Training a Steerable CNN for Guidewire Detection

Donghang Li, Adrian Barbu
Florida State University
600 W College Ave, Tallahassee, FL 32306, USA
dl14h@my.fsu.edu, abarbu@stat.fsu.edu

Abstract

Guidewires are thin wires used in coronary angioplasty to guide different tools to access and repair the obstructed artery. The whole procedure is monitored using fluoroscopic (real-time X-ray) images. Due to the guidewire being thin in the low quality fluoroscopic images, it is usually poorly visible. The poor quality of the X-ray images makes the guidewire detection a challenging problem in image-guided interventions. Localizing the guidewire could help in enhancing its visibility and for other automatic procedures. Guidewire localization methods usually contain a first step of computing a pixelwise guidewire response map on the entire image. In this paper, we present a steerable Convolutional Neural Network (CNN), which is a Fully Convolutional Neural Network (FCNN) that can detect objects rotated by an arbitrary 2D angle, without being rotation invariant. In fact, the steerable CNN has an angle parameter that can be changed to make it sensitive to objects rotated by that angle. We present an application of this idea to detecting the guidewire pixels, and compare it with an FCNN trained to be invariant to the guidewire orientation. Results reveal that the proposed method is a good choice, outperforming some popular filter-based and learning-based approaches such as Frangi Filter, Spherical Quadrature Filter, FCNN and a state of the art trained classifier based on hand-crafted feature.

1. Introduction

Convolutional Neural Networks(CNNs) are widely used and have an impressive performance in detecting and classifying objects. However, the CNNs performance is sensitive to variations in rotation, position or scaling of the objects to be detected. In [14], capsules were proposed as accurate generative models to handle such variations and obtain more accurate representations, with promising results. However, each capsule is trained to handle only a small range of such variations, hence the need for multiple capsules to handle the same object.

In this paper we propose a steerable CNN that can detect an object rotated by an arbitrary angle without being rotation invariant. The proposed model 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 steerable CNN to detect the guidewire in fluoroscopy (real-time X-ray) images. The guidewire is a thin wire used in coronary angioplasty interventions, which are visualized using fluoroscopic images. The fluoroscopic images are usually low-dose in order to limit the amount of radiation received by the patient. Under these conditions, the guidewire is a thin and poorly visible wire-like structure with different orientations, as shown in Figure 1. In this application, knowing the orientation of the guidewire is important for its detection, but the scaling is not important since all guidewires are one or two pixels wide.

In order to find the entire guidewire, a low level measurement that shows how the guidewire passes through any pixel of the image should be obtained first. More details about this procedure will be explained in Section 2. There are two main approaches to obtain the pixelwise detection map, filter-based approaches and learning-based approaches. For the filter-based approach, one applies a predefined filter (Frangi Filter, Steerable Filters, Spherical Quadrature Filters) to obtain a filtered response map. The learning-based approach is to find a per-pixel probability map by training a classifier with some Haar or hand-crafted features.

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.

Recently, a CNN (Convolutional Neural Network) was trained for this purpose, which learned its own features using the training data and obtained an invariant model that can detect guidewires at any orientation.

In this paper, we are interested in seeing what is to gain by training a CNN that is tuned to the guidewire orientation.

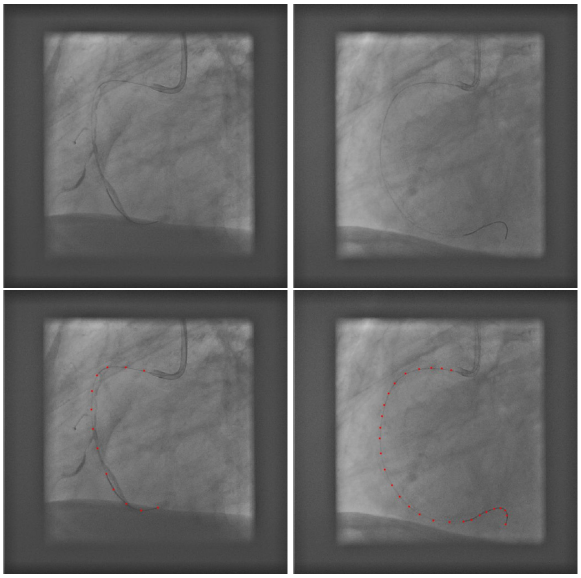

Figure 1. Two frames of the guidewire under X-ray images(top two images) and the X-ray frames of the guidewire with annotations(bottom two images)

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 CNN that can be trained on the original samples. This steerable CNN has a steering parameter θ that can be used to make it sensitive to the parts of the guidewire that have orientation θ. This way, the steerable CNN eliminates the need to rotate the image by many angles at detection time. This paper brings three contributions:

- It introduces a method for training steerable filters [13] by loss minimization. The steerable filters are rotated filters that can be obtained from a basis using an arbitrary rotation parameter. They were introduced in [13], but they were defined by an equation instead of being trainable.

- It introduces a model for a steerable CNN composed of a number of layers consisting of multiple steerable filters, and a method for training the steerable CNN from training examples.

- It presents an application to guidewire detection in fluoroscopic images. Our experiments indicate that the steerable CNN outperforms the regular CNN and the other guidewire detection methods such as the Frangi Filter and a trained classifier with Haar or hand-crafted features.

1.1. Related Work

Filter-based approaches include the Frangi Filter [12], 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 [1]. [7] used the Frangi Filter as the data term and fitted the guidewire with B-spline model in clinical X-ray videos. The beauty of the filter-based approaches consists in their simplicity and interpretability. [4] 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 [13] for detecting edges and ridges in images. A more recent and powerful type of steerable filters are the Spherical Quadrature Filters (SQF) [19] that were used by [17] for guidewire detection.

Learning-based approaches include [5, 2, 20, 26, 15, 22, 8]. In [2], the pixel detection step was trained with examples that were rotated for alignment, using a Probabilistic Boosting Tree (PBT) [24] 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 [20] to localize the guidewire. The PBT and hand-crafted features were also employed to track the guidewire in [26] and detecting vessels in [8]. [5] 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 [15]. It was trained on ridge and edge features. [9] detected the catheter and vascular structures using a Random Forest classifier of curvilinear structures trained on hand-crafted features. A method used the Region Proposal Network to detect the guidewire was presented in [25]. Different from our method which is aimed at obtaining a pixel-wise detection map using CNNs, their work is to place bounding boxes around the guidewires.

A Fully Convolutional Neural Network was trained in [17] 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 [11]. 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 [13]. 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 [27]. 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

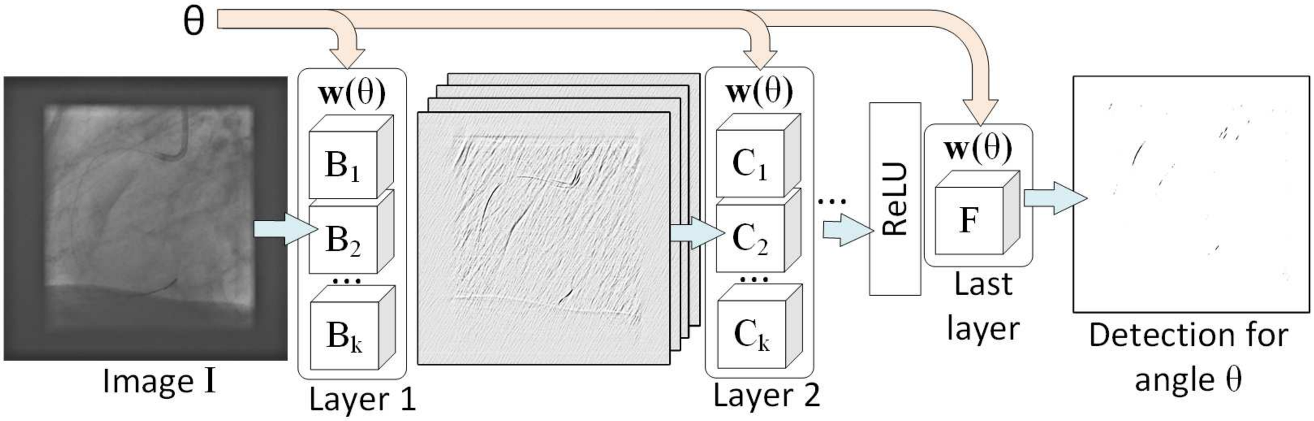

Figure 2. Diagram of the steerable CNN, steered (tuned) to an angle θ.

a basis that is not necessarily equivariant, but which is made close to equivariant by using a special loss function. Forthermore, the rotation and steering occurs only on the first layer for the SFCNN and is followed by several group-convolutional layers[10]. In our method, each layer is steerable by the same angle θ, making the entire CNN steerable. **Capsules** were introduced in [14] and improved in [23]. 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. The Steerable CNN

The steerable CNN consists of a number of steerable convolutional layers, as illustrated in Figure 2. The steerable convolution filters are described next.

2.1. The Trainable Steerable Filters

The steerable filters are oriented filters $f(\theta)$ that are obtained as a linear combination from a basis B, and can be rotated to any angle θ by a simple re-weighting of the basis as illustrated in Figure 3.

We are interested in deriving such a steerable representation for a filter $\mathbf{f}$. For that, we start with the steerable filter [13] of order 2, which for an angle θ is defined as

$$G_2^\theta = B \cdot \mathbf{a}(\theta) \tag{1}$$

where $\mathbf{a}(\theta) = (\cos^2\theta, -2\cos\theta\sin\theta, \sin^2\theta)^T$, $B = \frac{G}{\sigma^4} \cdot (x^2 - \sigma^2, -xy, y^2 - \sigma^2)$, and G is the 2-D Gaussian with variance σ^2. This inspires us to represent a filter as

$$\mathbf{f}(\theta) = B \cdot \mathbf{a}(\theta) \tag{2}$$

with some unknown $p^2 \times (d+1)$ matrix B that needs to be learned and

$$\mathbf{a}(\theta) = (\cos^d\theta, \cos^{d-1}\theta\sin\theta, ..., \sin^d\theta)^T. \tag{3}$$

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 will use an alternate steerable representation

$$\mathbf{f}(\theta) = B \cdot \mathbf{w}(\theta) \tag{4}$$

with

$$\mathbf{w}(\theta) = [1, \cos(2\theta), \sin(2\theta), ..., \cos(2d\theta), \sin(2d\theta)]^T, \tag{5}$$

where d controls the number of basis elements. We will denote the dimension $2d+1$ of the basis B as the rank of the steerable filter.

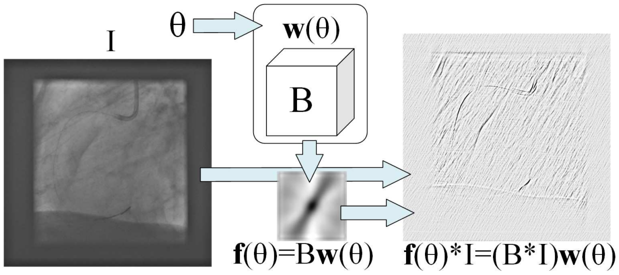

Figure 3. Diagram of a trainable steerable convolution filter.

Training steerable filters. Suppose we have n training examples $(\mathbf{x}_i, y_i), i = 1, ..., n$ where $\mathbf{x}_i \in \mathbb{R}^{p^2}$ is the extracted patch of size $p \times p$. The positives are patch with center on the guidewire, and the negatives have center at some distance from the guidewire. The label of the example is y_i, with $y_i = -1$ for negative patches and $y_i = k \in \{1, ..., K\}$ for positive patches where the tangent angle is in the interval $[\theta_k - \frac{\pi}{2K}, \theta_k + \frac{\pi}{2K})$, where $\theta_k = \frac{k\pi}{K}$.

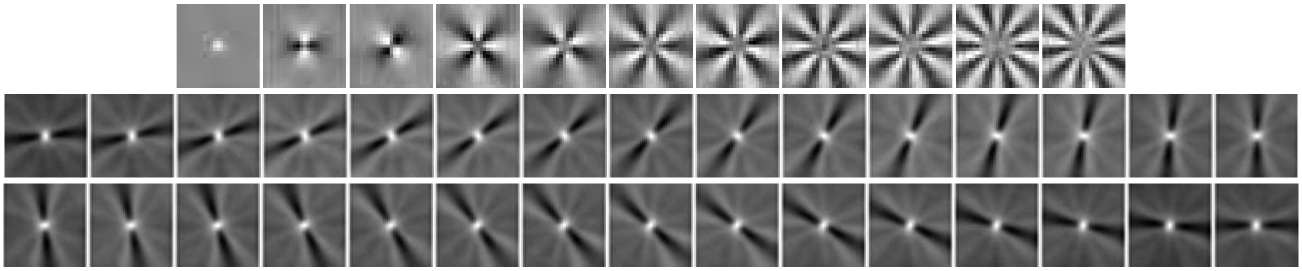

Figure 4. A trained steerable filter example. First row: trained filter basis B. Second and third row: obtained steered filters for different angles θ.

We can use the *foreground-background* (FB) loss

$$\begin{aligned} L(B) =& \frac{1}{n_{-1}} \sum_{i,y_i=-1} \sum_{k=1}^{K} \ell(-\mathbf{x}_i^T B \mathbf{w}(\theta_k)) \\ &+ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,y_i=k} \ell(\mathbf{x}_i^T B \mathbf{w}(\theta_k)) \end{aligned} \tag{6}$$

for training the steerable filters, where $\ell(u)$ is a per example loss function such as the Lorenz loss [3]

$$\ell(u) = \log(1 + ReLU(1-u)^2), \tag{7}$$

or the Focal loss [18]

$$\ell(u) = -\alpha_t (1 - p_t(u))^\gamma \log(p_t(u)). \tag{8}$$

where

$$p_t(u) = \begin{cases} \sigma(u), & \text{if } y = 1 \\ 1 - \sigma(u), & \text{if } y = -1 \end{cases}, \tag{9}$$

where $\sigma(u)$ is the sigmoid function

$$\sigma(u) = \frac{1}{1 + \exp(-u)}. \tag{10}$$

and $\alpha_t = \alpha = 0.25$ for class 1, $\gamma = 2$.

Examples of a trained B with the Lorenz loss (7) for $d = 5$ and some steered filters obtained from this B are shown in Figure 4.

Observe that because of linearity, the convolution of an image I with the steered filter $\mathbf{f}(\theta)$ is a linear combination with weight $\mathbf{w}(\theta)$ of the convolutions with the filters from B,

$$\mathbf{f}(\theta) * I = (B \cdot \mathbf{w}(\theta)) * I = (B * I) \cdot \mathbf{w}(\theta). \tag{11}$$

2.2. The Steerable CNN

The steerable CNN, illustrated in Figure 2, consists of a number of layers containing 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 kr responses corresponding to each group using the weight vector $\mathbf{w}(\theta)$ from Eq. (5).

2.3. Training the Steerable CNN

Suppose we are given n training examples $(\mathbf{x}_i, y_i, \alpha_i), i = 1, ..., n$ where $\mathbf{x}_i \in \mathbb{R}^{p^2}$ is the patch of size $p \times p$ either with center on the guidewire (a positive example) or at some distance 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 Spherical Quadrature Filter (SQF) [19]. The SQF is also used as a preprocessing step for detection, so the training examples are extracted only from locations with high SQF responses. 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 5.

Similar to section 2.1, 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 paper 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 [16]. 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 (5) 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 4-layer steerable CNN is show in Figure 6.

2.4. Implementation Details

CNN architecture. The Steerable CNN for this task consists of 4 steerable convolutional layers. The third steerable convolutional layer is followed by ReLU activation, and the

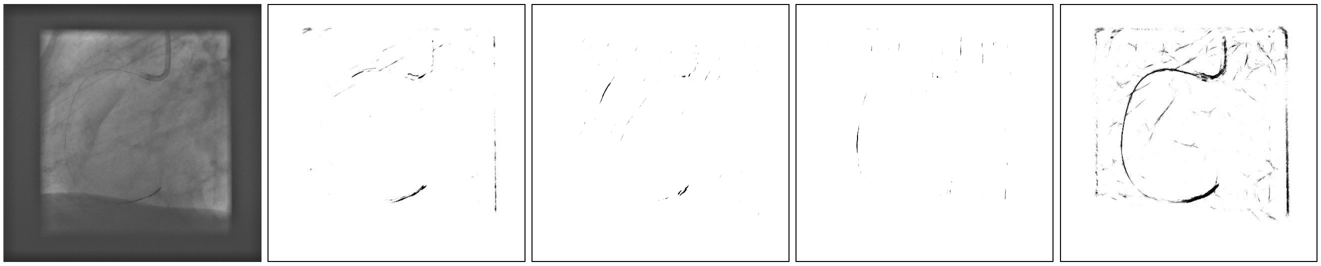

Figure 5. Examples of one frame Steerable CNN detection results: input image, response map for angle index $j = 5, 10, 15$, and final detection result.

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
```

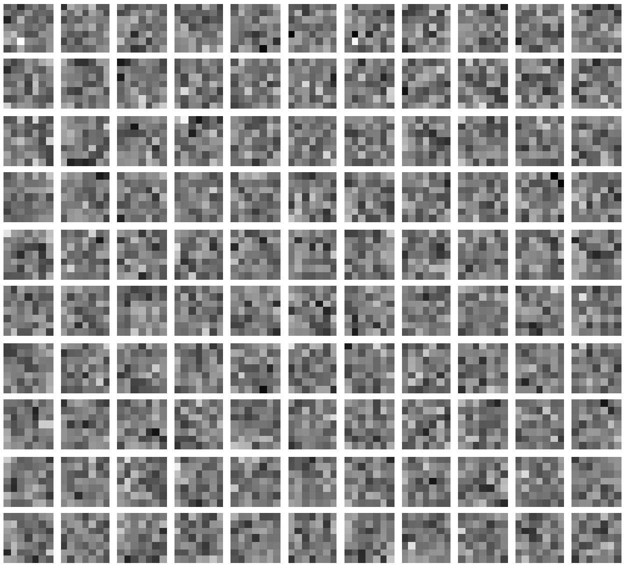

Figure 6. Trained basis B for the first layer of the rank 11 steerable CNN.

last one returns the response.

The 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. The receptive field of the SCNN is of size 25×25.

We also implemented a Fully Convolutional Network (FCNN) for comparison. The network consist of 4 convolutional layers, the third one is followed by ReLU, and the last convolutional layer returns a guidewire/non-guidewire response.

The first convolutional layer for a 25×25 receptive field of size contains 16 filters size 7×7, the next two convolutional layers contain 32 7×7 filters, and the last convolutional layer contains one 7×7 filter.

For both the FCNN and the steerable CNN we used the Pytorch[21] Soft Margin Loss

$$\ell(u, y) = \frac{1}{m} \sum_i \log(1 + \exp(-y_i u_i)) \tag{12}$$

to guide the training, where m is the minibatch size.

We also used the Focal loss [18] in Eq 8, since it can deal with the class imbalance.

Training details. The weights of convolutional layers were initialized from the normal distribution with standard deviation 0.01. For the Steerable CNN of rank 7 and rank 11, we started with a learning rate of $3 \cdot 10^{-6}$ and a mini-batch of 32. For training the FCNN, we started with learning rate 10^{-5} and mini-batch 32. After every 50 epochs the learning rate was multiplied by 0.8 and the minibatch was doubled. The training was done for a total of 300 epochs.

3. Experiments

Dataset. The evaluation results are conducted on 75 fluoroscopic sequences obtained during coronary angioplasty intervention. The sequences contain a total of 826 frames of various sizes and aspect ratios in the range $[512, 1024] \times [512, 960]$. In the 75 fluoroscopic sequences, 39 sequences are used for training, with 424 frames in total, and the remaining 36 sequences with 402 frames are used for testing. The guidewire annotations of each frame were obtained using B-splines. An examples of B-spline annotation is shown in Figure 7.

SQF NMS Alignment In [17], the authors observed that the guidewire annotation is imprecise because the wire is

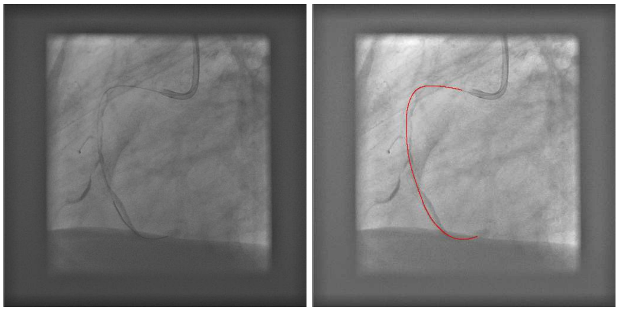

Figure 7. Example of input frame (left), and its B-spline annotation (right).

so thin and barely visible. For that reason we decided to use the Cauchy SQF filter followed by non-maximum suppression in the style of the Canny edge detection [6] as a preprocessing step.

Training examples. The size of the training patches are of size 25×25 for both the positive and negative examples.

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 above. 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 has 279,000 positives and 578,000 negatives. The training set with SQF NMS alignment has 193,000 positives and 535,000 negatives.

3.1. Angle estimation Experiments

In a first experiment, we evaluate the accuracy of different methods in estimating the guidewire angle. First, we show in Figure 8 the average responses for different methods on positive patches with true angle 60 degrees. We see that all methods have a peak at 60 degrees, and some other smaller peaks.

The evaluation of the Frangi filter, Cauchy SQF, Trained steerable filters of ranks 5-11 trained with different losses and the rank 11 steerable CNN on the training and test positive patches is shown in Table 1. We see that the trained steerable filters have better accuracy than the predefined steerable filters. Also, the steerable filters trained with NMS-aligned patches have a much better accuracy than those trained on non-NMS aligned patches.

3.2. Guidewire Detection Experiments

We present an evaluation of the pixelwise guidewire detection for both filter-based methods and learning based methods. As filter-based approaches we evaluated the popular Frangi Filter [12], as well as the Spherical Quadrature Filters (SQF) [19] with different types and ranks (dimension of the basis), and we show the performance of the best rank with different isometric filters. As learning based methods we compare the steerable CNN, the FCNN, and trained 25×25 steerable filters with the Lorenz loss[3] and Focal loss[18], which can be considered as using only the last layer of the steerable CNN, with a larger filter size. We also implemented [8] using about 100,000 oriented Haar features and PBT [24], and trained it on our training set without SQF NMS alignment.

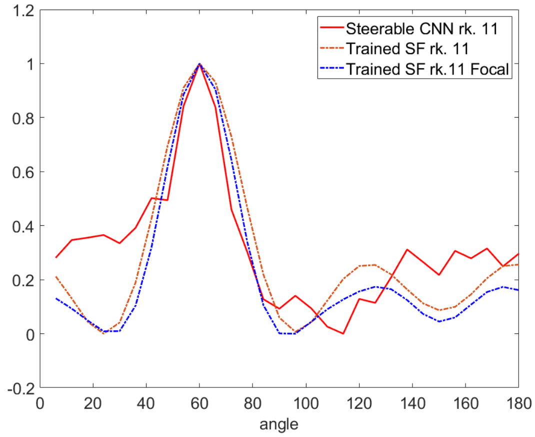

Figure 8. Average responses for test patches with angle 60°.

Method	Angle Error Train	Test
Frangi filter [12]	13.16	13.69
SQF [19] Cauchy, $f_0 = 1/6$, rk. 11	7.58	7.13
Steerable filter (6), rk. 5	8.89	9.72
Steerable filter (6), rk. 7	4.49	5.20
Steerable filter (6), rk. 9	4.85	5.69
Steerable filter (6), rk. 11	5.12	6.02
Steerable filter (6), rk. 11 w/ NMS	**2.14**	**2.42**
Steerable filter (6) Focal[18], rk. 11	6.62	7.51
Steerable filter (6), Focal [18], rk. 11 w/NMS	2.63	2.78
Steerable CNN, rk. 11 w/ NMS (Focal [18])	4.68	6.70

Table 1. Average angle estimation error (degrees) of different methods on 25×25 patches.

The guidewire detection examples are shown in Figures 9 (one frame from the train set) and 10 (one frame from the test set). A threshold was chosen to obtain the results with an average detection rate of 90% from the response map. Note that the input image shown in Figure 10 is noisier than the one from Figure 9.

The detection performance was evaluated on the training and test sets. The detected guidewire pixels were those that had a corresponding response above the detection threshold at a distance of at most 2 pixels. A response above the detection threshold was considered a false positive if it

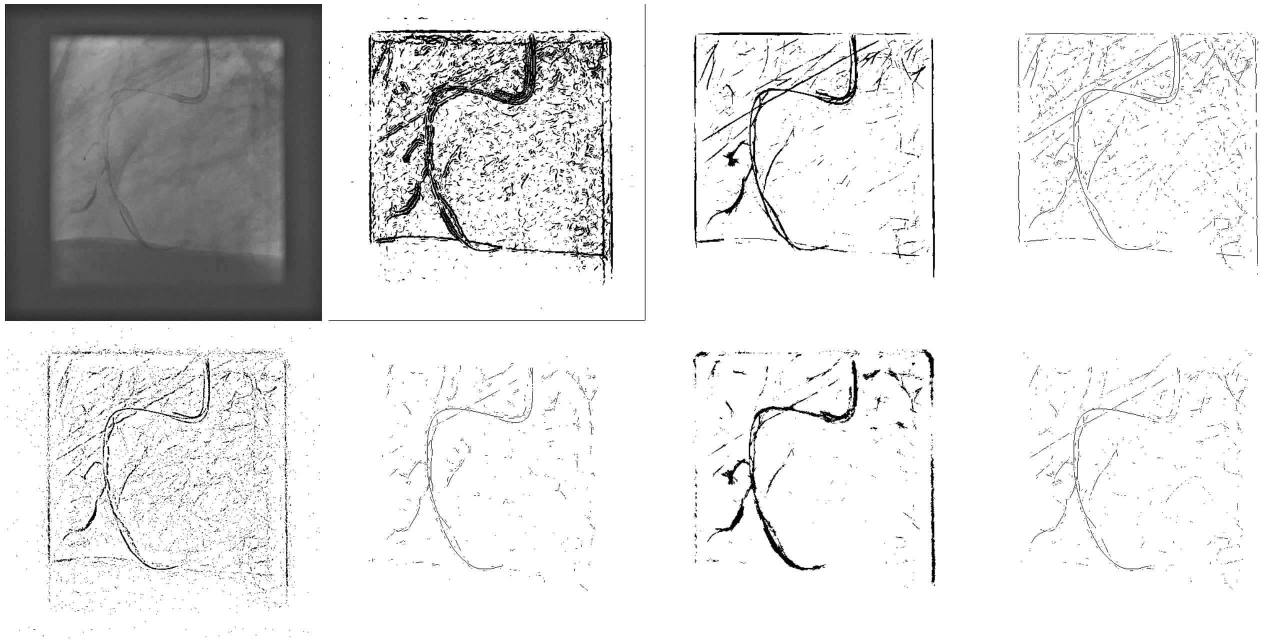

Figure 9. Guidewire detection **training** examples. First row: input image, Frangi filter [12], Cauchy SQF [19] of rank 11, trained Steerable filters(Focal loss[18] w/ NMS) rank 11. Second row: PBT with Haar features[2, 20, 26, 8], FCNN w/ NMS-aligned training examples, Steerable CNN of rank 11 w/o NMS-aligned training examples, Steerable CNN of rank 11(Focal loss[18]).

	Det. rate		FP rate		# of trained
Method	Train	Test	Train	Test	parameters
Frangi Filter [12]	90.44	90.44	26.99	24.19	-
SQF [19] Gauss deriv, f_0=1/2, rk. 7	90.00	90.08	6.92	7.04	-
SQF [19] Cauchy, $f_0 = 1/6$, rank 7	89.98	90.00	5.98	6.35	-
SQF [19] log-Gabor, $f_0 = 1/6$, rk. 11	90.04	90.03	5.32	5.21	-
SQF [19] Cauchy, $f_0 = 1/6$, rk. 11	90.13	90.02	5.12	5.87	-
SQF [19] Cauchy, $f_0 = 1/6$, rk. 11 w/ SQF NMS	90.03	90.00	4.19	3.93	-
Trained equivariant SFCNN Filters [27] w/ SQF NMS	89.99	90.00	6.31	6.46	0.12k
Trained rank 11 steerable filter, FB loss (6) + Lorenz loss[3] w/ NMS	90.09	89.98	2.48	2.94	6.9k
Trained rank 11 steerable filter, FB loss (6) + Focal loss[18] w/ NMS	90.01	90.11	2.91	3.44	6.9k
PBT and Haar features[2, 20, 26, 8]	90.07	90.19	3.87	3.98	8.4k
2-layer Steerable CNN rank 11, FB loss (6) + Lorenz loss[3] w/ NMS	89.78	89.96	2.17	2.89	18.7k
FCNN with NMS-aligned training examples	90.07	90.08	1.43	2.65	78k
4-layer Steerable CNN rank 7 w/ SQF NMS (Soft Margin loss (12))	90.08	90.02	0.94	2.01	217k
4-layer Steerable CNN rank 11 w/ SQF NMS (Soft Margin loss (12))	90.05	90.05	0.78	1.90	341k
4-layer Steerable CNN rank 11 w/ SQF NMS (Focal loss [18])	90.18	90.10	**0.76**	**1.82**	341k
FCNN w/o NMS-aligned training examples	90.09	90.01	3.72	8.28	78k
4-layer Steerable CNN rank 7 w/o SQF NMS (Soft Margin loss (12))	89.96	90.01	3.34	6.69	217k
4-layer Steerable CNN rank 11 w/o SQF NMS (Soft Margin loss (12))	90.01	90.07	2.95	5.94	341k
4-layer Steerable CNN rank 11 w/o SQF NMS (Focal loss [18])	90.07	90.01	2.92	6.08	341k

Table 2. Evaluation of different guidewire detection approaches.

was at distance of at least 3 pixels from the guidewires and catheters.

In Table 2 are shown the average per-image detection rates and false positive rates for the different methods evaluated. The steerable CNN obtains the lowest false positive rate on both the training and test set. It outperforms all the other methods. Among the filter-based methods, the SQF with a Cauchy filter of rank 11 with NMS performs the best, but it is outperformed by all the training based methods. The FCNN with NMS training examples has a very small training error but it does not generalize as well as the steerable CNN.

Ablation study. We also show in Table 2 the influence of training the FCNN with the NMS-based training examples vs examples extracted directly based on the annotation. We see that both training and test FP rate are lower using the

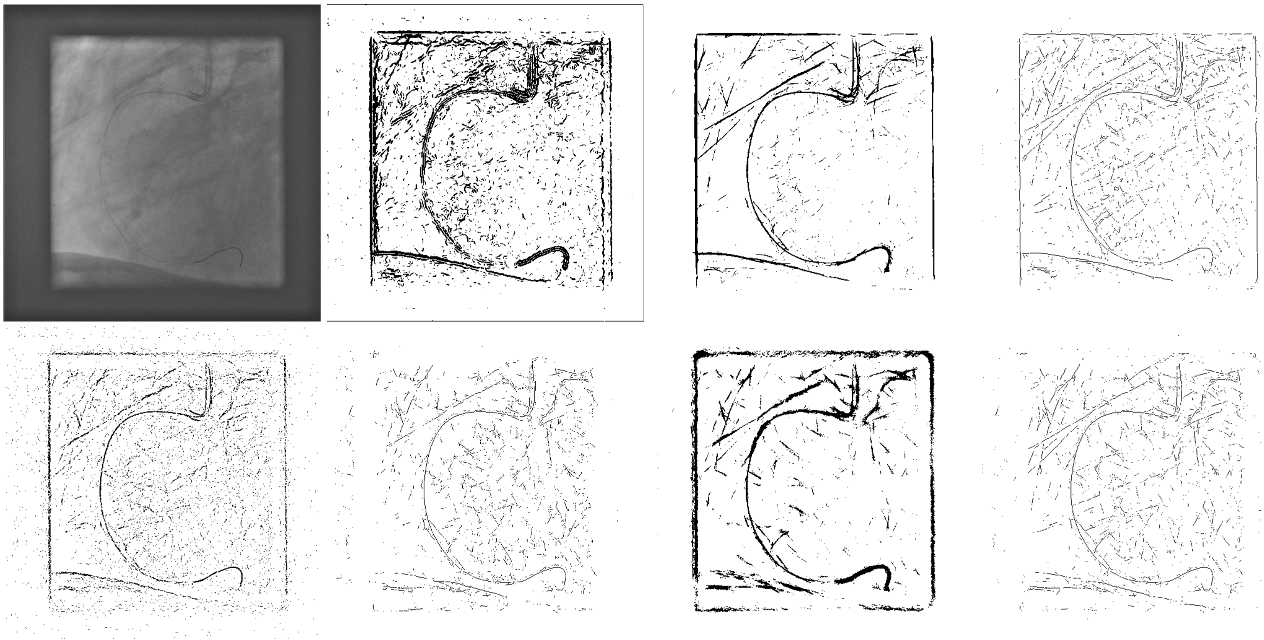

Figure 10. Guidewire detection **test** examples. First row: input image, Frangi filter [12], Cauchy SQF [19] of rank 11, trained Steerable filters(Focal loss[18] w/ NMS) rank 11. Second row: PBT with Haar features[2, 20, 26, 8], FCNN w/ NMS-aligned training examples, Steerable CNN of rank 11 w/o NMS-aligned training examples, Steerable CNN of rank 11(Focal loss[18]).

NMS-aligned examples.

To see whether the SQF are useful in screening the image and proposing the angle for the steerable CNN, we also evaluated in Table 2 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 NMS). Again we see that the SQF-based screening is useful, reducing the test FP rate from 6.08 to 1.82.

4. Conclusion

In this paper, we introduced a simple steerable CNN that can be tuned using a parameter θ to be sensitive to objects aligned to any orientation θ, instead of being rotation invariant.

We presented the mathematical formulation of the trainable steerable filters and the steerable CNN, and how to train it using examples at any orientation, without rotating them for alignment. As an 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 receptive field.

We reported the difficulties we encountered while training the steerable or the regular CNN due to the fact that the guidewire is thin and noisy, and imprecision in annotation makes the training more difficult. To address these issues, we explained how to obtain better aligned training patches using the Spherical Quadrature Filters [19] and non-maximal suppression.

Experiments reveal that the Steerable Convolutional Neural Network trained on SQF NMS-aligned data performed the best. We also observed that the learning based methods outperform the filter-based methods such as the Frangi filter [12] and the Spherical Quadrature Filters [19].

For the future study, we plan to employ the steerable CNN for automatic guidewire localization and for retina vessel detection. The guidewire localization is the higher-level process of finding in the image the entire guidewire as a curve. It will use the guidewire detection response as a data term to guide the search in the high dimensional space of smooth curves for the most likely guidewire location. The retina vessel detection problem has the added challenge that the vessel can have a wide range of widths, and a classifier should accommodate them in some way. It would be interesting to see whether a steerable CNN could be designed to be steered in both the orientation and the width of the vessel that needs to be detected.

References

[1] 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. 2

[2] 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. 2, 7, 8

[3] 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. 4, 6, 7

[4] 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. 2

[5] 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. 2

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 6

[7] 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. 2

[8] Terrence Chen, Gareth Funka-Lea, and Dorin Comaniciu. Robust and fast contrast inflow detection for 2d x-ray fluoroscopy. *MICCAI*, pages 243–250, 2011. 2, 6, 7, 8

[9] 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. 2

[10] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016. 3

[11] Taco S Cohen and Max Welling. Steerable cnns. *ICLR*, 2017. 2

[12] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *MICCAI*, pages 130–137, 1998. 2, 6, 7, 8

[13] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Trans. on PAMI*, 13(9):891–906, 1991. 2, 3

[14] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011. 1, 3

[15] Nicolas Honnorat, Régis Vaillant, and Nikos Paragios. Guide-wire extraction through perceptual organization of local segments in fluoroscopic images. *MICCAI*, pages 440–448, 2010. 2

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[17] 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. 2, 5

[18] 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. 4, 5, 6, 7, 8

[19] Ross Marchant and Paul Jackway. Feature detection from the maximal response to a spherical quadrature filter set. In *DICTA*, pages 1–8. IEEE, 2012. 2, 4, 6, 7, 8

[20] 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. 2, 7, 8

[21] 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. 5

[22] Olivier Pauly, Hauke Heibel, and Nassir Navab. A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences. *MICCAI*, pages 343–350, 2010. 2

[23] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017. 3

[24] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV*, volume 2, pages 1589–1596. IEEE, 2005. 2, 6

[25] 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. 2

[26] 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. 2, 7, 8

[27] 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. 2, 7