A U-Net Framework for Detecting K-Wires in Fluoroscopic Images
IE: 6380 - Deep Learning, Spring 2019

Josh Friede
University of Iowa
joshua-friede@uiowa.edu

Johnny Golec
University of Iowa
john-golec@uiowa.edu

Dominik Mattioli
University of Iowa
dominik-mattioli@uiowa.edu

Abstract

This work details the application of deep learning techniques to advance the image analysis of an orthopedic surgery simulator project. The methods applied include a prominent neural network architecture – U-Net – and the recently-proposed concept of using synthetic images to train a network for detecting objects in real-world images. While there is additional future work needed, the preliminary results are promising, and the implications exciting.

1. Introduction

1.1. Background

A surgery simulator project underway at The University of Iowa is focusing on simulating the navigation of a 'wire' through soft tissue and bone. This team's simulators currently mimic two surgeries defined by Wire Navigation: The Dynamic Hip Screw for femoral head fractures (DHS), and The Pediatric Supracondylar Humerus Fracture (PSHF). This project is one example of modern research efforts aiming to improve surgical residents' skills before entering the Operating Room — a high-risk and high-cost environment. By enhancing skills in a simulated setting and challenging the traditional paradigm of "on-the-fly" training hands-on experience, risks and costs may decrease.

A current gap in the orthopedic surgical simulator research community is the establishment of metrics that unambiguously quantify surgical skill. Motivations for the development of these simulators are multi-fold, but primarily focus on driving down of costs associated with training competent surgeons for tasks which are extremely costly in time and resources — both mental & physical[7].

Given the complexity of these procedures and the (motor) skills and situational awareness required to perform them adequately, there is a wealth of information to be had in the analysis of the decisions made by surgeons as they perform the operation. However, this information can often be lost during the standard evaluations of performance.

Generally, surgeons performing these surgeries are trying to minimize three things: drilling, duration, and radiation (fluoroscopy). Beyond accounting for these three variables, other metrics of "success" in orthopedic surgery are less clear and in need of development.

One specific aim of this team's project (as stated on the project proposal) is to "define simulator-based assessments that generalize to clinical performance", that is, a quantification of performance independent of OSATS (Objective Structured Assessment of Technical Skills) assessment — the current standard for evaluation of surgeries[6]. An issue with OSATS is that they can be neither unambiguous nor immune to simple biases, as they are a subjective assessments resembling a Likert Scale. The surgical simulator team's two focus surgeries — DHS and PSHF — both utilize the OSATS for performance assessment.

Additionally, both surgery procedures use fluoroscopic images (fluoros) to track progress as the surgeon navigates a K-Wire ("wire") through bone. The location of the wire's tip relative to its final destination is important because studies show that long-term recovery is contingent on an ideal wire placement[4, 8]. By tracking the surgeon's navigation of the wire's tip across multiple fluoros, a better understanding of their decision making and consequently their performance may be had. The caveat: acquiring this understanding will require a large set of image data — which can be very time consuming to process by hand — so a more efficient mechanism is desired to efficiently process these fluoros.

1.2. Motivation for Applying Deep Learning

To date, there are 3,252 fluoros — from 16 DHS surgeries and 15 PSHF surgeries — available (in-house) to this research team. Processing these images involves manually labeling the location of the wire and other relevant anatomical features (depending on the surgery) in a Graphical User Interface (GUI), as seen in Figure 1. While this process is time consuming, it is also a candidate for the application of deep learning (computer vision) techniques. The implementation of a deep learning model that predicts the loca-

tion of these features will greatly enhance the feasibility of this project. By reducing the time spent processing fluoros, more individual cases may be added to our cohort and analyzed in total.

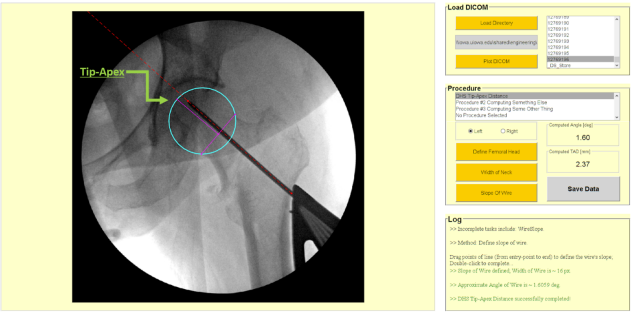

Figure 1. Example of the processing of a DHS fluoro via GUI, with the wire labeled with a red line, the femoral head labeled with a cyan ellipse, and the femoral neck labeled as a magenta line (with a bisector to indicate trajectory). The 'Tip-Apex' is defined as the intersection of the femoral neck bisector with the femoral head. While this GUI is designed for user-friendliness, it can be tedious to use for extended periods of time.

1.3. Problem Statement

As a proof of concept, this paper focuses on building a model solely for semantic segmentation of wires in fluoros. The issue at hand, as mentioned previously, is that processing these images by hand is quite time consuming. Thus, there is little reliable ground truth for this model to train a model on.

To our knowledge, there are no papers dedicated to automated strategies for detecting a wire in fluoros, nor are there papers describing the problem of manually performing such a task. Fortunately, a study by Tremblay *et al*. revealed some merit to training computer vision models with synthetic image data and receiving high accuracy on validation image data[10]. This work hopes to corroborate these results by training a model using only synthetic image data and then validate the model with real images, thus solving the issue of little available ground truth data.

2. Methods

2.1. The Data

The methodical process described by Tremblay *et al*. for generating the synthetic data focuses on the concept of Domain Randomization (DR). Tremblay *et al*. define DR as "an approach that intentionally abandons photo-realism by randomly perturbing the environment in non-photorealistic ways (*e.g*. by adding random textures) to force the network to learn to focus on the essential features of the [object]".

This work maintains the core concept of DR, but simplifies the methods used by Tremblay *et al*. to remain as a basic proof of concept. The simplified methodology is as follows:

1. Acquire a source image data set.
2. Acquire 1 wire template (binary mask).
3. Overlay several augmentations of the template to each source image to create a synthetic data set.
4. Train and test the model on 99% of the synthetic images.
5. Validate the model on 1% of the synthetic images and a handful of real images.

2.1.1 Source Image Data

Of the surgery simulator team's 3,252 fluoros, 385 of the images contain no visible wire. These 385 images form the basis of the source image data set. To increase variability, 385 additional images are randomly selected from 7,470 chest X-Ray images of the MedPix Database and added to the source images[3]. To complete the DR, 770 images are randomly selected from THe Coco Data Set's 5,000 random images[1]. This collection of images balances both random images (50%) with two specific medical image modalities (25% each) to ensure the model learns the distinct features of the wire irrespective of the context of its background and any signature features they may have.

2.1.2 Data Augmentation Techniques

A binary mask represents the pixel locations of a wire in the template seen in the bottom right (10th) image of Figure 2. This 1 template may be morphed 9 times to similar — but different length — wire templates that represent the various lengths of wire visible in real fluoros. From these 10 templates (resized to [256x256] to decrease computation), an augmented template is created and overlaid onto the source images. Each source image (1,540 images) is augmented $\sim$8 times each using MATLAB's *imageDataAugmenter* function (randomly selecting one of the 10 templates) giving a total of 12,463 synthetic images[2]. Those augmentations include a random combination of:

- Translation (Half of the image size in X- and Y- dimensions),
- Reflection (About the X- and Y- Axes), and
- Rotation (0–360 deg).

Additionally, the augmentations include several custom supplements to add more realism to the augmented wire features, such as:

- Duplication (2 augmented wires in an image),

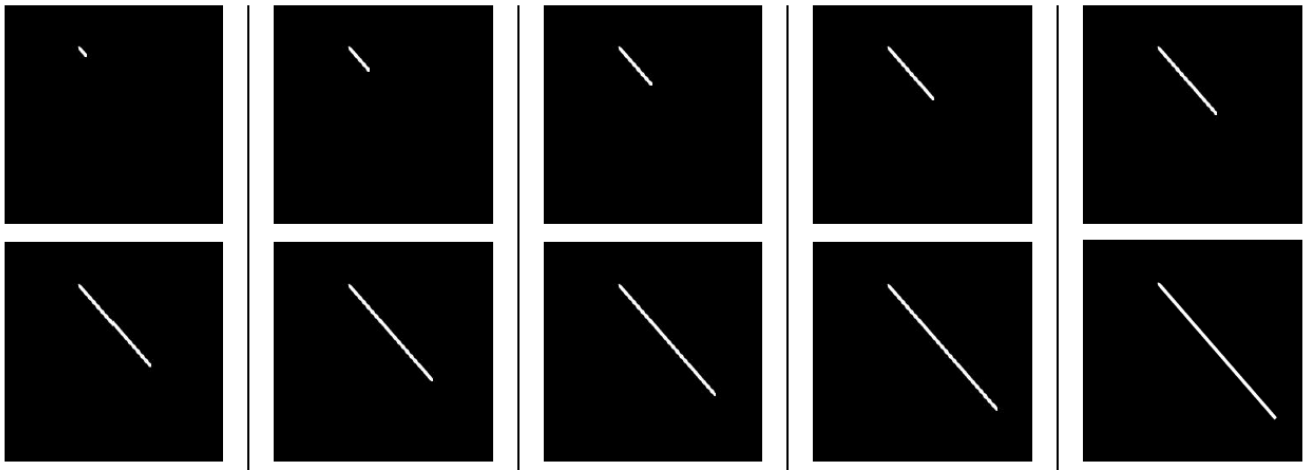

Figure 2. Wire templates to representing the varying length of a real wire, derived from the bottom right image.

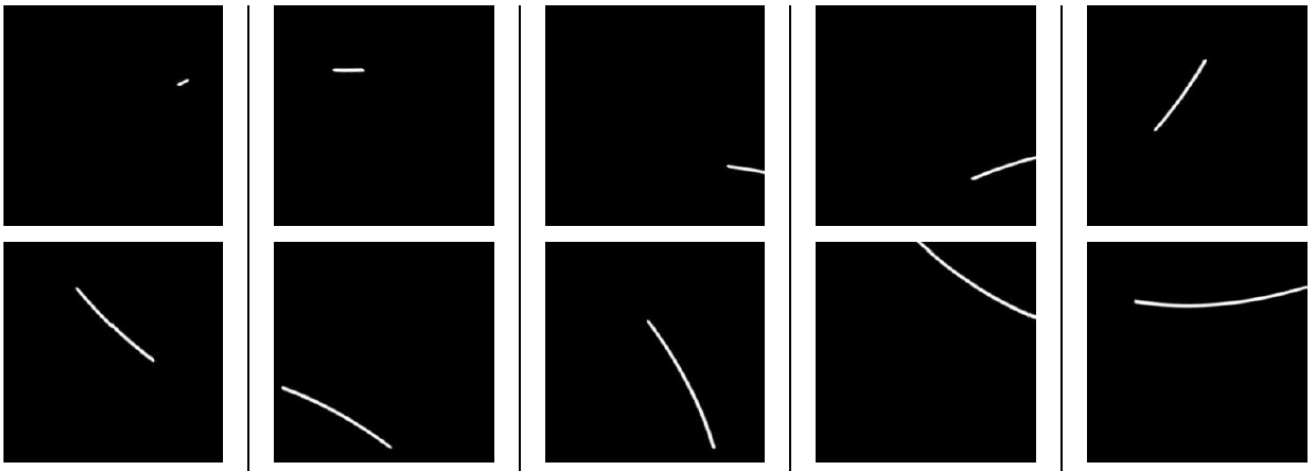

Figure 3. The application of data augmentations to the corresponding wire template images in Figure 2.

- Random pixel intensity between 128 – 255 (grayscale),
- 10% chance of being "negative" (*e.g*. no wire), and
- Bend.

where, Equation 4 defines the 'Bend' augmentation.

$$t(x) = x(:,1). * (\frac{\pi}{16}); \tag{1}$$

$$f_1(x) = x(:,1) \cdot \cos(t(x)) + x(:,2) \cdot \sin(t(x)); \tag{2}$$

$$f_2(x) = -x(:,1) \cdot sin(t(x)) + x(:,2) \cdot cos(t(x)); \tag{3}$$

$$F_(x) = f_1(x) + f_2(x); \tag{4}$$

The initial attempts at building this synthetic data set revealed several instances of bias that required redress. For instance, the initial assignment of 255 (to create a black overlaid wire) caused notable inaccuracy during early validation attempts of the model, as no real wire has complete opacity in real images, Therefore, subtracting a random gray intensity value between 128 and 255 allows for a softer edge with respect to the background and forces the model to learn the features of the wire without anticipating a specific pixel intensity or background.

Further, the exclusion of negatives (wire absence) led to a bias of assuming that a wire is always present. Conversely, inclusion of too many negatives causes the opposite bias — assuming a wire is typically not present. Thus, setting 10% of augmentations as negatives allows the model to reasonably predict the absence of a wire. Lastly, the *imageDataAugmenter* function allows for X- and Y- scaling and shearing, but these parameter inputs distort the wire template beyond a realistic representation of the wire. For example, scaling up the size of the wire makes the wire longer but also thicker, whereas shearing creates a staircase-like aesthetic. Other augmentation methods (using variety of different lengthened wire templates and bending the wire, respectively) negate the need for these augmentations.

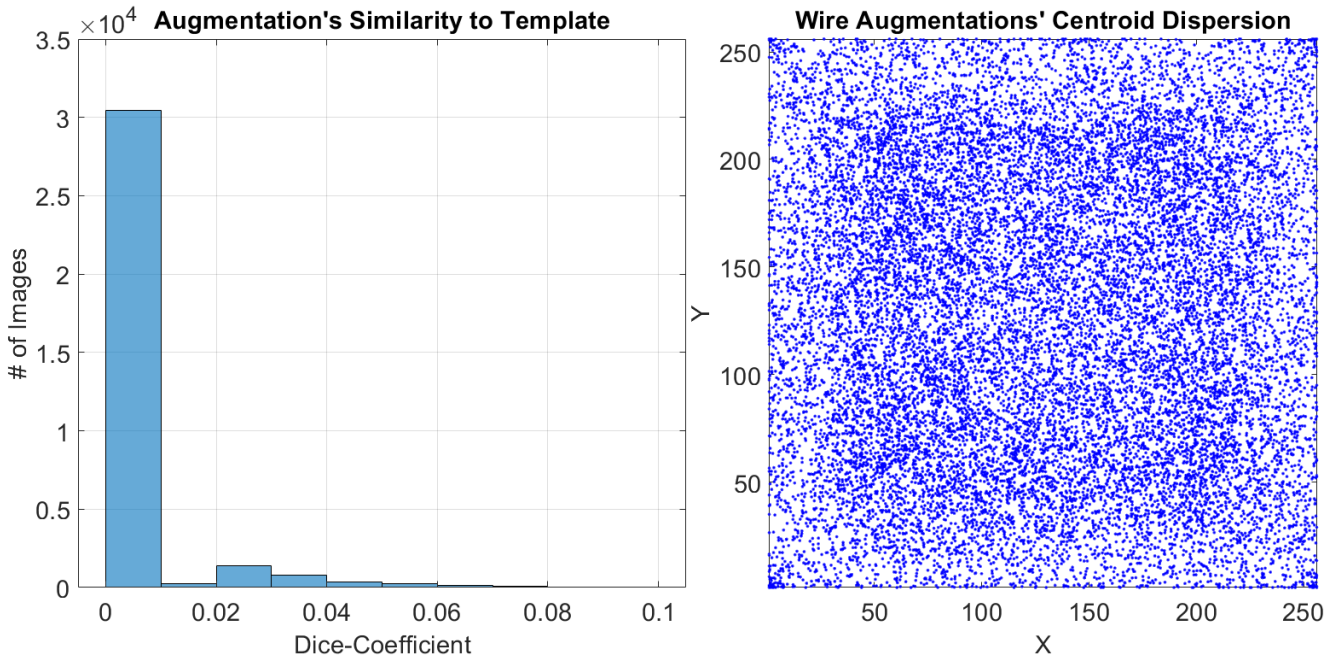

Figure 4. (Left) the similarity between every instance of an augmented wire template and its corresponding initial binary mask. See Equation 5 for computation of 'similarity' via *Dice Coefficient*; (Right) the spatial distribution of every augmented wire template's centroid, computed via MATLAB *imroi* bounding box functionality.

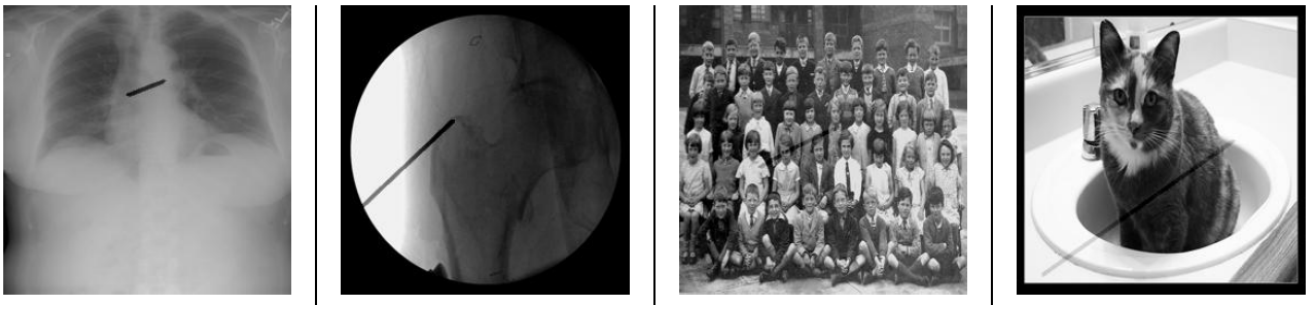

Figure 5. Actual synthetic images (y-train) used for training, where the intensity of the wire mimics the gradient of the background pixels.

Figure 3 shows that this set of augmentations produces a reasonable approximations of the real wire in various background images. The resulting synthetic image data set is a good step toward both realism and randomness. Figure 4 shows the degree of heterogeneity of these data augmentations in the synthetic image data set, where the resulting templates are nearly completely dissimilar to the template from which they were generated ($DiceCoefficient \approx 0$) and are not spatially biased to any one location. Lastly, Figure 5 shows an example of composite synthetic images following augmentation and overlay of the wire template. Note the relative intensity of the wire corresponding to its background, specifically in the 2nd, 3rd, and 4th images.

2.2. The Model

2.2.1 Overview

A prominent architecture of neural networks used for semantic segmentation is U-Net, which is a variation of a fully convolutional network (FCN)[9]. The FCN itself is an adaptation of the Convolutional Neural Network (CNN) and is designed for pixel-by-pixel segmentation — identifying the "what" in an image via feature map. The difference between U-Net and a FCN is also the reason for it's precision in image segmentation applications — a contraction path of layers (down-sampling) for identifying the key features of the image ("what"), and an expansion path of layers (up-sampling) for increasing the size of the feature map back to the exact size of the input image ("where" the "what" is in the image)[5].

This architecture (seen in Figure 6) enables U-Net to predict with high accuracy from just a few training images. Among its competitors, Rossenberger et al. shows U-Net outranking all of them in segmentation accuracy for multiple datasets in its inceptive publication[9]. This worm implements U-Net via Anaconda Navigator's Python (3.6) Jupyter Notebook, using Keras (2.2.4) and Tensorflow (1.13) deep learning library functionalities. The Python OpenCV library pre-processes the images before training.

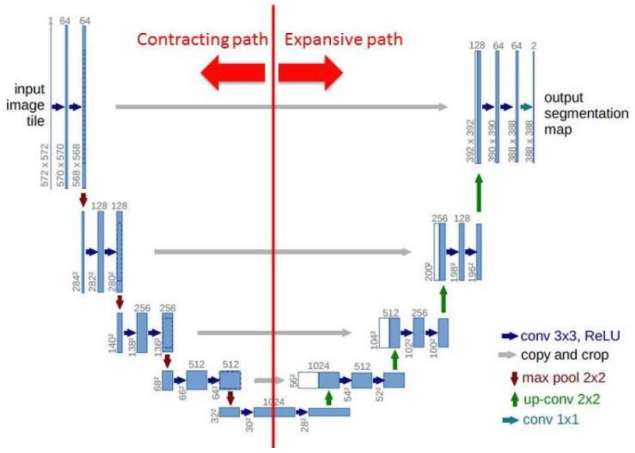

Figure 6. U-Net Model Architecture: A contracting path of layers followed by a expansive path of layers[9].

2.2.2 Hyperparamter Specification

This work utilizes U-Net primarily because its accuracy biomedical imaging applications. The only alterations architecture are the addition Dropout Layers before each Max Pooling Layer of the contraction path (to prevent overfitting), and one initial Gaussian Noise Layer (stddev = 0.10) following the input. Hyperparameters of the model include:

- 50 Epochs (twice)
- Batch Size = 32
- Validation Split = 0.20
- Adam Optimizer, learning rate = .0005
- Loss Function via the Dice Coefficient

$$DSC = \frac{2|X_{Train} \cap Y_{Train}|}{X_{Train} + Y_{Train}} \quad (5)$$

where, DSC is the similarity measurement of 2 images.

3. Results

Figure 7 shows the loss statistics that this model architecture and hyperparameter selection produces — a very promising (preliminary) result. It appears that the U-Net holds true to form in it's robust learning, however, given that all loss metrics only marginally improves after ∼20 epochs the learning potential from training data may be limited. After 50 epochs, the final Dice Coefficient, Jaccard Coefficient, and Loss Value are 0.8813, 0.6108, and -0.8804, respectively.

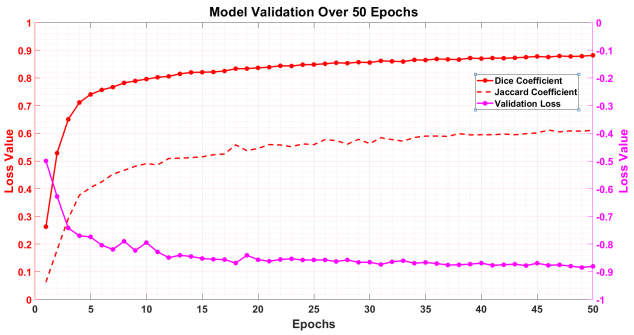

Figure 7. Validation and Loss Value per epoch.

3.1. Example Images

In total, 28 images are used to validate the model (14 real DHS images without ground truth, and 14 of the 12,463 synthetic images that were not trained on). Inputting these images into the saved model produced images like those shown in Figures 8 and 9 (for synthetic and real images, respectively). The results are in line with the final Dice Coefficient of 0.8813, as in, it appears that the model correctly predicts roughly 90% of the wire per image.

4. Conclusions

4.1. Discussion

The U-Net model performs well on the synthetic validation images that were excluded from the training data and on the real validation images, however, it was not feasible to capture the accuracy of the model on the validation images without their respective ground truths. Still, the model did not perform up to par with certain images, as illustrated by the liberal predictions of wire location seen in Figures 8 and 9. One source of the prediction error is likely a flaw in the training data is that sets all augmentations to the same $\frac{pi}{16}$ degree of bend, which could have been learned as a defining — but incorrectly assumed — feature of the wire.

Given these preliminary results, it is reasonable to expect improved accuracy with a larger, more diverse/robust synthetic data set. This could include different medical imaging modalities, an increased balance of random source images,

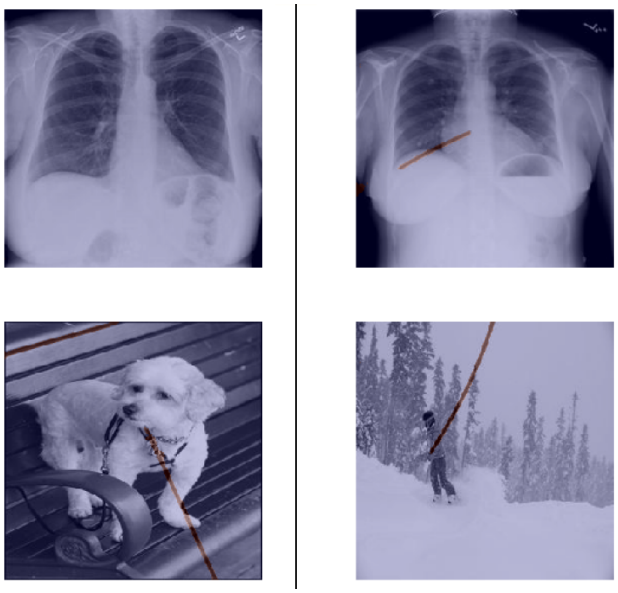

Figure 8. Model predictions on $\frac{4}{14}$ synthetic images. Note the errors in the bottom left and bottom right images, where the model is a bit overzealous with its prediction.

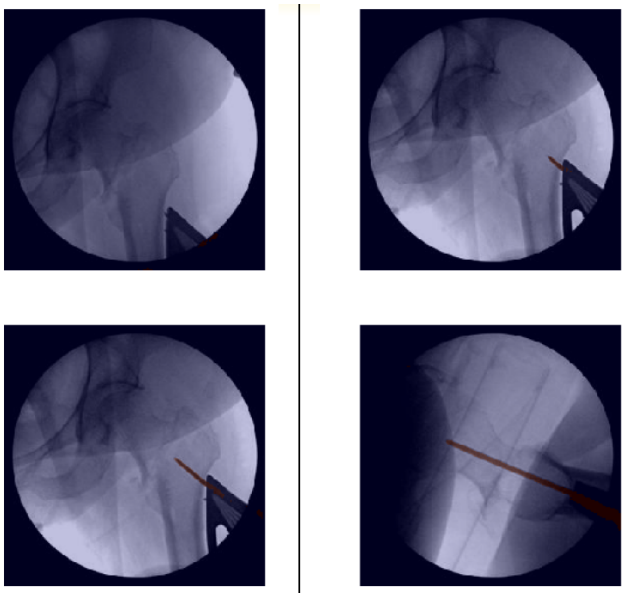

Figure 9. Model predictions on $\frac{4}{14}$ real images. Note the errors in the bottom left and bottom right images, where the model is a bit overzealous with its prediction.

more (or fewer) augmentations per image, etc. In particular, for the bottom left image in Figure 8, it is obvious to the human eye that the dog is sitting on a bench and a wire is overlaid on its torso-region. However, the model likely only had about 8 augmentations of this image to train on and was thus unable to differentiate between the wire and the wooden panels of the bench. This indicates a failure of learning that could be remedied with the inclusion of more augmentations of similar images. Additionally, the hyperparameters used for this model are cut-and-paste from other models, without any regard for the data set nor the context. Further improvement may be possible through refinement of these properties.

4.2. Implications

From the results, it is apparent that synthetic data can be useful in biomedical imaging applications when there is a lack of real data. The next step in this research is to train the model to also identify relevant anatomical features which can be used to identify the wire's relative location within the image, not just its absolute location. Automating the identification of features like the femoral head and neck labeled in Figure 1 would decrease the mental workload of a human manually processing these images. This would enable a greater ability to process and analyze these fluoros, allowing for a greater volume of processed data to draw conclusions from. These conclusions are likely to describe the decision making of surgeons as they navigate the wire through bone during the surgery, illustrated by their sequence of fluoros taken. Understanding this decision making could then pave the way for derived metrics which more adequately quantify performance than current methods like the OSATS.

4.3. Future Work

As mentioned previously, improving the methodology for generating the synthetic data can be improved. For example, simplicity purposes this model reduces the resolution of the training images significantly. Future training could use larger images (recall that the sizes of images this study uses are [256x256]) and could perhaps train on larger images ([512x512]) or even utilize the full RGB pixel information of each image rather than simplifying to gray-scale. Additionally, complimentary models should be trained for learning other information relevant to these surgical fluoros, such as image view (*i.e*. Anteroposterior v. Lateral). Lastly, the bend of each wire augmentation should vary randomly from some range $[pi, \frac{pi}{16}]$.

Acknowledgement

This work was made possible by the teachings and assistance of Dr. Stephen Baek, his assistant Joseph Choi, and the other reviewers of this course's poster presentations. Thank you for teaching this class.

References

[1] Coco - common objects in context; 2017 val images [5k/1gb]_2017, 2017.

[2] *MATLAB 2018a - Deep Learning Toolbox*. The Mathworks, Inc., 2019.

[3] Medpix chest x-ray dataset (7,470 images), 2019.

[4] I. Abdulkareem. A review of tip apex distance in dynamic hip screw fixation of osteoporotic hip fractures. *Nigerian Medical Journal*, 53(4):184, 2012.

[5] H. Lamba. Understanding semantic segmentation with unet. *Towards Data Science*, 2019.

[6] H. Niitsu, N. Hirabayashi, M. Yoshimitsu, T. Mimura, J. Taomoto, Y. Sugiyama, S. Murakami, S. Saeki, H. Mukaida, and W. Takiyama. Using the objective structured assessment of technical skills (osats) global rating scale to evaluate the skills of surgical trainees in the operating room. *Surgery Today*, 43(3):271–275, 2012.

[7] M. T. Nousiainen, S. A. McQueen, P. Ferguson, B. Alman, W. Kraemer, O. Safir, R. Reznick, and R. Sonnadara. Simulation for teaching orthopaedic residents in a competency-based curriculum: Do the benefits justify the increased costs? *Clinical Orthopaedics and Related Research*, 474(4):935–944, 2015.

[8] A. T. Pennock, M. Charles, M. Moor, T. P. Bastrom, and P. O. Newton. Potential causes of loss of reduction in supracondylar humerus fractures. *Journal of Pediatric Orthopaedics*, 34(7):691–697, 2014.

[9] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI 2015: International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 11 2015.

[10] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. 04 2018.