Fully Automatic Image Analysis Framework for Cervical Vertebra in X-ray Images

S M Masudur Rahman AL ARIF

Department of Computer Science

City, University of London

A thesis submitted in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

City, University of London January 2018

To my wonderful parents who have raised me to be the person I am today, for their unwavering support, love, prayers and sacrifices.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 65,000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures. I grant powers of discretion to the City, University of London librarian to allow the dissertation to be copied in whole or in part without further reference to myself (the author). This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

S M Masudur Rahman AL ARIF

January 2018

Acknowledgements

In the name of the God, the most beneficent, the most merciful. Praise be to Him, who has enabled me to pursue this program and allowed me to complete my dissertation in due time with good health.

My profound and hearty gratitude is to my supervisor Greg Slabaugh, for giving me the opportunity to work on a topic of my passion and guiding me throughout my journey as a researcher. I have been incredibly blessed to have a supervisor who was always approachable, available, friendly and willing to help. His work ethic and dedication towards his responsibilities are exemplary, which I would like to replicate in my future endeavors. Special thanks to Karen Knapp, Michael Gundry, Andy Appelboam and Adam Reuben, without their time and help with the datasets and clinical expertise, this dissertation would not have been possible. I am grateful to Michael Phillips for his technical support.

I would also thank my examiners, Nasir Rajpoot and Tillman Weyde, for their time and interest in my work, and for their feedback which influenced the quality of this dissertation. Tillman Weyde has also examined my MPhil-PhD transfer, so has had a significant influence on my journey. I would also like to extend my gratitude to Artur Garcez, Constantino Carlos Reyes-Aldasoro and Vladimir Stankovic for their interest, advice, and suggestions that had a high impact on my research. Special thanks to the senior tutors for research, Evangelia Kalyvianaki, for chairing my MPhil-PhD transfer, and Radu Jianu for approving my transfer to write-up status and chairing my PhD oral examination.

This research would not have been possible without the three-year studentship from City, University of London. I would like to thank the university, graduate school and my supervisor for supporting me financially to attend several international conferences. My experience at City would not have been smooth and jolly without the help of the academic support staff. My heartiest gratitude goes to Naina Bloom, Nathalie Chatelain, Mark Firman, Paula Green, David Mallo-Ferrer, and Gill Smith, for their help on numerous occasions.

I have been fortunate to have worked alongside a group of enthusiastic researchers. I owe a special thanks to Muhammad Asad for being a good friend, a mentor and an advisor, for the last three years. Last but not the least, I gratefully acknowledge the valuable feedback on my dissertation from Atif Riaz, Rilwan Remilekun Basaru, Nathan Olliverre and Aamir Gulistan.

Abstract

Despite the advancement in imaging technologies, a fifth of the injuries in the cervical spine remain unnoticed in the X-ray radiological exam. About a two-third of the subjects with unnoticed injuries suffer tragic consequences. Based on the success of computer-aided systems in several medical image modalities to enhance clinical interpretation, we have proposed a fully automatic image analysis framework for cervical vertebrae in X-ray images. The framework takes an X-ray image as input and highlights different vertebral features at the output. To the best of our knowledge, this is the first fully automatic system in the literature for the analysis of the cervical vertebrae.

The complete framework has been built by cascading specialized modules, each of which addresses a specific computer vision problem. This dissertation explores data-driven supervised machine learning solutions to these problems. Given an input X-ray image, the first module localizes the spinal region. The second module predicts vertebral centers from the spinal region which are then used to generate vertebral image patches. These patches are then passed through machine learning modules that detect vertebral corners, highlight vertebral boundaries, segment vertebral body and predict vertebral shapes.

In the process of building the complete framework, we have proposed and compared different solutions to the problems addressed by each of the modules. A novel region-aware dense classification deep neural network has been proposed for the first module to address the spine localization problem. The proposed network outperformed the standard dense classification network and random forest-based methods.

Location of the vertebral centers and corners vary based on human interpretation and thus are better represented by probability maps than single points. To learn the mapping between the vertebral image patches and the probability maps, a novel neural network capable of predicting a spatially distributed probabilistic distribution has been proposed. The network achieved expert-level performance in localizing vertebral centers and outperform the Harris corner detector and Hough forest-based methods for corner localization. The proposed network has also shown its capability for detecting vertebral boundaries and produced visually better results than the dense classification network-based boundary detectors.

Segmentation of the vertebral body is a crucial part of the proposed framework. A new shape-aware loss function has been proposed for training a segmentation network to encourage prediction of vertebra-like structures. The segmentation performance improved significantly, however, the pixel-wise nature of proposed loss function was not able to constrain the predictions adequately. To solve the problem a novel neural network was proposed which predicts vertebral shapes and trains on a loss function defined in the shape space. The proposed shape predictor network was capable of learning better topological information about the vertebra than the shape-aware segmentation network.

The methods proposed in this dissertation have been trained and tested on a challenging dataset of X-ray images collected from medical emergency rooms. The proposed, first-of-its-kind, fully automatic framework produces state-of-the-art results both quantitatively and qualitatively.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

This dissertation explores a set of computer vision problems related to X-ray image analysis of cervical vertebrae and proposes a fully automatic framework to be used as a supporting tool for image interpretation by clinical experts. In this first chapter, we begin by addressing the motivation behind the need for an automatic framework for the analysis of cervical radiographs. The research objectives and questions are then stated, followed by a list of key original contributions proposed in this dissertation. We end this chapter with a list of articles published during the course of the research along with an outline of the remaining chapters.

1.1 Motivation

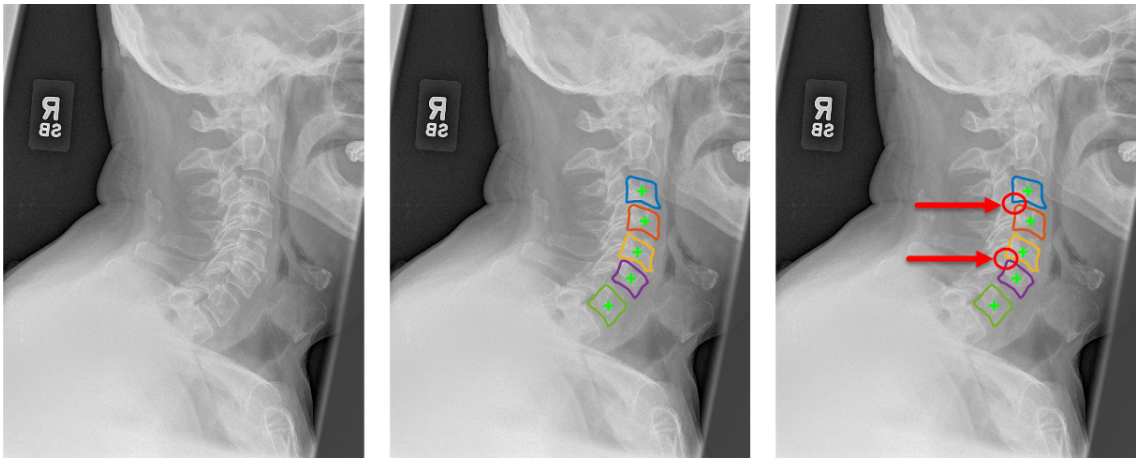
The cervical spine is a vital part of the human body, and due to its flexibility and position, is particularly vulnerable to trauma. Post-traumatic delayed or incorrect diagnosis can result in neurological deficit, paralysis or even death. Cervical spine injuries (CSIs) occur in a significant percentage of all trauma patients due to high energy impacts like automobile accidents, falls and dives into shallow water. Apart from these major accidents, minor injuries may also lead to CSI in elderly people and people with pre-existing bone abnormalities. About 43.9 to 61.5% of all spinal injuries occur in the cervical region, making it the most common injury-prone region of the whole spine [3]. Among different imaging techniques, X-ray is usually the first method of choice for the diagnosis of the cervical spine injuries in

the hospital emergency departments because of its quick results, low cost and availability. There are typically three views taken of the cervical spine: lateral view, anterior-posterior (AP) and odontoid process view. This dissertation focuses on the lateral view as it is most the informative and diagnostic for injury [4]. Despite standardisation and advances in imaging technologies, evaluation of a cervical spine X-ray image is a major radiological challenge for an emergency physician, particularly those with less experience. Failure to establish a correct diagnosis may result in death or serious disabilities. Clinical literature has reported up to 20% of CSI patients suffer tragic extension of their injuries due to delayed or missed diagnosis [5]. Early and accurate detection of CSI is critical to plan appropriate care and to prevent any tragic consequences. However, missed or delayed diagnosis of cervical spine injuries is still a common problem in hospital emergency departments. In one study [5], the most common cause (accounting for 44%) of missed cervical spine injuries was misinterpretation of the images. Another study [6] resulted in a similar number (47%) of missed or delayed diagnosis due to misinterpretation. Junior staff responsible for initial radiological examination failed to diagnose the injuries until experienced staff later performed a second evaluation of the radiographs. In [5], complications attributed to delayed or missed diagnosis ranged from motor and/or sensory neurologic deficits to complete quadriplegia. In other studies, 67% of patients with missed cervical fractures suffered neurological deterioration and nearly 30% of delayed CSI diagnosis developed permanent neurological deficit [7]. These numbers are alarming and the intention is to reduce these figures with the help of the state-of-the-art advances in computer vision algorithms.

Computer-aided diagnosis (CAD) systems have been used in clinical environment as a supporting tool for the experts for years. Most notably, CAD has been used for cancer detection in breast mammography [8, 9], radiography and computer tomography (CT) of lungs [10–12] and CT colonography [13, 14] with variable success rates. Other use of CAD systems includes detection of coronary artery diseases [15, 16], pathological brains [17, 18] and Alzheimer's diseases [19, 20]. Several studies reported that the inclusion of the CAD

system in the clinical environment has improved the diagnostic performance [14, 21, 22].

An overview of a conceptual computer-aided injury detection system for the lateral cervical X-ray image is shown in Fig. 1.1. From an input lateral cervical X-ray image, the system detects and highlights injuries to aid in clinical interpretation of the image by a physician.



(a) (b) (c)

Fig. 1.1 (a) An example cervical spine radiograph. This patient has retrolisthesis (displacement) of vertebra C3 onto C4 and C4 onto C5 (b) the conceptual injury detection system performs analysis of the image and predicts vertebral shapes (c) vertebral alignments are checked based on the predicted shapes, and possible location of abnormalities are highlighted to draw the radiologist's attention to the detected injury.

Keeping this overarching goal in the horizon, in this dissertation, **we limit our attention towards solving the computer vision aspects of the fully automatic injury detection system**. The evaluation of the proposed system's ability to improve human reading of images is beyond the scope of this dissertation. The performance of the algorithms proposed in this dissertation will be achieved in stand alone studies.

1.2 Research Question and Objectives

Many of the cervical spine injuries like vertebral displacement (retrolisthesis, and spondylolisthesis), spinal fusion, degenerative changes, osteoporosis, osteophytes and fractures (wedge, bi-concave and crush) can be detected by analyzing the size, shape, boundary and corners of the vertebrae. From a medical image computing perspective, the major challenge is to localize and detect different vertebral features in the image automatically. The main research question is 'Is it possible to develop a fully automatic image analysis framework for cervical vertebrae in X-ray images?'. The quest for a complete and fully automatic framework can be divided into several objectives:

1. Spine localization: Given an X-ray image, this algorithm will localize the spinal area in the image. We explore this objective in Chapter 3.
2. Center localization: Given the localized spinal column, this algorithm will be able to localize the vertebral centers. We explore the solution to this objective in Chapter 4.
3. Corner localization: Given the localized spinal column and centers, this algorithm will localize vertebral corners. We propose and compare different solutions to this algorithm in Chapter 5.
4. Vertebral boundary detection: Given localized vertebrae, this algorithm will detect vertebral boundaries. We discuss this problem in Chapter 6.
5. Vertebra segmentation: Given localized vertebrae, this algorithm will segment vertebral bodies. This algorithm has also been described in Chapter 6.
6. Vertebral shape prediction: Given localized vertebrae, this algorithm will predict vertebral shapes. The shape prediction is described in Chapter 7.

Once we can localize the spine, vertebral centers and corners, track vertebral boundaries, segment vertebral bodies and predict vertebral shapes, all these algorithms can be threaded together to build a complete and fully automatic image analysis framework which is reported in Chapter 8.

This dissertation explores data-driven machine learning-based solutions to the above-mentioned objectives. The models learn from a training dataset of images which have been annotated manually by clinical experts. Several random forest and deep learning-based models have been used, compared, investigated and innovated to build solutions for different vertebrae related computer vision problems in the objectives.

1.3 Original Contributions

The following are the key contributions of the work presented in this dissertation:

1. Region-aware deep convolutional neural network: A novel loss term has been included in the training of a deep convolutional neural network to encourage prediction of a single connected region. This region-aware network is used for the localization of the spinal region in X-ray images.

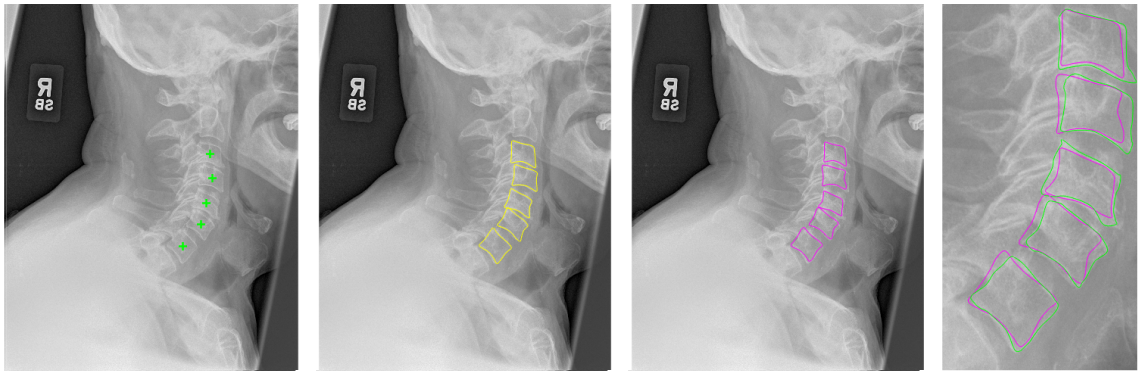
2. Shape-aware deep convolutional neural network: A shape-based loss term has been included in the training of a deep convolutional neural network to assist segmentation of vertebra-like shapes.

3. Deep spatial probabilistic regressor network: An innovative deep convolutional neural network is proposed for generating spatially distributed probabilistic maps. The proposed network has been used for vertebral center and corner localization, and also for vertebral boundary detection.

4. Deep spatial shape regressor network: A new convolutional neural network has been designed for prediction of vertebral shapes. The proposed network has been trained with a loss function defined in the shape space overcoming some of the limitations of the standard pixel-wise error-based loss function.

5. A fully automatic image analysis framework for cervical vertebrae: A first-of-its-kind fully automatic image analysis framework has been developed which is capable of taking an X-ray image as input and highlighting several vertebral features without any user intervention.

The work of this dissertation began with a vertebra segmentation framework that required vertebral centers to be clicked by the user at test time. The framework then used an active shape model to predict vertebral shapes. An example of this earlier framework is shown in Fig. 1.2.



(a) (b) (c) (d)

Fig. 1.2 Vertebra segmentation with manually clicked vertebral centers and active shape model (a) input image and manually clicked vertebral center points (+) (b) initialized active shape models on the vertebrae (—) (c) converged vertebral shapes (—) (d) converged vertebral shapes (—) with ground truth shapes (—).

The predicted shapes of this framework were sensitive to the variability of the manually clicked vertebral centers. Also, because of the complexity and sheer diversity in our dataset collected from real-life medical emergency rooms, the overall performance of this preliminary framework was poor. To lessen the effect of the manually clicked center points, we began our research by proposing two novel methods to localize the vertebral corners: a Harris corner detector-based naive Bayes approach and a Hough forest-based approach. However, the localized corners by these algorithms were not able to improve the overall performance of the framework. We then moved forward with our research to build a fully automatic framework by proposing a spine localization algorithm based on random classification forests. The algorithm was applied in a two-stage dense to coarse manner and able to localize the spine in a parallelogram box.

Deep learning has been at the center of the computer vision research since its outstanding performance in large-scale image classification challenge in 2012 [23]. However, the scarcity

of the training data was a roadblock for the deep learning methods to be applied to our problems. With time, our dataset of 90 images increased to a dataset of 296 images and using data augmentation techniques, we were able to use deep learning models on our problems. Based on the success of the fully convolutional neural networks in the literature, we proposed a novel spine localization algorithm using a region-aware deep fully convolutional neural network which is a key contribution in the present work. This algorithm outperformed our two-stage random classification forest-based spine localization algorithm.

After localizing spinal region robustly, we focused our attention to localize vertebral centers. We modified the fully convolutional network to generate a spatially distributed probability map indicating the location of the vertebral centers. The deep spatial probabilistic regressor network is a key contribution in this dissertation. We further improved the proposed deep spatial probabilistic regressor by introducing a novel loss function and a normalization layer. The improved spatial probabilistic regressor network was capable of localizing multiple image landmarks simultaneously. We applied this improved network to the corner localization problem which outperformed our previous Harris corner detector-based naive Bayes and Hough forest-based corner localization algorithms by a large margin. A final improvement to the spatial probabilistic regressor network was proposed by improving the normalization layer using a histogram-based approach. This network was applied on the vertebral boundary detection problem.

Another key contribution of this dissertation is a novel shape-aware deep vertebrae segmentation network. We proposed a novel shape-based loss term into the training of the segmentation network. The shape-aware network performed significantly better than the original segmentation network. However, the loss function was still a pixel-wise loss function where the segmentation results were not constrained into possible vertebra-like structures. This issue leads us to another key contribution presented in this work, a deep spatial shape regressor network. The proposed network is trained with a novel loss function defined in the

shape domain and predicts shapes directly instead of predicting segmentation masks.

After solving the spine localization, vertebral center and corner localization, vertebral boundary detection, segmentation and shape prediction problems, we combine these algorithms altogether in a seamless manner to build a complete and fully automatic image analysis framework for cervical vertebrae. This framework takes as input an X-ray image and highlights different vertebral features without any human input. An example of the fully automatic framework is shown in Fig. 1.3. For comparison, the same input image has been used in Fig. 1.2 where the earlier framework was illustrated.

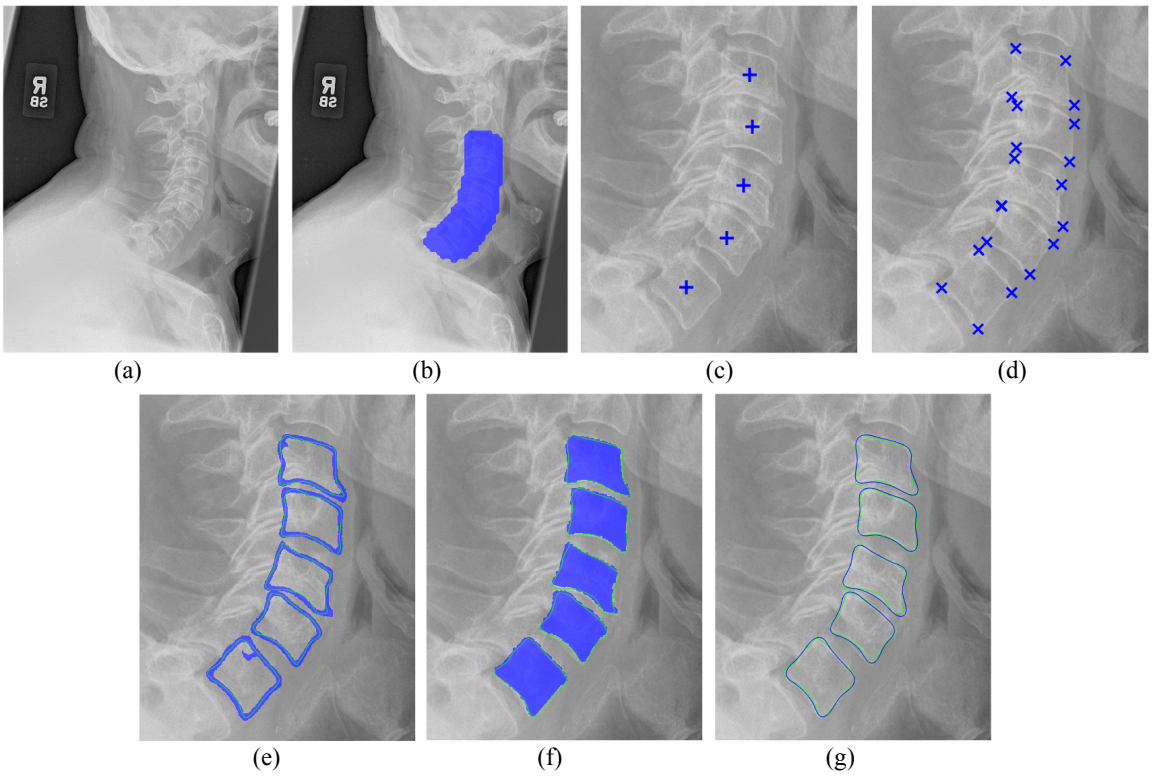


Fig. 1.3 Fully automatic vertebral image analysis framework (a) input image (b) localized spinal region (blue overlay) (c) localized vertebral centers (+) (d) localized vertebral corners (×) (d) predicted vertebral boundaries (blue overlay) with ground truth shape (−) (e) predicted segmentation masks (blue overlay) with ground truth shape (−) (f) predicted vertebral shapes (−) with ground truth shape (−).

1.4 List of Publications

The work in this dissertation is supported by articles published, or under review, in international workshops, conferences, and journals. Specifically, publications in Sec. 1.4.1-1.4.3 are directly related to this dissertation.

1.4.1 Journals

1. **S M Masudur Rahman Al-Arif**, Muhammad Asad, Karen Knapp, Micheal Gundry, and Greg Slabaugh. "Patch-based corner detection for cervical vertebrae in X-ray images." Elsevier Signal Processing: Image Communication, Volume 59, Page 27-36, November 2017.

2. **S M Masudur Rahman Al-Arif**, Karen Knapp, and Greg Slabaugh. "Fully automatic cervical vertebrae segmentation framework for X-ray images." Elsevier Computer Methods and Programs in Biomedicine. (https://doi.org/10.1016/j.cmpb.2018.01.006)

1.4.2 Conferences

1. **S M Masudur Rahman Al-Arif**, Muhammad Asad, Karen Knapp, Micheal Gundry, and Greg Slabaugh. Hough forest-based corner detection for cervical spine radiographs. In Proceedings of the 19th Conference on Medical Image Understanding and Analysis (MIUA), 2015, Lincoln, UK.

2. **S M Masudur Rahman Al-Arif**, Muhammad Asad, Karen Knapp, Micheal Gundry, and Greg Slabaugh. Cervical vertebral corner detection using Haar-like features and modified Hough forest. In Proceedings of the 5th International Conference on Image Processing Theory, Tools and Applications (IPTA), 2015, Orléans, France.

3. **S M Masudur Rahman Al-Arif**, Michael Gundry, Karen Knapp, and Greg Slabaugh. Global localization and orientation of the cervical spine in X-ray images. In Proceedings of the 4th International Workshop and Challenge In Computational Methods

and Clinical Applications for Spine Imaging (CSI), 2016, Held in Conjunction with MICCAI 2016, Athens, Greece. (**Best Paper Award**)

4. **S M Masudur Rahman Al-Arif**, Michael Gundry, Karen Knapp, and Greg Slabaugh. Improving an active shape model with random classification forest for segmentation of cervical vertebrae. In Proceedings of the 4th International Workshop and Challenge In Computational Methods and Clinical Applications for Spine Imaging (CSI), 2016, Held in Conjunction with MICCAI 2016, Athens, Greece.

5. **S M Masudur Rahman Al-Arif**, Karen Knapp, and Greg Slabaugh. Probabilistic Spatial Regression using a Deep Fully Convolutional Neural Network. In Proceedings of the British Machine Vision Conference (BMVC), 2017, London, UK.

6. **S M Masudur Rahman Al-Arif**, Karen Knapp, and Greg Slabaugh. Region-aware Deep Localization Framework for Cervical Vertebrae in X-Ray Images. In Proceedings of the workshop on Deep Learning in Medical Image Analysis (DLMIA), 2017, Held in Conjunction with MICCAI 2017, Quebec City, Canada.

7. **S M Masudur Rahman Al-Arif**, Karen Knapp, and Greg Slabaugh. Shape-aware Deep Convolutional Neural Network for Vertebrae Segmentation. In Proceedings of the workshop on Computational Methods & Clinical Applications in Musculoskeletal Imaging (MSKI), 2017, Held in Conjunction with MICCAI 2017, Quebec City, Canada.

1.4.3 Clinical Abstracts

1. Developing CSPINE CAD through machine learning algorithms: Inter-operator precision errors of user inputs, Watts V, Winzar C, Overington A, Rigby J, Gundry M, **Al-Arif SMMR**, Phillips M, Slabaugh G, Appelboam A, Reuben A, Knapp K. UKRC conference proceedings. P63, Liverpool, 29 June - 1 July, 2015.

2. Student radiographer perceptions of using CSPINE CAD software to assist cervical spine image interpretation and diagnosis, Watts V, Winzar C, Overington A, Rigby J, Gundry M, **Al-Arif SMMR**, Phillips M, Slabaugh G, Appelboam A, Reuben A,

Knapp K. UKRC conference proceedings. P108:P008, Liverpool, 29 June - 1 July, 2015.

3. Can CSPINE-CAD software increase diagnostic accuracy and confidence in c-spine imaging? Gundry M, Knapp K, Slabaugh G, Appelboam A, Reubens A, **Al-Arif SMMR**, Phillips M, UKRC conference proceedings. P186, Liverpool, 6 - 8 June, 2016.

1.4.4 Publications in Collaboration

1. Tim Albrecht, Gregory Slabaugh, Eduardo Alonso and **S M Masudur Rahman Al-Arif**. Deep Learning for Single-Molecule Science. Nanotechnology, Volume 28 (42), Page 423001, September 2017.

2. Atif Riaz, Muhammad Asad, **S M Masudur Rahman Al-Arif**, Eduardo Alonso, Danai Dima, Philip Corr and Greg Slabaugh. FCNet: A Convolutional Neural Network for Calculating Functional Connectivity from functional MRI. In Proceedings of the International Workshop on Connectomics in NeuroImaging (CNI), 2017, Held in Conjunction with MICCAI 2017, Quebec City, Canada.

3. Atif Riaz, Muhammad Asad, **S M Masudur Rahman Al-Arif**, Eduardo Alonso, Danai Dima, Philip Corr and Greg Slabaugh. Deep fMRI: An end-to-end deep network for classification of fMRI data. In Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI), 2018, Washington, D.C., USA.

1.5 Dissertation Outline

This dissertation is structured as followed: Chapter 2 provides a comprehensive literature review, an overview of the dataset used throughout the work and a brief discussion on some of the key concepts and algorithms. Chapter 3 deals with the spine localization problem and compares two proposed algorithms. The center localization method has been discussed in Chapter 4. Chapter 5 describes and compares three corner localization frameworks. Vertebral

boundary detection and segmentation problems have been addressed in Chapter 6. This is followed by the shape prediction framework in Chapter 7. Finally, in Chapter 8, all the algorithms are threaded together to build a complete and fully automatic image analysis framework for the cervical vertebrae. This leads to the conclusion of the dissertation in Chapter 9 where we discuss the limitations of the current framework, possible ways for improvements and direction towards future research on the topic.

Chapter 2

Background

This chapter is divided into four sections. We start by describing some of the key concepts related to the spine and the vertebrae. In the second section, we present a literature review by discussing the state of the research in related fields. The section ends with a table summarizing the most related articles. The dataset used for training and testing the proposed methods throughout this dissertation are reported in the next section highlighting the complexity and diversity of the images. We end this chapter by describing the initial framework from which the work of this dissertation evolved and with a brief introduction to the random forest and deep learning methods, both of which are extensively used in this dissertation.

2.1 Spine and Vertebrae

The vertebral column, or the spine, is a collection of bones that support the head and act as an attachment point for the ribs and muscles of the back and neck. An adult human vertebral column consists of 26 bones: the 24 vertebrae, the sacrum, and the coccyx bones [1]. The vertebrae are further divided into the seven cervical vertebrae, 12 thoracic vertebrae, and five lumbar vertebrae based on their position in the column (see Fig. 2.1 reproduced from [1]).

In this dissertation, we focus on the cervical vertebrae. As mention earlier, for X-rays, the cervical spine is scanned with three standard views: lateral, anterior-posterior (AP) and

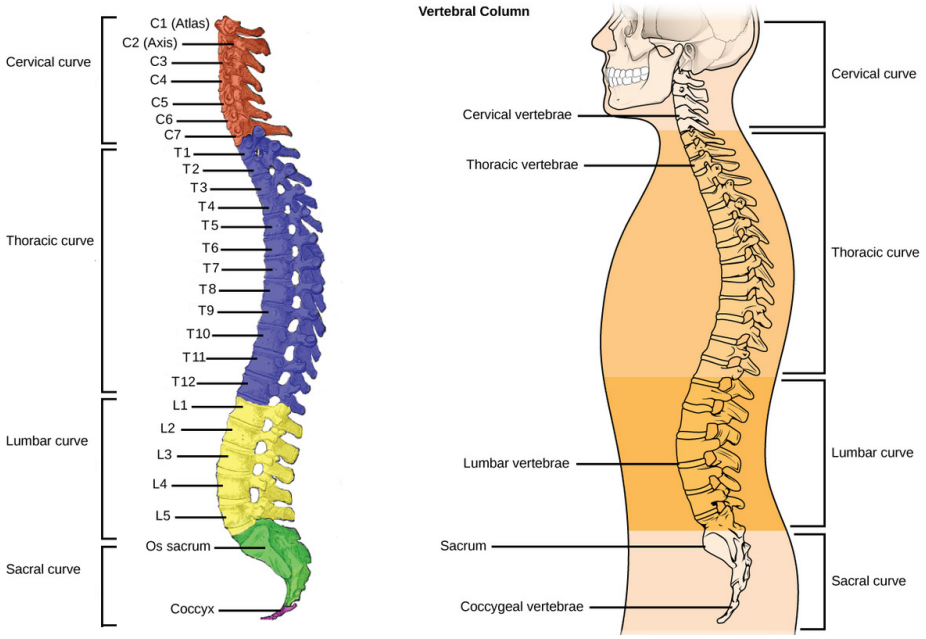


Fig. 2.1 Visualization of the vertebral column reproduced from [1].

odontoid process view [4]. Examples of these views are shown in Fig. 2.2. For general evaluation of the cervical spine, the lateral view is the most diagnostic. The other views are usually taken to focus on specific areas of the spine. For example, vertebra C1 and C2 overlap in the lateral view. Thus the odontoid peg view is appropriate to visualize these vertebrae. Similarly, the AP view is needed if specific focus on the anterior (right-side of Fig. 2.2a) and/or posterior (left-side of Fig. 2.2a) side is required. The work presented in this

(a) (b) (c)

Fig. 2.2 Standard views for cervical vertebrae (a) lateral (b) anterior-posterior (c) odontoid process.

dissertation deals with the most diagnostic lateral view of the cervical spine and vertebra C3-C7 (C1 and C2 are not considered because of their overlap, similar to other studies related to the cervical spine [24, 25]).

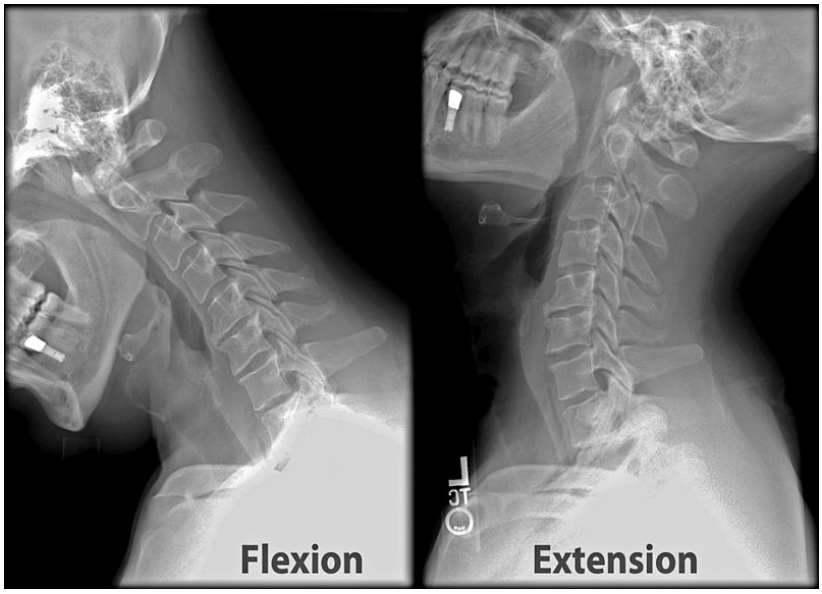


Fig. 2.3 Cervical spine at flexion and extension.

The cervical spine is a highly flexible anatomy, capable of flexion, extension, lateral flexion, and rotation [26]. Fig. 2.3 shows examples of lateral X-ray taken with the cervical spine in flexion and extension. Flexion is a movement by which subject's chin attempts to touch the chest whereas extension is a movement in the opposite direction. Lateral flexion is a similar movement but sideways, where the subject's ear tries to reach the shoulder. The lateral view can also be taken with subject's face rotated sideways.

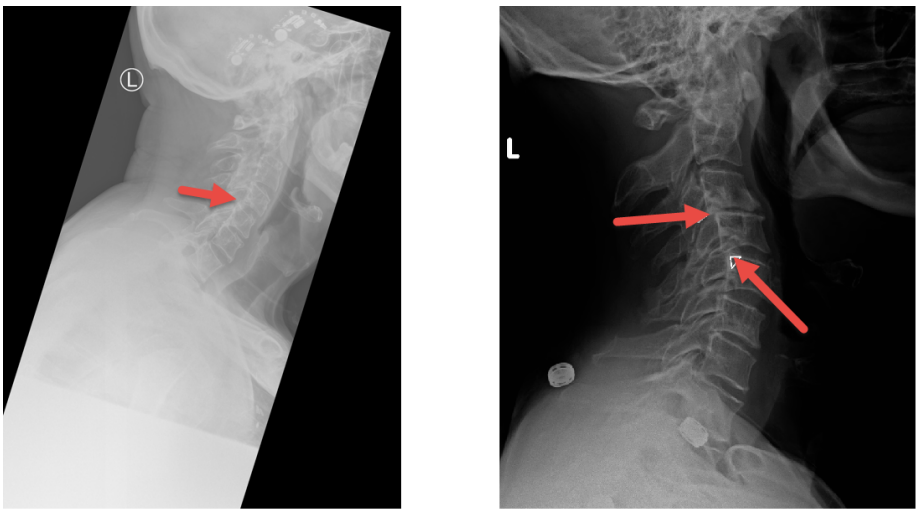
Due to this wide range of motion, the cervical spine is particularly vulnerable to injury. Automobile related injuries are the most common in the cervical spine. These injuries occur as the head and neck hit the dashboard, due to either being hit by another car or as the vehicle comes to a sudden stop. This causes either hyperflexion or hyperextension to the cervical spine. These mostly result in partial dislocation (subluxation) of the vertebral body. Diving head first into shallow water is another common cause of injuries in the cervical spine

resulting in compression injuries [27]. Apart from these, the cervical spine can also sustain injuries due to sudden distraction, rotational movement and age-related reasons. The next subsection describe some of the most common injuries related to the cervical spine.

2.1.1 Cervical Spine Injuries

2.1.1.1 Subluxation

Luxation is defined as the abnormal separation in the joint where two or more bones meet. A partial dislocation is referred to as a subluxation. In the cervical spine, subluxation of the vertebral body can occur on the anterior side or on the posterior side. The anterior displacement of one vertebral body on the adjacent vertebral body is termed as spondylolisthesis whereas the displacement in the posterior side is termed as retrolisthesis. Example of these subluxations are shown in Fig. 2.4.



(a) (b)

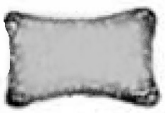
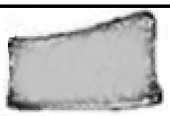
Fig. 2.4 Sublaxation injuries (a) spondylolisthesis (b) retrolisthesis.

2.1.1.2 Compression Fracture

The collapse of a vertebral body is identified as a compression fracture. A compression fracture is categorized based on the location of collapse. An anterior collapse is referred to

as a wedge fracture whereas a posterior collapse is called a crush fracture. A collapse of the vertebral body in the center is termed as a biconcave fracture. The severity of these fractures is often computed based on the anterior, medial and posterior heights of the vertebral body. A quantitative method, called Genant method, is widely used in clinical literature for the determination of the type and the severity of the compression fracture [28]. Different type of fractures are shown in Fig. 2.5.

(a)

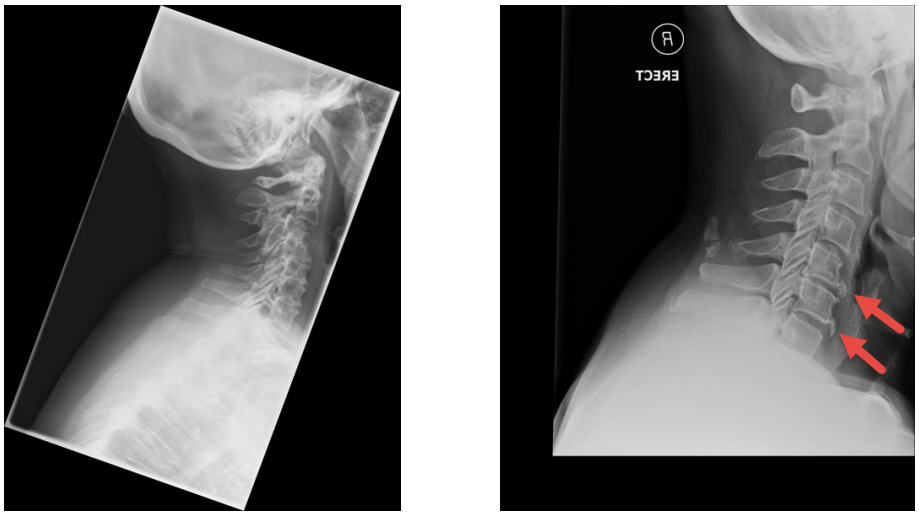
	Type of compression fracture		
Severity	Wedge	Biconcave	Crush
Grade 1			
□rad□2			
Grade 3			

(b)

Fig. 2.5 Vertebral fracture (a) normal vertebra (b) different types and grades of compression fractures.

2.1.1.3 Osteoporosis and Osteophytes

Vertebral osteoporosis is a condition characterized by gradual weakening vertebral bones, making them fragile. It develops over several years and usually not painful until a fracture occurs. Osteoporosis is often only diagnosed when a minor impact causes a fracture in the weakening bones. Fig. 2.6a shows an example of a cervical spine with osteoporosis. Osteophytes are another common anomaly in the cervical spine. Osteophytes are usually identified as bony projections that form along the vertebral boundaries. Both osteophytes and osteoporosis are sometimes categorized as degenerative changes of the spine and often related to ageing. An example of the cervical spine with osteophytes is shown in Fig. 2.6b.



(a) (b)

Fig. 2.6 Degenerative changes (a) osteoporosis (b) osteophytes.

In this subsection, we have presented a selection of clinical conditions most common to the cervical spine. The list is not exhaustive and there exist more complex and uncommon conditions. It should also be noted that the cervical spine injuries often come in a subtle form. Due to the difficulty in assessing the subtle injuries in X-ray images, a correct diagnosis can be delayed. Clinical literature suggests 61% of all cervical vertebra fractures and 36% of the subluxations remain unnoticed in X-rays [29].

2.2 Literature Review

The vertebral column is an important part of the human body. It can be imaged with different types of modalities. Two major types of image modalities used are radiographic imaging and magnetic resonance (MR) imaging. Radiographic imaging includes X-ray, computer tomography (CT) and dual energy X-ray absorptiometry (DXA). While MR and CT produce 3D volumetric scans, X-ray and DXA produce 2D images. Amongst the reviewed literature [24, 25, 30–42] work on X-ray images, [38, 43–49] use DXA scans, while CT and MR scans were investigated in [50–57] and [58], respectively. The literature diversely covers

different areas of the spine being studied i.e., cervical, thoracic and lumbar vertebra.

There are seven cervical vertebrae (C1-C7), 12 thoracic vertebrae (T1-T12) and five lumbar vertebrae (L1-L5). The column ends with the sacrum and the coccyx bone. Some of the literature works on the whole vertebrae column [37, 39, 50, 51], while most papers focus only on particular regions: cervical [24, 25, 30, 31, 33–36, 40, 41, 58], thoracolumbar [38, 42–44, 46, 48, 49, 52, 54, 56], lumbar [31, 32, 45, 47, 53, 59]. Our work is focused on five cervical vertebrae, C3-C7, in X-ray images. Like most of the reviewed articles concerning cervical vertebrae, C1 (atlas) and C2 (axis) are not considered in the current study because of their overlap in lateral X-ray images. These vertebrae are better visualized in 3D techniques like CT.

Different types of problems have been addressed in the reviewed literature. These objectives can be classified broadly into four classes: localization of the vertebral centers, end plates or spinal column [30, 38, 41, 42, 49, 51, 53, 54, 58], identification of the vertebral bone [39, 50, 51], segmentation [24, 25, 31, 32, 38, 40, 43–48, 50, 52, 55, 56] and fracture detection or morphometry [33, 36, 37, 43, 44, 48, 59]. The literature can also be categorized based on the type of methodology used. A few articles use non-data driven methods [34] while all other reviewed literature uses data-driven methods (statistical shape models (SSM), random forest (RF), AdaBoost, mean template etc.) which consists of an offline training phase and an online testing or prediction phase.

There is no gold standard dataset available publicly. Most of the research has been done on privately collected datasets. For X-ray images, a public dataset of cervical and lumbar vertebrae with manual segmentation, NHANES-II [60], has been used throughout the literature [24, 25, 30, 31, 33–36, 40]. This dataset consists of scanned analogue X-ray scans which often includes unnecessary artefacts and also missing information about the resolution of the data. These images were collected during second national health and nutrition examination survey (NHANES-II) conducted by the national center for health statistics (NCHS)

from 1976 to 1980 in the USA. As the dataset was not collected from hospital emergency departments, the images were not diverse in terms of injuries and other clinical conditions. Another publicly available CT dataset has been set up recently at SpineWeb [61, 62], which has been used in recent literature [52, 54, 56]. Our data is described in Sec. 2.3. The reviewed literature is summarized in Table 2.1.

Based on the reviewed literature, it can be understood that data-driven methods are more common than non-data driven methods. Among different methodologies, statistical shape model (SSM)-based methods (active shape model (ASM), active appearance model (AAM), deformable model (DM) and constrained local model (CLM)) have performed consistently well over different spinal regions and image modalities. As stated earlier, our goal in this dissertation is to solve different computer vision problems like localization of spinal region, localization of vertebral centers, corners and vertebral boundary detection, segmentation and shape prediction for cervical vertebrae in X-ray images. Concentrating on these objectives and selecting the work on 2D radiographic images, most of the related literature comes mainly from two groups: Benjelloun et al. [24, 25, 34, 35, 41, 42] and Cootes et al. [38, 44–49]. The first group works with the NHANES-II dataset of cervical X-ray images while the second group works on their own dataset of DXA images of the whole spine.

Earlier work of Benjelloun et al. [34, 35] address vertebral boundary detection and region selection using a polar signature and template matching, respectively. They implemented a semi-automatic segmentation framework based on ASM in [25, 49]. Our initial framework, described in the Sec 2.4, is a simplified version of this work. Their latest work on X-ray images [42] uses generalized Hough transform (GHT) to identify cervical vertebrae.

From the latter group, automatic segmentation and morphometry computation of vertebra on DXA images have been addressed by Robert et al. in [44–46] using an AAM. An improved segmentation is obtained with a part-based graph with AAM in [47]. Their method also showed vertebral fracture detection capability in [48]. AAM has been improved by Random

Reference No	Modality	Vertebra Region	Objective	Key Methodology	Dataset
Smyth 1999 [43]	DXA	Thoraco-Lumbar	Segmentation Morphometry	ASM	Own
Tezmol 2002 [30]	X-Ray	Cervical	Localization	Hough Transform (HT) Template Matching	NHANES-II
Zamora 2003 [31]	X-ray	Cervical Lumbar	Segmentation	GHT, ASM, DM	NHANES-II
Bruijne 2004 [32]	X-ray	Lumbar	Segmentation	k-NN classification PDM Particle filtering	Own
Chamarathy 2004 [33]	X-Ray	Cervical	Morphometry	K-means SOMs	NHANES-II
Roberts 2005 [44]	DXA	Thoraco-Lumbar	Segmentation Morphometry	AAM	Own-Cootes
Roberts 2006 [45]	DXA	Lumbar	Segmentation	AAM	Own-Cootes
Roberts 2006 [46]	DXA	Thoraco-Lumbar	Segmentation	AAM	Own-Cootes
Benjelloun 2006 [34]	X-ray	Cervical	Edge detection	Polar signature (NDD)	NHANES-II
Benjelloun 2006 [35]	X-ray	Cervical	Region selection	Template matching	NHANES-II
Aouache 2007 [36]	X-Ray	Cervical	Morphometry	ASM	NHANES-II
Casciaro 2007 [37]	X-Ray	Whole	Morphometry	Local Phase symmetry	Own
Bruijne 2007 [59]	X-Ray	Lumbar	Morphometry	Conditional ASM	Own
Roberts 2009 [47]	DXA	Lumbar	Segmentation	AAM, Part-based Graph	Own-Cootes
Klinder 2009 [50]	CT	Whole	Identification Segmentation	Generalized HT (GHT)	Own
Roberts 2010 [48]	DXA	Thoraco-Lumbar	Segmentation Morphometry	AAM	Own-Cootes
Mahmoudi 2010 [25]	X-Ray	Cervical	Segmentation	ASM	NHANES-II
Dong 2010 [39]	X-Ray	Whole	Identification	Probabilistic Graphical Model	Own
Benjelloun 2011 [24]	X-Ray	Cervical	Segmentation	ASM	NHANES-II
Xu 2012 [40]	X-Ray	Cervical	Segmentation	AAM	NHANES-II
Glocker 2012 [51]	CT	Whole	Localization Identification	Random regression forest Hidden Markov Model	Own
Larhmam 2012 [41]	X-Ray	Cervical	Localization	GHT	Own
Roberts 2012 [49]	DXA	Thoraco-Lumber	Localization	Regression forest, AAM	Own-Cootes
Larhmam 2014 [42]	X-Ray	Cervical	Localization	GHT, K-means	Own
Larhmam 2014 [58]	MR	Thoraco-Lumber	Localization	Ellipse fitting, curve detection	own
MICCAI 2014 [52]	CT	Thoraco-Lumber	Segmentaion	M1: Atlas-based M2,3,4,5:Statistical Shape Model	SpineWeb
Bromiley 2015 [38]	DXA	Thoraco-Lumber	Localization Segmentation	CLM with RFRV	Own-Cootes
Ibragimov 2015 [53]	CT	Lumber	Localization Rough Segmentation	Interpolation-based detection mean shape model	Own
Korez 2015 [54]	CT	Thoraco-Lumber	Localization Rough Segmentation	Interpolation-based detection Mean shape model	SpineWeb, Own
Korez 2015 [55]	CT	Lumber	Segmentation	Shape Constrained DM	Own
Korez 2015 [56]	CT	Thoraco-Lumber	Segmentation	Shape Constrained DM	SpineWeb
Embrahimi 2016 [63]	X-ray	Lumber	Localization	Corner detection using Haar-based features	Own
Bromiley 2016 [57]	CT	Thoraco-Lumber	Localization Segmentation	CLM with RFRV	Own-Cootes
Mehmood 2017 [64]	X-ray	Cervical	Localization	GHT, Fuzzy c-means	NHANES-II
Yang 2017 [65]	CT	Whole	Identification	Convolutional Neural Network	SpineWeb, Own

Table 2.1 Literature review.

Forest Regression Voting (RFRV) in [49]. Along with geometric constraints, AAM-RFRV performed better than the original AAM. Bromiley et al. [38, 57] is the latest work by their group where constrained local model (CLM), another improved version of SSM, is used with

RFRV. This method improved the segmentation accuracy for fractured vertebrae. [38] and [49] can be considered as the state-of-the-art in the field of vertebra segmentation on 2D radiographic images.

2.3 The Dataset

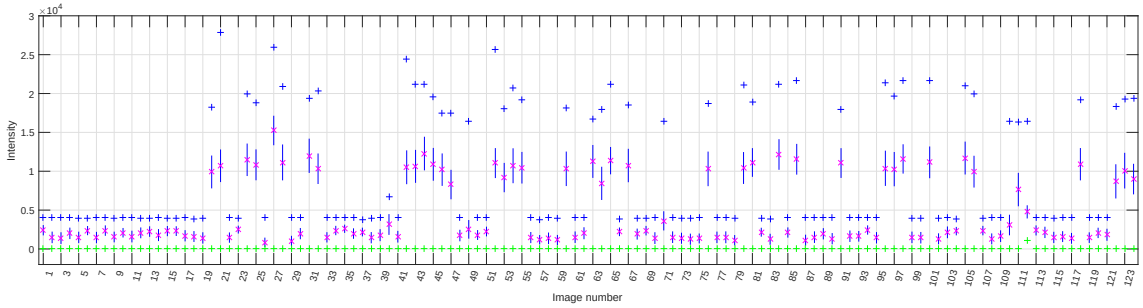
The data used in this dissertation comes from the 'Computer-Aided Detection of Cervical Spine Injuries: A Feasibility Project' grant funded by the EPSRC [66]. The images were received sequentially in stages. The images were provided by Royal Devon & Exeter Hospital in association with the University of Exeter. The first instalment of 138 images was received in late 2014. These images were scanned in the emergency rooms of the Royal Devon & Exeter Hospital from March to April of 2014. The work included in our initial publications [67–71] used a subset of 90 images from the initial 138 images. This dataset will be referred to as 'Dataset A' in the following chapters. The selection of these 90 images was done manually and has been described in Appendix A. The performance of the methods proposed in our initial set of publications was evaluated in a ten-fold cross-validation manner. As our methods developed from shallow models to deeper models requiring longer training time, doing ten-fold cross validation became infeasible. To train our first deep model (not reported in this dissertation), we used 124 randomly chosen images with data augmentation and the remaining 14 images were used as a validation set to check the over-fitting during training epochs.

In mid-2016, we received the second instalment of 158 images which were scanned in the same hospital from May 2014 to August 2015. For the research reported in this dissertation, we have used the randomly chosen 124 images from the first instalment as the training dataset and 172 images consisting of a randomly chosen 14 images from the first instalment and all the images from the second instalment as the test dataset.

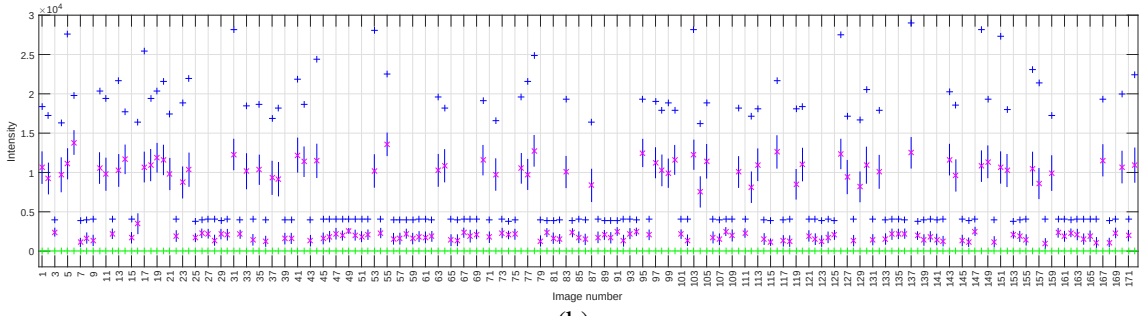
A final instalment of 40 images were received at the end of 2016. These images were used as a validation set in some of our experiments.

The data was collected from the subjects who have visited the emergency department of Royal Devon and Exeter hospital [66]. Each image in the dataset was de-identified of the personal information except the subject's age and gender. To make sure that the data collection complied with the confidentiality and personalization issues, approvals were taken from the ethics committee of the College of Engineering, Mathematics and Physical Sciences from University of Exeter, and a research committee of the National Health Service (NHS).

The dataset offers a range of challenging and practical issues. It contains normal, good contrast images to very low contrast, abnormal images. Fig. 2.7 shows a summary of the intensity variations of the images in the dataset. The images are recorded in 16-bit unsigned integer format thus the possible value of a pixel is limited from 0 to 65,535. Most of the

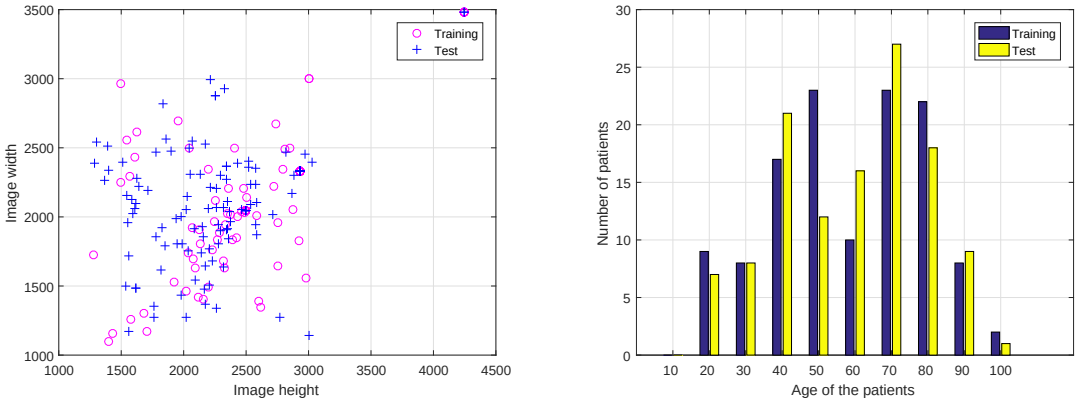


(a)



(b)

Fig. 2.7 Intensity variation in the (a) training and (b) test dataset: Maximum intensity (+), minimum intensity (+), mean intensity (×), length of the vertical blue line indicates the standard deviation of the intensity distribution per image.

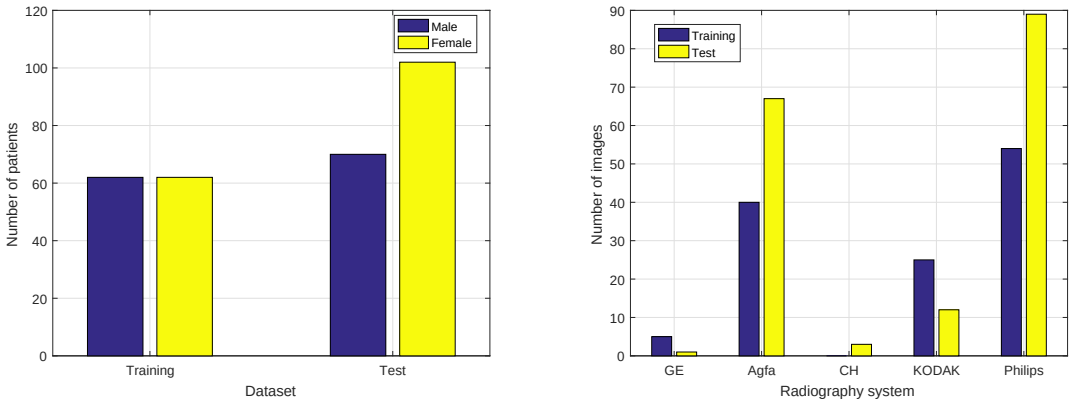


(a) (b)

Fig. 2.8 (a) Distribution of image resolution in the dataset (b) variation of patient age in the dataset.

images used only a part of this available range: from 0 to 5,000. However, some of the images had the upper range as high as 30,000. It can also be seen that the mean intensity and the standard deviation in each image also vary greatly, making the dataset very challenging.

Apart from the intensity variation, the heights and widths of the images are also diverse in our dataset. Fig. 2.8a shows the distribution of image sizes in the training and test dataset. The pixel spacing of the images varies from 0.1 to 0.194 millimeter per pixel. The age of the



(a) (b)

Fig. 2.9 (a) Variation of patient sex in the dataset (b) Radiography systems used for X-ray image acquisition.

patients varied from 17 to 96. The distribution of patient age and sex in the training and test dataset are reported in Fig. 2.8b and 2.9a, respectively. Radiographic systems used for the acquisition of the images come from five different manufacturers: Philips, Agfa, KODAK, General Electric (GE) and Carestream Healthcare (CH). Fig. 2.9b shows a bar plot of the number of images taken by each system.

Diversity in our dataset also comes in the form of patient view (left, right), patient position (standing, sitting, lying), spine orientation (flexion, extension) and clinical conditions (osteophytes, osteoporosis, degenerative changes, bone loss, fracture, bone implant etc.). We described some of these clinical conditions in Sec. 2.1. All the images were flipped and rotated accordingly so that the posterior side was on the left side of the image. Since our data is collected from the patients who had visited an emergency department, the majority of the

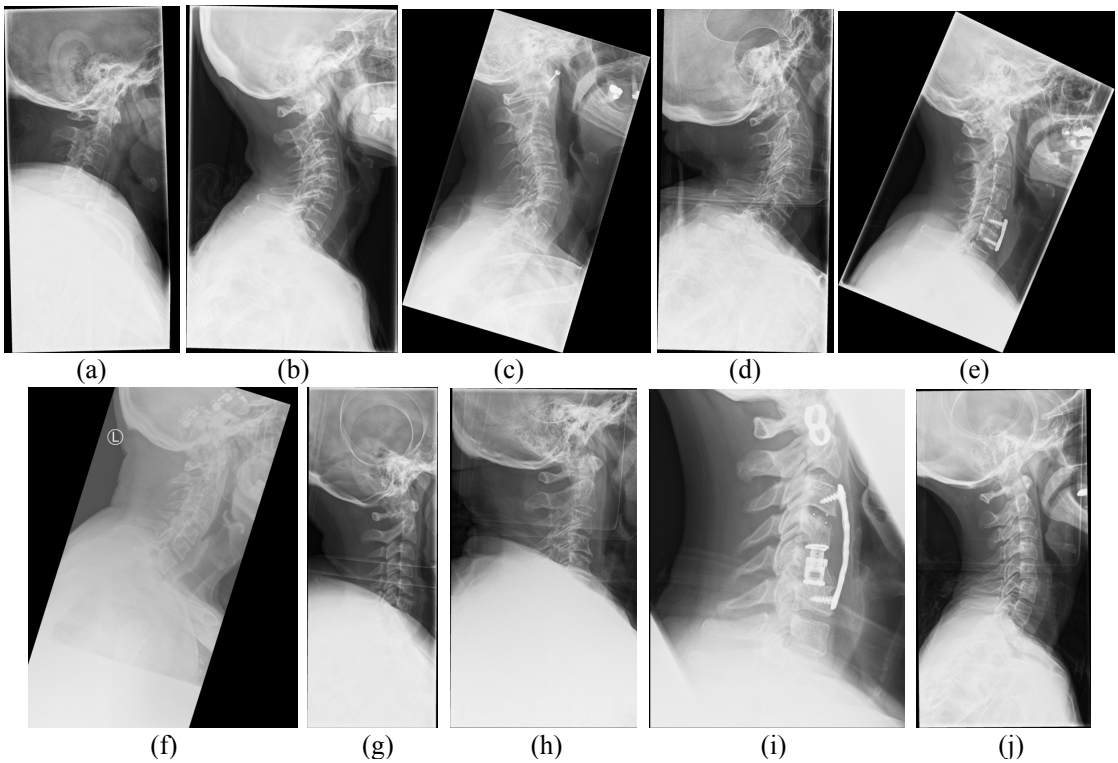


Fig. 2.10 Examples of images in the dataset (a) bone loss (b) osteophytes (c) degenerative changes (d) retrolisthesis (e) surgical implant (f) spondylolisthesis (g) image artefacts (h) compression fracture (i) surgical implant (j) retrolisthesis.

images contain clinical conditions. Fig. 2.10 illustrates some of the diversity in our training and test dataset.

2.3.1 Manual Annotation

The methods discussed in the dissertation are data-driven machine learning methods. We explore only supervised machine learning techniques which require input-output data pairs during training. These data pairs are also needed for evaluation purposes at the test time.

As stated earlier, our work is focused on five cervical vertebrae, C3 to C7. Our medical partners have provided us with the manual demarcation of the vertebral boundaries. Each of the vertebrae in the dataset was manually annotated by expert radiographers using a MATLAB graphical user interface. Each vertebra was annotated by 20 points along the vertebral boundaries with one point for the center. Four of the boundary points indicate the corners. The manual demarcation points for a few vertebrae are shown in Fig. 2.11. These sparsely annotated vertebral boundaries can be converted into a continuous curve by using a Catmull-Rom spline [72]. The blue lines in Fig. 2.11 represent the splined or interpolated vertebral boundary. The number of manually clicked boundary points per vertebrae i.e. 20 is chosen based on [24] so that it can represent the vertebral boundary curvature accurately. More manually clicked points would significantly increase the amount of human work needed for manual annotation whereas fewer points would make the boundary inconsistent when interpolated. As it can be seen in Fig. 2.11, using 20 manually clicked points, the interpolated continuous curve is able to follow the natural vertebral curvature accurately.

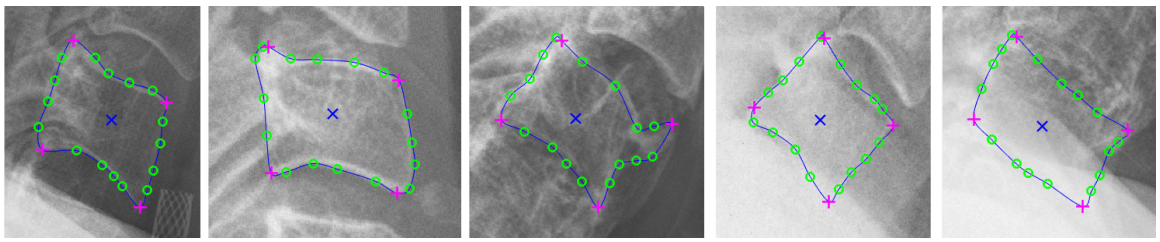


Fig. 2.11 Manual segmentation: manually demarcated center ($\times$), corner ($+$) and boundary (o) points. The blue curve ($-$) represents the splined vertebral boundary.

2.4 Initial Framework

The work of this dissertation evolved from a semi-automatic shape predictor framework. This framework is not a contribution of this dissertation, however, it serves as a starting point for the work to be presented. The framework uses an active shape model (ASM) to capture the shape variations within the training set. The ASM produces a mean shape, eigenvalues and eigenvectors which are also known as modes of variation. At the test time, using manually clicked vertebral centers, an approximation of the vertebra size and rotation is computed and mean shape is initialized on the vertebrae based on this information. Then using an iterative procedure, called ASM search, the initialized shape converges on the actual vertebral boundaries. In the following subsections, we describe the training and the test time procedures of this framework. This framework is designed based on the work of Benjelloun et al. [24, 25].

2.4.1 ASM Training

Active shape model (ASM) is a statistical model which captures the variation of a group of similar shapes defined by a set of points. In our case, vertebra shapes are defined by a set of 20 points. It can be seen in Fig. 2.11 that the spacings between the manually annotated points are not uniform which can add additional variation to the model. First, these manually induced variations are removed by reconfiguring the points between two consecutive corners with equal spacing with the help of Catmull-Rom spline interpolation [72]. Boundary points before and after this process can be seen in Fig. 2.12. Then, bearing in mind that different vertebra (C3-C7) may have different shape variation, an individual ASM is built per vertebra.

Let $\boldsymbol{x}_i$ be a vector of length $2m$ describing m 2D points of the i-th training vertebra, given by:

$$\boldsymbol{x}_i = [x_{i1}, y_{i1}, x_{i2}, y_{i2}, x_{i3}, y_{i3}, ..., x_{im}, y_{im}], \tag{2.1}$$

where (x_{ij}, y_{ij}) is the Cartesian coordinate of the j-th point of the i-th training vertebra and $m = 20$ for our case.

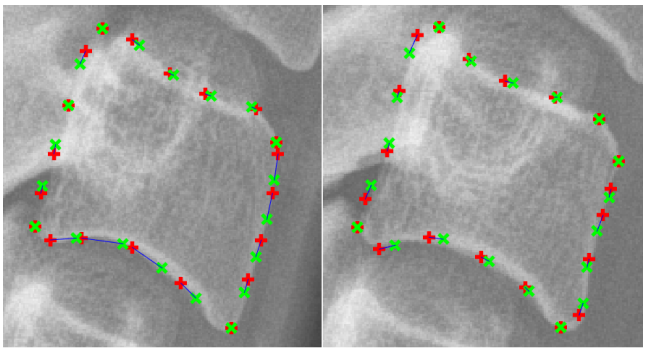


Fig. 2.12 Equally spaced reconfiguration of manually clicked points: original points (+) and reconfigured points (×).

All training vertebral shapes are then aligned/registered using Procrustes registration [73] to remove variability due to translation, scaling and rotation. Fig. 2.13 shows an example of Procrustes registration for two quadrilateral shapes. Aligned and non-aligned vertebral shapes are plotted in Fig. 2.14. Once the example vertebral shapes are aligned, a principal component analysis (PCA) is applied [74]. This allows any shape, $\boldsymbol{x}_i$ to be represented by a mean shape $\bar{\boldsymbol{x}}$, eigenvectors $\boldsymbol{p}_k(k = 1,2,...,2m)$ and corresponding shape parameters $\boldsymbol{b}$:

$$\boldsymbol{x}_i = \bar{\boldsymbol{x}} + \boldsymbol{P}_s \boldsymbol{b}_i; \boldsymbol{P}_s = [\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_l], \tag{2.2}$$

where $\boldsymbol{P}_s$ is the matrix consisting of the first l eigenvectors. The standard practice to select l is to find the first few eigenvalues that represent a percentage of the total variation in the training set. Now, for any known shape from the training data, $\boldsymbol{x}_i$, shape parameter $\boldsymbol{b}_i$ can be

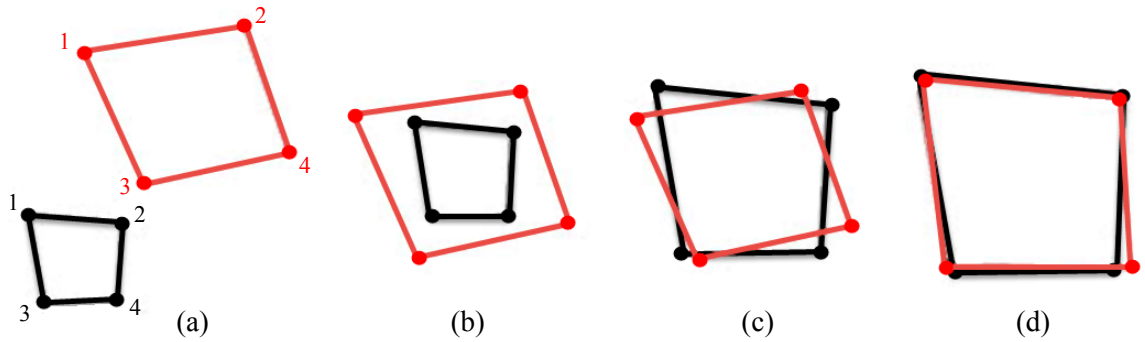
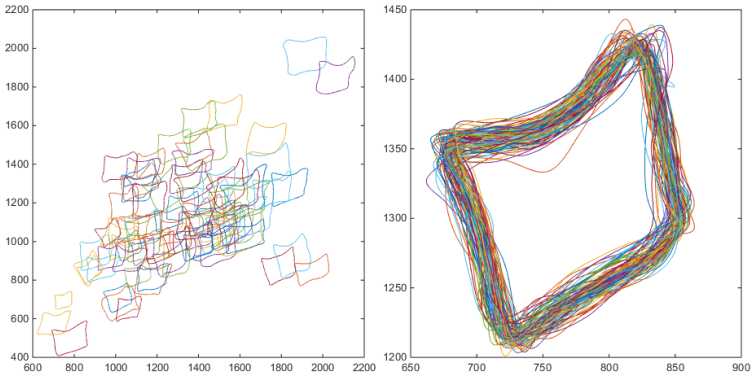


Fig. 2.13 Procrustes registration (a) unregistered shapes (b) centered (translation) (c) centered and scaled (d) centered, scaled and rotated (registered shapes) [2].



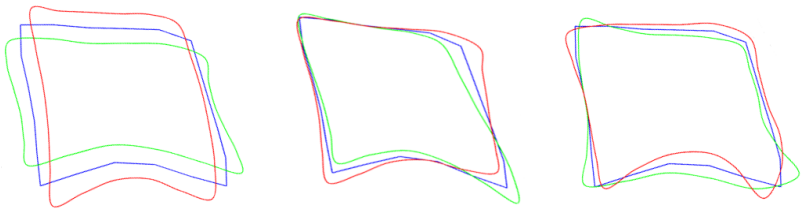
(a) (b)

Fig. 2.14 (a) Unregistered and (b) registered vertebral shapes for vertebra C3.

computed as:

$$\boldsymbol{b}_i = \boldsymbol{P}_s^T (\boldsymbol{x}_i - \bar{\boldsymbol{x}}). \tag{2.3}$$

Eqn. 2.3 establishes the relationship between a shape ($\boldsymbol{x}_i$) and its shape parameters ($\boldsymbol{b}_i$). For an unknown shape from the test dataset, $\boldsymbol{b}_i$ needs to be calculated through a procedure known as ASM search which is described in the next subsection.



(a) (b) (c)

Fig. 2.15 The blue shape represents the mean shape. The green and the red shapes represent variation in the positive and negative direction for the (a) first, (b) second and (c) third modes of variation.

2.4.2 ASM Search

At the test time, the mean shape first has to be initialized near the test vertebrae. The vertebra sizes in pixels vary considerably among images due to the difference in spatial resolution of the images. The size also varies in millimetres from patient to patient because of natural variation amongst the human population. Thus, it is more appropriate to compute the vertebra

orientation and size individually. This is where the user inputs are required, making the process semi-automatic. The user is asked to localize the vertebral centers. The orientation and size of the vertebrae are computed using these center points. For each vertebra, a vector is drawn from its center to the center of the vertebra above ($\boldsymbol{F}_u$) and below ($\boldsymbol{F}_d$). Then the orientation vector, $\bar{\boldsymbol{F}}$, can be computed as the average of these vectors.

$$\boldsymbol{F} = \frac{\boldsymbol{F}_u - \boldsymbol{F}_d}{2} \tag{2.4}$$

In case of the top vertebra, $\boldsymbol{F} = -\boldsymbol{F}_d$ and for the bottom vertebra, $\boldsymbol{F} = \boldsymbol{F}_u$. The magnitude of the vector $\boldsymbol{F}$ represents the coarse size of the vertebra. The vectors are visualized in Fig. 2.16.

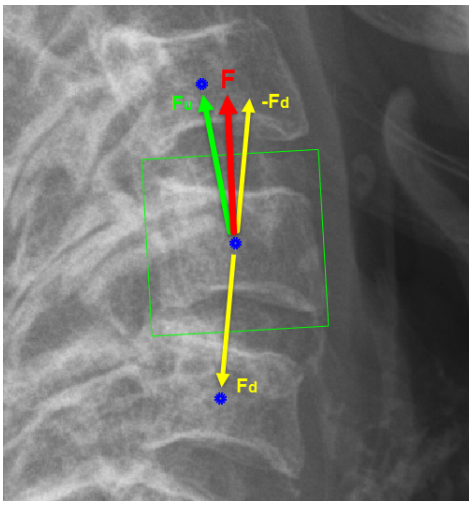


Fig. 2.16 Computation of the orientation vector, $\boldsymbol{F}$. Vertebrae centers (o). The green box approximately represents the size and orientation of the vertebrae.

Using the orientation and magnitude of the vector $\boldsymbol{F}$, the mean shape is initialized on the current vertebra, this mean shape is a vector of m (= 20) 2D points like Eqn. 2.1. Once the mean shape is initialized, normal profiles are extracted at each point (green lines in Fig. 2.17). Based on the gradient of each profile the edge is located at the maxima. Based on the current location of the point, the 2D offset to the maxima (dx_{ij}, dy_{ij}) is returned for each point j.

$$d\boldsymbol{X}_i = [dx_{i1}, dy_{i1}, dx_{i2}, dy_{i2}, dx_{i3}, dy_{i3}, ..., dx_{i20}, dy_{i20}], \tag{2.5}$$

$$d\boldsymbol{b}_i = \boldsymbol{P}_s d\boldsymbol{X}, \tag{2.6}$$

$$\boldsymbol{b}_i = \boldsymbol{b}_i + \boldsymbol{W} d\boldsymbol{b}_i, \tag{2.7}$$

where $\boldsymbol{b}_i$ is initialized as an all zero vector (this corresponds to the mean shape) and $\boldsymbol{W}$ is a diagonal matrix of weights, one for each mode. This can be identity, or each weight can be proportional to the standard deviation of the corresponding shape parameter over the training set. In our case, the identity matrix has been used. Once $\boldsymbol{b}_i$ is known, the shape model can be updated by Eqn. 2.2. This process is repeated until a maximum iteration threshold is reached or $d\boldsymbol{X}$ is negligible. Essentially this process converges to a state where the shape fits best. Fig. 1.2 in Chapter 1 shows an example of the performance of this framework. Due to its dependence on the manually clicked vertebrae centers for the initialization of the mean shapes and gradients of image intensity distributions for the shape convergence, this framework lacks robustness on our dataset containing challenging images. Novel approaches are proposed in this dissertation to build a fully automatic and robust image analysis framework for the cervical vertebrae.

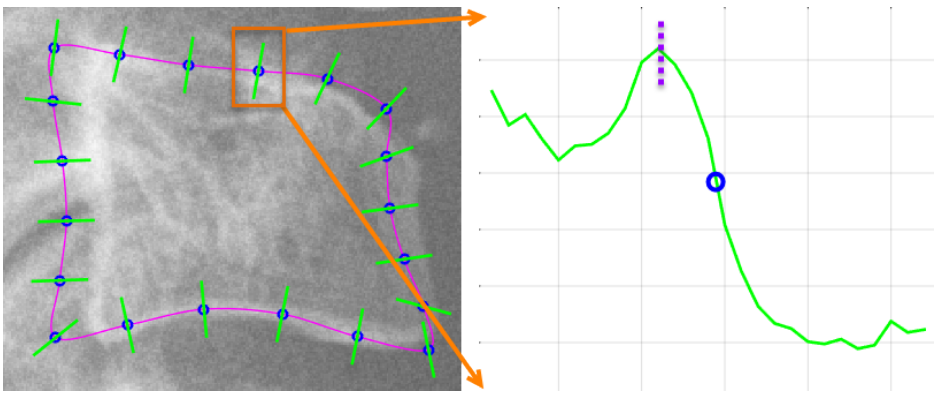


Fig. 2.17 ASM search: on the left image, the mean shape is shown in magenta, points on the mean shape are shown as blue circles and the normal profiles are shown with green lines. An example of intensity profile in the normal direction is shown on the right, with a dotted line demarcating a possible edge.

2.5 Machine Learning

The term 'machine learning' refers to a set of algorithms that allows the computers to learn from examples without being explicitly programmed. The examples, from which these algorithms learn, are provided through training data points. In Sec. 2.2, we have shown that most of the related literature in the field involves data-driven methods. These data-driven methods utilize different forms of machine learning techniques.

Machine learning algorithms can be divided broadly into two groups: supervised and unsupervised. This categorization depends on the nature of the training data. If the input training data points have known output values, the data is considered as labeled data. When the output variables of the labeled data points are used in the learning process, the algorithm can be classified as a supervised machine learning technique. On the other hand, unsupervised machine learning algorithms deal with the data where the output values or the labels are unavailable. There also exists a set of algorithms which falls in between this two categories. These algorithms can be classified as semi-supervised machine learning techniques. Semi-supervised algorithms learn from a training dataset where the output values are known for some samples and not known for other samples. In this dissertation, the manual annotation of the vertebrae works as target output values allowing us to use the supervised machine learning models.

Supervised learning can further be divided into classification and regression based on the nature of the output variables. If the output variable is discrete, it is called a classification problem. In case of continuous output variable, the problem is identified as regression. Both classification and regression problems are present in this dissertation. These problems are addressed using two of the most used machine learning algorithms in the literature: random forest and deep learning. Random forest algorithm has been used in Chapter 3 and 5, for spine localization and corner localization, respectively. Different deep learning-based methods are proposed, compared and investigated in Chapter 3, 5, 6 and 7. More details about both of these algorithms can be found in Appendix B.

Chapter 3

Spine Localization

This chapter explores global localization of the cervical spine in the X-ray images. Our aim here is to find the position and the orientation of the cervical spine in an X-ray image. In this chapter, we propose two algorithms to solve the localization problem. Given the fact that the cervical spine consists of several cervical vertebrae, our first algorithm looks for the vertebrae in the image using a sliding-window technique. This patch-based approach utilizes random classification forest and kernel density estimation. This algorithm has two-stages and localizes the spine in a coarse-to-fine manner within a bounding parallelogram. This algorithm has two drawbacks. First, the patch-based sliding-window technique often fails to take into account the global positioning of the cervical spine in the image because it only sees a small patch at a time. Second, the rigid output bounding parallelogram is not capable of capturing the flexibility of the cervical spine. To address these issues, a second algorithm is proposed. This method is a deep learning-based approach where we have converted the localization problem into a dense classification problem at a coarse resolution. Three fully convolutional networks have been utilized and compared. A novel region-aware loss term has been proposed which significantly improved the localization performance of all three networks. This algorithm can localize the spine with arbitrary shapes rather than a bounding parallelogram.

Global localization of the spine is a less studied area in the literature. Most of the articles use a manual or a semi-automatic way to reduce the search area in the image and then proceed with identification and/or segmentation problems [24, 32, 38, 44, 47]. However, a few methods have been presented in the literature for localization of the cervical spine in X-ray images. Most of these methods revolve around the generalized Hough transform (GHT) and random forest. Tezmol et al. [30] used a GHT-based framework using mean vertebra templates and an innovative voting accumulator structure. A more recent work proposed another template matching-based approach relying on the GHT which involves a training phase [41]. Glocker et al. presented a random regression forest-based localization and identification framework for vertebrae in arbitrary CT scans [51]. They proposed another framework using random classification forest which has shown better performance in the localization and identification of the vertebrae with pathological cases [75].

The contributions of this chapter are:

1. Two state-of-the-art algorithms for global localization of the cervical spine.
2. A comparison of three dense classification networks.
3. A novel region-aware loss function which significantly improved the localization ability of all three dense classification networks.
4. A major step towards the realization of a fully automatic image analysis framework for cervical vertebrae.

3.1 Spine Localization using Random Forest

Random forest is a popular machine learning algorithm [76]. It has been used in many medical image computing literature focused on vertebrae [38, 49, 51, 67, 68, 75]. Like the localization and identification work of [51, 75], our work also uses the random forest machine learning algorithm. But instead of localizing and identifying each vertebra, it finds the position and orientation of the vertebral column in lateral cervical X-ray images.

3.1.1 Overview

The main component of this framework is a random classification forest trained to distinguish between the vertebra and non-vertebra patches extracted from the images. The task is designed as a binary classification problem. The framework employs a two-stage coarse-to-fine approach. In the first coarse localization stage, a sliding window sparsely scans a test image to vote for vertebra patches. After this sparse voting, an accumulation phase converts the votes into a bounding parallelogram which indicates the position of the spinal column inside the image. The fine localization stage scans the resultant bounding parallelogram of the first stage densely with different patch sizes and orientations. The same voting accumulation phase is applied again and a refined bounding parallelogram is generated. This first stage limits the search area for the time consuming second stage and reduces the overall computation time for the algorithm.

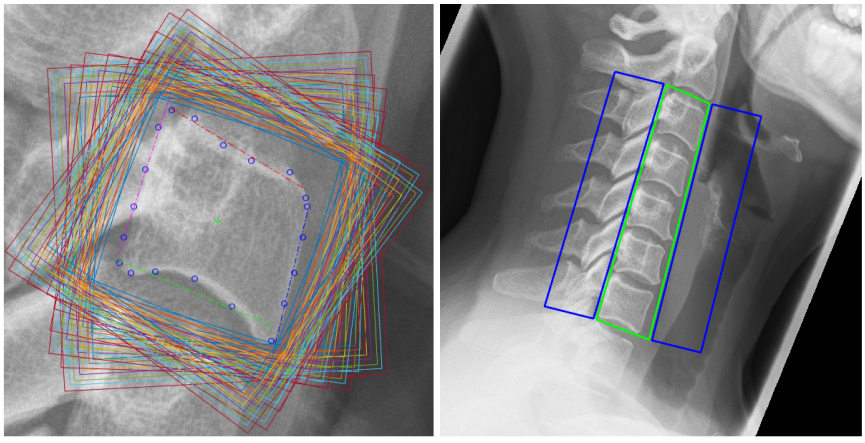
In the next subsections, we first describe how the training data was generated, followed by how the forest is trained. The localization procedure at test time is explained at the end.

3.1.2 Training Data for Random Classification Forest

The random forest utilized here is designed to classify image patches into two classes: vertebra and non-vertebra. To train the random forest, image patches are generated from the training dataset of 124 images and labelled into a vertebra class or non-vertebra class. The patches are considered with different sizes and orientations. To generate positive patches, the manual annotation of the vertebral center and boundary points are used.

The original vertebral size is computed from the manually annotated boundary points and the mean vertebral axis is computed from the vertebral centers. The mean vertebral axis is given by the orientation vector described in Sec. 2.4.2. To train the forest, seven different patch sizes with a step of 0.5 mm (starting from the original vertebral size) and 19 orientations of -45° to +45° with a step of 5° have been used. The orientation angle 0° is

the mean vertebral axis. Our training dataset of 124 images contains 586 vertebrae. Thus a total of $586 \times 7 \times 19 = 77,938$ vertebra (positive) image patches were generated. Fig. 3.1a illustrates boundaries of the extracted patches from a single vertebra. To balance the data, equal numbers of non-vertebra patches were generated from the rest part of each image with random sizes and orientations. In order to generate patches for the non-vertebra class, 50% of the patches are considered from both sides of the vertebral column and the rest is collected from other areas of the image. Fig. 3.1b shows the areas from which the positive and negative patches are collected; positive patches are collected from the green box, 50% of the negatives patches are collected from the blue boxes and other negative patches are collected from the remainder of the image randomly with random patch size and orientation. More importance is provided in the areas adjacent to the vertebral column for negative patch creation so that the forest has a better opportunity to distinguish these areas. These image patches are then converted to structured forest (SF) feature vectors [77, 78]. This feature vector collects the gradient magnitude and orientation information at different scales and angles. This feature vector has shown outstanding performance on the edge detection problem [78]. As vertebra patches are mostly filled with edge-like structures, this feature vector is chosen to distinguish



(a) (b)

Fig. 3.1 (a) Positive patch boundaries around a vertebra with different orientations and sizes (b) the green box indicates the region from where the positive patches collected and the blue boxes indicates the region from where 50% of the negative patches are collected.

vertebra from non-vertebra patches.

To compute the feature vector for a patch, first, the patch is resized to a dimension of 16×16 and the resized image is smoothed using a triangle filter. Then, the gradient magnitude is computed in two scales (original and down-sampled by a factor of 2). For each scale, gradient orientations are computed with four directions resulting in eight orientation channels. All the channels are visualized Fig. 3.2 using a hypothetical input patch. These eight orientation channels, two magnitude channels and the smoothed intensity channel (total 11) of size 16×16 are then resized to 5×5. These smaller size patches are then used to compute pair-wise difference vectors. The 11 channels of size 16×16 result in 2,816 feature candidates and pair-wise difference from the same number of channels result in $11 \times \frac{(5\times5)!}{2!(5\times5-2)!} = 3,300$ feature candidates from each patch. The total feature dimension stands at $2,816 + 3,300 = 6,116$ for a single patch. Creating 6116 feature candidates from a small image patch of size 16×16 is redundant. The redundancy is also visible in Fig. 3.2. However, this feature vector has been proven to perform outstandingly well with the random forest algorithm [77, 78]. A probable reason could be that the randomness introduced in the feature selection process at the split nodes of the random forest copes nicely with this redundancy. The feature vectors are computed using the public toolbox [79] provided by the authors of the articles [77, 78]. Once the feature vectors and corresponding binary labels are ready, a random classification forest is trained on the data.

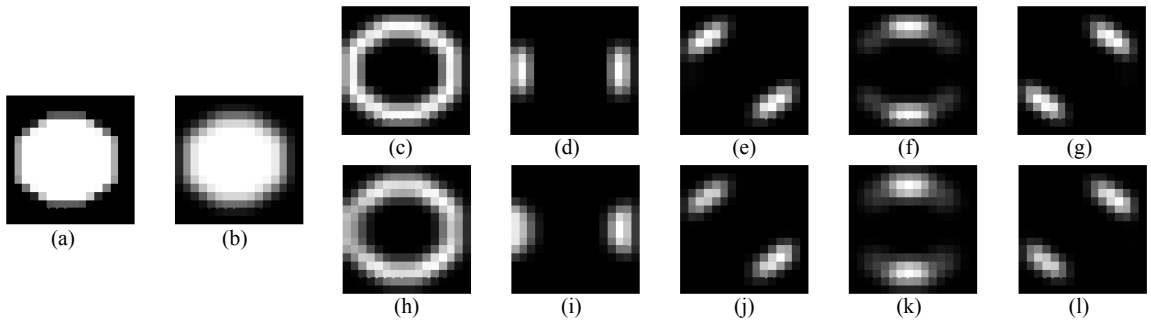


Fig. 3.2 (a) Input image patch of size 16×16 (b) smoothed input image (c) gradient magnitude at the original scale (d-g) gradient orientations with four different directions at original scale (h) gradient magnitude after down-sampling (i-l) gradient orientations with four different directions after down-sampling.

3.1.3 Training Random Forest

Random forest is an ensemble of decision trees (see Fig. B.1). Here, we are training a binary classification forest capable of distinguishing between a vertebra image patch and a non-vertebra patch. Each of tree only sees a random 25% of the training data, which means each tree begins training with 38,969 randomly chosen training samples at the root node. The data in the root node is then divided into left and right child node based on the information gain (IG). Our feature vectors have 6,116 variables. To split the data at each split node, only a few variables (*nVar*) are chosen randomly. Each variable is tested for only a handful of thresholds, $nThresh$. So, at each split node, a total of $nVar \times nThresh$ data splits are considered. For each split node, information gain is computed using the classification entropy, H. The split which achieves the maximum information gain is chosen to split the data. The information gain is computed as:

$$IG = H(S) - \sum_{i \in \{L,R\}} \frac{|S^i|}{|S|} H(S^i), \tag{3.1}$$

where S is a set of examples arriving at a node and S^L, S^R are the sets of data that travel left (L) and right (R), respectively. $H(S)$ is the classification entropy of the data S:

$$H(S) = -\sum_{c \in C} p(c) log(p(c)), \tag{3.2}$$

where C is the set of classes available at the considered node, $p(c)$ is the probability of the class c in the set C. In our case $C \subseteq \{0,1\}$, zero indicating non-vertebra patches and one indicating vertebra image patches.

The random forest has a number of hyper-parameters. We have already mentioned the number of variables to test at node split (*nVar*) and number of thresholds to choose from ($nThresh$). Apart from these two, we also have: maximum allowable tree depth (*nD*), minimum number of samples at a node (*nMin*) and number of trees ($nTree$) which should be set before training a forest. The parameters used for training the classification forest

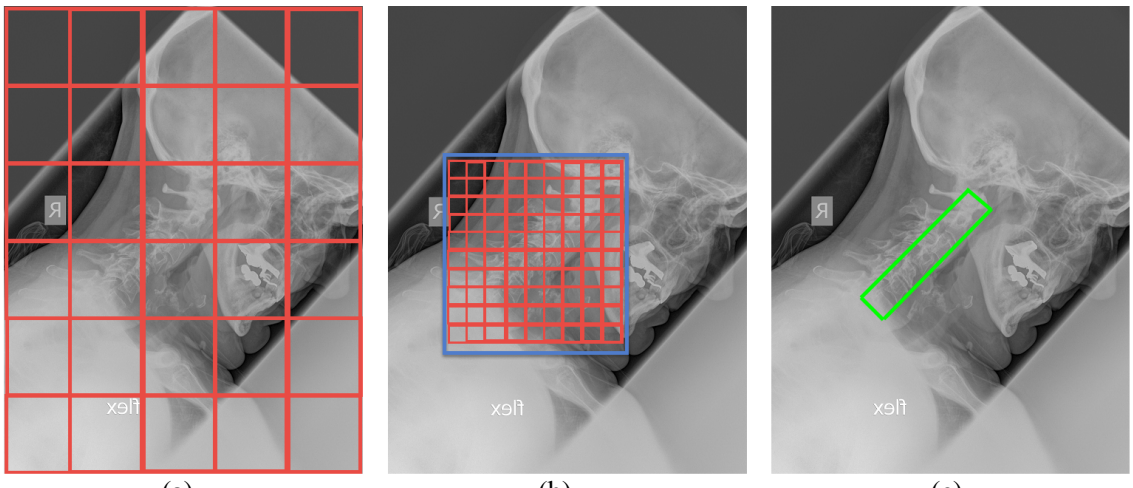
are reported in Table 3.1. These parameters are chosen based on a sequential parameter search. For the parameter search, we have used 80 images for training and 10 images for testing/validation. These images were randomly chosen from our Dataset A, described in Appendix A.

Parameters	Values
nD	10
nMin	50
nTree	10
nVar	85
nThresh	5

Table 3.1 Optimized hyper-parameters for random forest.

3.1.4 Spine Localization

At test time, a new image is fed into the framework for spine localization. The localization is done in two steps. First, we generate image patches sparsely from all over the image. These image patches are then fed into the trained forest, which provides the information



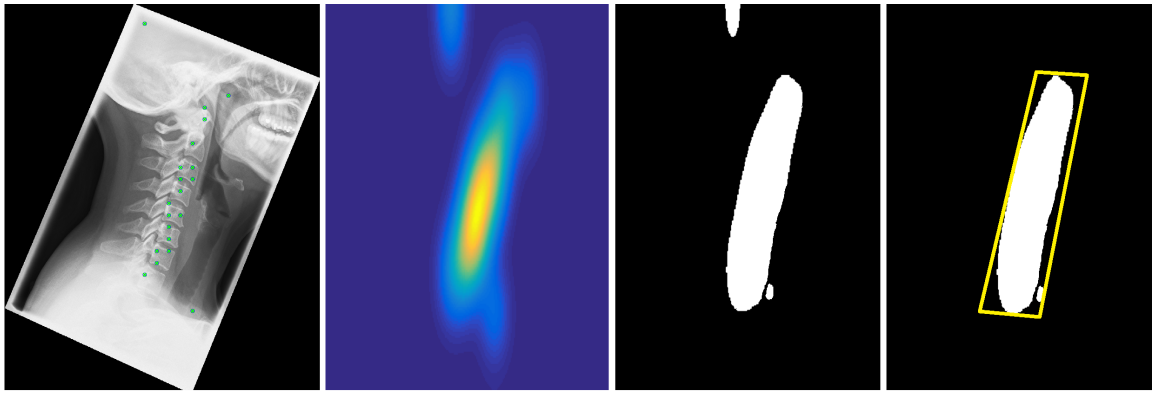
(a) (b) (c)

Fig. 3.3 (a) Sparsely generated image patches to be fed into the trained random forest (b) coarse bounding box (blue) with densely sampled patches for fine localization of the spine (c) final bounding box localizing the spinal region. For simplicity, multiple orientations, sizes, and overlapping patches have not been demonstrated.

about which patches are likely to be from the spinal region. Then a coarse bounding parallelogram is generated near the spinal region. The same process is then repeated within this bounding parallelogram by extracting patches with multiple scales and orientation. The process is summarized in the toy example of Fig. 3.3, and described in more detail in the next subsections.

3.1.4.1 Coarse Localization

For coarse localization of the spine, a set of test points is generated on the image at fixed step size (S_1). A single orientation 0° (O_1) and a fixed patch size (P_1), is considered to generate image patches, one at each of the test points. The generated image patches overlap neighboring image patches. The amount of overlapping is controlled by the parameters S_1 and P_1. These patches are fed into the trained forest. The forest determines which test points belong to the spinal region. These positive predicted points, $[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$, are then passed to the vote accumulation phase to generate a bounding parallelogram. The process is summarized in Fig. 3.4.



(a) (b) (c) (d)

Fig. 3.4 (a) Positive votes on the image (b) resultant distribution (H) (c) H after binarization (d) H after elimination of invalid areas with the minimum bound parallelogram (yellow).

3.1.4.2 Vote Accumulator

The vote accumulator adds a Gaussian kernel at each of the positive votes. The bandwidth, t, of these kernels are automatically estimated using a diffusion-based technique proposed by

Botev et al. [80]. This method allows the bandwidth (t) to change dynamically based on the vote distribution from image to image. The resultant distributions are then added together to form a single distribution, H, over the image space:

$$H(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi t}}e^{-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^2}{2t}}, \tag{3.3}$$

where N is the number of total positive votes coming to the accumulator.

This distribution over the image space is converted to a binary image, B, by dynamic thresholding:

$$B(\boldsymbol{x}) = \begin{cases} 1 & \text{if } H(\boldsymbol{x}) > H_t \\ 0 & \text{otherwise} \end{cases}, \tag{3.4}$$

where $H_t = K \times max(H)$ and K is a constant. As $max(H)$ is different for different images, H_t dynamically changes accordingly. The resulting binary image may be divided into a number of regions, B_j (Fig. 3.4c). The area of each of these parts is measured (A_j) and weighted (w_j) based on the distance from the image center (C_{image}) to the centroid of the concerned image part (C_{B_j}):

$$A_j = area(B_j), \tag{3.5}$$

$$w_j = \frac{1}{distance(C_{image}, C_{B_j})}, \tag{3.6}$$

and

$$wA_j = A_j \times w_j, \tag{3.7}$$

where $j = 1, 2, ..., M$; M is the number of disconnected areas in B and C_a denotes the centroid of the area a. In Fig. 3.4c, $M = 3$. As the images are taken to diagnose cervical spine related injuries, the assumption is that the spine should be located near the image center, not at any extreme corner of the image. Then some of these areas are eliminated if they are small

enough or located far from any adjacent areas:

$$\hat{B}_j = B_j = \begin{cases} valid\ (kept) & \text{if } wA_j > A_t \ \&\ d_{B_j} < d_t \\ invalid\ (eliminated) & \text{otherwise,} \end{cases} \tag{3.8}$$

where

$$d_{B_j} = min\Big(\Big\{distance(C_{B_k}, C_{B_j}) : k \in \{1,2,...,M\}\ and\ k \neq j\Big\}\Big), \tag{3.9}$$

where A_t and d_t are the area and the distance thresholds, respectively. This process reduces the chance of mis-detection. For example, one such mis-detection can be seen in the skull region of Fig. 3.4c. Finally, a minimal bounding parallelogram is generated to enclose the remaining areas [81]:

$$BoundingParallelogram_{coarse} = mBP\Big(\Big\{\hat{B}_j : j \in \{1,2,...,N_B\}\Big\}\Big), \tag{3.10}$$

where *mBP* computes the minimum bound parallelogram enclosing the given regions [81] and N_B is the number of valid disconnected regions. This parallelogram is the output of the coarse localization stage. In Fig. 3.4d, $N_B = 2$.

3.1.4.3 Fine Localization

The coarse localization operates on a single resolution and orientation. As a result, may struggle to find vertebra with uncommon orientation or size. As the bounding parallelogram of the previous stage is only meant to find the approximate area covered by the vertebrae, coarse localization is enough. But in order to find the orientation of the spinal curve, a finer localization with multiple patch resolutions and orientations is necessary. In this fine localization stage, a new set of test points is created within the coarse localization bounding parallelogram, with varying step sizes (S_2). At each test point, multiple patches are generated with different patch sizes (P_2) and angles (O_2). Then the same random forest, described in Sec. 3.1.3, is used for the patch classification and then, another vote accumulation process is conducted. This creates a refined bounding parallelogram within the first stage bounding

parallelogram. The orientation angle of this smaller bounding parallelogram is computed as the orientation of the vertebral column.

3.1.4.4 Localization Hyper-parameters

Apart from the random forest related hyper-parameters, the test time localization framework also has a set of free parameters mentioned in Sec 3.1.4.1 and 3.1.4.3. These parameters are chosen intuitively based on several factors like the training patch sizes, orientations, and the localization process. The values of these parameters are reported in Table 3.2.

Parameters	Values
P_1	24 mm
S_1	10 mm
O_1	0°
K	0.5
A_t	10 pixel
d_t	15 mm
P_2	20, 30, 40 mm
S_2	$P_2/2$
O_2	-45°, 0°, 45°

Table 3.2 Parameters and values for the random forest-based localization framework.

As mentioned in the beginning of this chapter, the random forest-based spine localization algorithm suffers from two drawbacks. First, the patch-based search process for the vertebra patches cannot utilize the topological information of spine being located between the skull and the body/shoulder. Second, the resulted rigid parallelogram fails to accommodate the flexibility of the cervical spine. In the next section, we describe a deep learning-based approach for spine localization which addresses both of these issues by localizing the spine with arbitrary shapes in a single shot from an X-ray image.

3.2 Deep Learning-based Spine Localization

The recent success of deep learning in medical image computing inspired us to solve the spine localization problem with the help of dense classification networks [82, 83]. We have formulated the localization problem as a dense classification problem at a lower resolution. Given a set of high-resolution images and manually segmented vertebral ground truth, at a lower resolution, the ground truth becomes a single connected region. We train a dense classification network to predict this region. To encourage the network to predict a single connected region, we introduce a novel term in the loss function which penalizes small disjoint areas and encourages single region prediction. This novel loss function has produced significant improvement in localization performance. In contrast with the random forest-based approach which generates a bounding parallelogram, the proposed framework can produce a localization map of arbitrary shape in a one-shot process and provides a localization result that models the cervical spine much better than a rigid parallelogram. There are two key contributions in this section. First, a novel loss function which constrains the segmentation to form a single connected region and second, the adaptation and application of dense classification networks to cervical spine localization in real-life emergency room X-ray images.

3.2.1 Overview

As mentioned earlier, we have approached the localization problem as a dense classification problem at a lower resolution. The X-ray images are converted into square images by padding an appropriate number of zeros in the smaller dimension and the square images are resized to a lower resolution using bicubic interpolation. This resolution can vary based on the available memory and size of the training networks. For our case, we chose this resolution to be 100×100 pixels. A binary ground truth was created for each image using the manual annotation of the vertebral boundaries. Each of these binary ground truth images is then resized to the same training resolution. At this resolution, the provided vertebral segmentation maps become a single connected area encompassing the cervical spine (blue

region in Fig. 3.5). For this work, we have experimented with modified versions of three dense classification networks found in the literature: fully convolutional network (FCN) [84], deconvolutional network (DeConvNet) [85] and UNet [82]. The networks have been trained from scratch. The networks take an input X-ray image of 100×100 pixels and produce a binary dense classification result of the same resolution.

3.2.1.1 Localization Ground Truth

As stated earlier, our target is to localize the spinal area in a cervical X-ray image using a dense classification network. For this purpose, we required labeled data in the form of a binary dense classification map where the spinal region will be marked as foreground and other parts of the image will be marked as the background class. We start with the manual annotation of the vertebral boundaries as described in Sec. 2.3.1. We first convert these into binary segmentation maps at the original resolution. As our networks are designed to produce an output dense classification map of 100×100 pixels, we must create our localization ground truth of the same size. Since our original image sizes are approximately in the range of 1000 to 5000 pixels, a simple bicubic interpolation-based resize of the vertebral segmentation maps produce a connected localization ground truth in the smaller dimension.

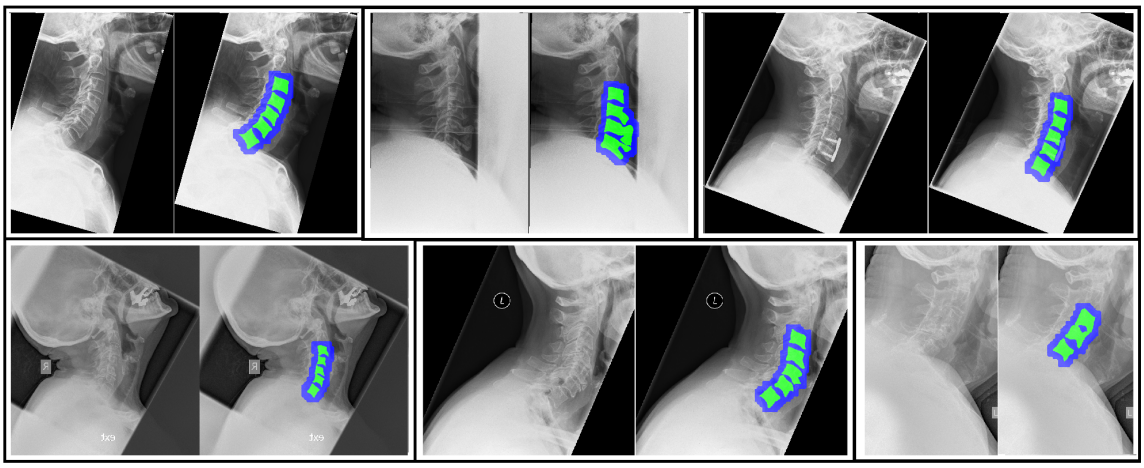


Fig. 3.5 Examples of X-ray images and corresponding ground truth. The ground truth is in blue and overlaid on the original image in the right of each image pair. The vertebrae are shown in green to highlight the difference between the spine localization ground truth and the actual vertebrae.

To visualize the ground truth, it can be transformed back to the original dimensions. The blue overlay in Fig. 3.5 shows how much area the localization ground truth covers apart from the actual vertebrae (green).

3.2.2 Network Architectures

The original FCN [84] and DeConvNet [85] were designed to tackle a semantic segmentation problem having multiple classes on natural images of size 224×224. Since our task here is to localize the spinal region, we essentially have a binary segmentation problem. Thus, we use a shallower version having fewer parameters. We also do not shrink the feature map to single levels like the original FCN and DeConvNet implementation (Fig. B.10 and B.11). This help us to keep more spatial context available for the final prediction. In our implementation, the FCN network has six convolutional layers and two pooling layers (size 2×2, stride 2). The two stages of pooling reduce the dimension from 100×100 to 25×25 thus creating an activation map of smaller size. The final deconvolutional layer upsamples the 25×25 activations to 100×100 pixels, producing an output map equal in size to the input. For our implementation of DeConvNet, we use our FCN as the contracting part of the network. The expanding path forms a mirrored version of the contracting convolutional path. Our UNet also follows a similar structure. However, in the original UNet architecture, the convolutional layers had no zero padding. Thus, the spatial dimensions of the input and the output were different. In our case, all the convolutional layers use zero padding, making the network

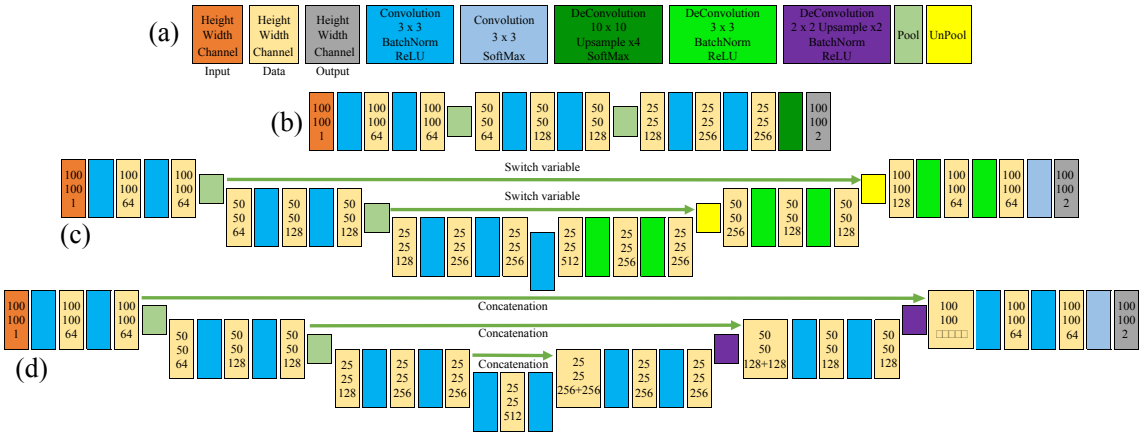


Fig. 3.6 (a) Legend (b) FCN (c) DeConvNet (d) UNet.

capable of producing output with the same spatial dimension as the input. Another important difference from the original implementations of all networks is that we have used a batch normalization layer after each convolutional layer of all three networks. We found that the convergence speed increases with the use of the normalization layer. Fig. 3.6 shows the network diagrams that include data sizes after each layer for a single input image. The number of filters in each layer can be tracked from the number of channels in the data blocks. In total, our FCN has 1,199,042 parameters whereas DeConvNet and UNet have 4,104,194 and 6,003,842 parameters, respectively.

3.2.3 Training

Our training dataset has 124 images. In order to train any network with a large number of parameters, 124 images are not enough. In order to increase the number of training data, we have augmented the images by rotating each image from 5° to 355° with a step of 5°. This results in a training set of 8,928 images. In other words, we now have 89,280,000 data pixels and corresponding ground truth to train our dense classification networks. The augmentation process also made the framework rotation invariant. Our choice for data augmentation was only limited to rigid transformations since non-rigid transformation will affect the natural appearance of the spine in the image. All the networks were trained from randomly-initialized weights using a mini-batch gradient descent optimization algorithm from this augmented training dataset.

Given a dataset of training image (x) - pixel-wise class label (y) pairs, training a deep neural network means finding a set of optimized parameters $\hat{\boldsymbol{W}}_o$ that minimize a loss function, L_t:

$$\hat{\boldsymbol{W}}_o = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} L_t(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W}), \tag{3.11}$$

where N is the number of training examples and $\{x^{(n)}, y^{(n)}\}$ represents n-th example in the training set with the corresponding ground truth. The simplest form of the loss function for

dense classification problem is the pixel-wise log loss also known as the cross-entropy loss:

$$L_t(\{x,y\};\boldsymbol{W}) = -\sum_{i\in\Omega_p}\sum_{j=1}^{M} y_i^j \log P(y_i^j = 1|x_i;\boldsymbol{W}), \tag{3.12}$$

where

$$P(y_i^j = 1|x_i;\boldsymbol{W}) = \frac{\exp(a_j(x_i))}{\sum_{k=1}^{M}\exp(a_k(x_i))}, \tag{3.13}$$

where $a_j(x_i)$ is the output of the final activation layer of the network for the pixel x_i, Ω_p represents the pixel space, M is the number of class labels and P are the corresponding class probabilities. However, this term doesn't constrain the predicted maps to be connected. Since the objective of the localization problem is to find a single connected region encompassing the spine area, we add a novel region-aware term in the loss function to force the network to learn to penalize small and disconnected regions.

3.2.3.1 Region-aware Term

We translate our domain knowledge into the training by proposing a region-based term, L_r. This term forces the network to produce a single region by penalizing small disjoint regions. This term can be defined as:

$$L_r(\{x,y\};\boldsymbol{W}) = \frac{1}{2}\sum_{i\in\Omega_p}\sum_{j=1}^{M} y_i^j E_i P^2(y_i^j = 1|x_i;\boldsymbol{W}), \tag{3.14}$$

where

$$E_i = \begin{cases} \max(N_r - N_t, 0)\frac{A_{max_t} - A_q}{A_{max_t}} & \text{if } i \in R_q \\ 0 & \text{otherwise} \end{cases}, \tag{3.15}$$

where N_r is the number of regions predicted as spine regions, N_t is the number of target regions we are looking for, A_q is the area of the q-th region, A_{max_t} is area of the t-th largest region, R_q is the set of pixels in the region q, and q represents the regions having area less than A_{max_t}. In our case, $N_t = 1$. Notice that, if N_r is equal to or less than N_t and/or $A_{max_t} = A_q$,

the region-based error becomes zero.

The $()^2$ in the region-aware loss term puts more emphasis on the disjoint regions with high probabilities. Eqn. 3.14 is differentiable with respect to the input $P(y_i^j = 1|x_i;\boldsymbol{W})$. The $\frac{1}{2}$ and $()^2$ in the equation make the derivative easily tractable. The differentiability ensures the backpropagation of this proposed loss through the network. The derivative can be computed as:

$$\frac{\partial L_r(\{x,y\};\boldsymbol{W})}{\partial P(y_i^j = 1|x_i;\boldsymbol{W})} = y_i^j E_i P(y_i^j = 1|x_i;\boldsymbol{W}). \tag{3.16}$$

3.2.3.2 Updated Loss Function and Optimization

Combining Eqn. 3.11 and 3.14 gives:

$$\hat{\boldsymbol{W}}_o = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} \Big(L_t(\{x^{(n)}, y^{(n)}\};\boldsymbol{W}) + L_r(\{x^{(n)}, y^{(n)}\};\boldsymbol{W}) \Big). \tag{3.17}$$

To train the network, Eqn. 3.17 is optimized using the mini-batch stochastic gradient descent (SGD) algorithm. An overview of the gradient descent algorithm can be found in [86]. Throughout this dissertation, the RMSprop version of the SGD algorithm has been used for training the neural networks. The networks proposed in this chapter was trained for 30 epochs with a batch-size of 10 images. The training took approximately 18 to 24 hours on a system with NVIDIA Quadro M4000 GPU.

The contribution of each term in the total loss can be controlled by introducing a weight parameter in Eqn. 3.17. Informal experiments were performed by multiplying the region-aware loss term, L_r, with a factor of 0.5, 1 and 2. However, the effect of these weights on the overall performance was negligible. Thus in the following sections, we only report the results from the network trained with the unity weight factor (i.e., Eqn. 3.17).

3.3 Experiments and Metrics

Both of the localization frameworks have been tested on the 172 images of our test dataset. The random forest-based algorithm takes as input a full resolution test image and produces two bounding parallelograms, one for each stage: coarse and fine. For the deep learning-based framework, a test image is padded with zeros to form a square, resized to 100×100 pixels and fed forward through the network to produce localization map of the same resolution, 100×100. In case, if the network results in multiple regions, only the largest connected region is kept. This map is converted into a single binary map and transformed (resized and unpadded) back to the original image resolution.

The predicted binary localization maps are then compared with the vertebra-level ground truth (green area in Fig. 3.5). We have reported four metrics 1) sensitivity 2) specificity 3) LM_{in}: percentage of landmark points inside the predicted region and 4) θ_e: difference in orientation angle. The sensitivity and specificity are computed based on the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) pixels between the prediction and the ground truth:

$$Sensitivity = \frac{|TP|}{|TP| + |FN|}, \tag{3.18}$$

$$Specificity = \frac{|TN|}{|TN| + |FP|}. \tag{3.19}$$

The percentage of landmark points inside the predicted region (LM_{in}) is computed by computing the percentage of vertebral boundary curve (the blue curve shown in Fig. 2.11) inside the predicted area. Finally, the ground truth orientation is measured by the angle of the smallest possible parallelogram that covers the vertebral ground truth. Similarly, the prediction angle is computed as the angle of the smallest possible parallelogram that covers the predicted region. For the random forest-based method, the prediction result itself is a parallelogram. The difference between the angle of the predicted parallelogram and the ground truth parallelogram is reported as θ_e.

3.4 Results

The average metrics over the test dataset for the algorithms are reported in Table 3.3. The deep learning-based framework has six different versions: three networks, FCN, DeConvNet and UNet, each of which was trained either with or without the region-aware term, '-R' signifies the use of the updated loss function of Eqn. 3.17. The bold font indicates significant improvement from the counterpart of the same network/method according to a paired t-test at a 5% significance level. An italicized font indicates the best performing method in terms of the metrics.

	Sensitivity	Specificity	LM_{in}	θ_e
RF-Coarse	**0.9523**	0.9156	**0.9308**	10.6062
RF-Fine	0.7967	***0.9818***	0.6631	**6.3769**
FCN	0.9433	**0.9744**	0.9204	3.5137
FCN-R	***0.9690***	0.9708	***0.9563***	*3.1236*
DeConvNet	0.8846	0.9738	0.8586	8.3893
DeConvNet-R	**0.9201**	**0.9762**	**0.9030**	**5.0269**
UNet	0.8769	**0.9761**	0.8697	4.8608
UNet-R	**0.8969**	0.9741	**0.8888**	4.8082

Table 3.3 Average metrics for spine localization.

The coarse stage of the random forest framework produces the best results in terms of sensitivity and LM_{in}, but produces the worst results in terms of specificity and θ_e. The fine stage improves specificity and θ_e but sensitivity and LM_{in} drops dramatically. This can be attributed to the bounding parallelograms generated at the fine localization stage. The second and third columns of Fig. 3.8 and 3.9 show the localized bounding parallelograms generated by the random forest-based framework. The coarse localization parallelogram covers a lager area encompassing the spinal region, thus the sensitivity and LM_{in} is very high for the coarse bounding parallelogram. However, the coarse parallelogram is not capable of tracking the spine orientation correctly. On the flip side, the fine localization stage creates a smaller bounding parallelogram which often excludes some part of the vertebral region. While this parallelogram represents the orientation of the column well but fails to perform in terms of sensitivity and LM_{in}.

The deep learning-based frameworks produce balanced results. Among three different networks, FCN produces better quantitative results than DeConvNet and UNet. The inclusion of the region-aware term significantly improves the performance of each of this networks in terms of sensitivity and LM_{in}. However, the specificity drops for FCN and UNet by a small amount. Based on the results, we can conclude that the FCN-R produces the best performance. The boxplot of the metrics are shown in Fig. 3.7. Although the average performance is good for all the networks, the inclusion of the region-aware term always reduces the standard deviation of the metrics, indicating the positive effect of the novel term. However, there are several outliers in the boxplots. Most of these outliers come from the challenging cases in our test dataset. Some of these challenging cases are shown in Fig. 3.9.

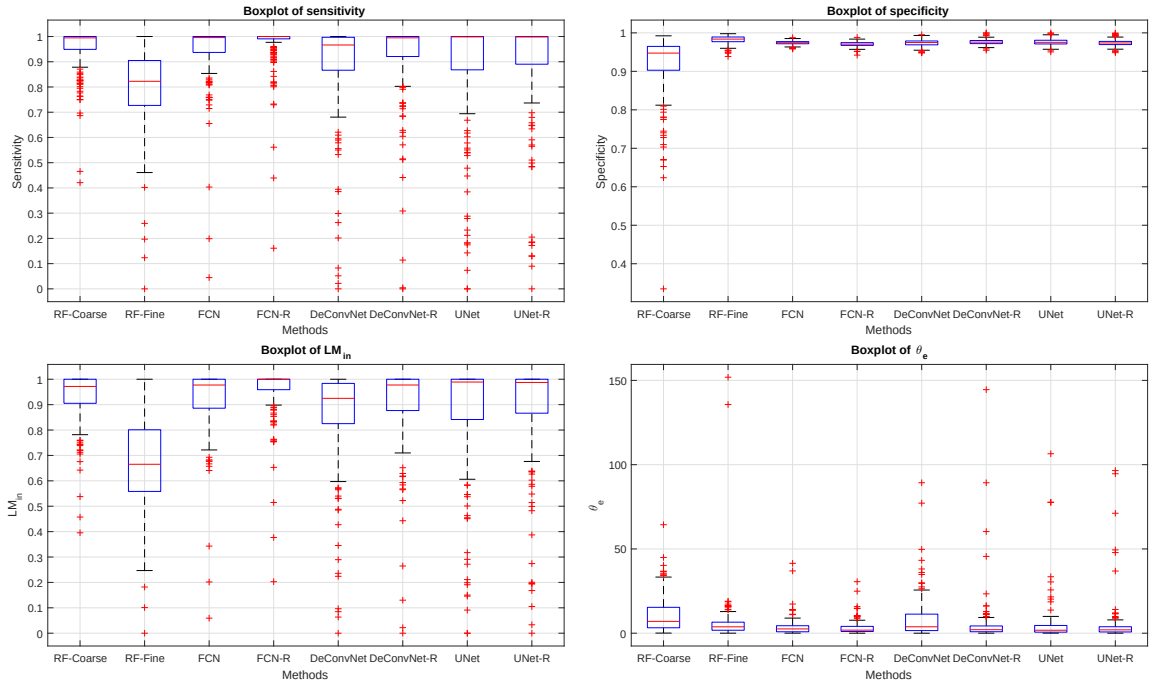


Fig. 3.7 Boxplots of the quantitative metrics.

Fig. 3.8 shows qualitative results for comparatively easier cases for all the methods. Because of the capability of generating arbitrarily shaped localization maps, the dense classification networks results follow the spinal column much better than the rigid bounding parallelograms generated by the random forest framework. The improvement is more noticeable when the spinal column is curved (Fig. 3.8a). The multi-stage stage upsampling the DeConvNet and UNet produces finer localization results than the FCN. On the flip side,

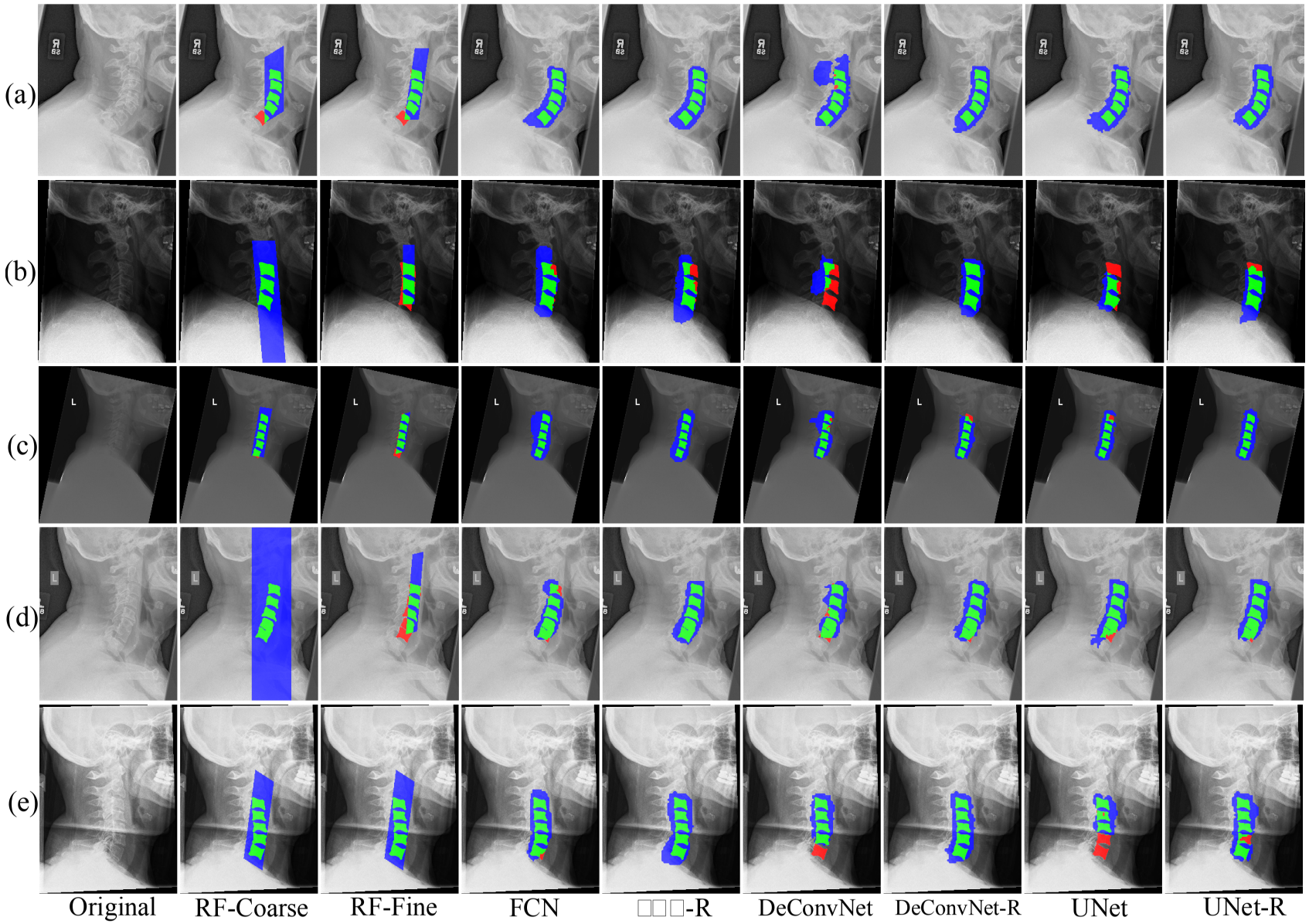


Fig. 3.8 Qualitative results. The green represents true positive (TP), the blue represents false positive (FP), and the red represents the false negative (FN) pixels.

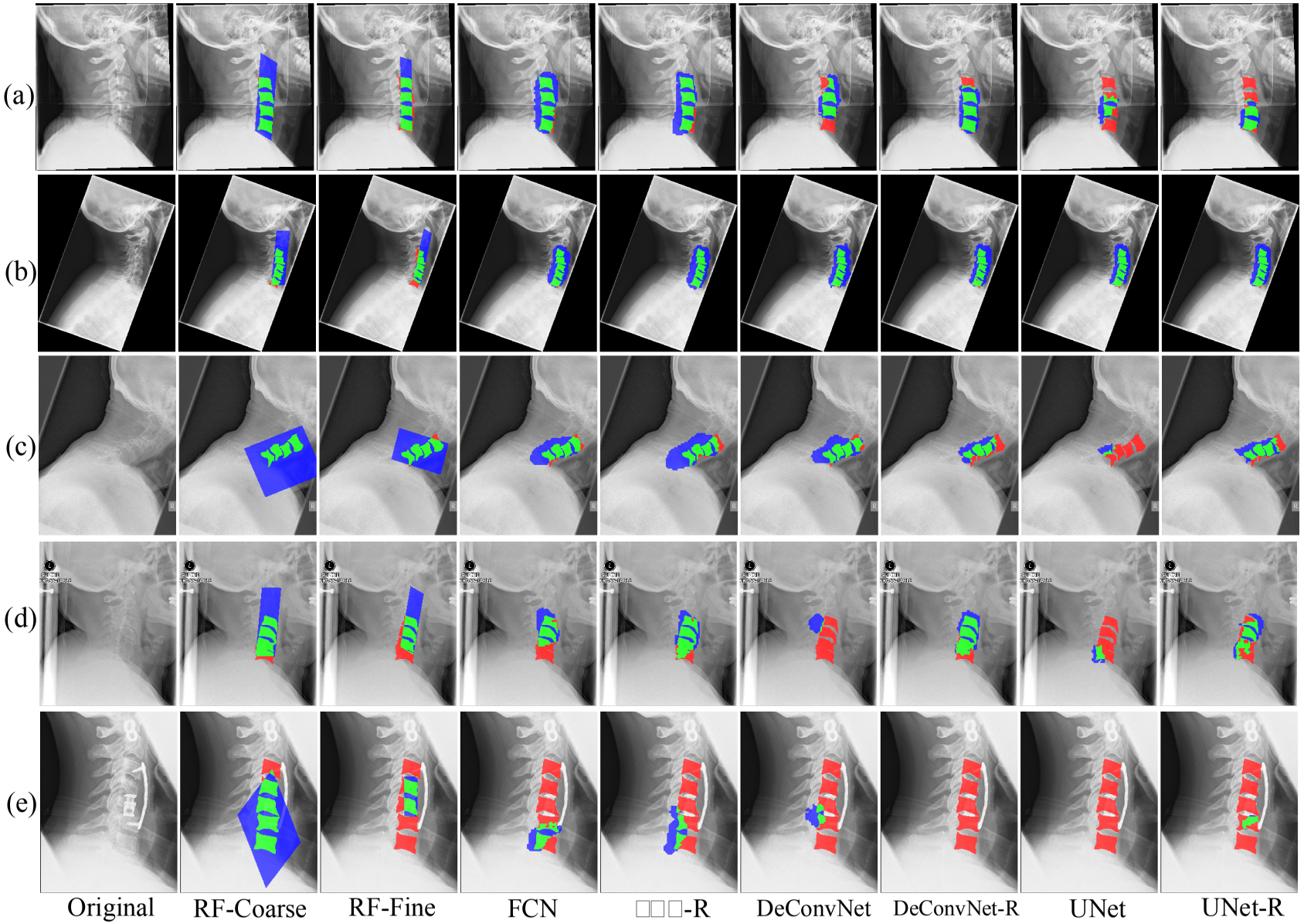


Fig. 3.9 Qualitative results for challenging cases. The green represents true positive (TP), the blue represents false positive (FP), and the red represents the false negative (FN) pixels.

which may cause a portion of the cervical vertebrae to fall outside the predicted region. This effect can be seen in Fig. 3.8c, 3.8d and 3.8e. The effect of the region-aware term is also noticeable in most of the cases. The novel term often produces better localization maps, especially visible for all the networks in Fig. 3.8b. Cases with image artefact are shown in Fig. 3.8e and 3.9a. Spinal columns with severe degenerative changes and fractures are shown in Fig. 3.9b, 3.9c and 3.9d. It can be seen that the dense classification networks suffer more than the random forest counterparts in these severe cases. The patch-based approach of random forest framework is less dependent on the overall look of the spine thus performs better than the deep networks which generate the output in one-shot. Another severe case with a surgical bone implant is shown in Fig. 3.9e where all the algorithms failed to localize the spine. This is the only example in our dataset where the spine is zoomed to the extent that the skull is not present in the image. As the dense classification networks are trained with images where the skulls are always visible partially, the networks fail to localize the spine for this particular test image. The performance is worsened by the presence of multiple prominent surgical implants.

In terms of the time required for the frameworks to make an inference, the slowest deep learning-based framework, UNet-R, is approximately 60 times faster than the random forest-based counterparts. The patch-based sliding window approach of the random forest-based framework requires more computation time, where the deep learning-based framework localizes the spine in a single shot. The deep learning-based framework is also generalizable and robust to different dataset. We have tested the proposed framework on cervical X-ray images of NHANES-II dataset [60] and even without any adaptation or transfer learning/fine tuning on the networks, it showed promising capability of generalization in localizing the cervical spine. However, due to insufficient ground truth information on this dataset, quantitative results are not available. A few qualitative localization results on this dataset are shown in Fig. 3.10. These results also illustrate the fact that our framework is invariant to rotation of the test image.

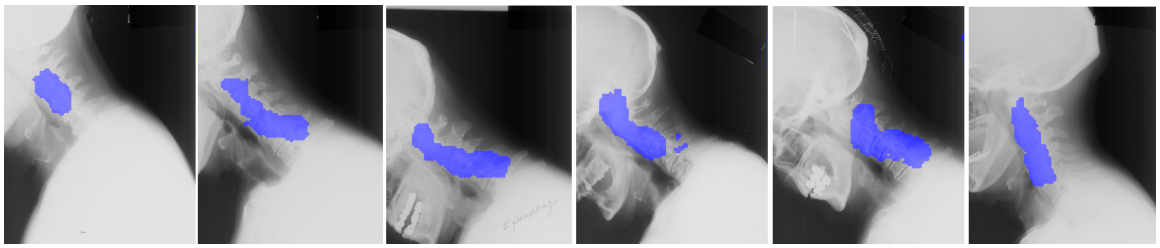


Fig. 3.10 Localization results (blue overlay) on NHANES-II dataset using FCN-R method.

Among the three networks compared in this chapter, quantitatively the FCN has performed better than DeConvNet and UNet. But the qualitative analysis from Fig. 3.8, reveals that FCN outputs are coarser than others. The single stage upsampling strategy utilized by the FCN causes this effect. Both DeConvNet and UNet have a better architecture where the upsampling is done in stages. These networks first extract the feature context through the contracting path and then refine the prediction by upsampling the feature maps in stages with the help of the information from the contracting path. The DeConvNet shares the information between the contracting and the expanding patch through switch variables and unpooling layers which produce sparse output matrices. The UNet, on the other hand, utilizes a more elegant solution. It upsamples the feature maps using deconvolutional layers and gets information from the contracting side through concatenation of the data matrices. This provides two benefits over the DeConvNet architecture. First, non-sparse output is generated ensuring better predictions and second, it strengthens the back-propagation of the loss through the concatenation path which helps the network to learn better and faster [87]. Thus, in the following chapters, the UNet architecture has been our choice for the neural network-based methods.

3.5 Conclusion

In this chapter, we have described two spine localization frameworks: a random forest-based framework and a deep learning-based framework. The random forest-based framework has two stages. The first stage localizes spine with a coarse bounding parallelogram whereas the second stage performs a dense search within the coarse parallelogram with multiple patch

sizes and orientations and results in a finer bounding parallelogram. However, the sliding window-based patch extraction process for this framework fails to see the global context of spine location in an X-ray image and produces a rigid parallelogram which is not capable of capturing the flexibility of the cervical spine. To address these issues, a dense classification network-based localization algorithm was proposed which is capable of producing arbitrarily shaped localization maps resembling the natural spinal curves in a single-shot from an X-ray image. Three dense classification networks were designed, and their performances were compared. A novel region-aware loss term has been proposed to encourage prediction of a single connected region which improved the localization performance significantly for the experimented networks. The dense classification network-based frameworks require more training time than the random forest-based framework, but much faster to produce localization results at the test time. The best performing network has also performed cervical spine localization in another dataset without any adaptation or fine-tuning, proving the robustness and generalization capabilities of the proposed method.

The localization algorithms proposed in this chapter can easily be adapted for other localization tasks. The machine learning modules have to be trained with the new data, and the algorithms should be modified/tuned accordingly to fit the new localization target. The current version of the random forest-based framework can only localize one target. However, the deep learning-based framework can be used for localization of multiple targets in a single image. The novel region-aware term is also generalizable and easily be extended from single region to localization of multiple fixed number of regions. One of the limitations of this region-aware term is that the number of regions has to be known before training. Future work can be performed to make the proposed region-aware term generalizable for any number of regions by dynamically learning the parameter that controls the number of predicted regions.

The frameworks described in this chapter localize the spinal region robustly with high accuracy. However, the exact location of the vertebrae inside the predicted region is still unknown. The next step in our quest for a fully automatic framework is to localize the

vertebral centers. In the next chapter, we describe a novel framework which is capable of localizing the vertebral centers from the predicted spinal region.

Chapter 4

Center Localization

Landmark localization is a fundamental problem in medical image computing. Many of the segmentation techniques require initialization of a mean proposal near the anatomy of interest [24, 25, 38, 88–90]. While many of these methods depend on manual interventions for landmark localization [24, 25, 89, 90], some propose semi-automatic methods [38, 88]. These semi-automatic methods depend on some prior knowledge and/or manual interventions, to reduce the region of interest in an image and then use automatic methods to localize the landmark of interest. In the previous chapter, we have described an automatic method for the localization of the spinal region in an X-ray image which already limits the region of interest. The next task for our fully automatic framework is to localize vertebral centers in the predicted spinal region. In this chapter, we propose a novel probabilistic regression method to localize the vertebral centers using a fully convolutional neural network.

Localization of vertebral landmarks in 2D radiographic images has been addressed in the state-of-the-art work of Bromiley et al. [38, 88]. These methods use a patch-based vector regression technique using random forests, similar to the object detection work proposed in [91]. Instead of the typical practice of regressing vectors pointing towards the location of image landmarks using random forests, we design our center localization algorithm to produce a probability map. Based on the success of the deep neural networks in the previous chapter, we propose a novel deep fully convolutional neural network to learn the mapping

between an image patch and a spatially distributed probability map indicating the locations of the vertebral centers. In contrast to the vector regression techniques, the proposed method can regress multiple landmarks from a single patch.

The proposed deep convolutional neural network (CNN) essentially solves a regression problem. In the previous chapter, we have used CNNs to solve dense classification problems. For classification problems, CNNs produce a probabilistic distribution over the output classes in the targets. However, regression using CNNs is not usually probabilistic [92, 93]. Among the existing literature, a CNN-based probabilistic regression method was proposed in [94] to address this issue. It utilizes a probabilistic interpretation of the Euclidean regression loss function to enforce a set of known constraints on the output space. Instead of finding a set of constraints on the output space like [94], we convert the output space into a 2D probability distribution having the same spatial resolution of the input and train a CNN to learn the mapping from the input image to the spatially distributed probability map. Another spatially constrained CNN was proposed in [95] to localize cell nuclei in histopathology images. This paper employs a patch-based approach where a single nucleus can be detected from a single patch. The method uses a sliding window technique to discover all the nuclei in a full image. In contrast, our proposed method is capable of detecting multiple vertebral centers from a single input patch.

The contributions of the work presented in this chapter are:

1. A novel deep convolutional neural network capable of producing spatially distributed probability map.
2. An innovative method for probabilistic regression using a convolutional neural network.
3. A fully automatic framework for localization of the vertebral centers in a lateral cervical X-ray image.
4. Expert-level performance in vertebral center localization.

4.1 Overview

Localizing vertebral centers is a crucial component in the fully automatic framework. The initial framework described in Sec. 2.4 used manually clicked vertebral centers, making the process semi-automatic. The location of the centers can vary based on interpretation, making the performance of that framework sensitive to user induced variations. To remove this variation and to make the framework reproducible, we propose a novel probabilistic vertebral center localization algorithm in this chapter. The centers inside an image patch are represented by a probability distribution defined over the same pixel space. A modified dense classification network is used to learn the mapping between the image patches and the spatially distributed probabilities. We call this novel network: probabilistic spatial regressor network (PSRN). The network is trained with a novel loss function. The proposed framework achieved human-level performance in localizing vertebral centers.

4.2 Ground Truth

As mentioned in Sec. 2.3, our medical partners have provided us with manually labeled center points for the vertebrae. However, the vertebral centers are not attached directly to any visible landmarks. Thus the human perception of the center varies to some extent. The ground truth vertebra centers (+) are provided by our medical partners when the images were received (see Sec. 2.3.1). To understand the extent of the expert interpretation of the vertebral centers, we asked two experts to annotate vertebral centers multiple times (three times per vertebra). This variation is illustrated in Fig. 4.1. This motivated us to convert the manually clicked centers into a probabilistic distribution. This was achieved by amending a Gaussian distribution at the original ground truth vertebral center (+).

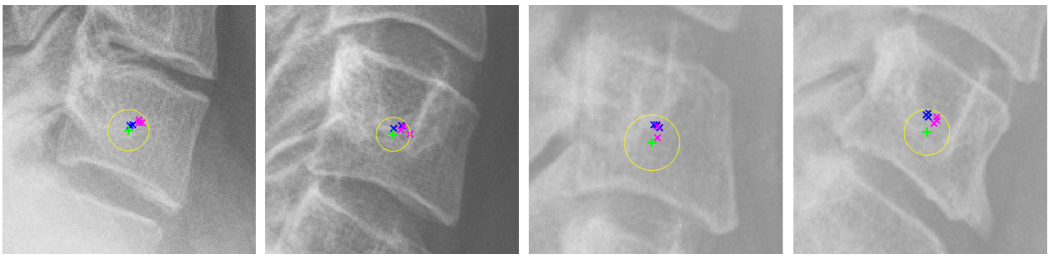


Fig. 4.1 Variation of manually clicked vertebral centers: ground truth center (+), centers clicked by two experts (×, ×) multiple times. The yellow circle represents a 3 mm distance from the ground truth center to illustrate the extent of variation for the expert clicked centers.

The probability distribution at a vertebral center (x_c, y_c) can be defined as a 2D anisotropic Gaussian distribution [96]:

$$F(x,y) = \frac{1}{2\pi\sqrt{v_w v_h}} e^{-\frac{1}{2v_x v_y}\left(a_1(x-x_c)^2 - 2a_2(x-x_c)(y-y_c) + a_3(y-y_c)^2\right)}, \tag{4.1}$$

where

$$a_1 = v_w \cos^2\theta + v_h \sin^2\theta, \tag{4.2}$$

$$a_2 = (v_w - v_h)\cos\theta\sin\theta, \tag{4.3}$$

$$a_3 = v_w \sin^2\theta + v_h \cos^2\theta, \tag{4.4}$$

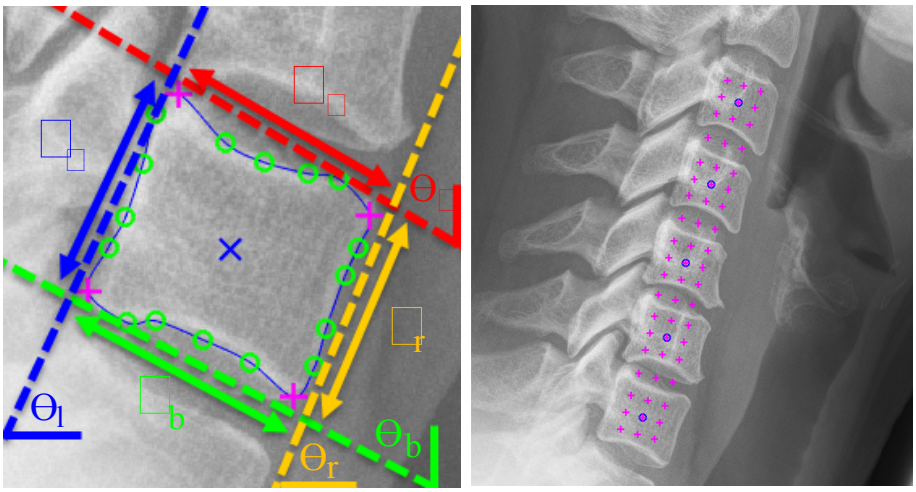
and

$$\theta = \frac{\theta_l + \theta_b + \theta_r + \theta_t}{4}, \tag{4.5}$$

$$v_w = \frac{\frac{w_t + w_b}{2} R}{k}, \tag{4.6}$$

$$v_h = \frac{\frac{h_l + h_r}{2} R}{k}, \tag{4.7}$$

where R is the pixel spacing (in millimeters per pixel) of the image, $k = 60$ is a constant chosen based on visual evaluation of the ground truth and θ_l, θ_b, θ_r, θ_t, w_t, w_b, h_l, h_r are computed from the manually annotated vertebral corners and demonstrated in Fig. 4.2a.



(a) (b)

Fig. 4.2 (a) Different parameter required for probabilistic ground truth generation (b) grid points for training patches.

The process is repeated for all the vertebral centers in an image and a single probabilistic distribution defined over the image space is generated. A few images with overlayed probabilistic center distributions are shown in Fig. 4.3.

The dense classification network used for center localization framework is trained on a dataset of 64×64 patches. To generate a training image patch and corresponding probability distributions, a grid of 9 uniformly spaced points were generated per vertebra and 3 points were generated in between two consecutive vertebrae. An example of these grid points is shown in Fig. 4.2b. From each of these grid points, patches were extracted with two scales (original vertebral size + 2mm and 4mm) and five orientations (-20° to 20° with a step of 5° where 0° is the mean vertebral axis). All these extracted patches are then resized to 64×64 pixels, the resolution at which the network will be trained. A total of 66,600 patches were generated from our 124 training images. Fig. 4.4 shows how these distributions look at the patch level.

Fig. 4.3 Probabilistic distribution for vertebral centers. The heatmap overlay represents the probability of the manually clicked centers.

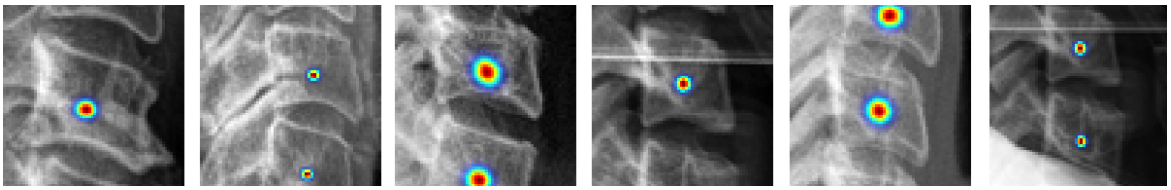


Fig. 4.4 Patch-level ground truth for center localization.

4.3 Methodology

Here the intention is to predict a two-dimensional probabilistic distribution for an input patch of 64×64 pixels. The predicted distribution should have the same spatial resolution as the input patch. The dense classification networks proposed in Chapter 3 are capable of fulfilling this requirement. However, the properties of the output here is different than the dense classification networks. Ideally, the predicted 2D distribution should be continuous (Gaussian) and have a low spread (standard deviation). The single step upsampling strategy of the FCN architecture may cause the prediction to have higher spread and the sparse output of the unpooling layers of the DeConvNet architecture may result in non-Gaussian distribution. Thus, for the probabilistic spatial regressor-based center localization framework, we used a modified version of the UNet [82] architecture.

4.3.1 Network

The UNet of the previous chapter was designed for an input-output pair of spatial resolution 100×100. A larger spatial resolution was chosen because the global localization problem required full image to be seen. Here the training patches only needs the vertebra-level images. Thus the network has been designed for a spatial resolution of 64×64. However, based on the assumption that the model is more complex and the fact that more training samples are available, the number layers in the network has also been increased. The contracting path of the network now has nine convolutional layers. Each convolutional layer is followed by a batch normalization and rectified linear unit (ReLU). Three maxpooling layers in between the convolutional layers downsample the spatial dimension from 64×64 to 8×8. As usual, the upsampling path forms a mirrored version of the downsampling path and upsampling is

achieved by deconvolutional layers. The network diagram is shown in Fig. 4.5. The number of filters in each layer can be tracked from the number of channels in the data blocks. The total number of parameters for the center localization UNet is 24,238,210.

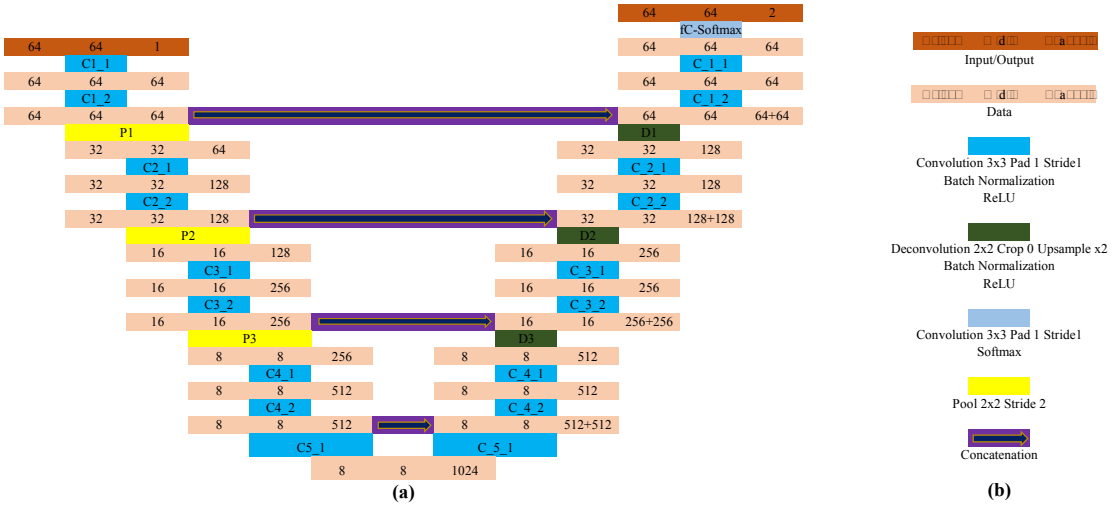


Fig. 4.5 Probabilistic spatial regressor UNet for center localization (a) network architecture (b) legend.

4.3.2 Training

The softmax layer at the end of the network creates a probabilistic two-channel output, just like in a dense pixel-wise classification problem. However, the ground truth here is a probabilistic map, not a binary segmentation map. Thus, the standard pixel-wise dense classification loss of Sec. 3.2.3, cannot be used. We formulate a novel loss function for training the network to predict a probabilistic map.

Loss function for probabilistic spatial regression: To match the two channel output of the final softmax layer, the ground truth probability (GT_p) is also converted to a softmax-like two channel distribution, P_{GT}:

$$P_{GT_{i,channel=1}} = \frac{GT_{p_i} - min(GT_p)}{max(GT_p) - min(GT_p)}, \tag{4.8}$$

$$P_{GT_{i,channel=2}} = 1 - P_{GT_{i,channel=1}}, \tag{4.9}$$

where $i \in \Omega_p$ is the pixel space. Notice that, $P_{GT_{channel=1}}$ is no longer a normalized probability distribution (i.e. doesn't integrate to unity), rather a stretched distribution where the maximum is unity and minimum is zero. This ensures that the softmax layer is able to produce similar distribution, as it squashes the input activations to the range from 0 to 1.

Training our UNet would then mean finding an optimized set of parameters $\hat{\boldsymbol{W}}_o$ which minimizes a loss, L, between the predicted $\hat{y}^{(n)}$ and updated ground truth $P_{GT}^{(n)}$ over the training dataset:

$$\hat{\boldsymbol{W}}_o = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} L(\{x^{(n)}, P_{GT}^{(n)}\}; \boldsymbol{W}), \tag{4.10}$$

where N is the number of training examples and $\{x^{(n)}, P_{GT}^{(n)}\}$ represents n-th example in the training set with corresponding ground truth probability of the regression target. Since the target probabilities are spatially distributed over the pixel space, we can define a pixel-wise loss function per training sample as:

$$L(\{x, P_{GT}\}; \boldsymbol{W}) = \frac{1}{2|\Omega_p|} \sum_{i \in \Omega_p} \sum_{j=1}^{2} w_i (\hat{y}_i^j - P_{GT_{i,channel=j}})^2, \tag{4.11}$$

where

$$w_i = \begin{cases} 1 & \text{if } i \in \Omega_{p_\phi} \\ \frac{|\Omega_{p_\phi}|}{|\Omega_p|} & \text{otherwise} \end{cases}, \tag{4.12}$$

where Ω_p is the pixel space and Ω_{p_ϕ} is set of pixels where the ground truth probabilities are not zero.

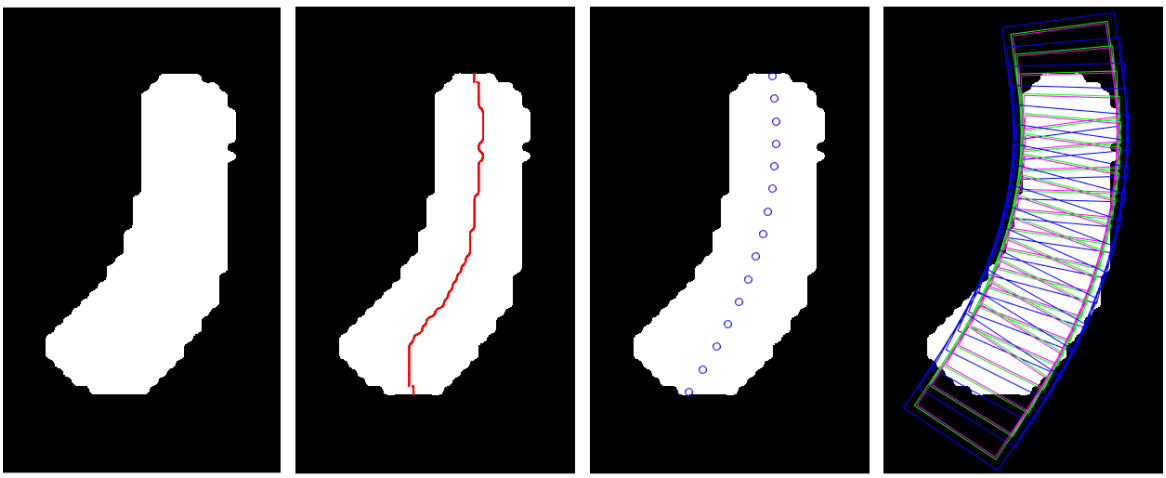
The term $(\hat{y}_i^j - P_{GT_{i,channel=j}})$ measures the difference between the prediction and the ground truth. This pixel-wise difference is weighted by w_i to address the problem of imbalanced regression targets. As most of the pixels in the output probability space have zero probabilities, without this weighting term the solution becomes biased towards the

probability of the majority pixels. In our case $< 5\%$ pixels have non-zero values, thus without the weighting term, the network converges to predict a flat distribution of zeros.

The network is trained on a system with a NVIDIA Pascal Titan X GPU[1] for 30 epochs with a batch-size of 25 image patches. The training took approximately 72 hours.

4.3.3 Inference and Post-processing

At test time, our spine localization algorithm discussed in Chapter 3 provides an automatic region of interest. Using this automatic localization result, we create a set of 15 uniformly distributed points along the approximate central axis of the localization result. The approximate central axis computed by setting a second order polynomial at the center pixels. From each point, three patches are generated with different scales and central axis aligned orientation. The patch sizes are based on the width of the localization area. At each of the 15 points, the width of localization area is computed. The maximum, median and mean of these widths



(a) (b) (c) (d)

Fig. 4.6 Test patch extraction process (a) localized spinal region (b) horizontal center points of the localized area (.) (c) 15 uniformly distributed at the approximate central axis of the region (o) (d) box drawn at the boundaries of each of the 45 extracted patches. Different colors indicate different patch sizes.

[1]We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

Fig. 4.7 Center localization post-processing (a) predicted probability map on the original image (b) thresholded map and potential centers (+) (c) filtered centers after proximity analysis (d) five most probable centers.

are considered as patch sizes. The process is summarized in Fig. 4.6. A total of 15 points $\times$ 3 patch sizes = 45 patches are extracted. These patches are passed through the center localization network to generate patch-level probability maps. The patch size, orientation and position of these probability maps on the original image are known from the patch creation process. These probability maps are then put back on the original image (Fig. 4.7a). The process includes scaling, rotation and translation i.e. affine transformation of the 64×64 pixel patch using the known patch size, orientation and position on the original image space. The probabilities on the original resolution are then thresholded to remove noise (Fig. 4.7b). The threshold is defined as 30% of the maximum probability. For every remaining proposal for a possible vertebral center, the pixel location with the maximum probability is considered as a potential center (Fig. 4.7b). Further post-processing is performed by removing multiple centers in close proximity by keeping the most probable center in a radius of 10 mm (Fig. 4.7c). The radius is chosen based on the size of the training vertebrae. Finally, we keep the maximum number of possible centers to five (C3-C7) and ignore the less probable center proposals when more than five centers are detected (Fig. 4.7d).

4.4 Experiments and Metrics

The center localization framework is tested on our 172 test images. At the patch level, the performance of the network is measured by comparing the predicted probability maps and ground truth maps using the Bhattacharyya coefficient (BC) [97]. The BC represent a measures of similarity between to two probability distribution thus suitable as a metric to evaluate patch-level performance of the proposed network. The Bhattacharyya coefficient (BC) is defined as:

$$BC(p,q) = \sum_{x \in \Omega} \sqrt{p(x)q(x)}, \tag{4.13}$$

where p is the predicted probability distribution, q is the ground truth probability distribution and Ω is space on which both p and q are defined. The BC varies between zero and one where higher values represent better matching.

After the post-processing step, the centers are localized on the original image. The predicted vertebral centers can be divided into three sets: true positive (TP), false positive (FP) and false negative (FN). The TP represents the set of vertebrae whose centers have been correctly detected. A correct detection is considered if the predicted center falls inside a vertebral body studied in this work, i.e., C3-C7. The FP represents the set of predicted centers which did not fall inside any of these vertebrae. Finally, the FN is the set of the studied vertebrae whose centers have not been detected. Based on the TP, FP and FN, we report two metrics: true positive rate (TPR) and false discovery rate (FDR) [98]:

$$TPR = \frac{|TP|}{|TP| + |FN|} \times 100\%,$$

$$FDR = \frac{|FP|}{|FP| + |TP|} \times 100\%.$$

The TPR is also known as the 'recall' and FDR is a complementary metric to 'precision'. While precision represents the percentage of the correct detections among all the detections, FDR represents the percentage of the incorrect detections. We also report the point to point Euclidean distance between the correctly detected centers and corresponding ground truth in mm as distance error.

4.5 Results

The fully automatic center localization algorithm uses the results of the spine localization algorithm from Chapter 3. For the results discussed in this section, we use the spine localization result of the FCN-R network. The performance of the center localization algorithm can also be measured independently. In this case, the uniform grid needed for the patch creation is generated using the spine localization ground truth (Fig. 3.5) instead of the prediction of the spine localization framework as mentioned in Sec. 4.3.3. We will present both results: semi-automatic that uses the spine localization ground truth and fully automatic that uses spine localization predicted by FCN-R.

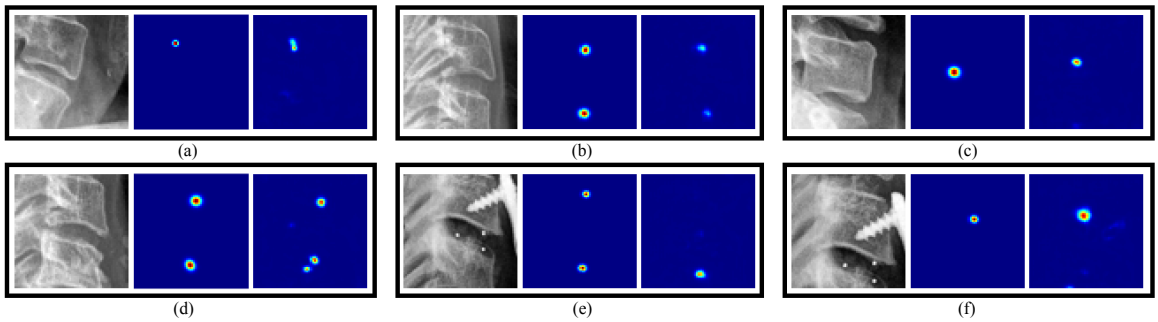


Fig. 4.8 Image patch (left), ground truth probability (middle) and predicted probability (right) with corresponding Bhattacharyya coefficients: (a) 0.8285 (b) 0.7153 (c) 0.3304 (d) 0.6149 (e) 0.4353 (f) 0.3715.

First, we present the Bhattacharyya coefficient (BC) between the patch-level inputs and predictions for the semi-automatic method. A Bhattacharyya coefficient (BC) of zero represents the worst result and one represents a perfect match between ground truth and prediction probability. Over all the test patches, an average BC of 0.58 has been achieved at the patch level. Some of the graphical results with corresponding BC are shown in Fig. 4.8. It can be seen that even with low BC (Fig. 4.8c and 4.8f), the results are similar. The histogram of the BC over all the patches is plotted in Fig. 4.9, a BC of > 0.5 was achieved for 71% of the test patches. A few qualitative results for center localization at the patch level are

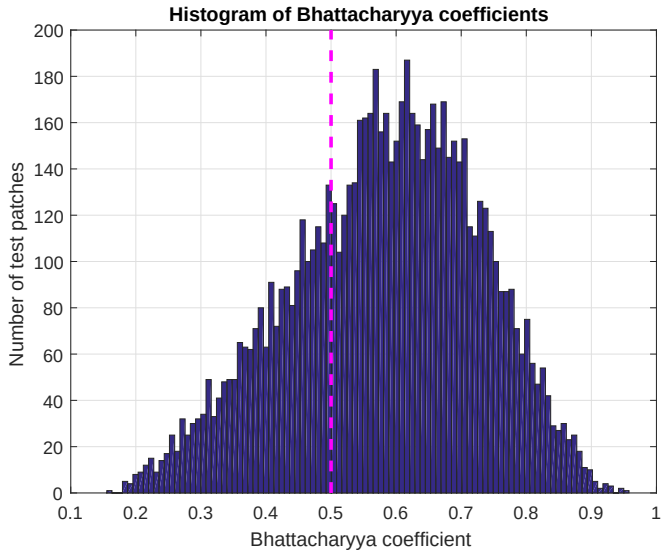


Fig. 4.9 Histogram of Bhattacharyya coefficients.

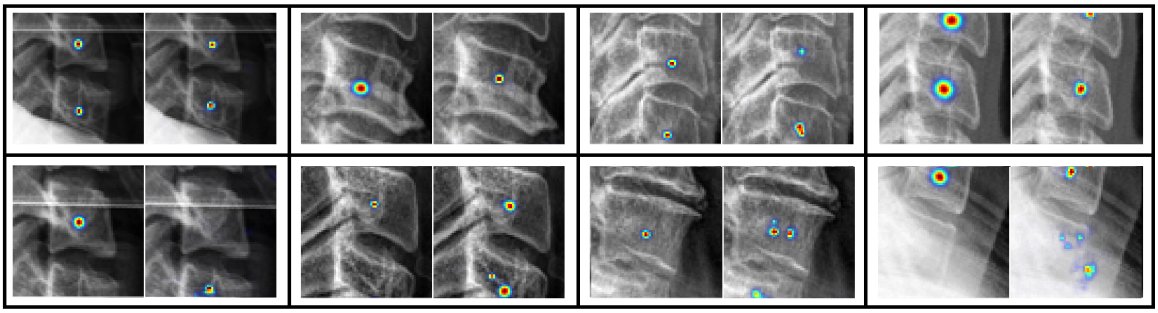


Fig. 4.10 Patch-level center localization results: ground truth (left) and prediction (right).

shown in Fig. 4.10. We have also tested the performance of the proposed network on vertebra patches collected from the NHANES-II dataset. Fig. 4.11 shows a few examples of the results from this dataset proving the robustness of the trained network.

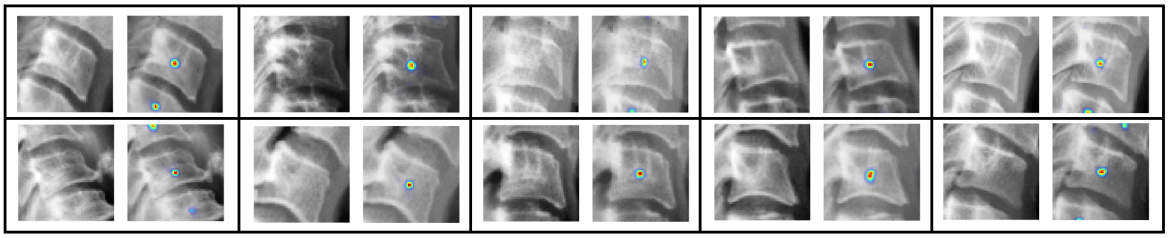


Fig. 4.11 Patch-level center localization results for vertebra patches collected from NHANES-II dataset: input image patch (left) and predicted probability map overlayed as a heatmap on the input image patch (right). The ground truth information was not available for this dataset.

After the post-processing phase, the centers are localized on the full resolution test image. Table 4.1 reports the true positive rate (TPR), false discovery rate (FDR) and distance error for the correctly detected centers in millimeters (mm).

Among 797 vertebrae from our 172 test images, 755 centers were detected with an average error of 1.80 mm. The number of false positives was 38, most of these belong to neighboring vertebrae C2 and T1. To compare the performance of the center localization algorithm with human performance, an expert radiographer was asked to click on the vertebral centers on ten random test images three times. These manually predicted centers are compared with the ground truth centers for those images. The average error was 1.92 mm which is higher than

Test patch creation	Semi-automatic				Fully automatic		
True positive rate (TPR)	94.73%				93.10%		
False discovery rate (FDR)	4.79%				9.40%		
	Median	Mean	Std		Median	Mean	Std
Distance error (mm)	1.62	1.80	0.96		1.54	1.72	0.99

Table 4.1 Performance of the center localization framework. The 'semi-automatic' patch creation process uses localization ground truth and the results reported below are independent of the accuracy of the global localization framework. Results from the fully automatic procedure which uses the localized spine from the global localization framework are reported in the right under the 'fully automatic' patch creation process.

the average error of correctly detected centers by our algorithm. The performance curve is shown in Fig. 4.12.

It can be seen that the distance error is < 3 mm for almost 90% of the correctly detected vertebral centers. For a qualitative comparison, the manually clicked vertebral centers shown in Fig. 4.1 also varied within a 3 mm radius from the ground truth center. The process is repeated by changing the uniform grid creation process in the beginning. In this case, the

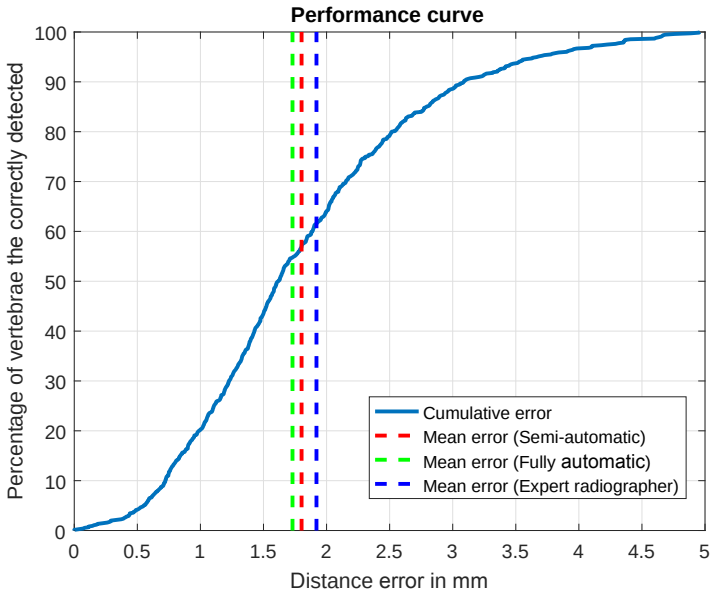


Fig. 4.12 Performance curve for center localization. The blue curve (—) represents what percentage of the correctly detected vertebrae (vertical axis) has a distance error (horizontal axis) lower than specific values.

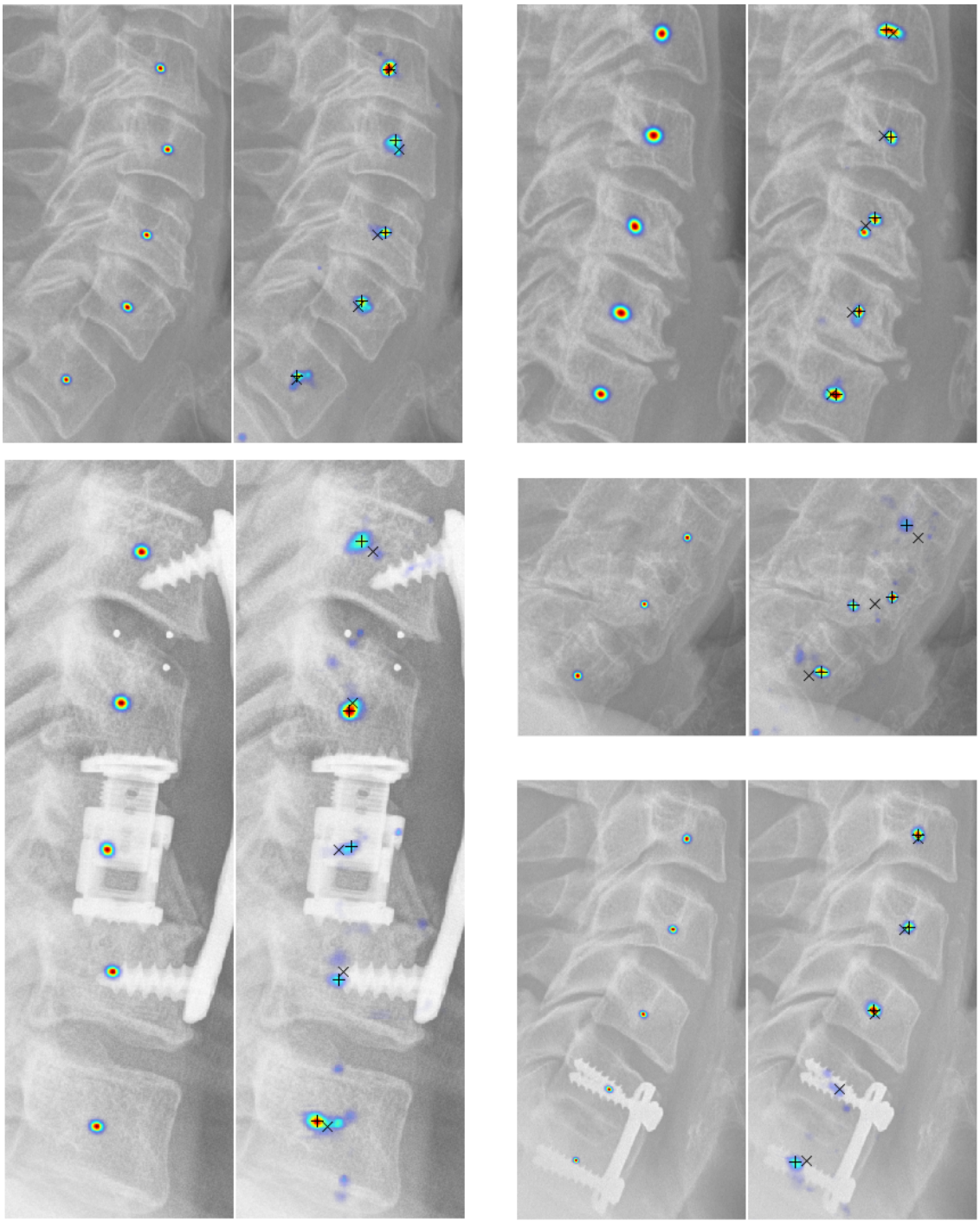


Fig. 4.13 Qualitative center localization results. For each pair, ground truth distribution is shown on the left, prediction distributions are shown on the right. On the predicted image, the ground truth center is denoted as a cross ($\times$) and predicted centers are denoted as plus ($+$).

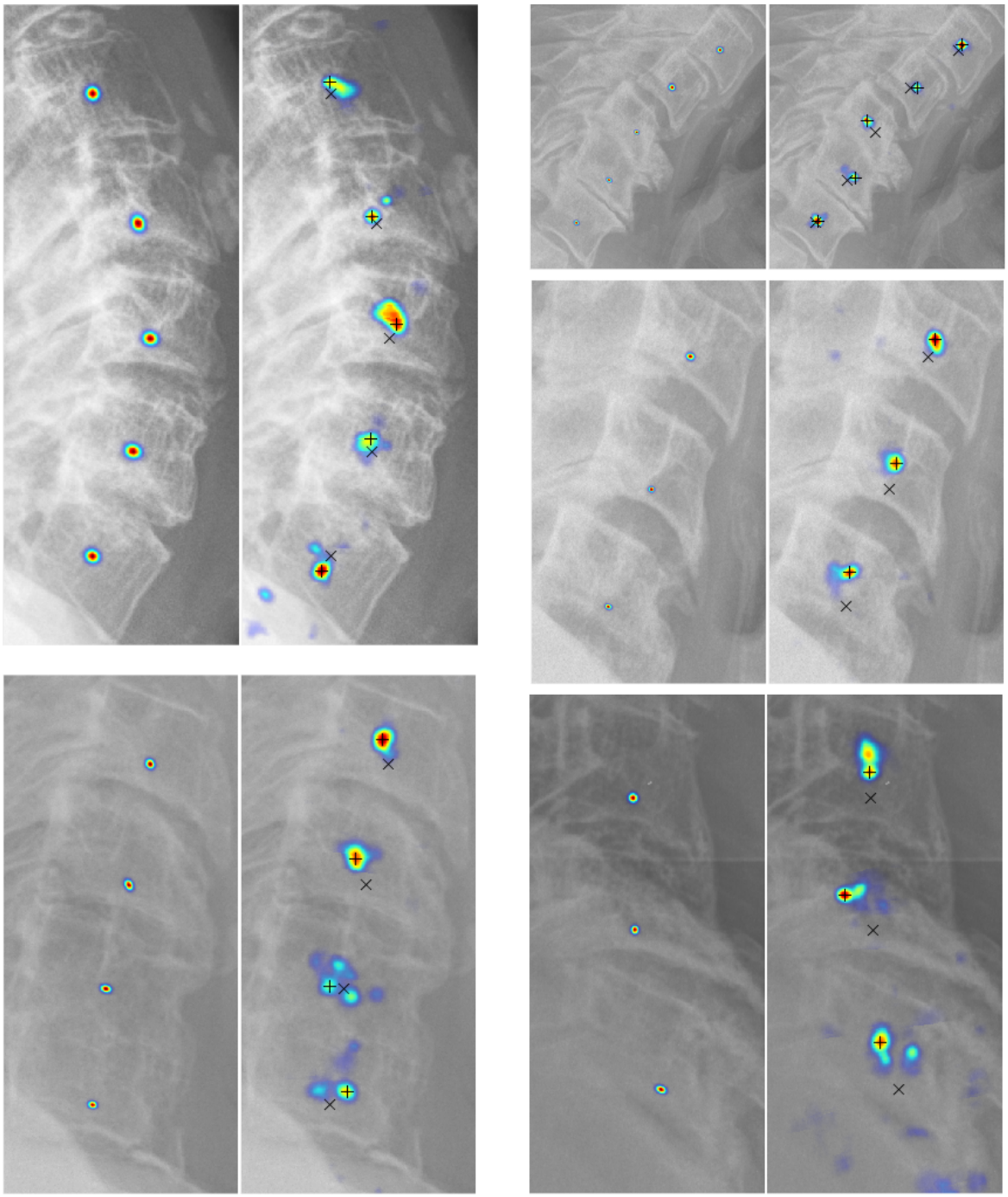


Fig. 4.14 More qualitative center localization results. Refer to the caption of Fig. 4.13 for legend.

uniform grid for patch generation is done using the area predicted by our global localization algorithm (instead of the global localization ground truth), as discussed in Sec. 4.3.3. The metrics are reported on the right side of Table 4.1. It can be seen the TPR dropped from

94.73% to 93.10%, while the FDR is increased from 4.79% to 9.40%. This degradation is because of the incorrect global localization results, as shown in Fig. 3.9. However, among the correctly detected centers, the distance error drops from 1.81 mm to 1.72 mm. The reason behind this is that much of the bad quality image areas have already been removed by the global localization prediction. So the remaining image areas are of comparatively of good quality thus center localization performs better on average on these image areas. Some graphical center localization results on the full resolution images are shown in Fig. 4.13 and 4.14.

4.6 Conclusion

We have described a novel vertebral center localization framework in this chapter. The vertebral center resides in the middle of the vertebral body without direct attachment to any visible image landmark. The perception of the center varies based on human interpretation. Thus the center location is better represented as a probability distribution. To learn the mapping between the vertebral image patch and the spatially distributed probability map of the vertebral centers, a novel deep convolutional neural network has been designed. The proposed network has been incorporated into a framework which localizes vertebral centers inside the spinal region. Combined with the spine localization framework proposed in Chapter 3, the framework proposed in this chapter is capable of predicting vertebral centers from an X-ray image without any manual intervention. The fully automatic vertebral center localization framework has achieved a true positive rate of 93.10% in center detection and an expert-level localization accuracy among the correctly detected centers.

The novel convolutional neural network proposed in this chapter essentially solves a regression problem for image landmark localization. The proposed solution can be adapted to approach various image landmark localization problems, traditionally solved by random forest-based methods [38, 88, 91]. The random forest-based methods often use votes from multiple image patches to generate a probability distribution for a single object. In contrast,

the proposed method is capable of generating a probability distribution localizing multiple objects of interest from a single patch.

Although classification using neural networks is probabilistic, regression using neural networks is often deterministic [92, 93]. The novel convolutional neural network described in this chapter proposes an innovative solution for probabilistic regression using neural networks. Unlike the work proposed in [94] which uses a set of known constraints to achieve probabilistic regression, our method learns to perform probabilistic regression automatically.

So far in this dissertation, we have described a fully automatic framework for localization of vertebral centers. In the next chapter, we will propose and compare several frameworks for localization of another essential vertebral landmark: corners. The probabilistic spatial regression network (PSRN) proposed in this chapter uses a pixel-wise loss function which doesn't constrain the prediction to be a valid probability distribution during training. To address the issue, we will propose major improvements to the PSRN by introducing a new network block and a novel probabilistic loss function.

Chapter 5

Corner Localization

Corners detection is a classical problem in computer vision. Early work in this topic involves segmentation of shapes, extracting the boundary as a sequence of points, and then searching for significant turnings in the boundary [99]. Dependence on the prior segmentation for corner detection was a major drawback of these methods. This was solved by detecting corners with the help of local operators [100–102]. These operators are applied on the image directly to detect corners. Like other topics in computer vision, recent literature on corner detection involves machine learning techniques [103, 104] which learn to detect corners in a supervised manner from manually annotated images. Similar to natural images, corners have great importance in medical images. It can provide key information about the size and the shape of anatomical organs which can then be used for other high-level purposes.

Earlier in Sec. 2.1.1, we listed a selection of injuries related to cervical vertebrae like subluxations (spondylolisthesis, retrolisthesis) and compression fractures (wedge, biconcave, crush). Vertebral corners can play a vital role for detection of these clinical conditions. Automatically predicted vertebral corners can also be used for initialization of several statistical shape model-based segmentation methods [57, 69, 89] to build a fully automatic segmentation framework. Motivated by the importance of the corners, in this chapter, we propose three novel methods for vertebral corner detection. Following the trend in the corner detection literature, our methods also develop from using classical operators to advanced

machine learning techniques.

The first method proposed in this chapter is a Harris corner detector-based framework [102]. It uses prior information from the distribution of corners from the training dataset and combines it with local edge-based features to detect vertebral corners. Second, a Hough forest-based framework which utilizes a patch-based method to regress corners of a vertebra. This framework is inspired from the object detection work proposed in [91]. Finally, a deep probabilistic spatial regressor network (PSRN)-based corner localization framework is proposed where we improve upon the work of the previous chapter by introducing a new spatial normalization layer and a novel probabilistic loss function. All three frameworks described in this chapter are semi-automatic and require centers of the vertebrae to be given. The process can be made fully automatic by augmenting the spine localization and the center localization frameworks from Chapter 3 and 4, respectively, with a corner localization framework proposed in this chapter.

The contributions of this chapter are following:

1. Three novel semi-automatic vertebral corner detection frameworks.
2. A normalization layer for the probabilistic spatial regressor network (PSRN) which generates a valid spatially probability map.
3. A novel loss function based on Bhattacharyya coefficient for training the improved PSRN.
4. A median error of less than a millimeter in localizing vertebral corners.

In the next two sections, we first describe the Harris-based naive Bayes corner detector (HarrisNB) and Hough forest-based method (HoughF) for vertebral corner localization. We have studied these two methods extensively in a prior publication [71] on Dataset A, described in Appendix A. In this chapter, we only report the best-performing methods for these frameworks. Detailed experimentations and results from [71] have been reported

in Appendix A which influenced different choices made in this chapter for HarrisNB and HoughF. In the third section, we describe the improved deep probabilistic spatial regressor network (PSRN)-based corner localization framework. This is followed by results and discussion where we evaluate and compare all the methods. Finally, we end the chapter with the conclusion.

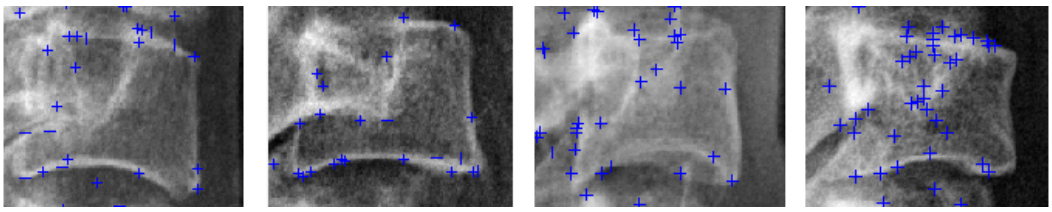


Fig. 5.1 Vertebral corners detected by the Harris corner detector (+).

5.1 Harris-based Naive Bayes Corner Detector

The Harris corner detector is a popular method to identify corners. It uses a second-moment matrix composed of image derivatives. As with other gradient-based methods, it suffers if the image has low contrast which is common to X-ray images. Initially, we attempted to apply standard implementation of the Harris corner detector for localizing vertebral corners, but results were poor due to the presence of noise in the X-ray image and the smooth transition of the vertebral boundary at the corners. Examples of the corners detected by Harris corner detection can be seen in Fig. 5.1. Therefore, we devised a novel multi-scale Harris corner

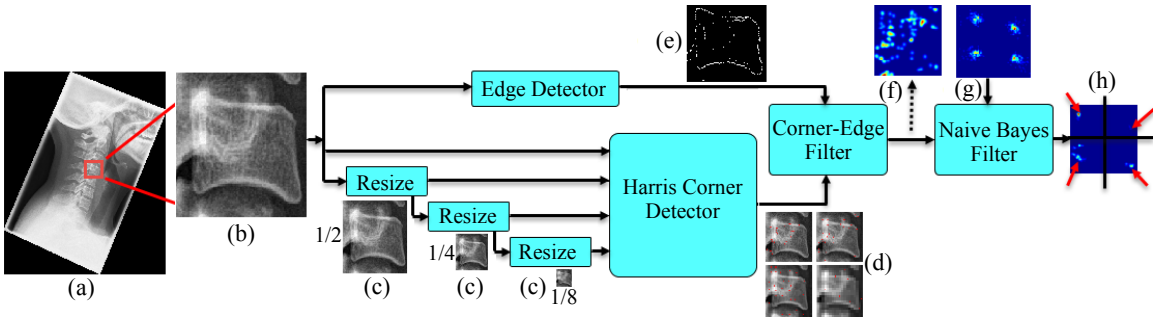


Fig. 5.2 Harris-based vertebral corner detector (a) original X-Ray (b) cropped ROI (c) ROI at different scales (d) Harris Corner detector output at each scale (e) binary edge image (EI_O) (f) output of Corner-Edge filter: $P(C|I)$ (g) $P(L|I)$ (h) final distribution: $P(C,L|I)$, corners are pointed out by red arrows.

detector-based framework with a spatial term trained from our manually labelled corners. The framework is summarized in Fig. 5.2 and described in the following subsections.

5.1.1 Vertebral Patch Extraction

Given a cervical X-ray image and manually annotated vertebral centers, the first step is to extract a vertebral region of interest (ROI). The coarse orientation and size of the vertebrae are computed using the center points. These are given by the orientation vector, $\boldsymbol{F}$, discussed in Sec. 2.4. The magnitude of the orientation vector, $\boldsymbol{F}$, represents the coarse size of the vertebra. Using this orientation and size, a bounding box is generated to identify a region of interest (ROI) around the vertebral center (green box in Fig. 5.3).

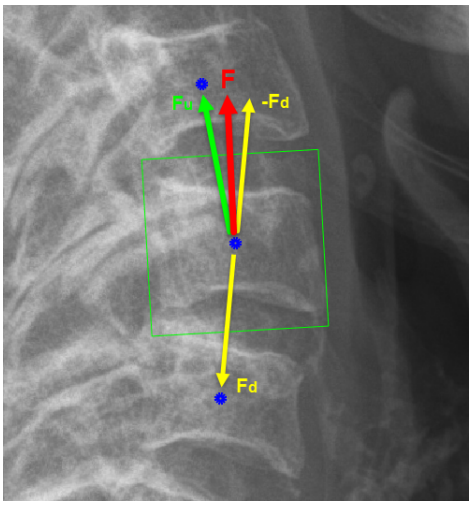


Fig. 5.3 Vertebral patch/ROI extraction.

The distribution of the corners around the center for each vertebra reveals that vertebral corners form a quadrilateral, that can be better approximated as a trapezoid (i.e., convex quadrilateral), as demonstrated in Fig. 5.4 which superimposes normalized corner distributions of the different cervical vertebral bodies in the dataset. Based on this insight, trapezoidal ROIs are extracted.

The size and the angles between the arms of the trapezoidal ROI are computed based on the distribution of the corners in the training dataset. The ROI requires an affine transforma-

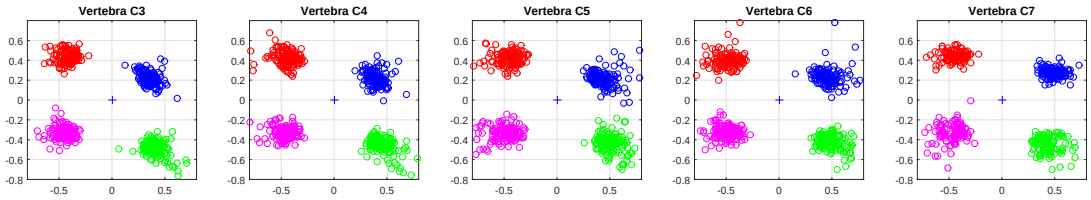
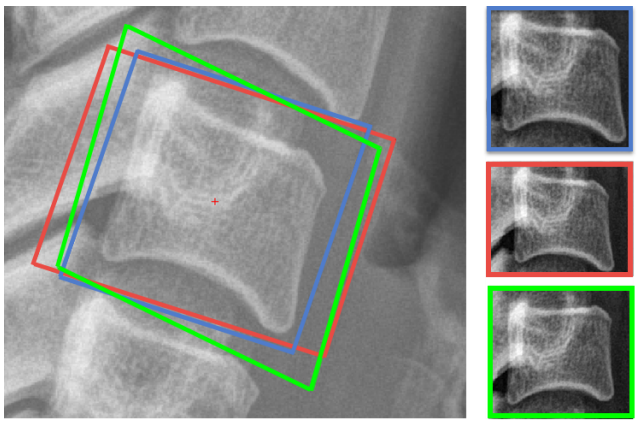


Fig. 5.4 Normalized corner distribution in the dataset.

tion to warp the extracted image patch. The warped image results in an axis-aligned vertebral body which is illustrated in Fig. 5.5b. This is significant because this framework involves corner and edge detection methods which detect edges better when they are axis-aligned. Experiments with square and rectangular ROI have been also been performed on Dataset A and results can be found in Table A.1.



(a) (b)

Fig. 5.5 (a) Different ROIs: square (blue), rectangle (red), trapezoid (green) (b) vertebra inside different ROI: square (top), rectangle (middle) and trapezoid (bottom).

5.1.2 Edge and Corner Detection

The extracted ROI is then passed through a collection of edge detectors. In this work, Canny edge detector, Sobel, Prewitt, Roberts operators and Laplacian of Gaussian (LoG)-based edge detector are used. Each of these detectors returns a binary image:

$$EI_{\psi} = EdgeImage(ROI, \psi), \tag{5.1}$$

where ψ is the edge detection method and $\psi \in \{\text{Canny, Sobel, Prewitt, Roberts, LoG}\}$. Then all the returned binary images are added together and an edge is determined only when more than two edge detectors agree:

$$EI = EI_{Canny} + EI_{Sobel} + EI_{Prewitt} + EI_{Roberts} + EI_{LoG}. \tag{5.2}$$

$$Output\ Edge\ Image,\ EI_O = \begin{cases} 0 & \text{when EI <3} \\ 1 & \text{otherwise} \end{cases}. \tag{5.3}$$

This process reduces noise and false edge detection.

The Harris corner detection is applied on the original ROI at four different scales (1, 1/2, 1/4 and 1/8). As Harris corner detection uses gradients in horizontal and vertical directions to determine a possible corner, the intensity profile at different scales increases the chance of detecting actual corners. Along with the corner location, it also returns a score which represents the likelihood of the detected corner. From each scale, the top 50 corners are selected. All the selected corners are then passed through a Corner-Edge filter. This filter only selects a corner if it falls within a five pixel radius of an edge pixel ($EI_O = 1$). These selected corners are then converted into a probability distribution, $P(C|I)$. A spatial prior probability distribution, $P(L|I)$, is created from the training data containing all the corners (Fig. 5.4). These $P(L|I)$ looks like Fig. 5.2g for different vertebrae. These $P(L|I)$ are then made corner specific $P(L_i|I)$ by considering single (i-th) corners only:

$$P(L_i|I) = \frac{1}{N}\sum_{j=1}^{N} \frac{1}{\sigma_c\sqrt{2\pi}} \exp^{-\frac{(x-C_{ij})^2}{2\sigma_c^2}}, \tag{5.4}$$

where N is the number of training examples, σ_c^2 is the variance of Gaussian distribution initialized at each training corner locations and $P(L_i|I)$ is the prior probability distribution for i-th corner. The variance, σ_c^2, is optimized experimentally using a set of validation images from Dataset A. A final posterior distribution, $P(C_i, L_i|I)$ is then computed by multiplying

$P(C|I)$ and $P(L_i|I)$ following a naive Bayes formulation:

$$P(C_i, L_i|I) = P(C|I)P(L_i|I), \tag{5.5}$$

where C_i is the i-th corner, L_i is the location of that corner, and I is the ROI. $P(C|I)$ comes from the edge-filtered multi-scale Harris corner detection and $P(L_i|I)$ comes from the training data. Final corner positions are found by localizing the maximum probability in each of the four posterior distributions $P(C_i, L_i|I)$. The complete framework is graphically explained in Fig. 5.2.

5.2 Hough Forest-based Vertebral Corner Detector

Hough forest [91], a variant of the random forest [76, 105] algorithm, has shown promising performance in object detection using votes from image patches. Here this algorithm has been adapted and customized in order to localize vertebral corners in X-ray images. In contrast to the random forest algorithm which performs either regression or classification, Hough forest performs both together in the same forest. During training the algorithm requires training

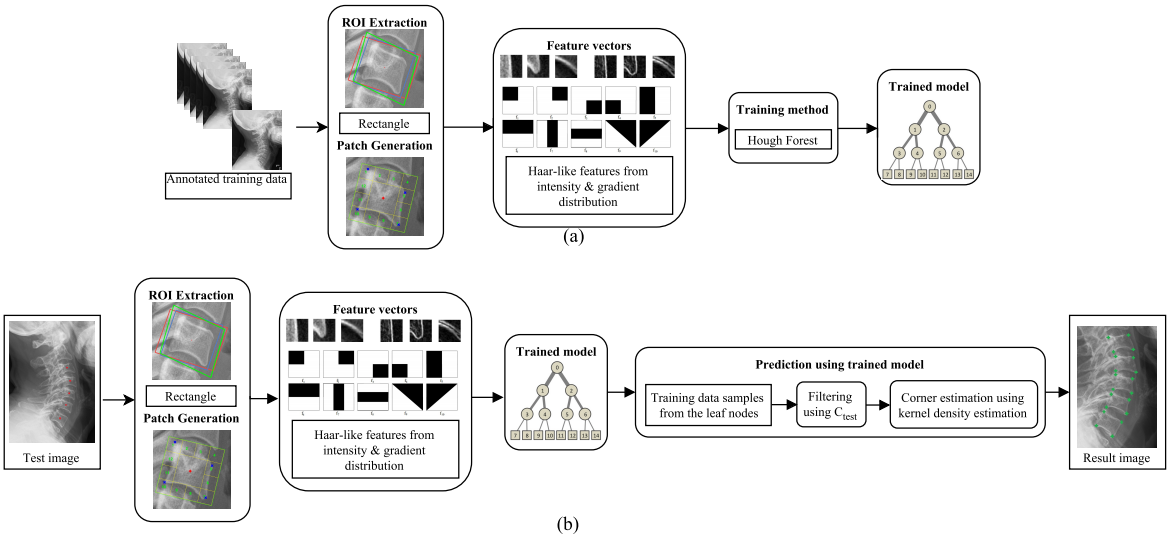


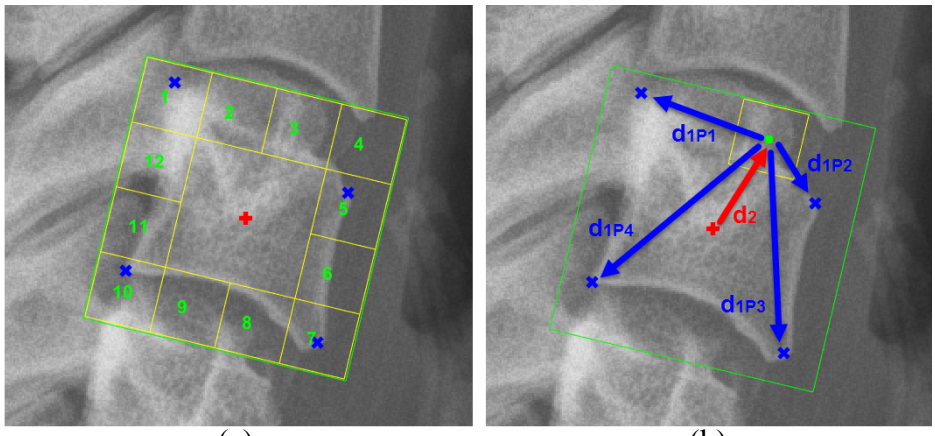
Fig. 5.6 (a) Training and (b) test flowcharts for the Hough forest-based vertebral corner detector.

data, each of which is associated with a class label and a vector.

Like any other machine learning-based frameworks, our Hough forest-based vertebral corner detection framework can be divided into two parts: training and testing; an overview is depicted graphically in Fig. 5.6. During training, the algorithm learns the relative position of the vertebral corners for different patches generated from the vertebra. The patches are generated from a region of interest (ROI) around the vertebra. These image patches are then converted into feature vectors using Haar-like features (Sec. 5.2.2). Training is performed by Hough forest where both classification and regression entropy are used together. The training process is summarized in Fig. 5.6a. Once the forest is trained, the framework can be used to predict corners for new vertebrae. At test time, a new image is provided with manually clicked vertebral centers. The ROIs are generated and patches are fed into trained forests, which then goes through a three-stage process to localize corners: forest prediction, filtering and corner estimation (Sec. 5.2.4). The test time process is summarized as a flowchart in Fig. 5.6b.

5.2.1 Patch Extraction and Labels

As mentioned in Sec. 5.1.1, three types of ROI geometry can be considered for evaluation. We have experimented with all three types of ROI in [71], the best performance was achieved



(a) (b)

Fig. 5.7 Hough forest training (a) class labels and (b) vectors.

by the rectangular ROI compared to the trapezoidal and square ROIs. Detailed results with all three ROI types are reported in Table A.3. Since the Hough forest framework does not depend on the classical corner and edge detection methods, the axis-aligned vertebral boundaries from the trapezoidal ROI do not improve the performance. After extracting the rectangular ROI, it is then divided into 16 equal-sized non-overlapping patches. Four center patches are discarded due to their homogeneous intensity distribution. Each of the boundary patches is associated with a class label (from 1 to 12) and five vectors. The class label (C_{patch}) represents the position of the patch within the ROI and four vectors ($\boldsymbol{d}_{1P1}, \boldsymbol{d}_{1P2}, \boldsymbol{d}_{1P3}$ and $\boldsymbol{d}_{1P4}$) point to four corners from the patch center and vector ($\boldsymbol{d}_2$) points to the vertebral center, as shown in Fig. 5.7. These image patches are converted to feature vectors and fed into a Hough forest algorithm for training.

5.2.2 Feature Vector

After creation, in order to train or test, the patches are converted into feature vectors. A detailed experimentation with different types of feature vectors was performed in [71] on Dataset A. The full description of these feature vectors and results can be found in Sec. A.3 and Table A.3, respectively. In this chapter, we only use the best performing feature vector: Haar-Mixed. To generate these feature vectors from the patches, we consider both the intensity and the gradient distribution of each patch. Gradients are calculated in horizontal and vertical direction. Then the root-mean-square (RMS) of the magnitudes are considered. Fig. 5.8 shows the intensity and gradient distributions of a few training patches. Both distributions contain complementary information which can be useful for training.

Fig. 5.8 Appearance of intensity and gradient patches.

Each of these distributions is then passed through ten Haar-like feature templates [106]. These templates are chosen based on the patch appearances in terms of intensity and gradient

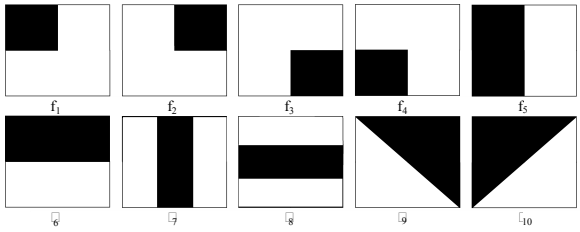


Fig. 5.9 Haar-like feature templates.

distribution (see Fig. 5.9). Each template returns a feature value when passed through an intensity (I) or gradient (G) patch based on the difference between the average intensity of the dark and bright area:

$$f_i = \bar{I}_{dark} - \bar{I}_{bright}, \tag{5.6}$$

$$g_i = \bar{G}_{dark} - \bar{G}_{bright}. \tag{5.7}$$

where $\bar{I}_x$ and $\bar{G}_x$ is the average value of intensity and gradient distribution of area denoted by x, respectively. Finally, feature vectors (H_v) are formed by the feature values from these feature templates:

$$H_v = [f_1, f_2, f_3, ..., f_{10}, g_1, g_2, g_3, ..., g_{10}], \tag{5.8}$$

5.2.3 Training

In contrast to the random forest method discussed in Sec. B.1 and 3.1.3, which can perform either regression or classification, Hough forest performs both together in the same forest. During training the algorithm requires training data, each of which is associated with a class label and a vector. In our case, patches generated from the vertebrae are converted into feature vectors (see Sec. 5.2.2). These feature vectors are considered as training data and corresponding class labels (C_{patch}) and vectors $\boldsymbol{d}_1$ are used to calculate the information gain (IG) using classification entropy (H_{class}) or regression entropy (H_{reg}). The choice of entropy at each node is random in Hough forest. The data flows down the tree until the maximum

tree depth (D) is reached or the number of elements in a node falls below a threshold ($nMin$).

The IG is calculated using Eqn. 5.9 which is the same equation described in Sec. 3.1.3:

$$IG = H(S) - \sum_{i \in \{L,R\}} \frac{|S^i|}{|S|} H(S^i), \tag{5.9}$$

where, S is a set of examples arriving at a node, S^L and S^R are the data that travel left (L) and right (R), respectively, and $H(S)$ is the entropy of the data S. Here, $H(S)$ can be either classification entropy, H_{class}, or regression entropy, H_{reg}. For Hough forest, at any node, this entropy is chosen at random. These can be calculated as below:

$$H_{class}(S) = -\sum_{c \in C} p(c) log(p(c)), \tag{5.10}$$

where C is the set of classes available at the considered node. In our case $C \subseteq \{1,2,3,...,12\}$.

$$H_{reg}(S) = \frac{1}{2} log((2\pi)^2 |\Lambda(D_1)|), \tag{5.11}$$

where D_1 is the set of $\boldsymbol{d}_1$ vectors arriving at the node and $\Lambda(D_1)$ is the covariance matrix of D_1. At each node, the optimum split (maximum IG) is chosen from a subset of all the possible splits. Each tree branch terminates at a leaf node. The leaf node contains the class labels (C_{patch}) and the $\boldsymbol{d}_1$ vectors of the patches that end up at that node.

5.2.4 Prediction

At the time of testing, image patches are generated using the manually annotated vertebral centers following the process described in Sec. 5.2.1. These test images patches are converted to feature vectors and fed into the trained forest. Each patch travels through each tree and reaches a leaf node. Each leaf node contains a set of $\boldsymbol{d}_1$ vectors and class labels (C_{patch}) from the training samples. As the image patches are created in a fixed manner based on the ROI, the class label of the test patch, C_{test}, is known at test time. Based on this known class label, a filtering stage discards all the $\boldsymbol{d}_1$ vectors belonging to classes other than C_{test} from the leaf

node training data samples:

$$\hat{\boldsymbol{d}}_{1_{tree}} = \hat{\boldsymbol{d}}_{1_{filtered}} = \{\forall \boldsymbol{d}_1 | c = C_{test}\}. \tag{5.12}$$

Each of these filtered $\boldsymbol{d}_1$ vectors is then combined with the corresponding $\boldsymbol{d}_2$ vector of that patch to find the corner location with respect to the vertebral center (see Fig. 5.7d):

$$\hat{\boldsymbol{d}}_{patch} = \{\hat{\boldsymbol{d}}_{1_{tree(1)}}, \hat{\boldsymbol{d}}_{1_{tree(2)}}, ..., \hat{\boldsymbol{d}}_{1_{tree(N)}}\} - \{\hat{\boldsymbol{d}}_{2_{patch}}, \hat{\boldsymbol{d}}_{2_{patch}}, ..., \hat{\boldsymbol{d}}_{2_{patch}}\}. \tag{5.13}$$

Then the final corner position $\boldsymbol{d}$ is predicted using a two-dimensional kernel density estimation (KDE) process over the collection of vectors coming from all the patches from the same vertebrae [107]:

$$\hat{\boldsymbol{d}} = KDE(\{\hat{\boldsymbol{d}}_{patch(1)}, \hat{\boldsymbol{d}}_{patch(2)}, ..., \hat{\boldsymbol{d}}_{patch(12)}\}). \tag{5.14}$$

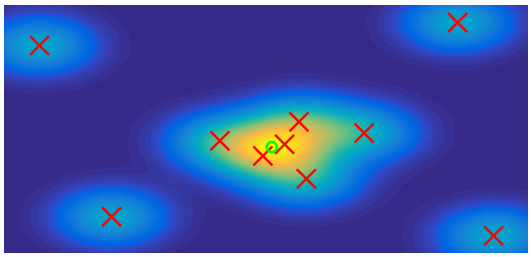


Fig. 5.10 KDE: The heat map denotes the confidence of the aggregated probability distribution $p(\boldsymbol{d}_{1\_out})$. Red crosses indicate the positions of the input $\boldsymbol{d}_{1\_in}$ vectors and green circle represents the maxima of $p(\boldsymbol{d}_{1\_out})$ and output vector $\boldsymbol{d}_{1\_out}$.

The *KDE* process takes a set of 2D vectors and regresses a possible output. Here at each vector ($\boldsymbol{d}_{1\_in}$) location a zero-mean 2D Gaussian distribution with isotropic variance σ_k^2 is set:

$$p(\boldsymbol{d}_{1\_in}) = \mathcal{N}(\boldsymbol{d}_{1\_in}, \sigma_k^2) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp^{-\frac{(\boldsymbol{x}-\boldsymbol{d}_{1\_in})^2}{2\sigma_k^2}}, \tag{5.15}$$

After placing a distribution at each of the vectors, a final probability map is calculated by adding all distributions as:

$$p(\boldsymbol{d}_{1\_out}) = \frac{1}{n}\sum_{i=1}^{n} p_i(\boldsymbol{d}_{1\_in}) \tag{5.16}$$

Finally, the maxima of this aggregated distribution are located and considered as the output vector ($\boldsymbol{d}_{1\_out}$). The process is also graphically summarized in Fig. 5.10.

5.2.5 Parameters

As for any random forest method, there are a few hyper-parameters that should be decided before training: number of trees ($nTree$), maximum allowed depth of a tree (D), minimum number of elements at a node ($nMin$), number of variables to look at in each split nodes ($nVar$) and number of thresholds ($nTresh$) to consider per variable. Apart from the random forest parameters, the kernel density estimation (KDE) function requires a given bandwidth (BW) which is the variance (σ_k^2) in Eqn. 5.15. The value of these parameters were optimized using a sequential parameter optimization process on Dataset A. The process is described in Sec. A.4 and the parameters for the Hough forest trained for this dissertation are reported in Table A.2.

5.3 Deep Probabilistic Vertebral Corner Localization

In the previous chapter, we have proposed a probabilistic spatial regressor network (PSRN) for the localization of vertebral centers. The network used a pixel-wise loss function for training which does not take into account the properties of a valid probability distribution like the summation must integrates to one. In this section, we propose major improvements to the PSRN by introducing a new spatial normalization layer and a novel probabilistic loss function. The improved network will be used in a novel framework to localize vertebral corners for lateral cervical X-ray images.

This section is structured as followed. First, in the next subsection, we discuss the ground truth used for training the improved PSRN. Followed by a summary of the whole corner localization framework. The new spatial normalization layer is discussed next, followed by the introduction of the novel loss function. The section ends with the discussion of the post-processing step needed to convert the probability maps into localized corners on the full resolution image.

5.3.1 Ground Truth

As discussed earlier, each vertebral body in the dataset was manually annotated for the vertebral boundaries and centers by expert radiographers. Two examples with the corresponding manual annotations are shown in Fig. 5.11a and 5.11b. The corner point of a cervical vertebra is often not well defined because of the smooth transition of the vertebral boundary. Thus manually clicked corner points vary substantially from expert to expert and from vertebra to vertebra. This variation makes it difficult for machine learning algorithms to learn a single deterministic model for corner prediction. This led us to consider probability distributions to represent the corners instead of a single point. The probability distribution is generated by applying the same 2D anisotropic Gaussian distribution used for generating probabilistic distributions for vertebral centers in Chapter 4. One distribution is added at each manually clicked corners. However, the full Gaussian distribution was not kept. Given the fact that the location of the corner can only vary along the vertebral boundaries, we only keep those values on the boundary. The resulting distributions over the image space can be seen in Fig. 5.11a and 5.11b.

To create the training image patches and the ground truth probability distributions from the image-level data-pairs, we follow the same grid-based multi-resolution multi-orientation procedure mentioned in Sec. 4.2. As a result, we have 66,600 training patches. Similarly, the patches were resized to a size of 64×64 pixels at which the proposed network is trained. A few vertebral image patches are shown in Fig. 5.11c with their corresponding ground truth distributions.

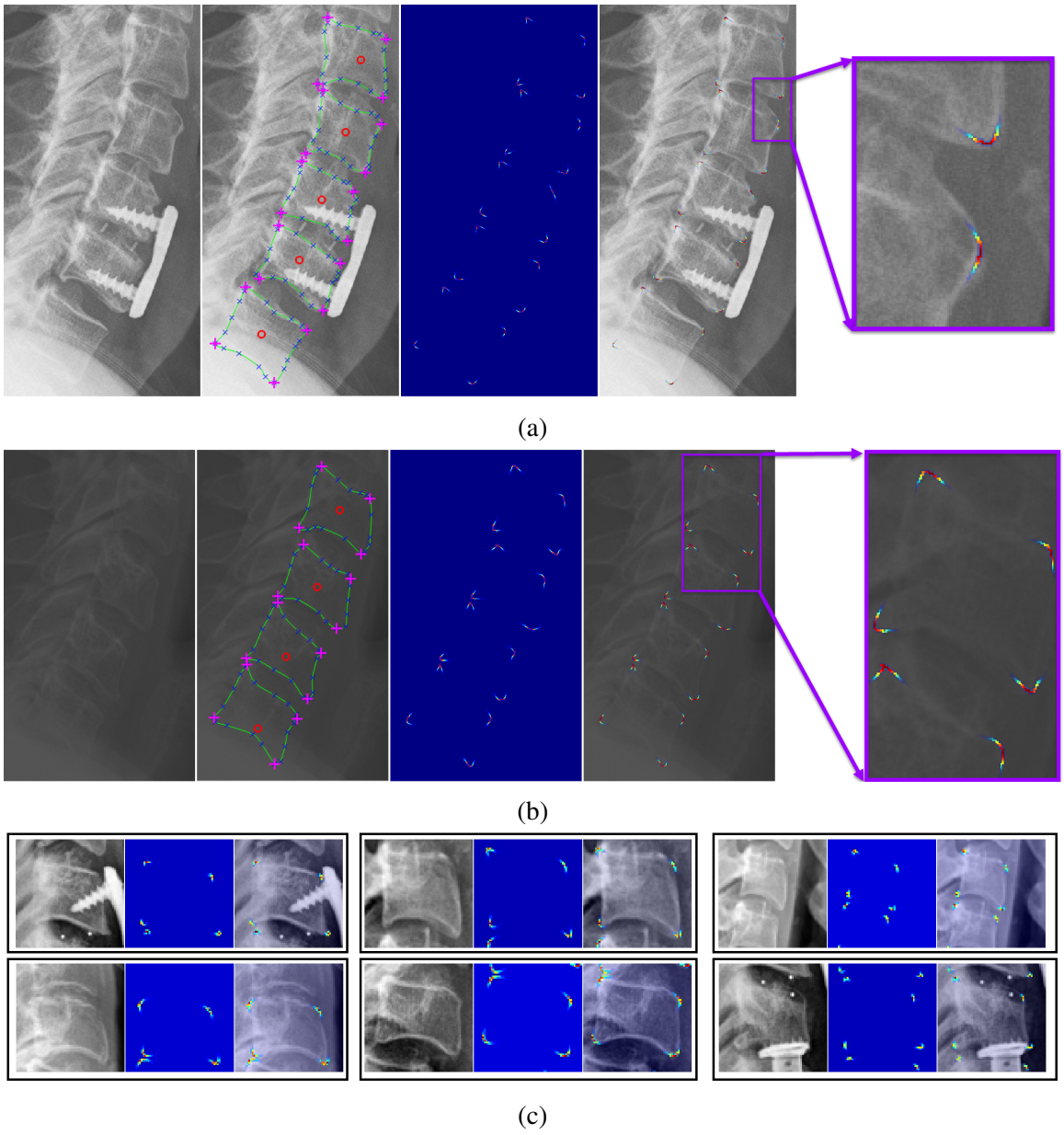


Fig. 5.11 (a-b) Zoomed X-ray images (left), manual annotations (middle-left): center (o), manually clicked boundary points (×), corner points (+) and splined vertebrae curve (−), heatmap of the probability distributions for the corners (middle-right) and heatmap overlayed on the X-ray image (c) training image patches and corresponding patch-level ground truth probability distributions.

5.3.2 Framework

The overview of the improved PSRN-based corner localization framework is summarized in Fig. 5.12. Like the previous two corner localization frameworks, we assume the ver-

tebral centers are given. From these manually clicked center points, a set of patches is generated. Each of these image patches is sent forward through the improved probabilistic spatial regression network described in Sec 5.3.3. The network generates patch-level spatial probability distributions for corners in each patch. The patches are then transformed back on the original image space using their known location, orientation and size. Finally, the vertebral corners are localized from the accumulated patch distribution. These last steps are part of the post-processing phase and described in Sec. 5.3.4.

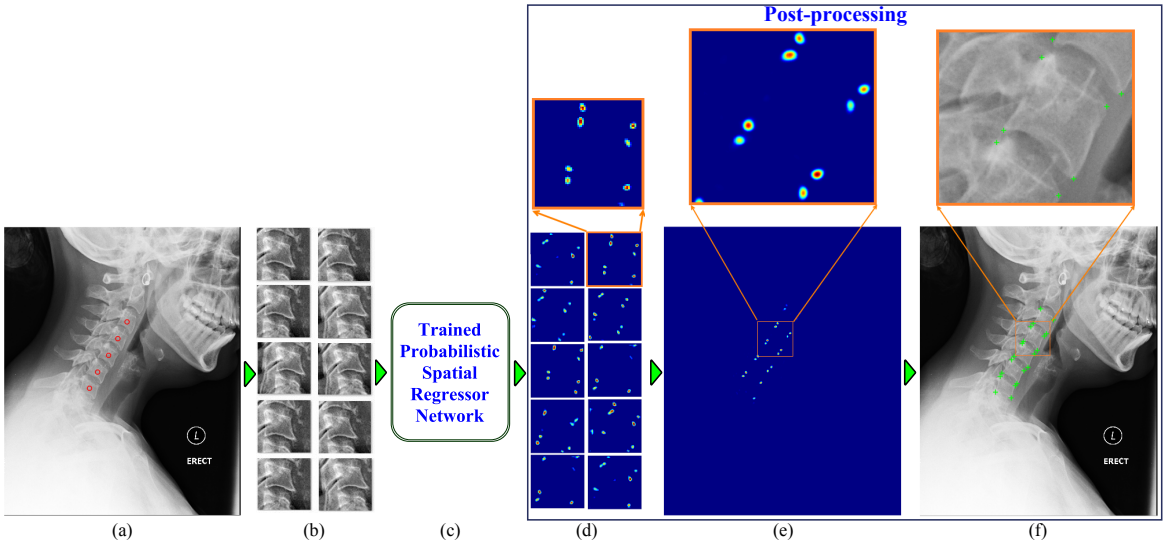


Fig. 5.12 Framework block diagram (a) input image with manually clicked vertebral centers (b) image patches (c) proposed network (d) patch-level predictions (e) image-level prediction (f) localized corners.

5.3.3 Network

The network in Fig. 5.12c is trained on the training image patches to learn a mapping for predicting spatially distributed probabilities for the vertebral corners. We have the same architecture used in the original PSRN proposed in Chapter 4 with the exception of the final convolutional and the softmax layer. Previously, the activation from the last convolution layer had two channels. In this chapter, the final activation of the network is a single channel output which will be compared with a 2D spatial probability distribution over the input image space.

Thus, we introduce a new layer to convert the final activation into a valid spatial probability distribution. One choice for this layer could have been doing softmax-like operation spatially, but as our input patches have multiple corners with high probabilities (Fig. 5.11b), the exponential nature of the softmax function often results in a single localized corner. Thus, we introduce a new spatial normalization layer, which converts the final activation of the network into a valid spatial probability distribution using a simple mathematical operation by forcing the minimum to be zero and the integration to be unity. The network is shown in Fig. 5.13. The total number of parameters in the network is 24,237,633 which is a few parameters less than the original PSRN. This is because of the difference in the last convolution layer which now produces a single channel output and contains half of the parameters compared with the network proposed in Chapter 4.

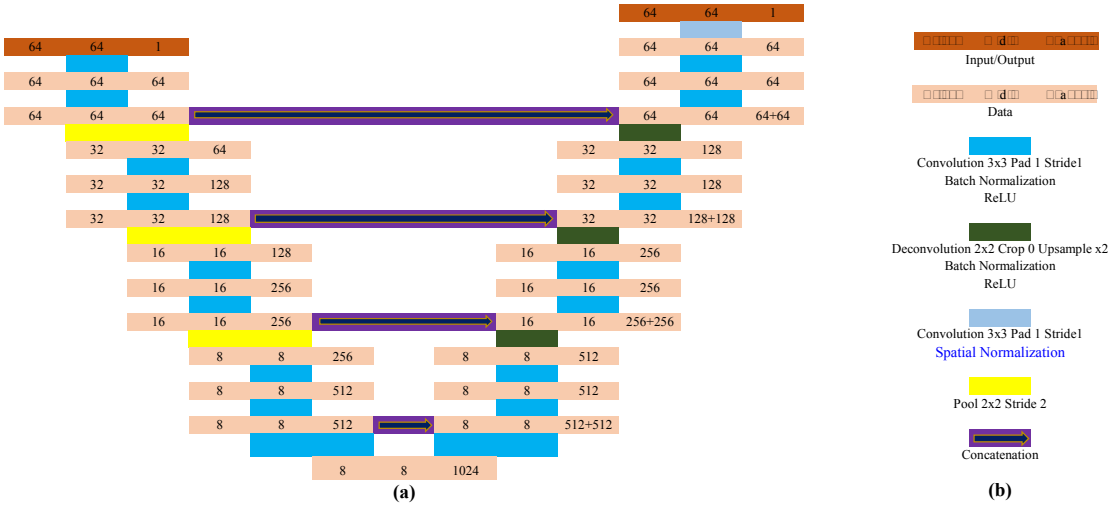


Fig. 5.13 (a) Network architecture (b) legend.

5.3.3.1 Training

Given a dataset of training patch (x) - ground truth probability distribution (y) pairs, training a deep neural network means finding a set of optimized parameters $\hat{\boldsymbol{W}}_o$ that minimizes a loss function, L:

$$\hat{\boldsymbol{W}}_o = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} L(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W}), \tag{5.17}$$

where N is the number of training examples and $\{x^{(n)}, y^{(n)}\}$ represents n-th example in the training set with corresponding ground truth corner probability distribution. We desire a network where the last layer, the spatial normalization layer, generates a valid probability distribution. Let $P(x)$ be the output of the network for the input x. We define a differentiable loss function that measures the similarity between the ground truth and prediction distributions. We have previously used Bhattacharyya coefficient (BC) to evaluate the similarity between two probability distributions. BC is zero if there is no similarity and increases to a maximum of unity as the similarity increases. Based on this knowledge, we define the loss function per input sample as following:

$$L(\{x,y\};\boldsymbol{W}) = -2BC(y,P(x)), \tag{5.18}$$

$$BC(y,P(x)) = \sum_{i\in\Omega_p} \sqrt{y_i P_i(x)}, \tag{5.19}$$

where Ω_p represents the pixel space and $P_i(x)$ is the probability at point x_i. Eqn. 5.18 is easily differentiable with respect to the input of the loss layer, $P(x)$. The pixel-wise derivative of Eqn. 5.18 with respect to $P_i(x)$ is used for the backpropagation of the loss during training:

$$\frac{\partial}{\partial P_i(x)} L_i(\{x,y\};\boldsymbol{W}) = -\sqrt{\frac{y_i}{P_i(x)}}. \tag{5.20}$$

The network is trained on the 66,600 image patches generated from the training images. The network is trained on a system with a NVIDIA Pascal Titan X GPU for 17 epochs with a batch-size of 25 image patches. The training took approximately 42 hours.

5.3.4 Post-processing

At the test time, given a test image and corresponding manually clicked vertebral centers, we create test patches following the same procedure described in Sec. 5.3.1. Each patch is then resized to 64×64 pixel and passed forward through the trained network which generates a patch-level spatial probability distribution. These probability distributions often have noise

and residual probabilities in the background. The residual probabilities are a result of the combined effects of the padding operations of the convolutional layers of the network and the introduced spatial normalization layer. Throughout the network, we have used zero padding in the convolutional layers to keep the output size similar to the input. This zero padding results in a lower value at the border of the output. As our spatial normalization layer simply forces the minimum to be zero, the border area of the final activation becomes zero and the rest of the background assumes small residual values. The effect can be seen in Fig. 5.14c, where the patch borders are visible and have probability values near zero. The range of values for the residual probability in each patch can be found by analyzing its histogram. In the next step, we remove these residual probabilities from the background and re-normalize the distributions to have a range between 0 and 1. These patch-level predictions are then transformed back on the original image space using their known size, orientation and location. The affine transformation process is the same as described in Sec. 4.3.3. The resultant distribution for each vertebra is then weighted by a prior probability distribution

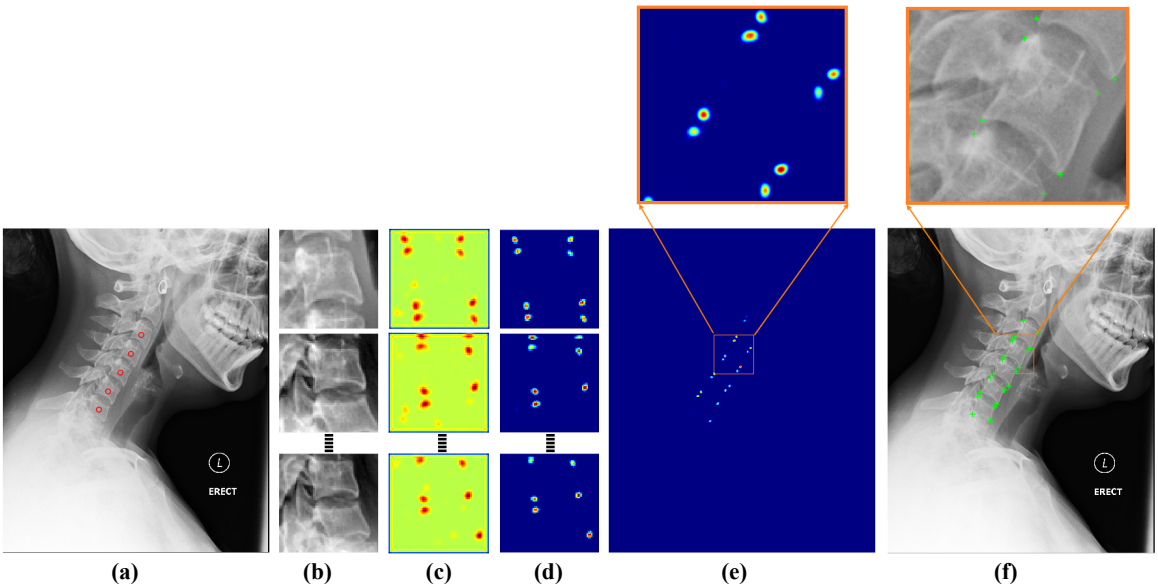


Fig. 5.14 Post-processing (a) input image with manually clicked vertebral centers (b) extracted image patches to be sent forward through the network (c) patch-level prediction results from the network (d) patch-level predictions after removing residual probabilities (e) image-level prediction (f) localized corners.

of corners for that vertebra learned from the training examples. Finally, on the original image space, the vertebral corners are localized by finding the maximum in each of four quadrants of each vertebra. The quadrants are defined using the manually clicked center points. The process is similar to the Harris-based naive Bayes corner detector discussed in Sec. 5.1.2. In case the algorithm does not find any probability distribution for a corner, which may be a result of occlusion, surgical implant and/or low contrast, it uses this prior distribution of corners determine a possible corner location. In the example of Fig. 5.14e, we show that the bottom-left corner is missing on the original image space because of very low contrast. The complete process of corner localization starting from a test image including the post-processing steps is summarized in Fig. 5.14.

5.4 Results and Discussion

We first evaluate the performance of the improved PSRN at the patch level by reporting the Bhattacharyya coefficient (BC) between each predicted spatial probability map and its corresponding ground truth probability for the 90,480 image patches generated from our 172 test images. The BC between two probability distribution is defined in Eqn. 5.19. An average BC of 0.9794 has been achieved over the test patches. A Bhattacharyya coefficient of 1 indicates a perfect match between two probability distributions. The histogram plot of the BC metrics is shown in Fig. 5.15. It can be noted that the BC is always in the

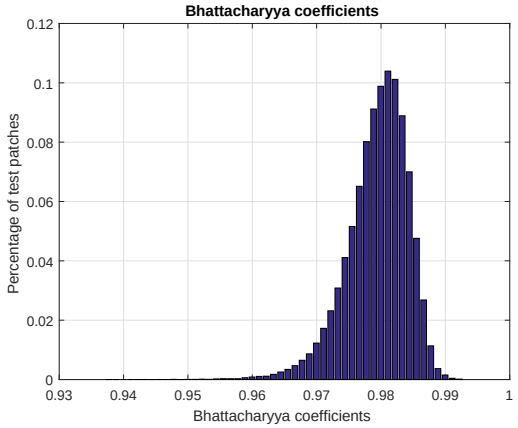


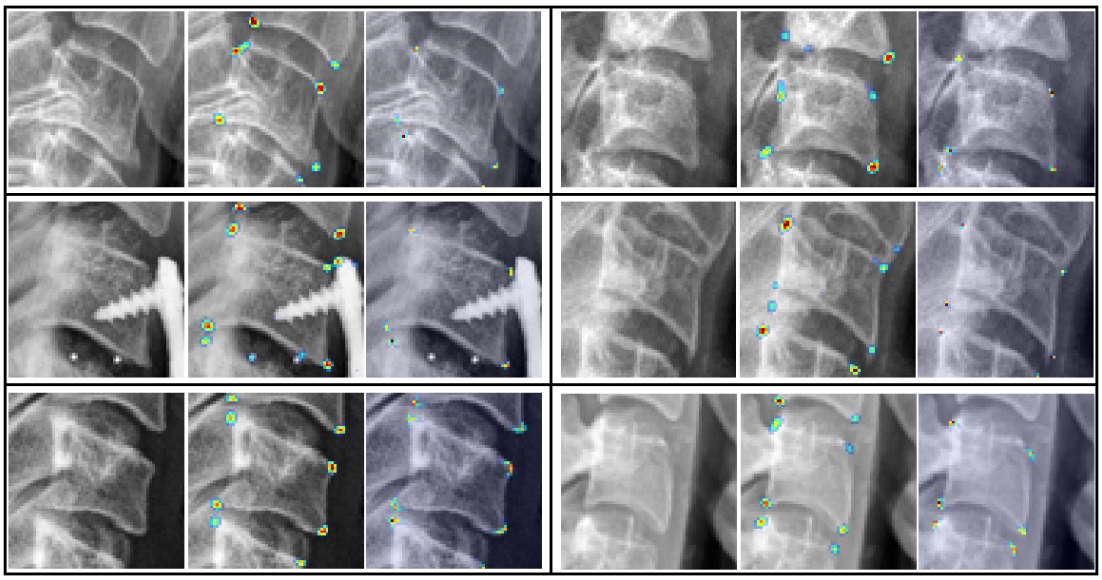
Fig. 5.15 Histogram plot of Bhattacharyya coefficients for patch-level predictions for PSRN.

high range of 0.96 to 0.99 for all the test patches. However, the BC has limitations in measuring the similarity between two distributions. Since the majority pixels on the ground truth probability distribution have zero values, it doesn't penalize if a small prediction probability is present in those places thus BC can be high even if the prediction looks different. This is why BC stays high even when there is border effects and small residual probabilities in the background (Fig. 5.14c). Although these results look different from the patch-level ground truth (Fig. 5.11b), the BC between them can be high as long as the locations of the maximum probabilities match. As our loss function is based on this metric, the trained network failed to remove the residual probabilities in the background (Fig. 5.14c). However, despite this limitation, the network robustly learns to predict high probabilities at the corner locations. A few patch-level qualitative results from our test dataset and also from the NHANES-II dataset are shown in Fig. 5.16a and 5.16b, respectively. The predicted probability heatmaps shown in these examples are after removing the residual background probabilities (Fig. 5.14d). Fig. 5.16a illustrates that the predicted probability distributions often have a higher variance than the corresponding ground truth. This could be an effect of multiple upsampling operations in the expanding path of the U-Net architecture. Instead of using deconvolutional layers to achieve upsampling, a sub-pixel convolution could be used which may reduce this effect [108].

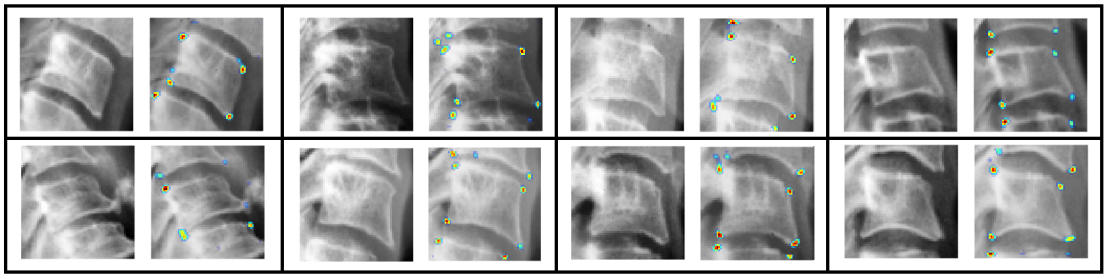
After the post-processing phase, the corners are localized on the original image. The HarrisNB and HoughF also localize the corner on the original image space. So, we can now compare all three proposed corner localization frameworks discussed in this chapter. The ground truth corners and the vertebral boundary curves are known from the manual annotations. We report two metrics:

1. Point to point ($P2P$): Euclidean distance between the predicted corner to manually annotated corner in millimeters (mm),

2. Point to curve ($P2C$): Distance between the predicted corner and manually annotated vertebral boundary curve (green lines in Fig. 5.11a). This metric is defined in Eqn. 5.21.



(a)



(b)

Fig. 5.16 Qualitative analysis of the predictions from PSRN (a) patches from the test dataset: input image patch - PSRN prediction (overlayed on the input patch) - ground truth distribution (overlayed on the input patch) (b) vertebra patches collected from NHANES-II dataset: input image patch - PSRN prediction.

$$P2C(\hat{C}, S_{gt}) = min\{D(\hat{C}, \boldsymbol{x}) : \boldsymbol{x} \in S_{gt}\}, \tag{5.21}$$

where $\hat{C}$ is the predicted corner, S_{gt} is set of points in the manually annotated vertebral boundary curve and $D(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between the point $\boldsymbol{x}$ and $\boldsymbol{y}$. For both, $P2P$ and $P2C$, lower values represent better results. The $P2P$ is a special case of $P2C$ where S_{gt} only contains a single point, i.e., the ground truth corner. The $P2C$ is more appropriate than $P2P$ when the corner area is smooth and determining a corner depends on human interpretation. We also report a third metric called fit failure [38]. We define fit failure as the

percentage of corners with a *P2P* error greater than 3 mm. The median, mean and standard deviation of these metrics over the 3,188 corners of the test dataset are reported in Table 5.1.

	Point to point (*P2P*) (mm)					Point to curve (*P2C*) (mm)		
	Median	Mean	Std	Fit failure (%)		Median	Mean	Std
HarrisNB	2.15	2.70	2.20	34.91		0.53	0.95	1.10
HoughF	1.99	2.48	1.98	27.13		0.88	1.12	1.07
PSRN	**0.99**	**1.54**	**1.74**	**11.70**		**0.35**	**0.58**	**0.76**

Table 5.1 Euclidean distance between predicted and manually annotated corners.

In terms of *P2P* error, the HoughF performs better than HarrisNB. The improved PSRN-based method achieved a 38% relative improvement (mean *P2P* 1.54 mm vs 2.48 mm) compared to HoughF. The median error for the proposed method achieved a large drop of 1 mm from the HoughF method. The percentage of vertebrae with fit failure also decreased by more than 15 percentage points. The cumulative *P2P* errors for the compared methods are shown in Fig. 5.17.

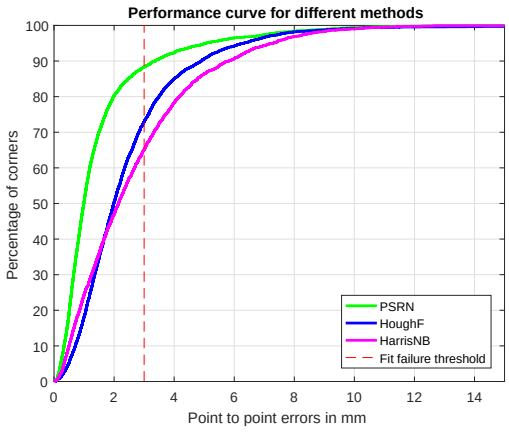
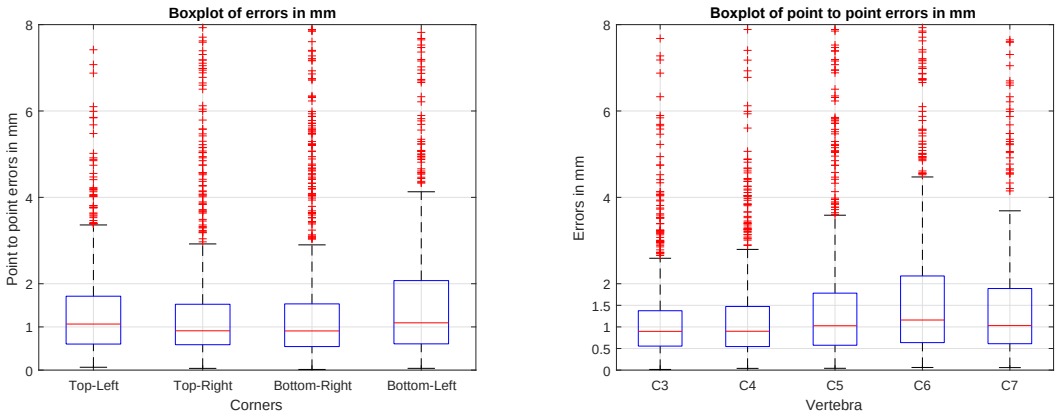


Fig. 5.17 Cumulative error curve for different corner localization methods.

It can be seen that the improved PSRN-based framework outperforms the other methods by a large margin. In terms of the *P2C* error, the HarrisNB outperforms the HoughF method. We believe this is because of the edge detection process utilized in the HarrisNB method, which forces the detected corners to be near vertebral boundaries. But the PSRN method still outperforms the HarrisNB method with a relative improvement of 39% in terms of the



(a) (b)

Fig. 5.18 (a) Boxplot of the errors for different corners and (b) boxplot of the errors for different vertebrae for PSRN-based corner localization method.

mean error. However, it can be noted that the standard deviation of the PSRN method is still somewhat high. This is because of the complexity in our test dataset. As we mentioned our data is not collected under a controlled environment, thus it contains challenging images full of clinical conditions, bone implants, image artefacts and contrast variations. Some of these challenging cases are shown in Fig. 5.19c and 5.19d. The boxplots of Fig. 5.18a and 5.18b also reveal that there are many outliers, most of which belong to the corners from these challenging cases. In Fig. 5.18a, we show a boxplot of the *P*2*P* errors for different corners for the PSRN method. It can be noted that the corners on the right (or anterior side) have comparatively lower errors than the left side (posterior). The probable reason behind this might be that the anterior side of the cervical spine often has better image contrast than the posterior side which contains posterior spinal arches and processes. The vertebral corners are also closer in between two vertebrae on the posterior side. The boxplot of corners for different vertebrae reveals that C3 and C4 have a lower error from the rest of the spine (Fig. 5.18b). As we go down the spine (from C3 to C7) the variation of the vertebrae increases as well as the image quality and contrast decrease to some extent, making it harder for the algorithm to predict corners.

Some vertebra-level results for all the compared methods are shown in Fig. 5.19. In the first row, Fig. 5.19a, we show some relatively easy cases where predictions of all the methods are comparatively good. In Fig. 5.19b, we show some more easy cases, where the improved PSRN-based methods outperformed the other methods. Some challenging cases with bone implants, low contrast, image artefacts and clinical condition are shown in Fig. 5.19c, where the improved PSRN method has produced good results. Finally, in Fig. 5.19d we show some more challenging cases where almost all the methods have failed.

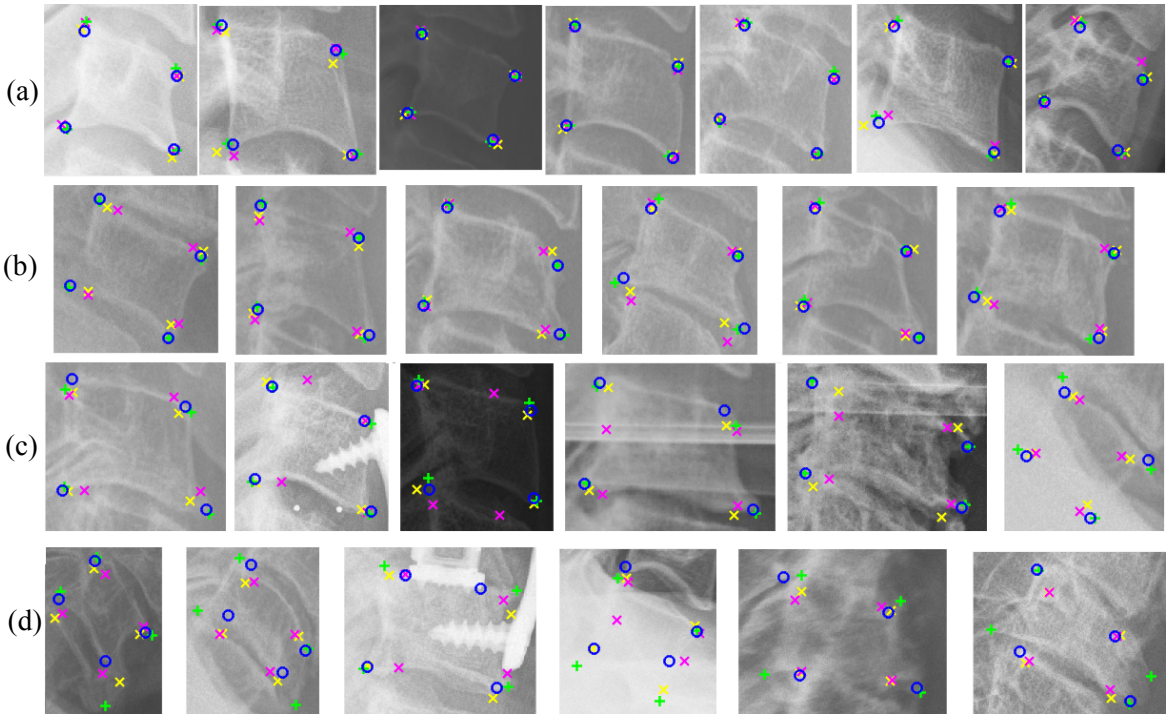
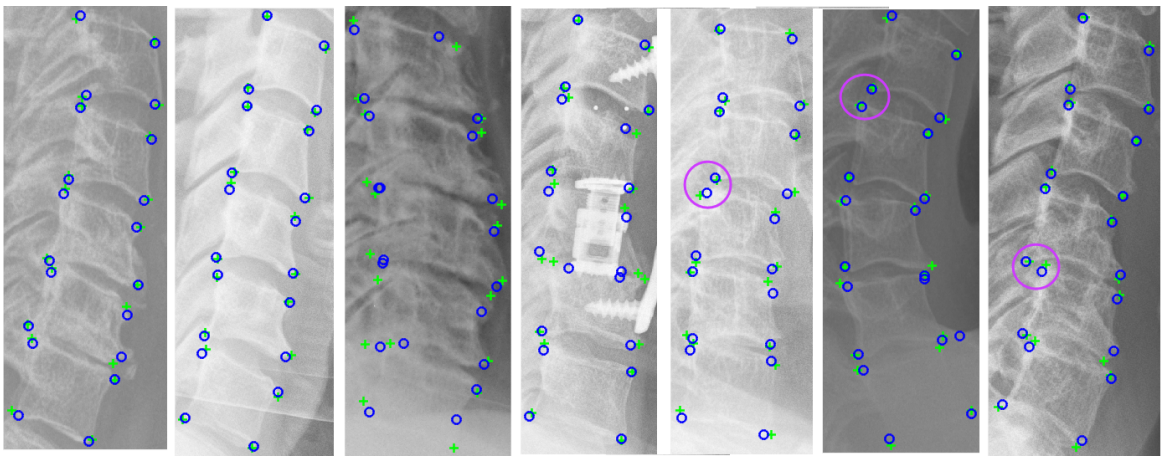


Fig. 5.19 Vertebra-level corner predictions: ground truth (+), PSRN (o), HarrisNB (×) and HoughF (×).

A few qualitative results with the full cervical spine with the predictions from the PSRN-based corner localization framework are shown Fig. 5.20. Fig. 5.20a and 5.20b show two examples of healthy spines where the prediction results are near perfect for almost all the corners. A severe case of bone loss, osteoporosis and low image contrast is shown in Fig. 5.20c. It can be seen even with such severe conditions, the prediction results are considerably correct. Fig. 5.20d shows an example with surgical bone implants, which affected some of the prediction results, especially at the C5-C6 area. However, because

of the patch-based framework, other corners are well detected. A few results for spinal misalignment (spondylolisthesis) are reported in the rest of the Fig. 5.20. Fig. 5.20e shows subluxation (partial dislocation) between C4-C5, Fig. 5.20f between C3-C4 and Fig. 5.20h between C5-C6. The predicted corners can be used to determine these injuries automatically.



(a) (b) (c) (d) (e) (f) (h)

Fig. 5.20 Vertebral corner prediction using PSRN-based framework: ground truth (+), PSRN-based corner prediction (o). The magenta circles (O) indicates the subluxation injuries.

5.5 Conclusion

In this chapter, we proposed three novel vertebral corner localization methods. The first method, HarrisNB, used classical corner and edge detection methods with a prior knowledge of possible corner locations from a training dataset. This method can be considered as a local method where the actual appearance of the corner plays an important role. However, because of the amount of noise present in the X-ray images, the local information can be corrupted. The HoughF framework tries to overcome this local dependency by localizing corners using votes generated by patches extracted from other parts of the region of interest. However, because this method has less dependency on the local properties, sometimes corners are localized in homogeneous image regions away from the actual vertebral boundaries. Finally, we have proposed major improvements to the probabilistic spatial regressor network by introducing a novel loss function and new spatial normalization layer. The improved

PSRN was incorporated in a corner localization framework which outperformed the first two frameworks by a large margin and achieved a median error of less than a millimeter.

The methods proposed in this chapter can be adapted to other domains. The HarrisNB framework proposes a methodological way of removing unnecessary corners detected by the classical Harris corner detector by incorporating a prior knowledge and several edge detection methods. This can be particularly useful for images with high amount of noise. The method is fast and doesn't require extensive training like other machine learning techniques. The proposed HarrisNB is also modular which means if necessary for certain applications, one can modify/improve the framework by removing/updating certain modules of the framework (Fig. 5.2).

The HoughF framework was inspired by the success of object detection method proposed in [91]. Unlike the HarrisNB framework which can only detect corners, the HoughF framework can easily be adapted to localize any other features and objects in the image. One only needs to update the vectors related to the training image patches with new vectors pointing towards the new features or objects of interest. The rest of the framework can stay the same.

In the conclusion of the previous chapter, we have already discussed the how the probabilistic spatial regressor network (PSRN) can be used for localization of image landmarks in the other domains of computer vision. In addition to those applications, the PSRN can also be used for probabilistic detection of other image features like edges and object boundaries. We will expand on this in the following chapter.

Like vertebral corners, vertebral boundaries can also play a vital role in the automatic detection of vertebrae related clinical conditions like subluxation and compression fracture. In the next chapter, we focus our attention on vertebral boundary detection. We will use our experience of the dense classification network and the probabilistic spatial regressor network (PSRN) to solve the boundary detection problem. The PSRN used in this chapter

suffered from residual background probability problem. It was a combined effect of the convolution operation with zero padding and the proposed spatial normalization layer. In the next chapter, we will propose another spatial normalization layer to solve the residual background probability problem.

Chapter 6

Boundary Detection and Segmentation

6.1 Introduction

Vertebral boundaries are arguably the most characteristic property of the X-ray images. It separates the vertebral body from the surrounding. The image intensities at the vertebral boundary are expected to change. However, in X-ray images, the change in the intensity along the edge of the vertebral body is not always apparent. In this chapter, we propose machine learning-based solutions to detect vertebral boundaries and segment vertebral bodies.

Boundary detection can be considered as a selective edge detection problem. An edge in the context of computer vision is defined as any sharp change in the image intensities. Edge detection is a fundamental problem in computer vision. Literature on this topic includes some of the seminal work from the 80's [109–113] to the recent state-of-the-art work like [78, 114, 115]. While the earlier research in the field focuses on finding suitable filters [111, 116, 117] and/or tackle the problem using variational approach [112], recent approaches in the literature are data-driven and designed as supervised machine learning problems pioneered by [118]. One of the benefits of these supervised methods is that the edge detection can be selective. Potentially, it is possible to train a machine learning model which identifies edges along the boundaries of specific objects while ignoring other edges in the images based on interest. In our case, we would like our model to track vertebral

boundaries while ignoring the edges from vertebral extensions and other image artifacts.

Most of the recent articles on boundary detection use deep dense classification [114, 115]. Following this trend and based on the success of deep networks in the previous chapters on vertebrae related problems, in this chapter, we propose two approaches to detect vertebral boundaries. First, a dense binary classification approach where pixels under the manually annotated vertebral boundary curve are considered as the foreground and others as the background. Then an UNet architecture is trained to classify each pixel in a vertebral image patch as a boundary pixel or a background pixel. Second, leveraging the method from the previous chapter, we take a probabilistic approach to boundary detection. Instead of representing the vertebral boundary as foreground and neighboring pixel as background, the vertebral boundary pixels are assigned a high probability of being an edge, and immediate neighboring pixels are also assigned a small probability of being an edge. The motivation behind this is that the manual annotation of vertebral boundary could be erroneous. The vertebral boundary is manually annotated by 20 sparse points per vertebra, which is then splined to form a continuous curve. Due to this operation, the curve may fall outside of a boundary pixel. To predict a spatial probabilistic map of the vertebral boundary, we train our probabilistic spatial regressor network on the vertebral image patches. We also propose an incremental improvement to this network where the spatial normalization layer has been modified to overcome the residual background problem suffered in the previous chapter.

Unfortunately, vertebral boundaries in X-ray images often lack visibility due to noise, clinical conditions and low bone densities. Furthermore, the presence of the vertebral extension in the posterior side of the vertebra makes it challenging to visualize vertebral boundary. These issues make the vertebral boundary detection problem very challenging. Thus, along with vertebral boundary detection, we also propose a novel method for segmenting the vertebral body. The expectation is that when trained to segment the whole vertebral body, the machine learning model will have better opportunity to learn the topological properties of a

vertebra.

Previous work in vertebral body segmentation has largely been dominated by statistical shape model (SSM)-based approaches [24, 38, 41, 46, 48, 49, 57, 69, 119]. These methods record statistical information about the shape and/or the appearance of the vertebrae based on a training set. Then the mean shape is initialized either manually or semi-automatically near the actual vertebra. The model then tries to converge to the actual vertebral boundary based on an ASM search procedure. Recent work utilizes random forest-based machine learning models in order to achieve shape convergence [38, 49, 57, 69]. In contrast to these methods, we propose a deep dense classification network-based method for vertebra segmentation. Instead of predicting the shape of a vertebra, our framework predicts the segmentation mask for a vertebral image patch. The ground truth here is a binary segmentation map where the foreground is defined as the whole vertebral body. A dense classification network is then trained with a standard loss function to learn the mapping between this ground truth and corresponding input vertebral image patch. The standard dense classification loss function computes the cross-entropy loss in a pixel-wise manner which does not encourage prediction of vertebrae-like shapes. Shape characteristics have long been used for medical image segmentation problems [120–123]. Medical image modalities, including X-rays, often produce noisy footprints of anatomical body parts, where segmentation problem must rely on the shape information to produce reliable results. Since vertebrae in lateral X-ray images have distinct shapes, we want to encourage our proposed network to predict vertebrae-like structures. However, combining shape information in a dense classification network is not straightforward. We try to solve this issue by introducing a novel shape-aware term in the loss function of the dense classification network.

The key contributions of this chapter are:

1. An innovative spatial probabilistic approach to boundary detection problem.
2. A new histogram-based normalization layer to solve the residual background problem of the probabilistic spatial regressor network.

3. A qualitative comparison of boundary detection performance by the dense classification network and the spatial probabilistic regressor network.

4. Introduction of a novel shape-aware term in the loss function of a deep dense classification network which learns to preserve the shape of the vertebral body.

6.2 Overview

In the next section, we describe the input image patches and the corresponding boundary detection and segmentation ground truth for training the deep networks. The network architectures and the relevant loss functions are explained in Sec. 6.4. A brief discussion of the compared algorithms and the quantitative metrics has been reported in Sec. 6.5. The results are discussed in Sec. 6.6 followed by the conclusion of the chapter in Sec. 6.7.

6.3 Ground Truth

We use the same training and test image split used in the previous chapters. However, we apply a different data augmentation scheme based on the patch extraction process during inference. To extract the image patches at the test time, one approach could be to use a sliding window technique. However, since the vertebra position, size and orientation vary a lot in our test dataset, the patch extraction would need to be performed all over the image space, possibly with overlapping patches and multiple scales and orientations increasing the processing time during inference. After we get all the predictions, other points to consider would be, how to reconcile between multiple predictions for the same vertebra and what to do with partially visible vertebra image patches. To avoid these complications and since we have already localized the vertebral centers and corners in the previous chapters, we propose to use them for an automatic test vertebrae patch extraction process. So, in this chapter, we assume that the approximate center point of the vertebra is known. For training images, these vertebral centers are provided by our clinical partner as mentioned in Sec. 2.3. At the test time, the user can first apply our spine localization algorithm of Chapter 3 to localize

the spine. Then the center localization framework of Chapter 4 can be applied to predict vertebral centers. One can also use the corner localization framework of Chapter 5 to localize the corners, and then compute the refined center location as the centroid of the four localized corners. However, in this chapter, to test the performance of the boundary detection and the segmentation algorithm independent of all the previous frameworks, we use the manually labeled vertebral centers. The fully automatic process will be discussed later in Chapter 8.

For data augmentation, we extract vertebral image patches from the training images with five different scales and nine different rotations/orientation. Based on the manually annotated vertebral center points, we compute the mean vertebral axis using the orientation vector described in Sec. 2.4.2. From this mean vertebral axis, we rotate the vertebrae from -20° to +20° with a step of 5° and for each rotation, we compute a base vertebral image patch size based on the distance from the center to the farthest point in the annotated vertebral boundary. The base size makes sure that the complete vertebra is inside the extracted vertebral image patch for every rotation. From the base patch size, we increase the size by 1 to 5 mm with a step of 1 mm to extract vertebral image patches with five different scales for each of the nine rotational angles. By augmenting the data using this approach, we end up with 26,370 vertebral image patches from training dataset of 124 images containing 586 vertebrae. All these extracted patches are resized to a common size of 64×64.

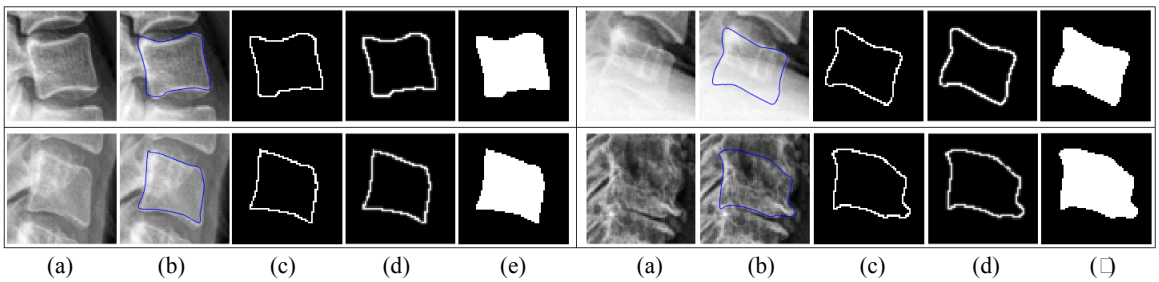


Fig. 6.1 Ground truth for edge detection networks (a) input vertebrae (b) manually annotated vertebral boundary (c) binary ground truth for boundary detection and (d) probabilistic ground truth for boundary detection and (e) binary ground truth for segmentation.

To create ground truth for these extracted image patches, the manually annotated vertebrae boundary curves are used. For the boundary detection problem, the pixels under the curve have been assigned foreground class label, and all other pixels are assigned the background class label. This binary ground truth is used to train the boundary detection dense classification network. The ground truth is then smoothed with a Gaussian kernel and normalized as a valid probability distribution to create the ground truth for the probabilistic spatial networks. We have used a Gaussian kernel with a kernel size of 0.55 pixel. The choice of the standard deviation is based on visual evaluation of the resulted probability distribution. For the vertebral body segmentation problem, the pixels inside the boundary curves are considered as the foreground class and outside are considered as the background class [124]. Fig. 6.1 shows four examples of all the ground truth from the augmented training dataset. It can be noted that the binary ground truth in Fig. 6.1c and 6.1e contain artifacts due to pixelation effect. This effect is caused by the manually annotated curve which is defined to the 20 pixel-level points at the original resolution (see Sec. 2.3.1). At the patch resolution of 64×64, the pixels under the curve produce pixelation artifacts which may affect the training of the dense classification networks. The effect is reduced to some extent by the Gaussian smoothing in the probabilistic boundary ground truth (Fig. 6.1d).

6.4 Network and Training

The dense classification networks for the boundary detection and the segmentation problem have a similar architecture as the UNet architecture used in the center localization problem. The network diagram is shown in Fig. 6.2. The network terminates with a softmax layer which predicts the probability of each pixel belonging to the foreground class or background class. As this network is performing a pixel-wise binary classification, the standard cross entropy loss function is used. The same loss function has been used in Chapter 3 for spine localization network. However, for the boundary detection ground truth, the ratio between the number of pixels in the foreground class and the background class in the training ground truth is <0.05. The majority of the pixels belongs to the background class thus training a

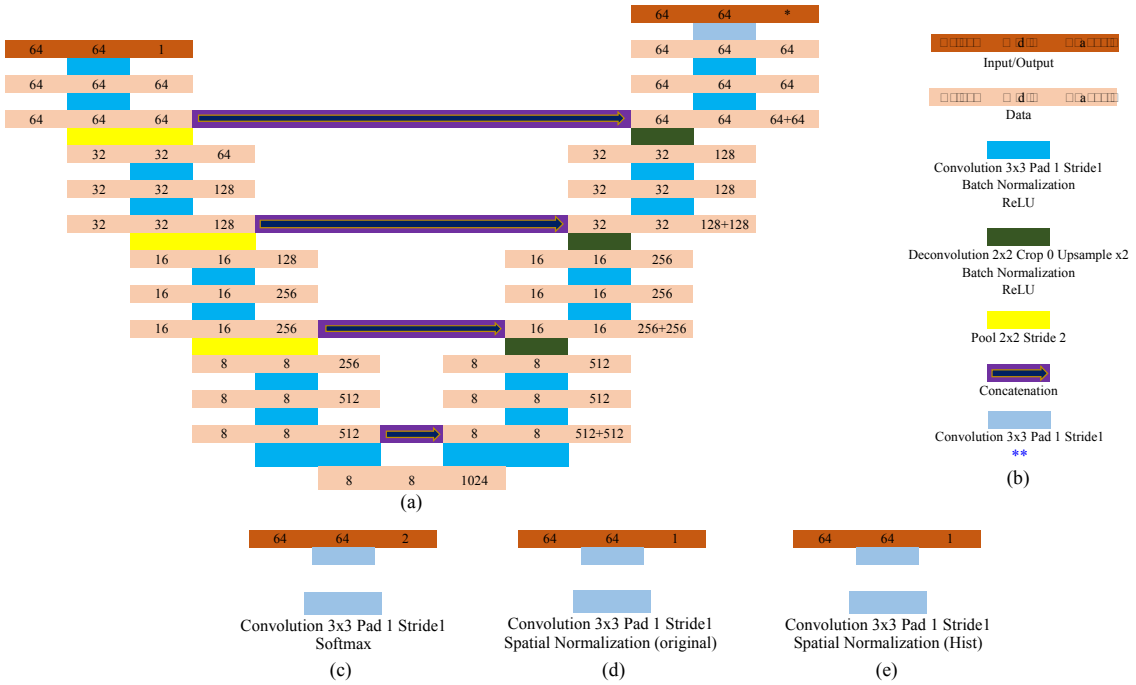


Fig. 6.2 Network architectures (a) common architecture and (b) legend (c) end modules for dense classification networks for boundary detection and segmentation (d) end modules for probabilistic networks: PSRN and (e) PSRN-H.

network with such imbalanced data may result in biased predictions. To handle this extreme data imbalance problem, we introduce a weight parameter into the loss function. This weight parameter balances the back-propagation of the loss term for the two classes. Similar balancing parameter has been used in [82]. The value of this parameter is dynamically computed based on the ratio of the foreground and background pixels.

Given a dataset of training image (x) - binary ground truth (y) pairs, training a dense classification network means finding a set of optimized parameters $\hat{\boldsymbol{W}}_o$ that minimize a loss function, L:

$$\hat{\boldsymbol{W}}_o = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} L(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W}), \tag{6.1}$$

where N is the number of training examples and $\{x^{(n)}, y^{(n)}\}$ represents n-th example in the training set with the corresponding ground truth. The loss function for the weighted dense

classification network is the pixel-wise log loss or the cross-entropy loss used before:

$$L(\{x,y\};\boldsymbol{W}) = -\sum_{i\in\Omega_p}\sum_{j=1}^{M} w_j y_i^j \log P(y_i^j = 1|x_i;\boldsymbol{W}), \tag{6.2}$$

$$P(y_i^j = 1|x_i;\boldsymbol{W}) = \frac{\exp(a_j(x_i))}{\sum_{k=1}^{M}\exp(a_k(x_i))}, \tag{6.3}$$

where $a_j(x_i)$ is the output of the penultimate activation layer of the network for the pixel x_i, Ω_p represents the pixel space, M is the number of class labels, and P are the corresponding class probabilities. The weight parameter w_j can be defined as:

$$w_j = \frac{\sum_{k=1}^{M}|\omega_k|}{|\omega_j|}, \tag{6.4}$$

where ω_k is the set of pixels having the k-th class label in a training image patch. Thus, the weight parameter dynamically changes its value from image to image based on the number of foreground and background pixels in each image.

For the segmentation ground truth, the ratio between the number of pixels in the foreground class and the background class varies between 0.3 to 0.7 based on the scaling factor of the data augmentation process. Since here this ratio is not as extreme as was it was in the case of boundary detection ground truth (< 0.05), the data balancing weight parameter was ignored (i.e., $w_j = 1$ is used). However, to encourage predicted segmentation mask to conform to possible vertebral shapes, a novel shape-aware term is added to the loss function of the vertebral body segmentation network. Along with the cross-entropy loss, this term further penalizes the predicted areas that do not match the ground truth, based on the known shape of the training vertebra. This term can be defined as:

$$L_s(\{x,y\};\boldsymbol{W}) = -E(x,y)\sum_{i\in\hat{\Omega}_p}\sum_{j=1}^{M} y_i^j \log P(y_i^j = 1|x_i;\boldsymbol{W}), \tag{6.5}$$

where

$$E(x,y) = mean\big\{min\{D(\boldsymbol{p},\boldsymbol{q}) : \boldsymbol{p} \in S_{gt}(y)\} : \boldsymbol{q} \in \hat{S}(x)\big\}, \tag{6.6}$$

and $\hat{S}(x)$ is the curve surrounding the predicted regions, $S_{GT}(y)$ is ground truth curve and $D(\boldsymbol{p},\boldsymbol{q})$ computes the Euclidean distance between the point $\boldsymbol{p}$ and $\boldsymbol{q}$. $\hat{S}(x)$ is generated by locating the boundary pixels of the predicted mask which is a function of the input image x. Similarly, $S_{gt}(y)$ is generated by locating the boundary pixels of the ground truth mask or the segmentation label, y. The redefined pixel space, $\hat{\Omega}_p$, contains the set of pixels where the prediction mask doesn't match the ground truth mask. These terms can also be explained using the toy example shown in Fig. 6.3. Given a ground truth mask (Fig. 6.3a) and a prediction mask (Fig. 6.3b), E is computed by measuring the average distance between the ground truth (green) curve and prediction (red) curve (Fig. 6.3c). Fig. 6.3d shows the redefined pixel space, $\hat{\Omega}_p$. The shape-aware term introduces an additional penalty proportional to the Euclidean distance between predicted and ground truth curve to the pixels that do not match the ground truth segmentation mask. In the case when the predicted mask is a cluster of small regions, especially during the first few epochs in training, E becomes very large because of the increase in the boundary perimeters from the disjoint predictions. Thus, this term also implicitly forces the network to learn to predict a single connected prediction mask faster. Instead of the average Euclidean distance, another potential choice for the distance function could be Hausdorff distance [125]. This distance function computes the maximum distance between two shapes. For the example shown in Fig. 6.3c, Hausdorff distance would ignore the fact that the majority parts of the two shapes coincide with each other and thus would

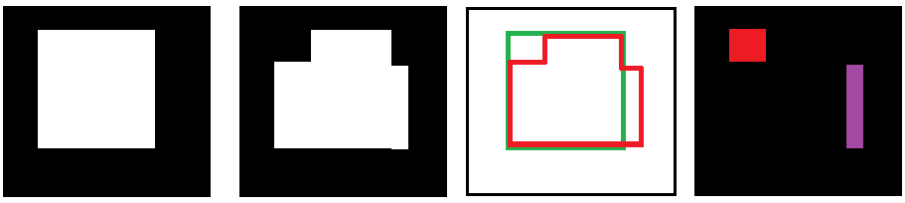


Fig. 6.3 Shape-aware loss (a) ground truth mask (b) prediction mask (c) ground truth shape, C_{GT} (green) and prediction shape, $\hat{C}$ (red) (d) refined pixel space, $\hat{\Omega}_p$: false positive (purple) and false negative (red).

result in a higher error. In case of multiple disjoint predicted regions, there would have been different Hausdorff distances for each of the separate regions making the computation of the loss in Eqn. 6.5 complicated.

Finally, incorporating the shape-aware term, the loss function of Eqn. 6.1 can be extended as:

$$\hat{\boldsymbol{W}}_o = \arg\min_{\boldsymbol{W}} \sum_{n=1}^{N} \Big(L(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W}) + L_s(\{x^{(n)}, y^{(n)}\}; \boldsymbol{W}) \Big). \tag{6.7}$$

The vertebral body segmentation network is trained using the above loss function.

The probabilistic spatial regressor network (PSRN) used in this chapter has the same architecture of the PSRN used for corner localization in Chapter 5. We also use the same Bhattacharyya coefficient (BC)-based loss function (see Eqn. 5.18 and 5.19). However, the spatial normalization layer has been improved with a histogram-based normalization layer that addresses the residual background probability issue mentioned in Chapter 5. The spatial normalization layer converts the last feature map from the network to a valid probability distribution. The previous spatial normalization layer forces the minimum to be zero and

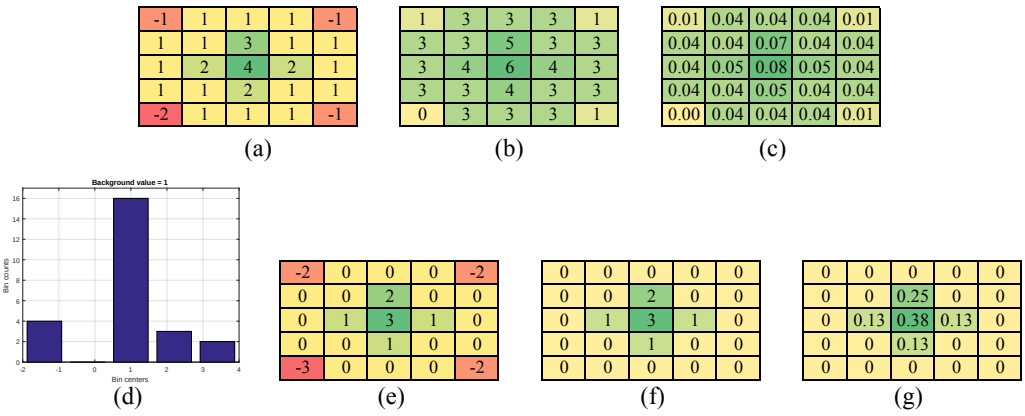


Fig. 6.4 Histogram-based spatial normalization layer. (a)-(c) illustrate the residual probability problem of the previous chapter. (d)-(g) summarizes the histogram-based solution to this problem. (a) input feature map (b) feature map after min subtraction (c) resulted probability distribution from the original spatial normalization layer (d) histogram of the input feature map (e) background value subtracted feature map (f) negative value replaced by zeros (g) resulting probability distribution from the histogram-based spatial normalization layer.

summation to be one. This process is illustrated in Fig. 6.4a, 6.4b and 6.4c. Instead of forcing the minimum value of the last feature map to zero, the new layer finds the residual background probability by analyzing the histogram of the feature maps and forces any pixel having values equal or less than the background probability to zero. This way the residual probability problem can be solved.

Assume X is the input to the normalization layer of the network. Histogram analysis of this input provides us with n bin centers and corresponding n bin counts representing the number of pixels with values around each bin center. We locate the maximum of the bin counts and consider that bin center as the background value for that particular feature map. We then subtract the background value from all predicted probabilities and replace any resulting negative values with zeros. Finally, we force the summation of the resultant feature to be unity. The process is summarized in Fig. 6.4a, 6.4d, 6.4e and 6.4f. In this hypothetical example, the input, X, is a 5×5 matrix and $n = 5$. It can be noticed how the background residual probability problem disappears when histogram-based spatial normalization has been applied, resulting in a sharper probability distribution. In our training, X is a 64×64 matrix, and we have used $n = 25$ histogram bins.

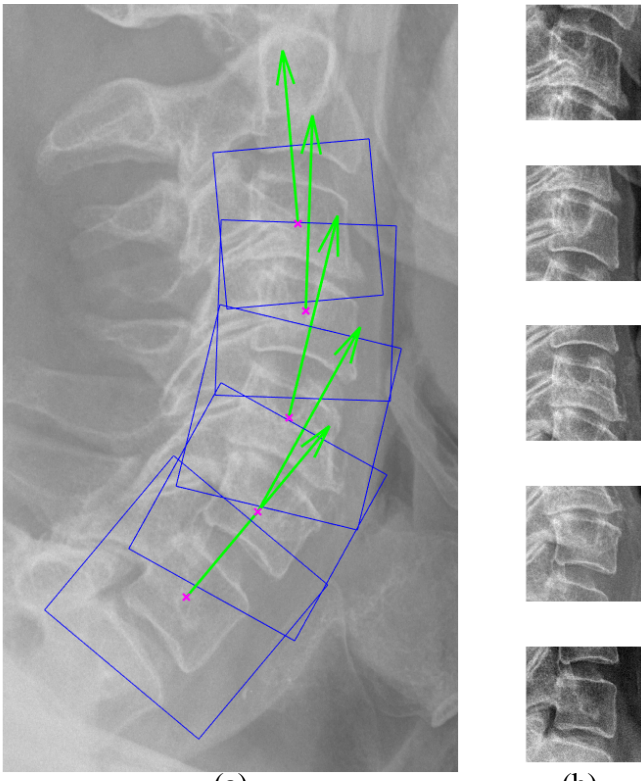
6.5 Experiments

We have trained four networks for the boundary detection problem. First, two dense classification networks: one without the introduced novel class balancing parameter (BDNet) and the other with the weight parameter of Eqn. 6.2 (BDNet-W). The class imbalanced network, BDNet, is trained with the same loss function with equal (unity) weight parameters for both foreground and background class. We have also trained two networks for the probabilistic boundary detection: one with the original spatial normalization layer (PSRN) and other with the histogram-based normalization layer (PSRN-H). For the vertebral body segmentation problem, we have trained two networks: SegNet and SegNet-S. '-S' signifies the use of updated shape-aware loss function of Eqn. 6.7. All networks share a common structure

except the last layer and the loss function. The differences in the networks are illustrated Fig. 6.2. The networks are trained on a system with a NVIDIA Pascal Titan X GPU for 30 epochs with a batch-size of 25 image patches. The training took approximately 28 hours for each network.

6.5.1 Test Patch Extraction

As previously stated, for this chapter, we assume the vertebral center points are provided manually during testing. Based on these points, the orientation vector, $\boldsymbol{F}$, can be computed as described in Sec. 2.4.2. The direction of this vector dictates the orientation, and the magnitude of this vector dictates the size of the extracted test patch. Our test dataset of 172 images contains 797 vertebrae. The extracted vertebral image patches are then resized to



(a) (b)

Fig. 6.5 Test patch extraction process (a) manually annotated centers (×), orientation vectors (↑) and patch boundaries in blue (b) extracted test patches.

the resolution of 64×64 and propagated through the networks to generate corresponding predictions. The patch extraction process is illustrated in Fig. 6.5.

6.5.2 Compared Algorithms

To compare with the deep neural network-based prediction results, three active shape model (ASM)-based shape prediction frameworks have been implemented. A simple maximum gradient-based image search-based ASM (ASM-G) [119], a Mahalanobis distance-based ASM (ASM-M) [24] and a random forest-based ASM (ASM-RF) [69]. The latter two have been used in cervical vertebrae segmentation in different datasets. These ASM-based models are different from the ASM-based framework discussed in Chapter 2. The ASM-based framework described in Sec. 2.4 works on the full resolution image, whereas in this chapter, the models are applied to the extracted 64×64 pixel test image patches. Another important difference is that a single model is trained for all the vertebrae, C3-C7. Previously, separate vertebral models were trained. The ASM models were trained on the same 26,370 vertebrae used for training the deep networks which include the scale and rotation variations introduced during data augmentation. Training the ASM models this way improved the shape prediction

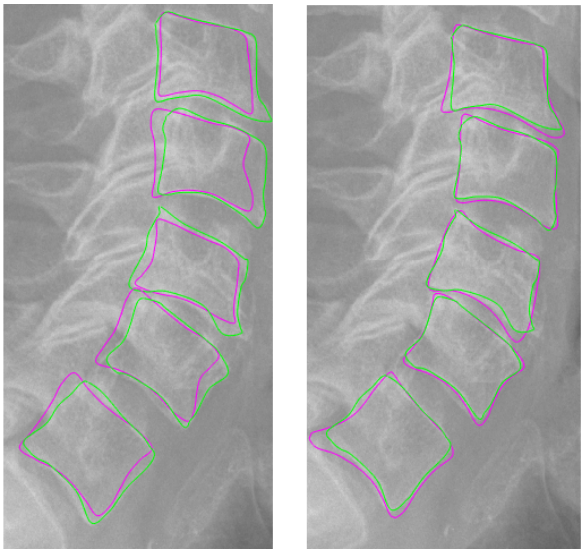


Fig. 6.6 Performance of the ASM-based initial framework (left) and performance of the ASM-G method trained in this chapter (right). Converged vertebral shapes (magenta) with ground truth shapes (green).

performance from the initial framework. One qualitative comparison between the initial framework and ASM-G trained for this chapter on the same test images is shown in Fig. 6.6.

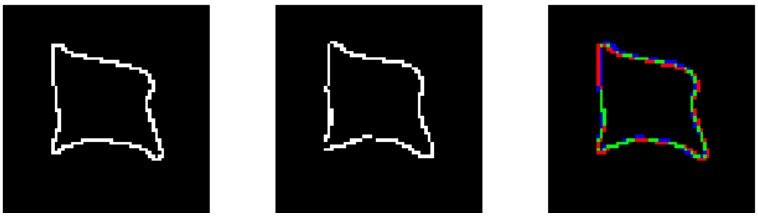
6.5.3 Inference and Metrics

During inference, 797 vertebrae from 172 test images are extracted following the process discussed in Sec. 6.5.1 based on the manually clicked vertebral centers. These patches are propagated through each of the networks to get the predictions.

The dense classification networks for boundary detection predict binary edge maps. The predicted edge map can be compared with the corresponding binary ground truth to categorize the pixels as true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The performance of the binary boundary detection results can then be compared based on the Dice similarity coefficient (DSC):

$$DSC = \frac{2|TP|}{2|TP| + |FP| + |FN|}. \tag{6.8}$$

DSC is also known as F-score or F-measure in the edge detection literature. The DSC between the ground truth edge map and predicted edge map could be computed with a matching distance (d). The matching distance is the maximum permissible distance when matching a predicted edge pixel with the ground truth. The use of optimal matching distance for comparing edge detection algorithms is standard in the literature [114, 115, 126, 127]. Here, we consider two cases: $d = 0$ and $d = 1$ pixel. The motivation behind computing the metrics with a matching distance, $d = 1$, is illustrated in Fig. 6.7. Although, the ground truth and the prediction are almost similar in Fig. 6.7a and 6.7b, the DSC is only 0.53 with $d = 0$. This happens because the prediction edge pixel is often displaced by a single pixel from the ground truth. While this is an erroneous prediction, but only computing DSC with $d = 0$ cannot address the fact the predicted edge pixel was adjacent to the ground truth. However, comparing the ground truth and prediction with a matching distance, $d = 1$, can consider adjacent pixel prediction as a correct prediction.



(a) (b) (c)

Fig. 6.7 Dice similarity coefficient (DSC) with different matching distances for boundary detection (a) binary ground truth (b) binary prediction (c) overlap between the ground truth and the prediction. Green indicates true positive, blue false positives and red false negatives. With matching distance, $d = 0$, the $DSC = 0.53$ and with $d = 1$, the $DSC = 0.94$.

The performance of the probabilistic spatial regressor networks are measured based on the Bhattacharyya coefficient between the predicted probability map and the ground truth probability maps.

For the segmentation networks, SegNet and SegNet-W, the performance is measured by the DSC without the matching distance. Along with the DSC, we also report the pixel-wise accuracy (pA) which is defined in Eqn. 6.9.

$$pA = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \times 100\%. \tag{6.9}$$

The DSC and pA metrics are well suited to capture the number of correctly segmented pixels, but they fail to capture the differences in the shape. In order to compare the shape of the predicted mask appropriately with the ground truth vertebral boundary, the predicted masks of the deep segmentation networks are converted into shapes by locating the boundary pixels. These shapes are then compared with the manually annotated vertebral boundary curves by measuring average point to curve Euclidean distance (E_{p2c}) between them. The error is defined in Eqn. 6.10. The networks are trained and tested on vertebral image patches of size 64×64 pixels. The image-level pixel spacing (millimeter per pixel) information is not representative in this normalized space. Thus, the error, E_{p2c}, is reported in pixels. We also report the fit failure. This metric has been used previously in Chapter 5 for comparing different vertebral corner localization methods. The definition can be found in Sec. 5.4. Here,

in this chapter, we redefine the fit failure as the percentage of vertebrae having an E_{p2c} of greater than two pixels.

$$E_{p2c}(\hat{S}, S_{gt}) = mean\big\{min\{D(\boldsymbol{m},\boldsymbol{n}) : \boldsymbol{n} \in S_{gt}\} : \boldsymbol{m} \in \hat{S}\big\}, \tag{6.10}$$

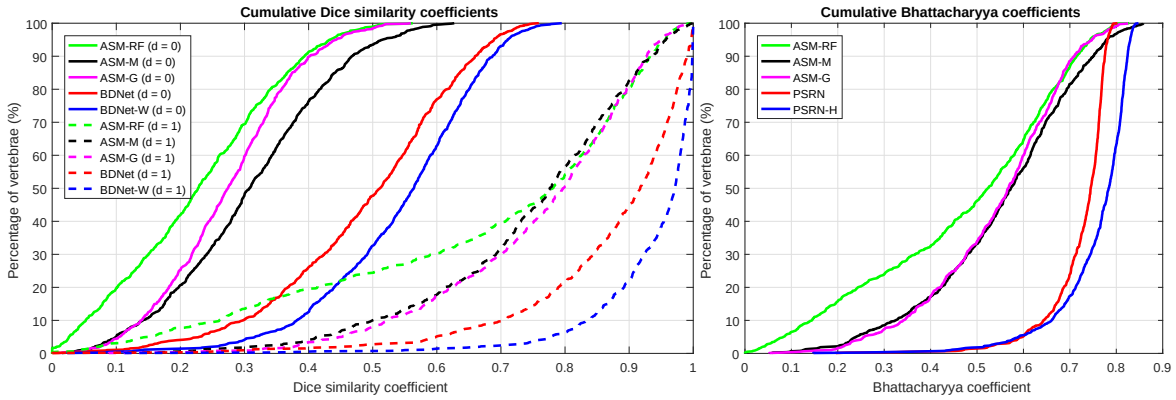
where $\hat{S}$ is set points in the predicted shape, S_{gt} is set of points in the manually annotated vertebral boundary curve and $D(\boldsymbol{m},\boldsymbol{n})$ is the Euclidean distance between the point $\boldsymbol{m}$ and $\boldsymbol{n}$.

Finally, it should be noted that the ASM-based methods predict shapes. To compare these algorithms with the proposed methods, these predicted shapes are converted to corresponding binary edge maps, segmentation masks or probability distributions following same procedure mentioned in Sec. 6.3.

6.6 Results

6.6.1 Boundary Detection

We present the DSC for the boundary detection dense classification networks in Table 6.1. The shapes predicted by the ASM-based methods are converted to corresponding binary maps for computing the DSC. It can be seen that the performance of the ASM-based methods is much lower than the dense classification networks. Among the two version of the dense classification networks, the performance has been significantly improved when the novel weighted loss function is used (BDNet-W). The DSC with $d = 0$ is low, but we have achieved a maximum average DSC of 0.936 with $d = 1$. The probabilistic spatial regressor networks are compared in Table 6.2 based on the Bhattacharyya coefficients (BC). For comparison, the predictions of the ASM-based methods are also converted to probabilistic outputs. These methods have achieved a maximum average BC of 0.544 where the PSRN-based methods achieved BC higher than 0.72. The introduction of the novel histogram-based normalization layer (PSRN-H) has resulted in significant improvement, pushing the average BC to 0.757. Following a Jarque-Bera test at the 5% significance level [128], it was found that the resulted



(a) (b)

Fig. 6.8 Cumulative metric curves (a) Dice similarity coefficients (b) Bhattacharyya coefficients.

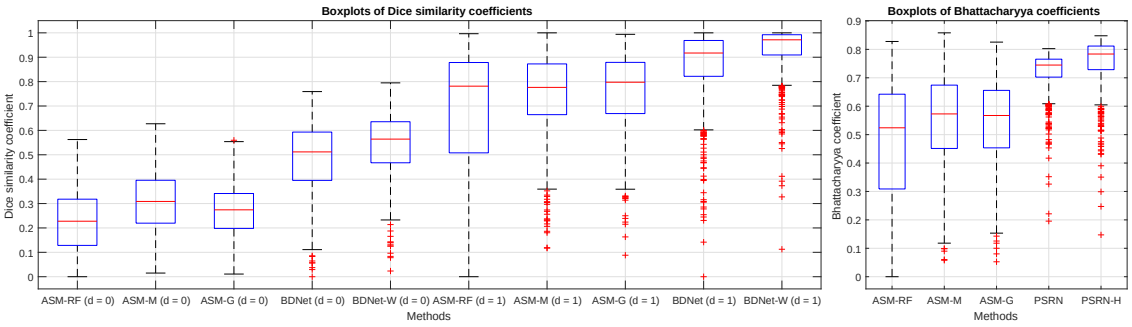
metrics are not normally distributed. Thus the significance tests reported in Table 6.1 and 6.2 are performed using the Wilcoxon signed-rank test [129] instead of the student's t-test.

Dice similarity coefficient					
Matching distance	$d = 0$		$d = 1$		Wilcoxon signed-rank test p-value
Method	Mean	Std	Mean	Std	
ASM-RF	0.228	0.125	0.678	0.261	
ASM-M	0.309	0.124	0.746	0.167	
ASM-G	0.273	0.102	0.758	0.158	
BDNet	0.489	0.143	0.872	0.138	$< 10^{-65}$
BDNet-W	**0.543**	0.124	**0.936**	0.090	

Table 6.1 Dice similarity coefficients for binary boundary detection networks.

Bhattacharyya coefficient			
Method	Mean	Std	p-value (Wilcoxon signed-rank test)
ASM-RF	0.473	0.212	
ASM-M	0.553	0.161	
ASM-G	0.544	0.145	
PSRN	0.723	0.067	$< 10^{-93}$
PSRN-H	**0.757**	0.083	

Table 6.2 Bhattacharyya coefficients for probabilistic boundary detection networks.



(a) (b)

Fig. 6.9 Boxplots of quantitative metrics (a) Dice similarity coefficients (b) Bhattacharyya coefficients.

The difference in the boundary detection performance between the ASM-based methods and the deep network-based methods are also visible in the cumulative metric curves shown in Fig. 6.8. In between the dense classification networks, the difference between the cumulative curves is more near the lower DSCs, indicating the fact that for difficult images the weighted term is more effective. As we move towards higher DSCs, the curves are nearer. For the probabilistic spatial regressor networks, we see the opposite pattern is observed. The networks perform similarly for lower BCs and difference is more for higher BCs. The boxplots of these metrics are shown in Fig. 6.9.

Qualitative predictions from the networks are shown in Fig. 6.10 and 6.11. The outputs of the dense classification networks are crisp [115, 130] and more accurate than the ASM-based methods but often discontinuous and broken. Even for relatively less challenging scenarios like Fig. 6.10a and 6.10b, the dense classification results are discontinuous. For challenging examples, the occurrence of discontinuity increases. The probabilistic spatial regressor networks produce smoother vertebral boundaries. But these predictions are thicker than the probabilistic ground truth. This issue can be attributed to the contracting path of network architecture, where much of the spatial information is lost. The dense classification network recovers the spatial information better than the probabilistic network in the expanding path through concatenation of data matrices. Overall, the probabilistic outputs give a better qualitative sense of the vertebral boundaries and less possibility of discontinuity. However,

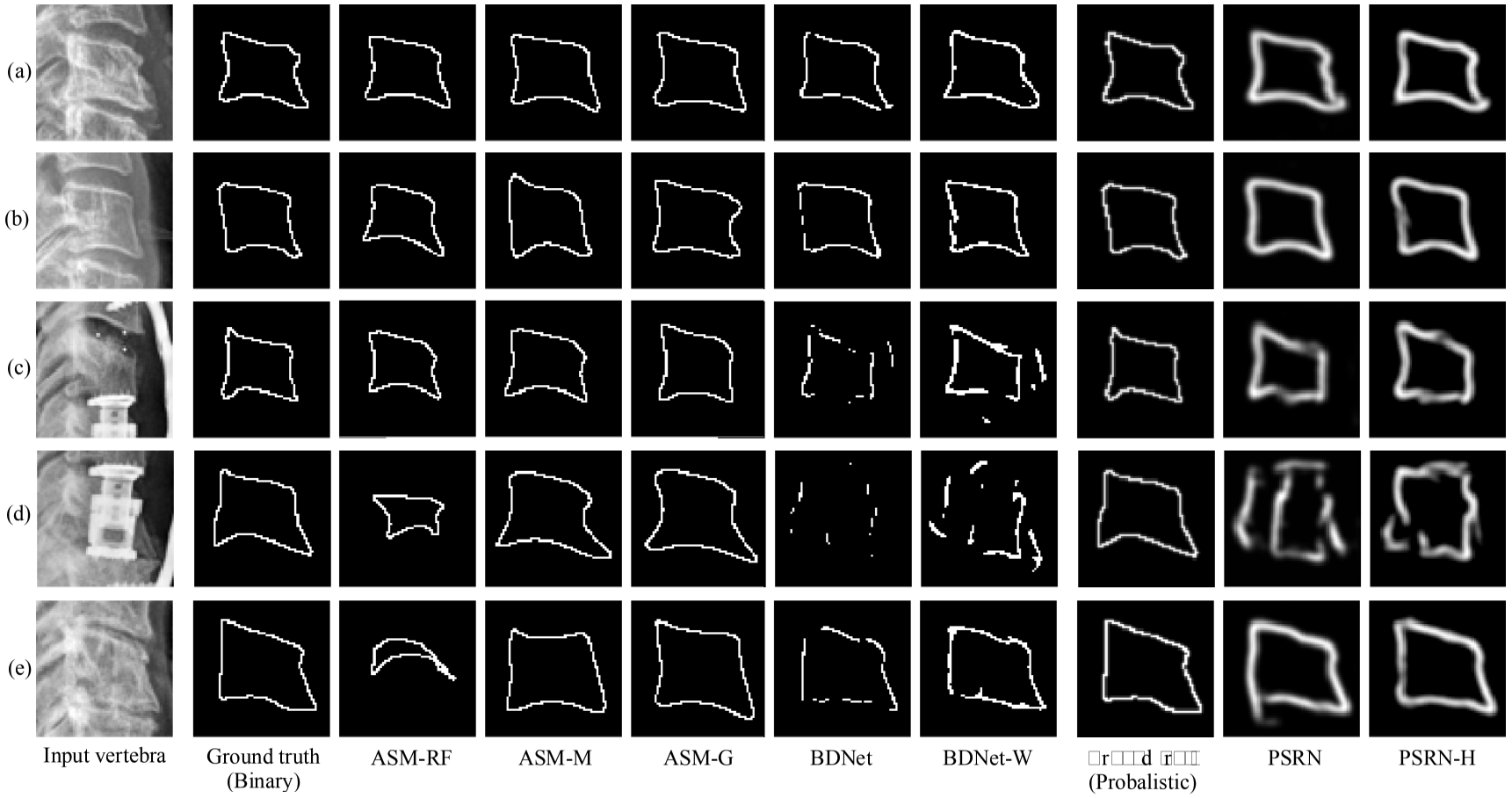


Fig. 6.10 Patch-level edge detection results 1.

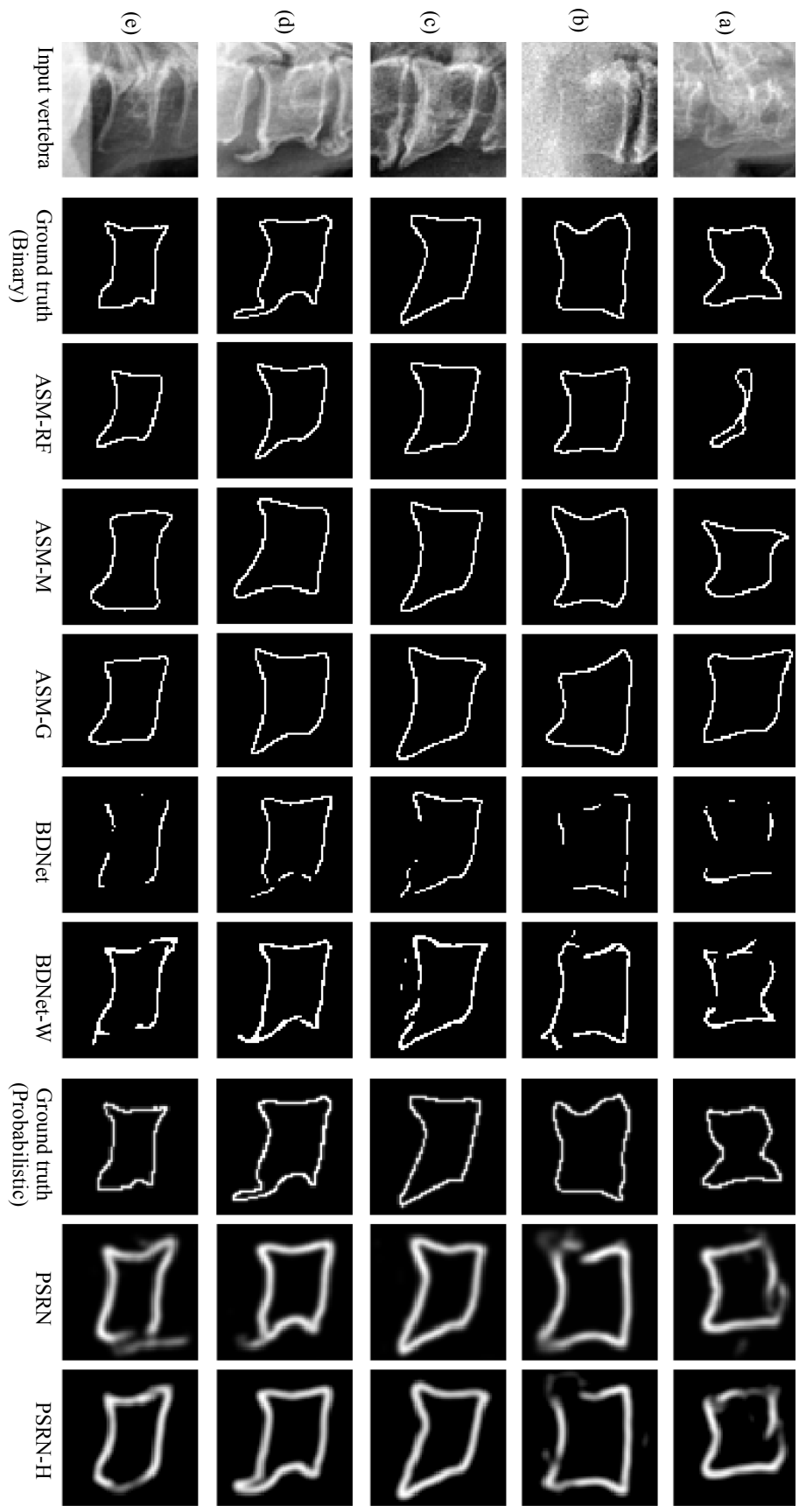


Fig. 6.11 Patch-level edge detection results 2.

the qualitative evaluation depends on human perception and may vary from user to user.

For visualization, a post-processing step can be introduced to reduce the thickness of the probabilistic boundary predictions. We have applied morphological erosion on the predictions with a four-neighborhood structuring element. Two examples of this thinning operation are shown in Fig. 6.12. Notice that after the thinning operation, the prediction thickness becomes similar to the ground truth thickness.

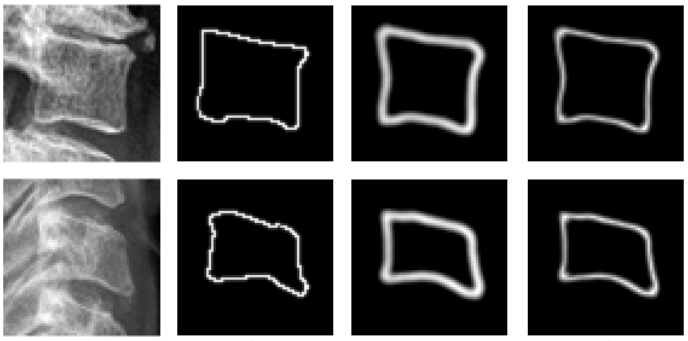


Fig. 6.12 Post-processing for reducing thickness of the predicted distribution (a) input test vertebrae (b) probabilistic ground truth (c) thick prediction of the probabilistic networks (d) eroded predictions (PSRN-H$_e$).

The patch-level predictions can be projected back on the original test images using affine transformation like previous chapters (Sec. 4.3.3 and 5.3.4). A few examples of image-level results from BDNet-W and PSRN-H networks are shown in Fig. 6.13 and 6.14. For the PSRN-H network, we have used the post-processed, thinned predictions (PSRN-H$_e$ in Fig. 6.12). We also show what the ground truth looks like when projected back on the original image. Fig. 6.13 shows examples of spinal columns from healthier subjects, and it can be seen the dense classification network produces discontinuous and noisy predictions for a few cases. But the predictions from the PSRN-H network is smooth and continuous in these examples. Examples of images with severe clinical conditions and bone implants are shown in Fig. 6.14. Both dense classification and probabilistic approaches suffer in the presence of these conditions. A problem related with the test patch extraction procedure can also be noticed in Fig. 6.14d and 6.14e. The ground truth is not continuous for some of the vertebrae.

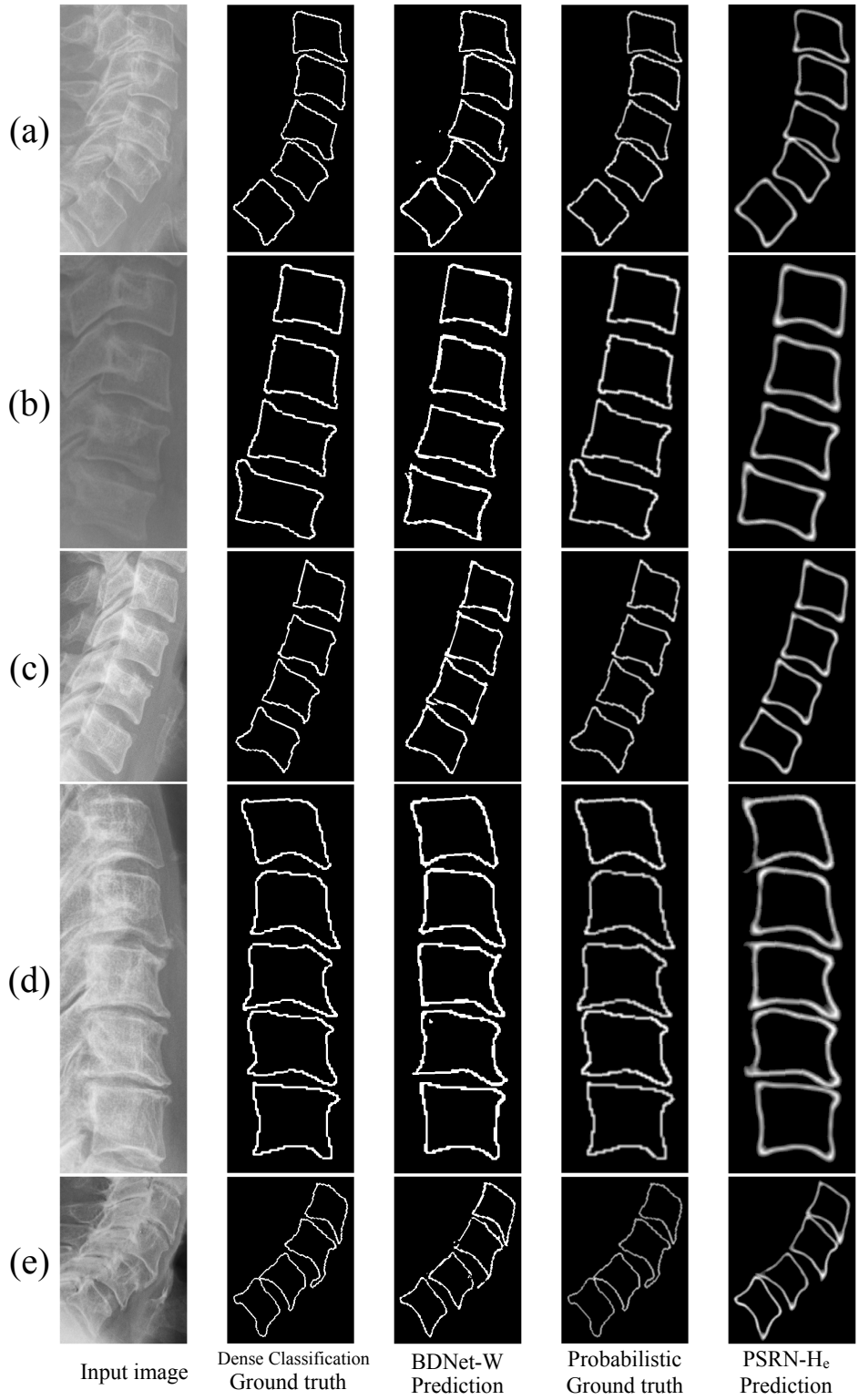


Fig. 6.13 Image-level edge detection results 1. PSRN-H_e indicates the eroded (thinned) patch-level predictions are used.

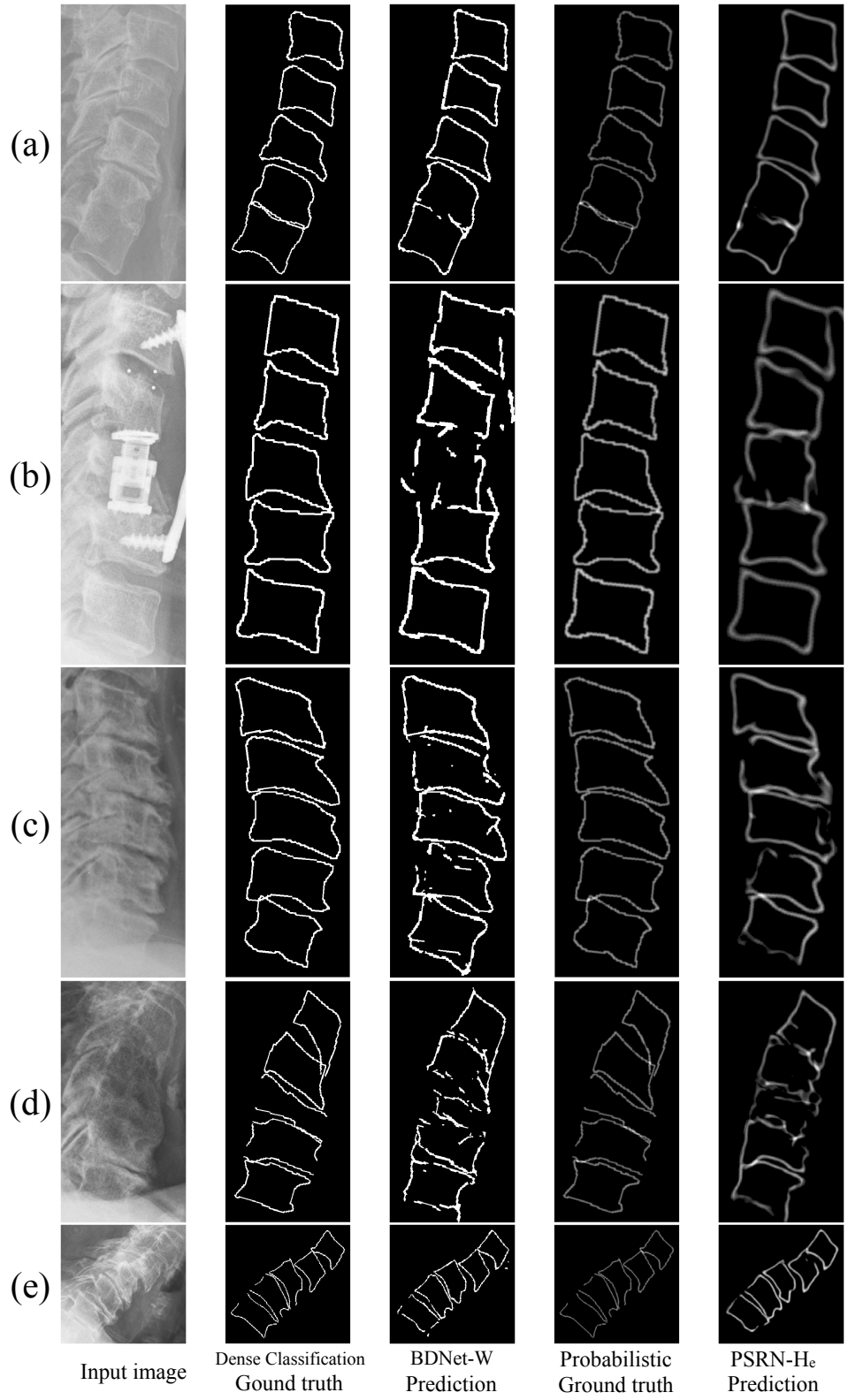


Fig. 6.14 Image-level edge detection results 2.

This is because of the severe degenerative changes which made the vertebrae to be prolonged exceptionally in the horizontal direction. Since the patch size at test time is proportional to the distances between the vertebral centers, the prolonged vertebrae get cut off at their horizontal extremes. As the networks are not trained with such examples, the predictions suffer greatly in this situation.

6.6.2 Segmentation

The median, mean and standard deviation (std) of the DSC and pA metrics over the test dataset of 797 vertebrae for segmentation methods are reported in Table 6.3. Deep segmentation network-based methods outperform the ASM-based methods. The SegNet achieves a 2.9% improvement in terms of pixel-wise accuracy and an increase of 0.055 for the Dice similarity coefficient. Among the two versions of the deep network, the use of novel loss function improves the performance by 0.31% in terms of pixel-wise accuracy. In terms of the Dice similarity coefficient, the improvement is in the range of 0.006. Although subtle, the improvements are statistically significant according to a paired t-test at a 5% significance level. Corresponding $p-values$ between the two versions of the network are reported in Table 6.3. Bold fonts indicate the best performing metrics. Interestingly, among the ASM-based methods, the simplest version, ASM-G, performs better than the alternatives. Recent methods [24, 69] have failed to perform robustly on our challenging dataset of test vertebrae.

	Pixel-wise accuracy (%)				Dice similarity coefficient			
	Median	Mean	Std	p-value	Median	Mean	Std	p-value
ASM-RF	95.09	90.77	8.98		0.881	0.774	0.220	
ASM-M	95.09	93.48	4.92		0.900	0.877	0.073	
ASM-G	95.34	93.75	4.48		0.906	0.883	0.066	
SegNet	97.71	96.69	3.04	$< 10^{-12}$	0.952	0.938	0.048	$< 10^{-12}$
SegNet-S	**97.92**	**97.01**	**2.79**		**0.957**	**0.944**	**0.044**	

Table 6.3 Average quantitative metrics for segmentation.

The average point to curve error (E_{p2c}) for the methods are reported in Table 6.4. The deep segmentation framework, SegNet, produced a 35% improvement over the ASM-based

methods in terms of the mean values. The introduction of the novel loss term in training further reduced the average error by 12% achieving the best error of 0.99 pixels. The most significant improvement can be seen in the fit failure which denotes the percentage of the test vertebrae having an average error of higher than 2 pixels. The novel shape-aware network, SegNet-S, has achieved a drop of around 37% from the ASM-RF method. The cumulative distribution of the point to curve error is also plotted in the performance curve of Fig. 6.15. It can be seen that the adaptation deep segmentation network provides a big improvement in the area under the curve.

	Average point to curve (E_{p2c}) error in pixels				Fit failure(%)
	Median	Mean	Std	p-value	
ASM-RF	1.82	2.59	1.85		43.43
ASM-M	1.54	1.88	1.05		32.70
ASM-G	1.38	1.73	0.99		26.89
SegNet	0.77	1.11	1.29	0.0043	8.59
SegNet-S	**0.78**	**0.999**	**0.67**		**6.06**

Table 6.4 Average quantitative metric for shape prediction.

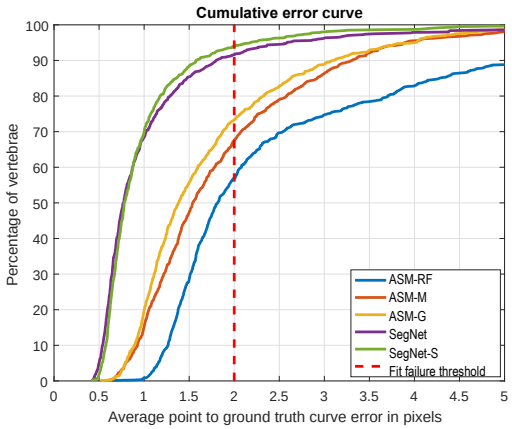
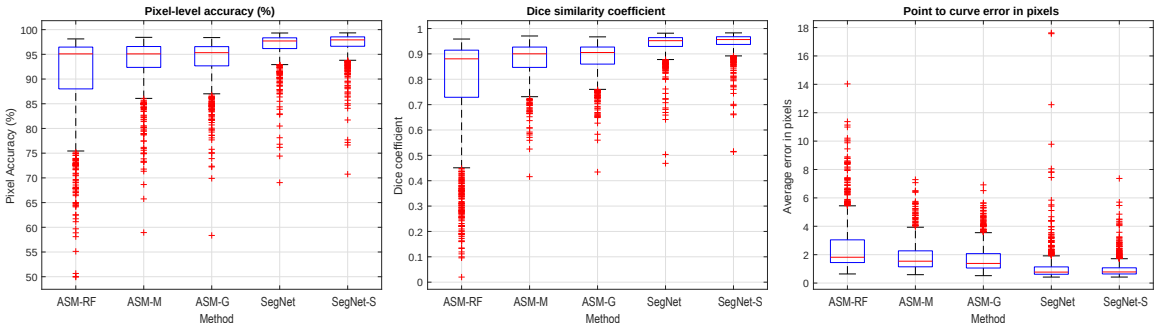


Fig. 6.15 Cumulative distribution of point to curve (E_{p2c}) errors.

The boxplots of the quantitative metrics are shown in Fig. 6.16. It can be seen that even the worst outlier for the shape-aware network, SegNet-S, has a pixel-wise accuracy higher than 70%, signifying the regularizing capability of the novel term. Most of the outliers are



(a) (b) (c)

Fig. 6.16 Boxplots of quantitative metrics (a) pixel-level accuracy (b) Dice similarity coefficients (c) point to ground truth curve error, E_{p2c}.

caused by bone implants, fractured vertebrae or abnormal artifacts in the images. A few examples for qualitative assessment are shown in Fig. 6.17. A relatively less challenging case is shown in Fig. 6.17a, where all the methods perform well. Examples with bone implants are shown in Fig. 6.17b and 6.17c. Fig. 6.17d and 6.17e show vertebrae with abrupt contrast change. Vertebrae with fracture and osteoporosis are shown in Fig. 6.17f and 6.17g. Fig. 6.17g also shows how SegNet-S has been able to capture the vertebral fracture patterns. Fig. 6.17h and 6.17i show vertebrae with image artefacts. A complete failure case is shown in Fig. 6.17j. In all cases, the shape-aware network, SegNet-S, has produced better segmentation results than its counterpart.

6.6.2.1 Analysis on Challenging Cases

Although statistically significant, the difference in performance between the SegNet and SegNet-S is subtle over the whole dataset of test vertebrae. This is because the majority of the vertebrae are healthy and shape-awareness does not improve the results by a large margin. To show the shape-awareness capability of SegNet-S, a selection of 52 vertebrae with severe clinical conditions is chosen. The average metrics for this subset of test vertebrae between SegNet and SegNet-S is reported in Table 6.5. An improvement of 1.2% and 0.02 have been achieved in terms of pixel-wise accuracy and Dice similarity coefficient, respectively. The differences over the whole dataset were only 0.31% and 0.006. The metric, E_{p2c} produces the most dramatic change. The novel shape-aware network, SegNet-S, reduced the error by

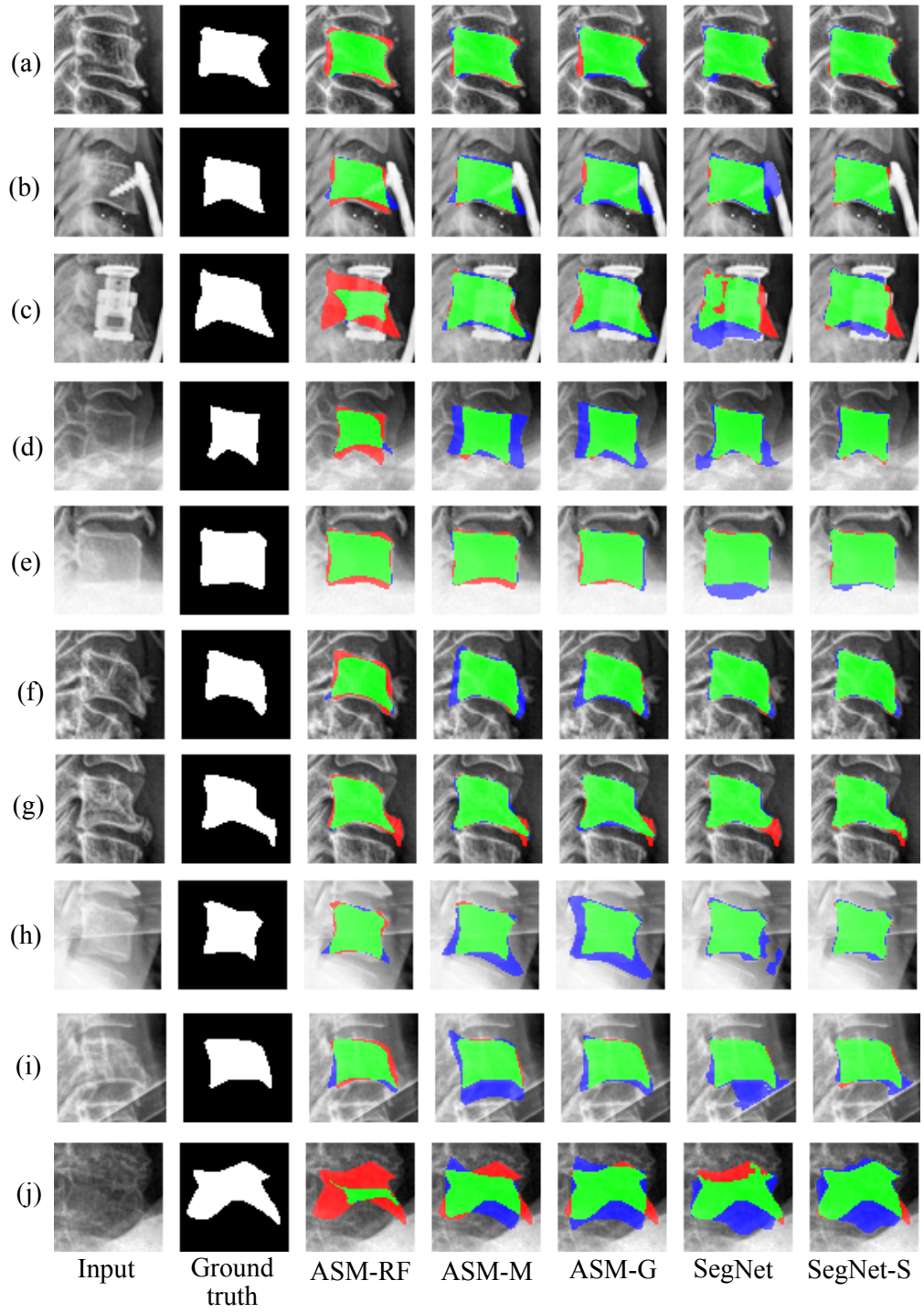


Fig. 6.17 Qualitative segmentation results: true positive (green), false positive (blue) and false negative (red).

22.91% for this subset of vertebrae with severe clinical conditions. Fig. 6.18 shows a few example of this subset of images.

	Average quantitative metrics		
	Pixel-wise accuracy (%)	Dice coefficient	Point to curve error (E_{p2c})
SegNet	94.01	0.91	1.61
SegNet-S	**95.21**	**0.93**	**1.24**

Table 6.5 Comparison between SegNet and SegNet-S for cases with severe clinical condition.

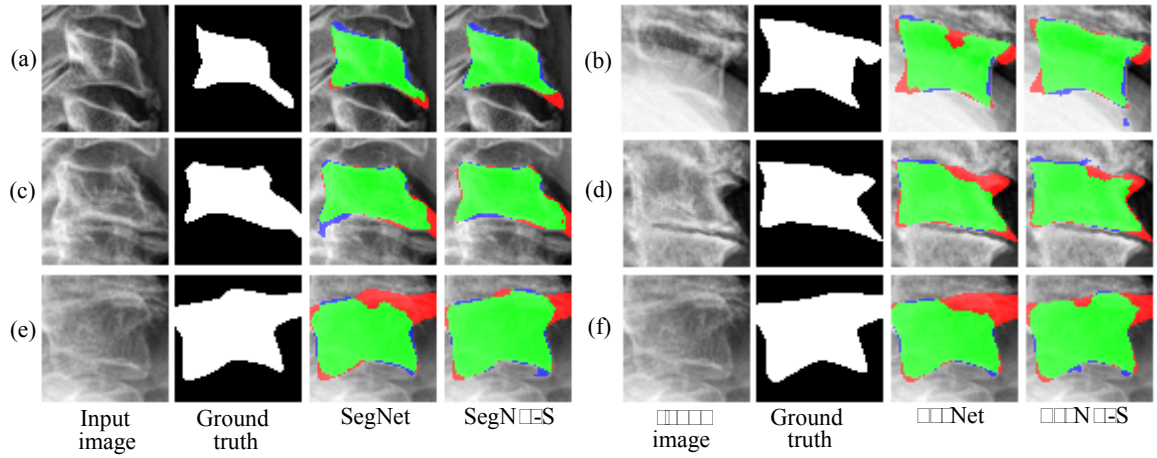


Fig. 6.18 Comparison of segmentation performance for vertebrae with severe clinical condition.

6.6.3 Qualitative Results on NHANES-II Dataset

We have applied our trained networks on the vertebra image patches collected from the NHANES-II dataset. A few qualitative results for these image patches are in shown in Fig. 6.19. The probabilistic boundaries are predicted by the PSRN-H network, and the segmentation masks are generated by the SegNet-S network. It can be seen that most of the predictions are accurate even though the networks were trained on a completely different dataset. This proves the robustness of the trained networks.

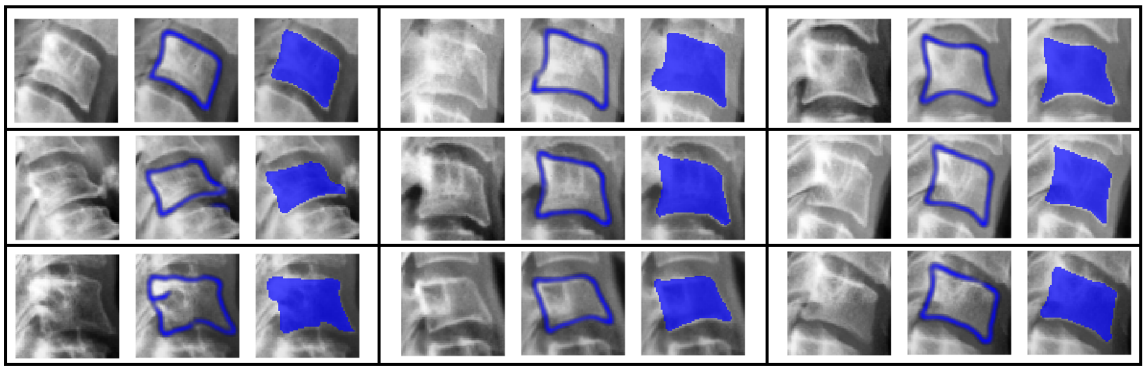


Fig. 6.19 Qualitative boundary detection and segmentation results for vertebrae collected from the NHANES-II: input image patch – predicted vertebral boundary – segmented vertebral body. The predictions are displayed on the input image patch as the blue overlay. Ground truth information is not available.

6.7 Conclusion

In this chapter, we have proposed novel methods for detecting vertebral boundaries and segmenting vertebral bodies. The boundary prediction problem is approached in two ways: a hard boundary detection using dense classification networks and a soft boundary detection approach using spatial probabilistic regressor networks. We proposed a weight parameter to compensate the class imbalance problem in the dense classification network. The weight dynamically changes based on the number of the pixels belonging to each of the classes. The dynamic weight parameter achieved significant improvement over unbalanced training scheme for dense classification networks. We have also proposed a novel histogram-based normalization layer for the spatial probabilistic regressor network (PSRN) to solve the residual background probability problem encountered in the previous chapter. The PSRN with the new normalization layer was able to significantly improve the boundary detection performance compared to the PSRN with the original normalization layer. For segmentation problem, we have proposed a robust semi-automatic framework using a dense classification network. The proposed deep segmentation method has outperformed the traditional active shape model (ASM)-based approaches by a significant margin. To incorporate the shape information with the mask prediction capability of the deep neural network, a novel shape-aware loss function has been formulated. The inclusion of this novel term in training

provided significant quantitative and qualitative improvements.

The dynamic weight parameter proposed for the dense classification network can easily be adapted to other classification neural networks. The weight parameter has been defined with a general formulation for multiple classes. Thus it can be utilized in any classification problem where the number of samples per class varies significantly [131]. This parameter could be particularly useful for several medical applications, where a classification between healthy and unhealthy subjects is required. The number of samples in the healthy category often surpass the number of samples in the unhealthy category in medical datasets [132, 133]. In future work, we can utilize this dynamic weight parameter for classification of the vertebrae with different types and grades of clinical conditions.

In this chapter, we have also proposed an innovative boundary detection approach using the spatial probabilistic regression network. The state-of-the-art work in the field of boundary detection is primarily dominated by dense classification networks [114, 115] which classifies each pixel in the image as being a boundary pixel or a non-boundary pixel. In contrast, we brought a probabilistic regression approach to the boundary detection problem where a spatial probability map is generated with high probabilities at the boundary locations. Potentially, this could change the way boundary detection problem is looked at and outperform the traditional classification approach where spatial continuity of the boundary is of utmost importance.

Like the vertebral corners, the predicted vertebral boundaries and bodies can be used to detect conditions like spondylolisthesis and retrolisthesis in the spinal column. The anterior, medial and posterior height of the vertebrae can easily be computed from the predictions, which can then be used to identify vertebrae with compression fractures. The type and the grade of the compression fracture can also be computed from these heights using the Genant scale [28]. The predicted vertebral boundaries can be further analyzed to identify conditions like osteophytes, osteoporosis and other degenerative changes. Moreover, the segmented

vertebral bodies can also be analyzed to detect complex conditions like low bone density (bone loss).

The novel shape-aware loss term proposed for the segmentation networks improved the overall accuracy significantly. The improvement was more noticeable for the vertebrae with clinical conditions. This can be crucial for correct identification of the clinical conditions and their severity. The Genant scale categorizes different grades of the compression fracture by the ratio of the anterior, medial and posterior heights of the fractured vertebra. The improved performance of the shape-aware network can prove critical in the correct computation of these heights for the fractured vertebra. Similarly, conditions like osteophytes are also better segmented when the shape-aware term has been used (see Fig. 6.18). Moreover, the formulation of the shape-aware term is general and can be adapted to any other anatomies in medical images or objects in other domains where preservation of the shape is critically important.

Although the boundary detection and the segmentation performance achieved in this chapter is very promising, two critical observations can be made from the qualitative results. First, the predictions of the deep networks are discontinuous and disjoint. The effect can be seen for boundary detection in Fig. 6.10d, Fig. 6.11a and Fig. 6.11b. Similar effects can also be noticed for segmentation with multiple disjoint predicted regions and/or holes inside the vertebral body in Fig. 6.17c:SegNet, 6.17h:SegNet and Fig. 6.18b:SegNet-S, 6.17c:SegNet-S. Second, although the shape-aware term helps, the predictions of the SegNet-S are still not constrained to strictly produce vertebra-like structures (Fig. 6.17i, 6.17j, 6.18d, 6.18e and 6.18f). Both of these issues can be attributed to the use of the loss functions for these networks which are defined in the pixel-space. The proposed Bhattacharyya coefficient-based loss function and the novel shape-aware term also works in a pixel-wise manner, thus, cannot completely solve the problem. In the next chapter, we propose a novel network that is trained on a loss function defined in the shape space, solving the issues discussed above.

Chapter 7

Shape Prediction

In the previous two chapters, we have used the encoder-decoder UNet architectures for vertebral boundary detection and vertebral body segmentation. The boundary detection framework was not able to predict continuous and closed vertebral boundaries in all cases. The problem was solved by the segmentation framework proposed in the previous chapter. However, we discovered two issues with the segmentation results. First, the prediction of multiple disjoint regions and second, the prediction did not always resemble vertebra-like structures. Both of these issues can be attributed to the loss function used for training the network which is defined in a pixel-wise manner. To solve these issues, in this chapter, we propose to predict shapes with the spatial encoder-decoder architecture. A novel loss function is proposed which computes the loss directly in the shape domain. The proposed shape predictor network outperformed the segmentation framework of the previous chapter both qualitatively and quantitatively.

7.1 Overview

Most of the work in vertebrae segmentation involves shape prediction. Active shape models [119] and level set-based segmentation models [134] have long been used in segmentation of objects in medical images [24, 49, 57, 69, 90, 135, 136]. Given the fact that a vertebra in an X-ray image mostly consists of homogeneous and noisy image regions separated by edges,

active shape model and level set-based segmentation methods try to converge to the edges and separate the two regions. While these methods work relatively well in many medical imaging modalities, the discontinuity of the edges in the vertebra and lack of the difference in image intensities and contrasts inside and outside the vertebra limits the performance of these methods in our challenging and real-life X-ray image datasets.

In Chapter 6, we used the UNet architecture for vertebral boundary detection and vertebral body segmentation. The segmentation framework solved the discontinuous boundary problem of the boundary detection framework to some extent, however, it failed to capture the high-level topological shape information. Moreover, it produced multiple regions and shapes that do not resemble possible vertebra-like structures. Our goal in this chapter is to learn the mapping between vertebral image patches and shapes directly. To this end, we use the same architecture used in the previous chapters and modify it to generate a shape representation function instead of dense classification probabilities or spatially distributed probabilities over the input image space.

The active shape model learns shape representation from a point cloud model of the object boundaries and requires a point to point correspondence during convergence on new example. Since our network's prediction is defined over the same input image space, converting this prediction to a point-based model with a point to point correspondence is not differentiable, and thus, end-to-end learning using back-propagation is not possible. Alternatively, the level set method proposes a different shape representation where shapes are represented implicitly by a signed distance function (SDF). The SDF is defined over the same input image space (Sec. 7.2.1). This gives a straightforward way to use our UNet architecture.

In this chapter, we modify the UNet architecture to generate an SDF from the input image. The predicted SDFs are converted to the shape parameters, and the loss is computed in the shape parameter domain. These shape parameters are related to the modes of variation of a set of training shapes which are computed based on the principal component analysis (PCA).

Sec. 7.2 describes how the manual annotation of the vertebral shapes are converted to SDFs and corresponding shape parameters. The modification of the UNet architecture and the loss function are discussed in Sec. 7.3. The following sections describe the experimentations and results, before ending the chapter with the conclusion.

7.2 Ground Truth Generation

7.2.1 Level-set Basics

In the level set segmentation method the shapes are represented implicitly by an auxiliary function, $\Phi(.)$. The shape, S, is denoted as the zero-level set of that function:

$$\boldsymbol{S} = \{\boldsymbol{p} | \Phi(\boldsymbol{p}) = 0\}, \tag{7.1}$$

where $\boldsymbol{p} \in \Omega_p$ and Ω_p is pixel-space over which the function is defined. The function, $\Phi(.)$, is a signed distance function (SDF) which is defined as:

$$\Phi(\boldsymbol{p}) = \begin{cases} d(\boldsymbol{p}, \boldsymbol{S}) & \text{if } \boldsymbol{p} \in \Omega_v^c \\ -d(\boldsymbol{p}, \boldsymbol{S}) & \text{if } \boldsymbol{p} \in \Omega_v \end{cases}, \tag{7.2}$$

where Ω_v is the set of pixels inside the object, which is a vertebra in our case, ^{c} represents the complement set and d is defined as:

$$d(\boldsymbol{p}, \boldsymbol{S}) := \inf_{\boldsymbol{x} \in \boldsymbol{S}} D(\boldsymbol{p}, \boldsymbol{x}), \tag{7.3}$$

where inf denotes infimum and $D(\boldsymbol{a}, \boldsymbol{b})$ denotes the Euclidean distance between pixel position $\boldsymbol{a}$ and $\boldsymbol{b}$.

7.2.2 Conversion of Manual Annotations to SDFs

The models in this chapter have been trained on the same augmented training data used in Chapter 6. Manual annotation for each of the 26,370 training vertebrae is converted into a signed distance function. To convert the vertebral shapes into a signed distance function the pixels lying on the manually annotated vertebral boundary curve have been assigned zero values. Then all other pixels are assigned values according to Eqn. 7.2, where S represents the set of pixels with zero values. A few examples of training vertebrae with corresponding zero-level set pixels and SDFs are illustrated in Fig. 7.1.

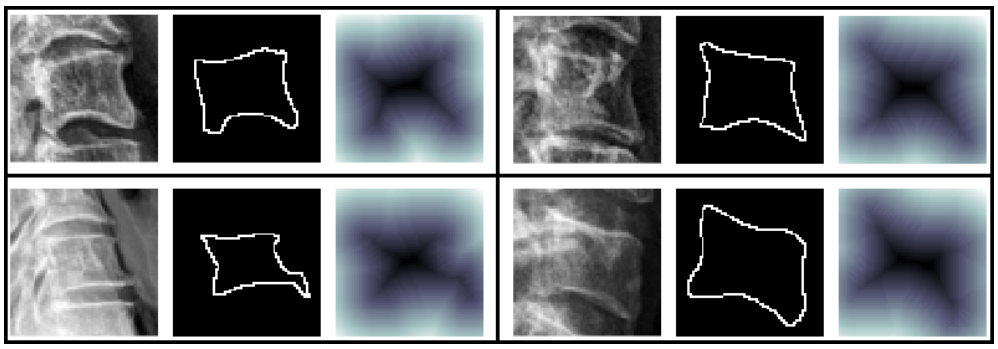


Fig. 7.1 Examples of training vertebrae: original image (left), pixels at the zero-level set of the SDF (center) and the SDF (right). Darker tone represents negative values.

7.2.3 Principal Component Analysis and Shape Parameters

Once all the training vertebral shapes are converted to corresponding signed distance functions, we can apply principal component analysis on the SDFs. First, we compute the mean SDF, $\bar{\Phi}$, as:

$$\bar{\Phi} = \frac{1}{N}\sum_{n=1}^{N}\Phi_n, \tag{7.4}$$

where N is the number of training samples. We then extract the difference SDF (Φ_{d_n}) by subtracting the mean ($\bar{\Phi}$) from each SDF (Φ_n):

$$\Phi_{d_n} = \Phi_n - \bar{\Phi}. \tag{7.5}$$

The vectorized Φ_{d_n} are then arranged in a matrix, M:

$$\boldsymbol{\phi}_{d_n} = vec(\Phi_{d_n}), \tag{7.6}$$

$$M = [\boldsymbol{\phi}_{d_1} | \boldsymbol{\phi}_{d_2} | .... | \boldsymbol{\phi}_{d_N}]. \tag{7.7}$$

The covariance matrix, C_M can then be computed as:

$$C_M = \frac{1}{N} M M^T. \tag{7.8}$$

The principal components of the variations of the training data can be extracted by singular value decomposition (SVD) of the matrix C_M:

$$[W, \Sigma, W_v^T] = svd(C_M), \tag{7.9}$$

where Σ is a diagonal matrix containing eigenvalues corresponding to the eigenvectors, which are arranged in a column-wise manner in W. The eigenvectors are sequentially arranged based on their corresponding eigenvalues. Now, each shape in the training data can be represented by the mean shape ($\bar{\boldsymbol{\phi}}$), matrix of eigenvectors (W) and a vector of shape parameters, $\boldsymbol{b}_n$:

$$\boldsymbol{\phi}_n = \bar{\boldsymbol{\phi}} + W \boldsymbol{b}_n. \tag{7.10}$$

For each training example we can compute $\boldsymbol{b}_n$ as:

$$\boldsymbol{b}_n = W^T (\boldsymbol{\phi}_n - \bar{\boldsymbol{\phi}}) = W^T \boldsymbol{\phi}_{d_n}. \tag{7.11}$$

These parameters are used as the ground truth ($\boldsymbol{b}_n^{GT}$) for training the proposed network. In the next section, we describe a deep UNet which is trained to produce the difference SDFs ($\hat{\Phi}_{d_n}$). $\hat{\Phi}_{d_n}$ is the matrix form of the vector $\hat{\boldsymbol{\phi}}_{d_n}$. We also propose a novel loss function that computes the error between the prediction and ground truth in the shape parameter domain.

For SDFs defined over a pixel space of size 64×64 and a training dataset with N samples, the dimensionality of the matrices and vectors discussed in this section are summarized in Table 7.1.

Dimension	Matrix/Vector
64×64	Φ_n, $\bar{\Phi}$, Φ_{d_n}
$4096 \times N$	M
4096×1	$\boldsymbol{\phi}_n$, $\bar{\boldsymbol{\phi}}$, $\boldsymbol{\phi}_{d_n}$, $\boldsymbol{b}_n$
4096×4096	C_M, W, V, U

Table 7.1 Dimensionality of different matrices and vectors.

7.3 Methodology

In the previous chapters, we have shown capabilities of the modified UNet to produce different spatial outputs. The expanding path and the concatenation of the data from the contracting path allows the network to produce outputs with the same resolution of input. Here, we want our proposed network, LevelSet-UNet or in short LS-UNet, to take a 64×64 vertebral image patch as input and produce its related difference SDF ($\hat{\Phi}_d$) which is also defined over the same pixel space (for simplicity the mathematics in this section is described for a single input image patch and the subscript $_n$ has been dropped). We use the same network architecture as we have used in the probabilistic spatial regressor network used in Chapter 5. The final normalization block, which used to convert the final activation to a valid probabilistic distribution has been removed. The last convolution layer outputs the difference signed distance function ($\hat{\Phi}_d$) which is then sent to the final layer where is it converted to shape parameter vector ($\hat{\boldsymbol{b}}$) and compared with the ground truth ($\boldsymbol{b}^{GT}$). The UNet that produces $\hat{\Phi}_d$ is illustrated in Fig. 7.2 and the final layer has been depicted in Fig. 7.3.

The forward pass through the final layer can be summarized below. First, the output of the last convolution layer of the UNet ($\hat{\Phi}_d$) is vectorized:

$$\hat{\boldsymbol{\phi}}_d = vec(\hat{\Phi}_d). \tag{7.12}$$

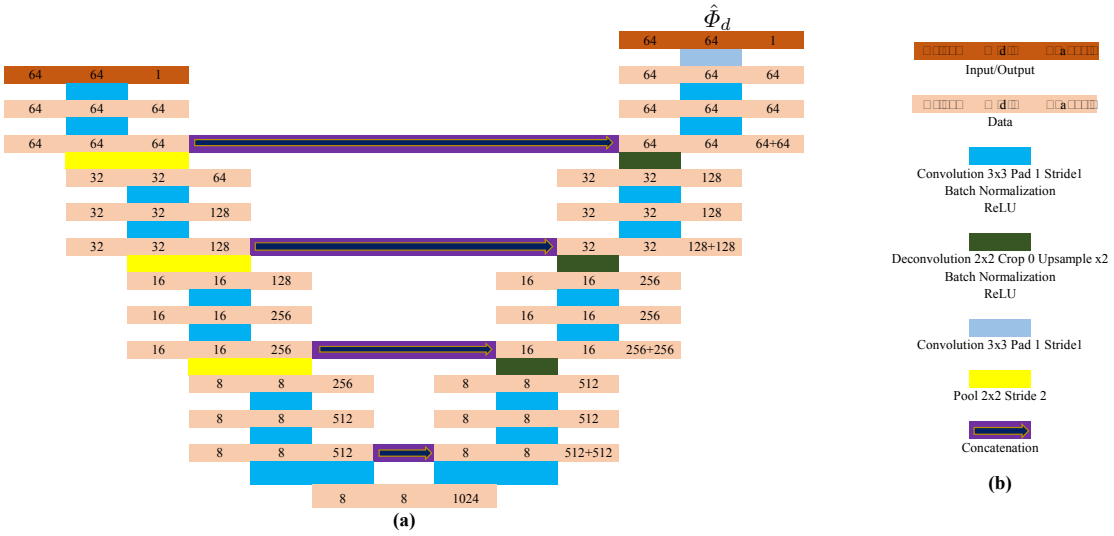


Fig. 7.2 UNet for shape prediction (a) network layers (except the final layer) (b) legend.

Then the final prediction of network is computed as $\hat{\boldsymbol{b}}$:

$$\hat{\boldsymbol{b}} = W^T \hat{\boldsymbol{\phi}}_d, \tag{7.13}$$

or in element-wise form:

$$\hat{b}_i = \sum_{j=1}^{k} w_{ij} \hat{\phi}_{d_j}, \tag{7.14}$$

where w_{ij} is the value at the i-th row and j-th column of the transposed eigenvector matrix (W^T) and k is the number of eigenvectors. Finally, the loss is defined as:

$$L = \sum_{i=1}^{k} L_i, \tag{7.15}$$

where

$$L_i = \frac{1}{2}(\hat{b}_i - b_i^{GT})^2. \tag{7.16}$$

The total number of eigenvectors is 4096. Thus, we have the same number of shape parameters. For back-propagation, the partial derivative of Eqn. 7.16 with respect to the input variable $\hat{b}_i$ can be expressed as:

$$\frac{\partial L_i}{\partial \hat{b}_i} = \hat{b}_i - b_i^{GT}. \tag{7.17}$$

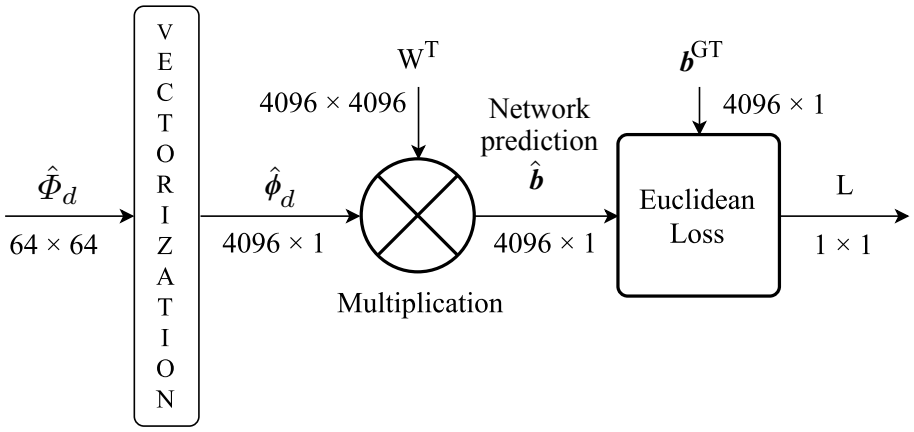


Fig. 7.3 Final layer.

Similarly, the partial derivative of Eqn. 7.14 with respect to the input, $\hat{\phi}_{d_j}$, can be expressed as:

$$\frac{\partial \hat{b}_i}{\partial \hat{\phi}_{d_j}} = w_{ij}. \tag{7.18}$$

7.4 Experiments

The proposed network (LS-UNet) has been trained on a system with a NVIDIA Pascal Titan X GPU for 30 epochs with a batch-size of 50 images. The network took approximately 22 hours to train. We have also implemented a traditional convolutional neural network (CNN) where we predict the shape parameter vector $\boldsymbol{b}$ directly using a Euclidean loss function. The network consists of the contracting path of the proposed UNet architecture, followed by two fully connected (FC) layers which regress the 4096 b-parameters at the output. This network will be mentioned as LS-FCNet in the following discussions. The LS-UNet has only 24,237,633 parameters where the LS-FCNet network has 110,123,968 trainable parameters. The FC layers cause a significant increase in the number of parameters. We have also shown results of vertebral shape prediction based on Chan-Vese level set segmentation method (LS-CV) [90, 136, 137]. However, this method is a parametric method and finding a common set of parameters for a challenging dataset like ours was difficult. A grid search method was followed to find a common set of parameters on a validation set of 40 images with 177 vertebrae. The images in validation set were collected recently and is not included in our

training and test dataset. We also had to constrain the b-parameters to 1.2 times the standard deviation to produce acceptable results for this method. Apart from these, we also compare our results with the segmentation results of the dense classification networks from Chapter 6, referred to as SegNet and SegNet-S. The foreground predictions of these networks have been converted into shapes by tracking the boundary pixels. For both of the deep level set networks, the predicted b-parameters are converted into a signed distance function following Eqn. 7.10. The final shape is found by locating the zero-level set of this function.

We compare the predicted shapes with the ground truth shapes using two error metrics. First, the average point to ground truth curve (E_{p2c}) error defined in Eqn. 6.10. Second, the Hausdorff distance (d_H) [125] between the prediction and ground truth shapes. This metric is defined in Eqn. 7.19. The E_{p2c} represents on-average how far the predicted shape points are from the ground truth, the second metric (d_H) denotes what is the maximum difference between the shapes. Both metrics are reported in pixels.

$$d_H(\hat{S}, S_{gt}) = max\{\sup_{x \in \hat{S}} \inf_{y \in S_{gt}} D(x,y), \sup_{y \in S_{gt}} \inf_{x \in \hat{S}} D(x,y)\}, \tag{7.19}$$

where $\hat{S}$ is set points in the predicted shape, S_{gt} is set of points in the manually annotated vertebral boundary curve, sup represents the supremum, inf represents the infimum and $D(x,y)$ is the Euclidean distance between the point x and y.

7.5 Results

We compare the three level set-based methods in Table 7.2. We report the mean and standard deviation of the metrics over 797 test vertebrae. The Chan-Vese method (LS-CV) achieves an average E_{p2c} of 3.11 pixels, where the fully connected version of the deep network (LS-FCNet) achieves 2.27 pixels and the proposed UNet-based network (LS-UNet) achieves 1.16 pixels only. Hausdorff distance (d_H) shows more difference between the LS-CV and the deep networks.

Metrics	Average E_{p2c}		Average d_H	
Methods	Mean	Std	Mean	Std
LS-CV	3.11	1.13	10.94	3.68
LS-FCNet	2.27	0.83	6.74	3.25
LS-UNet	**1.16**	**0.66**	**4.11**	**3.13**

Table 7.2 Comparison of deep shape predictor networks with the Chan-Vese model.

Both of these deep networks have been trained to regress all 4096 shape parameters. These parameters are related to the 4096 eigenvectors or modes of variations. The eigenvalues represent the variance in the training data along the corresponding eigenvectors. As the eigenvectors are ranked based on their eigenvalues, eigenvectors with small eigenvalues are often results of noise and can be ignored. In Table 7.3, we report performance of our proposed LS-UNet on the validation set of 177 vertebrae when we consider a certain percentage of total variations at test time. The second row of the table indicates how many parameters are left when a certain percentage of variation is considered. Other parameters are simply replaced with zeros when converting back to the signed distance function. It can be seen that the lowest errors are found when 98% of the total variation is considered and only 18 b-parameters are kept.

Variation (%)	90	95	98	99	99.5	99.8	100
No. of parameters	6	9	18	30	51	117	4096
Average E_{p2c}	1.34	1.23	**1.16**	1.19	1.21	1.23	1.25
Average d_H	4.83	4.62	**3.98**	4.15	4.31	4.49	4.68

Table 7.3 Effect of number of eigenvectors on errors for LS-UNet.

Based on this insight, we modified both versions of our level set-based deep networks to regress only 18 b-parameters and retrained the networks from randomly initialized weights. We report the performance of the retrained networks in Table 7.4. We also report the metrics for SegNet and SegNet-S networks from Chapter 6. It can be seen that our proposed LS-UNet-18, outperforms all other networks quantitatively. However, the improvement over SegNet-S is minuscule. Table 7.5 reports the results of the statistical significance test between our proposed LS-UNet-18 and all other methods reported in Table 7.4. It can be seen that the quantitative improvement of LS-UNet-18 over SegNet-S in terms of the E_{p2c} metric is not

statistically significant according to the paired t-test at a 5% significance level. However, the improvement in terms of Hausdorff distance (d_H) passes the significance test.

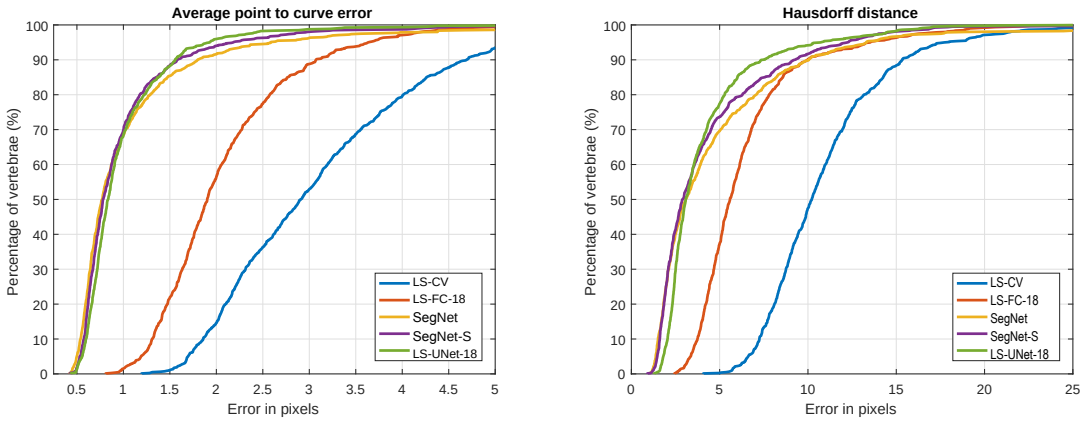
Metrics	Average E_{p2c}		Average d_H		$nVmR$	Fit failure (FF_s) %
Methods	Mean	Std	Mean	Std		
LS-CV	3.107	1.13	10.94	3.68	**0**	85.45
SegNet	1.114	1.29	5.06	6.11	57	8.53
SegNet-S	0.999	0.67	4.37	4.02	45	6.02
LS-FCNet-18	2.082	0.78	6.48	3.32	**0**	43.54
LS-UNet-18	**0.996**	**0.55**	**4.17**	**3.06**	**0**	**4.14**

Table 7.4 Quantitative comparison of different methods.

LS-UNet-18 compared with following methods:	Average E_{p2c}		Average d_H	
	h	p-value	h	p-value
LS-CV	1	$< 10^{-282}$	1	0
SegNet	1	0.003	1	$< 10^{-05}$
SegNet-S	0	0.827	1	0.035
LS-FCNet-18	1	$< 10^{-255}$	1	$< 10^{-151}$

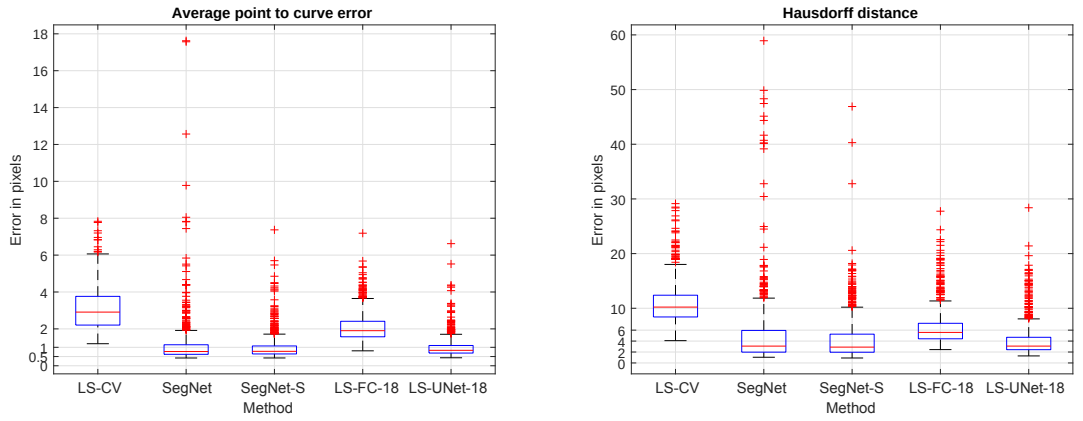
Table 7.5 Statistical significance test (t-test).

Another benefit of our proposed LS-UNet network over the original SegNet and SegNet-S is that the loss is computed in the shape domain, not in a pixel-wise manner. In the fifth column of the Table 7.4, we report the number of test vertebrae with multiple disjoint predicted regions ($nVmR$). The pixel-wise loss function-based networks learn the vertebral shape implicitly but this does not prevent multiple disjoint predictions for a single vertebra. The SegNet and SegNet-S produce 57 and 45 vertebrae, respectively with multiple predicted regions, whereas the proposed network does not have any such example indicating that the topological shape information has been learned based on the seen shapes. Few examples of these can be found in Fig. 7.7 and Fig. 7.8. We have also reported the fit failure (FF_s) for all the compared methods. Like in Chapter 6, the FF_s is defined as the percentage of the test vertebrae having an E_{p2c} of greater than 2 pixels. The proposed LS-UNet-18 achieves the lowest FF_s. The cumulative error curves and boxplots of the metrics are shown in Fig. 7.4 and Fig. 7.5, respectively. The proposed method achieves noticeable improvement in terms of Hausdorff distance (d_H). However, in terms of the E_{p2c} metric, the performance of the



(a) (b)

Fig. 7.4 Cumulative error curves (a) average point to curve error (E_{p2c}) and (b) Hausdorff distance (d_H).



(a) (b)

Fig. 7.5 Boxplots of quantitative metrics (a) average point to curve error (E_{p2c}) and (b) Hausdorff distance (d_H) on the right.

proposed method is very close with the SegNet and SegNet-S. Especially for E_{p2c} less than 1.5 pixels, both SegNet and SegNet-S marginally outperform LS-UNet-18.

However, the qualitative results in Fig. 7.6, 7.7 and 7.8 clearly show the benefit of using the proposed method. The SegNet and SegNet-S predict a binary mask and the predicted shape is located by tracking the boundary pixels. This is why the shapes are not smooth. In contrast, the level set-based methods predict b-parameters which are then converted to signed

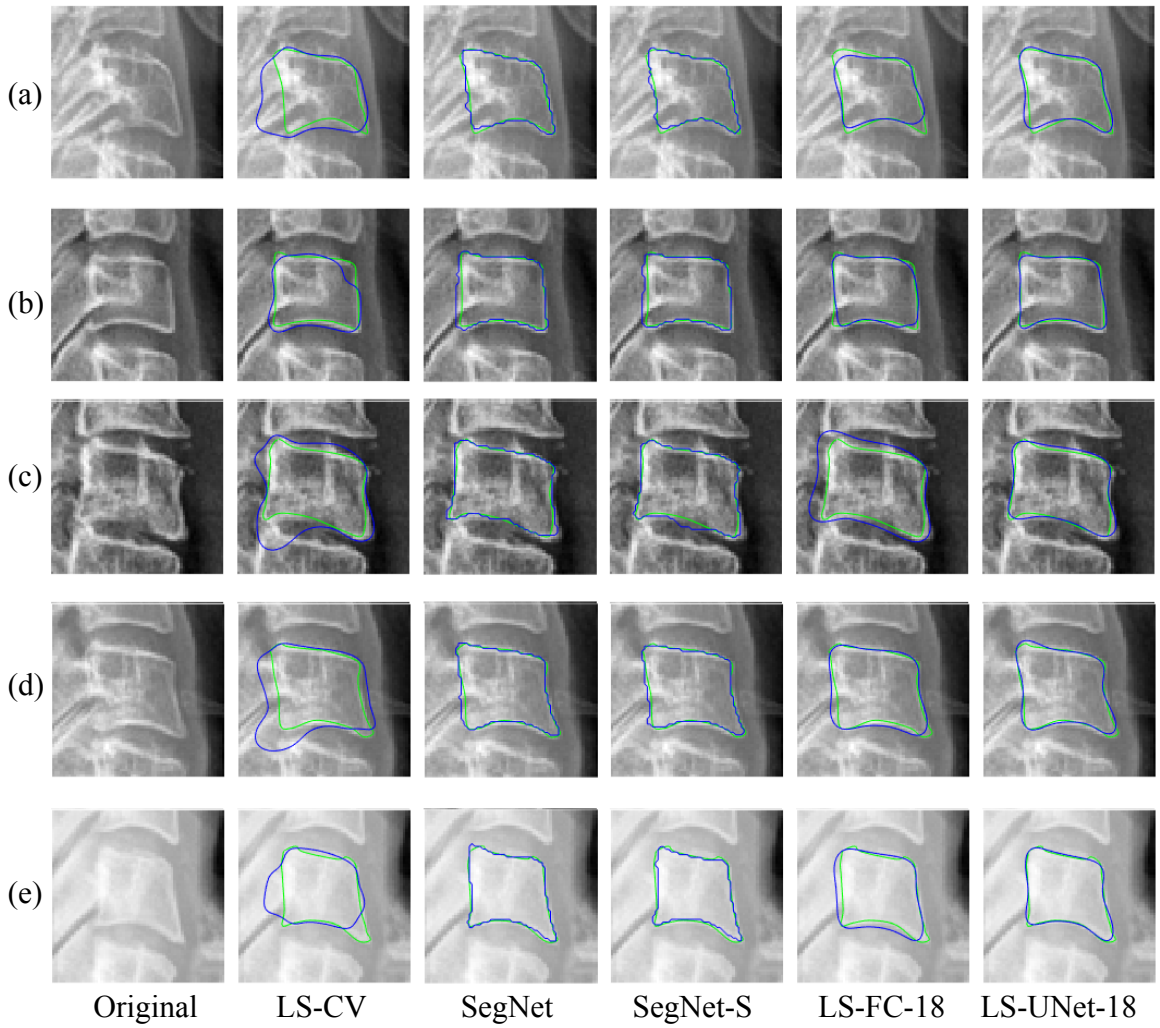


Fig. 7.6 Qualitative results for comparatively less challenging examples. The predicted shape is plotted in blue and the ground truth in green.

distance functions. The shape is then located based on the zero-level set of this function, resulting in smooth vertebral boundaries defined to the sub-pixel level which resembles the manually annotated vertebral boundary curves.

The worst performance is exhibited by the Chan-Vese methods. As mentioned earlier, finding a common set of parameters for our complex dataset was difficult, also lack of contrast in the anterior side of the vertebrae affects the results severely. The b-parameters predicted by the Deep LS methods were not constrained. The results of LS-FCNet-18

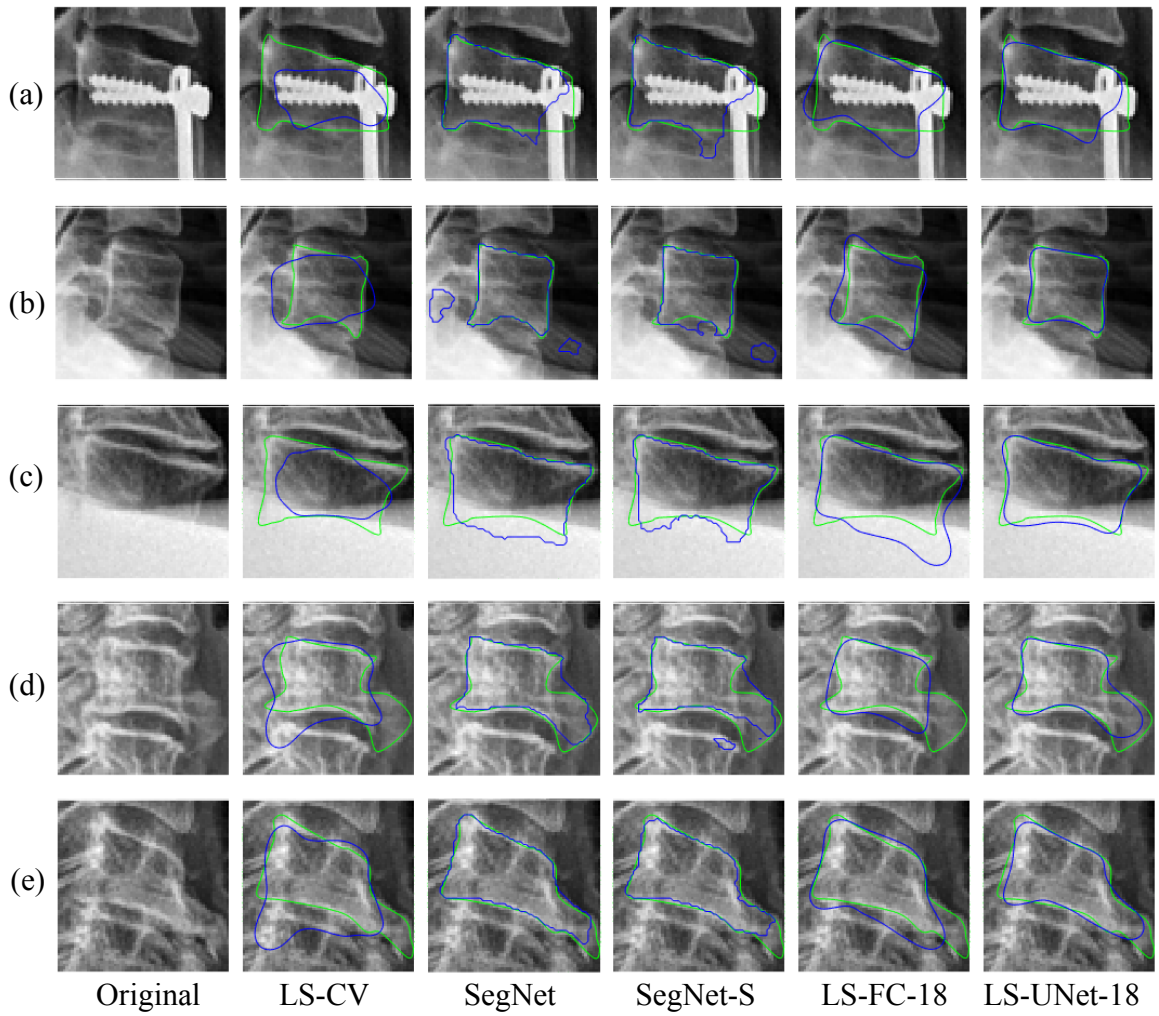


Fig. 7.7 Qualitative results for challenging examples. The predicted shape is plotted in blue and the ground truth in green.

is better than the traditional Chan-Vese model, but not comparable with the UNet-based methods. The reason can be attributed to the loss of spatial information because of the pooling operations. The UNet-based methods recover the spatial information in the expanding path by using concatenated data from the contracting path, thus performs much better than the fully connected version of the deep networks. We have shown some examples in Fig. 7.6, where the input vertebrae have less variations and better contrast. Harder examples are shown in Fig. 7.7 and Fig. 7.8. Examples with bone implants (Fig. 7.7a, 7.8a), abrupt contrast change (Fig. 7.7b, 7.7c), clinical conditions (Fig. 7.7d, 7.7e, 7.8d), image artefacts (Fig. 7.8d, 7.8e)

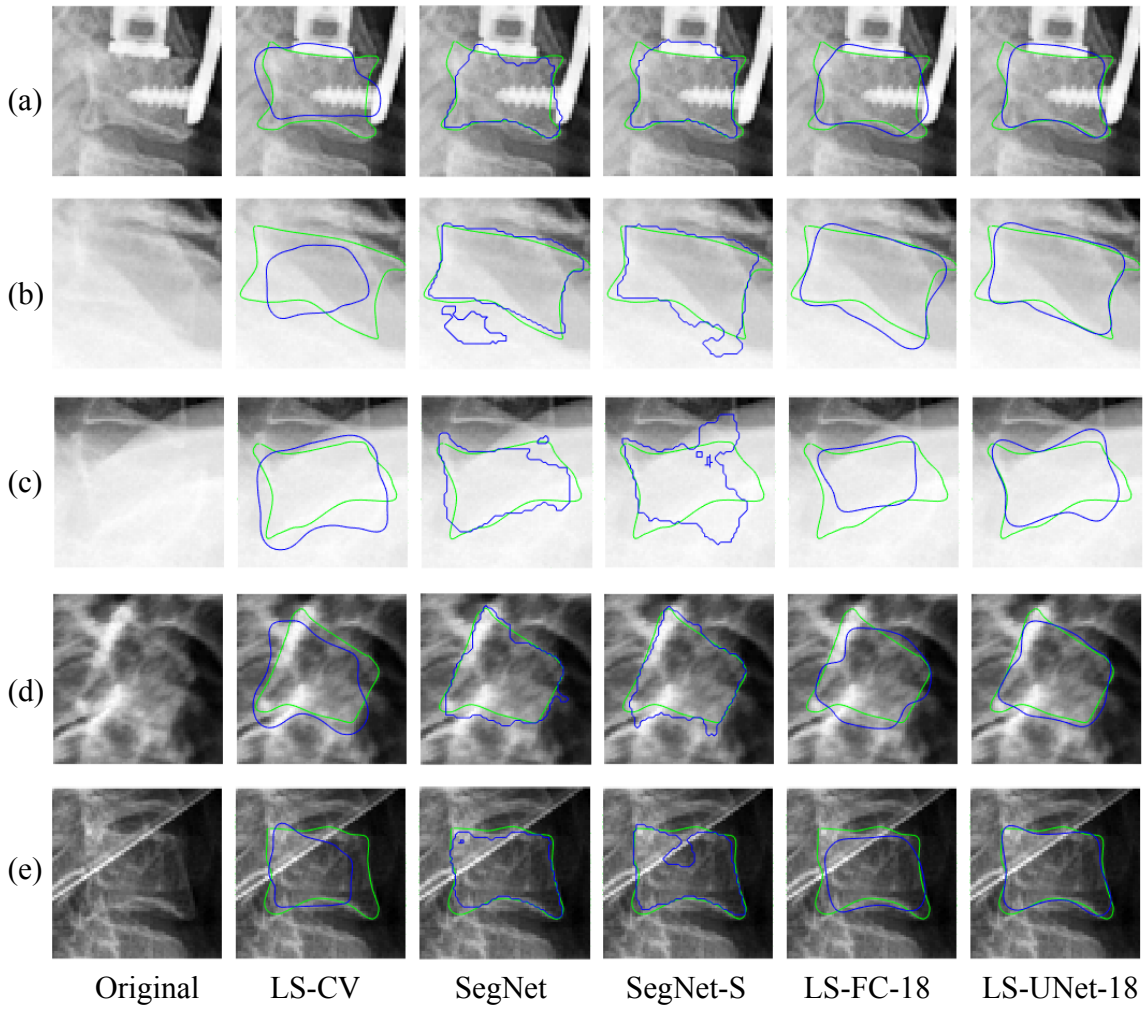


Fig. 7.8 Qualitative results for challenging examples.

and low contrasts (Fig. 7.8b, 7.8c) can be found in the qualitative results. It can be seen even in difficult situations like in Fig. 7.7 and Fig. 7.8, the LS-UNet-18 method predict shapes which resembles a vertebra where the pixel-wise loss function-based SegNet and SegNet-S

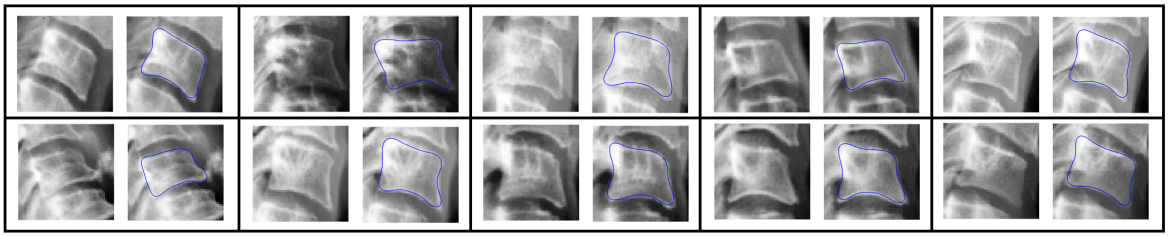


Fig. 7.9 Qualitative results from NHANES-II dataset using LS-UNet-18.

predict shapes with unnatural variations. The LS-UNet-18 has also been tested on vertebra from the NHANES-II dataset. A few examples from this dataset are shown in Fig. 7.9.

7.5.1 Corner Localization from Predicted Shapes

The framework discussed so far predicts $\hat{b}$-parameters, which is converted to corresponding signed distance function defined over a 64×64 pixel space. The predicted shape is then localized by locating the zero-level set of this function. The final shape can then be defined by a set of 200 evenly spaced points arranged sequentially in the clockwise direction. Inspired by earlier literature in the topic of corner detection [99], we can use these points to localize corners in the predicted shapes.

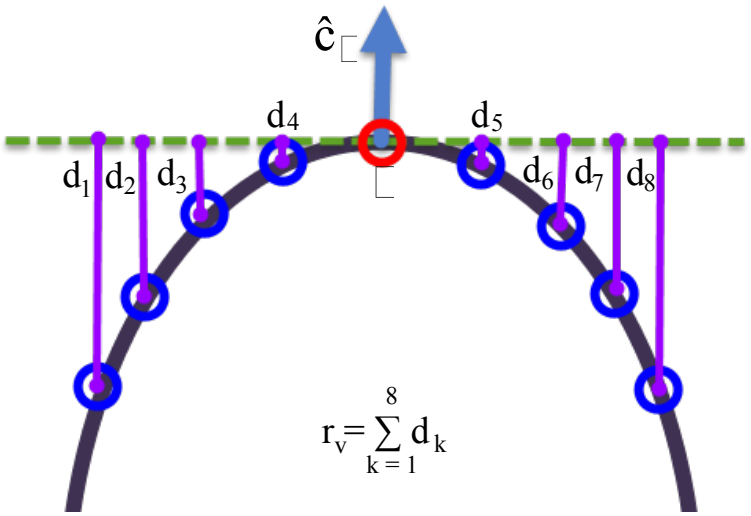


Fig. 7.10 Computing curvature of a point.

Since test vertebral image patches are extracted based on the manually clicked vertebral centers, the orientation is always upright. Thus it is safe to assume that the four vertebral corners are located in the four quadrants of the patch. Based on this assumption, we can divide the predicted shape into four parts. We then compute the curvature of each point in the shape following [138]. To compute the curvature, r_v, for a single point $\boldsymbol{v}$, we compute the summation of the distances from a 4-point neighborhood on each side of $\boldsymbol{v}$ to the line running through $\boldsymbol{v}$ and orthogonal to the surface normal $\hat{\boldsymbol{c}}_v$. This curvature computation process for

a single point is illustrated in Fig. 7.10. The point with the highest curvature from each shape quadrant is considered as the localization corner. The corner localization process is summarized in Fig. 7.11.

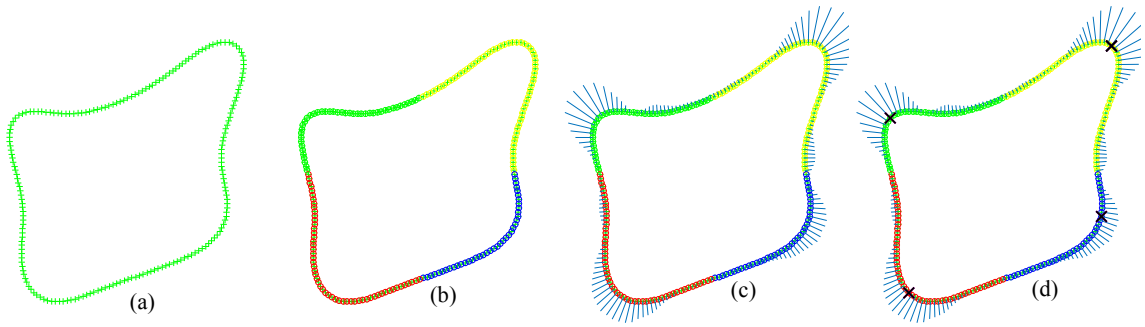


Fig. 7.11 Localization of corners from predicted shapes (a) predicted shape points (b) shape points divided into four quadrants (c) curvature magnitude plotted as a line in the normal direction (d) corners ($\times$) localized based on the maximum curvature magnitude in each quadrant.

Once the corners are located at the patch level, based on the known patch extraction process, they can be transformed back on the original image space and compared with the manually annotated corners. We report the *P2P* and *P2C* errors defined in Sec. 5.4 for evaluating corner localization frameworks, in Table 7.6. We have achieved a 3.9% relative improvements in terms of average *P2P* error and a drop of 4.08% in fit failure (FF_c)[1] from the probabilistic spatial regressor network (PSRN)-based corner localization framework proposed in Chapter 5. The improvement can be attributed to the fact that the localized corners are extracted from the predicted shapes, whereas for the PSRN-based framework, the corners were localized from a probability distribution. As the predicted distribution does not only spread on the actual vertebral boundary, the localized center has the possibility of staying a few pixel away from the actual boundary. This error is minimized to some extent as we are localizing the corner from the predicted shapes. However, the improvement in terms of the *P2C* is not noticeable. The average *P2C* error is the same with a lower standard deviation. Similar corner localization could also be performed from the detected vertebral

[1]Note that, the fit failure (FF_c) discussed in this subsection is different from the fit failure (FF_s) reported in earlier in Table 7.4. The FF_s is for shapes and defined in terms of pixels, whereas, FF_c is for corners and defined in terms of millimeters (see Sec. 5.4).

	Point to point ($P2P$) mm				Point to curve ($P2C$) mm	
Method	Mean	Std	Fit failure (FF_c) %		Mean	Std
PSRN	1.54	1.74	11.7		0.58	0.76
LS-UNet-18	1.48	1.26	7.62		0.58	0.61

Table 7.6 Corner localization from LS-UNet-18.

boundaries and segmented vertebral bodies from Chapter 6. However, both the boundary detection and the segmentation networks generate the pixel-level results. The pixelation artifacts are noticeable in the binary predictions of the boundary detection networks (see Fig. 6.10 and 6.11) and the segmentation networks (see Fig. 6.17, 6.18, 7.7 and 7.8). Thus the computation of the corners would have been coarse and erroneous. A post-processing smoothing of the pixelated shapes could have solved the issue to some extent but would make the shapes inaccurate and dependent on the accuracy of the smoothing process.

7.6 Conclusion

In this chapter, we have proposed a novel deep network capable of predicting vertebral shapes. The network learns to predict signed distance functions over the input image space which represents the vertebral shapes implicitly at the zero-level set. The proposed network has shown excellent qualitative improvement in performance over other deep architectures. Quantitatively, it produced a comparative performance with the SegNet-S mask prediction network. However, the mask predictor network fails to produce smooth vertebral boundaries and to constrain the prediction within the possible vertebra-like shapes when the input image is of poor quality. The proposed shape predictor network is also able to identify that the predicted shape should be a single connected region indicating the benefit of the proposed loss function which computes the error in the shape domains. Additionally, the predicted shapes have also been used to localize vertebral corners. The detected corners from the predicted shapes outperformed corner localization methods proposed in Chapter 5. In future, the predicted shapes can be used for automatic detection of vertebral fractures, spinal mis-

alignment, and other clinical conditions.

Shape is an important characteristic of an object and a fundamental topic in computer vision. In object segmentation, shape has been widely used in methods to constrain a segmentation result to a class of learned shapes [119, 134]. Although most of the topics in computer vision have been revolutionized by the advent of deep learning, shape prediction was mostly untouched. In this chapter, we have proposed a novel deep learning-based approach for shape prediction. The proposed method has been applied to a vertebral body segmentation problem and achieved a state-of-the-art performance. The deep shape predictor network designed in this chapter has been trained to predict shapes for a single object. However, the network is inherently capable of predicting multiple objects in the same image. This capacity comes because of the use of signed distance function to represent the shapes. By default, this function is capable of capturing topological changes of the shapes in terms of the number of regions and/or objects. Given a dataset of input images and corresponding ground truth with multiple and a variable number of objects per image, the same network with the same loss function can be trained.

So far in this dissertation, we have proposed separate methods for spine localization (Chapter 3), center localization (Chapter 4), corner localization (Chapter 5), boundary detection (Chapter 6), segmentation (Chapter 6) and shape prediction for cervical vertebrae. The proposed shape predictor, segmentation, boundary detector and corner localization frameworks require the vertebral centers to be given at the test time, making these frameworks semi-automatic. However, our center localization framework can localize center positions inside a localized spinal region, and our spine localization framework can localize spinal region without any human input. Thus, a fully automatic framework can now be built by connecting these frameworks appropriately. In the next chapter, we describe this fully automatic framework which was the primary research objective of this dissertation.

Chapter 8

Fully Automatic Framework

So far in this dissertation, we have solved several computer vision problems related to X-ray images of cervical vertebrae. Our anatomies of interest are the five cervical vertebrae in the spinal column: C3-C7. We have described semi-automatic methods for highlighting different features of these vertebrae: corners in Chapter 5, boundaries and vertebral bodies in Chapter 6 and vertebral shapes in Chapter 7. These methods work on the extracted test vertebral image patches of size 64×64 pixels. The extraction process is based on the manually annotated vertebral centers and thus, the methods were semi-automatic. To make the process fully automatic, the center localization algorithm of Chapter 4 can be utilized which can predict vertebral centers if the region of the spine in X-ray image has been localized. Moreover, this spinal region can be localized using the spine localization framework described in Chapter 3.

Having all the frameworks in place, we now have the capability of building a fully automatic image analysis framework for the cervical vertebrae in X-ray images. However, there are a few missing links which are addressed in the following sections.

8.1 Connecting the Dots

If the manually annotated vertebral centers are available, the test vertebral image patches are extracted based on the orientation vector ($\boldsymbol{F}$), which is described in Sec. 2.4.2. The patch

size depends on the magnitude of the vector and the orientation is given by its direction. This process works because when the manually annotated centers are available, the topological information about the identity of the vertebra is also known. Unfortunately, the vertebra identity is not available for the centers localized using the center localization algorithm. To solve the problem, based on our assumption that image is upright, we can arrange the predicted centers sequentially from top to bottom as C3 to C7 and proceed with the extraction process. This way we compute the orientation vector ($\boldsymbol{F}$). Although the direction of the vector is relevant for test patch extraction, the magnitude of this vector cannot be used as a measure of patch size because of the possibility of missing center in between two predicted centers. To solve this issue, a common patch size of 32 mm is chosen based on the distribution of training vertebral sizes. The vertebral size is defined as the distance from the vertebral center to the farthest point in the manually annotated vertebral boundary, similar to the base vertebral size discussed in Sec. 6.3. About 90% of the training vertebrae have a size smaller than 32 mm. A bigger patch size can increase this percentage. However, it would make a significant amount of the test vertebrae to appear smaller inside the extracted patch. This will adversely affect the following methods, as they are not trained on examples where vertebral size is smaller compared to the patch size. Fig. 8.1 shows the distribution of the vertebral sizes in our training dataset.

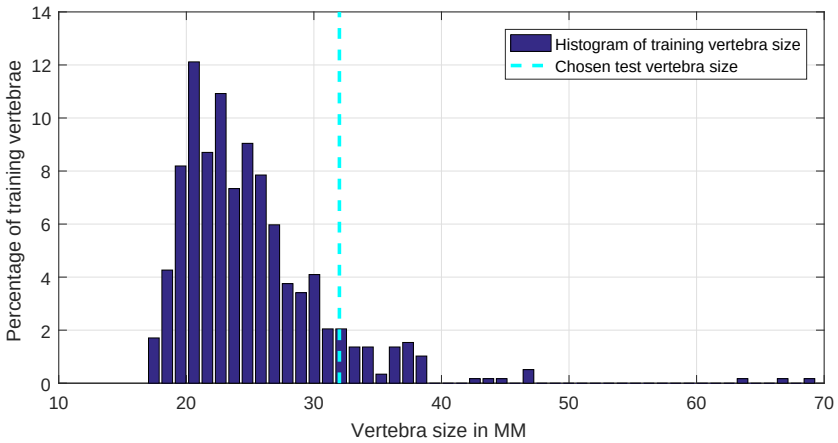


Fig. 8.1 Histogram plot of vertebral size in the training dataset.

Once we have solved the test patch extraction process, the patches can be extracted, and patch-level predictions can be generated. We have already discussed how to project back the patch-level predictions to original images space for corner localization and boundary detection frameworks in Sec. 4.3.3 and 5.3.4, respectively. The process involves affine transformation of the predictions using scaling, rotation, and translation which are known from the patch extraction process. These methods generate outputs defined over the 64×64 input image patch. The output of the vertebral segmentation framework is also similar and thus, can also be transformed back to the original image space using the same procedure. The shape prediction framework produces shape as a signed distance function. The resulting shape is given by the set of points at the zero-level set of this function. The set of points can also be scaled, rotated and translated accordingly to produce results on the original image space. Now, having the missing links solved, all the methods can be threaded together to produce a fully automatic image analysis tool for lateral cervical vertebrae in X-ray images.

8.2 Complete Framework

Given a high-resolution test image, the image can be zero-padded to form a square image and resized to 100×100 pixels. The resized image can be fed into the best performing spine localization network, FCN-R, to predict the spinal region. The network localizes the spinal region at the input resolution of 100×100 pixel, which can then be transformed back, i.e., resized and unpadded, to the original image. The process is summarized in Fig. 8.2:1.

Based on the spine localization result, a set of 45 patches are generated following the process described in Sec. 4.3.3. All the patches are then resized to 64×64 pixel and passed through the novel probabilistic spatial regressor network (PSRN) proposed in Chapter 4. Each patch generates a probability map of localized centers. These patch-level probabilities are then put back on the original image space. And the centers are localized using the post-processing steps of Sec. 4.3.3. Fig. 8.2:2 depicts the center localization process.

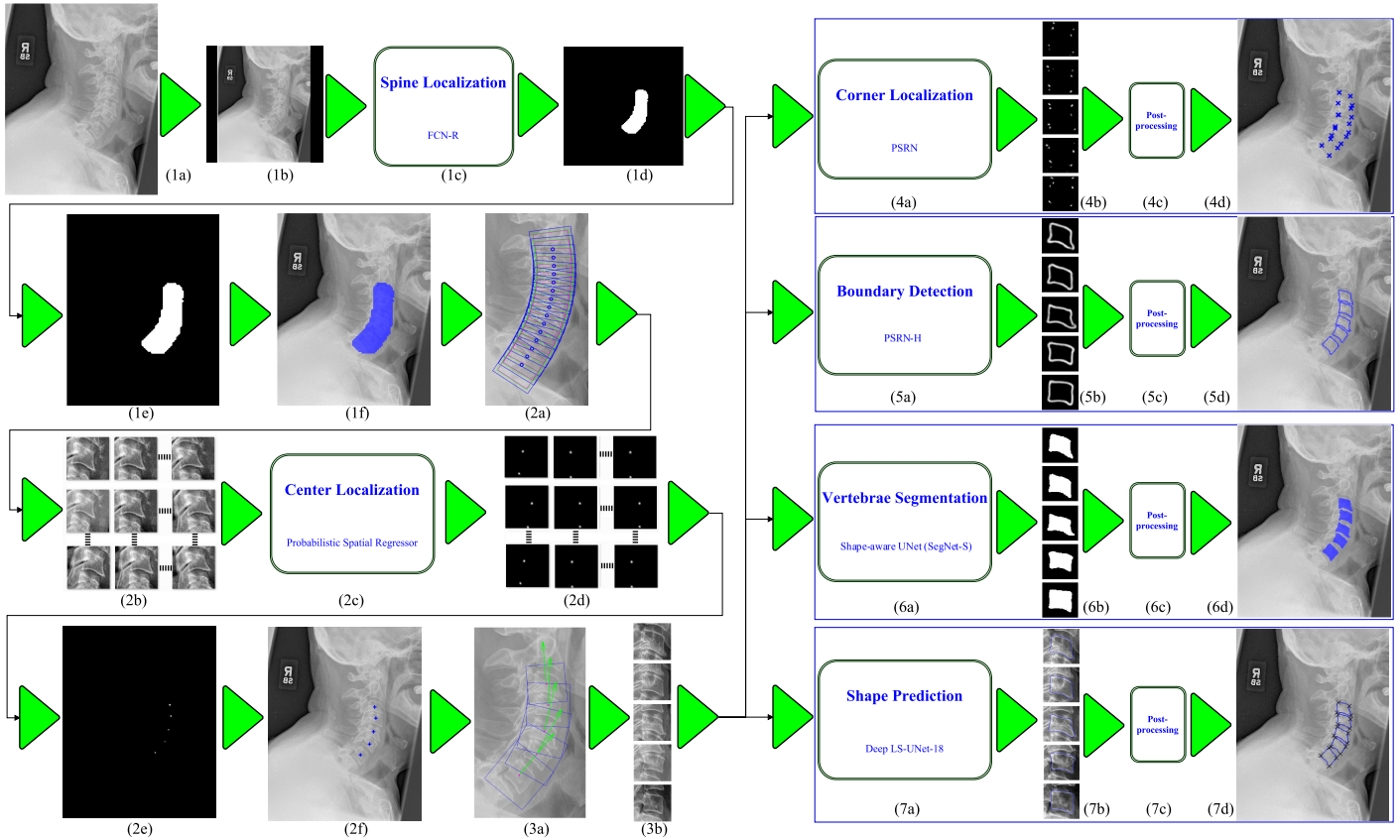


Fig. 8.2 Complete framework (1) spine localization: (1a) input image (1b) resized and padded image of size 100×100 (1c) region-aware spine localization network, FCN-R (1d) network output of size 100×100 (1e, 1f) image-level spine localization result (2) center localization (2a) patch extraction from localized spinal region (2b) extracted patches (2c) probabilistic spatial regressor network (PSRN) (2d) patch-level center probabilities (2e) image-level center probabilities (2f) localized centers (3a) vertebral image patch extraction (3b) extracted vertebral image patches (4) corner localization (4a) Bhattacharyya coefficient-based loss function equipped PSRN (4b) patch-level corner probabilities (4c, 4d) post-processing and image-level localized corners ($\times$) (5) boundary detection (5a) histogram-based normalization layer equipped PSRN (5b) patch-level edge probabilities (5c, 5d) post-processing and image-level vertebral boundaries (blue overlay) (6) segmentation (6a) shape-aware SegNet-S (6b) patch-level segmentation results (6c, 6d) post-processing and image-level segmented vertebrae (blue overlay) (7) shape prediction (7a) LS-UNet-18 (7b) patch-level predicted shapes (7c, 7d) post-processing and image-level predicted shape (blue) and localized corners ($\times$).

The predicted centers are sequentially arranged and a new set of vertebral test patches using the orientation vector ($\boldsymbol{F}$) and the fixed size discussed in Sec. 8.1 (Fig. 8.2:3a). One patch is extracted for each of the predicted centers. These patches are the again resized to the resolution of 64×64 and ready for further processing to generate final results (Fig. 8.2:3b).

For the next stage, we have four options: corner localization, boundary detection, segmentation and shape prediction. Each of the four methods produces a different output which can be used for various applications e.g., corners can be used for checking spinal alignment curve, predicted boundary probabilities could be used to detect the presence of osteophytes, segmented vertebral bodies and predicted shapes can be used to measure bone density, detect osteoporosis and other vertebral injuries. Also, while some of the information might be redundant in all four options, a user can prefer one output over other for further visual evaluation. As the goal of this dissertation is to produce a fully automatic image analysis tool for the cervical vertebra in X-ray images, we kept all options in our proposed complete framework. The framework ends with four parallel terminal modules that produce four visually different outputs.

The first terminal module localizes corners. Corner localization is done by Bhattacharyya coefficient-based loss function equipped PSRN network discussed in Chapter 5. However, there is a difference between the input patches produced by the extraction process discussed in Sec. 5.3.4 and the input test patches used here. In the prior case, a grid-based multi-resolution multi-orientation process was used to create multiple patches from the vertebrae and also from the intervertebral spaces. Whereas here, only a single patch per vertebrae has been used to keep the terminal modules similar to each other and also, to avoid complications in the grid creation process in case of missing center predictions. The second terminal module detects vertebral boundaries. The improved histogram-based normalization layer assisted PSRN discussed in Chapter 6 is used for boundary detection. The generated probability edge maps are then post-processed using morphological erosion to reduce the thickness. For segmentation the shape-aware SegNet-S of Chapter 6 has been used in the third terminal

module. Finally, in the last module, shape prediction is achieved by the novel shape predictor UNet described in Chapter 7. An additional set of corners are also localized from the predicted shapes using the process described in Sec. 7.5.1. All these patch-level predictions are then transformed back on the original image space and visualized. The complete process with all the terminal modules is summarized in Fig. 8.2.

8.3 Qualitative Evaluation

Qualitative results are shown in Fig. 8.3, 8.4 and 8.5. Four images with different overall intensity variations are shown in Fig. 8.3. It can be seen that the framework can produce good predictions for most of the vertebrae. Fig. 8.3a includes vertebrae with osteophyte (C5) and teardrop fracture (C6) which make the prediction challenging. However, the final results are accurate even with these conditions. Some deviations from the ground truth can be seen for the segmentation and shape prediction results for the affected vertebrae. However, the predicted boundaries were able to capture the variations induced by the clinical conditions. Another point of interest is the contrast between the corner localization results from the two available options. For the PSRN-based corner localization framework, the posterior corners ($\times$) for two consecutive vertebrae cannot be separated. However, for the corners ($\times$) localized from the predicted shapes, all the posterior corners are visible separately. The reason behind this can be attributed to the individual process of the two methods. The PSRN-based method creates a probability distribution for localized corners, and as the corners in the posterior sides are very close, two consecutive distribution merges in the final image and the process detects both corners at the same location. Whereas for the corners localized from the predicted shapes, first, the shapes are predicted and then corners are localized by finding the points with maximum curvature from the predicted shape points. Thus the corners are visible separately from each other, even when their locations are close. Fig. 8.3b shows an almost perfect results for all the outputs. The spine was localized with high accuracy, vertebral centers are almost at the middle of each vertebra, corners have been localized at the correct positions. Boundary detection, segmentation and shape prediction all match the ground truth accurately.

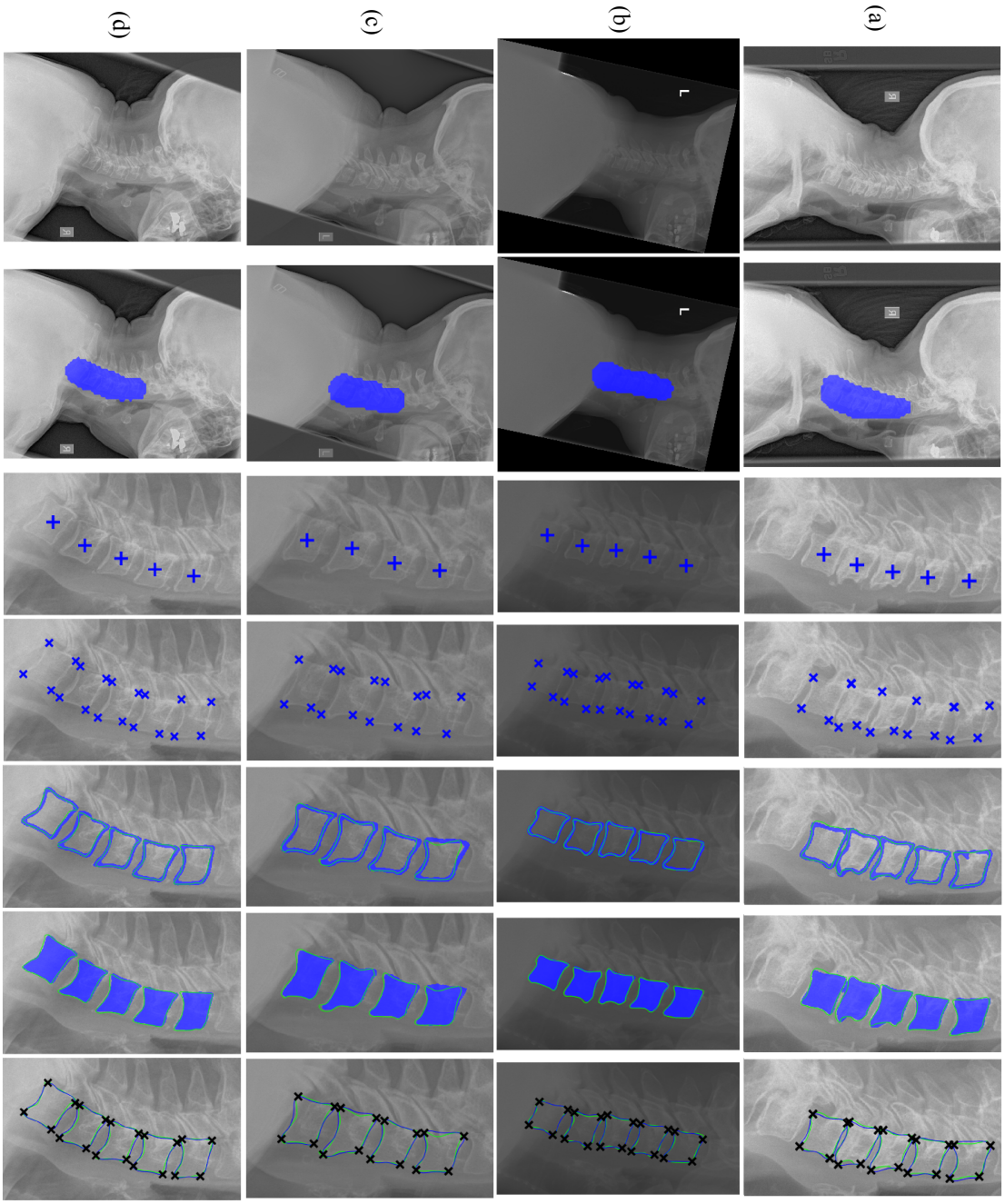


Fig. 8.3 Qualitative results 1. Manually annotated vertebral boundaries are plotted in green.

The subject has a spinal misalignment (retrolisthesis) between the vertebra C5 and C6, which is visible with all the highlighted predictions. A vertebra (C5) with uncommon structure and a probable fracture can be seen in the example shown in Fig. 8.3c. The final results from all four terminal modules have been able to follow the anomaly accurately. However, vertebra C3 of the same example shows disagreement between the ground truth and the results on the posterior side. Fig. 8.3d shows another example of near perfect results from all the modules.

The complete framework produces acceptable results for most of the images in the test dataset. But, the results are not always perfect. In Fig. 8.4, we report some interesting cases from the results. Our medical partners were not able to provide manual annotation of vertebrae C7 for the example in Fig. 8.4a due to lack of contrast in the image near that vertebra. But, the center localization framework have localized a vertebral center for C7 and although the results for corner localization, boundary detection and segmentation are inaccurate, the shape predicted by the LS-UNet-18 can be considered accurate. An opposite example can be found in Fig. 8.4b, where due to lack of contrast our method has failed to produce results for C7 but our medical partners have provided manual annotation. Fig. 8.4c shows a fault in our center localization post-processing process. When more than five centers have been localized, the post-processing described in Sec. 4.3.3, keeps the five most confident centers and ignores the rest. By doing so, for this particular example the algorithm missed C5 and detected T1, the first thoracic vertebra. And as the following terminal modules work on the extracted image patches, there was no scope of correcting the mistake. Another problem related to the center localization process is illustrated in the complicated example of Fig. 8.4d. The algorithm detected two false centers, one on the intervertebral disk between C2 and C3 and another in the vertebral extension of C2. Both of these false centers caused bad results from all the terminal modules.

Examples of images with severe degenerative changes, bone fusion, image artefacts and surgical bone implants are shown in Fig. 8.5. The example in Fig. 8.5a shows sign of severe degeneration in all the vertebrae. The spinal region in the image is also relatively

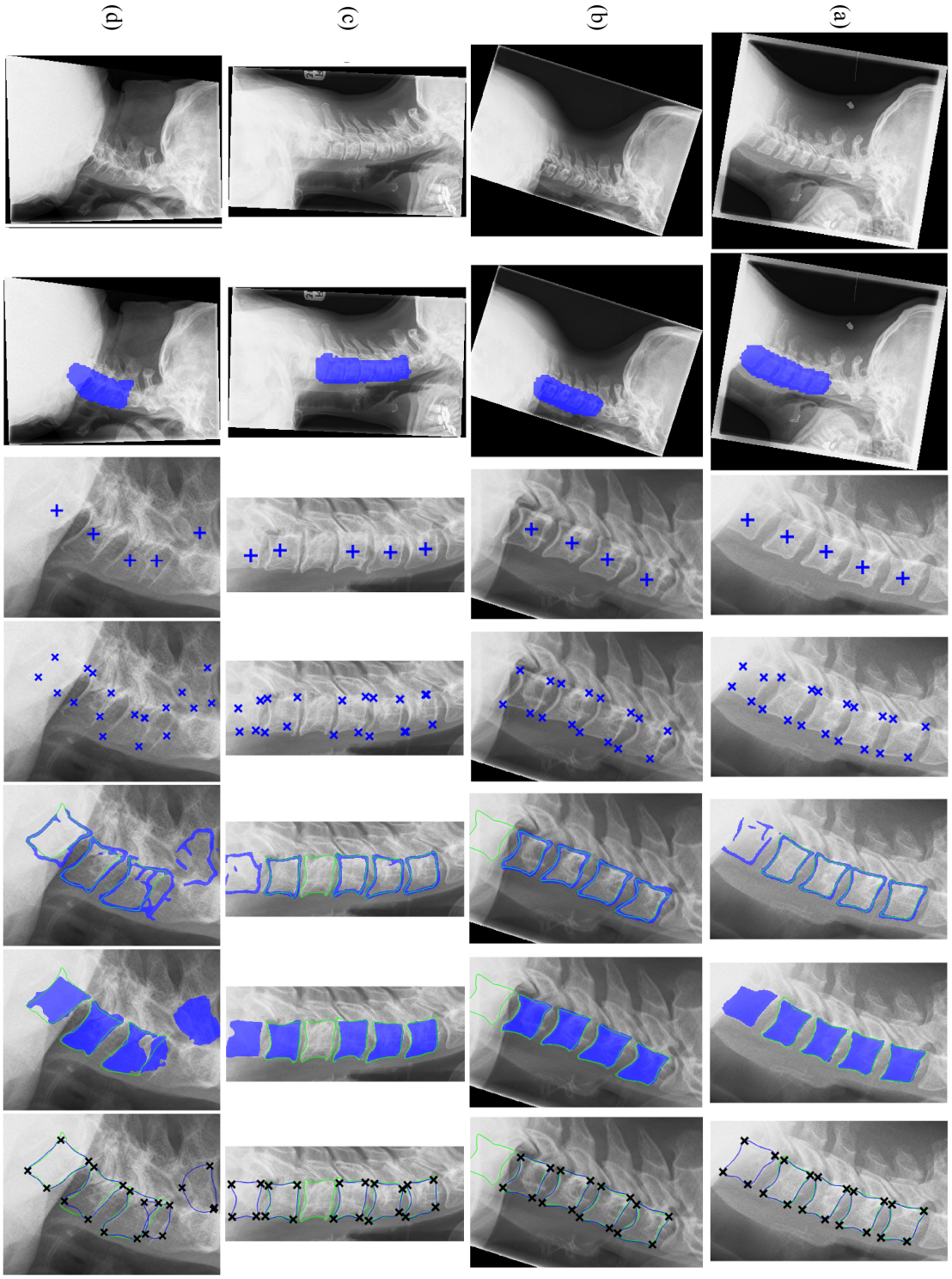


Fig. 8.4 Qualitative results 2. Manually annotated vertebral boundaries are plotted in green.

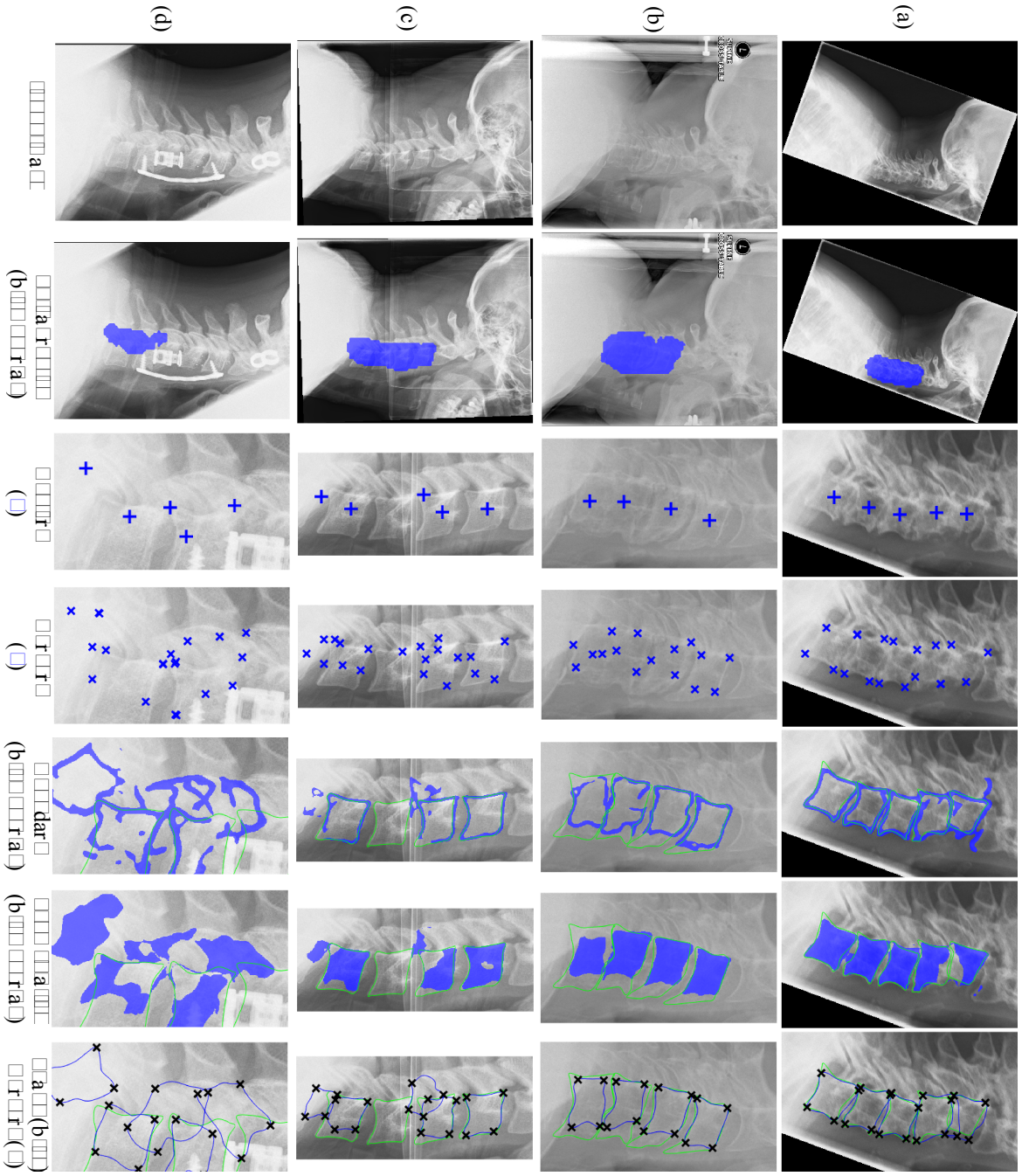


Fig. 8.5 Qualitative results 3. Manually annotated vertebral boundaries are plotted in green.

small. However, our spine localization and the center localization method were able to correctly localize the spine and vertebral centers. The results from terminal modules for the vertebra C3 and C4 are inaccurate because of the severity of the condition. Acceptable results were generated for following vertebrae where degeneration was less. Another example with extreme clinical conditions is reported in Fig. 8.5b. Although the spine localization and center localization results were good, the final vertebral image patch extraction process failed to compensate for the unnaturally long vertebrae. The generated patches with a fixed size of 32 mm were not able to encompass the complete vertebrae. So results from the terminal modules were not accurate. Fig. 8.5c reports an image with strong image artefacts which caused the center localization process to miss the center for C5 and caused two false positives. Because of the false positives, the computation of the orientation vector was affected and so were the extracted patches. The effects are visible in the results for vertebra C4 and C6. The predicted shapes and boundary for C3 were accurate. Finally, a complete failure of the fully automatic framework is reported in Fig. 8.5d. The presence of multiple surgical implants in the image caused the spine localization process to fail, and all the following modules suffered.

The fully automatic framework has also been applied to the images from the NHANES-II dataset. The results are shown in Fig. 8.6 and 8.7. It should be noted that the pixel spacings of these images are unknown. The fully automatic framework uses the pixel spacing in the center localization post-processing and during the vertebral patch extraction steps. For the NHANES-II images, we have chosen a fixed pixel spacing of 0.1 mm per pixel by comparing with images of similar sizes in our dataset. The images were also flipped and rotated accordingly so that the posterior side stays on the left side of each image, similar to the images in our dataset as explained in Sec. 2.3. Fig. 8.6 and 8.7 demonstrate the robustness of our proposed fully automatic framework.

Fig. 8.6 Qualitative results from NHANES-II dataset.

Fig. 8.7 Qualitative results from NHANES-II dataset 2.

8.4 Quantitative Evaluation

In the previous section, we have showcased the performance of our framework with qualitative results for a few cases from our test dataset where we have highlighted specific issues. For an overall evaluation of the performance, we report average quantitative metrics over the whole test dataset in this section.

The spine localization module of the complete framework uses FCN-R network reported in Chapter 3. The localization performance of this network has already been reported in Table 3.3. An average sensitivity of 0.97 and a mean orientation error of 3.12° have been achieved. The fully automatic center localization process has also been discussed in Chapter 4. A true positive rate of 93.10% and false discovery rate of 9.40% have been achieved. Out of our 797 test vertebrae, 742 centers have been correctly detected with an average error of 1.72 mm. The number of false positives was 77.

For the 742 vertebrae which have been correctly detected, the corners localized by the corner localization terminal module have an average error of 2.69 mm. This error is higher than the error reported in Table 5.1 because of three reasons. First, here the corners are localized using only one patch per detected vertebra, where the process described in Sec. 5.3.4 involves extraction of multiple patches from the spinal region. Second, the fixed size patch extraction process sometimes can not extract the whole vertebrae, and thus, the corners are not visible. And third, the post-processing stage involves the use of prior distribution corners which are defined per-vertebra. Because our center localization process does not provide topological information about the identity of the vertebra, a simple and error-prone approach has been utilized. The top center is considered as C3 and the following assigned sequentially without compensating for the probable missing centers. Because of all these issues, the corner localization results are poor making this terminal module the weakest in the complete framework. However, we have another set of corners predicted by the shape prediction terminal module which achieved a much lower average error of 1.74 mm. Unlike the corner localization terminal module, this module only suffers from the issue of fixed

patch size.

Finally, the vertebral body segmentation terminal module has achieved an average Dice similarity coefficient of 0.876 for overall vertebrae segmentation performance on the whole test dataset. The shape predicted by the shape predicted module can also be converted to segmentation masks. The converted masks achieved Dice similarity coefficient of 0.871.

The set of points predicted by the shape prediction module on the original image has also been compared with the manually annotated vertebral curves. This comparison is made with a metric similar to the E_{p2c} error described and used in Chapter 6 and 7. However, there are two differences. First, the error reported in these chapters was reported in pixels as it was computed on the extracted patch space where the pixel spacing information is not representative because of the resize operation. Here, in this chapter, the final predicted shape points are localized on the original image. Thus, the errors are reported in millimeters by using the known image pixel spacing values. Second, the original E_{p2c} errors were computed at the vertebra level. Here, because the prediction methods do not provide information about the identity of the vertebrae, the error is computed at the image level. Manually annotated vertebral boundary curves are converted to a single set of points and compared with the set of points predicted by the shape prediction module. An average error of 1.25 mm has been achieved with a median value of 0.54 mm. The segmentation masks predicted by the segmentation modules can also be converted to a set of points by tracking the boundary pixels of the predictions. These points achieved an average error of 1.24 mm.

The complete framework has been implemented in MATLAB, and the source codes have not been optimized for computational efficiency. On a computer without any GPU, the complete framework takes 16 seconds on average to produce all the results for a single image, about a half of which is taken by the center localization process because of the number of patches involved. Each of the deep networks used in the complete framework requires only

0.1 to 0.25 seconds to produce a single patch-level output. Most of the computational time is spent for the patch extraction and post-processing steps.

8.5 Future Work and Conclusion

The fully automatic image analysis tool described in this chapter was built by cascading individual frameworks in a seamless manner. Because of the sequential nature of the framework, errors in the first phases of framework propagate to the rest of the modules. If the spine localization module fails to localize the spinal region, vertebral centers will be incorrect. Thus, the patch extraction procedure will not be able to extract vertebral image patches and rest of modules will produce wrong predictions subsequently. An example of such failure was illustrated in Fig. 8.5d. When spine region has been localized correctly, there is still a possibility of detection of false vertebral centers. Our center localization algorithm has a high true positive rate of 93.10% but it also produces 77 false positive centers. The patches extracted from these false positive centers cause erroneous results for the terminal modules.

To solve the problem of error propagation, in the future version of the complete framework, new modules can be added which will check the validity or correctness of the previous modules. First, the validity of localized spinal region can be checked based on the confidence of the output from the FCN-R network. This dense classification network predicts probabilities of the foreground and background classes. These probability values might provide insight about the correctness of the spine localization prediction. A second validity check module can be added after the vertebral image patch extraction process (Fig. 8.2:3b). The extracted patches can be sent through a binary classifier which will check if the patch contains a vertebra or not. The random forest model described in Sec. 3.1 can be useful in this regard.

Apart from the addition of further modules, geometric analysis of the predicted center locations can also help to reduce the number of false positive centers. Based on the predicted

centers, random sample consensus (RANSAC) algorithm [139] can be utilized to check which predicted center does not belong to an approximate second-order polynomial curve. This process should be able to detect false center predictions like the examples shown in Fig. 8.4d and 8.5c. The missing centers like in the example shown in Fig. 8.4c can also be found by measuring the center to center distances and adding new centers in the middle where the distance is approximately double.

In this chapter, we have proposed a first-of-its-kind, fully automatic image analysis framework for cervical vertebrae in X-ray images. Although the framework has considerable scopes for improvement, the current version has been able to produce robust results for most of the images in our test dataset. Qualitative results for all the test images can be found at the following link - *https://goo.gl/BVwNe4*.

Chapter 9

Conclusion

9.1 Summary

The cervical spine is one of the most important anatomies of our body. Because of its relative location and its connection to the other organs, more than a half of all spinal injuries occur in the cervical spine. For any cervical spine related injuries, X-ray is commonly the first method of choice for diagnosis because of its quick results, low cost and ease of access. But even with the advancement in imaging technologies, a significant percentage of cervical spine injuries remain unnoticed. About two-third of the patients with missed injuries suffer neurological deterioration [7]. Towards building an automatic injury detection system which can reduce the amount of missing injuries, we have proposed a fully automatic image analysis framework for cervical vertebrae in X-ray images. The complete framework takes a lateral cervical X-ray image as the input and highlights several vertebral features in the output. Although the current framework does not detect vertebral injuries, it can provide the user with different views of the spine with highlighted vertebral features like corners, boundaries and shapes. These supplementary views generated by an automated system have the potential to augment the radiologist's decision and to reduce the number of unnoticed injuries.

The proposed complete framework consists of six specialized modules, each of which solves a particular computer vision problem related to the lateral cervical X-ray image. The

first module of the fully automatic framework localizes the spinal region in the image. The second module takes the localized spinal region and localizes vertebral centers. Based on the localized vertebral centers, vertebral image patches are extracted and fed into four parallel terminal modules. The first terminal module localizes vertebral corners and the second terminal module detects and highlights vertebral boundaries. Next, the segmentation of the vertebral bodies is performed by the third module. Finally, the fourth terminal module predicts vertebral shapes and also localizes corners from the predicted shapes.

9.2 Outcomes

The main research question posed at the beginning of this dissertation was: 'Is it possible to develop a fully automatic image analysis framework for cervical vertebrae in X-ray images?'. We approached the solution by dividing our goal into several objectives in Sec. 1.2, each of which can now be addressed.

1. Spine localization: The first objective was to localize the spinal region in the X-ray image. To address this objective, we have proposed and compared two spine localization algorithms in Chapter 3. The first algorithm uses random classification forest and localizes the spine using a coarse-to-fine approach. It produces a bounding parallelogram around the spinal region in the image. However, the bounding parallelogram was not able to capture the orientation and the flexibility of the spine. To localize the spinal region with arbitrary shapes, we have proposed a deep learning-based algorithm. A novel region-aware loss term is proposed which takes into account the connected nature of the predicted region in training. The proposed loss was able to improve the performance of all three dense classification networks which were compared in Sec. 3.2. An average sensitivity and specificity of 0.97 have been achieved in localizing the cervical spine.

2. Center localization: The purpose of this objective was to predict vertebral centers in the localized spinal region. We have proposed a novel probabilistic spatial regressor network (PSRN) to fulfil this objective. This is reported in Chapter 4. As vertebral

centers are not attached directly to any visible image landmark, the location of the center varies based on human interpretation. This motivated us to convert the manually clicked vertebral centers into probabilistic distributions. A dense classification network, UNet, is augmented with a novel loss function to generate spatially distributed probability distribution. The center localization algorithm has been able to detect 94.73% vertebral centers with an average error of 1.80 mm which was lower than the error computed from centers localized by an expert radiologist.

3. Corner localization: The third objective was to predict vertebral corners with help of the already localized vertebral centers. This objective was addressed in Chapter 5. Three innovative methods were proposed for localization of vertebral corners. The first method uses classical corner (Harris) and edge (Canny, Sobel, Prewitt, Roberts, LoG) detectors and combines the results with a prior distribution of corners in a naive Bayes formulation to localize vertebral corners. The second method involves Hough forest which localizes vertebral corners using a patch-based approach. Finally, a deep learning-based corner localization framework has been proposed where we improve upon the proposed probabilistic spatial regressor network (PSRN) of Chapter 4. A new spatial normalization layer and a novel loss function based on Bhattacharyya coefficient was proposed. We have achieved a median error of less than a millimeter for corner localization.

4. Vertebral boundary detection: After solving the localization issues, our next objective was to detect vertebral boundaries. A dense classification and a probabilistic approach were taken to address this issue. A novel weighted loss function is proposed to solve the data imbalance problem faced by the dense classification networks. For the probabilistic approach, the PSRN network of Chapter 5 has been improved further with a histogram-based normalization layer to solve residual background problem discovered in the previous chapter. Both of these amendments has proven to achieve a significant improvement in boundary detection performance.

5. Vertebral body segmentation: The objective of vertebra segmentation has also been addressed in Chapter 6. Vertebra segmentation was performed using a dense classification network. The standard dense classification network is trained using a pixel-wise loss function which is not capable of capturing shape information explicitly. To address the issue, a novel shape-aware loss term has been proposed which resulted in significant improvement in segmentation performance. We have shown that the shape-awareness helps the network to segment vertebrae with clinical conditions with higher accuracy. Overall, a high Dice similarity coefficient of 0.944 is achieved by the shape-aware segmentation network.

6. Vertebral shape prediction: A novel convolutional neural network was proposed to address the objective of vertebral shape prediction. The shape predictor network builds upon the success of the UNet architecture used in this dissertation for dense classification and probabilistic spatial regression tasks. The network has been modified to produce signed distance functions defined over the same pixel space as the input to the network. The shape is represented by the zero-level set of the predicted signed distance function. This novel shape predictor network has been reported in Chapter 7 which achieved an average shape error of less than a pixel.

9.2.1 Fully Automatic Framework

Finally, after addressing each of the objectives, we turn to our original quest of building a fully automatic framework which has been described in Chapter 8. The best-performing methods from each of the chapters were selected and joined together in a seamless manner. The complete framework takes a lateral cervical X-ray image and highlights several vertebral features at the end without any user input. The fully automatic framework has achieved a Dice similarity coefficient 0.876.

9.3 Future Work

In this dissertation, we worked on a dataset of X-ray images collected from real-life medical emergency rooms which posed a set of challenging and practical problems to overcome. We have been able to solve a set of computer vision related problems, but there are several limitations in our proposed solutions. In the next subsection, we will list some limitations and likely ways to address them. After that, we will list a number of things we have already explored without any fruitful outcomes so far. Finally, we will end this section with some directions of research which might be useful for other researchers interested in moving forward with the work presented in this dissertation.

9.3.1 Limitations

One of the limitation of the work presented here is our assumption that the X-ray image is upright and the anterior side in the left of the image. Although our spine localization algorithm is rotation and view invariant, the rest of the algorithms are not invariant to these assumptions. There are two possible ways to solve this issue. First, the models can be retrained with new data which will include these variations. A second approach can be to include a new module in the framework which will determine the orientation of the scanned subject in the image and will rotate and/or flip the image accordingly so that the following modules can work without the need of retraining. Similar modules for determination of anatomical pose and view can be found in the literature for classification of cardiac views in echocardiogram [140, 141]. Instead of adding a new module, we can also improve our spine localization algorithm to determine the orientation of the subject. This can be achieved as a post-processing step and/or the network can be augmented to produce appropriate outputs.

Another limitation of the current framework is that it cannot provide any information about the identity the predicted vertebral centers. A further geometrical analysis of the predicted centers can solve this issue to some extent with some assumptions like the image is upright, and the top vertebra is always C3. But these assumptions are not always valid. To

solve the problem completely, a new vertebra identification framework can be formulated and trained to identify and localize cervical vertebrae simultaneously.

The fixed size of the test patch extraction process is also a limitation in the proposed framework. According to the analysis of our training data in Fig. 8.1, about 10% of the vertebrae are larger than chosen test patch size. We need to improve the process to determine the patch size dynamically. An immediate solution could be to extract patches with multiple sizes. But that would require reconciliation between multiple predictions for the same vertebra.

The current version of the corner localization framework requires vertebral centers to be known. It uses the centers to select correct prior distribution which is required in the post-processing steps. Future work is needed to remove this dependency. Once removed, the vertebral corners can be localized directly from the localized spinal area. These corners can also be used to determine the size of the test patches dynamically to address the issue of the patch extraction process.

The current version of the boundary detection framework suffers from the problem of boundary thickness, common to the UNet like encoder-decoder architecture-based methods. Recent work by Wang et al. have addressed a similar problem by replacing the deconvolution layers with sub-pixel convolution [108, 115]. The use of sub-pixel convolution can be explored to improve the boundary detection performance.

The segmentation module and the shape prediction produce similar results with different representations. Both modules share a common deep architecture but trained with different loss functions. While each loss function has some advantages over the other, future work can be performed to combine the benefits of both loss functions in the same network.

Finally, the all the machine learning-based models proposed in this dissertation suffer when an abnormal vertebra is presented at the test time. This happens because the models are trained using a dataset mostly full of healthy vertebrae. As our primary focus was to create a prototype of the fully automatic image analysis framework, less attention was provided to recognize the abnormal situations. Moving forward, the proposed framework can work as a starting point, and more emphasis could be exerted on detecting the complex cases accurately where fewer examples are available to train the models.

9.3.2 Unsuccessful Attempts

There are several things that we have tried but ended up with unfruitful results. Below we present a list of a few of those unsuccessful attempts which seemed promising at the time.

- U-DeConvNet: A novel network architecture was developed by combining the UNet and DeConvNet architecture. The proposed network had an encoder-decoder architecture which shared information using both techniques from UNet and DeConvNet, i.e., concatenation data matrices and unpooling using switch variables. The intuition was that sharing more information from the contracting encoder path would result in better performance. However, the results from the proposed network were better than DeConvNet but worse than UNet. The reason was the use of unpooling layers. These layers produce sparse outputs which affect the overall performance.

- Autoencoder-based shape statistics compiler: In Sec. 7.4, we have described a convolutional neural network (CNN) for predicting the level set parameters directly from a vertebral image patch. In our initial set of experiments, we have also tried the same for active shape model (ASM)-based parameters. But the results were inaccurate. To build upon the idea, an attempt was made to replace the principal component analysis (PCA)-based shape statistics compiler with the use of deep autoencoder network. The ASM capture the shape statistics through PCA by computing mean shape, eigenvalues, and eigenvectors. Noisy information and variations from minor eigenvectors are discarded, and a full shape can be represented by a small number of parameters. The idea

was to achieve the same using autoencoders, given a detailed shape, an autoencoder will be trained to reproduce the same shape at the decoder output. After training, the encoder can be used to represent the shapes in a different space with a fewer number of parameters. These shapes could have been used as the ground truth for training a vertebral image patch to shape parameter network. The autoencoder was able to represent the shape with less number of parameters. But unfortunately, the decoded shapes were not close to the actual shape like ASM-based methods to proceed to the next step.

- Vertebra identification network: In the previous subsection, we have mentioned the need of a vertebra identification framework. This is an area of research that have not been included in the dissertation. However, we have tried to train a fully convolutional network to recognize vertebrae by predicting each vertebral center in a different channel. The method was partially successful, but future work is needed to make the results presentable. The network was parameter heavy as we had to increase the input resolution to 384×384. The network takes about four days to train with the existing facilities. Thus performing a complete set of experimentation with the network to improve the results was out of the scope of this dissertation.

9.3.3 Directions for Future Research

- Injury detection: In this dissertation, we have solved the computer vision part of a fully automatic injury detection system. The current framework highlights several vertebral features like centers, corners, boundaries, and shapes. The obvious next step is to utilize this information to detect anomalies in the spinal area. While injuries like vertebral misalignment, osteophytes, vertebral fractures (wedge and crushed) and reduction of intervertebral disk can be computed easily from the detected corners, boundaries and shapes, detection of other anomalies like osteoporosis, bone loss, and teardrop fracture, etc. needs further work.

- End-to-end framework: The work in this dissertation has shown that a fully automatic framework can be developed for X-ray images. However, the final framework is a combination of different modules, each of which solves a particular task. A fully end-to-end trainable framework can be developed in the future. This can be approached in various ways. One idea could be to use architectures like R-CNN [142, 143] to generate region proposal around the vertebrae from a full resolution image and then perform segmentation and/or shape prediction using the techniques proposed in Chapter **??** and 7. The networks can be formulated and trained in an end-to-end fashion. Another novel idea could be to formulate a neural network which will be a combination of a recurrent neural network module and a fully convolutional neural network module. The network will start with a low-resolution image and using recurrency, it will learn to zoom and crop parts of the image and provide a crisp vertebrae segmentation result at the end.

- Extension to other views of the cervical spine: As mentioned earlier, the X-ray images of the cervical spine is usually taken with three standard views: lateral, anterior-posterior (AP), odontoid process (Peg). These views have been shown in Fig.2.2. The methods proposed in this dissertation is focused only on the lateral view of the cervical spine. However, this dissertation presents a systematic way of building a fully automatic image analysis framework by addressing fundamental vision related problems like image landmark localization, boundary detection and segmentation. The same set of problems are also relevant to other views of the cervical spine. Thus, in future, if enough input images and corresponding manual annotations of the vertebrae are available, the models presented in this dissertation can be adapted and reformulated to work with the AP and Peg views.

- Extension to 3D images: The work presented in this dissertation are designed for 2D radiographic images. The spine can also be scanned using computer tomography (CT) and magnetic resonance imaging (MRI) techniques. Although the image acquisition times are longer than 2D radiographs, the CT and MRI techniques produce high quality

3D images of the spine which are more diagnostic for clinical evaluation. Most of the state-of-the-art methods proposed here use deep neural networks where loss functions are defined in a pixel-wise manner. Even the loss function for the shape predictor network computes an element-wise Euclidean distance between the shape parameters. Thus the formulations are easily generalizable from 2D to 3D. Theoretically, it should requires two major changes. First, the final loss of the proposed deep networks is usually computed using a summation operation over the pixel space (Ω_p) which should be replaced by the voxel space for the 3D data. And, second, the 2D fully convolutional neural network architectures should be replaced by appropriate 3D architectures. In this dissertation, we have extensively used the UNet-like architecture for solving different problems. A 3D version of the UNet architecture, VNet, has already been proposed in [144]. In future, the algorithms proposed in this dissertation can be updated using the VNet-like architectures to build fully automatic image analysis frameworks for 3D CT and MRI scans.

- Probabilistic regression: Regression using neural networks are mostly deterministic. In this dissertation, we have proposed a novel method for probabilistic regression using neural networks. By choosing an appropriate standard deviation, any regression target can be converted to a probability distribution defined over the output space. Then the proposed fully convolutional neural network can be adapted to learn the mapping between the input data and the output probability distribution. Theoretically, the method has the capacity to solve any regression problem. A systematic study of the proposed method for different regression problems, however, was outside the scope of this dissertation. In future, the applicability and generalizability of this proposed method can be evaluated on different regression datasets.

9.4 Personal Experience

My PhD research began with a set of real-life emergency room X-ray images and a semi-automatic framework that uses active shape model (ASM) to achieve vertebrae segmentation.

The framework was capable of producing acceptable results only for a handful of vertebrae. Since then it has been quite a journey through thick and thin, to the fully automatic framework that is capable of producing good results for the majority of the images in a test dataset full of complicated cases.

Although at the end of my PhD research I have been able to produce something that I am proud of, the journey was not smooth. Especially in the first year and a half, a good amount research was performed which did not produce expected results. The corner localization methods described in Sec. 5.1 and 5.2 were initially formulated to initialize the mean shape of the ASM-based initial framework with high accuracy. However, the performance of the corner localization was not accurate enough to improve the performance of the initial framework. Another attempt was made to improve the performance of active shape model-based framework by incorporating random forest models in the ASM search method. This work, ASM-RF, improved the segmentation performance by a small margin from other state-of-the-art ASM-based methods and was published in [69]. However, the method failed to outperform the simplest gradient-based ASM method when applied to the test dataset used in this dissertation at the resolution of 64×64 (see Table 6.3), and thus, excluded from this dissertation.

The real breakthrough in my research came after starting to incorporate deep learning into the solutions. Deep learning was making waves in the computer vision field since 2012. But lack of enough data samples to work with prevented us from using it in the beginning of my PhD. Slowly but surely, medical image communities also found ways to use deep learning-based solutions using data augmentation techniques. The UNet architecture was proposed in 2015 with great success in medical image segmentation. The number of images in our dataset also increased with time. Based on these developments, we first used the UNet and other dense classification networks to localize spinal region in the X-ray image. After getting robust localization results, we moved forward with deep learning solve other problems. And with time, we have been able to build a prototype fully automatic framework

that produces acceptable results for the majority of the images in our challenging test dataset.

Coming from an engineering background, I was always motivated to find solutions to any problem with existing tools and technologies without carefully thinking about original contributions, theoretical novelty, and innovation. This trend is noticeable in my initial research, especially in Sec. 3.1, 5.1, and 5.2, where I have used existing methods to solve different problems. The amount of theoretical novelty in these proposed solutions was limited, but they provided a practical way to solve the problems. However, with time, I have learned that original contributions, theoretical novelty, and innovation are necessary for a dissertation worthy of a PhD award. Thus, I have tried my best to include key contributions and innovation in the rest of work done for my PhD. The region-aware loss term for the spine localization framework and shape-aware loss term included in the segmentation framework are two examples of this effort. The novel probabilistic spatial regressor network proposed in Chapter 4 and improved in Chapter 5 and 6 is one of the two major contributions of this dissertation. I believe this novel spatial regressor network can be utilized beyond medical imaging to solve image landmark localization problems in a broader aspect. The other major contribution is the novel shape predictor network where we have described a practical way to predict shapes using a encoder-decoder architecture like UNet.

Going back to my original motivation, the personal aim for my PhD was to replace the initial framework with a robust fully automatic framework capable of detecting vertebral injuries. Unfortunately, I have not been able to work on the injury detection part within the limited time-line of my PhD research. Changing the title of my dissertation from ‘fully automatic injury detection system’ to ‘fully automatic image analysis framework’ was difficult for me. However, I believe given the outputs that the proposed fully automatic framework produces, injury detection system can be implemented with ease. Looking back, I am more than happy with what I have achieved during the last three years. I have learned a lot of lessons about research, work and the life itself. I have had an amazing experience, and

I hope these lessons will help me to grow as a researcher and most importantly as a good, responsible human-being on this earth.

References

[1] Vertebra column. http://voer.edu.vn/c/types-of-skeletal-systems/d4223f39/653a5049#fig-ch38_01_07. Accessed: 2016-02-07.

[2] Philipp Mitteroecker and Philipp Gunz. Advances in geometric morphometrics. *Evolutionary Biology*, 36(2):235–247, 2009.

[3] Anoushka Singh, Lindsay Tetreault, Suhkvinder Kalsi-Ryan, Aria Nouri, and Michael G Fehlings. Global prevalence and incidence of traumatic spinal cord injury. *Clinical epidemiology*, 6:309, 2014.

[4] Standard x-ray veiws for cervical spine. https://www.radiologymasterclass.co.uk/tutorials/musculoskeletal/x-ray_trauma_spinal/x-ray_c-spine_normal. Accessed: 2017-17-10.

[5] Patrick Platzer, Nicole Hauswirth, Manuela Jaindl, Sheila Chatwani, Vilmos Vecsei, and Christian Gaebler. Delayed or missed diagnosis of cervical spine injuries. *Journal of Trauma and Acute Care Surgery*, 61(1):150–155, 2006.

[6] James W Davis, David L Phreaner, David B Hoyt, and Robert C Mackersie. The etiology of missed cervical spine injuries. *Journal of Trauma and Acute Care Surgery*, 34(3):342–346, 1993.

[7] CGT Morris and E McCoy. Clearing the cervical spine in unconscious polytrauma victims, balancing risks and effective screening. *Anaesthesia*, 59(5):464–482, 2004.

[8] Per Skaane, Ashwini Kshirsagar, Sandra Stapleton, Kari Young, and Ronald A Castellino. Effect of computer-aided detection on independent double reading of paired screen-film and full-field digital screening mammograms. *American journal of roentgenology*, 188(2):377–384, 2007.

[9] Fiona J Gilbert, Susan M Astley, Maureen GC Gillan, Olorunsola F Agbaje, Matthew G Wallis, Jonathan James, Caroline RM Boggis, and Stephen W Duffy. Single reading with computer-aided detection for screening mammography. *New England Journal of Medicine*, 359(16):1675–1684, 2008.

[10] Sheng Chen and Kenji Suzuki. Computerized detection of lung nodules by means of "virtual dual-energy" radiography. *IEEE Transactions on Biomedical Engineering*, 60(2):369–378, 2013.

[11] Xujiong Ye, Xinyu Lin, Jamshid Dehmeshki, Greg Slabaugh, and Gareth Beddoe. Shape-based computer-aided detection of lung nodules in thoracic CT images. *IEEE Transactions on Biomedical Engineering*, 56(7):1810–1820, 2009.

[12] Hidetaka Arimura, Shigehiko Katsuragawa, Kenji Suzuki, Feng Li, Junji Shiraishi, Shusuke Sone, and Kunio Doi. Computerized scheme for automated detection of lung nodules in low-dose computed tomography images for lung cancer screening. *Academic Radiology*, 11(6):617–629, 2004.

[13] Greg Slabaugh, Xiaoyun Yang, Xujiong Ye, Richard Boyes, and Gareth Beddoe. A robust and fast system for CTC computer-aided detection of colorectal lesions. *Algorithms*, 3(1):21–43, 2010.

[14] Kenji Suzuki, Don C Rockey, and Abraham H Dachman. Ct colonography: Advanced computer-aided detection scheme utilizing mtanns for detection of "missed" polyps in a multicenter clinical trial. *Medical physics*, 37(1):12–21, 2010.

[15] Ethan J Halpern and David J Halpern. Diagnosis of coronary stenosis with ct angiography: comparison of automated computer diagnosis with expert readings. *Academic radiology*, 18(3):324–333, 2011.

[16] Ki-Woon Kang, Hyuk-Jae Chang, Hackjoon Shim, Young-Jin Kim, Byoung-Wook Choi, Woo-In Yang, Jee-Young Shim, Jongwon Ha, and Namsik Chung. Feasibility of an automatic computer-assisted algorithm for the detection of significant coronary artery disease in patients presenting with acute chest pain. *European journal of radiology*, 81(4):e640–e646, 2012.

[17] El-Sayed A El-Dahshan, Heba M Mohsen, Kenneth Revett, and Abdel-Badeeh M Salem. Computer-aided diagnosis of human brain tumor through mri: A survey and a new algorithm. *Expert systems with Applications*, 41(11):5526–5545, 2014.

[18] Xingxing Zhou, Shuihua Wang, Wei Xu, Genlin Ji, Preetha Phillips, Ping Sun, and Yudong Zhang. Detection of pathological brain in mri scanning based on wavelet-entropy and naive bayes classifier. In *IWBBIO (1)*, pages 201–209, 2015.

[19] Yudong Zhang, Shuihua Wang, and Zhengchao Dong. Classification of alzheimer disease based on structural magnetic resonance imaging by kernel support vector machine decision tree. *Progress In Electromagnetics Research*, 144:171–184, 2014.

[20] Yudong Zhang, Zhengchao Dong, Preetha Phillips, Shuihua Wang, Genlin Ji, Jiquan Yang, and Ti-Fei Yuan. Detection of subjects and brain regions related to alzheimer's disease using 3d mri scans based on eigenbrain and machine learning. *Frontiers in Computational Neuroscience*, 9, 2015.

[21] Feng Li, Hidetaka Arimura, Kenji Suzuki, Junji Shiraishi, Qiang Li, Hiroyuki Abe, Roger Engelmann, Shusuke Sone, Heber MacMahon, and Kunio Doi. Computer-aided detection of peripheral lung cancers missed at ct: Roc analyses without and with localization. *Radiology*, 237(2):684–690, 2005.

[22] Feng Li, Masahito Aoyama, Junji Shiraishi, Hiroyuki Abe, Qiang Li, Kenji Suzuki, Roger Engelmann, Shusuke Sone, Heber MacMahon, and Kunio Doi. Radiologists' performance for differentiating benign from malignant lung nodules on high-resolution ct using computer-estimated likelihood of malignancy. *American Journal of Roentgenology*, 183(5):1209–1215, 2004.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[24] Mohammed Benjelloun, Saïd Mahmoudi, and Fabian Lecron. A framework of vertebra segmentation using the active shape model-based approach. *Journal of Biomedical Imaging*, 2011:9, 2011.

[25] Sidi Ahmed Mahmoudi, Fabian Lecron, Pierre Manneback, Mohammed Benjelloun, and Säid Mahmoudi. GPU-based segmentation of cervical vertebra in X-ray images. In *Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

[26] Dr. Arun Pal Singh. Range of motion of cervical spine. http://boneandspine.com/range-motion-cervical-spine/. Accessed: 2017-11-09.

[27] Joel A Torretti and Dilip K Sengupta. Cervical spine trauma. *Indian journal of orthopaedics*, 41(4):255, 2007.

[28] Harry K Genant, Chun Y Wu, Cornelis van Kuijk, and Michael C Nevitt. Vertebral fracture assessment using a semiquantitative technique. *Journal of bone and mineral research*, 8(9):1137–1148, 1993.

[29] John H Woodring and Charles Lee. Limitations of cervical radiography in the evaluation of acute cervical trauma. *Journal of Trauma and Acute Care Surgery*, 34(1):32–39, 1993.

[30] Abraham Tezmol, Hamed Sari-Sarraf, Sunanda Mitra, Rodney Long, and Arunkumar Gururajan. Customized Hough transform for robust segmentation of cervical vertebrae from X-ray images. In *Image Analysis and Interpretation, 2002. Proceedings. Fifth IEEE Southwest Symposium on*, pages 224–228. IEEE, 2002.

[31] Gilberto Zamora, Hamed Sari-Sarraf, and L Rodney Long. Hierarchical segmentation of vertebrae from X-ray images. In *Medical Imaging 2003*, pages 631–642. International Society for Optics and Photonics, 2003.

[32] Marleen De Bruijne and Mads Nielsen. Image segmentation by shape particle filtering. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 722–725. IEEE, 2004.

[33] Pavan Chamarthy, R Joe Stanley, Gregory Cizek, Rodney Long, Sameer Antani, and George Thoma. Image analysis techniques for characterizing disc space narrowing in cervical vertebrae interfaces. *Computerized Medical Imaging and Graphics*, 28(1):39–50, 2004.

[34] Mohammed Benjelloun, Horacio Tellez, and Saïd Mahmoudi. Vertebra edge detection using polar signature. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 476–479. IEEE, 2006.

[35] Mohammed Benjelloun, H Téllez, and S Mahmoudi. Template matching method for vertebra region selection. In *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, volume 1, pages 1119–1124. IEEE, 2006.

[36] Mustapha Aouache, Aini Hussain, Salina Abdul Samad, AH Hamzaini, and AK Ariffin. Active shape modeling of medical images for vertebral fracture computer assisted assessment system. In *Research and Development, 2007. SCOReD 2007. 5th Student Conference on*, pages 1–6. IEEE, 2007.

[37] Sergio Casciaro and Laurent Massoptier. Automatic vertebral morphometry assessment. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5571–5574. IEEE, 2007.

[38] PA Bromiley, JE Adams, and TF Cootes. Localisation of vertebrae on DXA images using constrained local models with random forest regression voting. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 159–171. Springer, 2015.

[39] Xiao Dong and Guoyan Zheng. Automated vertebra identification from X-ray images. In *Image Analysis and Recognition*, pages 1–9. Springer, 2010.

[40] Xi Xu, Hong-Wei Hao, Xu-Cheng Yin, Ning Liu, and Shawkat Hasan Shafin. Automatic segmentation of cervical vertebrae in X-ray images. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

[41] Mohamed Amine Larhmam, Shadi Mahmoudi, and Mohammed Benjelloun. Semi-automatic detection of cervical vertebrae in X-ray images using generalized hough transform. In *Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on*, pages 396–401. IEEE, 2012.

[42] Mohamed Amine Larhmam, Mohammed Benjelloun, and Saïd Mahmoudi. Vertebra identification using template matching modelmp and K-means clustering. *International Journal of Computer Assisted Radiology and Surgery*, 9(2):177–187, 2014.

[43] Paul P Smyth, Christopher J Taylor, and Judith E Adams. Vertebral shape: Automatic measurement with active shape models. *Radiology*, 211(2):571–578, 1999.

[44] Martin G Roberts, Timothy F Cootes, and Judith E Adams. Vertebral shape: automatic measurement with dynamically sequenced active appearance models. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 733–740. Springer, 2005.

[45] MG Roberts, TF Cootes, and JE Adams. Automatic segmentation of lumbar vertebrae on digitised radiographs using linked active appearance models. In *Proc. Medical Image Understanding and Analysis*, volume 2, pages 120–124, 2006.

[46] Martin Roberts, Timothy F Cootes, and Judith E Adams. Vertebral morphometry: semiautomatic determination of detailed shape from dual-energy X-ray absorptiometry images using active appearance models. *Investigative Radiology*, 41(12):849–859, 2006.

[47] Martin G Roberts, Tim F Cootes, Elisa Pacheco, Teik Oh, and Judith E Adams. Segmentation of lumbar vertebrae using part-based graphs and active appearance models. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 1017–1024. Springer, 2009.

[48] MG Roberts, EMB Pacheco, R Mohankumar, TF Cootes, and JE Adams. Detection of vertebral fractures in DXA VFA images using statistical models of appearance and a semi-automatic segmentation. *Osteoporosis International*, 21(12):2037–2046, 2010.

[49] Martin G Roberts, Timothy F Cootes, and Judith E Adams. Automatic location of vertebrae on DXA images using random forest regression. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 361–368. Springer, 2012.

[50] Tobias Klinder, Jörn Ostermann, Matthias Ehm, Astrid Franz, Reinhard Kneser, and Cristian Lorenz. Automated model-based vertebra detection, identification, and segmentation in CT images. *Medical Image Analysis*, 13(3):471–482, 2009.

[51] Ben Glocker, Johannes Feulner, Antonio Criminisi, David R Haynor, and Ender Konukoglu. Automatic localization and identification of vertebrae in arbitrary field-of-view CT scans. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 590–598. Springer, 2012.

[52] Jianhua Yao and Shuo Li. Report of vertebra segmentation challenge in 2014 miccai workshop on computational spine imaging. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 247–259. Springer, 2015.

[53] Bulat Ibragimov, Robert Korez, Boštjan Likar, Franjo Pernuš, and Tomaž Vrtovec. Interpolation-based detection of lumbar vertebrae in CT spine images. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 73–84. Springer, 2015.

[54] Robert Korez, Bulat Ibragimov, Bostjan Likar, Franjo Pernus, and Tomaz Vrtovec. A framework for automated spine and vertebrae interpolation-based detection and model-based segmentation. *Medical Imaging, IEEE Transactions on*, 34(8):1649–1662, 2015.

[55] Robert Korez, Bulat Ibragimov, Boštjan Likar, Franjo Pernuš, and Tomaž Vrtovec. An improved shape-constrained deformable model for segmentation of vertebrae from CT lumbar spine images. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 85–94. Springer, 2015.

[56] Robert Korez, Bulat Ibragimov, Boštjan Likar, Franjo Pernuš, and Tomaž Vrtovec. Interpolation-based shape-constrained deformable model approach for segmentation of vertebrae from CT spine images. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 235–240. Springer, 2015.

[57] Timothy F Cootes. Fully automatic localisation of vertebrae in ct images using random forest regression voting. In *Computational Methods and Clinical Applications for Spine Imaging: 4th International Workshop and Challenge, CSI 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Revised Selected Papers*, volume 10182, page 51. Springer, 2017.

[58] Mohamed Amine Larhmam, Sidi Ahmed Mahmoudi, Mohammed Benjelloun, Saïd Mahmoudi, and Pierre Manneback. A portable multi-CPU/multi-GPU based vertebra localization in sagittal MR images. In *Image Analysis and Recognition*, pages 209–218. Springer, 2014.

[59] Marleen de Bruijne, Michael T Lund, László B Tankó, Paola C Pettersen, and Mads Nielsen. Quantitative vertebral morphometry using neighbor-conditional shape models. *Medical Image Analysis*, 11(5):503–512, 2007.

[60] NHANES-II Dataset. https://ceb.nlm.nih.gov/proj/ftp/ftp.php. Accessed: 2017-02-19.

[61] Jianhua Yao, Joseph E Burns, Daniel Forsberg, Alexander Seitel, Abtin Rasoulian, Purang Abolmaesumi, Kerstin Hammernik, Martin Urschler, Bulat Ibragimov, Robert Korez, et al. A multi-center milestone study of clinical vertebral CT segmentation. *Computerized Medical Imaging and Graphics*, 2016.

[62] SpineWeb. http://spineweb.digitalimaginggroup.ca/spineweb/index.php?n=Main. Datasets. Accessed: 2016-02-03.

[63] Shahin Ebrahimi, Elsa Angelini, Laurent Gajny, and Wafa Skalli. Lumbar spine posterior corner detection in x-rays using haar-based features. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 180–183. IEEE, 2016.

[64] Anum Mehmood, M. Usman Akram, and Anam Tariq. Vertebra localization and centroid detection from cervical radiographs. *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, pages 287–292, 2017.

[65] Dong Yang, Tao Xiong, Daguang Xu, Qiangui Huang, David Liu, S Kevin Zhou, Zhoubing Xu, JinHyeong Park, Mingqing Chen, Trac D Tran, et al. Automatic vertebra labeling in large-scale 3d ct using deep image-to-image network with message passing and sparsity regularization. In *International Conference on Information Processing in Medical Imaging*, pages 633–644. Springer, 2017.

[66] Computer-aided detection of cervical spine injuries: A feasibility project. http://gtr.rcuk.ac.uk/projects?ref=EP%2FK037641%2F1. Accessed: 2017-10-20.

[67] S M Masudur Rahman Al-Arif, Mohammad Asad, Karen Knapp, Micheal Gundry, and Greg Slabaugh. Hough forest-based corner detection for cervical spine radiographs. In *Medical Image Understanding and Analysis (MIUA), Proceedings of the 19th Conference on*, pages 183–188, 2015.

[68] S M Masudur Rahman Al-Arif, Mohammad Asad, Karen Knapp, Micheal Gundry, and Greg Slabaugh. Cervical vertebral corner detection using Haar-like features and modified Hough forest. In *Image Processing Theory, Tools and Applications (IPTA), 2015 5th International Conference on*. IEEE, 2015.

[69] S M Masudur Rahman Al-Arif, Michael Gundry, Karen Knapp, and Greg Slabaugh. Improving an active shape model with random classification forest for segmentation of cervical vertebrae. In *Computational Methods and Clinical Applications for Spine Imaging: 4th International Workshop and Challenge, CSI 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Revised Selected Papers*, volume 10182, page 3. Springer, 2017.

[70] S M Masudur Rahman Al-Arif, Michael Gundry, Karen Knapp, and Greg Slabaugh. Global localization and orientation of the cervical spine in x-ray images. In *Computational Methods and Clinical Applications for Spine Imaging: 4th International Workshop and Challenge, CSI 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Revised Selected Papers*, volume 10182, page 64. Springer, 2017.

[71] S M Masudur Rahman Al-Arif, Muhammad Asad, Michael Gundry, Karen Knapp, and Greg Slabaugh. Patch-based corner detection for cervical vertebrae in x-ray images. *Signal Processing: Image Communication*, 59(1):27–36, November 2017.

[72] Edwin Catmull and Raphael Rom. A class of local interpolating splines. *Computer Aided Geometric Design*, 74:317–326, 1974.

[73] David G Kendall. A survey of the statistical theory of shape. *Statistical Science*, pages 87–99, 1989.

[74] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[75] Ben Glocker, Darko Zikic, Ender Konukoglu, David R Haynor, and Antonio Criminisi. Vertebrae localization in pathological spine CT via dense classification from sparse annotations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 262–270. Springer, 2013.

[76] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[77] Piotr Dollár and C. Lawrence Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[78] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *PAMI*, 2015.

[79] Piotr Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). https://github.com/pdollar/toolbox.

[80] Zdravko I Botev, Joseph F Grotowski, Dirk P Kroese, et al. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

[81] Christian Schwarz, Jürgen Teich, Emo Welzl, and Brian Evans. *On finding a minimal enclosing parallelogram*. Citeseer, 1994.

[82] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[83] Hao Chen, Xiaojuan Qi, Jie-Zhi Cheng, and Pheng-Ann Heng. Deep contextual networks for neuronal structure segmentation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1167–1173. AAAI Press, 2016.

[84] Evan Shelhamer, Jonathon Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[85] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[86] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[87] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[88] Paul A Bromiley, Judith E Adams, and Timothy F Cootes. Automatic localisation of vertebrae in dxa images using random forest regression voting. In *International Workshop on Computational Methods and Clinical Applications for Spine Imaging*, pages 38–51. Springer, 2015.

[89] Juying Huang, Fengzeng Jian, Hao Wu, and Haiyun Li. An improved level set method for vertebra ct image segmentation. *Biomedical Engineering Online*, 12(1):48, 2013.

[90] Andy Tsai, Anthony Yezzi, William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W Eric Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE transactions on medical imaging*, 22(2):137–154, 2003.

[91] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

[92] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[93] Takuya Narihira, Michael Maire, and Stella X Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2992–2992, 2015.

[94] Deepak Pathak, Philipp Krähenbühl, Stella X Yu, and Trevor Darrell. Constrained structured regression with convolutional neural networks. *arXiv preprint arXiv:1511.07497*, 2015.

[95] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David Snead, Ian Cree, and Nasir Rajpoot. A spatially constrained deep learning framework for detection of epithelial tumor nuclei in cancer histology images. In *International Workshop on Patch-based Techniques in Medical Imaging*, pages 154–162. Springer, 2015.

[96] Greg Slabaugh, Quynh Dinh, and Gozde Unal. A variational approach to the evolution of radial basis functions for image segmentation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[97] A Bhattachayya. On a measure of divergence between two statistical population defined by their population distributions. *Bulletin Calcutta Mathematical Society*, 35(99-109):28, 1943.

[98] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[99] W Rutkowski and A Rosenfeld. A comparison of corner-detection techniques for chain-coded curves, university of maryland. *Computer Science Center, TR-623*, 1978.

[100] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.

[101] Les Kitchen and Azriel Rosenfeld. Gray-level corner detection. *Pattern recognition letters*, 1(2):95–102, 1982.

[102] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, page 50. CITESEER, 1988.

[103] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443, 2006.

[104] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2010.

[105] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[106] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[107] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[108] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[109] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167):187–217, 1980.

[110] V. Torre and T. Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:147–163, 1984.

[111] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[112] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.

[113] Alan L Yuille and Tomaso A Poggio. Scaling theorems for zero crossings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):15–25, 1986.

[114] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, pages 1–16, 2017.

[115] Yupei Wang, Xin Zhao, and Kaiqi Huang. Deep crisp boundaries. In *Computer Vision and Pattern Recognition, 2017. CVPR 2017. Proceedings of the 2017 IEEE Computer Society Conference on*. IEEE, 2017.

[116] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[117] Judith MS Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

[118] Piotr Dollar, Zhuowen Tu, and Serge Belongie. Supervised learning of edges and object boundaries. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1964–1971. IEEE, 2006.

[119] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[120] Paul A Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[121] Charnchai Pluempitiwiriyawej, José MF Moura, Yi-Jen Lin Wu, and Chien Ho. Stacs: New active contour scheme for cardiac mr image segmentation. *IEEE Transactions on Medical Imaging*, 24(5):593–603, 2005.

[122] Jürgen Weese, Irina Wächter-Stehle, Lyubomir Zagorchev, and Jochen Peters. Shape-constrained deformable models and applications in medical imaging. In *Shape Analysis in Medical Image Analysis*, pages 151–184. Springer, 2014.

[123] Amal A Farag, Ahmed Shalaby, Hossam Abd El Munim, and Aly Farag. Variational shape representation for modeling, elastic registration and segmentation. In *Shape Analysis in Medical Image Analysis*, pages 95–121. Springer, 2014.

[124] Convert region of interest polygon to region mask. https://www.mathworks.com/help/images/ref/poly2mask.html#f6-465457/. Accessed: 2017-09-18.

[125] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.

[126] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.

[127] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. *arXiv preprint arXiv:1612.02103*, 2016.

[128] Carlos M Jarque and Anil K Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics letters*, 6(3):255–259, 1980.

[129] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[130] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Crisp boundary detection using pointwise mutual information. In *European Conference on Computer Vision*, pages 799–814. Springer, 2014.

[131] M. Lichman. Dataset: Anneal, auto, glass, lymp, splice, vehicle from UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

[132] ADHD-200 Consortium et al. The adhd-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Frontiers in systems neuroscience*, 6, 2012.

[133] Automatic vertebral fracture analysis and identification from VFA by DXA. http://lit.fe.uni-lj.si/xVertSeg/database.php. Accessed: 2017-10-23.

[134] Michael E Leventon, W Eric L Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *Computer vision and pattern recognition, 2000. Proceedings. IEEE conference on*, volume 1, pages 316–323. IEEE, 2000.

[135] Tenn Francis Chen. Medical image segmentation using level sets. *Technical Report. Canada, University of Waterloo*, pages 1–8, 2008.

[136] Yan Zhang, Bogdan J Matuszewski, Lik-Kwan Shark, and Christopher J Moore. Medical image segmentation using new hybrid level-set method. In *BioMedical Visualization, 2008. MEDIVIS'08. Fifth International Conference*, pages 71–76. IEEE, 2008.

[137] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[138] Tahir Nawaz and Greg Slabaugh. A bottom-up approach for the analysis of haustral fold ridges in ctc-cad. *Annals of the BMVA*, 2012(8):1–15, 2012.

[139] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[140] Jin Hyeong Park, Shaohua Kevin Zhou, Costas Simopoulos, Joanne Otsuki, and Dorin Comaniciu. Automatic cardiac view classification of echocardiogram. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[141] GN Balaji, TS Subashini, and N Chidambaram. Automatic classification of cardiac views in echocardiogram using histogram and statistical features. *Procedia Computer Science*, 46:1569–1576, 2015.

[142] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[143] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[144] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[145] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[146] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[147] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR), 2015*. http://arxiv.org/abs/1409.1556, 2015.

[148] MatConvNet. http://http://www.vlfeat.org/matconvnet/. Accessed: 2016-01-26.

[149] Sam Hallman and Charless C Fowlkes. Oriented edge forests for boundary detection. *arXiv preprint arXiv:1412.4181*, 2014.

[150] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[151] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[152] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[153] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[154] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[155] David H Hubel. Integrative processes in central visual pathways of the cat. *JOSA*, 53(1):58–66, 1963.

[156] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[157] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[158] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[159] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

Appendix A

Supplementary Experiments and Results

This appendix includes additional experiments and results performed in [71] on Dataset A, that influenced the methods described in Chapter 5.

A.1 Dataset A

The dataset used in this dissertation was received in batches. The first batch contained 138 images which were collected at the beginning of my PhD. The rest was not received until mid-2016. Our work before receiving the rest of the data was performed on a subset of 90 images from 138 images of the first batch. These 90 images were selected manually to reduce the complexity by ignoring most of the images with low contrasts and severe clinical conditions. However, some complex samples were intentionally kept to investigate

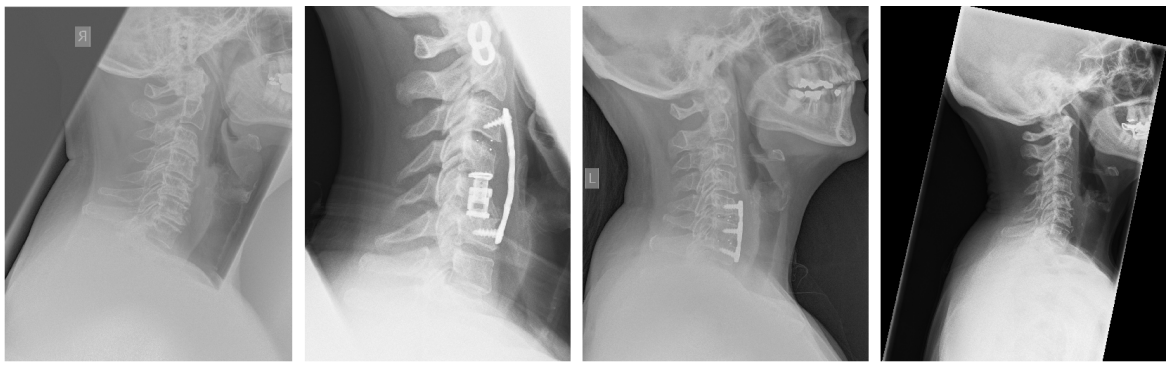


Fig. A.1 Example of images in Dataset A.

the response of the proposed methods on these variations. A few images from this dataset are shown in Fig. A.1.

A.2 Effect of ROI Selection on HarrisNB

As described in Sec. 5.1.1, the extracted region of interest (ROI) for the Harris-based naive Bayes corner detector (HarrisNB) around a vertebra can be a square, rectangle or trapezoid (Fig. 5.5). The results of this method with all three ROIs are reported in Table A.1. The results show an increase of error from the square ROI to the rectangle ROI and a sharp decrease for the trapezoid ROI. The increased area of the rectangle from the square sometimes triggers false corners whereas, for the trapezoid ROI, the axis-aligned vertebral boundaries result in much better edge and corner detection which decrease the error to 2.06 mm.

ROI type	Error in MM
Square	2.52
Rectangle	2.54
Trapezoid	**2.06**

Table A.1 Effect of different ROIs on HarrisNB.

A.3 Additional Feature Vectors for HoughF

The Hough forest-based vertebral corner detector (HoughF) also works on the patches extracted based on different ROIs. The patch extraction process has been discussed in Sec. 5.2.1. After extraction, the patches are converted into feature vectors. Several feature vectors were evaluated in [71] which have not been included in Sec. 5.2.

Intensity and gradient distribution-based feature vectors: The patches contain grey-level intensity distributions (I). Patch sizes in pixels vary from image to image based on the corresponding X-ray resolution. To generate feature vectors of similar length and distributions range, patches are resized to form a square shape and distributions are normalized. Four

sizes are considered: 30×30, 10×10, 5×5, and 3×3 pixels. These resized patches are then converted into feature vectors of length 900 (I30), 100 (I10), 25 (I5) and 9 (I3), respectively. The same procedure is repeated with the gradient distribution of each patch. Gradients are calculated in horizontal and vertical direction. After that the root-mean-square (RMS) of the magnitudes are considered. This process also produced feature vectors of length 900 (G30), 100 (G10), 25 (G5) and 9 (G3). A few examples of the resized patches for intensity and gradient distributions are shown in Fig. A.2.

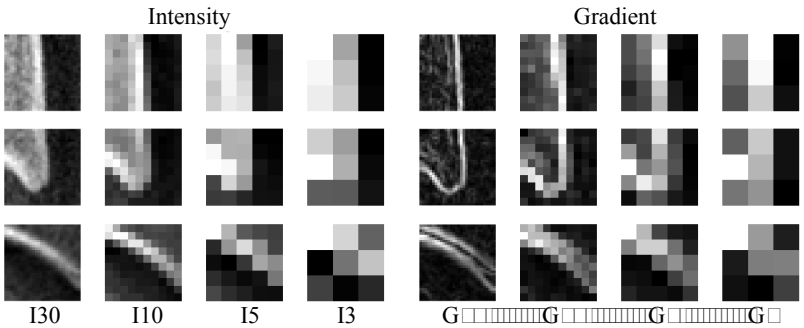


Fig. A.2 Appearance of intensity and gradient patches of different sizes.

Intensity and gradient-based Haar-like feature vectors: The Haar-Mixed feature vector described in Sec. 5.2.2 can be divided into two different feature vectors by splitting H_v of Eqn. 5.8 as Haar-Intensity (H_{v_i}) and Haar-Gradient (H_{v_g}), where

$$H_{v_i} = [f_1, f_2, f_3, ..., f_{10}], \tag{A.1}$$

$$H_{v_g} == [g_1, g_2, g_3, ..., g_{10}], \tag{A.2}$$

where f_x's and g_x's are from Eqn. 5.8.

Random Mirrored Feature (RMF): RMF is a novel feature vector introduced in [71]. For all the patches that arrive at a particular node, vectors are generated randomly from the patch center ($\boldsymbol{p}_c$). The length of the vectors L_x varies between a lower limit (L_{min}) to a higher limit (L_{max}). L_{max} is proportional to the size of the vertebra and not limited by the patch

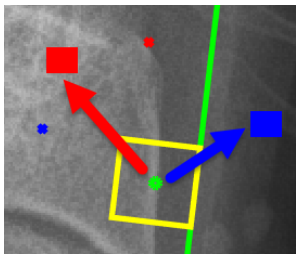


Fig. A.3 Random Mirrored Feature (RMF).

size as the feature vector is created from the original image. For each pair of vectors, one direction ($\boldsymbol{\delta}_1$) is randomly chosen. The direction of the other vector ($\boldsymbol{\delta}_2$) is computed by mirroring $\boldsymbol{\delta}_1$ either horizontally or vertically. This mirroring is applied so that two vectors can reach regions inside and outside the vertebra (see Fig. A.3). One feature value (f) is created from one pair of vectors by calculating the difference between the intensities (I) at those vector locations:

$$f = I(\boldsymbol{p}_c + L_1\boldsymbol{\delta}_1) - I(\boldsymbol{p}_c + L_2\boldsymbol{\delta}_2). \tag{A.3}$$

During training, 85 feature values are computed at a split node and the one that maximizes the information gain is chosen. This number is chosen based on an optimization process (Sec. A.4). The RMF feature vector is inspired by the work of [145], where random displacement feature vectors are used in 3D on depth images. Here, two types of RMF are used. The first type considers only one pixel at each of the random vector locations (RMF-1P) and the other type considers the average intensity of a 3×3 pixel box centered at each vector location (RMF-B) to calculate the feature values using Eqn. A.3.

Convolutional Neural Net (CNN) feature vectors: Off-the-shelf CNN feature vectors using a pre-trained deep neural network have been used for various complicated image processing tasks with great success [146]. Based on this insight, a pre-trained deep network (VGG-16) is used to convert the patches into CNN feature vectors [147, 148]. Two sizes of patches are considered for CNN feature vectors. The first one is the standard non-overlapping patch size used for generating the Haar-like features described in Sec. 5.2.1 and 5.2.2. The second is a larger and overlapping patch size to provide more spatial information about the patch to the network. The size of the larger patch is the same as L_{max} used for RMF

feature vectors. The CNN feature vector created from the standard size patch is denoted by CNN-S and the other is denoted by CNN-B. The pre-trained network has 37 layers including the final classification layer. The output response of the 36th layer is used as the feature vector. A detailed description of this pre-trained network can be found in [147]. This network requires a three channel RGB image of size 224×224 pixels as an input. To conform to this requirement, our patches (both standard and large) are resized to 224×224 and same grey-level intensity channel is repeated in each of the three channels. The length of the CNN feature vectors is 1,000.

Structured forest (SF) feature vectors: Edge detection is a fundamental problem in image processing. Recent work [77, 78, 149] in this domain have shown significant improvement from the baseline [150]. All these articles use a complex feature vector. We call this feature vector structured forest (SF) feature. The computation process of this feature vector has been described in Sec. 3.1.2. The length of this feature vector is 6,116. Both patch sizes are considered to compute this vector: features computed from standard size patch is denoted by SF-S and larger patch (described in the previous paragraph) is denoted as SF-B.

A.4 Optimization of Parameters for HoughF

As described in Sec. 5.2.5, there are some hyper-parameters in HoughF methods. They can be listed as: number of trees ($nTree$), maximum allowed depth of a tree (D), minimum number

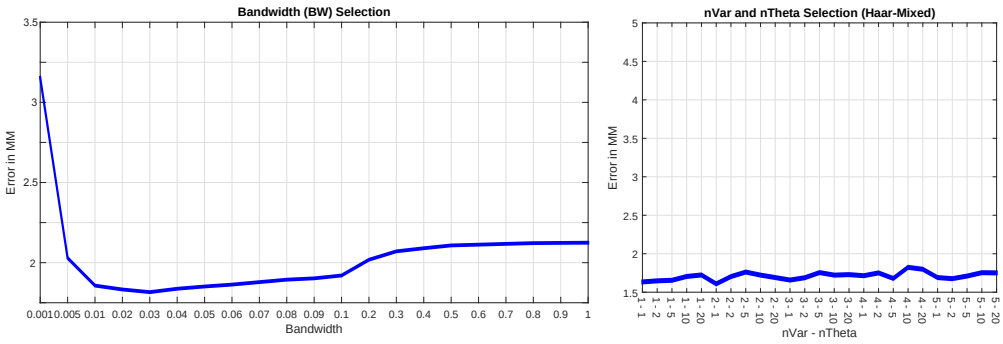


Fig. A.4 Bandwidth (BW), number of variables ($nVar$) and number of thresholds ($nThresh$) selection.

of elements at a node (*nMin*), number of variables to look at in each split nodes (*nVar*) and number of thresholds (*nTresh*) to consider per variable. The kernel density estimation (KDE) function also requires a bandwidth (*BW*) which is the variance, σ_k^2, in Eqn. 5.15. To accurately optimize the forest for all these parameters, the experiment is conducted for a single corner of a vertebra for all the images in Dataset A in a 10-fold manner, and the localization algorithm is repeated with different parameters. Due to time constraints, a sequential optimization approach is followed instead of a multi-dimensional approach. The sequence of the parameters was chosen based on the understanding of the problem. The sequence followed is: the bandwidth (*BW*), number of trees (*nTree*), maximum tree depth (*D*), minimum element at a node (*nMin*) and number of variables (*nVar*) & number of thresholds (*nThresh*) for each feature vector in a two-dimensional fashion. Most related variables were chosen for the two-dimensional optimization. The cost function for the optimization is the

Parameters	Feature type	Feature length	Value
BW (σ_k^2)	N/A		0.03
nTree			100
D			10
nMin			5
nVar-nThresh	SF	6116	85-10
	RMF	Infinity[a] (5000)	
	CNN	1000	30-10
	I30	900	15-20
	G30		
	I10	100	10-2
	G10		
	I5	25	2-1
	G5		
	Haar-Mixed	20	
	I3	9	1-5
	G3		
	Haar-Intensity	10	
	Haar-Gradient		

Table A.2 Optimized parameters for corner localization.

[a] textRMF is dynamically created with random vectors. Technically, the feature length is infinity. However, for the sake of implementation, the choice of possible vector locations was quantized such that maximum possible feature length becomes 5000.

average Euclidean error between corners predicted by the forest and annotated manually by experts. The error vs. parameter value graphs are shown in Fig. A.4. The selection of the BW is reported in Fig. A.4a and $nVar\text{-}nTheta$ selection for Haar-Mixed features is reported Fig. A.4b. The BW selection graph indicates that the lowest error occurs with $BW = 0.03$. Lowest error for $nVar\text{-}nTheta$ selection occurs at 2-1. The next option could have been 4-5, but that would increase the computational cost by a factor of ten. Similarly, all the parameters are optimized based on the lowest error and corresponding computational cost. The chosen parameters are reported in Table A.2.

A.5 Additional Results for HoughF

The Hough forest-based vertebral corner detector (HoughF) has been evaluated for different ROIs and feature vectors described in Sec. A.3. The results are summarized in Table A.3. The average performance in the last column shows that the rectangle and trapezoid ROIs perform marginally better than the square ROI. In between the rectangle and trapezoid ROIs, the earlier is slightly better. However, the pattern is not consistent with all the features. The lowest error of 2.01 mm is achieved with the Haar-Mixed feature for the rectangle ROI. Results reported in Table A.1 for HarrisNB clearly show that the trapezoid ROI can help when the algorithm uses horizontal and vertical gradients to detect edges and corners as this ROI aligns the vertebral edges (see Fig. 5.5). But the forest framework does not use this information explicitly. Thus the differences are not noticeable. Among all the features, Haar-Mixed performed the best followed by Intensity 5 (I5). The advanced features (RMF, CNN, SF) do not perform very well in terms of the error.

ROI type	Feature vectors																		Average
	Intensity				Gradient				Haar-like features			CNN		SF		RMF			
	I30	I10	I5	I3	G30	G10	G5	G3	Intensity	Gradient	Mixed	S	B	S	B	IP	B		
Square	2.11	2.09	2.07	2.09	2.13	2.12	2.09	2.12	2.09	2.14	2.05	2.09	2.11	2.08	2.09	2.08	2.08		2.10
Rectangle	2.10	2.09	2.04	2.04	2.13	2.10	2.05	2.07	2.08	2.07	**2.01**	2.08	2.09	2.07	2.08	2.08	2.06		2.07
Trapezoid	2.12	2.08	2.03	2.07	2.12	2.08	2.05	2.08	2.09	2.10	2.05	2.10	2.08	2.08	2.07	2.09	2.08		2.08

Table A.3 Effect of different ROIs on HoughF for different feature vectors.

Appendix B

Random Forest and Deep Learning

In this appendix, we briefly introduce two key machine learning algorithms that have been explored in this dissertation: random forest and deep learning.

B.1 Random Forest

Random forest is a popular machine learning algorithm that has been used in many computer vision tasks with great accuracy and performance [76, 105]. The algorithm can be used for supervised learning e.g. classification or regression and also for unsupervised learning e.g. clustering. The algorithm utilizes the concepts of randomness and generalization. It is faster to train than its deep learning counterparts. The algorithm also requires less data for training making it a popular choice in the medical image analysis domain. It has been used in many related state-of-the-art articles [38, 49, 51, 57].

A random forest is an ensemble of binary decision trees. A decision tree consists of several nodes (see Fig. B.1). The root node is where all data points start their journey. Other nodes can be categorized as split nodes and leaf nodes. Split nodes are where the data gets divided into left and right branches based on a cost function called the information gain (IG). The data splits are performed with sparse optimization of the cost function. The cost function considers only a small subset of all possible data splits and optimizes within that small subset.

This optimization is intentionally kept imperfect. Each data point ends at a leaf node when the tree reaches a maximum depth or number of data points in a node become lower than a threshold.

Random forest exploits generalization among the trees. The accuracy of an individual tree is sacrificed by introducing randomness and weak optimization of split nodes to achieve better results over the forest. The randomization is added by sampling random subsets of data for training each tree (bagging) and by randomly choosing only a subset of variables and thresholds to optimize the IG at each node (RNO: randomized node optimization). Both types of randomization help to reduce overfitting of the model and improve the performance

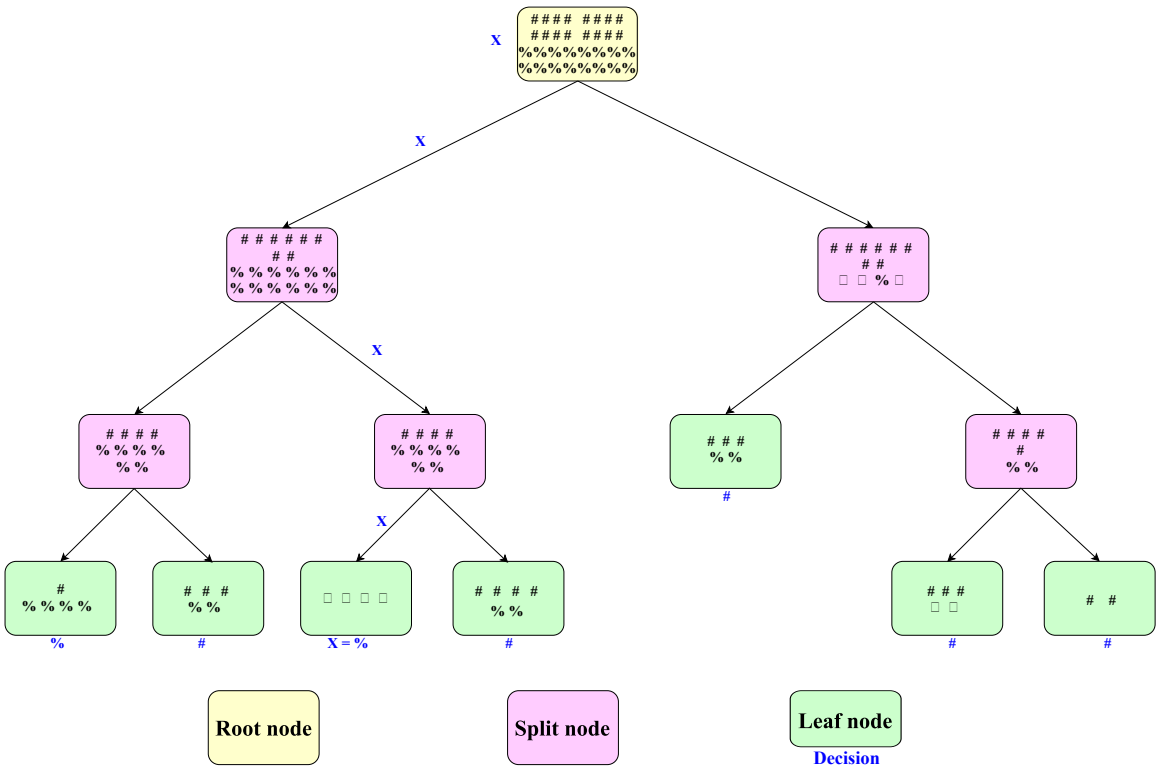


Fig. B.1 Decision tree: a tree starts with a set of training data at the root node. Based on a cost function the data is divided into left and right child nodes. The process is repeated at the split nodes. Each branch of the tree ends with a leaf node. Leaf nodes are associated with a decision based on the set of training data it contains. At test time, a new data point, X, starts at the root node and follows a tree branch based on the splits learned during training. A decision can be taken based on which leaf node it reaches. In this toy example, we show a decision tree for a set of 32 characters containing two letters: '#' and '%'.

of the overall forest prediction.

If the forest predicts discrete values, then the forest is called a classification forest, if it predicts continuous values then the forest is called a regression forest. The difference during training is to calculate the information gain (IG) accordingly. The IG can be calculated with different types of entropy based on the application. Entropy is a measure of randomness of a collection of data points, the split node divides the data so that the overall entropy becomes less. The leaf nodes are responsible for predicting a class label or for regressing an output variable. Several prediction models for classification and regression have been used for different problems.

We have used random classification forests for the spine localization problem which is discussed in Chapter 3. Another hybrid random forest that performs classification and regression together, called Hough forest, is described in Chapter 5 for vertebral corner localization.

B.2 Deep Learning

Recently, the term 'deep learning' has become very popular in the field of machine learning, artificial intelligence and computer vision. The term covers a range of artificial neural network-based machine learning techniques. In this section, we start by describing a single perceptron which is essentially the atomic unit of any neural network. Building on this, we then continue by introducing multi-layer networks (MLP) or fully connected networks, convolutional neural networks (CNN) and finally, fully convolutional networks (FCN), the last of which has been extensively used in this dissertation to solve several localization, segmentation and prediction problems.

B.2.1 Perceptron

The initial idea of a perceptron dates back to the work of Warren McCulloch and Walter Pitts in 1943 [151], who drew an analogy between biological neurons and simple logic gates with binary outputs. In more intuitive terms, neurons can be understood as the sub-units of a neural network in a biological brain. Here, the signals of variable magnitudes arrive at the dendrites. Those input signals are then accumulated in the cell body of the neuron and if the accumulated signal exceeds a certain threshold, an output signal is generated which is passed on by the axon. The process is summarized in Fig. B.2.

Based on this simplified understanding of the neuron, Frank Rosenblatt proposed the perceptron learning rule [152]. The key idea was to define an algorithm to learn the values of a set of weights, $\boldsymbol{w}$, that are multiplied with the input features in order to make a decision whether a neuron fires or not, essentially, solving a binary classification problem. The structure of a perceptron is highly motivated by the structure of the biological neuron.

Given an input vector $\boldsymbol{x}$, the output of the network can be found as:

$$\hat{z} = f(\boldsymbol{w}^T \boldsymbol{x} + b) \tag{B.1}$$

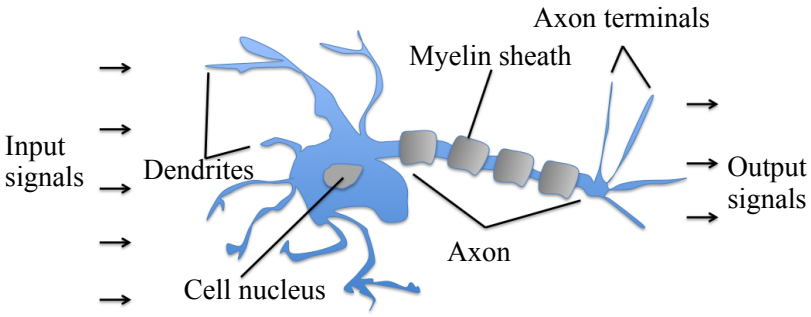


Fig. B.2 Schematic of a biological neuron.

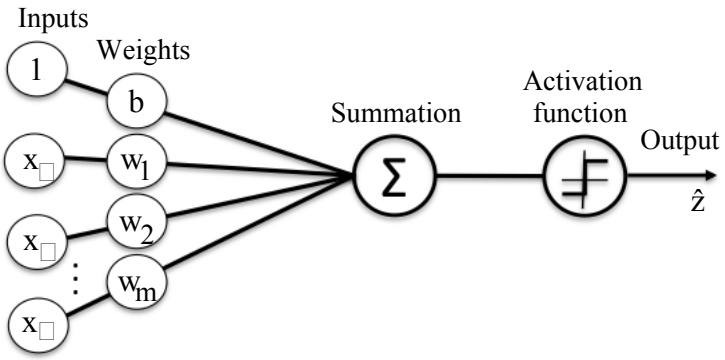


Fig. B.3 Schematic of Rosenblatt perceptron.

where f is the activation function. For probabilistic output, the activation function is usually a sigmoid function which squashes the input to a valid probabilistic range of 0 to 1.

$$\sigma(a) = \frac{1}{1+e^{-a}} \tag{B.2}$$

Now, if we know the actual label, z for out input vector $\boldsymbol{x}$, then we can define a loss term, L,

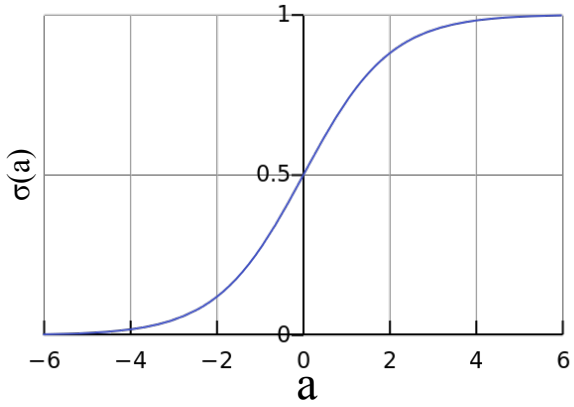


Fig. B.4 Sigmoid function.

as:

$$L = \frac{1}{2}(z-\hat{z})^2 \tag{B.3}$$

For dataset containing N training data with known labels, the loss function can be defined over the dataset as:

$$L = \frac{1}{2N}\sum_{i=1}^{N} L_i \tag{B.4}$$

$$L_i = (z_i - \hat{z}_i)^2 \tag{B.5}$$

or

$$L_i(\boldsymbol{w}, \boldsymbol{x}) = \left(z_i - \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x}_i + b)}} \right)^2 \tag{B.6}$$

Now, we can compute the derivatives:

$$\frac{\partial L_i}{\partial w_j} = 2(z_i - \hat{z}_i)\hat{z}_i(1 - \hat{z}_i)x_{ij} \tag{B.7}$$

$$\frac{\partial L_i}{\partial b} = 2(z_i - \hat{z}_i)\hat{z}_i(1 - \hat{z}_i) \tag{B.8}$$

Then the weights can be updated using gradient descent algorithm [86]:

$$w_j^t = w_j^{t-1} - \eta \frac{\partial L}{\partial w_j^{t-1}} \tag{B.9}$$

$$b^t = b^{t-1} - \eta \frac{\partial L}{\partial b^{t-1}} \tag{B.10}$$

where t represents optimization iteration number and η is the step size of the gradient descent algorithm, also known as the learning rate. b^0 and w_j^0's are initialized randomly. The training data is usually randomly sorted and passed through the perceptron several times (epochs), to allow the optimization process to reach convergence.

The perceptron described above is applicable for binary classification problems. For multiclass problems, multiple neurons (equal to the number of classes in the dataset) can be used, followed by a softmax layer. The softmax layer normalizes the output values to form a valid probability distribution over the output classes. For example, for a three-class problem (z_1, z_2 and z_3), the network of Fig. B.5 can be used.

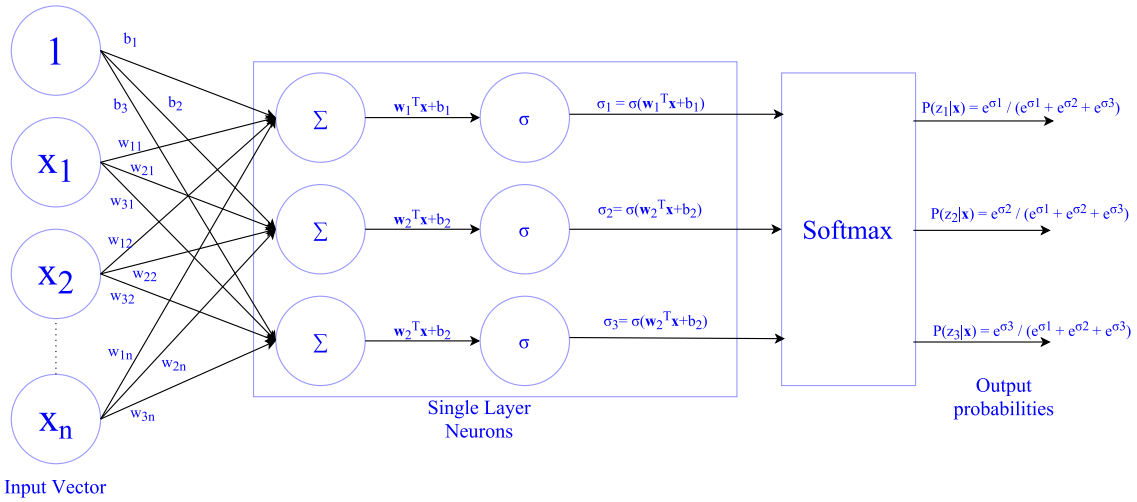


Fig. B.5 Multiclass classification using perceptrons.

B.2.2 Multi-layer Perceptron

The network of Fig. B.5 can solve multiclass problems but is still not suitable for complex classification tasks. The model has $(n+1)\times 3$ parameters only, where n is the length of the input vector. To model complex problems, the number of parameters can be increased by cascading more layers between the input and output layers, these layers are often termed as 'hidden layers' in the literature. As long as each of the layers is differentiable, gradient descent can be used to optimize the weights. The propagation of the derivative of the error or the loss function from the output layer to the input layer is called backpropagation. A deep multi-layer perceptron is shown in Fig. B.6. This network is also known as the fully connected network as each neuron is connected to all other neurons of the consecutive layers through weights.

B.2.3 Convolutional Neural Network

The multi-layer perceptron is a powerful machine learning tool. Researchers have proven that given enough neurons it can approximate any function [153]. But it has a huge number of parameters that require memory and computation power. Also, the network is not suitable for images. Although one can vectorize the image and use an MLP for a particular problem [154], such an approach fails to take into account the fact that the image consists of simple and

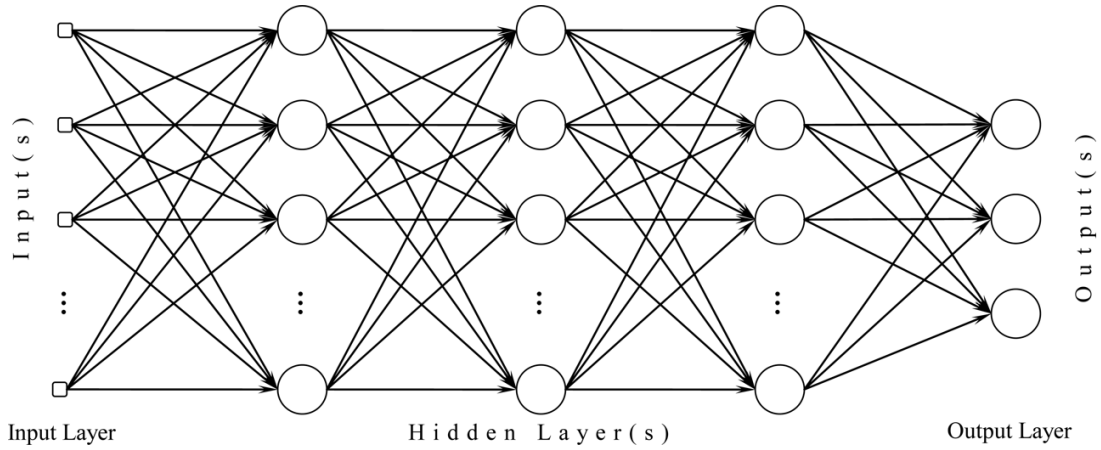


Fig. B.6 Multi-layer perceptron or fully connected network.

repetitive structures such as edges of different orientations, distributed over a 2D plane. Research into the brain's visual pathway also supported the fact that the first layer in the path detects edges, from which the next layers detect higher level features and finally it decides what the image is or is about [155]. Based on these developments in different disciplines, in 1998, Yann LeCunn proposed convolutional neural networks (CNN) for handwritten digit classification [156]. The network is shown in Fig. B.7. The network takes a single channel 32×32 pixel image of handwritten digits and classifies these into ten classes representing digits from 0 to 9. Detailed descriptions of different layers used in CNNs can be found in Sec. B.2.5.

But, CNNs were mostly unused for complex computer vision problems because of their requirements for computing power, large memory and huge training datasets. Other machine learning algorithms, like SVM, AdaBoost and random forest, became the state-of-the-art in the field. In 2012, Alex Krizhevsky reintroduced the CNN in the computer vision community and won the ImageNet classification challenge by outperforming the previous state-of-the-art by a large margin [23]. This network is known as AlexNet. The challenge had a thousand image categories and 1.5 million images to train on. Since 2012, several innovative variants of CNNs, like VGGNet [147], ResNet [87], GoogLeNet [157] have been proposed achieving better and better performance. In 2015, PReLU-nets achieved an error of 4.94% surpassing

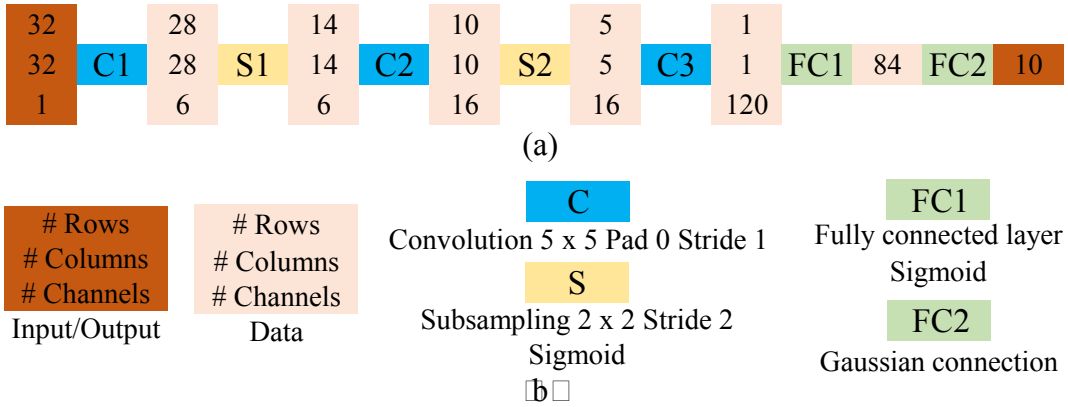


Fig. B.7 Convolutional Neural Network for digit classification (a) network architecture (b) legend.

the human performance level (5.1%) [158]. Fig. B.9 shows the architecture of the VGG-16 network. The network uses the same convolution and pooling throughout the network. This is one of the most used CNN architectures in the literature.

B.2.4 Fully Convolutional Network

The fully convolutional network (FCN) is an evolution of the CNN, suited for image segmentation problems. In segmentation problems, instead of classifying the whole image into a certain class, each pixel has to be classified. This problem has been termed as a dense classification problem. In [84], the fully connected layers are replaced by convolutional layers. The

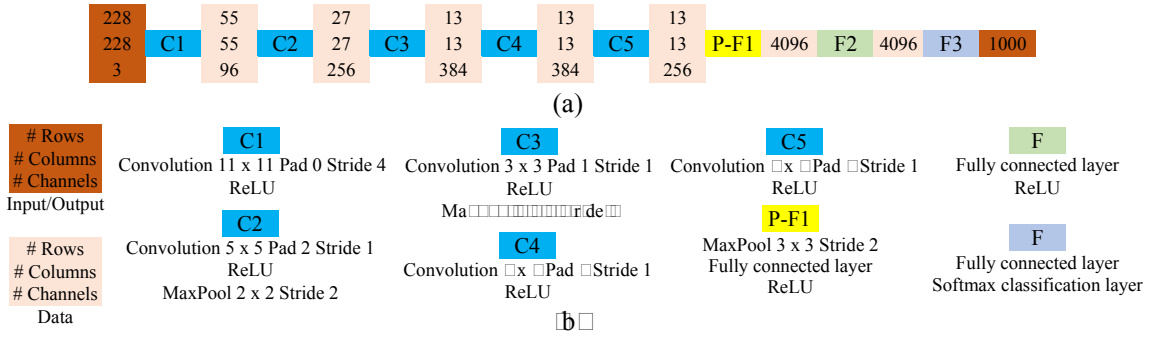


Fig. B.8 CNN (AlexNet) for large-scale image categorization (a) network architecture (b) legend.

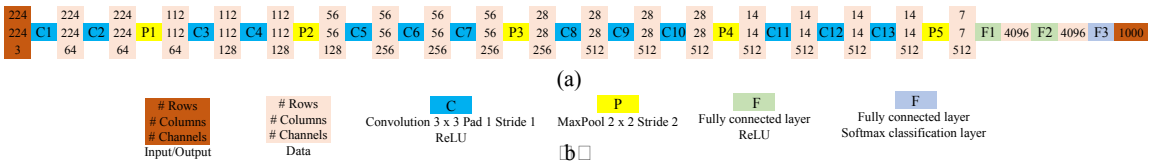


Fig. B.9 VGG-16 Net (a) network architecture (b) legend.

final map is then upsampled using a deconvolution layer to produce a segmentation result of the same size as the input. We show the fully convolutional version of the VGG-16 Net in Fig. B.10. This network has been used for segmentation of 21 object classes of the PASCAL VOC 2011 dataset [159], thus the final layer consists of 21 channels.

The single step upsampling strategy causes the segmentation map to become coarse losing fine details about the object boundaries. To solve this problem, DeConvNet [85] was proposed. This network consists of two paths: a contracting path that reduces the spatial dimension through pooling and an expanding path that upsamples the contracted activations into the same input resolution in stages. The DeConvNet achieves upsampling through unpooling which uses the maxpooling indexes (called switch variables) to upsample the input map.

A similar dense classification network, UNet, is proposed in [82]. This network also has a contracting path and an expanding path. But it uses a different upsampling strategy. Instead of unpooling layers, it uses a learnable deconvolution layers to achieve upsampling in the expanding path. The network also shares information between the contracting path and the expanding path by concatenation of data matrices. UNet is designed for biomedical

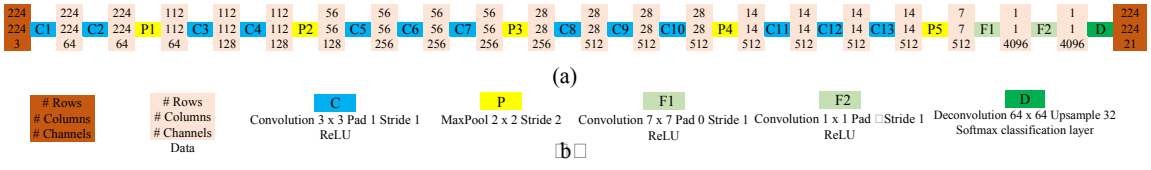


Fig. B.10 Fully convolutional network for image segmentation (VGG-16 FCN) (a) network architecture (b) legend.

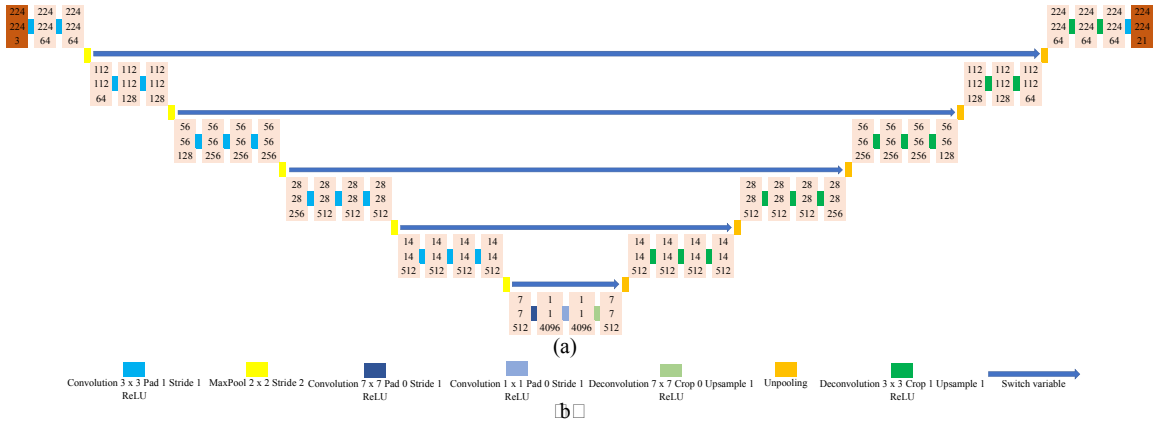


Fig. B.11 Deconvolutional network for image segmentation (a) network architecture (b) legend.

image segmentation. The network takes an input image of size 572×572 and creates a segmentation map of size 388×388. The problem is a binary dense classification problem thus the final output has only two channels.

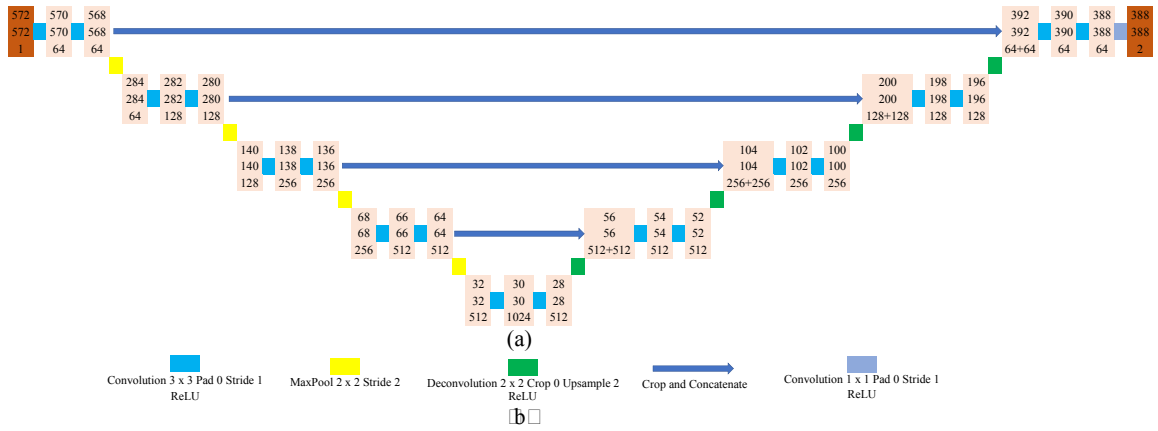


Fig. B.12 UNet for medical image segmentation (a) network architecture (b) legend.

B.2.5 Layers in a Deep Neural Network

All the deep convolutional neural network discussed above consists of several layers. Each layer performs certain computation on the input and produces an output feature map which is

usually fed to a next layer in the network. In the following subsections, we look into different layers used in those deep architectures in more detail.

B.2.5.1 Convolutional Layer

The convolutional layer performs 2D convolution on the input feature map x with K number of filters f and produces output feature map y. The process is graphically illustrated in Fig. B.13. Mathematically, the values in the output feature map can be computed as:

$$y(i,j,k) = \sum_{c=1}^{C} \sum_{P_i=-P}^{P} \sum_{P_j=-P}^{P} x(Si-S+1+P_i, Sj-S+1+P_j, c) f(P_i+P+1, P_j+P+1, c, k) + b_k, \tag{B.11}$$

where $k = 1,2,...,K$; S is a hyper-parameter called 'stride' in the literature and other symbols bear the same meaning described in Fig. B.13d. The number of channels in each filter is determined by the number of channels in the input feature map (C). The number of channels in the output feature map is determined by the number of filters (K). The number of rows (H_y) and columns (W_y) in the output feature map is determined by the following equations:

$$H_y = \frac{H_x - H_f + 2P}{S} + 1, W_y = \frac{W_x - W_f + 2P}{S} + 1. \tag{B.12}$$

In the example illustrated in Fig. B.13, $H_x = W_x = 5$, $H_f = W_f = 3$, $P = 1$ and $S = 2$. The $f(*,*,*,*)$ and b_* are the trainable parameters of the layer. The convolutional layers have been used throughout this dissertation in the all the proposed networks.

B.2.5.2 Subsampling and Maxpooling

The subsampling and maxpooling layers reduce the spatial size of the input feature map and produce a smaller output feature map. These operations are performed in each channel separately, thus do not change the number of channels in the output feature map. For a single channel input, the process is illustrated in Fig. B.14. For subsampling layer, the values in the

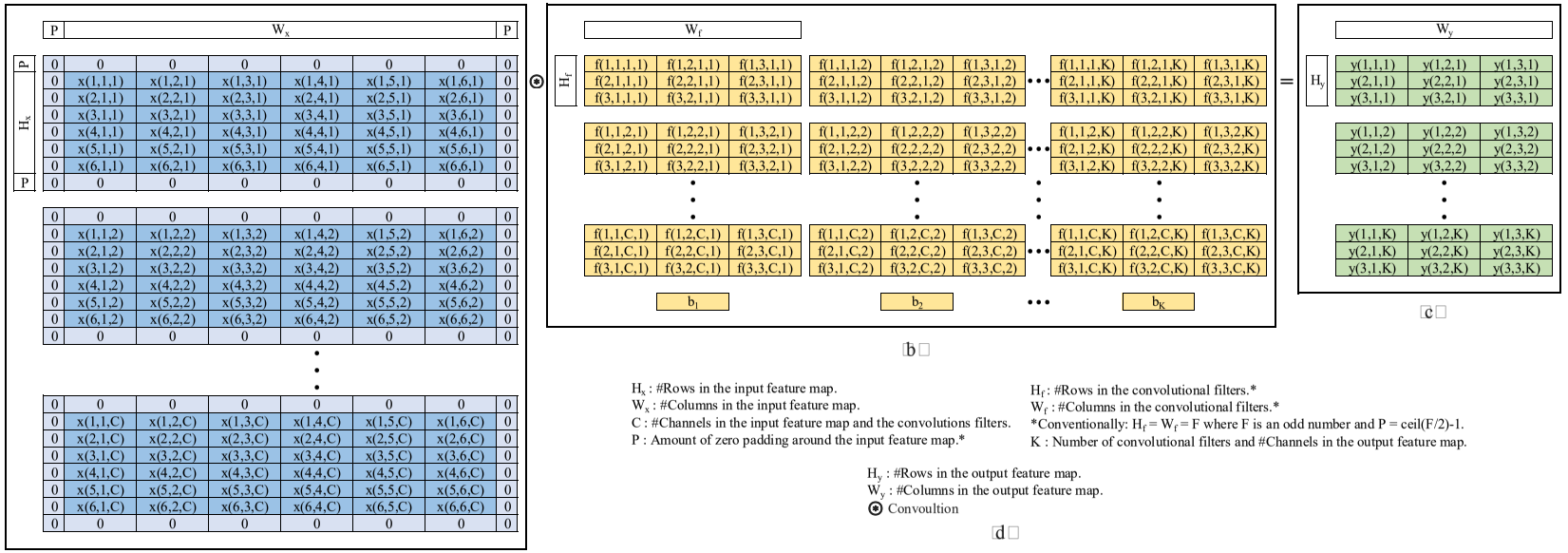


Fig. B.13 Convolutional layer (a) input feature map (b) filters (c) output feature map (d) legend.

output feature map can be computed as:

$$y(i,j) = w_{ij} \sum_{m_i=1}^{M} \sum_{m_j=1}^{M} x(Si - S + m_i, Sj - S + m_j) + b_{ij}, \tag{B.13}$$

where S is the stride, $M \times M$ is the size of the receptive field and w_{ij}s and b_{ij}s are trainable parameters of the layer. For non-overlapping subsampling, $S = M$. For the example shown in Fig. B.14 and subsampling layers used in the CNN of Fig. B.7, $S = M = 2$.

In contrast to the subsampling layers, the maxpooling layers do not have any trainable parameters. The maxpooling operation simply choose the maximum value from the input feature map under the receptive field.

$$y(i,j) = \max\{x(Si - S + m_i, Sj - S + m_j) : m_i = 1,2,...,M, m_j = 1,2,...,M\}. \tag{B.14}$$

The maxpooling layers have been used throughout this dissertation and mentioned as 'MaxPool' in the network diagrams.

B.2.5.3 Gaussian Connection

The Gaussian connection layer is a particular version of the fully connected layer where the output values are computed using Eqn. B.15 [156]. This layer is illustrated in Fig. B.15. The

x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,1)	x(4,2)	x(4,3)	x(4,4)

(a)

y(1,1)	y(1,2)
y(2,1)	y(2,2)

(b)

Fig. B.14 Subsampling and maxpooling (a) input feature map (b) output feature map.

output values (y) can be computed as:

$$y_i = \sum_{j}^{L_{in}} (x_j - w_{ij})^2; i = 1, 2, ..., L_{out}, \tag{B.15}$$

where L_{in} and L_{out} are the length of the input vector and output vector, respectively. This layer has been used in the CNN proposed in [156] (see Fig. B.7).

B.2.5.4 Rectified Linear Unit (ReLU)

The rectified linear unit (ReLU) performs a simple non-linear operation and used as a replacement of the sigmoid operation:

$$y(x) = \begin{cases} x & if x > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{B.16}$$

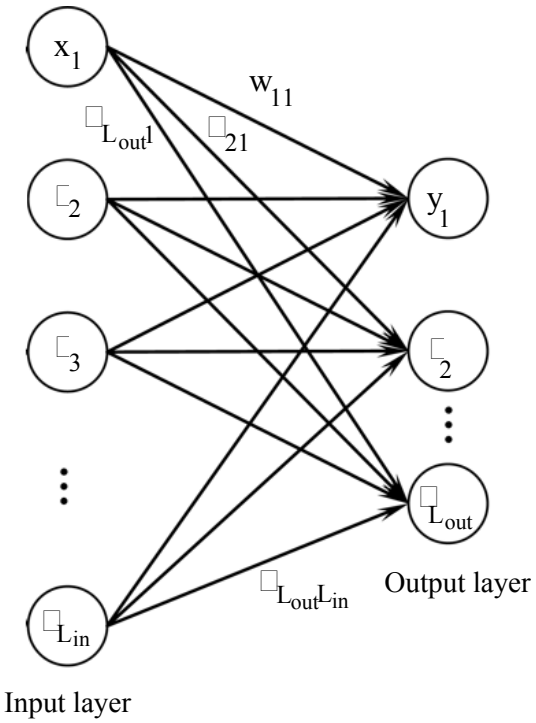


Fig. B.15 Gaussian connection.

The operation is also illustrated in Fig. B.16. The rectified linear units have been used throughout this dissertation.

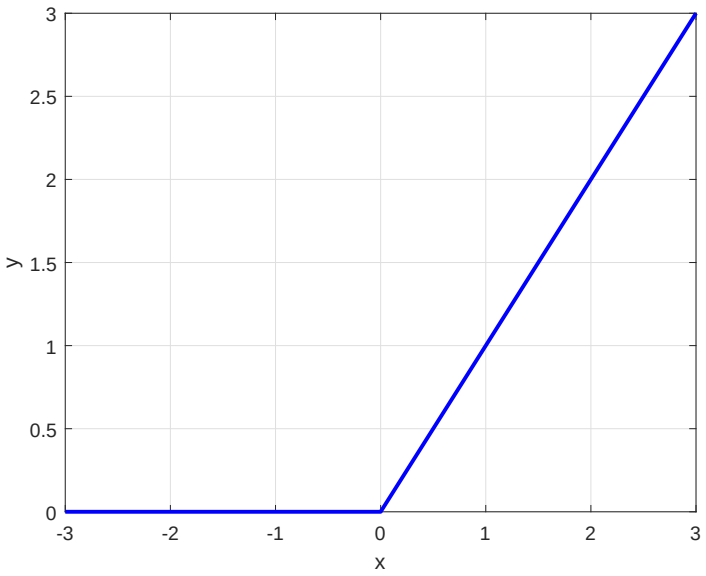


Fig. B.16 Rectified linear unit (ReLU).

B.2.5.5 Deconvolutional Layer

The deconvolutional layer performs a backward convolution operation. In Fig. B.13, we have illustrated a forward pass of the convolution operation with a stride, $S = 2$, where the spatial size of the output was reduced by a factor of S. During backpropagation, this convolution layer has to perform an upsampling operation. The deconvolutional layer simply switches the forward pass and the backward pass operations of the convolutional layer. The operation is also termed as backward convolution. Due to its backward nature, the hyper-parameter 'stride' for the deconvolutional layer is renamed as 'upsample' and the term 'padding' is replaced by 'cropping'. This layer has been used in the networks described in Sec. B.2.4.

B.2.5.6 Unpooling and Switch Variable

The unpooling operation is introduced in [85] to perform a backward maxpooling operation. The maxpooling operation chooses the maximum value from a receptive field. The index

of this maximum value is saved as 'switch variable'. At the time of unpooling, this switch variable is used to upsample the input feature map. The process is graphically illustrated in Fig. B.17 for a maxpooling operation with 'stride' 2 and receptive field of size 2×2. This layer has been used in the DeConvNets described in Sec. B.2.4 and 3.2.2.

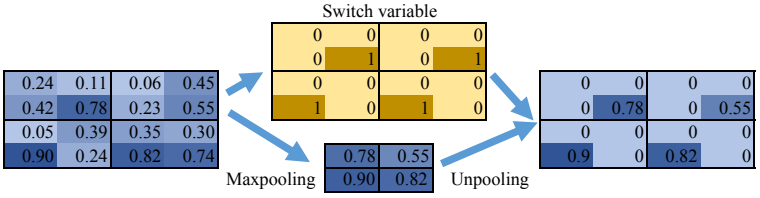


Fig. B.17 Unpooling and switch variable.