Implementing Contrastive Triplet Loss to Improve Classification on Chest X-Rays

Jessica Breda and Claira Fucetola

Introduction

The emergence of deep learning models has led to significant advances in computer aided medical diagnostics. Yet, despite its importance, this progress has been limited relative to other classification tasks in the field of computer vision (Wang et al. 2018). The primary cause of this is that high-quality labeled medical data is limited. Collection and annotation of medical data is time consuming and requires expertise. Moreover, there are additional privacy considerations when sharing medical data. Such challenges make it difficult to train and implement models that perform well and generalize across individuals and pathologies. Progress in this field is important given that the predictions from these models can pertain to life-threatening diseases.

Furthermore, due the growing knowledge on the long term effects of COVID-19, a substantial amount of research has been dedicated to performing classification tasks on Chest-XRay datasets to identify COVID-19 induced pneumonia (Maghadid et al, 2021; Narin et al. 2021; Tschandl 2021). To address the challenge of limited high-quality labeled data, researchers have explored strategies such as data augmentation, contrastive learning pre-training, and a variety of neural network architectures to improve classification accuracy for medical diagnosis. A successful approach for improving feature extraction in deep learning for pneumonia diagnosis is to pre-train a base network using a contrastive learning triplet loss function (Gupta et al. 2018; Sanz et al. 2021). This approach is less computationally expensive than other contrastive loss functions (e.g. SimCLR, InfoNCE), yet is capable of showing state of the art performance (Gupta et al. 2018).

Here, we are interested in replicating the results demonstrating that supervised pre-training with a contrastive triplet loss function can improve multi-class classification on Chest-X Rays. We begin by examining the effect of supervised pre-training on classification accuracy in a simple convolutional neural network (CNN). Then, we experiment with data augmentation, training data sampling, and modeling complexity. In summary we found that pre-training with a contrastive triplet loss on triplets where the positive image was augmented resulted in a 6% increase in accuracy and a reduced tendency to randomly classify images when compared to no pre-training.

Data

A subset of MNIST datasets is MedMNIST which holds a collection of datasets for a variety of medical tasks, including ChestX-Rays: ChestMNIST (Yang et al. 2023). The train set, test set, and validation set initially hold 78,468, 22,433, and 11,219 data points respectively. ChestMNIST is a multi-label dataset corresponding to 15 classes: Normal, Atelectasis, Cariomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Edema, Emphysema, Fibrosis, Pleural, and Hernia (**Figure 1**).

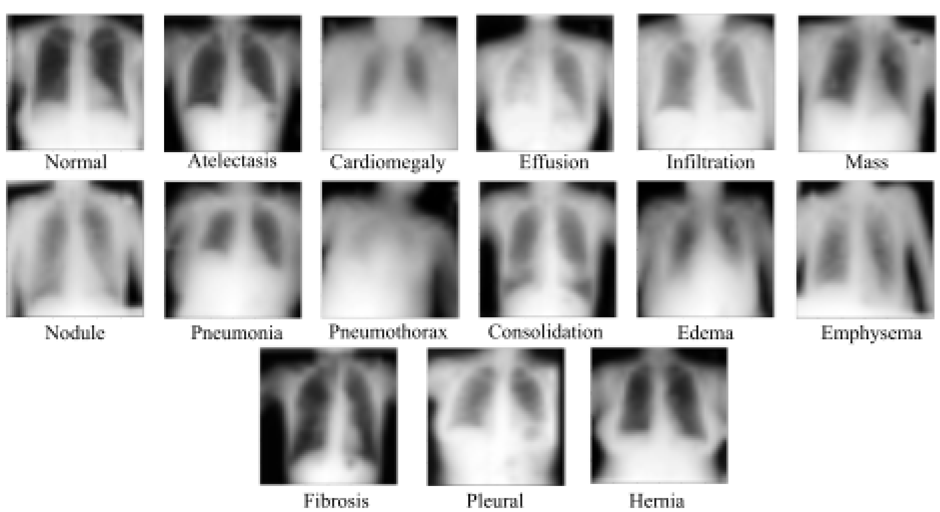

Figure 1: Images & labels corresponding to each of the 14 classes in the ChestMNIST dataset

Data Preparation

The images in the dataset were resized from 1x28x28 grayscale, to 3x32x32 grayscale. turned to a tensor, and normalized using PyTorch transforms. The data contains multi-label classes, such that each image might belong to multiple classes. To address this, pruning was performed to remove multi-class instances from the dataset for the purposes of training resulting in a training set, test set, and validation of sizes 64007, 18187, and 9191 respectively. Moreover there was an imbalance in the number of images that are "normal". To mitigate this, the WeightedRandomSampler was used to upsample underrepresented classes and downsample over represented classes in our data loader.

Augmentation

Augmentation for medical images requires techniques which are considered clinically practical. Otherwise, the data can be changed in a way that leads to misclassification and learning incorrect features. For this reason, transformation such as reflections across the x or y axis are not feasible as they drastically change the orientation of the image. Techniques that are considered to be clinically practical are translation across x or y axis, slight rotations between -5° and 5° (Elgendi et al. 2021), and cropping of field of view (Madani et al. 2018). The following transformations were applied to the training set using PyTorch transforms: random crop (image upscaled to 300x300, 250x250 crop taken, and then downscaled back to 32x32), random rotation (min: -5°, max: +5°), and random affine translation about the x and y axis (0.1, 0.1) (**Figure 2**).

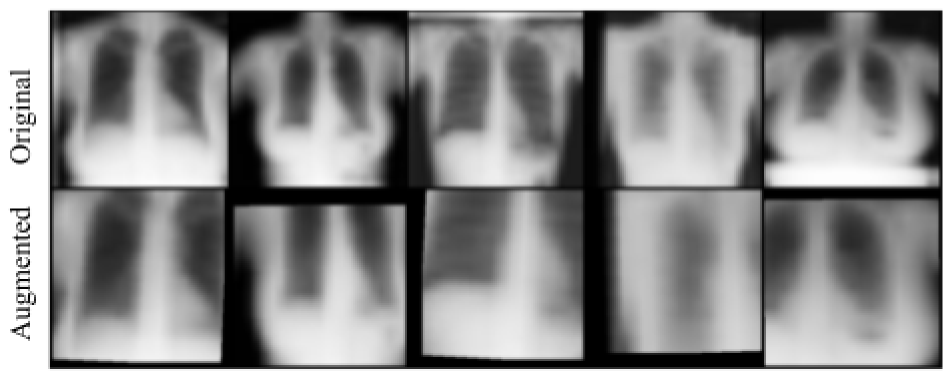

Figure 2: Visualization of augmentation techniques on five data points where the original image (top) has been transformed with random crop, x-axis translation, and rotation (bottom). Each displayed image and corresponding augmented image is labeled "Normal"

Methods

Baseline Model

The baseline model consist of 3-layer CNN followed by a flattening layer, h(*), for feature extraction with an added classification head, k(*). The classification head consists of two fully connected layers with ReLU activation, with a dropout layer added in between, and softmax activation on the outputs (**Figure 3, left**). The encoding feature network h(*) was inspired by the VGG architecture with smaller (3x3) filters, but significantly reduced in depth to allow for more efficient experimentation given computational resources. We trained this model for 50 epochs and batch size 32 using cross entropy loss and stochastic gradient descent (SGD) with dropout rate 0.7 (30% output ignored), learning rate 0.1 and momentum 0.9. Weights for all models were initialized with the Xavier method.

Triplets

A triplet is a group of anchor (a), positive (p), and negative (n) data points. The anchor and positive data points are of the same class and the negative is of a different class (**Figure 4**). Supervised pre-training with triplet loss aims to maximize the distance between the embeddings of the anchor and the embeddings of the negative and minimize the distance between the embeddings of the anchor and the embeddings of the positive datapoint.

$$L(a, p, n) = max\{d(a_i, p_i) - d(a_i, n_i) + m, 0\}$$

where $d(x_i, y_i) = ||x_i - y_i||_2$ and m is the margin

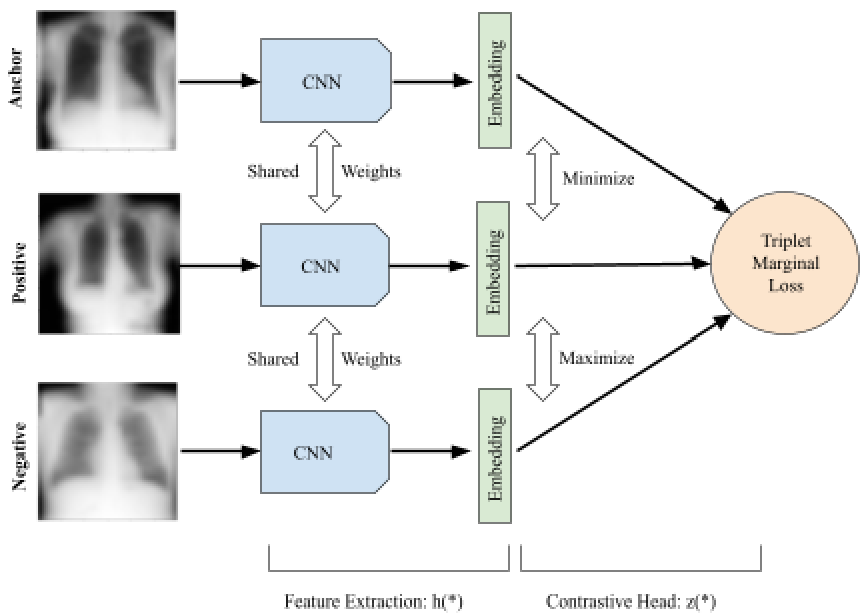

Figure 4: Visualization of Supervised Contrastive Pre-training Training with Triplets.

Pre-Training with Contrastive Learning

To improve performance of the network, we added a supervised pre-training step using a contrastive head z(*) before the classification head that consisted of two fully connected layers with a ReLU activation (**Figure 3, right**). We trained the model for 250 epochs and a batch size of 32. We used a triplet loss function (**Equation 1**) to train the network with margin (m) 0.3, and SGD with learning rate 0.01 and momentum 0.99.

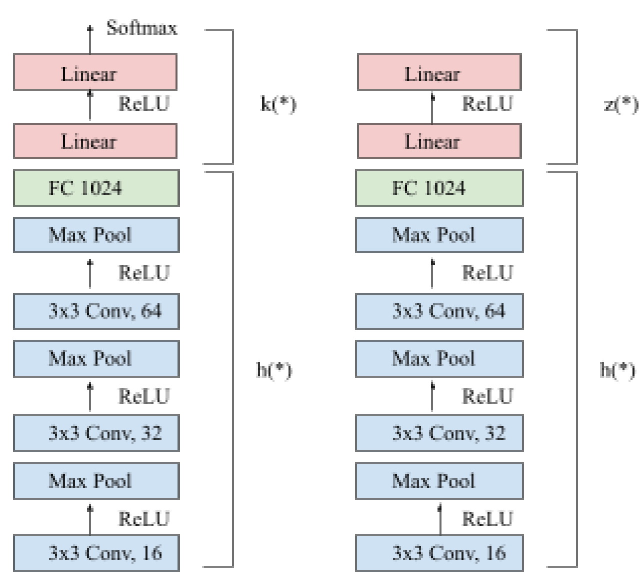

Figure 3: Architecture for the classification model (left) and contrastive model (right)

When augmented positive triplets were used the parameters were 100 epochs with a learning rate of 0.1, momentum of 0.9, margin of 0.5, and batch size of 32.

There are 15 classes in this dataset, meaning there are 15 possible unique anchors. Triplets were sampled in three different ways. The “normal” was to iteratively select a positive and a negative image given an anchor on each epoch. The “hard” approach was to sample the most similar negative and most distance positive to compose a triplet (Schroff, Florian, et al.). The “augmented” approach was to apply an augmentation (Figure 2) to the positive image in the triplet.

To implement hard triplet loss, we imported “batch_hard_triplet_loss” from “online_triplet_loss” which, for each embedding, finds the “hardest” negative and positive image to use in the loss function (Rishaug). The motivation behind hard triplet loss is to find the negative image that is most similar to the anchor and the positive image that is most dissimilar from the anchor, and then maximize and minimize the distance respectively. This way the extrema for each image has been considered (Schroff, Florian, et al.).

Fine Tuning with Classification

Once pre-training was complete, we froze the weights of the feature layers, h(*) and returned to using the classification projection head k(*). We fine tuned this head on the classification task as described in the baseline model with 60 epochs, 32 batchsize, 0.01 learning rate, dropout at 0.7 (30% of outputs ignored) and 0.9 momentum. When augmented positive triplets were used the parameters were 150 epochs with a learning rate of 0.02, momentum of 0.9, and batch size of 32.

The experiment that received the best performance (pre-training with augmented positive triplets) went through an additional classification. The features are frozen h(*) and the classification head is fine tuned with a larger training set. To achieve this, the training set doubled by augmenting all the images in the training set and adding them to the training set (training set size 128014 images). The augmentation here is independent of the triplets.

Increasing Base Model Complexity

To test if a deeper, more complex model for improved classification accuracy with and without contrastive learning, a set of experiments was run using EfficientNet as the feature extractor. Similar to above, this model consists of a backbone model built by EfficientNet (h(*)) for feature extraction and a classification head (k(*)), **Figure 5**).

The backbone is constructed by importing efficentnet-b0 with pretrained weights (Tan & Le 2019). Since EfficientNet was trained on ImageNet, and ChestMNIST is quite a different dataset, fine-tuning on parameters needed to be deeper. Therefore the top 150 parameters are learned and the bottom 80 parameters are frozen. The classification head consists of three fully connected (FC)

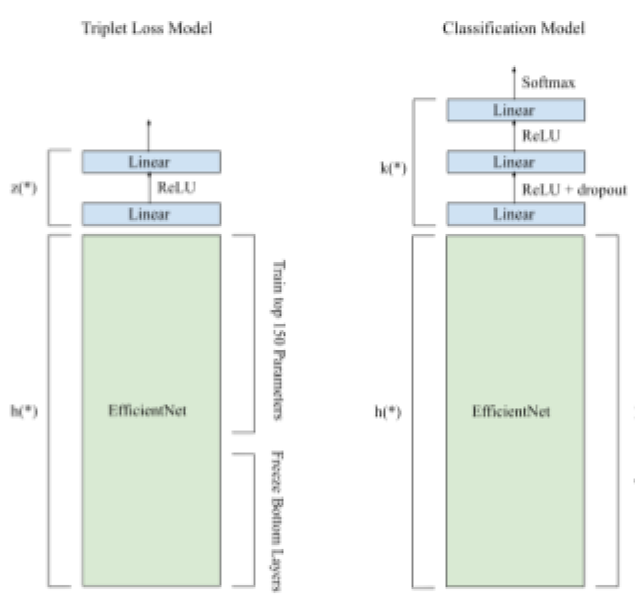

Figure 5: Model Architecture for Triplet Loss and Classification with EfficientNet. Frozen parameters are indicated for both contrastive learning with triplet loss and classification.

layers with dropout between two of the FC layers for regularization, ReLU activation, and softmax activation on the outputs. Weights for all models (apart from the frozen layers) were initialized with the Xzavior method. We trained this model for 30 epochs and batch size 32 using cross entropy loss and stochastic gradient descent (SGD) with learning rate 0.001 and momentum 0.9, and drop out rate of 0.7.

Hyperparameter Selection

A suitable range for learning rate was found using random search and tracking the training loss and accuracy across epochs and trials. Additionally, margin, batch size, and dropout rate was decided by tracking the loss function for both the training set and validation set and updating parameters accordingly to achieve appropriate convergence.

Analysis

We performed supervised pre-training and classification by training our models on the Train Set and tuning parameters based on the Validation Set. The embeddings were evaluated in terms of training and validation loss, and T-SNE clustering, while the classification performance was assessed based on top-1 accuracy, top-5 accuracy, network accuracy, and AUC (Area Under receiver operating characteristic Curve). Additionally, we evaluated the effectiveness of supervised pre-training with contrastive learning on classification.

Experiments

1. **Classification with and without contrastive learning pre-training.** In this experiment, we tested whether contrastive learning with "normal" triplets improves classification performance within a mini-VGG network (**Figure 3**).

2. **Triplet modulation.** In this experiment, we tested if and how three different types of triplets affected contrastive training. Specifically, we compared embeddings and performance when normal, hard, or augmented triplet sets were used (see **Methods- Pre-Training with Contrastive Learning**). We then applied augmentation to increase the size of the training set (independent of the triplets) to compare the performance on classification (see **Methods- Pre-Training with Contrastive Learning**).

3. **Model complexity.** In this experiment, we tested if a deeper, more complex base model would improve classification and post contrastive-learning (**Figure 5**).

Results

Experiment 1: Baseline Model Network Classification

Classification via baseline model (**Figure 3, left**) reached a highest validation accuracy of 19.03%. Overall the model received an accuracy of 13%,with a top-1 accuracy of 13% and top-5 accuracy of 47% when running on the test set (**Figure 13**). The model was able to best recognize classes 2 and 10 (Cariomegaly, Edema) achieving 0.7131 AUC and 0.7049 AUC respectively, which is reflected in the confusion matrix (**Figure 6**). Full training of the model performed particularly poorly on classes 0, 3, 7, and 11 (normal, effusion, pneumonia, emphysema). The model was unable to identify any images from class 3 (effusion) and only incorrectly identified images for class 11 (emphysema). Furthermore, the plot of the AUC for each class shows a majority of classes performing equal or near to random, with class 0 (normal) performing below random.

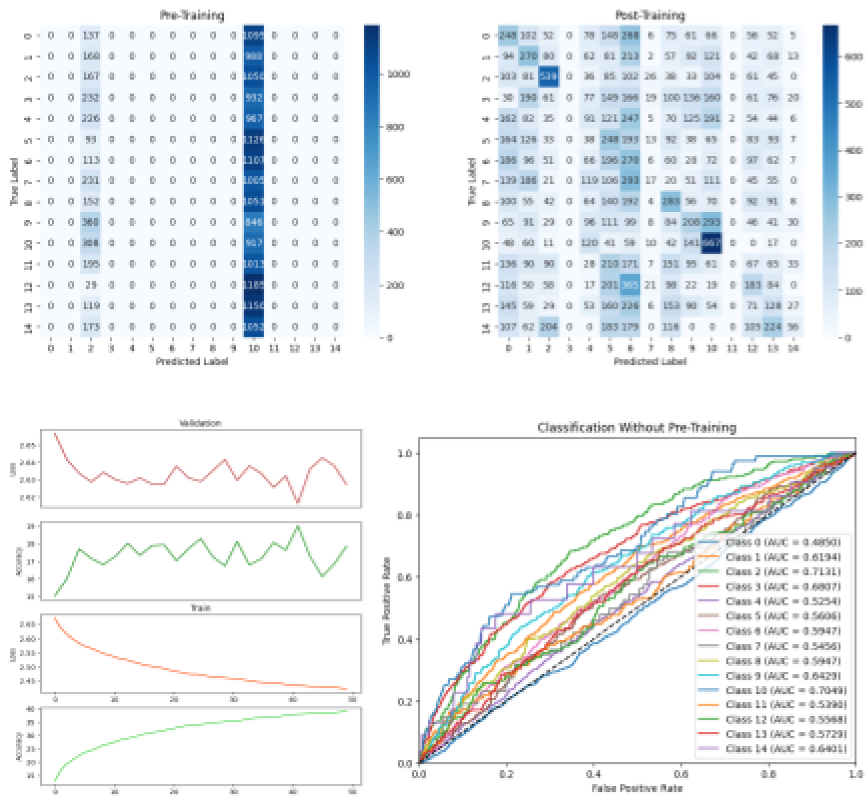

Figure 6: classification results on the baseline model with no contrastive learning pre-training

Experiment 1: Contrastive learning embeddings with Normal Triplets

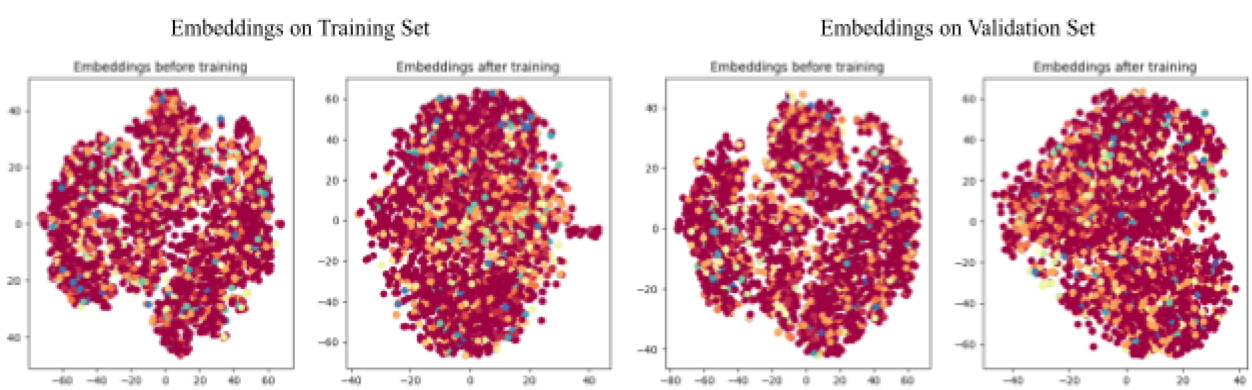

Figure 7: Embeddings from the Train Set (left) and the Validation Set (right) after training h(*) with normal Triplets

When using the contrastive head of the model with a triplet loss (**Figure 3, right**), the training loss was 0.08223 and the validation loss was 0.231 at the end of training. The lowest validation loss achieved using this method was 0.182. In the T-SNE plots of the embeddings (**Figure 7**), each color is a different class and red represents the "normal" chest. It can be noted the majority of the samples are normal, and this is why weighted random sampling was used. When comparing the embeddings before and after training, there appears to be some clustering of the normal samples, but the overall structure is not cleanly separated across classes post training.

Experiment 1: Fine Tuning Classification post contrastive learning using Normal Triplets

After pre-training with contrastive learning, fine tuning was performed by loading the weights pre-trained, freezing h(*), and learning the weights on the liner layers for classification, k(*) (**Figure 3, left**). Overall, the model with fine tuning received an accuracy of 18%, top-1 accuracy of 18%, and top-5 accuracy of 53% when running on the test set (**Figure 13**). The model received the highest validation accuracy during the training process of 20.25%. The AUC shows an affinity towards classes 2 and 10 (Cariomegaly, Edema), receiving 0.7680 AUC and 0.7698 AUC respectively. There remains a visible weakness towards classifying images in class 7 (pneumonia) (**Figure 8**), but this model shows significant improvement in classifying class 3 (emphysema) with 431 True Positives (TP).

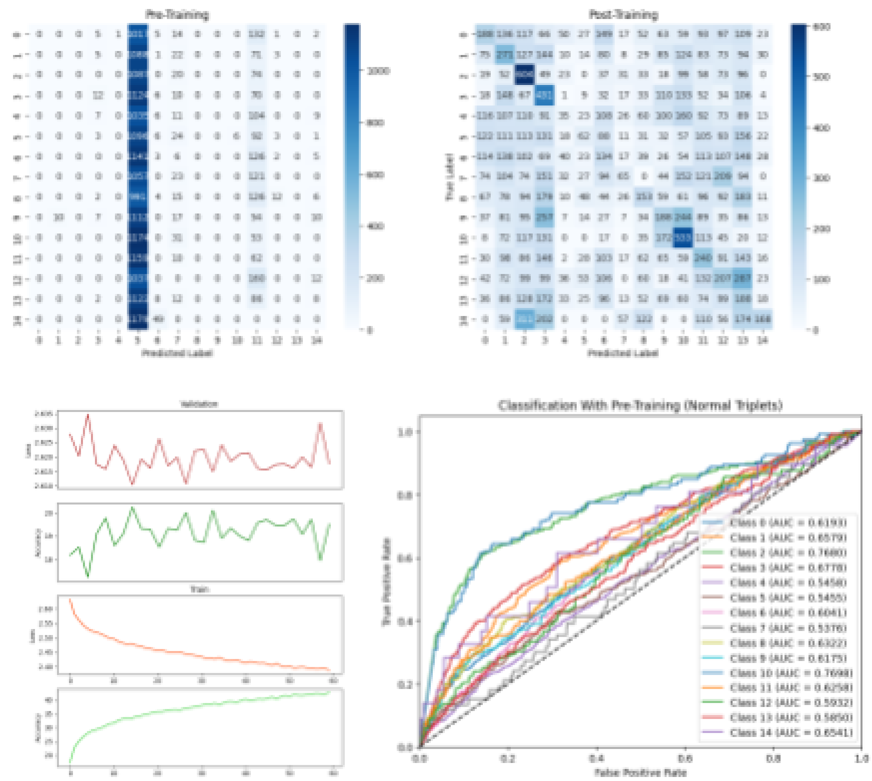

Figure 8: classification results on the baseline model with contrastive learning pre-training on normal triplets

Experiment 2: Adding Augmentation to Positive Images in Triplets during Pre-Training

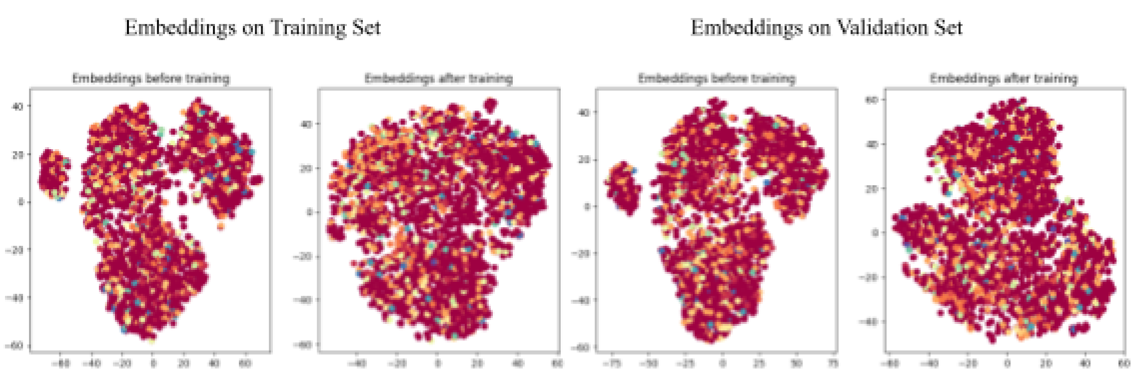

Figure 9: Embeddings from training set (left) and validation set (right) after training h(*) with triplets where the positive image is augmented.

In this experiment, the positive images for the triplets were augmented using the transforms described in the Data section. The training and validation loss reached a minimum of 0.12 and 0.43 respectively. Similar to **Figure 9**, there is no clear distinction between classes but there is an observable affinity to push classes that are not “normal” to the outside and cluster the “normal” class together.

Classification accuracy improved when fine tuning classification with positive triplet augmentation. The model reached a high for validation accuracy during the first 50 epochs of 20.70% and then plateaued for the remaining at about 19% validation accuracy (**Figure 13**). The training loss curve indicates an appropriate learning rate and the confusion matrix displays more of a diagonal trend. Overall the model received an accuracy of 19%, top-1 accuracy of 19%, and top-5 accuracy of 55% when running on the test set. The AUC continues to show a preference in the model towards classes 2 and 10 (Cardiomegaly, Edema), receiving 0.8115 AUC and 0.8093 AUC respectively, which is further improvement from fine tuning from the normal triplets. There is continued difficulty towards classifying class 7 (pneumonia), and class 4 (infiltration) performed close to random based on the AUC. However, more classes were pushed farther away from random which likely accounted for the performance improvement (**Figure 10**).

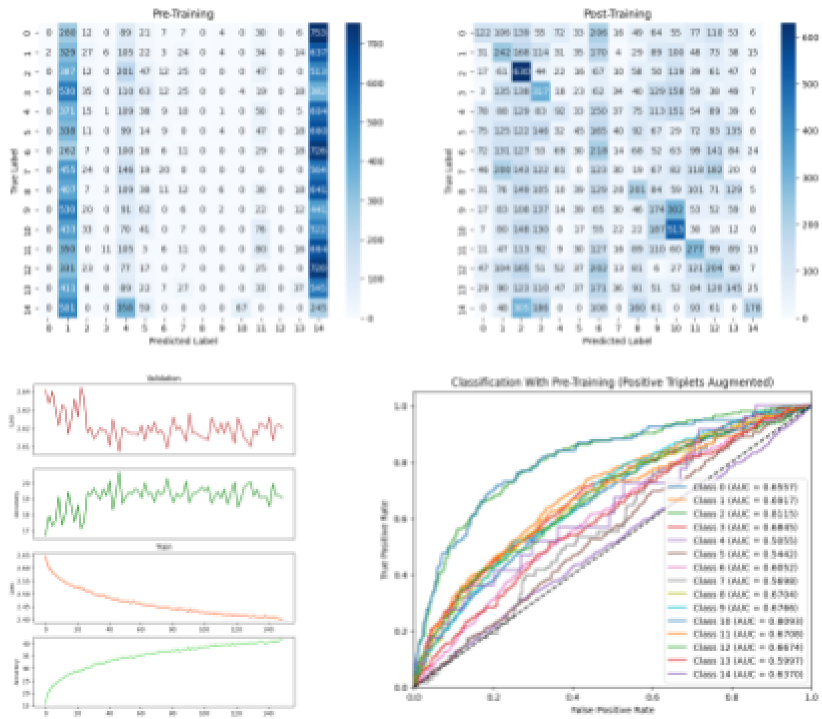

Figure 10: classification results on the baseline model with contrastive learning pre-training on triplets with positive image augmented.

Experiment 2: Adding Augmented Data to Increase Train Set for Best Embedding

By adding augmented data (see **Methods - Pre-Training with Contrastive Learning**), we doubled our train set and added an element of randomness. This further improved the performance of our model resulting in model accuracy of 19%, top-1 accuracy of 19%, and top-5 accuracy of 56% when running on the test set (**Figure 13**). While the AUC still shows a preference towards classes 2 and 10 (Cardiomegaly, Edema), receiving 0.7976 AUC and 0.8087 AUC respectively, all classes perform above random (**Figure 11**).

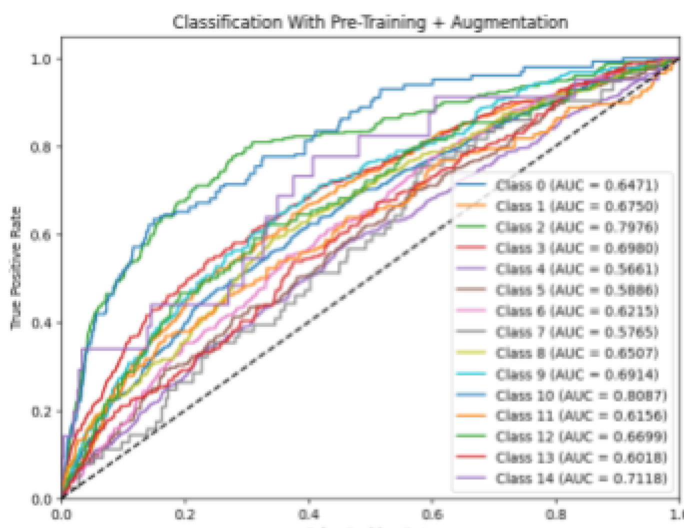

Figure 11: AUC classification results on the baseline model with contrastive learning pre-training on triplets with positive image augmented and with Training Set increased by addition of augmented images.

Experiment 2: Using Hard Triplets

The model failed to accurately classify images with contrastive pre-training using Hard Triplets. The hard triplet embeddings achieved a minimum training loss of 0.52 and a minimum validation loss of 0.5. When applying the frozen embeddings to the classification task, it resulted in overall model accuracy of 6%, top-1 accuracy of 6%, top-5 accuracy of 32%, and highest validation accuracy of 6.5% (**Figure 13)**.

Experiment 3: Efficient Net

The embeddings resulting from transfer learning on EfficientNet with normal triplets were the best embeddings we were able to find. The minimum training loss found was 0.002458 and the minimum validation loss found was 0.580. From the embeddings above (**Figure 12**) a clear separation of the data is beginning to form, where the bottom cluster holds majority class 0 and the top holds the other classes, separating the normal and abnormal chest x-rays.

Applying these embeddings to the fine tuning classification task resulted in poor performance. The model reached a maximum training accuracy of 6.88% and validation accuracy of 7.64%. Overall, the model received an accuracy of 6%, top-1 accuracy of 6%, and top-5 accuracy of 31%. (**Figure 13**) This results in performance equivalent to chance for each of the classes based on the AUC curve.

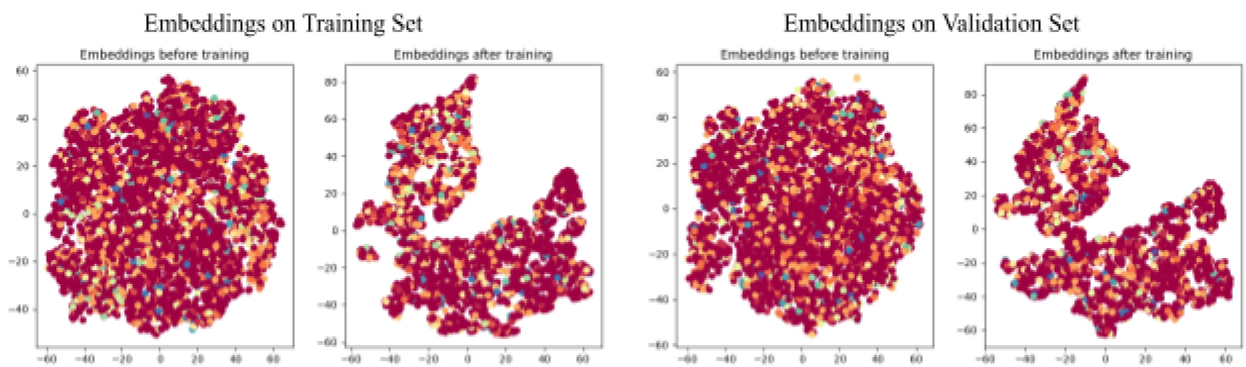

Figure 12: Embeddings from training set (left) and validation set (right) after training h(*) with normal Triplets using EfficientNet.

Summary of Results

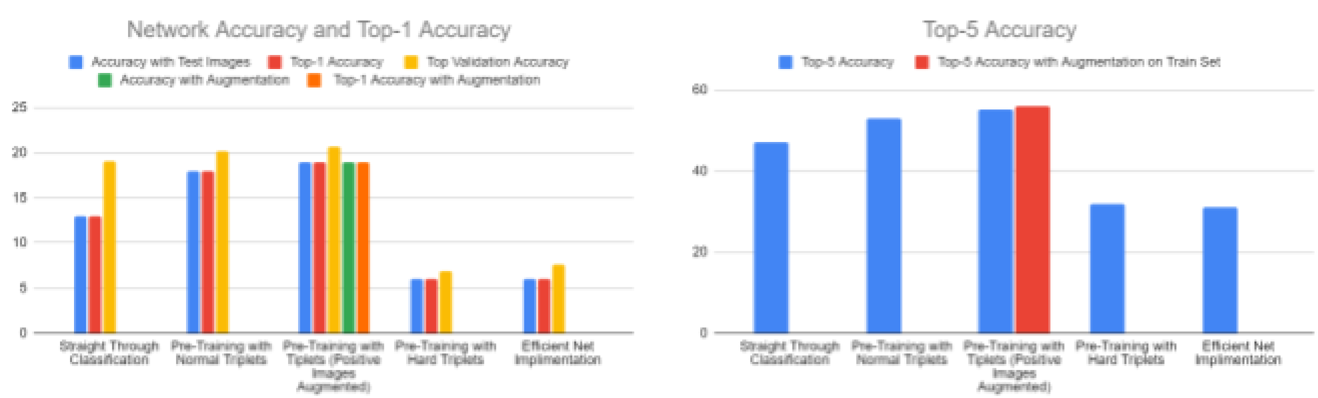

Figure 13: Network Accuracy, top-1 Accuracy, and top-5 Accuracy across all experiments.

The model accuracy, top-1 accuracy, and top-5 accuracy on the test set for each experiment is shown below (**Figure 13**). From the chart, it is evident that augmenting the positive images in the triplets resulted in creating the most meaningful embeddings for the classification task. Furthermore, there is consistent improvement from the normal classification task to pre-training with contrastive triplet loss. However, it is clear that the hard triplets were unable to produce meaningful embeddings to increase classification performance and that the more complex model (EfficentNet) was unable to accurately classify images.

Limitations

Dataset

One of the limitations we encountered at first was the distribution of the classes we were working with. There were significantly more "normal" images compared to other classes. This affected the ability of our model to classify images. Rather than learning the representation of multiple classes, the model would overfit to the normal image and achieve upwards of 60% accuracy by solely classifying each image as normal.

The dataset is also limited by the size of the images. The images from ChestMNIST are small, 28x28. Due to computational resources, the images were only upscaled to 32x32. However, working with small images impacts our models ability to learn as diseases in the chest are sometimes subtle. By having smaller images, our data loses possibly important features for developing accurate classification.

Computational Power

Computation Power limited our ability to perform contrastive pre-training with triplet loss. In order for triplet loss to work best, a combination of large batch sizes, many epochs, and intentional selection of triplets are needed. Triplet loss performs best with batch sizes above 1000 and above 1000 epochs, while creating meaningful triplets by finding hard triplets either online or offline (Schroff, Florian, et al.). While we attempted to find meaningful triplets by using Rishaug's implementation of hard triplets, we were unable to train with the recommended quantity of batch size or epochs due to our limitations with runtime and GPU.

Our computational power limited us to a maximum of 300 epochs (depending on the task) as the runtime became upper bounds of 7 hours and our session would crash shortly after. Despite this fact, we were still able to observe improved performance with contrastive learning.

Conclusion/Discussion

We found that pre-training the data with contrastive learning via triplet loss with positive augmented triplets lead to significant improvements when compared to the full classification task without pre-training. Specifically, when pre-training using positive augmented triplets and increasing the training set during the classification task with augmentation we found a 6% increase in accuracy on the test set (**Figure 13**). Additionally, the overall performance of the model increased and moved away from random level classification of any class. When adding augmentation to the positive image in the triplet, we made the contrastive task harder by increasing the distance between the positive and anchor image that had to be minimized. By increasing the training set with augmented images, we added an element of randomness that helps generalize the training to perform well on the validation set.

The results of pre-training with hard triplets were surprising as they did not create meaningful embeddings and significantly decreased the performance of the classification task, so much so that it was below the performance obtained without pre-training. This could be due to the inability to create substantial batch sizes. Additionally, this could be due to the low resolution images, as many abnormalities in chest-x rays can be subtle and a 32x32 image may not be large enough to capture these (Schroff, Florian, et al.).

Similarly, the results of EfficentNet were interesting because while they appeared to create the most meaningful embeddings of the experiment, the embeddings did not improve the classification task. Some reasoning as to why this occurred is that, similar to the issue with hard embedding, the images did not have enough information. Here it is possible that too many features were trying to be extracted by EfficentNet and while this helped to separate images from one another, it was harmful in classifying the images. EfficentNet B-0 (the model we used for pretrained parameters) is designed to take in images of 224x224, and it is reasonable to assume that when sending in 32x32 images the performance of EfficentNet would decrease (Tan & Le 2019).

The models appeared to have consistent difficulty classifying classes 4 and 7 (infiltration, pneumonia). This could be due to images in classes 4 and 7 being similar to images from different classes. We then assume that the model attributed these features with other classes and grew an affinity towards the other classes.

Overall, we found an improvement in classification performance on the ChestMNIST dataset using contrastive learning pre-training with a triplet loss as seen in previous work (Gupta et al. 2018; Sanz et al. 2021), despite the fact that the embeddings found did not visually appear to have significant structure. This shows that contrastive learning pre-training with a triplet loss, where meaningful embeddings are created, can help fine-tune classification models that perform better when generalizing across individuals and pathologies, thereby progressing the field of computer aided diagnostics.

Bibliography

Elgendi, M., Nasir, M. U., Tang, Q., Smith, D., Grenier, J.-P., Batte, C., Spieler, B., Leslie, W. D., Menon, C., Fletcher, R. R., Howard, N., Ward, R., Parker, W., & Nicolaou, S. (2021). The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective. Frontiers in Medicine, 8, 629134. https://doi.org/10.3389/fmed.2021.629134

Gupta, R., Agrawal, A., Yamada, M., & Srivastava, M. (2018). ChestXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv preprint arXiv:1711.05225.

Madani, A., Moradi, M., Karargyris, A., & Syeda-Mahmood, T. (2018). Chest x-ray generation and data augmentation for cardiovascular abnormality classification. In Medical Imaging 2018: Image Processing (Vol. 10574, p. 105741M). International Society for Optics and Photonics. https://doi.org/10.1117/12.2293971

Maghdid, H. S., Asaad, A. T., Ghafoor, K. Z., Sadiq, A. S., & Khan, M. K. (2021). Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms. Applied Sciences, 11(11), 4910.

Narin, A., Kaya, C., & Pamuk, Z. (2021). Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Analysis and Applications, 24(1), 1207-1220.

Rishaug, Joakim. “Online_triplet_loss.” *Online_triplet_loss | Online_triplet_loss*, 2021, www.jrishaug.com/OnlineMiningTripletLoss/.

Sanz, J.L.C., Sánchez, C.I., López-Martínez, D., Luengo-Oroz, M.A., Antón-Rodríguez, M., & García-Ortega, D. (2021). Multi-view triplet loss network for COVID-19 diagnosis with chest X-ray images. Biomedical Signal Processing and Control, 68, 102667.

Schroff, Florian, et al. “FaceNet: A Unified Embedding for Face Recognition and Clustering.” *arXiv.Org*, 17 June 2015, arxiv.org/abs/1503.03832.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In International Conference on Machine Learning (pp. 6105-6114).

Tschandl, P., Rinner, C., Aponte, J. D., & Kittler, H. (2021). simCLR Improves Chest X-Ray Diagnostic Accuracy Without Labels. arXiv preprint arXiv:2103.14331.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2018). ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3462-3471).

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., & Ni, B. (2023). MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification. Scientific Data. https://doi.org/10.1038/s41597-023-01603-4

This paper and project is a representation of our own work in accordance with the Honor Code and University Regulations

/s/ Jessica Breda
/s/ Claira Fucetola