FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

STEERABLE CONVOLUTIONAL NEURAL NETWORKS

By

DONGHANG LI

A Dissertation submitted to the
Department of Statistics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2020

Donghang Li defended this dissertation on 6/18/2020.
The members of the supervisory committee were:

Adrian Barbu

Professor Directing Dissertation

Anke Meyer-Baese

University Representative

Jonathan Bradley

Committee Member

Chong Wu

Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

Dedicated to my family and friends.

ACKNOWLEDGMENTS

First I would like to thank my major advisor Dr. Adrian Barbu, who not only guided and encouraged me in my research, but also provided experience on career development throughout my PhD study.

I am also grateful to my committee members, Dr. Anke Meyer-Baese, Dr. Jonathan Bradley, and Dr. Chong Wu, for their patience, time and comments on improving this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Following is the list of abbreviations in this dissertation.

- **CNN:** Convolutional Neural Network
- **SCNN:** Steerable Convolutional Neural Network
- **SFCNN:** Steerable Filter Convolutional Neural Network
- **FCN:** Fully Convolutional Network
- **SQF:** Spherical Quadrature Filters
- **PBT:** Probabilistic Boosting Tree
- **NMS:** Non-maximal Suppression
- **TSF:** Trained Steerable Filters
- **FB:** foreground-background
- **BFGS:** Broyden–Fletcher–Goldfarb–Shanno
- **ROC:** Receiver Operating Characteristics
- **AUC:** Area Under the ROC Curve

ABSTRACT

Convolutional Neural Networks (CNNs) are widely used and have an impressive performance in detecting and classifying objects. However, the CNN's performance is sensitive to variations in rotation, position or scaling of the objects to be detected. Fully Convolutional Neural Networks were trained for guidewire detection and retinal vessel detection in this dissertation. We highlight what challenges are encountered during training for the guidewire detection. We present a novel method for simultaneously detecting the guidewire pixels and predicting the guidewire orientation using trained oriented filters. We also show how to train, in the same framework, these oriented filters as steerable filters in a low rank representation. We introduce the Spherical Quadrature Filters (SQF) for guidewire detection and show how they can be used to improve the training data. We propose a steerable CNN that can detect an object rotated by an arbitrary angle without being rotation invariant. The proposed steerable CNN is discriminative like a regular CNN, but it has a latent parameter representing the object's 2D orientation. For any value of this parameter, the steerable CNN will be sensitive to detect only objects having that orientation. We apply the SQF, CNN and steerable CNN to detect the guidewire in fluoroscopy (real-time X-ray) images and to detect vessels in retina (fundus) images. The guidewire is a thin wire used in coronary angioplasty interventions, which are visualized using fluoroscopic images. The fundus images are noisy because of the similarity between the background and the vessels. Experiments show that the steerable CNN outperforms the regular CNN and other popular approaches such as the Frangi filter, the Steerable Quadrature Filter and a state of the art trained classifier based on hand-crafted feature.

CHAPTER 1

INTRODUCTION

The guidewire is a thin medical equipment used in coronary angioplasty interventions. During the intervention, a catheter is inserted through the femoral artery all the way to the heart, and a guidewire is used to guide different tools beyond the catheter, inside the heart. Then the cardiologist inserts a balloon into the obstructed coronary artery, inflates it to widen the narrowing, and places a stent there to keep the blood vessel open. All these operations are monitored by the cardiologist using real-time X-ray (fluoroscopy) images. The fluoroscopy images are usually low-dose to limit the amount of radiation received by the patient, which makes the images noisy and the guidewire poorly visible. Guidewire detection is a challenging problem with wide applications in coronary angioplasty interventions.

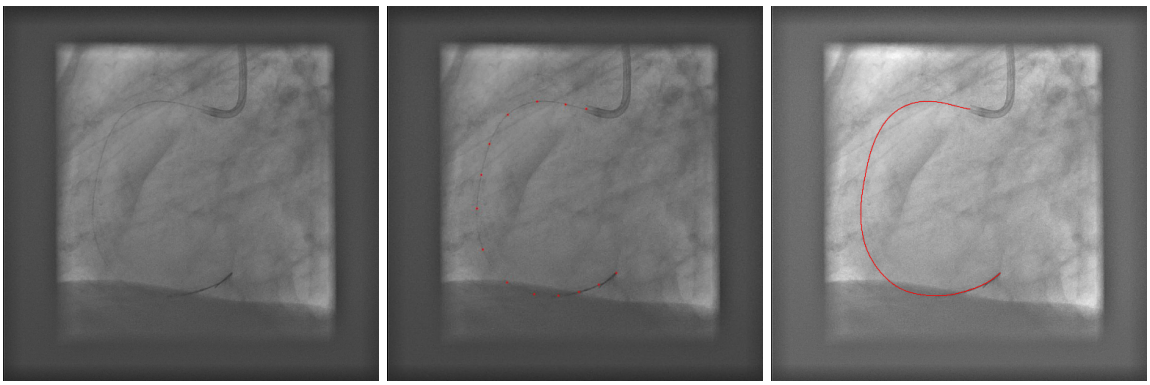

Figure 1.1: From left to right: an example of the fluoroscopic images of guidewire, an example of fluoroscopic images with annotations and an example of guidewire annotated using B-spline.

Examples of fluoroscopic images of guidewire are shown in Figure 1.1. As the figure reveals, the guidewire is thin and hardly visible. Thus, robust detection of the guidewire could help the cardiologist have a better visualization during the intervention. Furthermore, robust detection of guidewire can possibly further reduce the radiation dose administered to the patient.

Filter and learning based approaches have been applied to solving this task. In this dissertation, we proposed various methods for detecting the guidewire in fluoroscopic images. Our proposed Steerable Convolutional Neural Network can be applied to this task, and it outperforms the existing popular methods and a state of art trained classifier with Haar features. The details of guidewire detection will be introduced in Chapter 2.

The retinal blood vessels transport blood to the eye. Blood is essential for the retina to ensure that the eye gets nutrition and oxygen and removes the waste the retina produces. The retina consists of several layers and it is the light-sensitive tissue of the eye. There are diseases such as diabetes, hypertension, glaucoma, etc that affect the retina and its blood vessels. By measuring the diameter of the retinal arteries and vein, some recent studies [38, 39, 29, 42] show that diseases such as cardiac, endocrine, ophthalmic and neural, can be tracked at an early stage. More details about the related work and methods will be discussed in Chapter 3.

Images of the retina, also known as fundus images and shown in Figure 1.2, are used to visualize the retinal blood vessels. These visualize the bottom of the eye. Applying image processing techniques in analysing the retinal blood vessels is helpful for diagnosis of these diseases.

Figure 1.2: Examples of the fundus image (left) and the annotations(right).

1.1 Outline

Below we give a brief description about how the dissertation is organized.

In Chapter 2, we introduce four novel methods for guidewire detection, a Fully Convolutional Network, the trained steerable filters, the Spherical Quadrature Filters, and a Steerable Convolutional Neural Network. We show our experimental results, and compare them with current methods. In Chapter 3, we present another application of our methods on retinal vessel detection. We discuss how to obtain the training annotations specific to our method. Chapter 4 concludes our results.

CHAPTER 2

GUIDEWIRE DETECTION

2.1 Introduction

Guidewire detection is a challenging problem in image-guided intervention. To detect the guidewire one needs to first obtain a low level detection layer that tells how likely is the guidewire to pass through any pixel of the image. As it will be discussed in more detail in the related work section below, guidewire detection work has two main types of approaches to obtain this first level of pixelwise guidewire detection. The first approach is filter based, which uses a predefined filter (Frangi Filter, Steerable Filters, etc.) to obtain a filter response map. The second approach is learning-based, and uses a learning algorithm (Boosting, Random Forest, etc.) together with hand-crafted features (e.g. Haar or rotated Haar) to obtain a per-pixel probability map.

The best performing methods are trained on rotation-aligned samples and search for the maximum response rotation angle at detection time. This is done by rotating the image by a number of angles and applying the classifier to the rotated images. A third approach would be to train a Convolutional Neural Network (CNN) for this purpose, which will learn its own features using the training data. We could not find any work that trains a CNN for detecting the guidewire pixels, which is why it will be investigated in this dissertation.

We applied four methods for guidewire detection, such as the Fully Convolutional Neural Networks (FCNN), trained steerable filters, Spherical Quadrature Filters, and a Steerable Convolutional Neural Network (SCNN). Because the guidewire is thin and hardly visible, it is difficult to train the CNN directly. The loss function becomes flat near the random initialization. We will show how to overcome this issue, through a better initialization obtained from starting the training on a single sequence.

As mentioned, the best performing methods are trained on rotation-aligned samples. We are also interested in training a CNN that is tuned to the guidewire orientation. However, instead of training a CNN on rotation-aligned samples, which would require us to apply it to rotated images for detection, we introduce a Steerable Convolutional Neural Network (SCNN) that can be used to

make it sensitive to parts of the guidewire that have orientation θ. This way, the SCNN eliminates the need to rotate the image by many angles at detection time.

Another guidewire specific issue is that imprecisions in the annotation make the positive examples misaligned. We will also show how to obtain better aligned training data using Spherical Quadrature Filters (SQF) [33] and non-maximum suppression. The Spherical Quadrature Filters (SQF) [33] are a type of steerable filters derived analytically to obtain maximal responses to edge, line or wedge structures. The steerable filters, first introduced in [18], are oriented filters obtained from a basis using predefined weights that depend on the rotation angle. Moreover, the oriented filter response can be computed using the same predefined weights from the response maps obtained by the basis.

The SQFs have been used in [9] for person identification from grayscale images of the ear and in [45] for detecting faint streaks (space debris) in astronomical images. The ear images have edge/ridge structures, and that is why the SQF were a good fit, but the ear images have no noise. In this dissertation, we introduce another potential application of SQF, guidewire detection in fluoroscopy images. As mentioned in Chapter 1, guidewire in X-ray images is in low quality. To our knowledge, we are the first to apply the SQF for this problem.

This dissertation brings the following contributions:

- It shows how to train a Fully Convolutional Neural Network (FCNN) for guidewire detection and how to escape the flat energy landscape present near a random initialization.

- It presents a framework for training steerable filters by loss minimization, and multiple loss functions for this purpose.

- It introduces the Spherical Quadrature Filters (SQF) for guidewire detection, which work better than the popular Frangi filters.

- It introduces a simple model for a Steerable Convolutional Neural Network and shows how to train it for guidewire detection.

- It shows how to address another challenge in training a CNN, which is due to the imprecision in the manual annotation of the thin guidewire. For that, it shows how to use the SQF response map to obtain better aligned examples.

Our experiments reveal that the steerable CNN trained with SQF-aligned examples is the best, followed by the FCNN, and then the SQF and trained steerable filters. Furthermore, all

those methods introduced in this dissertation greatly outperform the existing guidewire detection methods such as the Frangi filters, Steerable filters and a state of the art trained classifier with hand-crafted features.

2.2 Related Work

All guidewire localization methods rely on a first level of pixelwise guidewire detection that applies either a predefined filter or a trained classifier to all locations of the image to obtain a pixelwise guidewire response map.

Filter-based approaches include the Frangi Filter [17], which is based on the sorted eigenvalues (λ_1, λ_2) of the Hessian matrix. It is widely applied to vascular image analysis. The sorted eigenvalues of the Hessian matrix were used to extract and track the guidewire through a spline optimization in [2]. [10] used the Frangi Filter as the data term and fitted the guidewire with a B-spline model in clinical X-ray videos. The beauty of the filter-based approaches consists in their simplicity and interpretability. [5] proposed a method that votes on many candidate curves through all pixels, and the method was compared with the Frangi Filter. Results showed that the Frangi Filter was inferior to the path voting approach.

Steerable Filters have been introduced in [18] for detecting edges and ridges in images. A more recent and powerful type of steerable filters are the Spherical Quadrature Filters (SQF) [33] that were used by [27] for guidewire detection. The SQFs will be introduced in 2.3 in detail.

Learning-based approaches include [7, 3, 35, 50, 24, 41, 11]. In [3], the pixel detection step was trained with examples that were rotated for alignment, using a Probabilistic Boosting Tree (PBT) [47] and Haar features. The trained classifier was applied to rotated images by many angles to obtain the guidewire detection result. A user-constrained algorithm with PBT was proposed in [35] to localize the guidewire. The PBT and hand-crafted features were also employed to track the guidewire in [50] and to detect vessels in [11]. [7] introduced a framework using Boosting and Haar features for catheter detection, and the method was compared with the Frangi Filter. The tracking error results obtained by the learning-based approach were smaller than the results of the filter-based approach. A boosted classifier was used to obtain the low-level detection of the guidewire in [24]. It was trained on ridge and edge features. [12] detected the catheter and vascular structures using a Random Forest classifier of curvilinear structures trained on hand-crafted features. A

method that used the Region Proposal Network to detect the guidewire was presented in [49]. Different from our method which is aimed at obtaining a pixel-wise detection map using CNNs, their work aimed to place bounding boxes around the guidewires.

A Fully Convolutional Neural Network was trained in [27] for guidewire detection. The CNN was invariant to the guidewire orientation, and difficulties in training were reported. In contrast, the steerable CNN is sensitive to the guidewire orientation, alleviating some of the training difficulties and obtaining better detection results.

A steerable CNN theory was presented in [15]. The theory is very generic and only discusses rotations by multiples of 90°, lacking any specific details on how to apply it for steering by arbitrary angles. Moreover, the theory is directed towards invariant models, whereas our steerable CNN obtains models tuned to any orientation, in the spirit of the steerable filters [18]. Furthermore, the rotation angle can be estimated in our method as the angle of maximal response, together with its uncertainty.

Steerable Filter CNNs were developed in [51]. The SFCNNs are both translational and rotational equivariant. The SFCNNs learn the weights of a set of predefined basis of equivariant steerable filters, while our formulation learns a basis that is not necessarily equivariant, but which is made close to equivariant by using a special loss function. Furthermore, the rotation and steering occurs only on the first layer for the SFCNN and is followed by several group-convolutional layers [14]. In our method, each layer is steerable by the same angle θ, making the entire CNN steerable.

Capsules were introduced in [23] and improved in [44]. The capsules represent object detectors together with precise values of the deformations and viewing parameters specific to each object instance. Each capsule is sensitive to a small range of rotation angles and many capsules are needed to cover the entire rotation range. In contrast, our steerable CNN is a single detector that can be rotated to an arbitrary angle, thus it achieves the rotation goal of multiple capsules.

2.3 Overview of the Spherical Quadrature Filters (SQFs)

When training the steerable filters for guidewire detection, we observed that the trained filters resemble the Spherical Quadrature Filters as shown in 2.1. We apply the Spherical Quadrature Filters (SQF) to guidewire detection and compare them with the other methods.

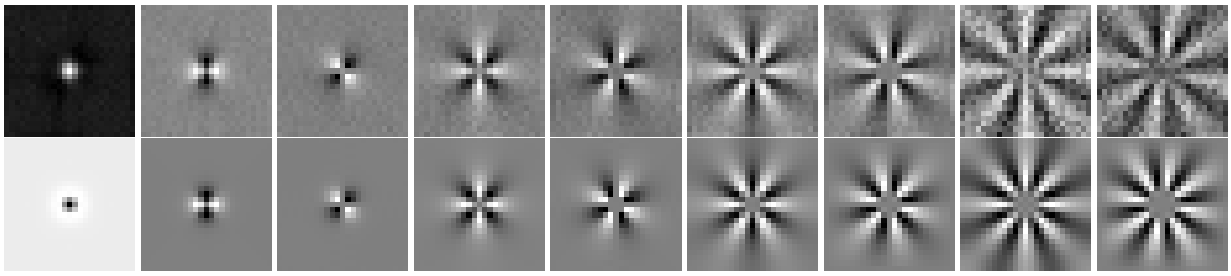

Figure 2.1: Top: Basis B of trained steerable filters of rank 9 with Focal loss[30]. Bottom: Cauchy Spherical Quadrature Filters [33] of order 0,2,4,6,8. Notice the similarity between the trained steerable filters and the analytically derived SQF.

The Spherical Quadrature Filters (SQF) [33] are obtained by the convolution of a generalized Hilbert transform kernel with an isometric filter. The n-th order SQF has the following form in the spatial domain:

$$SQF^{(n)}(x,y) = G(x,y) * \left(\frac{n}{2\pi} \frac{(-(x+\mathrm{i}y))^n}{\|x+\mathrm{i}y\|^{2+n}} \right) \tag{2.1}$$

where "$*$" denotes convolution, and in the Fourier domain

$$\hat{SQF}^{(n)}(\mathbf{u}) = \hat{G}(\mathbf{u}) \cdot \left(i \frac{\mathbf{u}}{\|\mathbf{u}\|} \right)^n \tag{2.2}$$

where $x, y \in \mathbb{R}, \mathbf{u} \in \mathbb{R}^2, n \in \mathbb{N}^*$ and $G(x,y)$ is a bandpass isometric filter. In this dissertation we will generate the SQFs using bandpass filters such as the log-Gabor filters [6]

$$G_l(\omega) = n_c \exp\left(-\frac{\log^2(\omega/\omega_0)}{2\log^2(\sigma)} \right), \tag{2.3}$$

Gaussian derivative filters [6]

$$G_d(\omega) = \begin{cases} n_c \omega^{(\omega_0\sigma)^2} \exp(-(\sigma\omega)^2), & \text{if } \omega \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

and Cauchy filters [6]

$$G_{Cauchy}(\omega) = \begin{cases} n_c \omega^{\omega_0\sigma} \exp(-\sigma\omega), & \text{if } \omega \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

where $\omega_0 \in \mathbb{R}$ is the peak tuning frequency, and $\sigma \in \mathbb{R}$ such that $\omega_0\sigma \geq 1$. Eq. (2.3), (2.4) and (2.5) are defined in the frequency domain. For more details see [33].

Observe that except for the order 0 SQF, the higher order SQFs come in pairs as the real and imaginary part of eq. (2.1) or (2.2). For ridge detection, we only need the even order (symmetric)

SQFs, and we will use all the even SQFs of order $n < r$ where r is an odd number. Observe that for any odd number $r > 0$ there are exactly r SQFs of even order $n < r$, and we will call them the SQF of rank r.

The SQF bank of rank r can be steered to an angle θ by dot product multiplication with the following weight vector:

$$\mathbf{w}(\theta) = [-1, -\cos(2\theta_k), \sin(2\theta_k), ..., -\cos(2r\theta_k), \sin(2r\theta_k)]. \tag{2.6}$$

Figure 2.2: Steered Cauchy Spherical Quadrature Filters of rank 11

An example of steered Cauchy SQFs of rank 11 is shown in Figure 2.2.

Examples of filter response maps obtained using the SQFs and non-maximal suppression are shown in Figure 2.6.

2.4 Fully Convolutional Networks (FCN)

Fully Convolutional Networks are convolutional neural networks that are composed of convolutional layers without any fully-connected layers. In [32], the author showed how to train an FCNN end-to-end for pixel-wise prediction. Fully Convolutional Networks can be applied on an input of any size.

Assume we are given n training patches $(\mathbf{x}_i, y_i), i = 1, ..., n$ where $\mathbf{x}_i \in \mathbb{R}^{p^2}$ is the image of a patch of size $p \times p$ either centered on the guidewire (a positive example) or away from the guidewire (a negative example), and y_i is the label. The labels are $y_i = -1$ for negative patches and $y_i = 1$ for patches centered on the guidewire.

2.4.1 CNN Architecture

We implemented a Fully Convolutional Neural Network for guidewire detection. The network (Figure 2.3) is composed of 5 convolutional layers, the first three layers are followed by 2×2 max-pooling with stride 1, while the fourth layer is followed by ReLU (Rectified Linear Unit). The last convolutional layer obtains the binary guidewire/non-guidewire response.

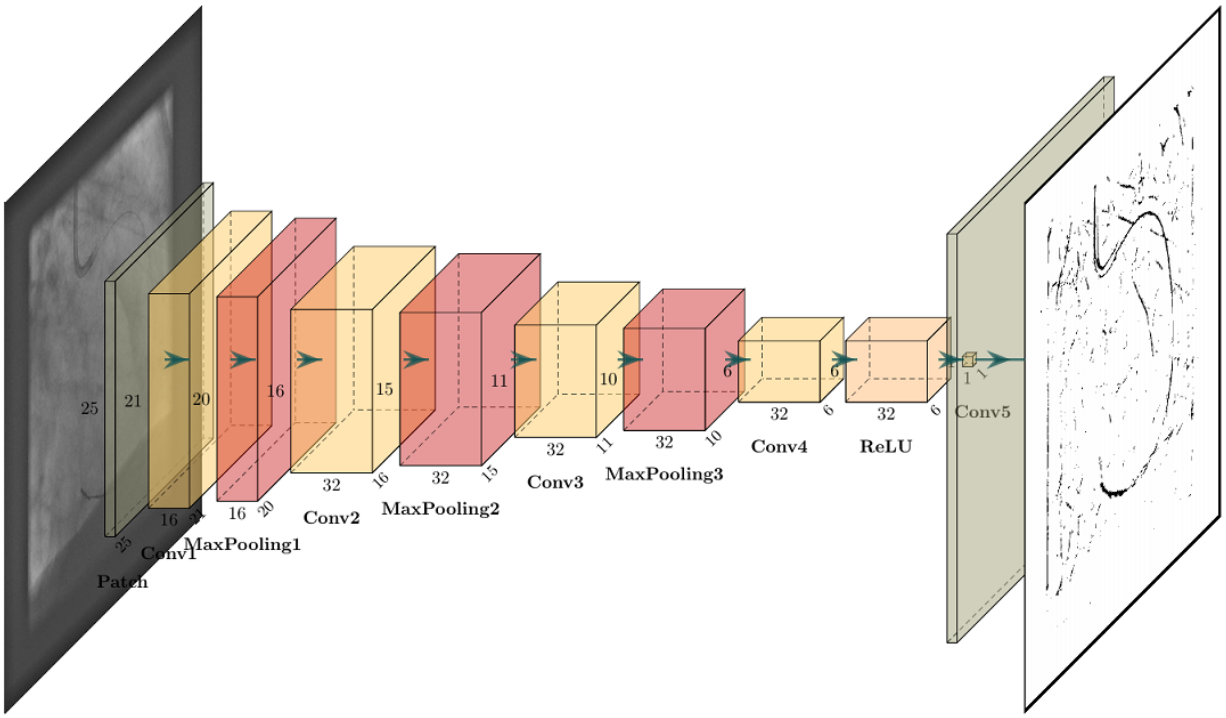

Figure 2.3: The Fully Convolutional Neural Network (FCNN) used in this dissertation with input patches of size 25×25.

We first trained a FCNN based on patches of size 15×15. The first convolutional layer has 16 filters of size 3×3, and the next three layers have 32 filters of size 3×3 each. The last layer is 4×4. The experiment shows that the FCNN trained on 25×25 patches performs better than the one trained on 15×15. For a receptive field of size 25×25, the first convolutional layer has 16 filters of size 5×5, and the next three layers have 32 filters of size 5×5 each. The last layer is 6×6.

We also implemented a FCNN composed of 4 convolutional layers for patches of size 25×25 and 33×33. During the experiment, we noticed that the inclusion or exclusion of the max-pooling layers did not affect the performance of the network. An FCNN architecture without Maxpooling Layers is shown in 2.4. For a receptive field of size 25×25, only the third convolutional layer is followed by a ReLU layer. The first convolutional layer has 16 filters of size 7×7, the following layers have 32 filters of size 7×7. The last one contains 1 filter of size 7×7. For a receptive field of size 33×33, it has the same framework as the 25×25, except the sizes of the filters are 8×8. All the convolutional layers have no padding and a stride $S = 1$.

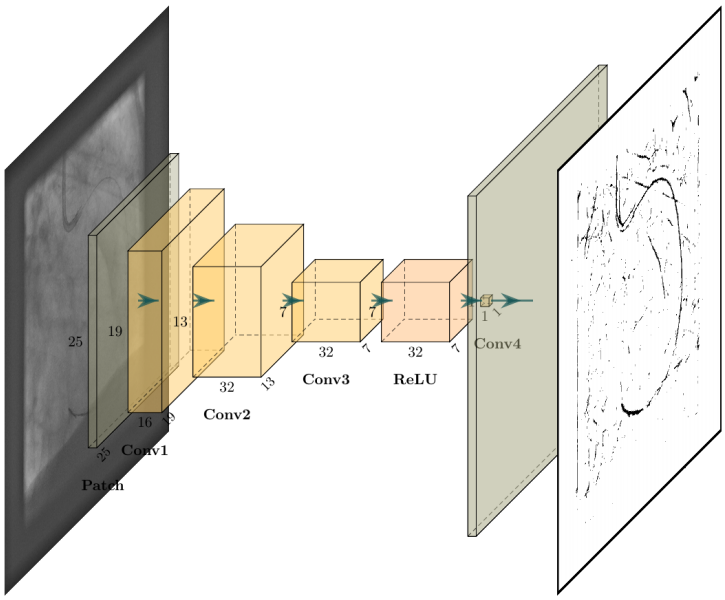

Figure 2.4: The FCNN used in this dissertation without Maxpooling Layers of input patch of size 25×25.

Loss Function. Denoting all the parameters of the CNN as W, the loss for training the CNN $f_W(\mathbf{x})$ is

$$L(W) = \sum_{i=1}^{n} \ell(y_i f_W(\mathbf{x}_i)) \tag{2.7}$$

where n denotes the number of examples and ℓ denotes a per example loss function. In terms of per example loss function, one can use the Lorenz loss [4]

$$\ell(u) = \log(1 + \max(1 - u, 0)^2) \tag{2.8}$$

due to its ease of training and robustness to outliers, the Logistic loss

$$\ell(u) = \log(1 + \exp(-u)), \tag{2.9}$$

or the Focal loss [30] as a comparison. The Focal loss can deal with the class imbalance and can adjust the weight for both easy and hard examples.

It has the form

$$\ell(u) = -\alpha_t (1 - p_t(u))^\gamma \log(p_t(u)). \tag{2.10}$$

where

$$p_t(u) = \begin{cases} p(u), & \text{if } y = 1 \\ 1 - p(u), & y = -1 \end{cases}, \tag{2.11}$$

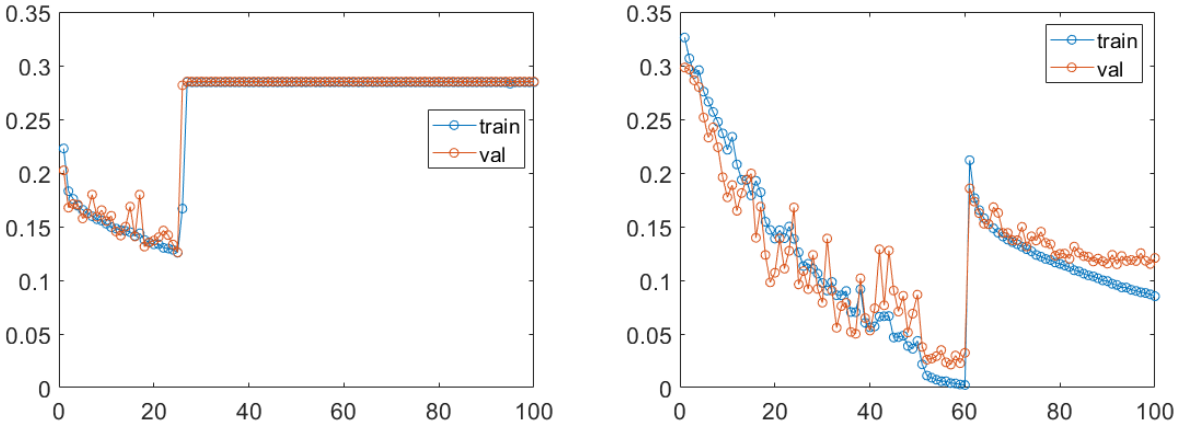

Figure 2.5: The plot of Lorenz loss function(Eq 2.8) for 100 epochs of a patch size 25×25. Top left: training loss with all training examples. Top right: training loss with our approach.

$\alpha_t = 0.25$ for class 1, $\gamma = 2$ and p is obtained by using sigmoid function

$$p(u) = \frac{1}{1 + \exp(-u)}. \tag{2.12}$$

2.4.2 Training Examples

The guidewire annotations are obtained using B-spline. As positive examples we used image patches at distance at most 1 pixel from the annotation, while negative examples were at distance at least 8 pixels from annotation.

2.4.3 Training Initialization

All weights were initialized with random Gaussian values with std 0.01. The training was performed using the Adam optimizer[25] with the initial learning rate 0.00001 and an initial mini-batch size 32. The learning rate was multiplied by 0.8 and the minibatch was doubled every 50 epochs, for a total of 300 epochs.

Training the FCNN directly from a random initialization does not work because the guidewire is very thin and the energy landscape becomes flat near the random initialization. Indeed, as shown on the left side of Figure 2.5, the loss becomes flat at around 0.285 after epoch 26. In this case, results show that every pixel of the response map is considered detected. To overcome this problem

we started by training the first 40-60 epochs using the training examples from only one sequence as shown on the right side of Figure 2.5. After that, training was done on all training examples.

2.5 NMS-based Alignment

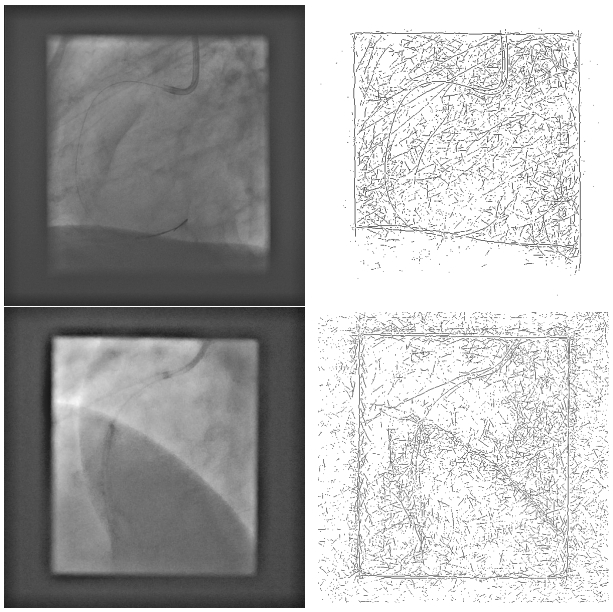

Figure 2.6: SQF NMS map of two frames: input images(first column), SQF NMS images(second column).

There are two main issues during training the Convolutional Neural Network. First, the training doesn't work directly from a random initialization because the guidewire is very thin with only one to two pixels wide. We proposed how to overcome this difficulty in Section 2.4.

Another issue we observed was that the annotation was not precise enough to obtain a good alignment of the positive examples. As a result, the false positive rate, while quite good, was still rather high. To obtain a better alignment we used the Cauchy rank 11 SQF maximum response map on which we performed non-maximal suppression (NMS) in the direction of the image gradients. An example of the SQF NMS map is shown in Figure 2.6. Then we used as training examples only the patches centered on the NMS response map.

In Figure 2.7 are shown the training loss functions for the FCNN illustrated in Section 2.4 without max-pooling layers, trained on regular examples (left) and NMS-aligned examples (right).

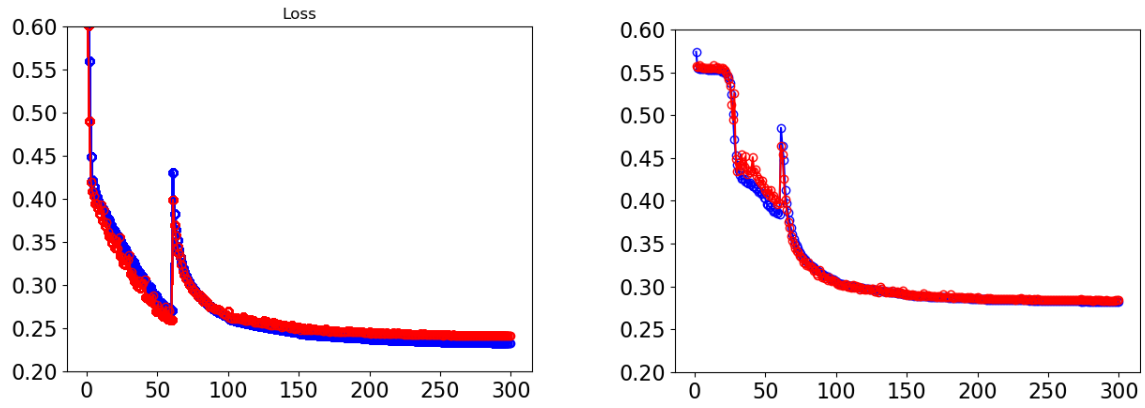

Figure 2.7: The Soft Margin loss function (Eq 2.9) of the FCNN. Left: training loss without NMS-aligned examples. Right: training with NMS-aligned examples.

2.6 Trained Steerable Filters(TSF)

The training patches are similar to the Fully Convolutional Neural Network except the labels are $y_i = k \in \{1, ..., K\}$ for patches centered on the guidewire (positive examples) where the tangent angle is in the interval $[(k-1)\pi/K, k\pi/K)$. The labels for the negative patches are $y_i = -1$. In our experiments, K is set to 30.

2.6.1 Trainable Filters

We are interested in training filters $\mathbf{f}_i \in \mathbb{R}^{p^2}, i = 1, ..., K$ that achieve both a good detection of the guidewire points and estimation of the guidewire direction. These filters can be organized as columns in a filter bank of size $p^2 \times K$ matrix $F = (\mathbf{f}_1, ..., \mathbf{f}_K)$.

Given an image patch $\mathbf{x} \in \mathbb{R}^{p^2}$ we compute the K filter responses $(r_1, ..., r_K) = \mathbf{x}^T F$ and obtain the classifier $f_F(\mathbf{x}) = \max(\mathbf{x}^T F)$ and the angle prediction $a_F(\mathbf{x}) = \text{argmax}(\mathbf{x}^T F)$. For example in Figure 2.8, the classifier response is about 13 with an angle prediction 60°.

The training loss function aims to find filters that achieve good separation between positive and negative examples and a small error in predicting the correct angle.

One of the simplest loss functions that we will study is the *foreground-background* (FB) loss, which encourages the correct class responses of the positives to be high and all responses for the negatives to be low. It also equalizes the total weight for the positives corresponding to each angle

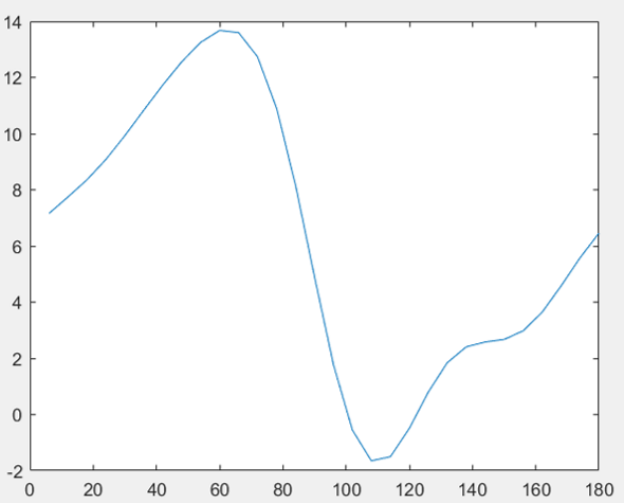

Figure 2.8: Example of classifier response and angle prediction of one patch.

and has the form

$$L^{FB}(F) = \frac{1}{n_{-1}} \sum_{i,y_i=-1} \sum_{k=1}^{K} \ell(-\mathbf{x}_i^T \mathbf{f}_k) + \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,y_i=k} \ell(\mathbf{x}_i^T \mathbf{f}_k) \tag{2.13}$$

where $n_k = |\{i, y_i = k\}|$ and $\ell(u)$ is a per example loss function such as the logistic loss $\ell(u) = \log(1 + \exp(-u))$, hinge loss $\ell(u) = \max(1 - u, 0)$, the Lorenz loss [4] in Eq. 2.8 or the Focal loss [30] (that assumes only labels 1 and -1) as described in Eq. 2.10.

The FB loss function is very well behaved and it is convex when $\ell(u)$ is convex (such as the logistic or the hinge loss).

To discourage high responses for other angle filters than the true angle for positive examples we can add to the FB loss above the following Vapnik loss [48]:

$$L^{V}(F) = \frac{1}{n+} \sum_{i,y_i>0} \sum_{k \neq y_i} \ell(\mathbf{x}_i^T \mathbf{f}_{y_i} - \mathbf{x}_i^T \mathbf{f}_k) \tag{2.14}$$

where $n_+ = |\{i, y_i > 0\}|$.

The *Max loss* aims to directly maximize the margin of the classifier $f_F(\mathbf{x}) = \max(\mathbf{x}^T F)$:

$$L^{M}(F) = \frac{1}{n_{-1}} \sum_{i,y_i=-1} \ell(-f_F(\mathbf{x}_i)) + \frac{1}{n_+} \sum_{i,y_i>0} \ell(f_F(\mathbf{x}_i)) \tag{2.15}$$

The max loss has many local optima which are not equivalent. We observed that we can obtain better results by initializing the optimization with the results obtained by the FB loss as opposed to a random initialization.

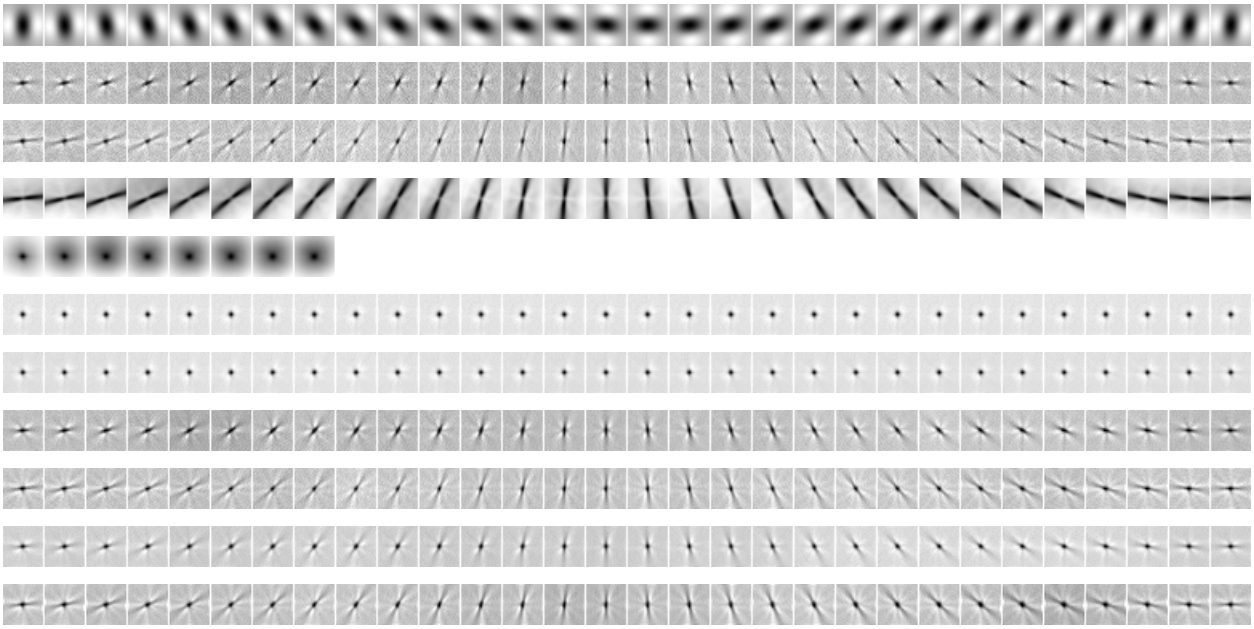

Figure 2.9: Filters, from top to bottom, one row each: steerable filters, trained FB filters, trained FB filters with NMS patches, trained FB+Vapnik filters, trained Max filters (initialized with zero, randomly, with FB result respectively), trained steerable rank 11 filters(Lorenz loss, Lorenz loss with NMS patches, Focal loss, and Focal loss with NMS patches respectively).

One could directly train a filter bank F using these losses or one could obtain a low rank representation of the filter bank using steerable filters as described below. Examples of trained filters are shown in Figure 2.9.

2.6.2 Trainable Steerable Filters

We are interested in deriving a steerable representation for the filter bank $F = (\mathbf{f}_1, ..., \mathbf{f}_K)$. For that, we start with the steerable filter [18] of order 2, which for an angle θ is defined by

$$G_2^\theta = B \cdot \mathbf{w}_\theta \tag{2.16}$$

where $\mathbf{w}_\theta = (\cos^2\theta, -2\cos\theta\sin\theta, \sin^2\theta)^T$,

$$B = \frac{G}{\sigma^4} \cdot (x^2 - \sigma^2, -xy, y^2 - \sigma^2). \tag{2.17}$$

and G is the 2-D Gaussian with variance σ^2.

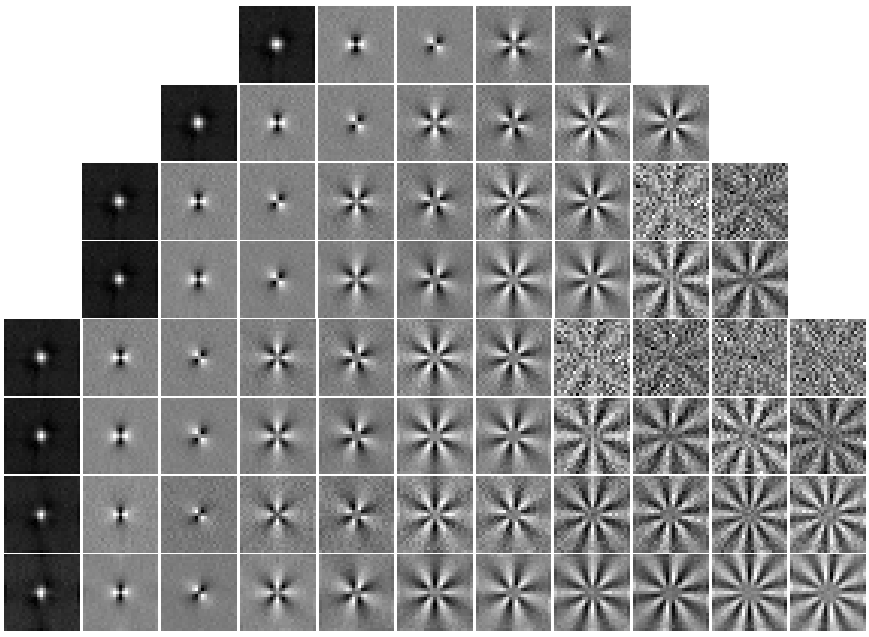

Figure 2.10: Trained basis B of steerable filters for $d = 2$ to 5 (ranks 5 to 11) with per example loss Lorenz loss(row 1, 2, 3, 5, 7) and Focal loss(row 4, 6, 8), trained with SQF + NMS patches(last 2 rows).

This inspires us to represent the filters as $\mathbf{f}_k = B \cdot \mathbf{w}_k$ which is a linear combination of some unknown $p^2 \times (d+1)$ matrix B that needs to be learned and a weight vector

$$\mathbf{w}_k = (\cos^d \theta_k, \sin^{d-1} \theta_k \cos \theta_k, ..., \sin^d \theta_k)^T. \tag{2.18}$$

However, higher powers d result in numerical instability, and since the even powers of the sin and cos are related to the sin and cos of the angle multiples, we propose an alternate steerable representation $\mathbf{f}_k = B \cdot \mathbf{a}_k$ with

$$\mathbf{a}_k = [1, -\cos(2\theta_k), \sin(2\theta_k), ..., -\cos(2d\theta_k), \sin(2d\theta_k)]^T. \tag{2.19}$$

Denoting by $A = (\mathbf{a}_1, ..., \mathbf{a}_K)$, we obtain $F = BA$ with A being a fixed and known matrix. This is a rank $2d+1$ steerable representation of F. Examples of trained B for $d = 1$ to 5 are shown in Figure 2.10.

This representation can be used in Eq (2.13) to obtain the loss for training the steerable filters:

$$L^S(B) = \frac{1}{n_{-1}} \sum_{i, y_i = -1} \sum_{k=1}^{K} \ell(-\mathbf{x}_i^T B \mathbf{a}_k) + \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i, y_i = k} \ell(\mathbf{x}_i^T B \mathbf{a}_k). \tag{2.20}$$

When training with SQF NMS-aligned examples, we are given training examples $(\mathbf{x}_i, y_i, k_i)$, where $y_i \in \{-1, 1\}$ and k_i is the angle index for each observation, given as the angle of maximum SQF response. In this case the FB loss has the following form:

$$L^S_{NMS}(B) = \frac{1}{n_{-1}} \sum_{i,y_i=-1} \ell(-\mathbf{x}_i^T B \mathbf{a}_{k_i}) + \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,y_i=1,k_i=k} \ell(\mathbf{x}_i^T B \mathbf{a}_k). \tag{2.21}$$

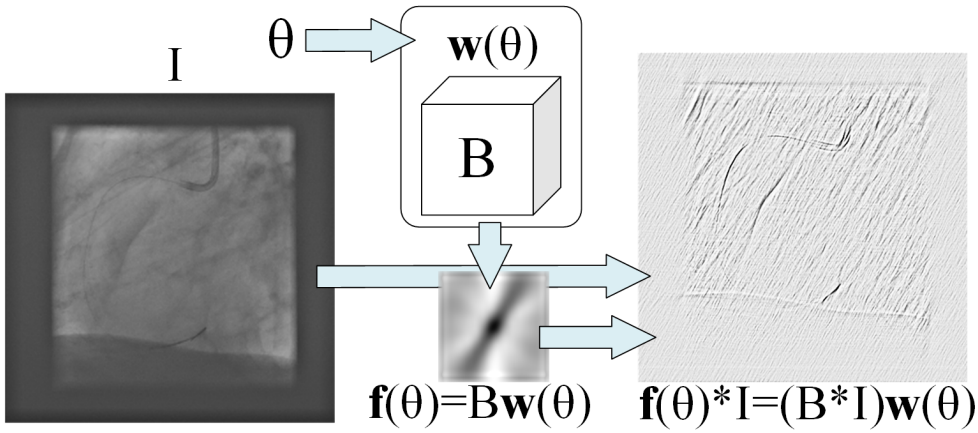

Figure 2.11: Diagram of the steerable convolution filters.

2.7 Steerable Convolutional Neural Network

As we saw, the steerable filters are oriented filters that are obtained as a linear combination from a basis B, and can be rotated by a simple re-weighting of the basis as illustrated in Figure 2.11.

2.7.1 The Steerable CNN

The steerable CNN, illustrated in Figure 2.12, consists of a number of layers with multiple steerable filters.

If the basis of each steerable filter contains r filters, then a layer with k steerable filters will contain $r \cdot k$ filters in total, grouped in k groups of r filters. The response maps of that layer for any angle θ can be obtained by convolution with all the rk filters, followed by linearly combining the k responses corresponding to each group using the weight vector $\mathbf{w}(\theta)$ from Eq. (2.6).

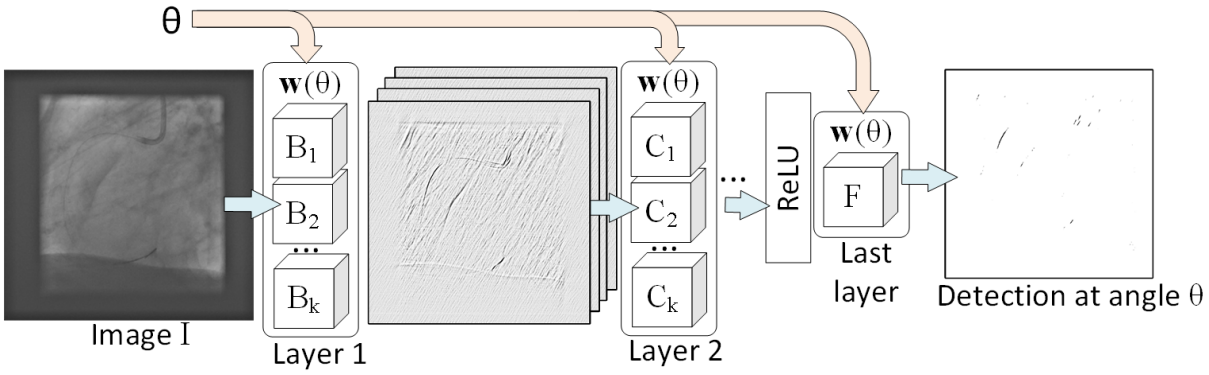

Figure 2.12: Diagram of the steerable CNN, steered (tuned) to an angle θ.

2.7.2 Training the Steerable CNN

Assume we are given n training patches $(\mathbf{x}_i, y_i, \alpha_i), i = 1, ..., n$ where $\mathbf{x}_i \in \mathbb{R}^{p^2}$ is the image of a patch of size $p \times p$ either centered on the guidewire (a positive example) or away from the guidewire (a negative), $y_i \in \{-1, 1\}$ is the label, and $\alpha_i \in [0, \pi)$ is the orientation. The orientation at the center location of each patch is obtained by a Steerable Quadrature Filter (SQF) [33]. The SQF is

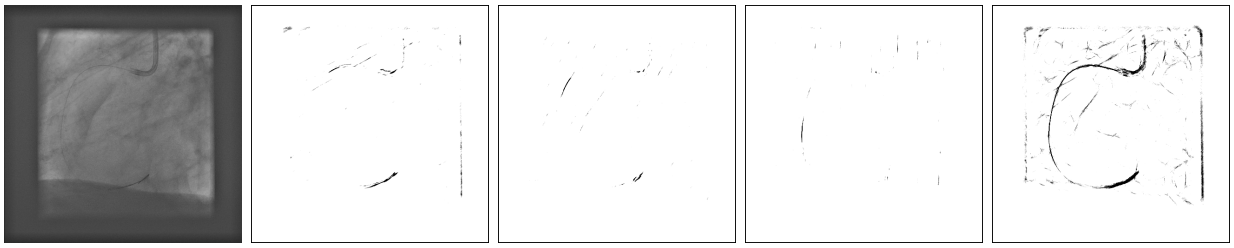

Figure 2.13: Examples of one frame Steerable CNN detection results: input image, response map for angle index $j = 5, 10, 15$, and final detection result.

also used as a preprocessing step for detection, so the training examples are extracted only from locations with high SQF responses (NMS-based examples). This way the angle information α_i for each training patch has a reliable value. Alternatively, the steerable CNN can be applied for a number of discrete angles and the maximum response can be used as detection map, as illustrated in Figure 2.13.

Similar to Section 2.6.2, the range $[0, \pi)$ is discretized (modulo π) into a number of equally spaced angle bins $b_j = [\theta_j - \frac{\pi}{2K}, \theta_j + \frac{\pi}{2K}), j \in \{1, ..., K\}$, where $\theta_j = \frac{j\pi}{K}$ (in this dissertation we

used $K = 30$ angle bins). Then the orientation angles α_i of the training examples are converted to angle bin indices $a_i \in \{1, ..., K\}$ and the examples with the same angle index j are collected into the set $S_j = \{(x_i, y_i, a_i), a_i = j\}$. For simplicity, we assume that all angles are equally represented, so $|S_j| = |S_k|, \forall j, k \in \{1, ..., K\}$.

Training is done using the Adam optimizer [25]. For each minibatch, an angle index $j \in \{1, ..., K\}$ is chosen and only examples with $a_i = j$ are selected, so they have approximately the same angle θ_j, the center of the bin b_j. In this case, the examples share the same weight vector $\mathbf{w}_j = \mathbf{w}(\theta_j)$ from Eq. (2.19) and the network is equivalent to a CNN where each convolution layer is followed by a linear layer that takes each group of k responses and combines them linearly with weights $\mathbf{w}_j$. One epoch of the training is described in Algorithm 1 below. An example of the trained basis B of the first layer in the rank 11 steerable CNN is show in Figure 2.14. The filters appear like a set of random filters in a 4-layer SCNN. As a comparison, steerable filter of basis in a 1-layer SCNN and the steerable filters in the first layer of a 2-layer SCNN are shown in Figure 2.15 and Figure 2.16. The steered filters are shown in Figure 2.17.

Algorithm 1 One epoch of **Steerable CNN Training**

Input: Training patches $\{(\mathbf{x}_i, y_i, a_i)\}_{i=1}^N$, minibatch size m

Output: Trained steerable CNN.

Set $N^{batch} = \lfloor |S_j|/m \rfloor$.
for j = 1 to K **do**
  Shuffle the set S_j.
end for
for b = 1 to N^{batch} **do**
  for j = 1 to K **do**
    Set the steerable CNN angle $\theta = \alpha_j$, so $\mathbf{w}(\theta) = \mathbf{w}_j$
    Use the b-th minibatch from S_j to update the weights by backpropagation.
  end for
end for

2.7.3 Implementation Details

CNN Architecture. A 1-layer Steerable CNN contains only one steerable convolutional layer, it returns the guidewire/non-guidewire response directly. A 2-layer Steerable CNN are also trained which consists of 2 steerable convolutional layers. The first convolutional layer is followed

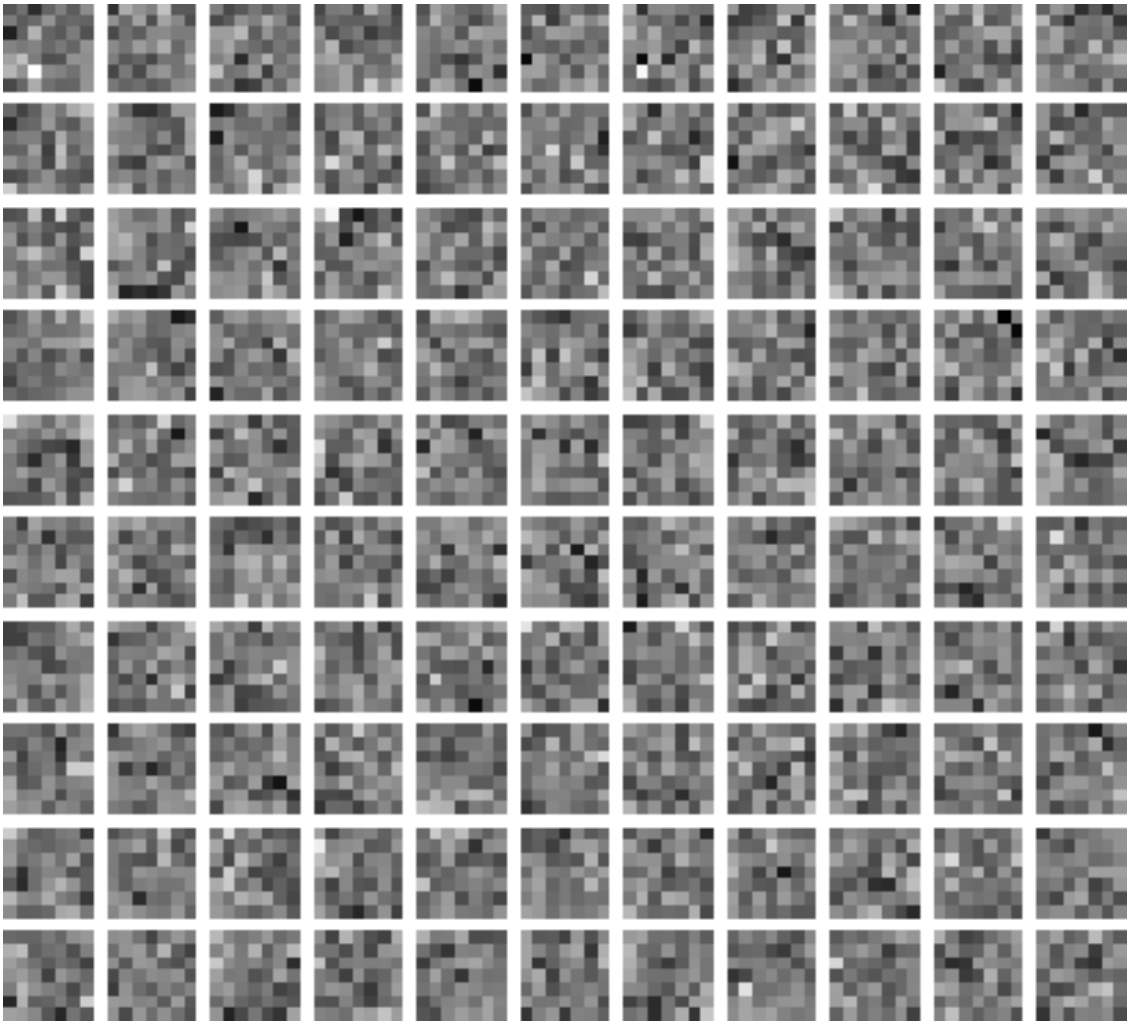

Figure 2.14: Trained basis B for the first layer of the rank 11 4-layer SCNN.

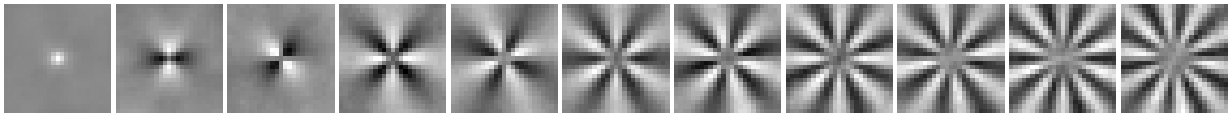

Figure 2.15: Trained basis B for 1-layer SCNN.

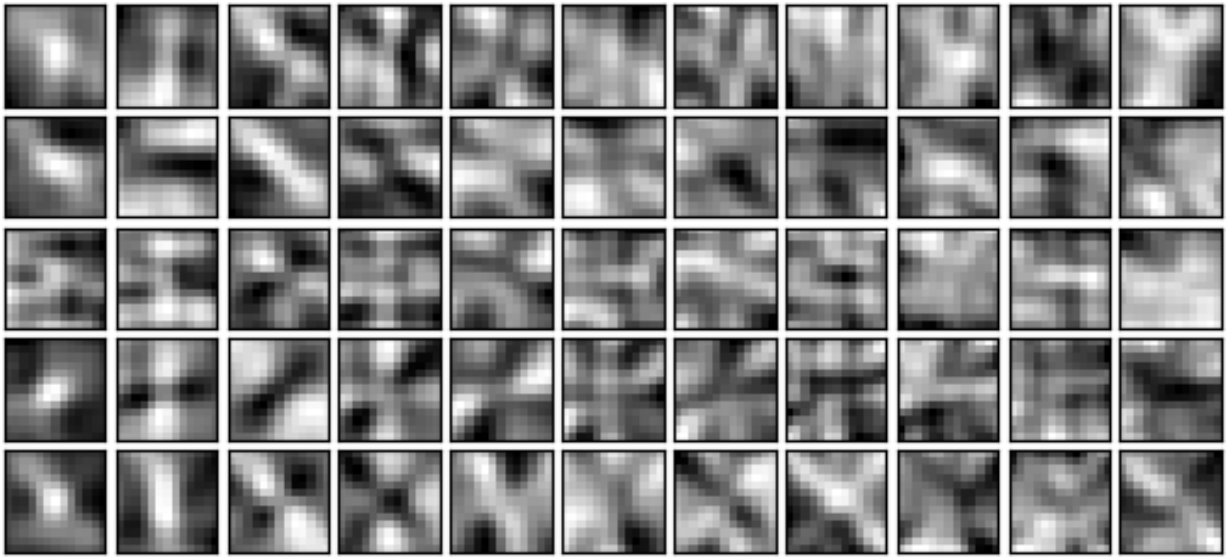

Figure 2.16: Trained basis B for the first layer of 2-layer SCNN.

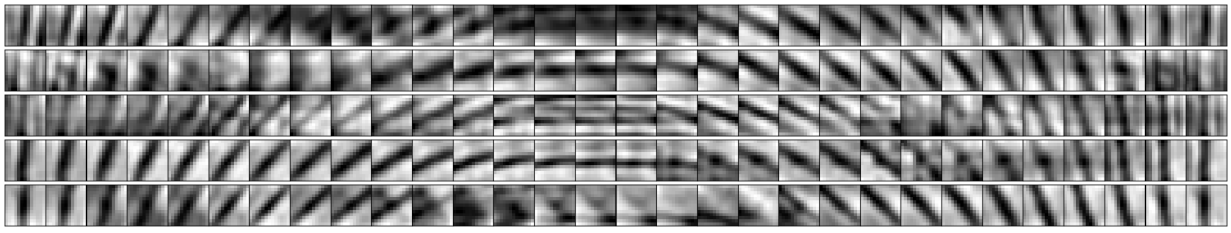

Figure 2.17: Steered filters for the first layer of 2-layer SCNN

by a ReLU layer, and the second one gives the response. A 4-layer Steerable CNN for this task consists of 4 steerable convolutional layers. The third steerable convolutional layers is followed by ReLU activation, and the last one returns the response.

For the 1-layer SCNN, the filters are of size 25×25. The layer has 1 steerable filter. With a basis of rank $r = 11$, there are 11 filters in the layer. The steerable filters of a 2-layer SCNN are of size 13×13. For a basis of rank $r = 11$, there are 5 steerable filters in the first layer (55 filters in total) and 1 steerable filter in the second layer. The 4-layer SCNN steerable filters are of size 7×7, with their basis containing $r = 7$ filters or $r = 11$ filters. The first layer has 10 steerable filters (thus the layer has 70 or 110 total filters), the second and the third one have 20 steerable filters and the last one has 1 steerable filter. It has a receptive field of size 25×25.

For both the FCNN and the steerable CNN we used the Pytorch[40] Soft Margin Loss in Eq 2.9 to guide the training.

Training Initialization. All weights were initialized with either 0 or random normal values with std 0.01. For the Steerable CNN, the training was performed using the Adam optimizer[25] with the initial learning rate 3×10^{-7}, with mini-batch size 32, and with L_2 regularization $\lambda = 0.001$. The learning rate was multiplied by 0.8 and the minibatch was doubled every 50 epochs, for a total of 300 epochs.

For training the Steerable CNN, we start training the first 40 epochs using all positive patches, and the negative patches are subsampled 40%. After that, training was done on all negative examples.

In Figure 2.18 are shown the training loss functions for the steerable CNNs of rank 7 and 11 with Soft Margin loss(Eq. 2.9) and Focal loss [30].

2.8 Experiments

2.8.1 Dataset

Experiments are conducted on a dataset of 75 fluoroscopic sequences with a total of 826 frames of different sizes in the range $[512, 960] \times [512, 1024]$. The sequences were divided into a training set containing 39 sequences with 424 frames and a test set containing 36 sequences with 402 frames. The guidewire was manually annotated in all the frames using B-splines. One example of the spline result, positive mask and negative mask are shown in Figure 2.19.

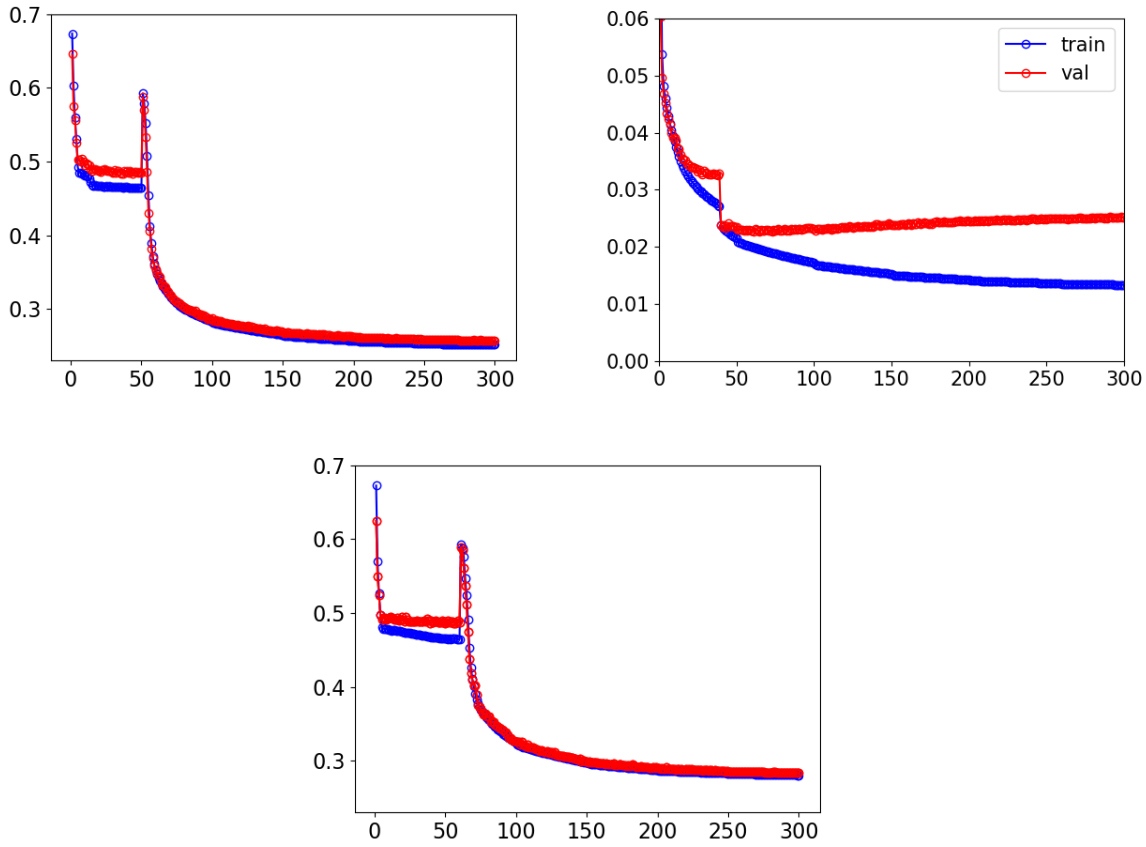

Figure 2.18: The different loss functions of the Steerable CNN during training. Top left: Rank 11 steerable CNN training started with Soft Margin loss function(Eq 2.9). Top right: Rank 11 steerable CNN training started with Focal loss function(Eq 2.10). Bottom: Rank 7 steerable CNN training started with Soft Margin loss function(Eq 2.9).

2.8.2 Training Details for FCNN and SCNN

For training the FCNN in Section 2.4, we used positive and negative patches of size 15×15 or 25×25. We extracted positive patches centered on the guidewire, and also extracted negative patches that are at a distance between 8 and 30 pixels from the guidewire, subsampled to 5%. In both cases the training set contains about 213,000 positives and about twice as many negatives. The training set using NMS alignment contains about 91,000 positives and as about twice as many negatives.

For training the FCNN and SCNN in Section 2.7, we used positive and negative patches of size

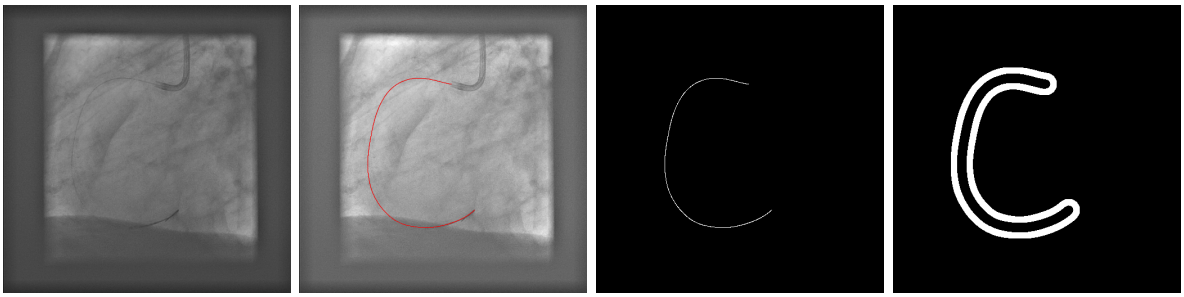

Figure 2.19: From left to right: original frame, annotated using B-splines, positive mask and negative mask.

25×25. For the training set without SQF NMS alignment we extracted positive patches centered on the guidewire, and negative patches at a distance between 8 and 30 pixels from the guidewire, subsampled to 5%. The SQF NMS aligned training examples were extracted from locations that were detected by the SQF with NMS described 2.5. The positives patches were centered at distance at most 2 from the guidewire annotation and the negatives at distance at least 5, subsampled to 5%. The training set without SQF NMS alignment contains about 279,000 positives and 578,000 negatives. The training set with SQF NMS alignment contains about 193,000 positives and 535,000 negatives.

2.8.3 Training Details for Trained Steerable Filters

We used the positive and negative patches of size 25×25 for training and evaluation of the detection performance in terms of area under the ROC curve and angle estimation error. We extract the positive patches centered on the guidewire with a label corresponding to the angle of the tangent to the guidewire, discretized in 30 possible values. We trained the loss functions and the trainable steerable filters by using the same examples as mentioned in 2.8.2 (both non-NMS-aligned examples and NMS-aligned examples).

The filters were trained using the BFGS optimization algorithm [8], with maximum $2,000$ iterations, initialized with all filters having value 0 unless otherwise specified. Since the loss gradient could be computed analytically, training of each filter bank took between 20 and 40 minutes on a Matlab implementation using the built-in `fminunc` optimization procedure. Examples of trained filters are shown in Figure 2.9. The basis B of the trained steerable filters of ranks 5 to 11 are shown in Figure 2.10.

Table 2.1: Evaluation of the different filters on 25×25 patches.

	AUC		Angle Error	
Method	Train	Test	Train	Test
Frangi Filter [17]	0.7366	0.7296	13.16	13.69
Steerable Filter [18]	0.6348	0.6418	22.25	24.55
SQF [33] log-Gabor, $f_0 = 1/6$, rk. 11	0.8340	0.8251	9.04	9.90
Lorenz FB (Eq. 2.13)	0.8951	0.8730	6.14	7.23
Lorenz FB (Eq. 2.13)(w/ NMS)	0.8618	0.8295	2.71	3.24
Lorenz FB (Eq. 2.13) + Vapnik 2.14	0.7530	0.7286	18.83	20.55
Lorenz Max (Eq. 2.15), zero init.	0.7773	0.7566	44.34	45.40
Lorenz Max (Eq. 2.15), random init.	0.8654	0.8362	46.54	49.68
Lorenz Max (Eq. 2.15), FB init.	0.8652	0.8364	19.53	21.47
Steerable (Eq. 2.20), rk. 5	0.8925	0.8719	8.89	9.72
Steerable (Eq. 2.20), rk. 7	0.8956	0.8755	4.49	5.20
Steerable (Eq. 2.20), rk. 9	0.8966	0.8764	4.85	5.69
Steerable (Eq. 2.20), rk. 11	0.8971	0.8766	5.12	6.02
Steerable (Eq. 2.20)(w/ NMS), rk. 11	0.8622	0.8305	**2.14**	**2.42**
Steerable (Eq. 2.20)(Focal[30]) rk. 11	**0.8991**	**0.8776**	6.62	7.51
Steerable (Eq. 2.20)(Focal[30] w/ NMS) rk. 11	0.8682	0.8353	2.63	2.78
Steerable (Eq. 2.20), rank 13	0.8972	0.8764	12.45	14.42

2.8.4 Patch Based Evaluation

On the training and test datasets and using the optimization described in Section 2.8.3, we evaluated the FB loss (2.13), the FB+Vapnik loss (2.14), the Max loss (2.15), all using the Lorenz loss (2.8). The Max loss was trained using different initializations: a zero initialization, a random initialization, and initialization with the filters obtained by the FB loss. The steerable filter loss (2.20) was trained with B of ranks 5 to 13. We also evaluated the Frangi filter [17], Steerable filter [18] and the Spherical Quadrature filters [33].

The training and test AUCs for these methods are shown in Table 2.1. We see that indeed the Max loss result does depend on initialization and FB init. performs better than the zero and randomly initialization. We also see that the trained loss and trained steerable filters with Focal loss [30] have better train and test AUC than the other methods.

We also show in Table 2.1 the angle estimation error as the average difference between the true angle (from 0 to 180°) and the estimated angle modulo 180°. We see that the methods with NMS examples have smaller angle error of all filters. The trained steerable filters with rank 11 have the lowest angle error on both training and test set.

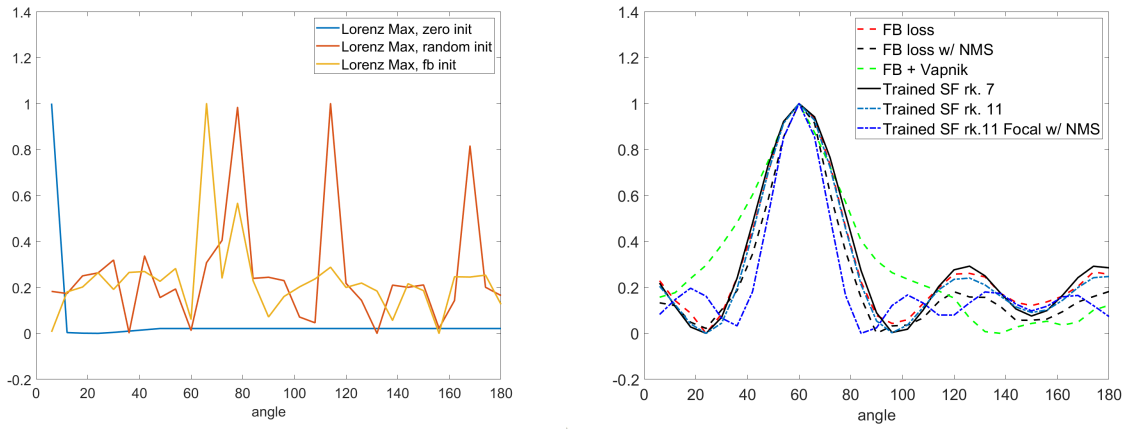

Figure 2.20: Average responses for test patches with angle 60°.

In Figure 2.20 are shown the average responses for test (unseen) patches from class $k = 10$, corresponding to angles around 60°. On the right plot are shown the average responses of the filters trained with the FB loss, FB loss w/ NMS patches, FB + Vapnik loss, steerable FB loss of rank 7 and rank 11 with Lorenz loss and rank 11(Focal loss[30]) with NMS patches. All these averages have a strong maximum at the true angle, with the average response of the steerable filters of rank 11 with Focal loss being the smoothest of all.

2.8.5 Comparison with Other Methods

We also present an evaluation of the pixelwise guidewire detection results on the entire images instead of patches for filter based methods, learning based methods and Convolutional Neural Networks. As filter based methods we evaluated the popular Frangi Filter [17], Steerable Filters [18], and the steerable Spherical Quadrature Filters (SQF) [33] with different types of isometric filters and ranks (dimension of the basis). As learning based approach, we implemented the trained steerable filters and also the approach from [11] based on about 100,000 oriented Haar features and a Probabilistic Boosting Tree (PBT) [47] and trained it on the same data. As CNN approaches, we evaluated the SFCNNs [51] filters on the NMS patches. We compare the steerable CNN, the FCNN, and we also trained a 25×25 steerable filter, which can be considered as using only the last layer of the steerable CNN, with a larger filter size.

Table 2.2: Per-image evaluation of different filter based and training based guidewire detection methods

Method	Det. rate Train	Det. rate Test	FP rate Train	FP rate Test	# of trained parameters
Frangi Filter [17]	90.44	90.44	26.99	24.19	-
Steerable Filters [18]	90.06	89.93	67.07	76.48	-
SQF [33] Gauss deriv, f_0=1/2, rk. 7	90.00	90.08	6.92	7.04	-
SQF [33] Gauss deriv, f_0=1/2, rk. 9	89.97	90.09	6.86	6.98	-
SQF [33] Cauchy, $f_0 = 1/6$, rank 7	89.98	90.00	5.98	6.35	-
SQF [33] Cauchy, $f_0 = 1/6$, rank 9	90.13	90.05	5.91	6.06	-
SQF [33] Cauchy, $f_0 = 1/6$, rank 11	90.13	90.02	5.12	5.87	-
SQF [33] Cauchy, $f_0 = 1/6$, rank 11 w/ NMS	90.03	90.00	4.19	3.93	-
SQF [33] log-Gabor, $f_0 = 1/6$, rank 7	90.13	90.10	5.82	5.95	-
SQF [33] log-Gabor, $f_0 = 1/6$, rank 9	90.11	90.14	5.63	5.57	-
SQF [33] log-Gabor, $f_0 = 1/6$, rank 11	90.04	90.03	5.32	5.21	-
Lorenz FB	90.04	90.12	8.54	11.97	18.7k
Lorenz FB + Vapnik	90.29	90.21	60.49	79.81	18.7k
Lorenz Max (Random init.)	89.95	89.91	11.92	19.38	18.7k
Lorenz Max (FB init.)	89.85	89.98	12.14	19.66	18.7k
Trained steerable filter(Lorenz loss[4]) rank 11	90.03	90.04	8.43	11.70	6.9k
Trained steerable filter(Focal loss[30]) rank 11	90.18	90.06	8.15	11.53	6.9k
PBT and Haar features[3, 35, 50, 11]	90.07	90.19	3.87	3.98	8.4k
Trained equivariant SFCNN Filters [51] w/ SQF NMS	89.99	90.00	6.31	6.46	0.12k
FCNN with NMS-aligned training examples	90.07	90.08	1.43	2.65	78k
1-layer SCNN rank 11, FB loss (2.20) + Lorenz loss[4] w/ NMS	90.09	89.98	2.48	2.94	6.9k
2-layer SCNN rank 11, FB loss (2.20) + Lorenz loss[4] w/ NMS	89.78	89.96	2.17	2.89	18.7k
4-layer SCNN rank 7 w/ SQF NMS (Soft Margin loss (2.9))	90.08	90.02	0.94	2.01	217k
4-layer SCNN rank 11 w/ SQF NMS (Soft Margin loss (2.9))	90.05	90.05	0.78	1.90	341k
4-layer SCNN rank 11 w/ SQF NMS (Focal loss [30])	90.18	90.10	**0.76**	**1.82**	341k

The response map obtained by any method was thresholded to obtain a binary detection image as shown in Figure 2.22 (one frame in the training set) and Figure 2.23 (one frame in the test set), with the threshold chosen so that the average detection rate was about 90%. The frame shown in Figure 2.23 is noisier than the one in Figure 2.22 .

We evaluated the detection performance on the training and test images. A guidewire pixel was considered detected if there is a detection (response above the threshold) at distance at most 2 pixels from it. A detection was considered a false positive if it is at distance at least 3 pixels from the guidewire or any catheter.

The results are shown in Table 2.2. The SFCNN filters[51] are also evaluated for comparison, obtaining a 6.46 test FP rate. The trained filters are shown in Figure 2.21. The 4-layer SCNN of

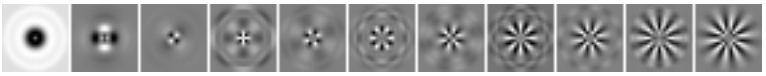

Figure 2.21: Trained SFCNN filters [51] basis.

rank 11 (Focal loss[30]) has the lowest false positive rate on the training and test set. It outperforms all the other methods including the FCNN and the PBT with Haar features. Among the filter based methods, the SQF with Cauchy filter of rank 11 with NMS performs the best on the training and test set, but it is outperformed by the training based methods.

2.8.6 Ablation Study

Table 2.3: Patch based evaluation: w/o NMS-aligned vs w/ NMS-aligned

	AUC		Angle Error	
Method	Train	Test	Train	Test
Lorenz FB (2.13)	0.8951	0.8730	6.14	7.23
Lorenz FB (2.13)(w/ NMS)	0.8618	0.8295	**2.71**	**3.24**
Steerable (2.20), rk. 11	0.8971	0.8766	5.12	6.02
Steerable (2.20)(w/ NMS), rk. 11	0.8622	0.8305	**2.14**	**2.42**
Steerable (2.20)(Focal[30]) rk. 11	0.8991	0.8776	6.62	7.51
Steerable (2.20)(Focal[30] w/ NMS) rk. 11	0.8682	0.8353	**2.63**	**2.78**

We show in Table 2.3 the influence of training the loss function and steerable filters with Lorenz loss [4] and Focal loss [30] based on NMS-aligned examples vs the examples obtained directly based on the annotation (w/o NMS-aligned examples). We see that the angle error on training and test sets has better performance using the NMS-aligned examples. However, the AUC on training and test set has better performance not using the NMS-aligned examples, since the NMS-aligned examples have more hard negative patches than the examples obtained directly based on the annotation.

We also show in Table 2.4 the evaluation of different methods on the entire images based on NMS-aligned examples and examples based on annotation. We see that both training and test FP rate are lower using the NMS-aligned examples. From the trained steerable filter results, the Focal loss [30] performs better than the Lorenz loss [4]. On training the FCNN with NMS-based positive examples vs positives extracted directly based on the annotation (w/o NMS-aligned

Table 2.4: Evaluate on the entire image: w/o NMS-aligned vs w/ NMS-aligned

	Det. rate		FP rate	
Method	Train	Test	Train	Test
SQF [33] Cauchy, $f_0 = 1/6$, rank 11	90.13	90.02	5.12	5.87
SQF [33] Cauchy, $f_0 = 1/6$, rank 11 w/ NMS	90.03	90.00	**4.19**	**3.93**
Lorenz FB	90.04	90.12	8.54	11.97
Lorenz FB(w/ NMS)	90.08	90.00	**3.13**	**3.75**
Trained steerable filter(Lorenz loss[4]) rk. 11	90.03	90.04	8.43	11.70
Trained steerable filter(Lorenz loss[4] w/ NMS) rk. 11	90.02	90.05	**3.02**	**3.62**
Trained steerable filter(Focal loss[30]) rk. 11	90.18	90.06	8.15	11.53
Trained steerable filter(Focal loss[30] w/ NMS) rk. 11	90.01	90.11	**2.91**	**3.44**
FCNN w/o NMS-aligned training examples	90.09	90.01	3.72	8.28
FCNN with NMS-aligned training examples	90.18	90.07	**1.43**	**2.65**
Steerable CNN rank 11 w/o SQF (Focal loss [30])	90.07	90.01	2.92	6.08
Steerable CNN rank 11 with SQF (Focal loss [30])	90.18	90.10	**0.76**	**1.82**

training examples), we see that both training and test FP rates are lower using the NMS-aligned examples.

To see whether the SQF are useful in screening the image and proposing good angle prediction for the steerable CNN, we also evaluated in Table 2.4 the trained steerable CNN by directly applying it to the whole image and obtaining the maximum response from 30 discrete angles in the range $[0, \pi]$ (Steerable CNN w/o SQF). Again we see that the SQF-based screening is useful, reducing the test FP rate from 6.08 to 1.82.

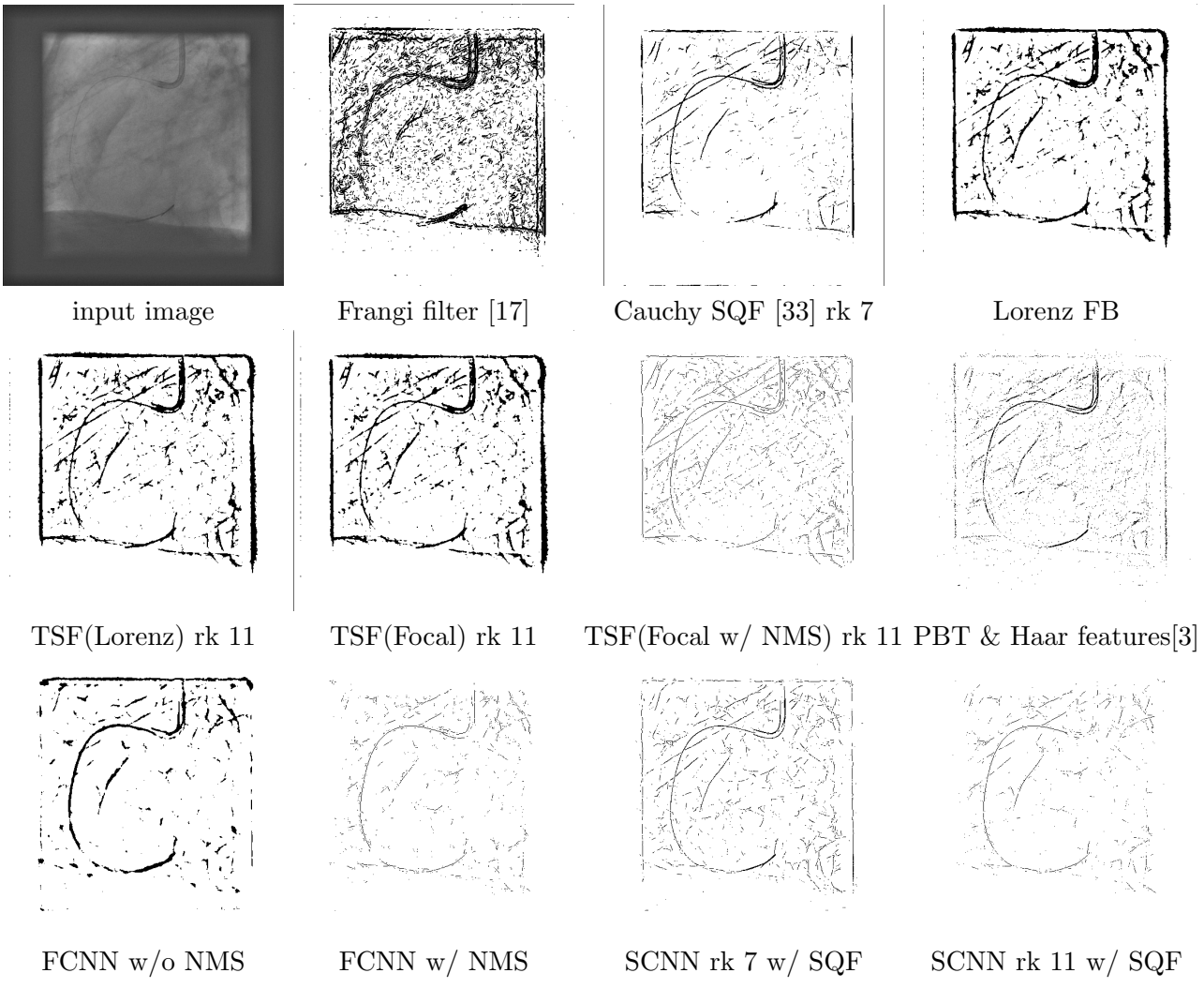

Figure 2.22: Guidewire detection examples of different methods for one training frame.

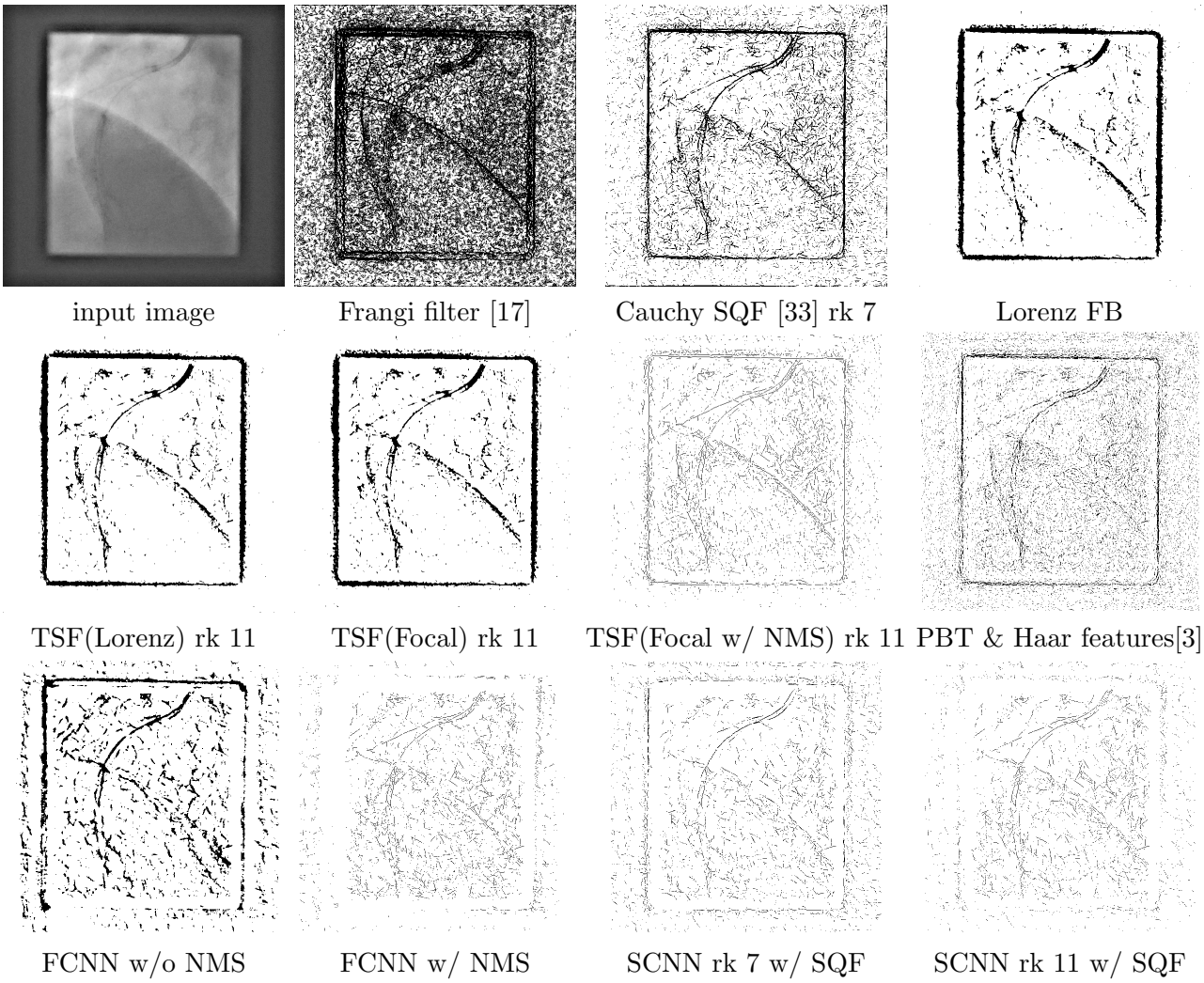

Figure 2.23: Guidewire detection examples of different methods for one test frame.

CHAPTER 3

RETINAL VESSEL DETECTION

3.1 Introduction

A fundus camera is used to photograph the rear of the eye that includes retina, optic cup, optic disc, macula, fovea, arteries and veins. The fundus is the only microcirculation system that is visible throughout the human body. By analyzing the fundus images, ophthalmologists can provide diagnosis and treatment of different eye diseases as well as diabetes, hypertension, leukemia, arteriosclerosis and anemia. The ophthalmologists will check the cup-to-disc ratio as a measurement to diagnose the glaucoma. Meanwhile, the arteries which provide enough blood supply to the eye and veins which carry the blood back to the heart are examined by the ophthalmologists. The veins are thicker than the arteries. The normal ratio of the diameter between arteries and veins in fundus images would be approximately $2 : 3$. Based on the vascular elasticity and hardness, ophthalmologists can make a diagnosis whether there's an arteriosclerosis or some other diseases.

The diagnosis of some retinal diseases can be made by analyzing the arteries and veins in fundus images, as the structure of the retinal vessels has different thicknesses and orientations depending on the disease. Due to those reasons, retinal vessel detection is important to recognize the disease automatically or to help the ophthalmologists diagnose and treat the diseases. Robust retinal vessel segmentation is necessary. Some diseases might be identified in an early stage so that the patient can receive better treatment. Automatic retinal vessel detection is a challenging problem due to the noisy background in the fundus images.

There are two main approaches to detect the vessels. First one consists of morphology-based approaches. One can use a predefined filter to obtain a filtered response map, and then with some vessel tracking methods to get the detection map. The second approach contains learning-based approaches, either supervised or unsupervised. For supervised learning one can use k-nearest neighbors, support vector machines (SVM) and neural networks. For unsupervised learning methods, K-means is used to identify whether a pixel is a vessel or not.

In this dissertation, a novel filter based method is presented that uses Spherical Quadrature Filters with morphological approaches. Fully Convolutional Networks that have the same architecture as in Section 2.4 are trained on solving this task and a Steerable CNN with equivariant steerable filters is also trained to solve the retinal vessel detection.

3.2 Related Work

Various approaches have been applied to blood vessel segmentation in fundus image, and can be grouped into filter based methods, morphological processing, vessel tracking and model based algorithms.

Filter based methods, such as Steerable Filters [18] were used for vessel segmentation. In [54], a matched filter with First-Order Derivative of Gaussian (MF-FDOG) was used to extract the retinal vessels. The thick blood vessels are detected by using large-scale MF-FDOG and small-scale MF-FDOG is used to detect the thin vessels. The detection maps are the combinations of the large-scale and small-scale results. A Second-Order Gaussian filter was proposed in [19] for detecting the retinal vessels in fundus images, but fails on most thin vessels.

For morphological processing, a method that combined the filters with morphological reconstruction to detect the centerlines of the retinal blood vessels and to preform vessel segmentation was presented in [36]. In [37], a discrete curvelet transform was used to enhance the retinal images for a better detection. By using morphological operators the falsely detected ridges were filtered out.

Kalman Filter and a Second-Order derivative Gaussian filter were employed in [13] for retinal vessel detection and tracking. The center and width were obtained by the Second-Order derivative Gaussian filter. Based on the center, an extended Kalman Filter was used to estimate the next position of the vessel.

Deep learning approaches can achieve state of the art results in learning patterns of the features directly from the images. Various Convolutional Neural Networks have been developed in image processing, such as Alex Net[26], VGG Net[46], R-CNN[20], U-Net[43], Fully Convolutional Networks[32], ResNet[22] and CapsNet[23, 44] etc. CNNs are employed to segment the vessels in fundus images.

In [31], a Convolution Neural Network with three convolutional layers, one pooling layer and two fully connected layers was trained for detecting the blood vessels under fundus images. The results outperformed the previous methods on accuracy and AUC. A similar conventional CNN with two convolutional layers, two pooling layers, a dropout layer and one fully connected layer has been employed to solve the retinal vessel detection in [21]. A reinforcement learning strategy is also applied to train the CNN more efficiently. In [1], another deep learning based approach based on the combination of U-Net, Recurrent Convolutional Neural Network and Residual Networks was proposed on retina vessel segmentation and other medical image segmentation. [16] proposed a method to apply neural architecture search and U-Net to segment the retinal vessels.

3.3 Annotation

The vessel annotation for the fundus images consists of pixelwise masks for the vessels in the image. This is a simple type of annotation that does not have the concept of vessel, vessel orientation, etc. The vessel orientation is important for our Steerable CNN as the Steerable CNN is trained to be sensitive to specific orientations.

For this reason, we are first investigating an automatic approach of extracting a higher level annotation that contains the vessel centerlines, vessel width and vessel orientation from the pixelwise annotation. The approach consists of the medial axis transform, distance transform and B-spline interpolation. The medial axis(skeleton) is the loci of the centers of maximal circles that can fit in the foreground region in 2-D. First, skeletonization is performed on the pixelwise annotation to obtain a set of points that are centered along the retinal vessel tree. Then, a distance transform is used to measure the thickness(the width) of each point of the pixelwise annotation. Then, short segments are fitted using B-spline on the set of center points. Finally, the set of points and the vessel thickness are used to reconstruct the retinal vessel.

In Figure 3.1 are shown the manual pixelwise annotation, the skeletonization result, the B-spline interpolation result and a reconstructed map from the B-spline vessel tree and thickness information. The Dice Similarity Coefficient is used for evaluating the reconstructed map. The Dice Similarity Coefficient is in the range $[0, 1]$, it is a popular performance measure for image segmentation methods. It can be written as

$$DSC = \frac{2TP}{2TP + FP + FN}. \tag{3.1}$$

True positive (TP) is the number of vessel pixels that are correctly classified as vessel pixels. True negative (TN) is the number of non-vessel pixels that are correctly classified as non-vessel pixels. False positive (FP) is the number of non-vessel pixels that are misclassified as vessel pixels. False negative (FN) is the number of vessel pixels that are misclassified as non-vessel pixels.

The average Dice Similarity Coefficient of the reconstruction of the vessel tree using the automatic method from the pixelwise annotation is 93.91% on the training and test datasets in DRIVE.

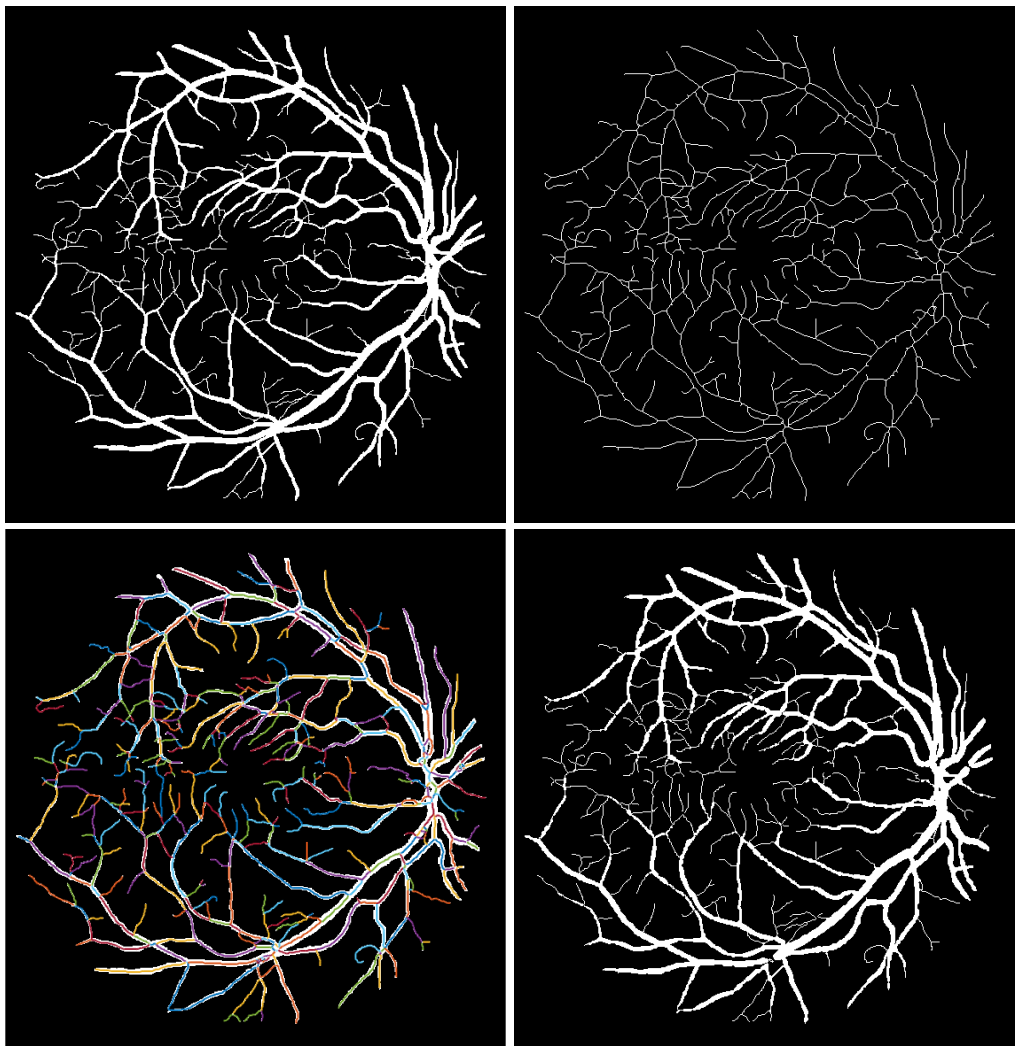

Figure 3.1: Top row: Examples of manual annotation and skeletonization result. Bottom row: Examples of B-spline of short segments and a reconstructed map

3.4 Methods

3.4.1 Region of Interest(ROI)

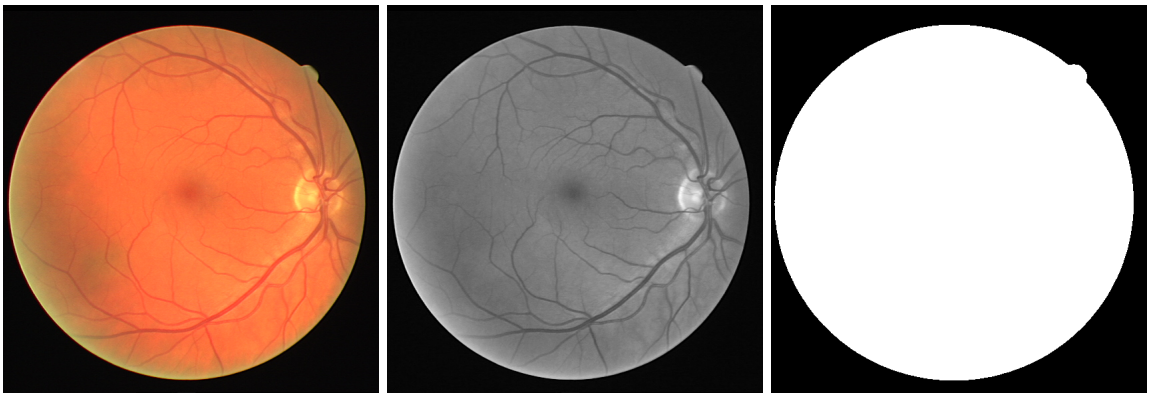

Figure 3.2: From Left to right: the fundus image, the grayscale image and region of interest mask.

The fundus images of the DRIVE dataset are of size 584×565. However, the retinal region is circular, as shown in Figure 3.2, left. First, the region of interest should be generated as a mask automatically to analyze the segmentation results. A threshold is chosen directly to separate the ROI and the non retinal part. This will make some pixels in the region of interest as the non retinal part. To overcome this, [53] proposed a method of generating the ROI by calculating the sum of difference among the RGB values. In our study, we used a similar method that obtains the mask ROI according to

$$ROI(R,G,B) = \begin{cases} 0, & \text{if } |R-G| + |R-B| + |G-B| < t, \\ 1, & otherwise \end{cases}, \tag{3.2}$$

where t is the threshold and is set to 41. An example of the ROI is shown in Figure 3.2,right. The average accuracy is 99.80%.

3.4.2 Vessel Light Reflex Removal

Before removing the vessel central light reflection, the RGB image is converted to a grayscale image, using the following equation

$$I(R,G,B) = c_r R + c_g G + c_b B \tag{3.3}$$

where R, G and B are the intensities of three channels and $c_r,\ c_g\ and\ c_b$ are the coefficients. An averaging-based method will average the RGB values, using $c_r = \frac{1}{3},\ c_g = \frac{1}{3},\ c_b = \frac{1}{3}$. The luminosity method sets the coefficients to $c_r = 0.3,\ c_g = 0.59,\ c_b = 0.11$. In retinal vessel detection methods, the green channel is used as grayscale image because the green channel has the highest contrast[52]. The green channel is used as the gray level intensity map in this dissertation. An example is shown in Figure 3.2, middle.

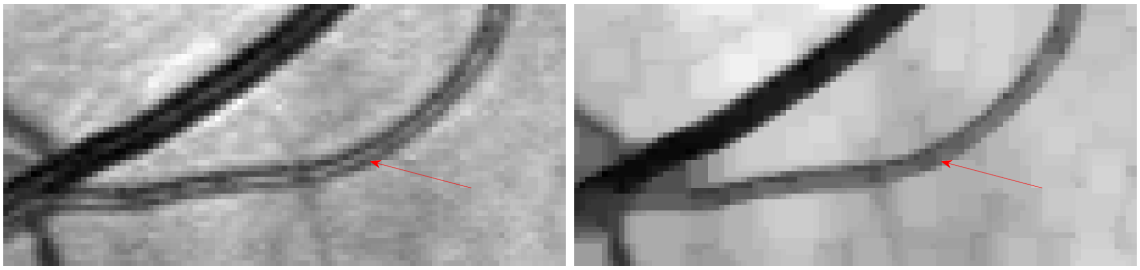

Figure 3.3: Left: vessel with central light reflection. Right: after light reflection removal by morphological opening.

The retinal vessels have lower brightness than the retinal surface. However, some blood vessels may include a light reflection along the vessel centerline, as shown in Figure 3.3, left. To remove the vessel light reflection, a morphological opening operation is used with a three-pixel diameter disc and a square grid by using eight-connectivity [34]. An example of removal result are shown in Figure 3.3, right.

3.4.3 Spherical Quadrature Filters

The Cauchy Spherical Quadrature Filters [33] of rank 9 are employed to obtain the filtered response, the SQFs equation is described in Eq. (2.1) with a Cauchy isometric filter in Eq. (2.5). The peak tuning frequency ω_0 is set to $\frac{1}{3}$. In guidewire detection, the SQFs are shown to detect the thinner wire-like structures accurately. However, there is a problem in vessel detection that the center of some thicker vessels is considered as background pixels, as shown in Figure 3.4, left. To improve classification, the fundus images are downscaled to half size and the SQFs are applied on the downscaled images.

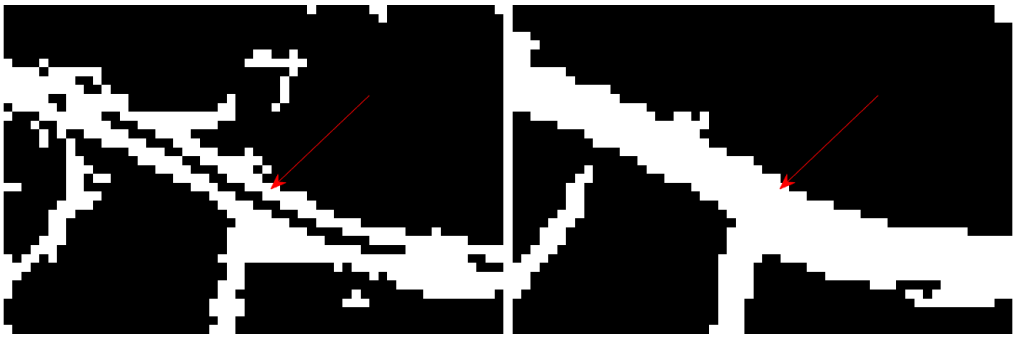

Figure 3.4: Left: SQF detection results of full-scale fundus images. Right: SQF detection results of downscaled fundus images.

Then, top-hat transform is applied after the filtered responses. The white top-hat transform is defined as the difference between the input image and the morphologically open image

$$I_{Top-Hat} = I - I \circ rn \tag{3.4}$$

where "$\circ$" denotes the opening operation, r is the radius and n is the number of line structuring elements. In this dissertation, the best performance is obtained by a combination of a disk-shaped structuring element with radius of 3 and 8 line structuring elements.

Finally, the filtered downscaled images are resized back to the original size of 584×565. An example of obtained SQF detection result is shown in Figure 3.4, right.

3.4.4 Fully Convolutional Neural Network

Assume we are given n training patches $(\mathbf{x}_i, y_i), i = 1, ..., n$ where $\mathbf{x}_i \in \mathbb{R}^{p^2}$ is the image of a patch of size $p \times p$ either centered on the retinal vessels (a positive example) or away from the retinal vessels (a negative example), and y_i is the label. The labels are $y_i = -1$ for negative patches and $y_i = 1$ for patches centered on the retinal vessel.

FCN Architecture. A Fully Convolutional Neural Network is employed on detecting the retinal vessels. The network architecture is the same as the one in Figure 2.4. For a receptive field of 25×25, there are 4 convolutional layers, and the third one followed by a ReLU layer (Rectified Linear Unit). The last convolutional layer obtains the binary response whether the patch is a retinal vessel or not. The first convolutional layer has 16 filters of size 7×7, the following layers

have 32 filters of size 7×7. The last one contains 1 filter of size 7×7. We used the Focal loss [30] from Eq. (2.10) to guide the training.

Training Initialization. All weights were initialized with random Gaussian values with std 0.01. The training was performed using the Adam optimizer[25] with the weight decay 0.0001, the initial learning rate 0.00001 and an initial mini-batch size 32. The learning rate was multiplied by 0.8 and the minibatch was doubled every 50 epochs, for a total of 300 epochs.

3.4.5 Steerable CNN

The Steerable CNN is similar to the one in Section 2.7. The difference is that the steerable filters of the SCNN in retinal vessel detection is tuned to be equivariant as illustrated in [51]. Assume we are given n training patches $(\mathbf{x}_i, y_i, \alpha_i), i = 1, ..., n$ where $\mathbf{x}_i \in \mathbb{R}^{p^2}$ is the image of a patch of size $p \times p$ either centered on the retinal vessel (a positive example) or away from the retinal vessel (a negative), $y_i \in \{-1, 1\}$ is the label, and $\alpha_i \in [0, \pi)$ is the orientation. The orientation at the center location of each patch is obtained by Steerable Quadrature Filters (SQF) [33].

SCNN Architecture. A 2-layer Steerable CNN and a 4-layer Steerable CNN are trained. The steerable filters in each layer has been forced to be equivariant based on [51]. The basis of each steerable filters is set to rank 11. The learned steerable filters are defined as a linear combination of some weight w and the basis B. All basis B's of steerable filters were initialized with

$$B = Re\ (\exp(-(r - \mu_j)^2/2)e^{ik\phi}), \tag{3.5}$$

where Re denotes the real part of the equation, $\mu_j = j$ is the radius with $j = 1, 2..., J$ and k is the angular frequency $k = 0, 1, ..., K$.

A neuron of the first layer of the SCNN with a equivariant steerable filters for any angle θ_k has the following form:

$$f(\mathbf{x}, \theta_i) =< (\mathbf{x} * \mathbf{B}) \cdot \mathbf{A}_k, \mathbf{w} > \tag{3.6}$$

where $\mathbf{x}$ is an input patch, $\mathbf{B}$ is the equivariant basis from Eq. 3.5, "$*$" denotes the convolution,"$\cdot$" denotes elementwise multiplication, $\mathbf{w}$ denotes a learned weight vector of size $JK + 1$, "$<, >$" denotes the dot product, and $\mathbf{A}^{(i)}$ is a vector of size $JK + 1$:

$$A_k = [\underbrace{1, ..., 1}_{\text{J+1}}, \underbrace{\cos(2\theta_k), \sin(2\theta_k), ..., \cos(2\theta_k), \sin(2\theta_k)}_{\text{2J}}, ..., \underbrace{\cos(2d\theta_k), \sin(2d\theta_k), ..., \cos(2d\theta_k), \sin(2d\theta_k)}_{\text{2J}}]$$

where $d = \frac{K-1}{2}$.

Training Initialization. The initial learning rate was 0.00003 and the initial mini-batch size was 32. The mini-batches size was doubled every 50 epochs. We set the maximum mini-batch size to 4096. After the mini-batch size became 4096, the learning rate was multiplied by 0.5 every 50 epochs, for a total of 500 epochs. The training was performed using Adam optimizer [25].

3.4.6 Post-processing

A post-processing step is applied after obtaining the binary response map of the retinal vessels. Due to the detection result having gaps, a filling step is performed. If the pixel is considered not detected and at least 6 neighbors are classified as vessels, the pixel is also classified as a vessel pixel.

3.5 Experiments

The experiments are performed on a single GPU machine with NVIDIA GeForce GTX 1070 with 6GB memory and 64GB of RAM. The Pytorch [40] framework is used for training the FCN, SCNN and for vessel detection.

3.5.1 Dataset

The DRIVE (Digital Retinal Images for Vessel Extraction) database is used for evaluation. The photographs for the DRIVE database were obtained from a diabetic retinopathy screening program in the Netherlands. The images are of size 584×565.

For the learning based approaches that require a training set, 20 images are used for training and the other 20 are used for testing. The training patches are extracted from the images of size 25×25. The annotations are manually obtained by two experts, and we used the results obtained by the first expert as the ground truth.

The training set of FCN contains 362,320 positives and 4,580,866 negatives. The training set of SCNN contains 510,041 positives and 1,034,628 negatives.

3.5.2 Performance Metrics

In this dissertation, we evaluated the performance of the methods in terms of sensitivity, specificity and accuracy. They are defined as:

$$Sensitivity = \frac{TP}{TP + FN}, \tag{3.7}$$

$$Specificity = \frac{TN}{TN + FP}, \tag{3.8}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}. \tag{3.9}$$

where TP, TN, FP, FN have been defined in Section 3.3. The area under curve (AUC) is also used as an evaluation measure.

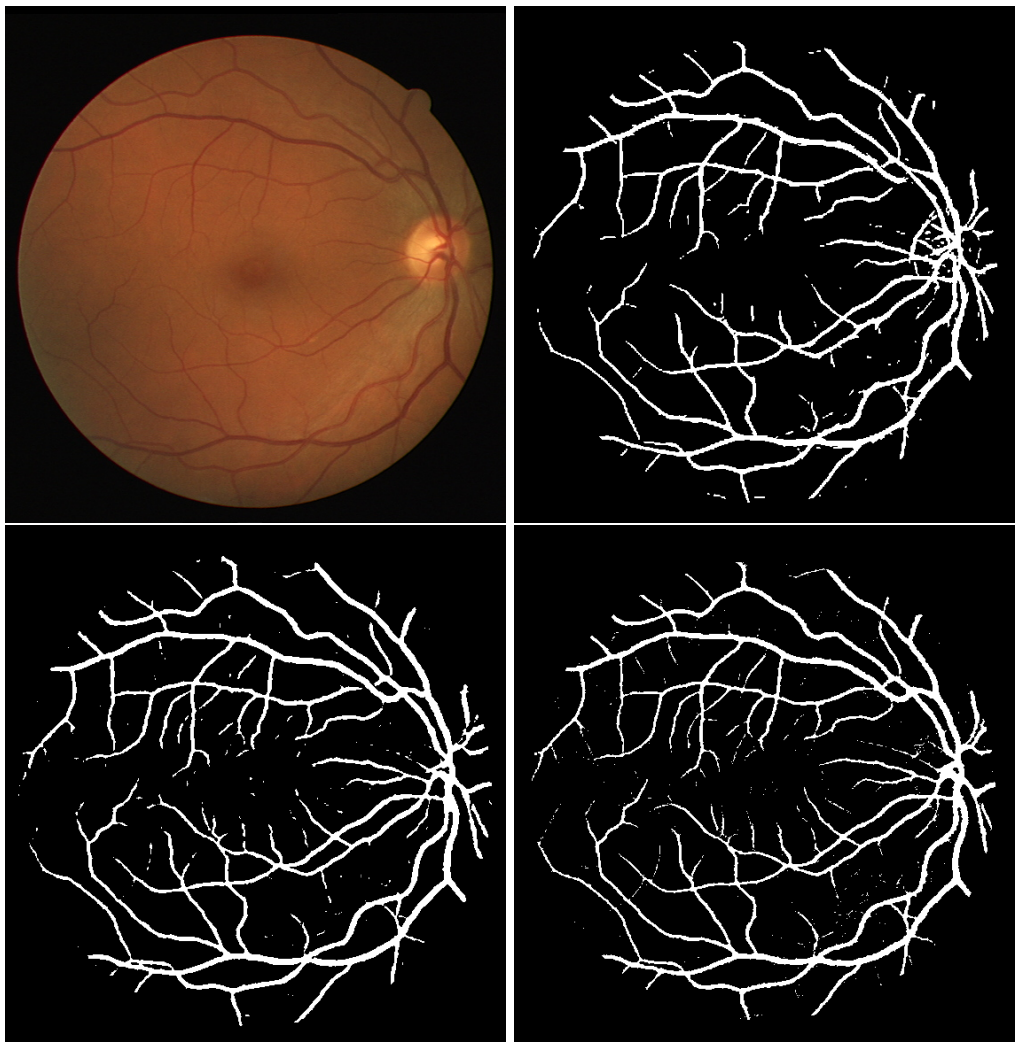

Figure 3.5: Detection examples of fundus images, top row: input image, Cauchy SQF rank 9. Bottom row: FCN detection result, SCNN 4-layer detection result.

3.5.3 Results

In Table 3.1 are shown the results of different methods and our proposed methods on the DRIVE dataset. The first part contains the filter based and morphological approaches. The proposed SQF rank 9 outperformed the other methods on both Sensitivity and Accuracy. The second part contains the learning based approaches. Our proposed FCN method outperformed other existing methods on Sensitivity and Accuracy. The SCNN with 4 layers outperformed all the other methods on Sensitivity and Accuracy. The Specificity of 4-layer SCNN is slightly lower than the Li et al. [28] and AUC is lower than the other methods. The detection examples of different methods are shown in Figure 3.5. The ROC curves of FCN, 2-layer SCNN and 4-layer SCNN are shown in Figure 3.6.

Table 3.1: Comparison with different methods on 25×25 patches.

	Sensitivity	Specificity	Accuracy	AUC
Mendonca et al. [36]	0.7344	0.9764	0.9452	-
Zhang et al. [54]	0.7120	0.9724	0.9382	-
Miri et al. [37]	0.7352	**0.9795**	0.9458	-
Yavuz et al. [53]	0.6779	0.9786	0.9400	-
SQF [33] rank 9	**0.7353**	0.9706	**0.9498**	0.8108
Liskowski et al. [31]	0.7811	0.9807	0.9535	0.9790
Alom et al. [1]	0.7792	0.9813	0.9556	0.9784
Li et al. [28]	0.7791	**0.9831**	0.9574	**0.9813**
Proposed FCN	0.7856	0.9806	0.9633	0.9743
Proposed FCN w/ sqf	0.7847	0.9719	0.9554	0.9526
SCNN 2L	0.7846	0.9803	0.9629	0.9044
SCNN 4L	**0.7873**	0.9820	**0.9646**	0.9050

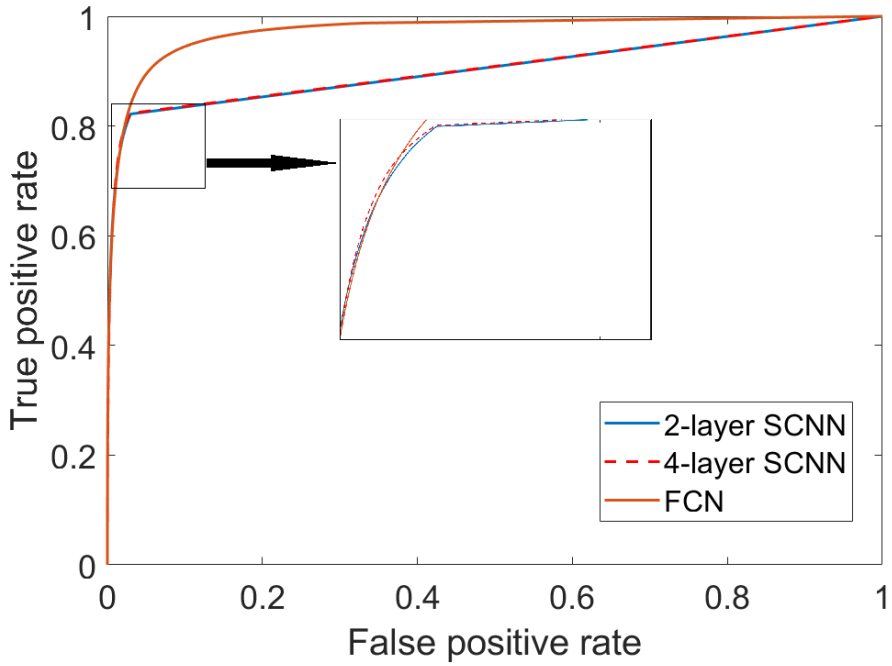

Figure 3.6: ROC for FCN and SCNN.

CHAPTER 4

CONCLUSION

In this dissertation, we introduced a simple steerable CNN that can be tuned using an orientation parameter θ to be sensitive to objects aligned to that orientation, instead of being rotation invariant. Another contribution was to apply the Spherical Quadrature Filters to the X-ray images and fundus images to obtain a detection map. We also proposed a learning-based method for training a class of oriented filters and training Fully Convolutional Networks for guidewire detection in fluoroscopic images.

We presented the mathematical formulation of the steerable CNN, and how to train it using examples at any orientation, without rotating them for alignment. As application, we used the steerable CNN to detect guidewire pixels in fluoroscopic images, where a regular CNN overfits because the wire is very thin and covers only a small percentage of the input.

We reported the difficulties we encountered while training the regular CNN due to the fact that the guidewire is thin and noisy, and imprecision in annotation makes the training even more difficult. To address these issues, we presented two training-related measures specific to this problem. First, we showed how to get a better initialization using training examples from one image or one sequence only. Then, we explained how to obtain better aligned training examples using the Spherical Quadrature Filters [33] and non-maximal suppression.

The experiments of guidewire detection show that the Steerable Convolutional Neural Network trained with our SQF-aligned data outperformed all other methods evaluated. In terms of filter-based methods, we evaluated the Frangi filter [17] and the Spherical Quadrature Filters [33] and we observed that the learning based methods outperform the filter based methods.

SQFs were also used as a new filter based approach to segment the retinal vessels, and can obtain a higher Sensitivity and Accuracy than the existing filter based methods. Also our trained Fully Convolutional Network and Steerable Convolutional Neural Network can obtain better retinal vessel segmentation Sensitivity and Accuracy, with a sightly lower Specificity and AUC.

In the future, we plan to apply the steerable CNN to automatic guidewire localization. The guidewire localization is a higher level process that uses the pixelwise detection as a data term to find the most likely position of the guidewire by searching the high dimensional space of all possible curves.

BIBLIOGRAPHY

[1] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M Taha, and Vijayan K Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*, 2018.

[2] Shirley AM Baert, Wiro J Niessen, Erik HW Meijering, Alejandro F Frangi, and Max A Viergever. Guide wire tracking during endovascular interventions. In *MICCAI*, pages 727–734, 2000.

[3] Adrian Barbu, Vassilis Athitsos, Bogdan Georgescu, Stefan Boehm, Peter Durlak, and Dorin Comaniciu. Hierarchical learning of curves application to guidewire localization in fluoroscopy. In *CVPR*, pages 1–8, 2007.

[4] Adrian Barbu, Yiyuan She, Liangjing Ding, and Gary Gramajo. Feature selection with annealing for computer vision and big data learning. *IEEE Trans. on PAMI*, 39(2):272–286, 2017.

[5] Vincent Bismuth, Régis Vaillant, Hugues Talbot, and Laurent Najman. Curvilinear structure enhancement with the polygonal path image-application to guide-wire segmentation in x-ray fluoroscopy. In *MICCAI*, pages 9–16, 2012.

[6] Djamal Boukerroui, J Alison Noble, and Michael Brady. On the choice of band-pass quadrature filters. *Journal of Mathematical Imaging and Vision*, 21(1-2):53–80, 2004.

[7] Alexander Brost, Andreas Wimmer, Rui Liao, Joachim Hornegger, and Norbert Strobel. Catheter tracking: Filter-based vs. learning-based. In *DAGM-Symposium*, pages 293–302, 2010.

[8] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[9] Tak-Shing Chan and Ajay Kumar. Reliable ear identification using 2-d quadrature filters. *Pattern Recognition Letters*, 33(14):1870–1881, 2012.

[10] Bor-Jeng Chen, Ziyan Wu, Shanhui Sun, Dong Zhang, and Terrence Chen. Guidewire tracking using a novel sequential segment optimization method in interventional x-ray videos. In *ISBI*, pages 103–106, 2016.

[11] Terrence Chen, Gareth Funka-Lea, and Dorin Comaniciu. Robust and fast contrast inflow detection for 2d x-ray fluoroscopy. *MICCAI*, pages 243–250, 2011.

[12] Erkang Cheng, Yu Pang, Ying Zhu, Jingyi Yu, and Haibin Ling. Curvilinear structure tracking by low rank tensor approximation with model propagation. In *CVPR*, pages 3057–3064, 2014.

[13] O Chutatape, Liu Zheng, and Shankar M Krishnan. Retinal blood vessel detection and tracking by matched gaussian and kalman filters. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)*, volume 6, pages 3144–3149. IEEE, 1998.

[14] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

[15] Taco S Cohen and Max Welling. Steerable cnns. *ICLR*, 2017.

[16] Zhun Fan, Jiahong Wei, Guijie Zhu, Jiajie Mo, and Wenji Li. Enas u-net: Evolutionary neural architecture search for retinal vessel segmentation. *arXiv preprint arXiv:2001.06678*, 2020.

[17] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *MICCAI*, pages 130–137, 1998.

[18] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Trans. on PAMI*, 13(9):891–906, 1991.

[19] Luo Gang, Opas Chutatape, and Shankar M Krishnan. Detection and measurement of retinal vessels in fundus images using amplitude modified second-order gaussian filter. *IEEE transactions on Biomedical Engineering*, 49(2):168–172, 2002.

[20] 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.

[21] Yanhui Guo, Ümit Budak, Lucas J Vespa, Elham Khorasani, and Abdulkadir Şengür. A retinal vessel detection approach using convolution neural network with reinforcement sample learning strategy. *Measurement*, 125:586–591, 2018.

[22] 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.

[23] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[24] Nicolas Honnorat, Régis Vaillant, and Nikos Paragios. Guide-wire extraction through perceptual organization of local segments in fluoroscopic images. *MICCAI*, pages 440–448, 2010.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[27] Donghang Li and Adrian Barbu. Training a cnn for guidewire detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2214–2218. IEEE, 2019.

[28] Liangzhi Li, Manisha Verma, Yuta Nakashima, Hajime Nagahara, and Ryo Kawasaki. Iternet: Retinal image segmentation utilizing structural redundancy in vessel networks. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3656–3665, 2020.

[29] Laurence S Lim, Lieng H Ling, Peng Guan Ong, Wallace Foulds, E Shyong Tai, and Tien Yin Wong. Dynamic responses in retinal vessel caliber with flicker light stimulation and risk of diabetic retinopathy and its progression. *Investigative ophthalmology & visual science*, 58(5):2449–2455, 2017.

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[31] Paweł Liskowski and Krzysztof Krawiec. Segmenting retinal blood vessels with deep neural networks. *IEEE transactions on medical imaging*, 35(11):2369–2380, 2016.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[33] Ross Marchant and Paul Jackway. Feature detection from the maximal response to a spherical quadrature filter set. In *DICTA*, pages 1–8. IEEE, 2012.

[34] Diego Marín, Arturo Aquino, Manuel Emilio Gegúndez-Arias, and José Manuel Bravo. A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. *IEEE transactions on medical imaging*, 30(1):146–158, 2010.

[35] Philippe Mazouer, Terrence Chen, Ying Zhu, Peng Wang, Peter Durlak, Jean-Philippe Thiran, and Dorin Comaniciu. User-constrained guidewire localization in fluoroscopy. In *SPIE Med. Imag.*, volume 7259, pages 72591K–72591K, 2009.

[36] Ana Maria Mendonca and Aurelio Campilho. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE transactions on medical imaging*, 25(9):1200–1213, 2006.

[37] Mohammad Saleh Miri and Ali Mahloojifar. Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction. *IEEE Transactions on Biomedical Engineering*, 58(5):1183–1192, 2010.

[38] Matthias P Naegele, Jens Barthelmes, Valeria Ludovici, Silviya Cantatore, Arnold von Eckardstein, Frank Enseleit, Thomas F Lüscher, Frank Ruschitzka, Isabella Sudano, and Andreas J Flammer. Retinal microvascular dysfunction in heart failure. *European heart journal*, 39(1):47–56, 2018.

[39] Matthias P Nägele, Jens Barthelmes, Valeria Ludovici, Silviya Cantatore, Michelle Frank, Frank Ruschitzka, Andreas J Flammer, and Isabella Sudano. Retinal microvascular dysfunction in hypercholesterolemia. *Journal of clinical lipidology*, 12(6):1523–1531, 2018.

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[41] Olivier Pauly, Hauke Heibel, and Nassir Navab. A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences. *MICCAI*, pages 343–350, 2010.

[42] Giuseppe Querques, Enrico Borrelli, Riccardo Sacconi, Luigi De Vitis, Letizia Leocani, Roberto Santangelo, Giuseppe Magnani, Giancarlo Comi, and Francesco Bandello. Functional and morphological changes of the retinal vessels in alzheimer's disease and mild cognitive impairment. *Scientific reports*, 9(1):1–10, 2019.

[43] 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.

[44] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[45] Radim Sara and Vojtech Cvrcek. Faint streak detection with certificate by adaptive multi-level bayesian inference. *European Conference on Space Debris*, 2017.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[47] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV*, volume 2, pages 1589–1596. IEEE, 2005.

[48] Vladimir Naumovich Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[49] Li Wang, Xiao-Liang Xie, Gui-Bin Bian, Zeng-Guang Hou, Xiao-Ran Cheng, and Pusit Prasong. Guide-wire detection using region proposal network for x-ray image-guided navigation. In *IJCNN*, pages 3169–3175. IEEE, 2017.

[50] Peng Wang, Terrence Chen, Ying Zhu, Wei Zhang, S Kevin Zhou, and Dorin Comaniciu. Robust guidewire tracking in fluoroscopy. In *CVPR*, pages 691–698, 2009.

[51] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018.

[52] Bo Wu, Weifang Zhu, Fei Shi, Shuxia Zhu, and Xinjian Chen. Automatic detection of microaneurysms in retinal fundus images. *Computerized Medical Imaging and Graphics*, 55:106–112, 2017.

[53] Zafer Yavuz and Cemal Köse. Blood vessel extraction in color retinal fundus images with enhancement filtering and unsupervised classification. *Journal of healthcare engineering*, 2017, 2017.

[54] Bob Zhang, Lin Zhang, Lei Zhang, and Fakhri Karray. Retinal vessel extraction by matched filter with first-order derivative of gaussian. *Computers in biology and medicine*, 40(4):438–445, 2010.

BIOGRAPHICAL SKETCH

Donghang Li received his Bachelor's degree in Information Engineering from Shanghai University, Shanghai, China in 2012. He completed his M.S. degree in Mathematics in University of Massachusetts, Lowell in 2014.

He started his PhD program in Department of Statistics in Florida State University in 2014 under the supervision of Dr. Adrian Barbu. He worked on projects in medical image processing such as guidewire detection and retinal vessel detection.

His research interests include medical image processing and machine learning.