COVID-19 detection and model optimisation using Vision Transformers

Prithvi S Naidu , Prajwal S Naidu , Mohammed Minhaas BS
psn5377 , pn2140, mb7979
psn5377@nyu.edu, pn2140@nyu.edu, mb7979@nyu.edu

Abstract

COVID-19 is the disease caused by the corona virus called SARS-CoV-2. The initial step in the treatment of COVID-19 is screening of patients at a health facility via a PCR test. However a detailed diagnosis of the patient is done by a medical official by screening the patient's lung x-ray sample through which the progress of the infection can be determined and the level of damage caused to the lung.Medical imaging is simple and fast, thus helping doctors to identify diseases and their effects more quickly. Chest X-rays have been shown in the literature to be a potential source of testing for COVID-19 patients, but manually checking X-ray reports is time-consuming and error-prone. Computed tomography scan (CT scan) and X-ray images are being widely used in the clinic as alternative diagnostic tools for detecting COVID-19 and to find the effects of the virus. We propose a Vision transformer (ViT) model which deep learning pipeline for the detection of COVID-19 from chest X-ray based imaging.Our model successfully differentiates COVID-19, normal and pneumonia patients X-rays with a 61% for train and 58% accuracy in the multi-classification task.

1 Introduction

As of April 2022, there have been 520 million COVID-19 cases worldwide, with new cases rapidly increasing at an alarming rate and showing no signs of abating. The main factor and important break through for stopping the spread of COVID was the early detection and isolation of the patients to prevent its spread. Considering the scale of the pandemic and its rapid spread in a short period of time the burden on the medical facility made it viable to implement deep learning methods using the available data and image recognition algorithms for quick detection of COVID-19.

X-ray imaging has been frequently utilized for COVID-19 screening in comparison to CT imaging as it takes less imaging time, is less expensive, and X-ray scanners are commonly available even in remote regions. Because of the complicated morphological patterns of lung involvement, which can fluctuate in degree and appearance over time, the accuracy of a COVID-19 infection diagnosis using chest imaging is strongly reliant on radio logical proficiency. Considering the scarcity of valid tested radiography lung image we have tried to implement our model on one such unexplored lung radiography dataset.

Our implementation of a Vision Transformer [9] model for image classification demonstrated on the COVID-19 dataset. The ViT model applies the Transformer architecture with self-attention to sequences of image patches, without using convolution layers. We hope to achieve a reasonable level of accuracy with the implementation of the ViT model which is based on transformers and learns by measuring the relationship between input token pairs. In this project we have carried out extensive ablation study as a part of a scope of novelty which is discussed in detail going further.

1.1 Data

The COVID-19 [1] dataset contains 6334 black and white images (Fig. 1) The distribution of images in classes are further shown below (Fig. 2 and Fig. 3). The dataset contains four classes (a) Negative for Pneumonia, (b) Typical Appearance, (c) Intermediate Appearance, (d) Atypical Appearance respectively.

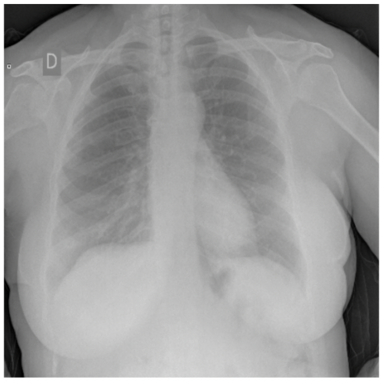


Figure 1: Lung image from the COVID-19 dataset

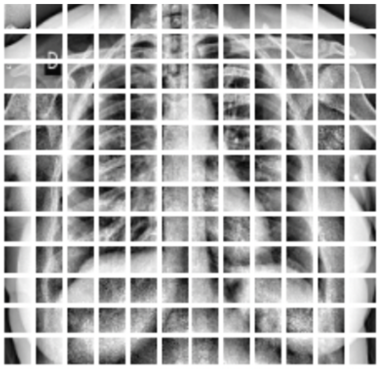


Figure 2: Lung image after patch splitting

2 Model Architecture and Pipeline

2.1 Architecture

After the success of Transformers in solving natural language processing problems [10], Dosovitskiy et al. in [9] presented the Vision Transformer (ViT) model. When trained on enough data, ViT beats state-of-the-art CNN with approximately four times less computing resources. ViT tries to resemble the original transformer architecture [2] as much as possible. We designed a COVID-19 detection pipeline utilizing the Vision Transformer model and fine-tuned it on our dataset with a custom MLP block. The initial part of the network has a Patch Encoder layer which reshapes the input image into multiple flattened patches. Along with the patches, positional embeddings are added to form a sequence, because only sequential data are compatible with the Transformer encoders. The Transformer encoder used contains multi-headed self-attention layers and multiple Multi-layer Perceptron blocks. ViT's self-attention layer enables it to integrate information globally throughout the full picture. Self-attention has a quadratic cost as each pixel in the image is given as input,

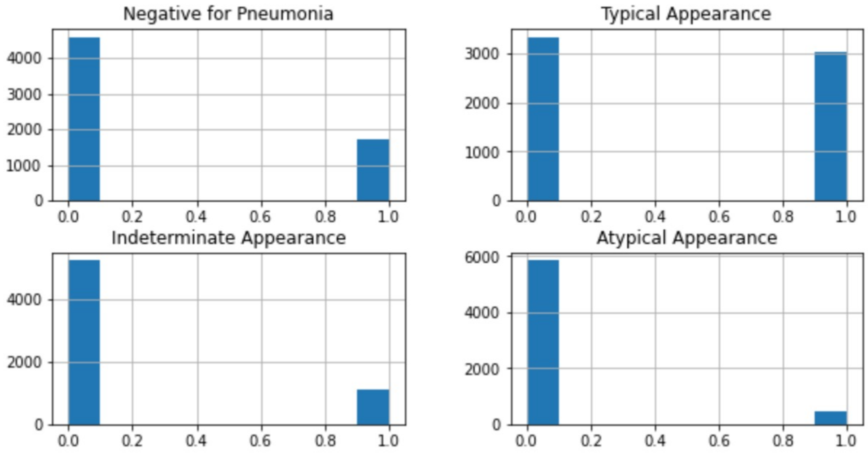


Figure 3: Dataset Class Distribution

self-attention requires each pixel to pay attention to every other pixel. Because the quadratic cost of self-attention is prohibitively expensive and does not scale to a reasonable input size, the image is separated into patches. Because it does not establish any additional dependencies between the training images, Bach Norm is used before each block which assists in reducing training time and improving generalization performance.

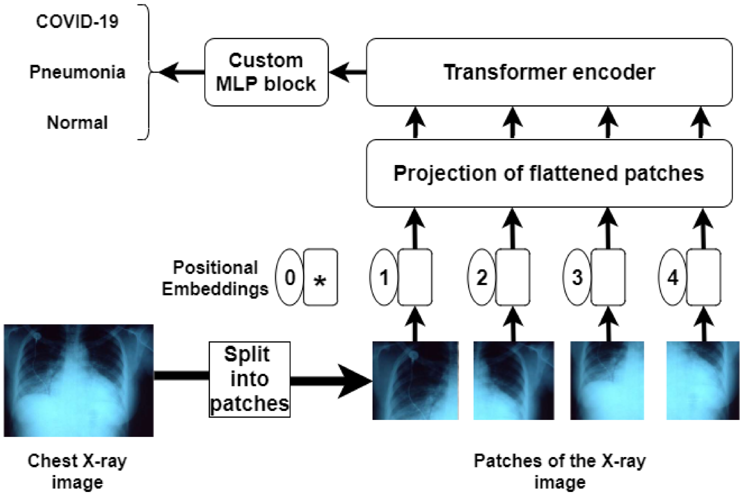


Figure 4: Model architecture

2.2 Fine Tuning

In our implementation we initially started of with ViT-L/16 variant with 16×16 input patch size which had initial pre trained weights from the image net data. We removed the pre trained MLP block and then customized the block and attached an untrained set of feed forward network consisting of MLP block which is shown in fig 4. The flattened result of the last transformer encoder is sent through two arrangements of batch norm and dense layers comprising the MLP block.

Batch norm is a brain network layer that permits the model's different layers to learn all the more autonomously. It is utilized to make the result of the former layers more normal and to make the activation scale the input layer. Learning turns out to be more effective when batch norm is used, and it might likewise be utilized as a regularization to forestall model over-fitting. Although GELU is a very frequently used activation function for the transformer models during our implementation we realized with experimentation RELU activation function performed better with higher accuracy. We also use dropout layer in our MLP block and softmax function in the last layer in order to reduce the effect of over-fitting in our model which helped us achieve a balanced performance noticeable on how our model performs for the train and test set.

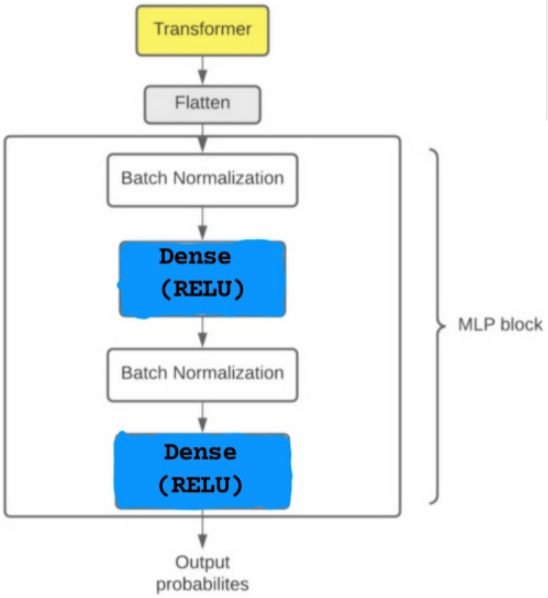


Figure 5: Custom MLP Block

2.3 Model Training Mechanism

In our implementation we use ADAMW optimizer to train our model for multi class classification and categorical Cross entropy loss function for the case of binary classification. We also tried to implement label smoothing by setting the factor to 0.3 in order to have more generalization on the unseen data by adding noise. We experimented with different optimizers such as NovoGrad , ADAM , ADAMW, SGD and based on our results we concluded that ADAMW optimizer's performance for our model gave us a comparably better accuracy than the rest. Our performance metrics during training were as follows:

1) Accuracy: The most well-known performance metric in any classification problem is the accuracy metric. The binary classification included the two fold precision metric which estimates how often the predicted label matches the true label for the chest X-ray images. For the multi-class classification, the downright precision was picked which looks like the average accuracy over every one of the three classes of chest X-ray input images. (figure 6) 2) AUC : That is, AUC measures the entire

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Figure 6: Accuracy calculation

two-dimensional area underneath the entire ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.

3) Loss: Loss value implies how poorly or well a model behaves after each iteration of optimization.

2.4 Preprocessing

Each image in the gathered data set is passed through a minimal image pre-processing pipeline which ensures to make all images compatible for the model training. We followed these steps in the pipeline.

2.4.1 Data Augmentation

Random Crop: A preprocessing layer which randomly crops images during training. During training, this layer will randomly choose a location to crop images down to a target size. The layer will crop all the images in the same batch to the same cropping location. [6] [5]

Normalization : Normalize the activations of the previous layer for each given example in a batch independently, rather than across a batch like Batch Normalization.

Resizing: A preprocessing layer which resizes images. This layer resizes an image input to a target height and width. As neural network models have a fixed-size input layer, all images must be scaled to the same size. Therefore, we resize all the images in the data set to 224 × 224 pixels. [8]

RandomFlip : A preprocessing layer which randomly flips images during training. This layer will flip the images horizontally based on the mode attribute. During inference time, the output will be identical to input.

RandomRotation : A preprocessing layer which randomly rotates images during training. This layer will apply random rotations to each image. By default, random rotations are only applied during training. At inference time, the layer does nothing.

RandomZoom: A preprocessing layer which randomly zooms images during training. This layer will randomly zoom in or out on each axis of an image independently, filling empty space.

The final transformations we used:
- RandomCrop(size=[224,224])
- Normalization()
- RandomRotation(20)
- RandomZoom(20,20)
- Resizing(224,224)
- RandomHorizontalFlip("horizontal")

2.5 Model Evaluation

Our proposed COVID-Transformer was evaluated over the test set of both of our multi-class and binary classification data sets. Although the number of false positives and true negatives is lower, the model sometimes confuses between COVID-19 and other types of pneumonia, which is acceptable as COVID-19 is itself a form of pneumonia and it is very tough even for expert radiologists to distinguish between the two.

The multi-class classification model works well, with an test accuracy of 57.60% and an AUC score of 78.59% and train accuracy 61İn this situation, the accuracy is lower than the AUC score because only images projected as pneumonia but really COVID-19 are misclassified, while all other categories are correctly classified, hence the AUC score is not much affected.

From the above results we notice that out model is very balanced and not overfitting given that the train and test accuracy is very close to each other. We have worked on a dataset which has not been well explored and has a lot of anomalies in it. After performing various preprocessing on our data next, through our ablation study have tried to build a strong model which tries to perform the best for our dataset. The relativeness between our train and test accuracy shows that our model's performance is comparably very good.

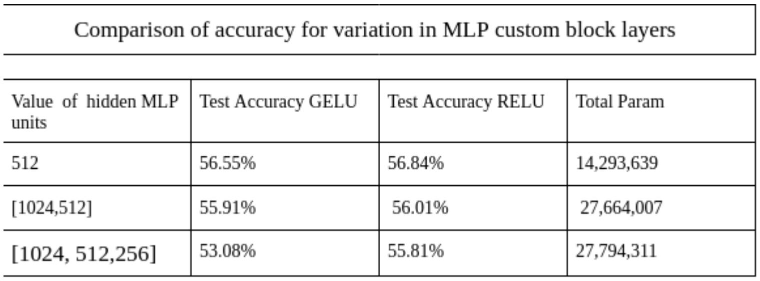
3 Methodology

There are multiple deep learning networks that have been used to diagnose COVID-19, from which CNN is one of the most widely used techniques that has been experimented with the available COVID-19 dataset. We have increased the amount of data by generating new data points from existing data by the means of data augmentation such as to work on a more regularized dataset thereby helping us reduce overfitting when training a machine learning model.. In the project we proposed the usage of the ViT model to learn mapping from the input chest image to the correct class label. The architecture of ViT has been able to achieve state-of-the-art (SOTA) performance in machine translation and other natural language-processing applications. The Transformer architecture is made up of encoder–decoder blocks that allow sequential data to be handled in parallel without the use of any recurrent networks. The self-attention mechanism quantifies the pairwise entity interactions that help a network to learn the hierarchies and alignments present inside input data. The ViT transformer first splits the data into patches of 16x16. These image patches are flattened and lower-dimensional linear embeddings are created from these flattened image patches. The feature map is translated by a tokenizer into a sequence of tokens that are then inputted into the transformer. The transformer then applies the layers.MultiHeadAttention layer as a self-attention mechanism applied to create a sequence of output tokens. Eventually, a projector reconnects the output tokens to the feature map. The Transformer block produces a tensor, which is processed via a classifier head with softmax to produce the final class probabilities output.Then a learn-able embedding to sequence the encoded patches is added for image representation.The outputs of the final Transformer block is then reshaped to using the flatten() and passed as an image representation input for the ViT classifier. The image are classified with and accuracy of 57.72% . The area under curve (AUC) for the testing dataset was 78.79%.

4 Ablation Study

In order to ensure that our transfer learning architecture is optimal, we conduct a comprehensive ablation study on the multi-class classification data set. With reference to the appendix A in [9] we were inspired to experiment with various parameters mentioned and bring about the best tuned parameter settings for our model in this project.

In the first part of our study we have focused on experimenting with modifying the custom block using different of combinations MLP head units and its effect with different activation function namely RELU and GELU [4]. From the results as obtained in (Figure 6) we observe a higher accuracy is obtained by the usage of a single combination dense layer unit and dropout layer in the MLP block with Relu as the activation function.This shows that ReLU activation is slightly more effective in processing the outputs of the stacked transformer encoders compared to the GeLU activation for our model and data.

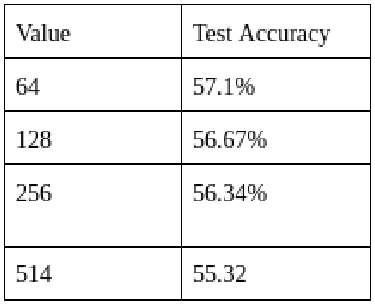


Comparison of accuracy for variation in MLP custom block layers

Value of hidden MLP units	Test Accuracy GELU	Test Accuracy RELU	Total Param
512	56.55%	56.84%	14,293,639
[1024,512]	55.91%	56.01%	27,664,007
[1024, 512,256]	53.08%	55.81%	27,794,311

Figure 7: Comparison of accuracy for variation in MLP custom block layers

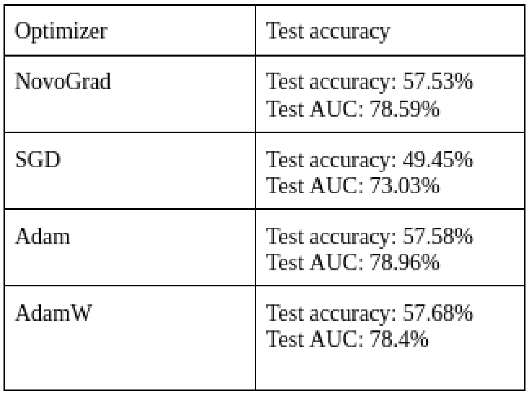
Next, we further experiment with the trade off between different batch sizes and accuracy as shown in (Figure 7).We observe clearly that the accuracy obtained declined as the batch size increased. Even though greater batch sizes means a higher computation time in general, since we are using GPU's, they utilize parallel computing which weakens the effect of increased batch size on the computation time. We are able to increase it and get less noisy gradient estimations. Although we can go for higher batch sizes, we will still be in the small batch size regime because they observe that larger batch causes a degradation in the quality of the model's ability to generalize. So we have chosen to use batch size of 64 mainly because it capitalized on GPU's parallelization better hence was faster to train.



Value	Test Accuracy
64	57.1%
128	56.67%
256	56.34%
514	55.32

Figure 8: Variation of Accuracy with Batch Size

Next,we further experiment with the trade off between different Optimizer and Accuracy as shown in (Figure 8). For the optimizer we have chosen to try SGD and ADAM. We have tried them at the beginning of our tests with the model from starter code. Figure 2 shows the performance of NovoGrad and SGD, Adam and AdamW optimizers. We can clearly see that for our model with AdamW is a better choice since AdamW yields better training loss and that the models generalize much better than models trained with Adam allowing the new version to compete with SGD with momentum. We have have chosen the stick with default beta values.

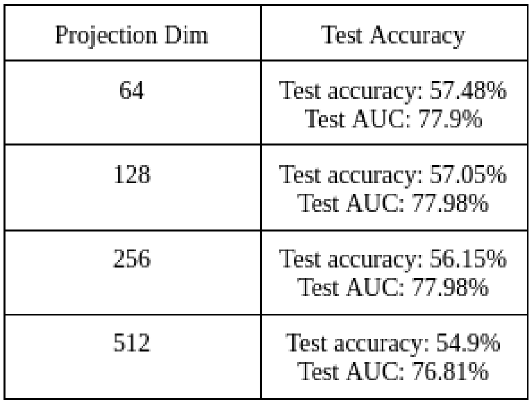


Optimizer	Test accuracy
NovoGrad	Test accuracy: 57.53% Test AUC: 78.59%
SGD	Test accuracy: 49.45% Test AUC: 73.03%
Adam	Test accuracy: 57.58% Test AUC: 78.96%
AdamW	Test accuracy: 57.68% Test AUC: 78.4%

Figure 9: Variation of Accuracy with Activation Functions

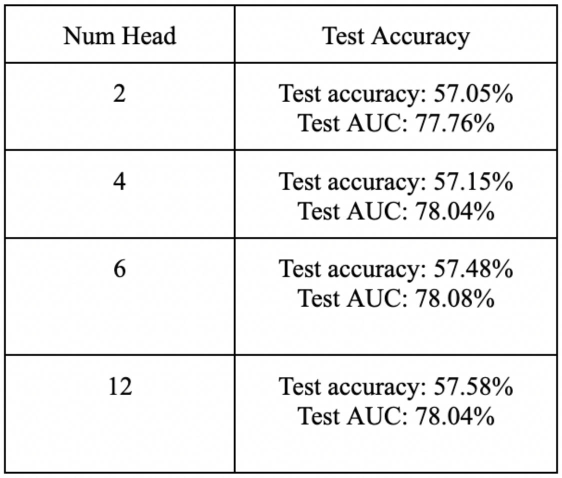
Further on,we discuss about the effect of projection dimensions in our MLP block on accuracy of our model. In our model the projection dimension parameter is used to increase the number of transformer units in the ViT model. We have experimented with multiples of 2 started from 64 and observed the test accuracy and AUC with increase in the Projection dimensions. It is clearly observed that our test accuracy was the best for projection dim of 64 which also helps us understand that merely increasing the number of transformer units does not increase the accuracy of the model. The results are dispalyed in (figure 9).

Changing the number of heads changes the number of learnable parameters. If you have more heads, training will take longer. When we have several heads per layer the heads are independent of each other. This means that the model can learn different patterns with each head. We experimented with multiples of 2 started from 2 and observed the test accuracy and AUC with increase with increase in num heads. (figure 11)



Projection Dim	Test Accuracy
64	Test accuracy: 57.48% Test AUC: 77.9%
128	Test accuracy: 57.05% Test AUC: 77.98%
256	Test accuracy: 56.15% Test AUC: 77.98%
512	Test accuracy: 54.9% Test AUC: 76.81%

Figure 10: Variation of Accuracy with Projection Dimensions



Num Head	Test Accuracy
2	Test accuracy: 57.05% Test AUC: 77.76%
4	Test accuracy: 57.15% Test AUC: 78.04%
6	Test accuracy: 57.48% Test AUC: 78.08%
12	Test accuracy: 57.58% Test AUC: 78.04%

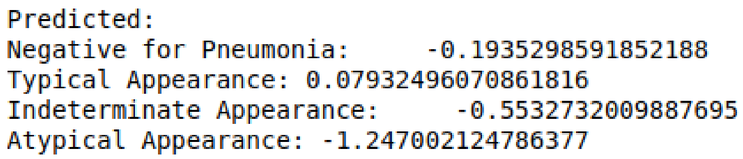
Figure 11: Variation of Number of Heads and Projection Dim of Multihead Attention Layer(QKV)

5 Results

In this exploration, we proposed a strong and interpretable Vision Transformer model that can efficiently analyze COVID-19 in real-circumstances for medical services. The model architecture chosen was based on the Vision Transformer and it showed high performance with accuracy and AUC score of 58.6% and 78.79%, respectively.We have identified and tested different combinations of ViT architecture specifc hyperparameters, general hyperparameters, and transformations. For making our model trustworthy, we made an interpretable inference pipeline with extensive ablation study as shown above.

Though this model achieves comparably good results in classifying COVID-19 lung radiography images, there is still scope for development. Given that noise plays a key factor in radiography that affects the model's performance we could apply Generative Adversarial Network [7] based noise reduction [9] techniques on the dataset can greatly improve the performance of our model. Using a large version of ViT [9] with a larger dataset can surely reach higher performance metrics which was one of the main reasons we achieved lower accuracy rates in our project.Our future work will focus on proposing another variant of the Vision Transformer for further improving the performance, given the availability of larger and more robust data sets.

We present the following 3 curves (Fig 11, 12 and 13) in support of our results.



```
Predicted:
Negative for Pneumonia:      -0.1935298591852188
Typical Appearance: 0.07932496070861816
Indeterminate Appearance:      -0.5532732009887695
Atypical Appearance: -1.247002124786377
```

Figure 12: Final Prediction

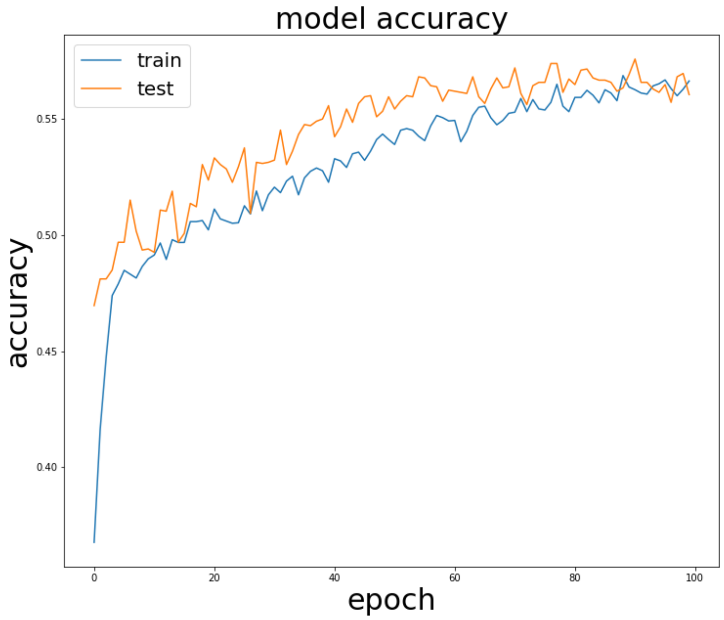


Figure 13: Accuracy vs Number of epochs

6 Code and References

You can find the entire code of our project here [3].

References

[1] *Annotation and Standard Exam Classification of COVID-19 Chest Radiographs*. https://osf.io/532ek/. Accessed: 2022-03-24.

[2] *Attention Is All You Need*. https://arxiv.org/abs/1706.03762/. Accessed: 2022-03-24.

[3] *Code and steps to run*. https://github.com/Minhaas/COVID19_ViT. Accessed: 2022-03-24.

[4] *COVID-Transformer: Interpretable COVID-19 Detection Using Vision Transformer for Healthcare*. https://www.mdpi.com/1660-4601/18/21/11086/htm#B47-ijerph-18-11086/. Accessed: 2022-03-24.

[5] *Deep Learning for Multigrade Brain Tumor Classification in Smart Healthcare Systems: A Prospective Survey*. https://pubmed.ncbi.nlm.nih.gov/32603291/. Accessed: 2022-03-24.

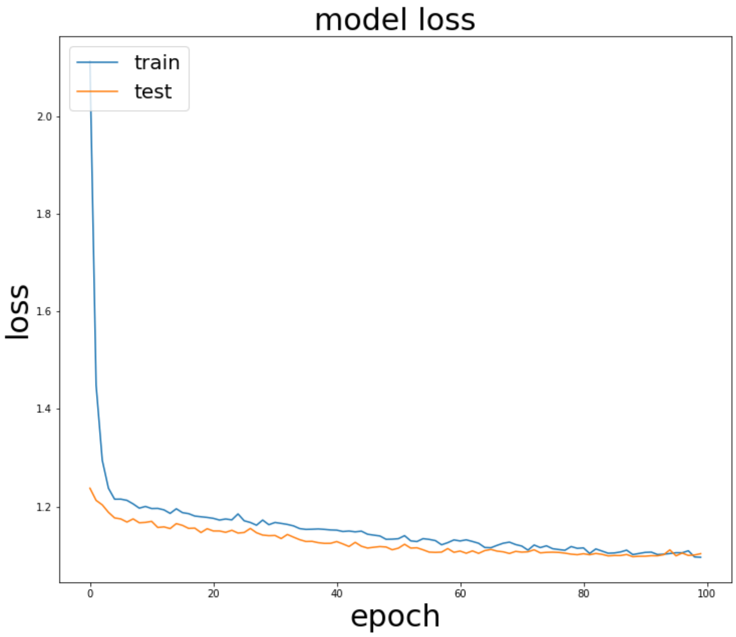


Figure 14: Loss vs Number of epochs

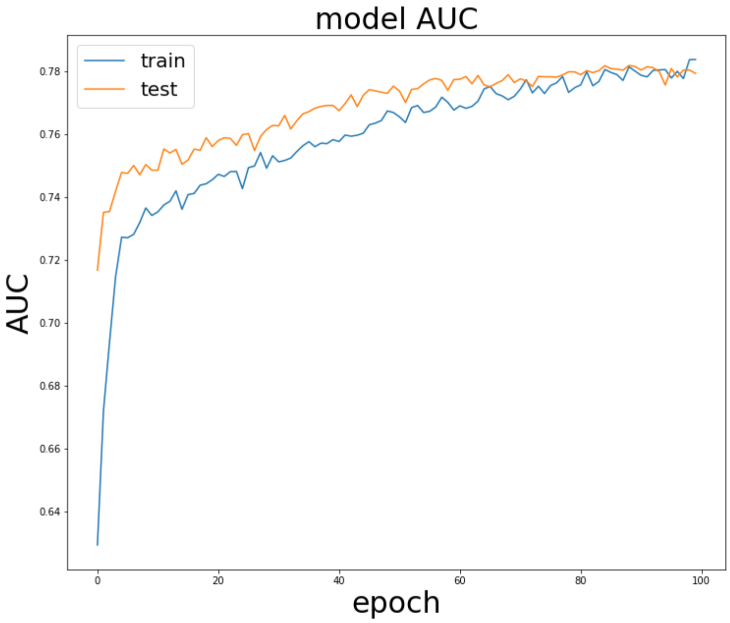


Figure 15: AUC vs Number of epochs

[6] *Differential Data Augmentation Techniques for Medical Imaging Classification Tasks*. `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977656/`. Accessed: 2022-03-24.

[7] *Generative Adversarial Networks for Hyperspectral Image Classification*. `https://ieeexplore.ieee.org/abstract/document/8307247/`. Accessed: 2022-03-24.

[8] *ImageNet Classification with Deep Convolutional Neural Networks*. `https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html/`. Accessed: 2022-03-24.

[9] Alexander Kolesnikov et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: (2021).

[10] *Pre-trained Language Models in Biomedical Domain: A Systematic Survey*. https://arxiv.org/abs/2110.05006/. Accessed: 2022-03-24.