Deep Learning for Fluoroscopic Feature Detection

Liujiang Yan
Mentor: Robert Grupp, Professor Russell Taylor

Abstract: Efficient and robust feature detection is essential for performing 2D-3D perspective transformation between X-ray images and pre-operative models. In this project, we utilize deep learning methods, especially deep convolutional neural networks to perform landmarks position regression and contour detection task in 2D X-ray images. We utilize convolutional pose machines to regress belief map for each landmark. Also, we utilize holistical-nested network for contour detection. In order to address potential tool in the field, we introduce a data augmentation approach to simulate such situation. To validate the performance, we first use hard-tissue simulated data as start, and then expand to soft-tissue simulated data, and real X-ray data. Thorough and detailed experiments have been performed for analysis.

1. Introduction

The perspective transformation between intra-operation 2D X-ray image and pre-operative 3D-model play an essential role for surgery. The wide-spread feature driven 2D-3D registration methods need detecting the corresponding features in both 2D images and 3D models and then an optimization based algorithm is performed to find the optimal transformation. It is not trivial to design hand-crafted feature that is easy to detect and correspond in both X-Ray images and 3D model. Therefore, researchers use well-defined anatomical landmarks as features, and try to detect them in 2D X-Ray image. Also, instead of point feature, researchers also use contour as feature and develop contour based registration method correspondingly.

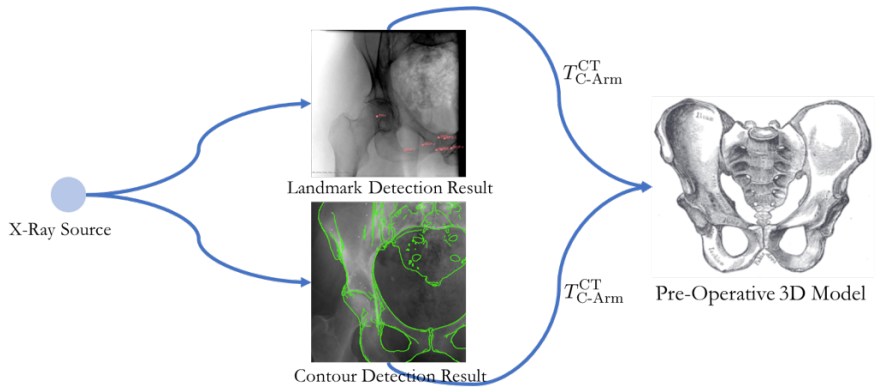

Figure 1 Feature Based 2D-3D Registration

To replace the time consuming handy detecting process, this project aims to utilize learning based method and annotated training data to train a deep convolutional neural networks model to perform feature detection in real time and end to end manner. The whole desired pipeline for automated registration pipeline includes data preparation, data labeling, deep learning network architecture design, training and evaluation process, and the final registration process.

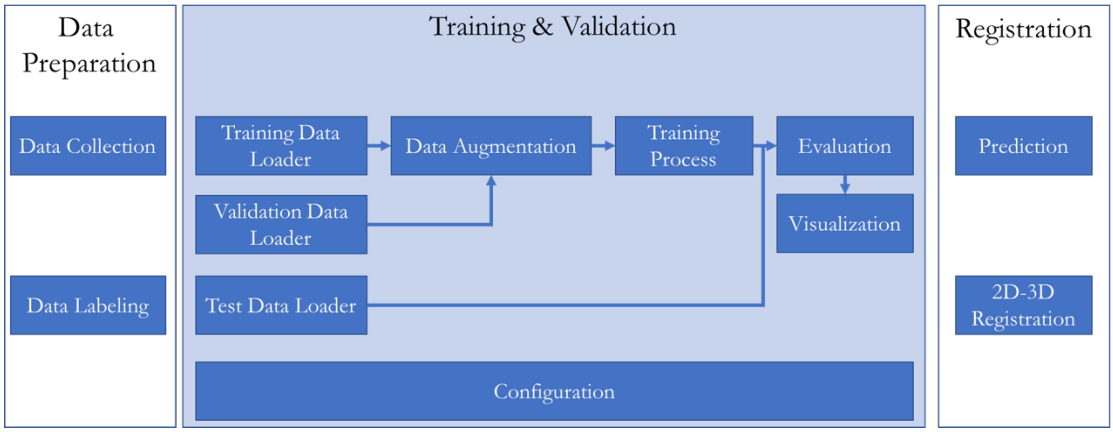

Figure 2 Workflow for Registration

This project focuses on the model training and evaluation pipeline implementation. Convolution neural networks and the variants achieve significantly outstanding performance in various computer vision tasks, from natural images to medical images. Therefore, here we start with convolution neural networks as feature extractor and discriminator to regress landmarks' and contours' image coordinates. A basic truth is that training a supervised deep learning model needs great amount of labeled data while medical images are usually rare for this usage. Therefore, we start with simulated data with annotated landmarks, and then expand to real X-ray images. We evaluate our results by pixel-wise distance for each landmark.

2. Methods

2.1. Data Preparation and Annotating

Training deep neural networks needs a large number of data with ground truth corresponding to the complexity of the model. However, in our project, it is not feasible to gather enough real X-Ray images with landmark and contour annotated for training the model from ground. Therefore, we propose to use simulated data as a start, adding complexity step by step and evaluating the model's performance. Once we have a trained and out-performed model on complex simulated data, we could use a small real X-Ray dataset to fine tune the pretrained model.

Figure 3 Data: Hard Tissue - Soft Tissue - Real X-Ray

The simulated data is provided by Robert Grupp, and here we briefly go through how we

generate the data and annotated the data. A pre-operative CAD model is given and different viewpoints and focal lengths have been set to generate various data. Since the 3D spatial positions of landmarks and contour are well-formed, and the projective transformation is determined, the annotation could be done in 2D simulated X-Ray images. Different simulation software and methods are used to generate different level of complexity of simulated X-Ray images, from bone-only, hard tissue to soft tissue.

2.2. Network Architecture

2.2.1. Convolutional Pose Machine for Landmark Detection

The landmark detection could be formulated as a fixed number key points regression problem. Given image I, the goal is to predict the image location Y for all P parts. In order to capture the landmark, it is not enough to look at the local image patch but needs to model the long-range dependencies among all the landmarks. Previous works introduce graphical model to explicitly model the spatial relation, while it is not trivial to learn and infer such graphical model.

In this project, we use the convolutional pose machines for our landmark detection task. Convolutional pose machines are originally introduced to perform human pose detection, which is similar to our task since they all require to regress a fixed number image coordinates with spatial relations.

Convolutional pose machines consist a sequence of convolutional networks that repeatedly produce 2D belief map for each part, where each pixel represents the probability of existence of the corresponding part. At each stage, the network first of all generate a feature map from input through a weight shared convolution based feature extractor block. Then the image features and the belief maps produced by the previous stage are used as input for a convolution based classifier block to generate the updated belief maps. The belief maps at each stage represent the non-parametric encoding of the spatial uncertainly of the location for each part, therefore through refining the belief maps stage by stage, CPM is able to learn rich spatial relations between parts. Another intuition is that, composition of multiple convolution operation will enlarge the receptive field of the network so that it could learn local to global feature stage by stage.

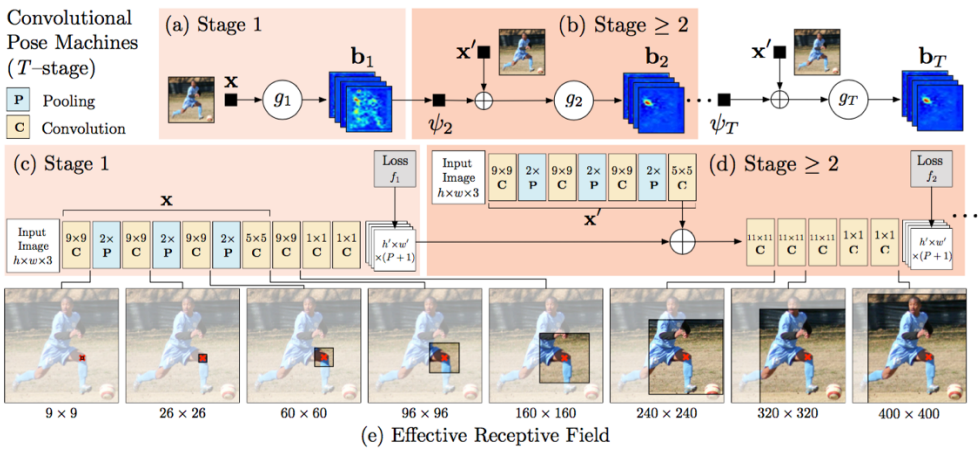

Figure 4 Convolution Pose Machine

Training deep neural networks can be prone to the problem of vanishing gradients where the magnitude of back propagated gradients decreases in strength with increasing intermediate layers between the output layer and the input layer. To alleviate the effect, in the training process a loss function for each stage is computed to minimize the L2 distance defined by the belief map from each stage and the target belief map for each part. And the overall loss is the sum of all stages, where t stands for stage, p for parts, x for all the pixels.

$$F = \sum_{t=1}^{T} \sum_{p}^{P+1} \sum_{x} ||b_t^p(x) - b_{gt}^p(x)||_2^2$$

To generate the target belief map, we define a 2D Gaussian distribution with mean as the landmark position, and certain variance (2 pixel in our implementation). The variance here is a hyper parameter and intuitively reflects how certain the annotation is.

There are several potential variants to this architecture that we could tune.

Basic Block: The basic blocks of Convolution pose machines consists of a feature extractor and a belief map regressor. The original implementation is a pure composition of convolution layer and pooling layer. There are other well-performed architectures for visual tasks that achieve outstanding performance that could be applied as basic block here. One is residual layer which utilize skip layer connection to alleviate gradient vanishing. The other is auto-encoder like U-Net to alleviate the downsample effect and capture features in different scales. In our implementation, we use the original design and decouple the block definition from the model architecture and leave room for switching to other blocks.

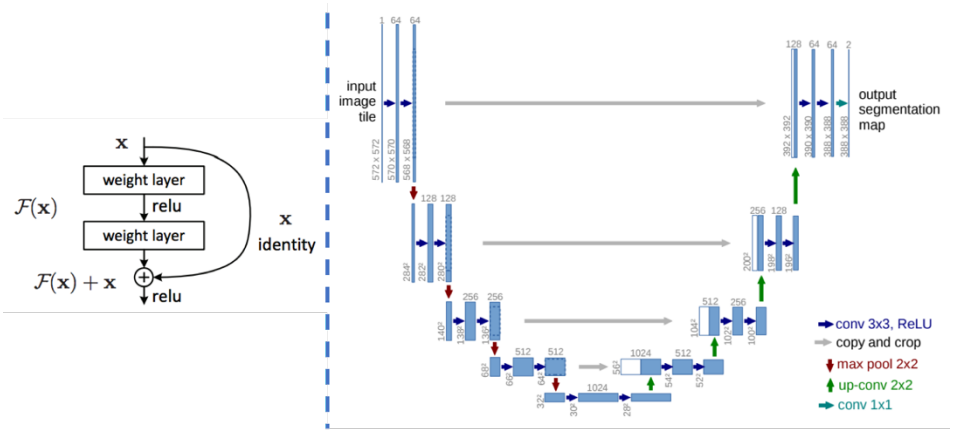

Figure 5 Residual Block; U-Net

Stages Number: The number of stages corresponds to the model complexity and spatial range that the model could capture. As for our project, in one aspect, landmark detection is simpler than human pose detection since the spatial relation is rigid among all the landmarks while the spatial relation is highly deformable for human pose. In other aspect, X-Ray images are complex than natural image due to the composition of different tissues. Through experiments on different settings, we use a six-stage CPM for our landmark detection task, while also keep the implementation configurable.

Loss Function: The original mean of squared error is one metric to measure the

difference between two belief maps. However, if we treat the belief map as a probability distribution then we may use KL divergence loss to measure the difference between two distributions. In our implementation, we use the MSE loss but leave the loss configurable.

2.2.2.Holistically-nested Network for Contour Detection

The contour detection problem could be formulated as, given image I, the goal is to regress a same size binary image T that each non-zero value pixel represent the existence of edges. Previous researches like gradient method or Canny detection perform well on natural images, while does not work for X-Ray images. There have been recent wave of development using deep convolutional neural networks treating edge detection as supervised regression problem. Holistically-Nested edge detection method is currently state of the art, and tackles two critical issues: it performs holistic image training and prediction, and it performs multi-scale feature learning by intermediate supervision.

The fundamental architecture is the convolutional stages of VGG nets original for image classification, which could be treated as a multi-scale feature learning in the form of increasingly larger receptive field and downsample layer. On each stage, a side output layer is introduced by a convolution layer and a upsample layer. Each side output is associated with a classifier and then lead to a stage-wise loss.

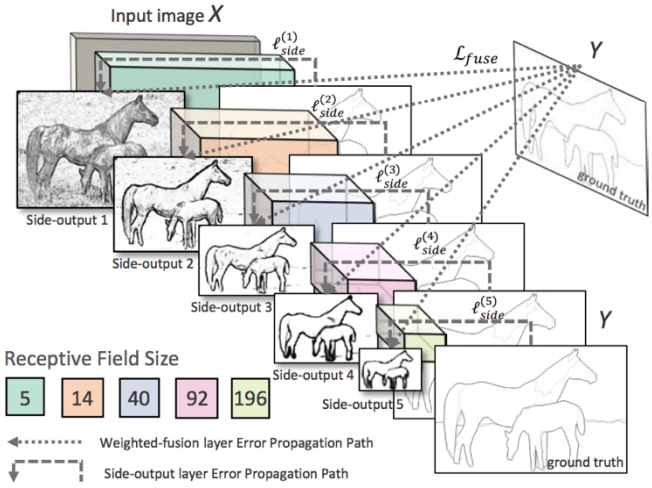

Figure 6 Holistically-Nested Edge Detection

Essentially, the edge detection problem could be treated as pixel-wise classification, and therefore binary cross entropy loss is suitable. Also, due to the highly unbalance between edge and non-edge pixels, a class-weighted cross entropy loss is introduced.

$$L = -\beta \sum_{j \in Y^+} \log(y_j = 1|X) - (1-\beta) \sum_{j \in Y^-} \log(y_j = 0|X)$$

Beyond side output edge map predictions given by each stage, a fusion layer is introduced to combine all the side outputs in different scale, through a learnable convolution layer. Then in testing phase the final prediction could be obtained by averaging all the outputs.

There are several potential variants to this architecture that we could tune.

Fundamental Networks: As we discussed above, other network architectures like deep residual net and U-Net perform well in numerous visual tasks, therefore could be a feasible alternative to the original VGG net. In our implementation, we stick with the original VGG net as fundamental network.

Learnable Upsampling: The upsampling layer in the original architecture uses bilinear interpolation. An arbitrary interpolation function could be learned through replacing the original upsampling layer with a learnable deconvolutional layer.

Stage Numbers: The number of stages represents the level of global contour that the network could capture. It is highly dependent on the task as well as the quality of annotation. Also, weights among different scales could be also introduced to embed the priority about the task. Through experiments, stage number as 3 is sufficient for our contour detection task.

2.3. Data Augmentation

The ultimate goal of this project is to perform robust and accurate feature detection during operation. Therefore, the field of view in the operation time X-Ray images may not be that clear as the simulated data as well as measured real data. One of the main difference is that the surgical tools may occur in the field and mask some regions as well as the features.

It is not feasible to simulate such effect through simulation as well as collecting such real data for training. Therefore, here we introduce a self-defined data augmentation method to simulate such mask effect. Generally, data augmentation introduces reasonable artificial variance to the original dataset in order to avoid overfitting.

Our method is to randomly mask a region of arbitrary shape, arbitrary size, and arbitrary constant intensity, as following shows. Noting that, we only mask the input image, but keep the target landmark and contour image as the same, assuming that the network is able to regress the information from the global cues.

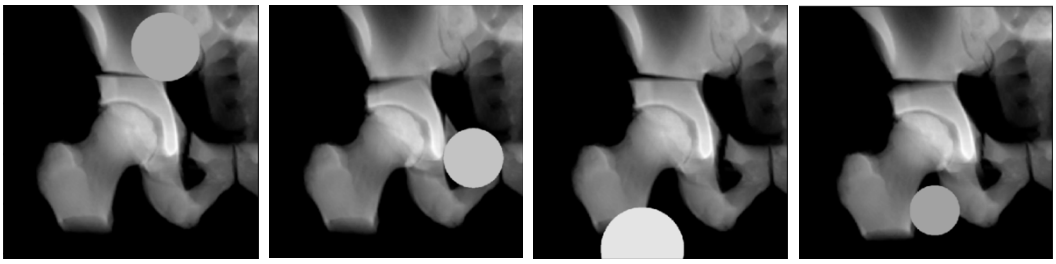

Figure 7 Data Augmentation: Random Region Mask

2.4. Implementation

The whole pipeline is implemented in Python, and based on one of the deep learning framework PyTorch. Our code structure consists of following several parts:

- configuration: dictionary for dataset configuration, network architecture configuration, and training configuration;
- models: implementation of deep neural networks, and unit tests;
- utility: implementation of support functions and visualization methods;

- main: script for training and inference with given configuration.

As shown, we decouple each component by the functionality for flexibility and maintaining.

3. Results

3.1. Experiment Setting

For each dataset, we split it to three separate parts: training data, validation data and test data, with ration 0.6 : 0.2 : 0.2. Since the images is ordered subject by subject, we also evaluate whether the model could be transferred to other subjects. At each epoch, we use training data to update the model parameters, and then evaluate the validation loss by validation data, and if the validation loss keeps going down, we are confident that the model is not overfitting. Once we have validated the model and well-tuned the hyper parameters, we deploy the trained model for the separate test data.

Table 1 Summary of Dataset Statistics

	Training Data	Evaluation Data	Test Data
Hard Tissue Simulated	3110	518	518
Soft Tissue Simulated	3110	518	518
Measured Real Data	24	4	4

3.2. Training Details Summary

The training details for models is different, and consist of several parts: output activation function, loss function, optimizer and the corresponding parameters, and number of epochs.

Table 2 Summary of Training Details

	CPM for Landmark Detection	HED for Contour Detection
Activation Function	identity	logsoftmax
Loss Function	MSE	Class-Weighted NLL Loss
Optimizer	SGD	SGD
Learning Rate	0.002	0.05
Momentum	0.9	0.9
Number of Epochs	30	30

3.3. Results and Summary

3.3.1. Landmark Detection

The metric we use for evaluating the performance of given model on landmark detection is the image coordinates distance between prediction and target for each landmark. A feasible and more meaningful alternative is to perform landmark based 2D-3D registration method and compute the difference between prediction-led transformation and ground truth transformation. In our experiments, we use image coordinate distance.

We first train our model on hard-tissue simulated data, and evaluate the performance in the following. From the following visualization we could see, the model is capable of

capturing the landmark in reasonable spatial accuracy, while there exist some cases that the model gives prediction with large error.

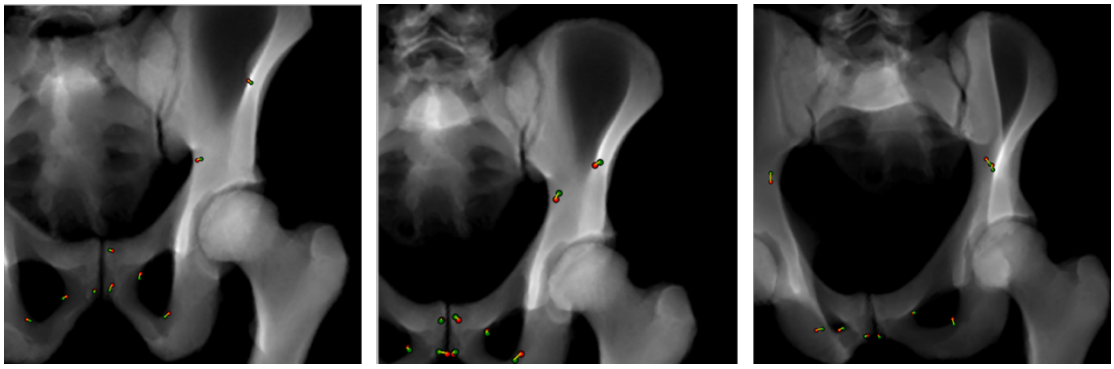

Figure 8 Good Prediction for Hard Tissue Simulated Data; Green: Target, Red: Prediction

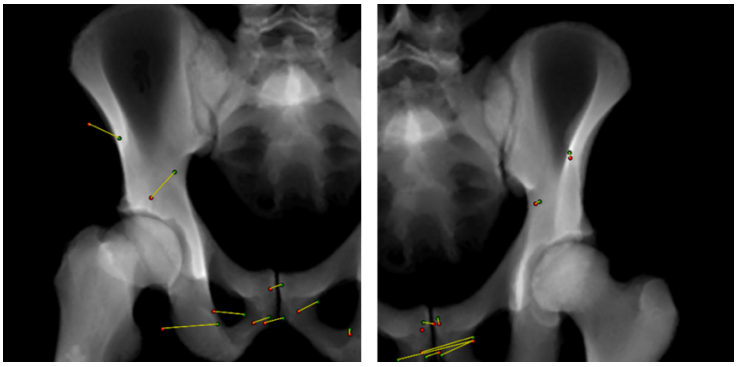

Figure 9 Poor Prediction for Hard Tissue Simulated Data; Green: Target, Red: Prediction

We evaluate the prediction error from target by image coordinate distance for each landmark, and carry out the following plot. We could see that, in most of the cases, the predictions are accurate for most cases, while there exist more than 100 pixels' error for some cases.

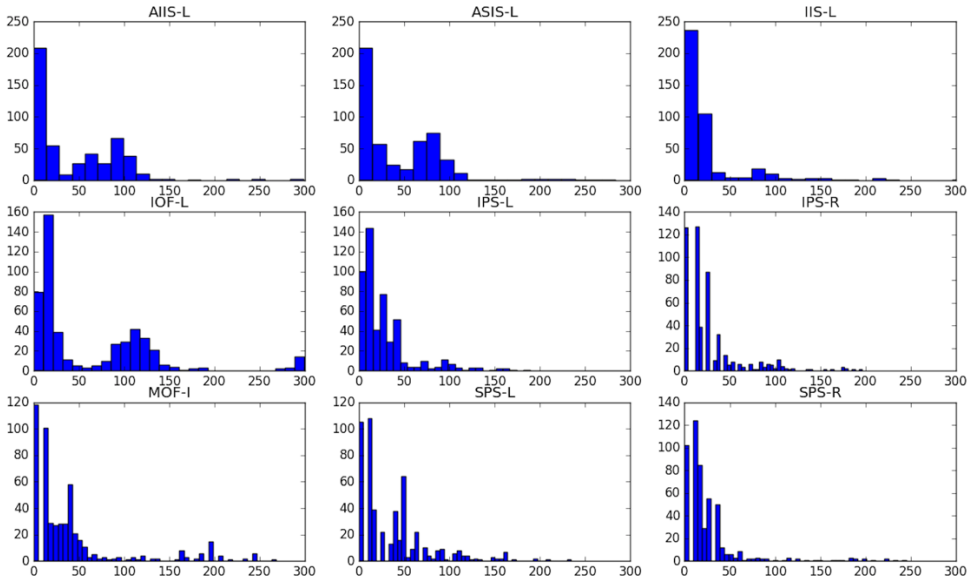

Figure 10 Image Coordinates' Error for Hard-Tissue Simulated Dataset

Then we look at the soft-tissue simulated dataset. This dataset differs from the previous data set with its increasing complexity in simulation and involving of soft tissue in the field of view. We keep our network as the same and train the model on training data and evaluate on test data. Here we also list some cases for intuition.

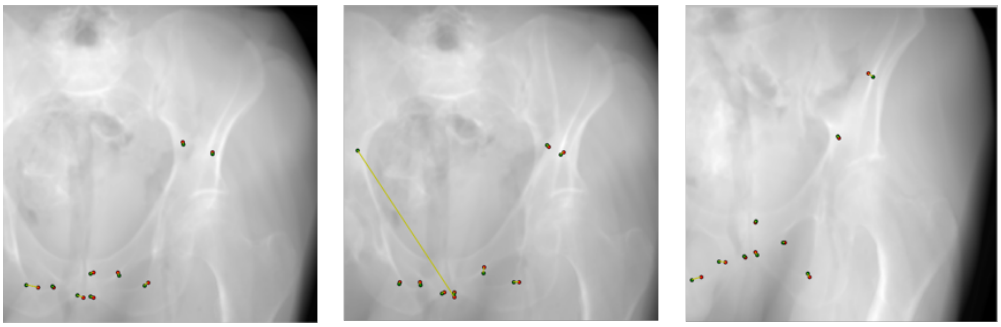

Figure 11 Good Prediction for Soft Tissue Simulated Data; Green: Target, Red: Prediction

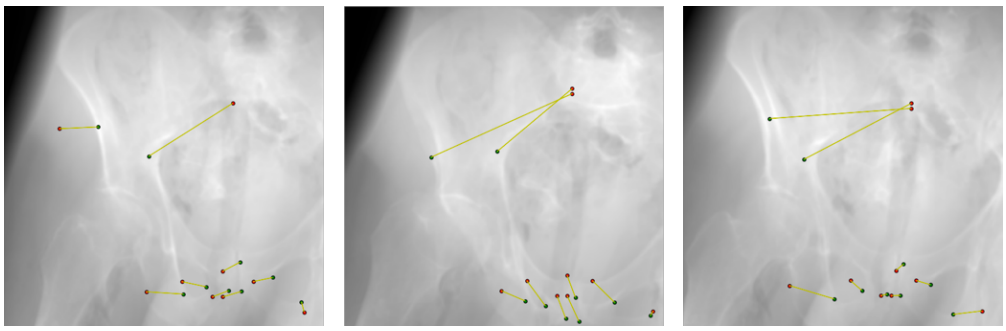

Figure 12 Poor Prediction for Soft Tissue Simulated Data; Green: Target, Red: Prediction

We evaluate the prediction error from target by image coordinate distance for each landmark, and carry out the following plot. Comparing to the hard-tissue simulated data, the performance under this circumstance is not as well.

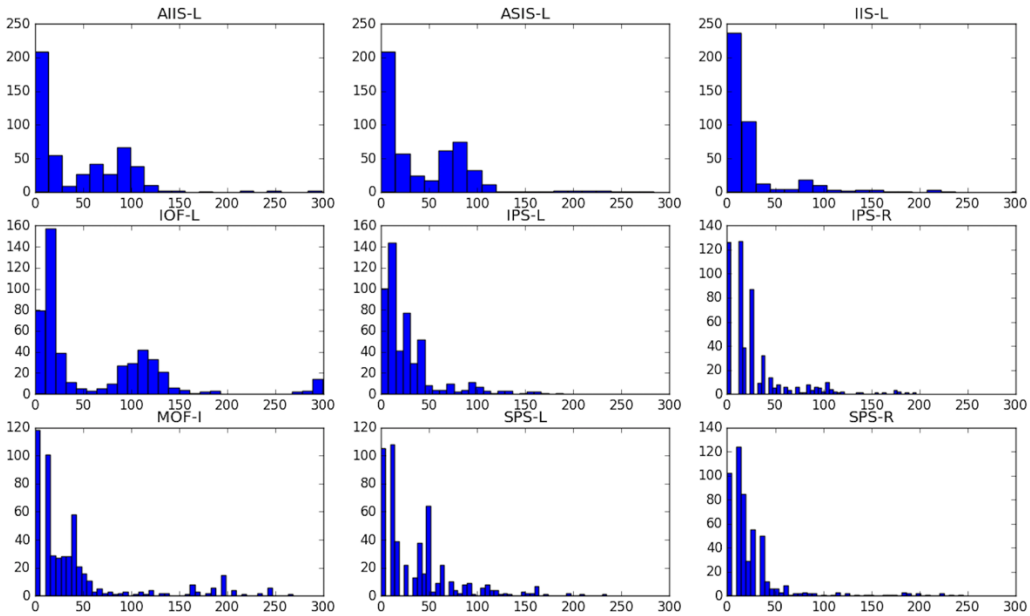

Figure 13 Image Coordinates' Error for Soft-Tissue Simulated Dataset

Finally, we deploy the model trained on soft-tissue simulated data to the real measured X-Ray data without fine-tuning. We could see that the model trained on simulated data is not capable of transferring to real X-Ray data, with relatively large offset.

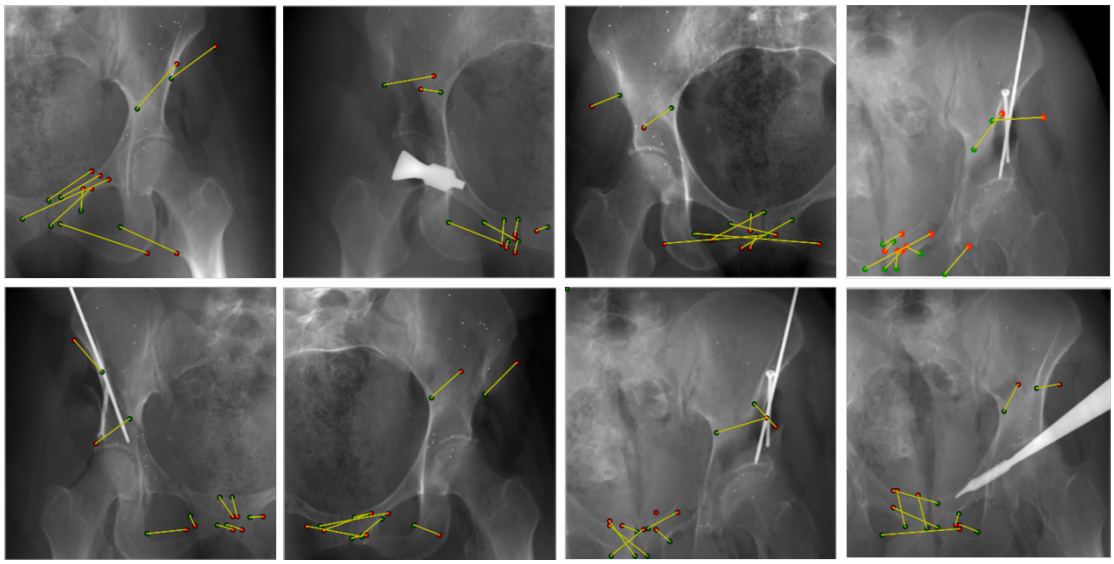

Figure 14 Prediction for Measured Real X-Ray Dataset

Then we evaluate the prediction error from target by image coordinate distance for each landmark, and carry out the following plot. Still, we could see that there exist cases that the model could give relatively accurate prediction. That raises an question that how we could detect such poor predictions, and throw them out from evolving into the following registration objective function.

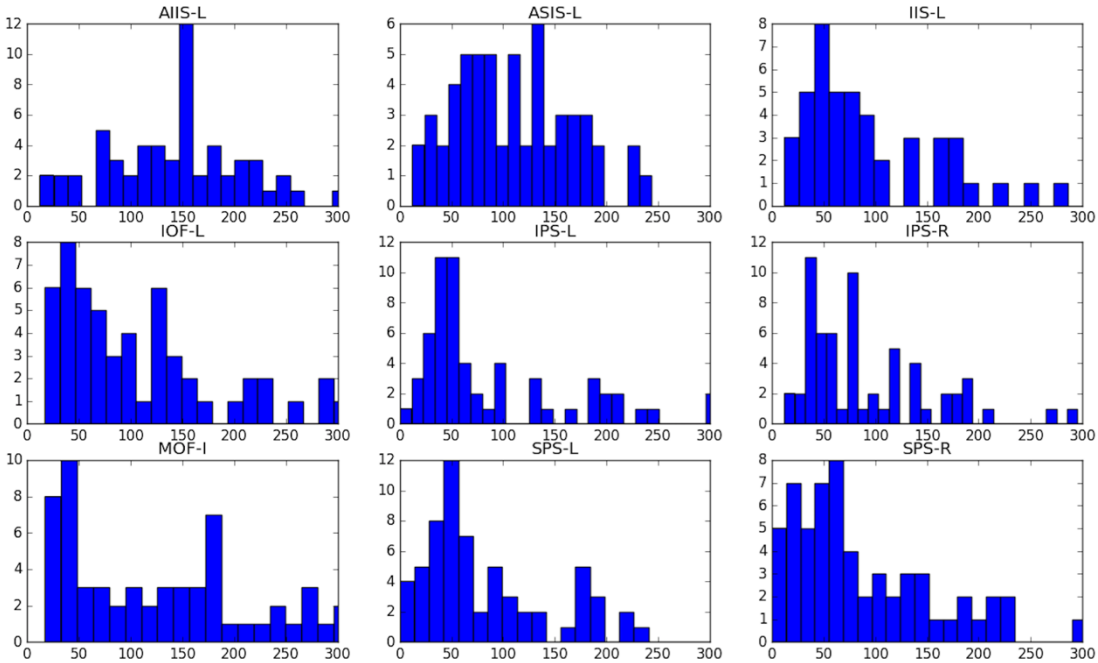

Figure 15 Image Coordinates' Error for Soft-Tissue Measured Dataset

3.3.2.Contour Detection

The metric we use for contour detection is accuracy, and F1 score, based on the classification result. Accuracy is given by the ratio of correctly classified pixels to all pixels. Since the ratio of edge pixels to non-edge pixels is highly unbalances, we also use other metrics as precision, recall and F1 score, given definition as following.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

As above, we first of all train our model in hard-tissue dataset, and evaluate on a separate test dataset. Here we list some results from test phase, with both intermediate side outputs and final prediction.

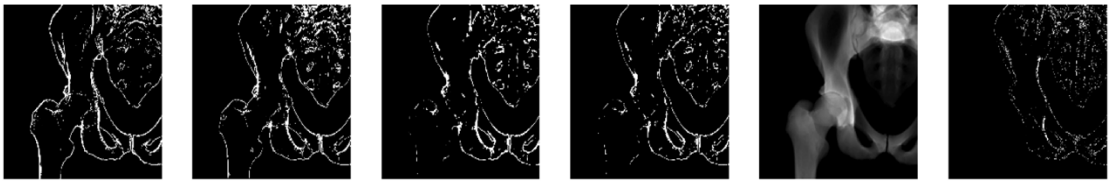

Figure 16 Stage 1-3, Fusion Prediction, Input, Target

Then we perform the metric to the test phase, and carry out the statistics summary about the performance of HED on hard tissue simulated dataset.

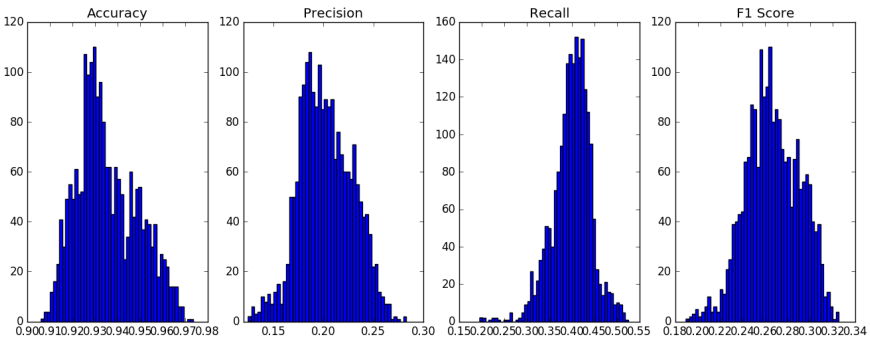

Figure 17 Accuracy, Precision, Recall, F1 Score on Hard Tissue Simulated Dataset

The ratio of edge pixels to non-edge pixels is around 0.1 and therefore around 90% accuracy is not decent. However, if we look at the prediction and target, we shall see that, the given target edge map is sparse and not continuous along the contour, while the edges of given prediction are thick and continuous. Therefore, we overlay the prediction as well as the target edge map on the input image, and analyze qualitatively. From the following overlay images, we could see the prediction align the contour accurately, though the metrics defined above is not outstanding.

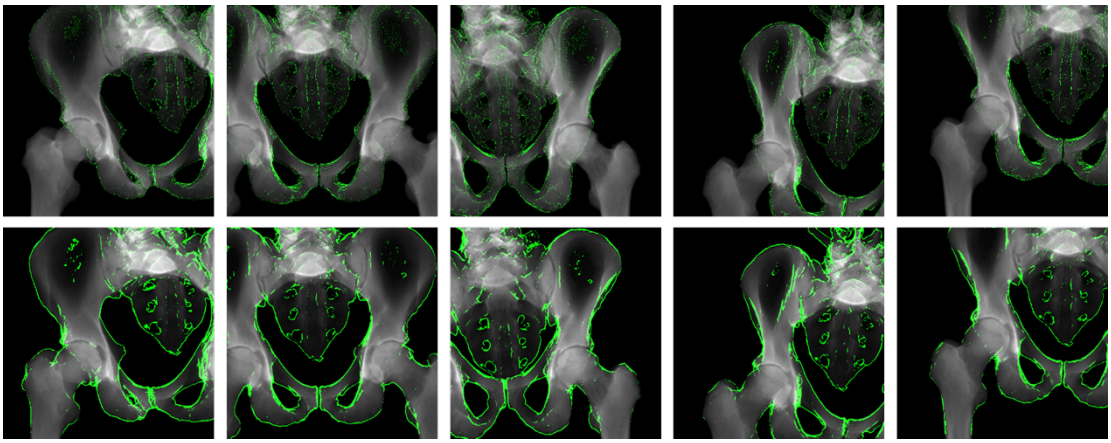

Figure 18 Edge Overlay Images. Top: Target, Bottom: Prediction

Through the evaluation on hard-tissue simulated dataset, we illustrated that HED is capable to capture the contour in X-Ray images in a supervised learning manner. Furthermore, we perform the same experiment on soft tissue dataset.

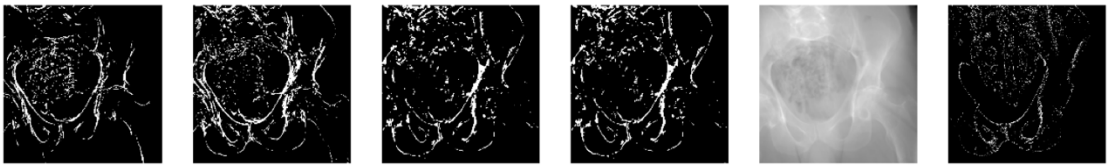

Figure 19 Stage 1-3, Fusion Prediction, Input, Target

Then we perform the metric to the test phase, and carry out the statistics summary about the performance of HED on soft tissue simulated dataset. Again, the metric gives poor result regarding the predictions and the targets.

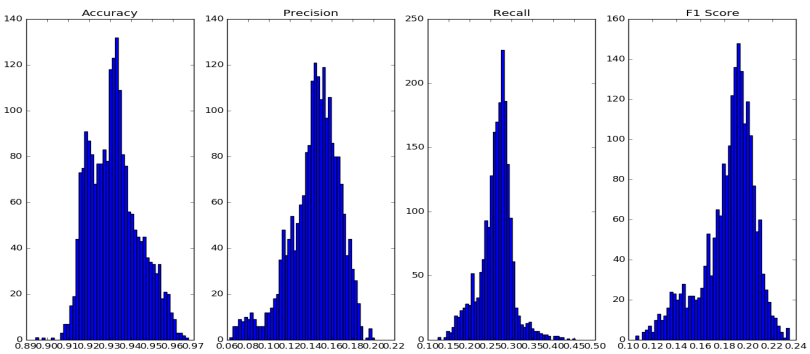

Figure 20 Accuracy, Precision, Recall, F1 Score on Hard Tissue Simulated Dataset

Through the overlay images, we could see that the predictions give thick contours which well-align the pelvis, which illustrates the effectiveness of HED on capturing the contour in soft-tissue simulated data.

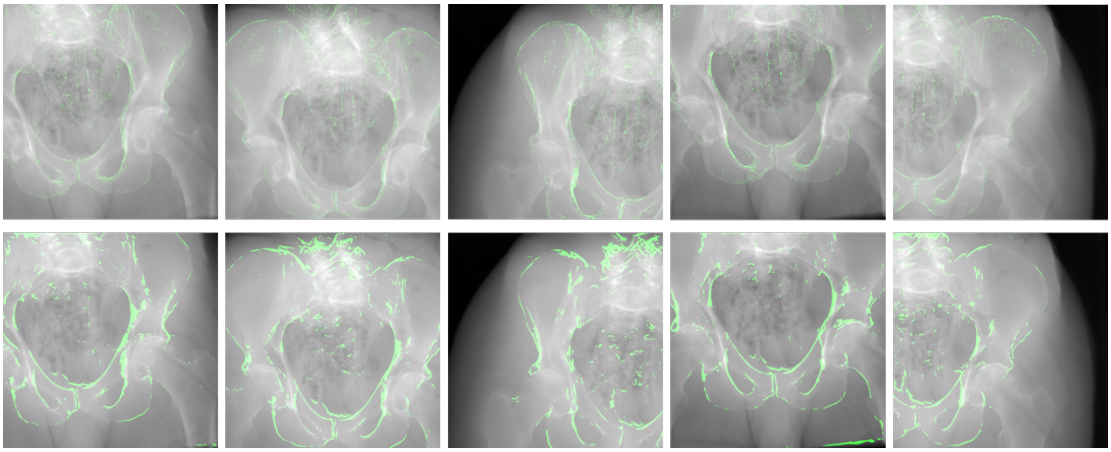

Figure 21 Edge Overlay Images Top: Target, Bottom: Prediction

Then we perform the metric to the test phase, and carry out the statistics summary about the performance of HED on soft tissue simulated dataset.

It is interesting to see whether model trained on soft-tissue simulated data is capable to transfer and infer accurate in real data, given the similarity of the image domain. Therefore, here we directly deploy the pretrained model by soft-tissue simulated data to real data, and list several results in the following.

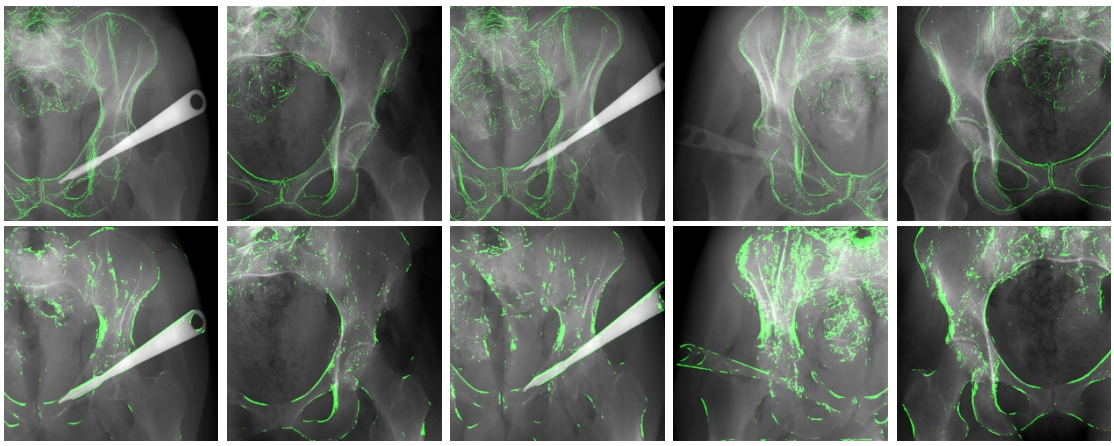

Figure 22 Edge Overlay Images Top: Target, Bottom: Prediction

The overlay images show that the model trained on soft-tissue simulated data could smoothly transfer to real X-Ray data without fine-tuning on an extra separate real X-Ray dataset. However, we shall see that since the model does not encounter tools throughout the training process, it detects the contour of tools as well. We may apply random mask augmentation to alleviate such effect.

4. Discussions and Future Works

4.1. Landmark Detection

Downsample The output size of convolutional pose machines has downsampled resolution due to the pooling operations. Therefore, one pixel error in the output belief map will lead to tens of pixel error in the original input image, therefore will lead to poor localization. The straightforward approach is to replace large kernel convolutional layer

with a composition of small kernel convolutional layer to keep the same receptive field. Another more decent approach is to introduce auto-encoder like architecture, therefore could keep the output belief map size as the input but still capture mutli-scale feature intra-block.

Robustness From the visualization results, we could see that the network is not robust towards the noisy input. One approach is to preprocess the images, such as normalization and denosing. Another general alternative is to add dropout layer and batch normalization layer to stabilize the network during the training process, where dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data, and batch normalization helps stabilize the network by keep the inputs for next layer with zero mean /unit variance for a given batch in a training iteration.

Belief Map In the implementation and in this project, we only use the regressed belief map to get the point estimate of the landmark image coordinates by the maximum response. The belief map actually reflects the uncertainty measure of the network towards the whole image range. Therefore, we could utilize the uncertainty measure qualitatively and quantitatively to enhance the following registration. Qualitatively, through belief map, we could derive the divergence to threshold the poor predictions and not involve those into registration objective, treating as outliers. Quantitatively, since our target belief is assumed Gaussian, we could model our regressed belief maps as Gaussian distribution as well. Then, by rewriting the reconstruction objective function as,

$$\underset{\theta \in SE(3)}{\arg\min} \sum_i \frac{1}{2}[P^i_{2D} - \mathcal{P}(P^i_{3D};\theta)]^T \Sigma^{-1} [P^i_{2D} - \mathcal{P}(P^i_{3D};\theta)]$$

We could embed the whole belief map into the registration and hopefully could lead to more robust performance.

4.2. Contour Detection

Metrics Through the analysis, we could see that treating edge detection as pixel-wise classification problem and applying corresponding metrics as precision, recall, and F1 score is not suitable and highly prone to the annotation. Instead of one or zero loss, we could alternatively compute the Euclidean distance map between the predicted edge map and target edge map. For each edge pixel in predicted map, find the nearest edge pixel in target edge map, and average the distance for the whole image. Through such metric, we could measure the difference between two edge maps.

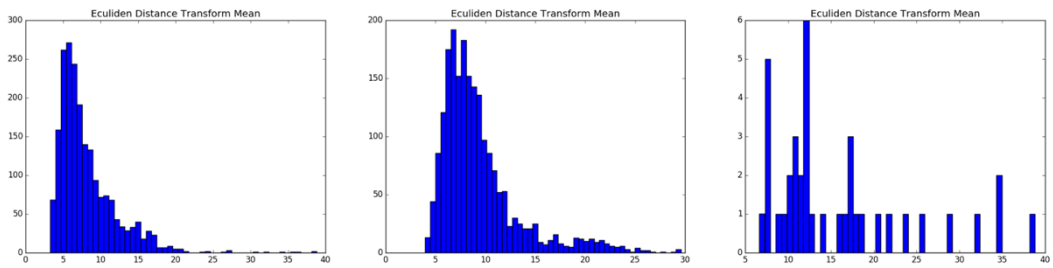

Figure 23 Distance Transform on Edge Map

5. Deliverables

The goal of this project is to develop deep learning based pipeline for landmark detection and contour detection, starting by simulated data and transferring to real data.

Task	Status	02/22	03/01	03/15	03/29	04/05	04/12	04/19	04/26	05/03	05/10
Project Proposal	Done									**Minimum**	
Environment Set Up	Done									**Medium**	
Initialize Neural Network	Done									**Maximum**	
Seminar Report	Done										
Training on Larger data	Done										
Evaluating Accuracy	Done										
Final Report / Poster	Done										
Tools in field of view	Done										
Contour Detection	Done										

Figure 24 Schedule and Deliverables

We have adjusted the expected deliverables throughout the semester. Specifically, the data simulating part is taken charge by Robert Grupp, and Liujiang is focused on the learning part. Therefore, we eliminate the simulation task from the list. Also, due to the time constraint and absence of accurate registration annotated data, we eliminate the end-to-end transformation regression task from the list and focus on feature detection.

We have achieved the minimum deliverables on landmark detection, by using Convolutional pose machines, and we performed experiments to analyze the performance and limitations of such method. We have achieved the medium deliverables on addressing tools in field of view, by introducing random region mask as data augmentation, and states that belief map as uncertainty measure could be used to threshold poor prediction and embed extra information for robust optimization based registration method. We have achieved the maximum deliverables on contour detection, by utilizing HED network, and we successfully transferred the model trained by soft tissue simulated data to real measured X-ray data.

References

Wei, Shih-En, et al. "Convolutional pose machines." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

Xie, Saining, and Zhuowen Tu. "Holistically-nested edge detection." *Proceedings of the IEEE international conference on computer vision*. 2015.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).