GUIDEWIRE POSITION AND DIRECTION ESTIMATION IN FLUOROSCOPIC IMAGES

Krishnan Venkataraman
{kvenkat9@jhu.edu}

ABSTRACT

A guidewire is a medical wire inserted into the body during surgery to assist insertion of larger instruments and provide positional heuristics in real time. The accurate tracking of the guidewires requires longer exposure time on a patient and also due to anatomical and occluding hindrances in the environment the tracking becomes quite difficult. For these problems, in this report, proposing an automated deep learning-based solution that can be useful in locating a guidewire and detecting its direction and position. The proposed solution is built on two stages . Stage 1 - to identify a guidewire tip and approximately bound it , Stage 2 - to predict the direction and position of the guidewire from the approximate bound. The solution proposed is developed and evaluated on different surgical sites data and the results are produced in the report. .

Keywords guidewire · automated · position · direction · deep-learning

1 Introduction

A guidewire is a thin, flexible, medical wire inserted into the body to guide a larger instrument, such as a catheter, central venous line, or feeding tube[2]. Guidewires are majorly used in orthopaedic surgeries as guiding heuristics for inserting and screw fixation tasks[1]. Surgeons identify the position and direction of these guidewires to navigate and perform the major procedure.

In current standard of care, the accurate positioning of guidewires often involves trial-and- error and long fluoroscopic exposure times[1]. To avoid these long exposure times on patients and to assist the surgeons better **there is a need for an automated system to locate these guidewires and to provide real time tracking of direction and position of these guidewires**.

Tracking of these guidewires is quite challenging because the guidewires are inside a patient's body during surgery and in real time surgical data, we expect a lot of occlusion and fuzziness in the images. Apart from the occlusions, since the guidewires can be used in multiple sites, the anatomical configurations of each body part are different which could contribute different kinds of artefacts to the captured fluoroscopic images in different ways.

These challenges make it hard to use traditional image segmentation or object identification algorithms. AI/ML based solutions are better suited to address these problem because they were designed to perform well in **object detection and pose identification tasks with similar challenges such as occlusions**. Hence, for this medical imaging application going forward with an AI/ML based solution is a better choice.

In this report, the proposed solution is a two staged AI/ML based solution. Stage 1 uses a Faster RCNN deep framework to locate and bound guidewire tips while stage 2 uses a resent101 based deep framework to predict the angle and offset of the guidewire tip using the bounds from stage 1. The solution as a whole is used to estimate the position (in pixels) and direction(in angle) of the guidewires in a given fluoroscopic image.

2 Materials

Dataset:
The dataset provided is as a single .npz file that contains images and ground truth annotations of guidewire positions

and directions. Images were acquired from multiple cadaveric specimens, captured at multiple anatomical sites (pelvis, lumbar spine, thorax, and shoulders), and were acquired using a mobile C-arm (Siemens Cios Spin, Erlangen, Germany) [1].

The npz file contains ‘images’, ‘position’ and ‘direction’ components.

‘images’ Component contains,

Total Number of Images = 314

Image dimension = 976 x 976

Each Image has 2 guidewires

‘position’ and ‘direction’ components contain,

Position for each guidewire in the corresponding image – **x and y components**

Direction of each guidewire in the corresponding image - **displacement delta x and delta y components**

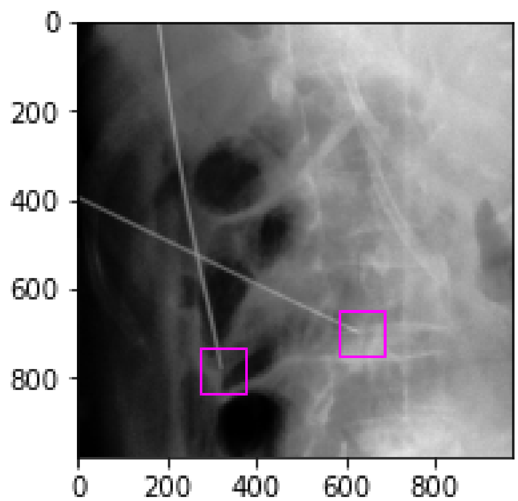

Figure 1: Guide Wire in a fluoroscopic image. Guide wire tip bounded in magenta color box

Splitting the dataset into Train-Validation- Test:
The entire dataset is randomly divided into train , validation, and test sub-datasets. The Train dataset consists of 70 percent of the data, the Validation dataset consists of 10 percent of the data and the Test dataset consists of 20 percent of the data. Since all the images are from different anatomical sites and are distributed without any order information, random sampling is performed for the train-val-test splitting.

Total Images in Train = 230

Total Images in Validation = 21

Total Images in Test = 63

Structuring the dataset for the two stages:
As discussed previously, the entire pipeline is divided into two stages (Guidewire detection and Angle-Pos prediction).

For Stage 1,

The images in the dataset are retained in their original size (976 x 976). Two bounding boxes are created as ground truths for each image. A single bounding box takes the position of the guidewire tip as centre and has a width and height of 101x101. An odd number (101) is chosen as width and height to get a proper centre pixel.

Total Images in Train for Stage 1 = 230

Total Images in Validation for Stage 1= 21

Total Images in Test for Stage 1 = 63

For Stage 2,

The bounding boxes created in stage 1 are used to generate dataset for stage 2. For each image, two new images of size 101 x 101 are created by slicing the original image at the bounding box. Before slicing the image at the bounding box, a random offset of +/- 15 pixels is performed. This is done, so that the models built in the future are capable of learning the needed goal even with a centre offset. For train, validation and test sets we have a new dataset each. Hence,

Total Images in Train for Stage 2 = 460

Total Images in Validation for Stage 2= 42

Total Images in Test for Stage 2 = 126

For both Stage 1 and 2, the images undergo a pre-processing step which will be discussed in the methods section.

3 Methods

Stage 1:

The goal of this stage is to detect the guide wire tips in a given fluoroscopic image. For the object detection tasks the below processes were followed,

1. Data pre-processing

2. Building network architecture

3. Training the model and optimizing through experiments

Data pre-processing:
The images in the dataset had different intensity distributions overall. So, to adjust the intensity variation and keep all images in a standard contrast, histogram equalization was performed. The histogram equalized images were then standardized to a range of 0 to 1 by performing min-max scaling. The guidewires are darker while the background is lighter in contrast. Usually, objects that are lighter in comparison with background are picked up well by neural network frameworks. Hence to assist the model training better the image was inverted. Now the guidewires are brighter in contrast with the background. After that each image was duplicated twice and stacked together so that they form three channels like structure.

Model Architecture:
For the model architecture, pretrained FasterRCNN–Resnet50 model from torch vision was used. The box predictor layer of the architecture was modified to accept two output classes to detect whether or a not a region is a guide wire tip. Since FasterRCNN architecture is faster as well as quite robust, it seemed to a desirable choice for this application.

Model Training and Optimization:
The model was trained with stochastic gradient descent SGD optimizer with the four different loss functions default for faster RCNN networks. Hyper parameter tuning was performed using the validation dataset and finally the below parameters were found to be the best parameters for learning,

Learning rate = 0.001

Momentum = 0.9

Number of epochs = 50

Score threshold after sweeping through multiple thresholds = 0.5

The stage 1 train figure shows train– val loss curves during the training,

We see that both the training and validation losses are coming down from a higher value and are saturating. There is no abnormal overfitting or underfitting trend observed from the loss curves.

.

Stage 2:

The goal of this stage is to detect the angle and centre offset of the guide wire tips. For this task the below processes were followed,

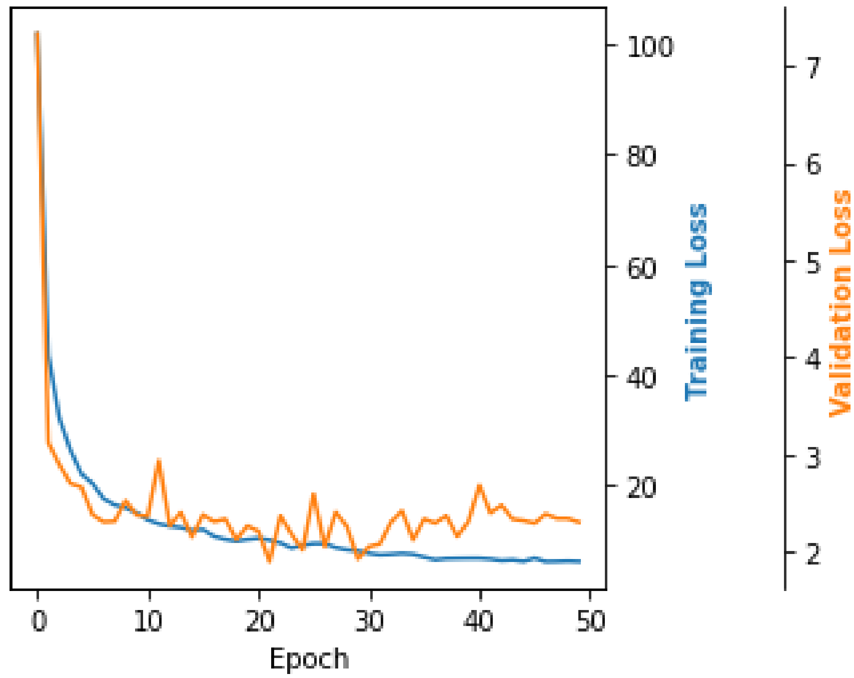

Figure 2: stage 1 training

1. Data pre-processing
2. Building network architecture
3. Training the model and optimizing through experiments

Data pre-processing:
The images were made to undergo the same pre-processing steps. Histogram equalization was performed followed by min-max scaling and then inversion. The parameters for histogram equalization and min-max scaling were extracted from the initial higher dimension images used in stage 1.

Model Architecture:
For the model architecture, pretrained Resnet101 model was taken from torch vision. The input layer of the architecture was modified to accept single channel input and the output layer of the architecture was modified to produce four outputs. First two outputs were designed to produce the sine and cosine parts of the angle and the last two outputs were designed to get the x and y offsets from the centre, respectively. ReLU activation function was applied to the last output layer. But since the first two outputs are desired to be sine and cosine terms, those outputs are again passed through a tanh activation function so that the range is from -1 to 1.

Reasoning for the architecture:
Resnet 101 architecture seemed to be a desirable choice since resents were used in literature to calculate the poses of objects (which was similar to our application) and resnet 101 performed better than resnet50 in our application.

Predicting sine and cosine terms instead of the angles directly because the angles have a circular nature, meaning after 359 degrees the rotation is back to 0 . So circular functions like sine and cosine could be helpful in dealing with this situation.

The inputs architecture was modified to accept single channel input because multiple experiments were conducted with different parameters and three channel inputs with custom normalisation of mean and std, but the single channel input yielded the best loss and validation results.

Model Training and Optimization:
The model was trained with stochastic gradient descent SGD optimizer with customised loss function.The loss function designed is shown in the loss function figure.

Hyper parameter tuning and optimization was performed using the validation dataset and finally the below parameters were found to be the best parameters for learning,

Learning rate = 0.001

Momentum = 0.9

Number of epochs = 300

Cost for x pos in loss function = 0.1

Cost for y pos in loss function= 0.1

Cost for Sine term in loss function= 1

Cost for cosine term in loss function = 1

The stage 2 train figure shows the train– val loss curves during the training,

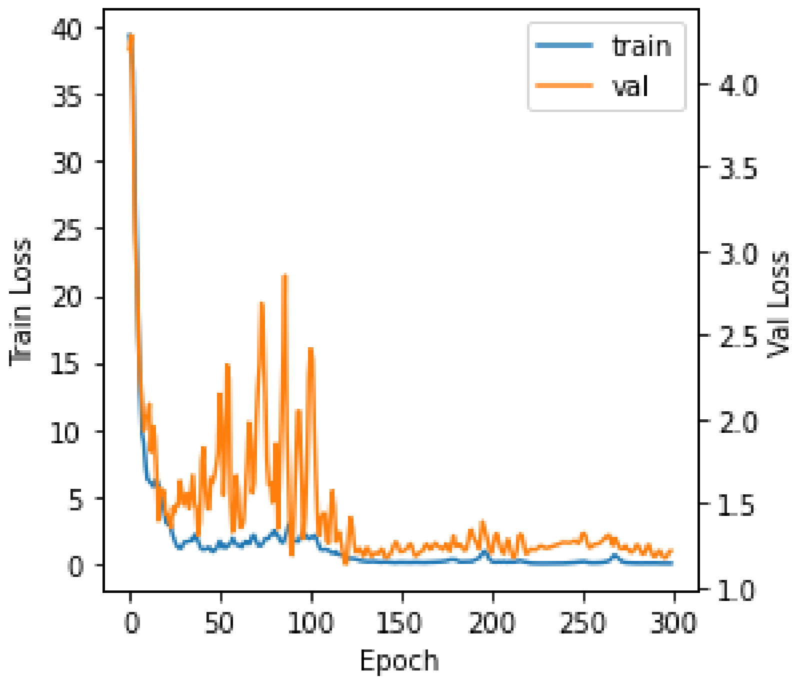

Figure 3: stage 2 training

We see that both the training and validation losses are coming down from a higher value and are saturating after some fluctuations in the middle epochs. There is no abnormal overfitting or underfitting trend observed from the loss curves.

Integrating Stage 1 and Stage 2 – Automated Solution Pipeline:

The models in stage 1 and stage 2 are saved. The automated solution combines both stage 1 and stage 2 models. When a fluoroscopic image is given, it is pre-processed, and then stage 1 model is applied. Stage 1 gives us the bounding boxes for guide wire tips. These bounding boxes are cropped from the pre-processed image, and they are fed to stage 2 model. For each guidewire, stage 2 model gives its respective angle and centre offset. The centre offset is added with the bounding box absolute centre co-ordinates to get the absolute position of the guide wire. Hence for each guide wire we get a direction in angle and position in pixel.

4 Results:

The evaluation of the solution is based on three metrics, 1. Correct-Wrong-Missed guidewires 2. Mean and Median Euclidean distance errors 3. Mean and Median Angular deviation errors

Correct-Wrong-Missed guidewires percentage :

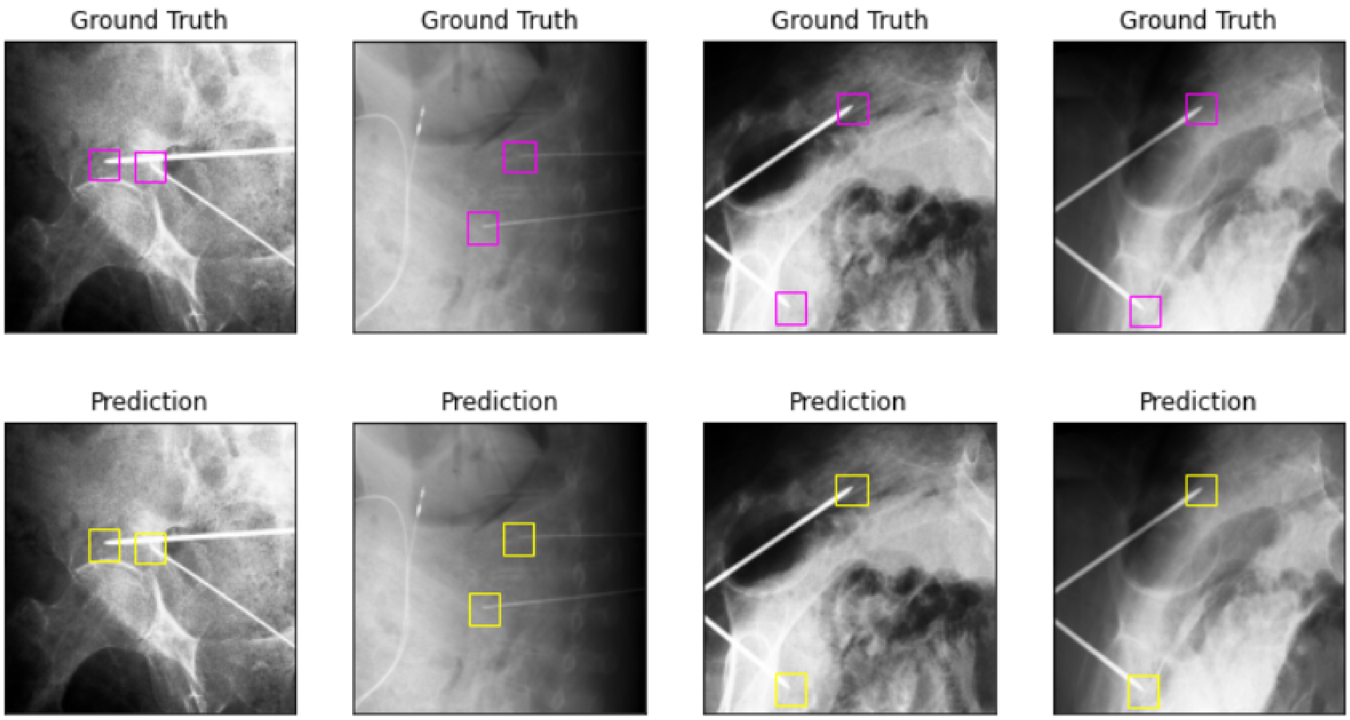

Figure 4: stage 1 results

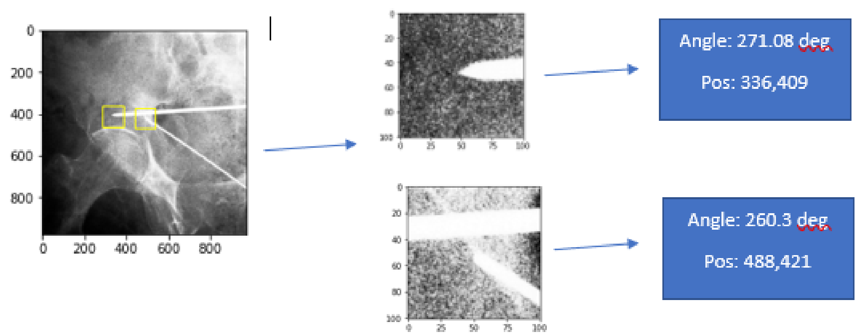

Figure 5: Full pipeline result

This metric is to calculate how many guidewires are identified correctly by the solution, how many regions are wrongly identified as guidewires and how many guidewires are missed.

This is majorly to evaluate the FasterRCNN part, to check whether the bounding boxes are on rightly on the guidewires. Intersection over union IOU measure is used to check whether a bounding box is on a guidewire,

If IOU > 70 percent - then the box is considered as correct. If is less than 70 percent then wrongly identified. If a guidewire ground truth does not have a box at all then it is considered missed. Seventy percent is set because it is related to the random offset, we used in stage 2 dataset generation.

The Correct-Wrong-Missed guidewires results figure shows the evaluation results with respect to this metric,

Correct-Wrong-Missed guidewires			
	Train	Validation	Test
Total Boxes	460	42	126
Correct	450	39	117
Wrong	0	0	0
Missed	10	3	9
Correct-Wrong-Missed guidewires %			
	Train	Validation	Test
Total Boxes	100	100	100
Correct	97.82	92.85	92.85
Wrong	0	0	0
Missed	2.17	7.14	7.14

Figure 6: Correct-Wrong-Missed guidewires results

We see that there some guidewires that are missed. For the automated solution to be more robust we could introduce some rules like minimum two guidewires are to be predicted otherwise discard the prediction.

Mean and Median Euclidean distance errors:

Euclidean distance between the predicted position and the GT position of the guide wire is calculated. For each guidewire that is bounded by the box we calculate Euclidean error. For the entire train, val and test sets , for all the guidewires predicted in them we calculate Euclidean error and then we generate the mean and median for the collected errors.

The ED error results figure shows the evaluation of the solution with respect to this metric,

Mean and Median Euclidean distance errors in pixels			
	Train	Validation	Test
Mean	3.99	18.72	8.53
Median	3.32	3.68	4.41

Figure 7: ED error results

Mean and Median Angular deviation errors:

Absolute Angle difference between the predicted angle and the GT angle is calculated. For each guidewire that is bounded by the box we calculate angular deviation error. For the entire train, val and test sets , for all the guidewires predicted in them we calculate angular deviation error and then we generate the mean and median for all the collected errors.

The AD error results figure shows the evaluation of the solution with respect to this metric,

Mean and Median Angular deviation errors in degrees			
	Train	Validation	Test
Mean	9.95	8.47	11.64
Median	7.15	7.16	7.92

Figure 8: AD error results

5 Conclusion

From the results, we see that, given a fluoroscopic image the proposed solution is capable of identifying the guidewire tips with 92.85 percent accuracy , measure the position under +/- 8.53 pixels (on an average) and predicted the direction in +/- 11.64 degrees (on an average). The overall solution looks decent and with some auxiliary rules and safety algorithms the solution could be put to use in real time or simulated scenarios for testing and refinement.

References

[1] Project description document

[2] Blog: medicaldesignbriefs medicaldesignbriefs.com/component/content/article/mdb/features/applications/23536

[3] Designing Deep Convolutional Neural Networks for Continuous Object Orientation Estimation by Kota Hara, Raviteja Vemulapalli and Rama Chellappa - https://arxiv.org/pdf/1702.01499.pdf

[4] Geometric loss functions for camera pose regression with deep learning by Alex Kendall and Roberto Cipolla - https://arxiv.org/pdf/1704.00390v2.pdf

[5] Blog: wayfair https://www.aboutwayfair.com/tech-innovation/shot-angle-prediction-estimating-pose-angle-with-deep-learning-for-furniture-items-using-images-generated-from-3d-models