Deep Vein Thrombosis Detection from CT Scans

Anirudh Joshi
Stanford University
anirudhjoshi@stanford.edu

Ishan Shah
Stanford University
ijshah@stanford.edu

Rui Liu
Stanford University
rui3@stanford.edu

Abstract

Patients who get CT scans for any concerns are generally discharged before radiologists evaluate the scans. For patients with deep vein thrombosis (blood clots), it is critical for them to get on medication as soon as clots are discovered from the CT scan [1]. An automated system to detect clots in veins would help early diagnosis at the point of testing and would prevent fatalities from delayed radiologist involvement. We explored solutions to this problem ranging from classical computer vision like SIFT to deep neural networks like DenseNet-161 and finally multitask segmentation and classification neural networks. Our results show that the multitask model which takes into account more local information is best suited for tackling this problem. Our results are currently the best reported numbers on the Stanford Medicine deep vein thrombosis dataset.

1 Introduction

Deep vein thrombosis (DVT) is a blood clot commonly found in deep veins of the lower extremities. Every year, 60,000-100,000 Americans die of complications arising from DVT [1]. Despite its high prevalence and fatal impacts, DVT continues to be under-diagnosed in the clinical setting.

To tackle this problem, Computed Tomography (CT) has shown superior diagnostic power. For our study, we aim to apply computer vision techniques and mainly implement deep neural networks (DNN) to process 3D datasets of CT images to detect DVT. More specifically, the successful detection of a DVT will rely on (1) localization of targeted deep veins in a CT slice, and (2) classification of whether a DVT is present in multiple slices.

2 Related work

Deep learning has shown effectiveness across various domains in the medical field for detecting pathologies from images. Irvin et al. and Rajpurkar et al. showed that DenseNet-161 was effective on large chest-xray datasets to achieve radiologist level performance for 14 different pathologies [2], [3]. Gulshan et al. showed that deep neural networks could identify diabetic retinopathy from photographs of the retina [4]. The strengths of these papers were training very deep classification models on 2-D medical images. However these papers differ from our setup given that the images we have are naturally 3D and are sliced to generate 2D images. Further these models have not been able to generalize to noise in the data which is an inherent part of our dataset [5]. UNet is a network built for the task of medical image segmentation which takes advantage of highly localized information about an image. This work can aid us in classification by incorporating that local knowledge in the hidden representations [6]. To the best of our knowledge deep learning for DVT detection from CT scans has not been studied in the literature. The current state of the art model performance on the VITAL dataset curated by Dr. Hofmann at Stanford is an AUC of 0.6 based on unpublished work in his lab while radiologists have an estimated AUC of 0.9.

,

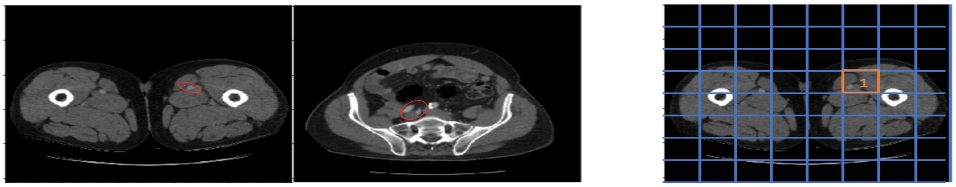


Figure 1: Presence of DVT in Femoral (L), Iliac veins (M) and 8x8 patch-wise splitting of scan (R)

3 Dataset and Features

The data used in this project originates from the Stanford Radiology Venous InTerventionAL (VITAL) database. VITAL contains 25 years of imaging data from all DVT patients at Stanford. Experienced radiologists then spent a considerable amount of time locating the blood clots on these images, resulting in a high percentage having complementary data that marks veins through labeled pixel values. The images are grayscale of size 512x512.

Initially, there were 89 unique patients in the dataset, each having anywhere from one to four separate studies done on them, with up to dozens of CT scans each. When initially viewing the scans, we noticed a clear dichotomy in the anatomy shown Figure 1. This was a result of some scans being taken above the groin crease that target iliac veins, and some below that target mainly femoral veins. To prevent a model from learning two entirely different anatomies and trying to generalize, we decided to focus our analysis solely on femoral veins, which made up about half the scans. In total, there were 460 CT scans done on femoral veins from 76 unique patients, and 29447 2D slices from them. Scan position indicated in the labels allowed us to isolate the femoral images and produce binary labels indicating presence of DVT.

We decided to choose our test and validation data from scans that have near the median amount of slices, to assure that we are testing on scans that most commonly represent what is seen in practice. We isolated 156 scans from the 33rd to 66th percentile of number of slices, and then randomly assigned half to the validation set and half to the test set. This led to the train-validation-test split of 2D slices being approximately 84%-8%-8%.

These scans' complementary segmentation files contained a small group of pixels labelled as 1 over a blank image, indicating the location of the vein, no matter the presence of a DVT. These locations can be very subtle and difficult to detect, as seen in Figure 1. A patchwise array of vein locations was created from the images' segmentation masks and was used in our error analysis, also demonstrated in Figure 1. In the end, our DataLoader inputs a csv file with the above information, and outputs the corresponding image, segmentation file, patch array, and "clean label" (DVT or no DVT).

4 Methods

4.1 Baseline Method - Bag-of-Word (BoW) Model with SIFT Image Features

For our baseline, we used a non-deep-learning method which builds a bag-of-word (BoW) model with SIFT image feature. SIFT image feature has become popular in image recognition classification tasks attributed to its scale and rotation invariance. SIFT creates representation of an image based on its gradient. We first extracted and encoded a set of SIFT features for each image. Thereafter, a BoW model was built to aggregate the encoded SIFT features into a vector representation for each image. Finally, we implemented a SVM classifer on the vector representation to classify the images into DVT versus non-DVT classes.

4.2 Entire Image Classification

Annotations Each image in the training set were annotated with binary labels as to whether or not deep vein thrombosis was present in the image.

Model We used various convolutional neural network architectures for the classification task. Convolutional neural networks convolve a filter across inputs to generate representations of visual features. Since they work on local regions at a time, they are translation invariant and computationally efficient.

The architectures we explored in for the classification task are VGG-16 and DenseNet-161. VGG-16 being a shallower network was used as a deep learning baseline. DenseNet is deeper and uses skip connections between all layers in a block to make learning tractable for deep networks. Since DenseNet is pretrained on ImageNet with a 1000 classes, we replace the final classification layer. The loss function used was Binary Cross entropy.

$$L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y * \log \hat{y} + (1 - y) * \log(1 - \hat{y})) \tag{1}$$

4.3 Multitask Learning

Annotations Given that veins are a very small part of the larger image, we hypothesized that giving the model more local context would allow it to learn the classification task better. The segementation masks for each image were used along with classification labels and the model is trained to jointly optimize both segmentation and classification.

Model We built a custom architecture that is combination of the ResNet and UNet. The encoder in the model is ResNet18. We create two branches after the encoder; one for classification and the other for segmentation. The segmentation branch follows the UNet scheme of transpose convolutions upto the original image size while the classification branch is another ResNet 18 network.

The loss function has 3 terms and is a weighted sum of the losses from classification and segmentation. The first term being the classification loss which is the cross entropy across the two classification labels. The second term is the weighted pixelwise binary cross entropy loss applied against the output of the segmentation head and the segmentation mask. The third term is the DICE loss.

$$PixelwiseBCE(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(w_0 * y * \log \hat{y} + w_1 * (1 - y) * \log(1 - \hat{y})) \tag{2}$$

$$Dice(p, \hat{p}) = 1 - \frac{2 * p * \hat{p} + 1}{p + \hat{p} + 1} \tag{3}$$

$$L(y, \hat{y}) = 0.5 * (ClassificationBCE) + 0.5 * (0.25 * PixelwiseBCE + 0.75 * Dice) \tag{4}$$

where n is the length of the flattened pixel array, w_0 and w_1 are class weights, p $\in \{0, 1\}, \hat{p} \in \{0, 1\}$

4.4 Sliding Window Vein Identification

As part of error analysis we train a neural network to identify presence of a vein (regardless of clots) in patches of the input image.

Annotations Since the size of the veins are very small relative to the images, we proposed that breaking down the image into 16x16 blocks of 32x32 pixels and then classifying each block for the presence of a vein would allow the model to accurately target image partitions that contained DVTs.

Model We used ResNet50 and a fully convolutional implementation of the patchwise sliding window classification. The loss function was weighted binary cross-entropy similar to equation (2), since the vast majority of patches would be labeled as 0.

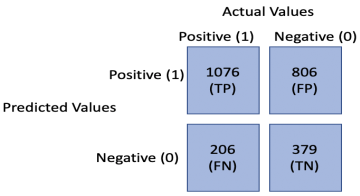
5 Results and Discussion

5.1 Baseline Experiment

The non-deep-learning method applied KMeans clustering and Support Vector Machine (SVM) classifier. KMeans clustering was used to categorize the encoded SIFT features into different clusters. The hyperparameter tuned during training was the number of clusters. Thereafter, we applied a SVM classifier to classify an image based on its clusters of encoded SIFT features. As we changed the hyperparameter (i.e., the number of clusters) in our model, we obtained the best performance at a 100 clusters with an AUC of 0.55, a precision of 0.56 and a recall of 0.72.

Experiment	AUC	Precision	Recall
SIFT Baseline	0.55	0.56	0.72
VGG-16	0.50	0.54	0.40
DenseNet-161 (train)	0.78	0.72	0.94
DenseNet-161 (test)	0.51	0.6	0.6
DenseNet-161 (test) with dropout	0.66	0.63	0.88
Multitask ResNetUNet	**0.69**	**0.68**	**0.73**

Table 1: Classification performance across models



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	1076 (TP)	806 (FP)
	Negative (0)	206 (FN)	379 (TN)

Figure 2: Confusion Matrix for DenseNet Classification (drop=0.2)

5.2 Full Image Classification Experiments

We finetuned convolutional networks pretrained on ImageNet like VGG-16 and DenseNet-161 on a binary classification task for presence of a thrombosis. After experimenting with different learning rates and optimizers, a learning rate of 0.001 was used with the Adam optimizer. Adam utilizes exponential moving averages to allow past gradient updates to influence future updates, preventing noisy updates and leading to faster convergence. RMSProp and SGD led to oscillating losses even with small learning rates. Initially experiments were run with a fixed learning rate but later a learning rate scheduler which divided the learning rate by 10 after every epoch was used. The reason for using a decaying learning rate is that smaller learning rates can escape local optima and saddle points which could occur mid training. A batch size of 10 was used for training due to GPU memory constraints preventing large batch sizes with DenseNet-161.

A shallower network like VGG-16 was used initially for the classification task. It was clear that training loss was saturating too early which indicating underfitting so model capacity needed to be increased. DenseNet-161 was chosen for increased model capacity and the architecture's effectiveness for medical images [2]. Table 1 shows that DenseNet's performance on the test set (AUC 0.51) was not much better than VGG (AUC 0.5) however it was clear the model was overfitting given the gap between training performance (AUC 0.78). A dropout of 0.2 was introduced and the results show that regularization helped AUCs on the test set (AUC 0.66). These results beat our baseline (AUC 0.54) and also beat all prior unpublished work in Dr. Hofmann's lab with this dataset.

5.3 Multitask Experiments

Given that the task of detection of DVT relies heavily on picking out extremely small regions in the image, we hypothesize that giving the model locality context would improve classification. To enable more local context, a multitask model for both segmentation and classification was trained with separate output heads for each task. Having separate output heads enabled the network to have some shared parameters to create common representations between the tasks however it also enabled task specific parameters. A common issue in multitask training is if the gradients from one task are against the gradients of the other, the loss of one task would go down and the other may not. We did not see this issue as the two tasks are complimentary to each other (Figure 3). The results show that the addition of segmentation information and backpropagating against a combined segmentation and classification loss led to significantly higher performance on the test set (AUC of 0.69) and was the best model that we tested (Table 1).

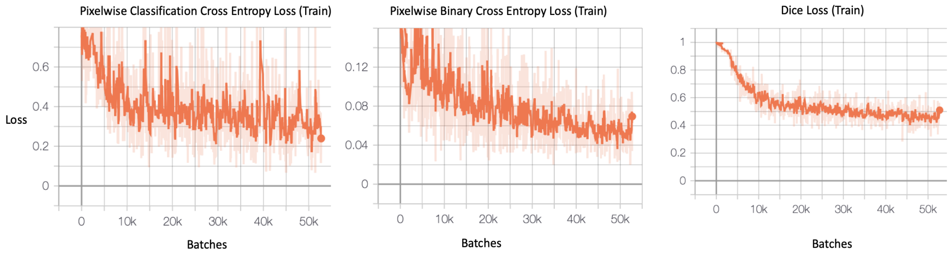


Figure 3: Training Losses for each component of the multitask loss demonstrating task similarity in the optimization process

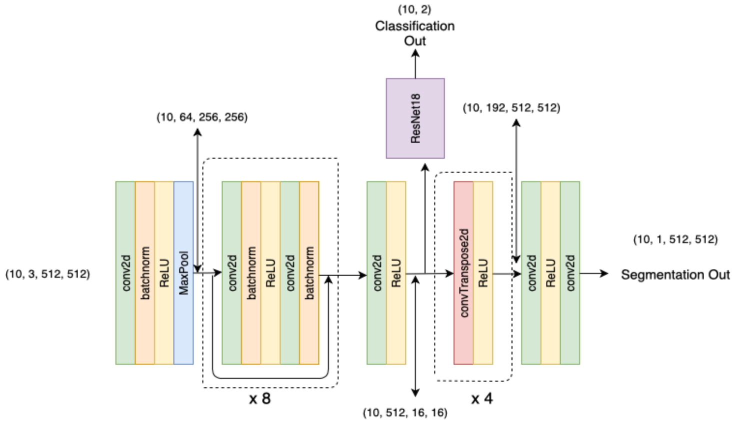


Figure 4: Custom Architecture for joint optimization of segmentation and classification

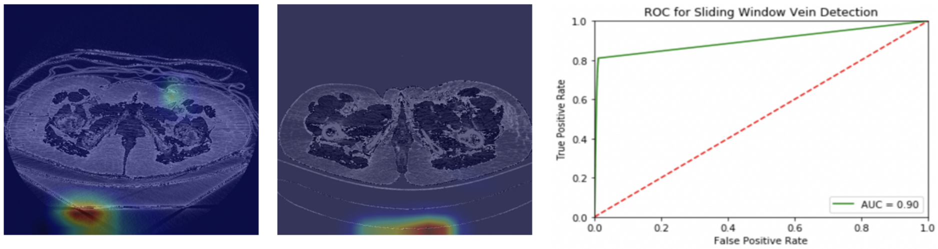


Figure 5: Error Analysis of the Classification Models. (Left) CAMs for incorrectly classified images. (Right) ROC Curve for Patchwise Sliding Window Vein Detection

5.4 Error Analysis

To understand why the classification models were not performing as well as expected, we looked at the class activation maps (CAMs) for images the model gets wrong. A Class activation map for a particular class indicates the discriminative region used by Convolutional Neural Network to identify the class [7]. We see from the CAM that the model is using the edge of the image as opposed to the veins to make the classification indicating an issue with learning the right features (Figure 5).

Based on the above finding, we wanted to know if our models were having issues identifying locations of veins which was contributing to low accuracy for classification of clotted veins. We trained a patchwise sliding window classification model to classify patches of the input image for presence of veins. The patchwise model scores a test **AUC of 0.9** on the vein detection task indicating that the models can identify locations of veins but could not determine if they were clotted.

6 Conclusion and Future Work

We demonstrate that deep learning is effective towards the identification of deep vein thrombosis which could reduce fatalities caused by late detection. It is clear from the experiments that multi-task models that incorporate localizing information through segmentation masks helps the model perform better by focusing on the regions that truly matter for classification. For future work, a cascaded framework with vein identification followed by a thrombosis classification model could boost performance.

The biggest limitation was the quality of the dataset which only contained patients who had abnormalities and lacked consistency in scan quality. For better model performance, we would need scans from patients without any abnormalities and consistency in the quality of the CT scans across patients.

7 Contributions

Our team wrote custom neural network architectures for the multitask and patch classification tasks. We also modified pretrained architectures (DenseNet, VGG-Net) for classification. We wrote custom training, evaluation loops and dataloaders in Pytorch. We also designed loss functions for each task.

Anirudh Joshi structured the problem as a deep learning task and architected the multitask and classification models. He also wrote the training, evaluation loops, loss functions and ran the experiments.

Ishan Shah architected the patch classification model used for error analysis and built the dataloaders for each task. He identified data splits that enhanced model training.

Rui Liu built the baseline non-deep-learning model using Bag-of-Word(BoW) with encoded SIFT image features.

8 Acknowledgements

We would like to acknowledge Dr. Hofmann and his team for supporting us during this project. Special thanks to Anoop and Andre for guidance.

References

[1] M. G. Beckman, W. C. Hooper, S. E. Critchley, and T. L. Ortel, "Venous thromboembolism: A public health concern," *American journal of preventive medicine*, Apr. 2010. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/20331949.

[2] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," *CoRR*, vol. abs/1901.07031, 2019. arXiv: 1901.07031. [Online]. Available: http://arxiv.org/abs/1901.07031.

[3] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, 2017. arXiv: 1711.05225. [Online]. Available: http://arxiv.org/abs/1711.05225.

[4] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, Dec. 2016, ISSN: 0098-7484. DOI: 10.1001/jama.2016.17216. eprint: https://jamanetwork.com/journals/jama/articlepdf/2588763/joi160132.pdf. [Online]. Available: https://doi.org/10.1001/jama.2016.17216.

[5] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study," *PLOS Medicine*, [Online]. Available: https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1002683.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597.

[7] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *CoRR*, vol. abs/1512.04150, 2015. arXiv: 1512.04150. [Online]. Available: http://arxiv.org/abs/1512.04150.