Abnormality Detection in Carotid Ultrasounds with Convolutional Networks

E. Sabri Eyuboglu*
Computer Science
Stanford University
eyuboglu@stanford.edu

Geoffrey Angus*
Computer Science
Stanford University
gangus@stanford.edu

Pierce Freeman*
Computer Science
Stanford University
piercef@stanford.edu

Bhavik Patel
Stanford University
bhavikp@stanford.edu

Mu Zhou
Stanford University
muzhou1@stanford.edu

Katie Shpanskaya
Stanford University
kss@stanford.edu

Kristen Yeom
Stanford University
kyeom@stanford.edu

Matthew Lungren
Stanford University
mlungren@stanford.edu

Abstract

Carotid artery stenosis is a common disease responsible for roughly a quarter of all strokes. In carotid stenosis, plaque deposits in the carotid artery narrow the vessel and reduce or block blood flow, increasing the risk of stroke. In practice, a patient is diagnosed with carotid stenosis using a combination of gray-scale, color Doppler and spectral Doppler ultrasounds.

We present two deep learning methods for automating gray-scale carotid ultrasound screening. The first is an object localization model that crops-out extraneous graphical and textual information in gray-scale ultrasounds. Our object-localization model achieves an intersection-over-union score of 92.1%*. The second is a convolutional neural network trained to detect signs of stenosis in gray-scale carotid ultrasounds. We present a robust analysis of current architectures and their failings in being able to reason about current datasets; this points to a clear need for more granularly annotated data. As the first model trained on this dataset, this outcome serves to further influence the efforts of our radiologist partners.*

1. Introduction

Blood flows to the brain via the carotid artery, a blood vessel in the neck. In a condition known as carotid stenosis, plaque deposits in the carotid artery narrow the vessel and reduce or block blood flow, increasing the risk of stroke. Carotid stenosis affects about 10% of the population, and is the root cause of roughly 25% of strokes [3]. The condition is typically diagnosed by taking some combination of gray-scale, color Doppler and spectral Doppler ultrasounds of the carotid artery [10]. A radiologist interprets the ultrasounds and reports the degree of blockage in the artery. While more advanced Doppler techniques are useful in diagnostics, they are also significantly more expensive and thus limit their deployment in less medically developed nations. Our project investigates deep learning methods for automating the process of gray-scale carotid ultrasound screening. It consists of two parts: (1) carotid artery ultrasound localization in ultrasound machine screen-captures and (2) abnormality detection in carotid artery ultrasounds.

1.1. Carotid Ultrasound Localization

Carotid artery ultrasounds are usually stored as ultrasound machine screen-captures like the one shown in figure 1. These screen-captures include the actual ultrasound image surrounded by extraneous textual and graphical features. A machine learning model meant to interpret carotid artery ultrasounds should not be trained on raw screen-captures because the model could over-fit to textual information surrounding the image or otherwise be hindered by extraneous graphical features. The exact format of the ultrasound operator window varies substantially from exam to exam so cropping the actual ultrasound image is a non-trivial computational task. In the past, ultrasounds have been manually separated from operator windows by researchers.

We present a convolutional neural network (CNN) for automatically cropping ultrasound images in ultrasound machine screen-captures. Our model was trained on a train set of 400 hand-cropped screen-captures and achieves a validation intersection-over-union (IOU) score of 92.1%. We

used our CNN to crop all $21,831$ screen-captures in our dataset of carotid artery ultrasound.

1.2. Carotid Ultrasound Abnormality Detection

Gray-scale carotid ultrasound abnormality detection is a binary classification task that consists of labeling gray-scale ultrasounds as abnormal (i.e. showing stenosis) or normal (i.e. not showing stenosis). Carotid ultrasound abnormality detection is naturally framed as a **multiple instance learning** (MIL) problem [1]. Within electronic medical records, carotid ultrasounds are labeled at the exam level. As a result, our actual datapoints (the ultrasound images themselves) are not individually labeled. In a given abnormal exam, only a small fraction of the images will actually show signs of stenosis so naively applying an exam's label to all of its images and training a classifier is not effective.

Indeed, the sparsity of abnormal images in an abnormal exam presents a significant challenge. To address it, we train a classifier at the *exam level* and add a *sparsity* term to the loss to reflect the fact that only a few images in an abnormal exam will actually show signs of stenosis. Our model achieves an accuracy of 69% on the validation set but performs no better than random chance on the test set, indicating need for further improvement.

2. Previous Work

Convolutional neural networks have previously been applied to the problem of abnormality detection in medical images [2, 6, 8, 11]. For example, Rajpurkar *et al.* used a pre-trained DenseNet to detect Pneumonia in chest x-rays. [9] Together, these studies demonstrate convolutional neural networks' strength in interpreting medical images.

Most of the models from the studies above were trained in a fully-supervised setting; that is, each image in the training set was labeled. Our carotid abnormality detection algorithm cannot be trained in a fully-supervised setting because labels for ultrasound exams are provided at the exam level. Thus, we framed carotid ultrasound abnormality detection as multiple instance learning (MIL) problem. Previously, Zhu *et al.* applied an MIL approach to whole mammogram classification, by splitting an image into grid regions and comparing against the whole image's label [11].

Lekadir *et al.* used convolutional neural networks to characterize the composition of carotid artery plaque [7]. However, convolutional neural networks have not yet been used directly detect to carotid stenosis.

3. Data

We acquired our dataset through Stanford's Center for Artificial Intelligence in Medicine and Imaging (AIMI). The dataset includes $21,831$ gray-scale ultrasound images across 500 exams performed at Stanford Hospital. On average, an exam consists of 43 images and they range in size from 15 to 130 images. The images in a carotid ultrasound exam cover different regions of the carotid artery. If a carotid artery exhibits stenosis, the stenosis will only appear in a subset of the images. Indeed, in a typical abnormal exam, fewer than 10% of the images will show stenosis. An exam is labeled abnormal if any of its images show stenosis, however the images themselves remain unlabeled.

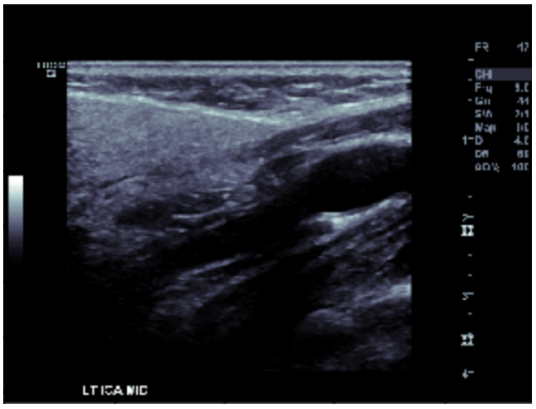

Figure 1. An ultrasound machine screen-capture from the AIMI dataset. The screen-capture includes extraneous measurements and graphics. The ultrasound blends into the background in the bottom left-hand corner, making it challenging to crop. The format of screen-captures vary substantially across the dataset, these is but one example.

As mentioned in the introduction, the images in our dataset are not isolated ultrasound images, but rather ultrasound machine screen-captures. In these screen-captures, the ultrasound image is surrounded by significant textual and graphical information that could bias or hinder a classification model. For example, patient data could be leaked to a classification model through the text surrounding the image. Cropping out the ultrasound from these screen-captures is non-trivial because the formats of the screen-captures vary across exams and the ultrasound image often blends into the background as it does in Figure 1.

To train an object localization model we needed a small dataset of cropped ultrasound. We built a simple annotation GUI pipeline that could be hosted on servers within our closed network. This interface let us navigate to an image, label its four corners and save our labels to a database. It randomized the examples that were displayed so we could annotate in parallel across our whole group. We annotated a dataset of 500 screen-captures.

4. Ultrasound Localization

4.1. Problem Definition

Our raw dataset is comprised of ultrasound machine screen-captures taken at the time of the examination (see

Figure 1). Ultrasound localization or cropping is the task of finding the minimum bounding-box of the ultrasound image in a screen-capture.

We frame the ultrasound localization problem as a regression task on four continuous variables. More formally, given some raw screen-shot $X^{(i)}$ predict the bounding box coordinates of the ultrasound image $Y^{(i)}$:

$$\text{Input: } X^{(i)} \in \mathbb{R}^{H \times W}$$

$$\text{Output: } Y^{(i)} = \{x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}\}$$

where H and W are the height and width of the image. With accurate bounding-box coordinates in $Y^{(i)}$, one can crop the ultrasound image from the screen-capture.

4.2. Methods

Let us define our model G to be some function parameterized by a set of weights θ. G takes an image $X^{(i)}$ as input and outputs bounding box coordinates $\hat{Y}^{(i)}$:

$$G(X^{(i)}; \theta) = \hat{Y}^{(i)} \tag{1}$$

To train our model, we optimize the mean squared error (MSE) loss on the dataset using Adam, a first-order gradient based optimization method [5].

$$L_{\text{MSE}}(Y, \hat{Y}) = \frac{1}{mk} \sum_{i=1}^{m} \sum_{j=1}^{4} (\hat{Y}_j^{(i)} - Y_j^{(i)})^2 \tag{2}$$

4.2.1 Model Architecture

Our model G is a light-weight convolutional neural network (CNN). We intentionally kept our CNN small so that it could be trained with a small number of examples. The model consists of two convolutional layers with 32 filters, batch normalization, dropout and max-pooling. The model architecture is depicted in Figure 2.

4.3. Experiments

Our dataset of 500 annotated images was split into a train set of 400 train images and 100 validation images. Note that in lieu of a test set, we visually verified a random subset of $1,000$ images after cropping to confirm accuracy of the system.

Our first experiment was based on conventional CV techniques, separate from our CNN approach. We performed edge detection on the given examples and ran a contour search to find the bounding box. Initial results showed that the algorithm identified other boxes in the image (eg. diagnostic information) but not the image itself. Upon further analysis, we discovered that the images tended to fade into the background, which was too little of an edge for this approach to be effective.

4.3.1 Evaluation Metrics

The primary evaluation metric for our ultrasound localization model is intersection-over-union (IOU):

$$IOU(Y^{(i)}, \hat{Y^{(i)}}) = \frac{\text{OverlapArea}(Y^{(i)}, \hat{Y^{(i)}})}{\text{UnionArea}(Y^{(i)}, \hat{Y^{(i)}})} \tag{3}$$

We report the average IOU over the train and test set.

4.3.2 Results

Our model was trained for 150 epochs annealing the learning rate by a factor of 10 every 40 epochs.

The localization model successfully performed its task, creating bounding boxes that approximated the borders of the ultrasound images within operating windows with near-perfect accuracy. We report a 94.3% IOU score on the training set at the end of 150 epochs of training, and a 92.1% IOU score on the evaluation set. Upon compiling several hundred bounded images into a video format, we verified the model's efficacy in the regression task by manually validating the regions cropped by the model.

4.4. Post-Processing

After refining the model on our dev set and achieving promising test accuracy, we passed our entire dataset through the localization algorithm. Upon manual analysis, these crops did indeed isolate the ultrasound image by cropping out supplementary diagnostic data.

We noticed, however, that some screen-shots contained metadata instead of ultrasound images. These included patient information, recorded observations, and other manual radiologist information. We denote these "operator images," since they aid operators but shouldn't be included within our model. By our estimates, about 5% of our data were these supplementary images. These posed a real danger of polluting our exams with information that the model could use to overfit.

4.4.1 Histogram Classification

To prune these instances, we first searched for four files that were representative of operator images. We labeled these with a ground-truth label, along with four standard exam images.

We then passed these image datapoints X along with their labels y to a histogram classifier that we implemented. This classifier creates a histogram over three buckets with values from 1-8, resulting in a 3D embedding for each image $\hat{X}$. We save the embeddings of the training data $\hat{X}$ alongside the y labels.

During test time, we run through each new example to get its histogram embedding. We then perform a 1-NN

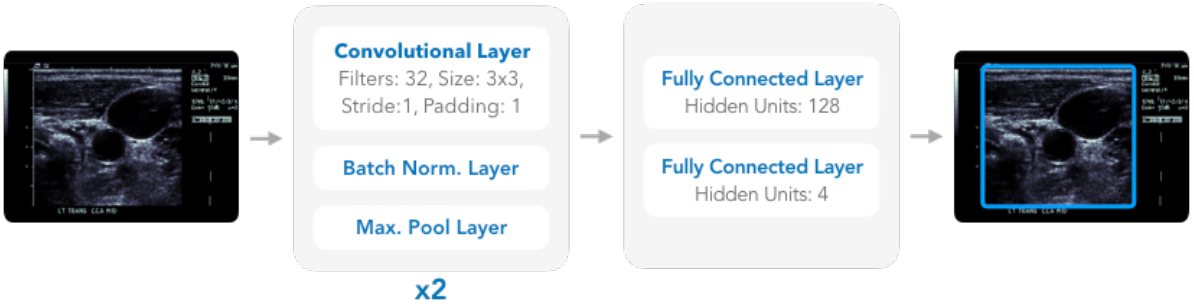

Figure 2. Light-weight convolutional neural network for ultrasound localization. The model accepts an ultrasound machine screen-capture and outputs the minimum bounding box around the ultrasound image.

Model Metrics: Ultrasound Localization	
Set	IOU
Train	94.3%
Validation	92.1%

search on this point to find the most similar labeled neighbor. This postprocessing process decreased the presence of these data frames about tenfold, to around .5% of our total dataset.

5. Abnormality Detection

5.1. Problem Definition

We frame the carotid ultrasound abnormality detection problem as a *multiple instance learning* (MIL) problem. In typical MIL, a label Y is provided for a whole bag of examples $X = \{X^{(1)}, ..., X^{(n)}\}$. In the context of our task, we ascribe $X^{(i)}$ to be the set of images associated with some patient i, and $Y^{(i)} \in 0, 1$, where 0 is a normal diagnosis and 1 is an abnormal diagnosis.

In the event where some patient i has a normal diagnosis, we expect that all $x \in X^{(i)}$ have the attributes of a normal image. However, in the event of an abnormal diagnosis, we expect that a small number of $x \in X^{(i)}$ have the attributes of an abnormal image. As we mentioned previously, only around 3 of 100 have these attributes.

We constructed a multi-stage model consisting of two primary modules in order to accomplish this MIL problem: 1.) an image module, responsible for encoding the most salient features of all given images and 2.) an aggregation module, responsible for aggregating the image encodings and making the final classification for some patient.

5.2. Methods

Data Augmentation

In order to increase model performance on unseen data, we chose to randomly apply transforms to training images from epoch to epoch. The transforms applied included the following:

1. Random Horizontal Flip
2. Random Rotation (up to k° where k is a tuneable hyperparameter)
3. Random Color Jitter (offsets on saturation, contrast, and brightness)

These transforms were applied along with a padding operation, which pasted the image onto a black background in order to standardize the image size required for some of our convolutional networks. With these transforms in place, we were able to create a data pipeline capable of feeding in an unlimited quantity of permuted data.

5.3. Model Architecture

5.3.1 Image Module

DenseNet-121

DenseNets, first proposed by Huang *et al*. [4], have been shown to be effective in abnormality detection on medical images [9]. Our first iteration of the abnormality detection model is simply DenseNet pretrained on ImageNet. We removed the final layer of DenseNet and added a sigmoid activation layer in order to turn it into a binary classifier. We did this knowing that the model itself was pretrained on natural images; however, we also see in that this is something that has been done before and thus a reasonable baseline for our classification task.

ResNet-18 and AlexNet

Our preliminary tests using the DenseNet image module exhibited symptoms of dramatic overfitting. In the interest of regularization, we decided to experiment with smaller architectures with less parameters, such as ResNet-18 and AlexNet. AlexNet was not successful in training and

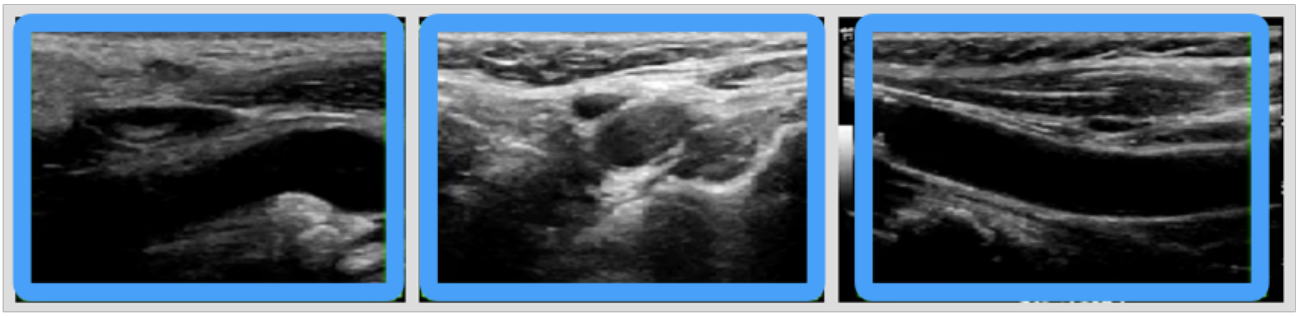

Figure 3. The results of our image localization algorithm. The model performed a regression task on coordinate points, achieving a 92.1% Intersection Over Union (IOU) score.

ResNet-18 exhibited similar behavior as DenseNet-121.

VAE Classifier

There are nearly two orders of magnitude more data in ImageNet than in our ultrasound dataset. Our previously described models - which perform well on ImageNet - have enough parameters to easily overfit on a dataset of our size. Additionally, the GPU memory requirements of these models forced us to split an exam into multiple batches, which is a slight deviation from our original problem formulation. As part of a low-memory regularizing strategy, we explored VAEs as an unsupervised method to decrease our feature space.

Variational Autoencoders (VAE) have shown great promise in extracting distinguishing features within images. They rely on learning a compression that's able to roughly decode an original image I from an embedding X. In essence, for an encoder CNN E and decoder CNN D, the VAE attempts to optimize $I = D(E(X))$. We trained 32-dimensional embeddings with the encoder structured as: CONV4-32 $\rightarrow$ ReLU $\rightarrow$ CONV4-64 $\rightarrow$ ReLU $\rightarrow$ CONV4-128 $\rightarrow$ Conv4-256 $\rightarrow$ 3xFC-32.

VAE's are unique in a few ways when compared to their vanilla autoencoder counterparts. One, embeddings are forced into a continuous space which allows extrapolation between datapoints. Relationships like addition and subtraction can have more interesting dynamics within this space. They can also be clustered and visualized, as we've done below.

5.3.2 Aggregation Module

The aggregation modules were developed over time, as we learned more about the problem definition. The first aggregation module we implemented took the average of the 1024-element feature embeddings outputted by the image module. The resulting 1024-vector was then passed through a fully connected layer in order to output a 2-vector containing un-normalized outputs to be fed into a softmax loss function. See Figure 4.

We experimented with a variety of other aggregation modules in an attempt to address the fact that abnormal-labeled exams contain many images that would be otherwise labeled normal. One such aggregation module utilizes a fully connected layer in order to output classification scores for each of the images, and then outputting the score of the image with the maximum abnormal classification score. Intuitively, we do this in an attempt to use the classification scores of only the most abnormally-perceived images at train time. Formally, we formulate the classification output formula for this aggregation module with the following, where $Z^{(i)}$ is the set of image embeddings for exam i and W and b represent the weights and bias of the final linear layer:

$$\hat{y}^{(i)} = \max_{z \in Z^{(i)}} (W^T z + b)_2$$

Additionally, we implemented an RNN-based aggregation module with the underlying assumption that context is necessary to detect overarching abnormality across ultrasound images. This aggregation module first passes the image module output through a linear layer to some prescribed embedding size, then passes the embeddings into a many-to-one multi-layer bidirectional LSTM as sequence data. Upon passing through all of the images in a given exam, the LSTM would output some final hidden state, which would then be passed through a linear classification layer.

Finally, we implemented an aggregation module that combined the two aforementioned methods. We implemented a many-to-many multi-layer bidirectional LSTM that outputted learned embeddings for each image instead of generating some final embedding meant to be interpretable by a linear layer. These outputted hidden states for each image were then fed into a linear classification layer, which was then maxed in order to find the most abnormal image from within the exam. By combining the two methods, we address the possibility that anomalous artifacts are often times more evident when put in comparison to a number of data-points.

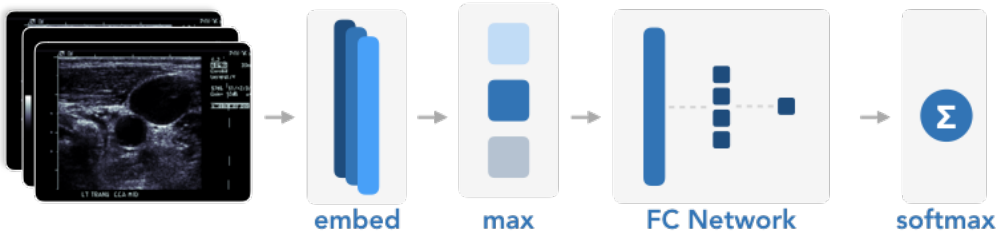

Figure 4. Diagram of aggregation module. Images are converted to their embedding space, either as an output of a ImageNet module or a VAE. They are then passed through a max layer, to accentuate the one that is most likely to be abnormal. This embedding is then passed to a

5.3.3 Sparsity Loss

As mentioned above, as few as 10% of the images in an abnormal exam will actually show stenosis. As a result, if we naively apply cross-entropy loss to the exam labels and the outputs of the aggregation module, our model will likely learn to label a huge number of normal images as abnormal. To overcome this issue, we add a term to the loss function that encourages the image module to only output a few abnormal images per exam. More formally, to enforce sparsity in the image module predictions we add a sparsity term to the loss function as was done by Zhu *et al.* [11]. The complete sparsity loss is then:

$$\ell(Y^{(i)}, \hat{Y}^{(i)}) = \text{CE}(Y^{(i)}, \hat{Y}^{(i)}) + \alpha \sum_{j}^{|X^{(i)}|} \text{CE}(0, \hat{Y}_j^{(i)}) \quad (4)$$

where CE is the cross-entropy loss, α is a tuneable hyper-parameter, and $\hat{Y}_j^{(i)}$ is the image modules prediction on the j^{th} image.

5.4. Experiments

We ran several dozen experiments utilizing different combinations of image modules and aggregation modules in order find the model best suited for the task of ultrasound abnormality detection. Preliminary tests involved pairing a pre-trained Densenet-121 and the average merge aggregation module. These tests showed symptoms of dramatic overfitting, and thus the task ahead of us involved the exploration of various regularizing techniques and architecture reformulation with the ultimate goal of increasing model performance on unseen data.

Each of our experiments struggled to combat overfitting. L2 regularization and dropout were unsuccessful in bringing up the accuracy on the evaluation set and this caused us to try many other regularization techniques, such as the variety of data augmenting transforms described prior. The assumptions we made about both sparsity and the effect of the RNN and the max function held true to some extent as evidenced in the results of our top-3 models.

	Eval Acc	Test Acc
DenseNet-AvgMerge	56.0%	48.0%
DenseNet-MaxSparsityMerge	65.0%	60.0%
DenseNet-RNNMaxMerge	69.8%	50.0%

Table 1. The results of our top-3 models. Our models overfit dramatically and while a couple show promising evaluation set results, we see here that the model produced results on the test set that approximately equaled random.

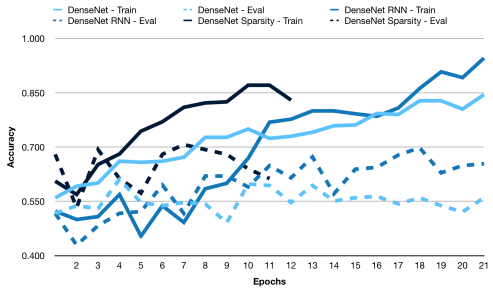

Figure 5. Experiments with differing aggregation modules. Train accuracy (solid lines) climbs across the board while dev (dotted line) stagnates around 50%, which would be near random.

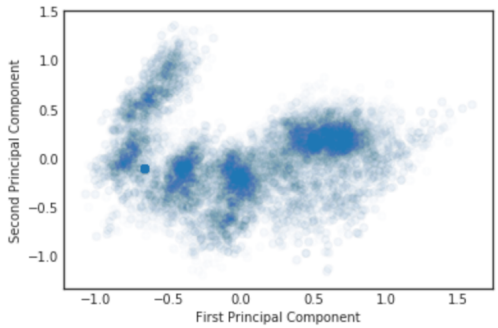

Figure 6. A Principal Component projection of the image embeddings learned through 50 epochs of a CNN VAE encoder.

5.4.1 VAE Classifier

We tested a variety of fully connected layer combinations to classify our VAE embeddings. Depending on how many parameters we fixed as our hidden layer sizes, the model

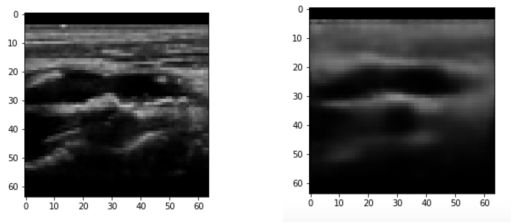

Figure 7. Left: Original Image, Right: Decoded image from the learned VAE encoding

showed varying abilities to overfit. Within all these situations, however, the dev accuracy oscillated around 50%.

Despite learning something fundamental about the positive/negative space (as seen in Figure 1, it wasn't able to generalize to other examples. We hypothesize this is simply because of the structure of our data. Our model is not only having to accommodate an imbalance in data, but it's also having to self-determine which ones are malignant because only the exam is labeled.

6. Conclusion & Future Work

Our experiments confirmed that abnormality detection in carotid ultrasound image data is indeed a challenging task. While we did not get the results that we hoped to achieve, we conducted several dozen experiments across a significant number of model architectures in hopes of reducing the high variance in model performance. Along the way, we built a robust medical imaging data pipeline that allowed us to develop and test our model both efficiently and systematically.

We are fortunate in the fact that we can deem our implementation of an image localization model for medical imaging data a success. According to our team at AIMI, the model we have built is capable of completing a task in seconds that which currently medical staff months to complete by hand.

We believe that there are multiple avenues that can be taken to see more success on this particular task. Due to the fact that at most 10% of the images in the dataset contained abnormal attributes, we believe that more data is necessary for the learning task required of our models. Further, there are many solutions to MIL tasks documented that remain unimplemented by our team. Future work would include implementing and assessing these other solutions, with the understanding that nearly half the images in the dataset that would otherwise be considered normal are labeled abnormal due to the patient's overarching condition.

Overall, our endeavor saw mixed results, but the data and experimentation pipeline and image localization models trained are outcomes that hold high promise for those interested in continuing to apply deep learning methods to medical imaging tasks.

7. Special Thanks

We'd like to thank our partners at AIMI for their help in acquiring this dataset and discussing diagnostic techniques. Nishith Khandwala also helped with discussing architecture choices and brainstorming on bug hunts. We'd also like to thank the pytorch vision team for their modal zoo, which provides pre-trained weights for common vision tasks. In addition, we'd like to thank GitHub user 'sksq96' for his implementation of a CNN VAE.

References

[1] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 5 2018.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2 2017.

[3] A. Y. Fung and J. Saw. Epidemiology and Significance of Carotid Artery Stenosis.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks.

[5] D. P. Kingma and J. L. Ba. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.

[6] P. Lakhani and B. Sundaram. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. *Radiology*, 284(2):574–582, 8 2017.

[7] K. Lekadir, A. Galimzianova, A. Betriu, M. del Mar Vila, L. Igual, D. L. Rubin, E. Fernandez, P. Radeva, and S. Napel. A Convolutional Neural Network for Automatic Characterization of Plaque Composition in Carotid Ultrasound. *IEEE Journal of Biomedical and Health Informatics*, 21(1):48–55, 1 2017.

[8] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, A. Ng, and p. S. Edu¿. MURA Dataset: Towards Radiologist-Level Abnormality Detection in Musculoskeletal Radiographs.

[9] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning.

[10] H. R. Tahmasebpour, A. R. Buckley, P. L. Cooperberg, and C. H. Fix. Sonographic Examination of the Carotid Arteries. *RadioGraphics*, 25(6):1561–1575, 11 2005.

[11] W. Zhu, Q. Lou, Y. S. Vang, and X. Xie. Deep Multi-instance Networks with Sparse Label Assignment for Whole Mammogram Classification.