A REPORT

ON

VISION TRANSFORMERS - LEVERAGING THE TRANSFORMER ARCHITECTURE FOR IMPROVED PNEUMONIA DIAGNOSIS THROUGH CHEST X-RAY

BY

BHASKAR RUTHVIK BIKKINA	2021A7PS1345H
SIDDHANT SRINIVAS	2021A7PS0050H
NIKHIL DHANARAJ	2021A7PS0427H
SOUMIL RAY	2021A7PS2652H
SHIVAM ATUL TRIVEDI	2021A7PS1512H
ADARSH DAS	2021A7PS1511H

CS F425

DEEP LEARNING

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Nov 2023

Contents

Introduction and Problem Statement

In the world of computer vision, Convolution architectures such as LeNet and AlexNet have remained dominant. Transformers, with their high computational efficiency and scalability are primarily used for Natural Language Processing (NLP) tasks. As such, the Vision Transformer Architecture (ViT) is an attempt to adapt the original transformer architecture to computer vision problems. This model was proposed by Dosovitskiy, A. *et al.* (2020) *An Image is worth 16x16 words: Using Vision Transformers for Image Recognition at scale*. This is a multi-headed self-attention based architecture. A Transformer architecture is generally considered to be any neural network that uses the attention mechanism) as its primary learning layer. We apply a transformer to images by splitting an image into smaller 'patches' and passing the sequence of linear embeddings of these patches as an input to the Transformer, likening these 'patches' to words in an NLP application.

Problem Statement: As the advancements in the world of Deep Learning and Machine Learning continue, we find that their use in various industries is rising rapidly. One such industry that would be helped a lot by this AI revolution is the field of Medicine. Diagnosing diseases through a simple picture without having to consult with the doctor is a convenience that is now becoming a reality. As such we choose the task of diagnosing whether a patient has pneumonia or not through a Chest X-Ray scan. As we have seen during the Covid-19 pandemic, the need for being able to diagnose diseases remotely is now more imminent than ever. We see the same study where researchers have tried to diagnose covid using these Chest X-Ray scans in (Ibrahim et al., 2021). As such we need models that perform more accurately while also being portable enough that they can run without issues on low-powered hardware

Methodology

We first implement the model (ViT-Base) proposed by Dosovitskiy, A. *et al.* (2020) *An Image is worth 16x16 words: Using Vision Transformers for Image Recognition at scale*. from scratch using the architecture presented in the paper. Next, we test out the results on the Cifar-10 dataset without any training and after training for a few epochs. We then use various techniques such as pretraining and fine-tuning to improve the results of the model without requiring much GPU compute. These results are then compared without other pretrained models fine-tuned for the same number of epochs to see if the architecture produces better results. We then try to reduce the model size while maintaining similar accuracy for faster inference and better portability by tinkering with the encoder block. We also perform various experiments to see the impact of small changes in the patch embeddings and the linear layers to see how they would impact the size and accuracy of our model and if they are feasible changes. Note that due to the nature of the dataset, the data transforms used in the ViT architecture(Dosovitskiy, A. *et al.*, 2020) did not have any impact on the accuracy and as such to maintain uniformity among all the models being tested the images were merely resized to a size of 224x224 and converted to tensors.

Results of Paper Implementation

We implement the architecture using the PyTorch library and define a class with adjustable patch size, stride length, number of encoder layers etc. to allow for experimentation on the model. We compare this model with the ViT-b16 model available in the datasets module of the Torchvision library and use the ImageNet-1K weights to initialize the model. This provides us with a similar model which has already been pretrained on a large dataset that can then be fine-tuned for our use case scenario. As in the paper (Dosovitskiy, A. *et al.*, 2020), we use the Adam optimizer to train the model with a learning rate of 0.003 and learning rate decay parameter of 0.3. The betas used are (0.9 and 0.999) for the optimizer. We use the Categorical Cross Entropy Loss Function as our loss function and train each model for a max of 10 epochs.

While Dosovitskiy, A. *et al.* (2020) was tested on various benchmark image classification datasets, due to GPU constraints we were only able to reliably test the models on the Cifar-10 dataset. The results obtained are shown below:

Table 1 - Accuracies of models on Cifar-10

Model name	Number of Epochs	Accuracy (%)
ViT-B16 (Vanilla)	10	10.01
ViT-16 (P)	-	13.02
ViT-16 (P)	1	85.95
ViT-16 (P)	10	87.01

As seen, the model with no pre-training when trained on a few epochs performs extremely poorly with an accuracy of 10% which would be the exact same as the probability of a random guess being the correctly predicted class. This is certainly due to the fact that the weights have been initialized to random values and as the model contains over 85 million parameters it needs to be trained for an exponentially greater number of epochs.

The pretrained model with no fine-tuning surprisingly also performed very poorly with an accuracy of 13% but this may be attributed to the fact that the linear layer of the model was changed as the ImageNet-1K dataset contains 1000 classes but the Cifar-10 dataset contains only 10 classes.

Upon freezing the embedding and encoder layers and fine tuning the model by updating the linear layer weights we immediately see a huge spike in the accuracy which increases by almost 650%. Continuing this process for 9 more epochs we can see that the model reaches a very respectable accuracy of 87.01 which is expected. After implementing the model and checking its results we now apply it to solve our problem.

We take the Chest X-Ray dataset from Kaggle. We now compare the ViT model (Dosovitskiy, A. *et al.*, 2020) with 2 other state of the art models - Resnet-50 (He et al., 2016) and ConvNeXt-B (Liu et al., 2022).

We fine-tune each model for 10-epochs and ensure that every hyperparameter is exactly the same including the dataset on which the pre-training weights are chosen. Upon performing fine-tuning on our Chest X-Ray dataset we obtain the following results. The table below shows the average accuracy per epoch of fine-tuning for each of the models and the max accuracy achieved and the standard deviation as each model reaches its maximum accuracy value at different epochs:

Table 2 - Comparison with State of the Art models

Model name	Avg. Accuracy (%)	Max Accuracy (%)
Resnet-50	75.17 ± 3.34	80.47
ConvNeXt-B	76.50 ± 2.44	79.84
ViT-B16	**77.73 ± 3.69**	**83.13**

We can see that the ViT (Dosovitskiy, A. *et al.*, 2020) performs the best out of all the 3 models. This shows that using the power of transformers does provide an appreciable difference in accuracy as compared to other state of the art models. We now try to improve the best model by performing a few tweaks in the architecture and leverage the fact that the dataset size is much smaller than the number of parameters which will help us safely reduce the number of parameters without negatively affecting the accuracy.

Proposed Improvements

1. As we can clearly see that pre-training improves accuracy drastically, we choose to use a pretrained model which we will further improve by modifying the architecture
2. We change the patch size of the embeddings from 16x16 which causes the kernel and stride sizes in the conv layer to be 16x16 and change it to 32x32. We now get more general information in each embedding rather than localized data in an image.
3. The most important change being made to the model is on the encoder block used on the vision transformer. We remove each of the odd layers of the encoder block except the first block and use the last encoder block recursively to refine the embedding that would otherwise directly be passed into the linear layer. Now, immediately this reduces the number of parameters in the model drastically which reduces the time of inference of the model which is crucial when the model is run on CPUs and also increases the portability of the model which would help it run on mobile devices. We check how this impacts the accuracy and if similar accuracy is obtained this could be considered an improved model as it provides the same accuracy but with a much smaller size.

Results of Proposed Improvements

We first find the improvements in the model caused by changing the patch size from 16 to 32. This increases the parameters by 2 million which is a small increase compared to the total number of parameters in the model. It maintains almost the same training times per epoch and the same inference time but performs better when it comes to accuracy and also obtains a higher peak accuracy as compared to all the other models that this was tested on.

We then reduce the size of the encoder block by removing the odd layers in the block except for the first layer. This is because we felt that the first layer is already pre-trained to handle the patch embeddings it receives and as such removing it could result in drastic changes since the encoder weights will be frozen during the fine-tuning process.

We decided to go for this as this seems the safest way to reduce the number of parameters without losing out on the advantages obtained from pretraining. We then add the last encoder layer once again into the encoder block to refine the encodings obtained before it is passed onto the linear layer for classification.

We notice that these changes immediately reduce the parameters from 87 million to 52 million which is a 40% reduction in the number of parameters. This makes it easier to fine-tune parts of the encoder block on the dataset if that is required and also achieves our goal of obtaining a smaller model with faster inference speed while maintaining similar accuracy.

The table below contains all the results from the changes individually and combined together:

Table 3 - Improved Model Accuracies compared

Model name	Avg. Accuracy (%)	Max Accuracy (%)
Resnet-50	75.17 ± 3.34	80.47
ConvNeXt-B	76.50 ± 2.44	79.84
ViT-B16	77.73 ± 3.69	83.13
ViT-B32 (52M)	**77.88 ± 4.88**	**82.66**
ViT-B32 (87M)	**78.01 ± 3.01**	**83.59**

It is evident that the results shown above have only minor differences in accuracy. This is due to the size of the fine-tuning dataset which contains only about 5200 images mixed with the small number of epochs that the model was trained on.

Experiments

The following experiments were performed to try and improve the performance of our Vision Transformer :-

Experiment	Observed changes	Possible reasons why
Changing Patch Size from 16 to 32	Positive impact on accuracy	A larger patch size makes it easier for the model to learn features that are not localized in smaller patches
Retaining only first encoder block	Drastic reduction in testing accuracy	Due to the loss of numerous important weights that would provide a better encoding for the linear layer to classify
Changing kernel size to 16 on patch 32 model and adding max pool	Drastic reduction in testing accuracy	Gets more localized information and loses more data due to max pooling
Changing kernel to 16 on patch 32 and adding max pool and another conv2d layer	Drastic reduction in testing accuracy	Multiple convolutions distort the patch embeddings which are then not very distinguishable by the encoder
Introducing more layers into the feed forward network	Moderately reduced accuracy of architecture	Vanishing/exploding gradient may make it difficult to for the weights to be updated optimally
Increasing dropout rate to reduce overfitting	Reduced overfitting but also reduced overall accuracy of model	Reducing the number of neurons makes it harder for the network to learn the optimum weights
Using the encoder block twice by stacking each encoder layer over itself	Very slight dip in accuracy	Since the parameters are frozen during fine-tuning, the model finds it hard to adjust to the exponential increase in parameters

Findings and Accomplishments

Our findings from the project include :-

1. Increasing the number of parameters often negatively impacted the accuracy of the model. This could be due to the large size of the fine-tuning dataset.
2. We also noticed that the introduction of pooling into the convolution layers distorted the positional embeddings which led to poor performance.
3. In most cases, the size of the dataset determined the model's training time rather than the total number of parameters.
4. The experiments that were performed on the implemented model without pretraining led to no changes in accuracy and very slight changes in training loss since the model was only trained for a small number of epochs and the weights were still mostly random.
5. Every model took around 90 seconds on average to fine-tune during each epoch.
6. We also noticed that the training losses were almost always lower than the testing losses which suggests overfitting but this is mostly due to the imbalance in the number of samples for training and testing in the dataset.

Our Accomplishments from the project:

1. This project establishes that making use of the transformer architecture to perform pneumonia detection using Chest X-Ray images is more accurate than some of the state of the art models that use convolution based methods.
2. We realize the benefits of pretraining and the benefits of freezing certain parameters and only training some weights,
3. The model we propose in this report provides a much smaller model that performs at similar accuracy to or better than most of the other models that are already present.
4. We present a model that also speeds up the time of inference thus making the diagnosis process faster on low-powered hardware such as a mobile phone.

Contributions by Group Members

Name of Group Member	Contributions Made
BHASKAR RUTHVIK BIKKINA	Testing models such as BiseNet V1 and V2,finding datasets to fine-tune on,implementing ViT from scratch and comparative study on cifar-10
SIDDHANT SRINIVAS	Testing models such as LETNet, implementing ViT from scratch and drafting the report.
NIKHIL DHANARAJ	Fine-tuning ViT, implementing proposed improvements and experimenting with kernel size.
SOUMIL RAY	Working on inference times and experimenting with Encoder architecture and dropout.
SHIVAM ATUL TRIVEDI	Working on improvements, making observations and drafting the report
ADARSH DAS	Finding datasets to fine-tune on, comparative study of fine-tuned models and conclusions

References

[1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

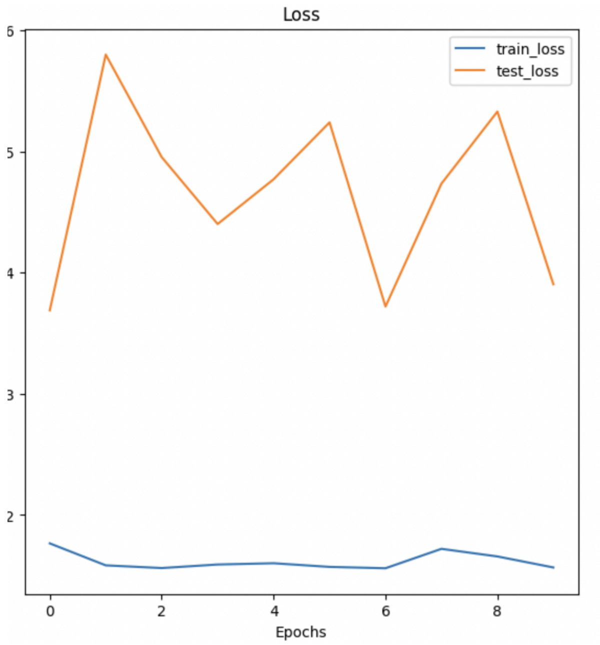
[3] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11976-11986).

[4] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*, 211-252.

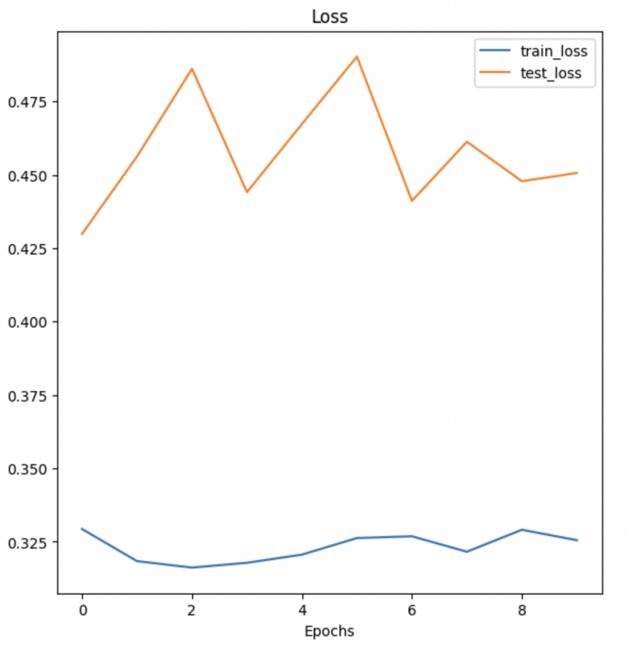
[5] Ibrahim, A. U., Ozsoz, M., Serte, S., Al-Turjman, F., & Yakoi, P. S. (2021). Pneumonia classification using deep learning from chest X-ray images during COVID-19. *Cognitive Computation*, 1-13.

Appendix A - Plots of Loss Curves

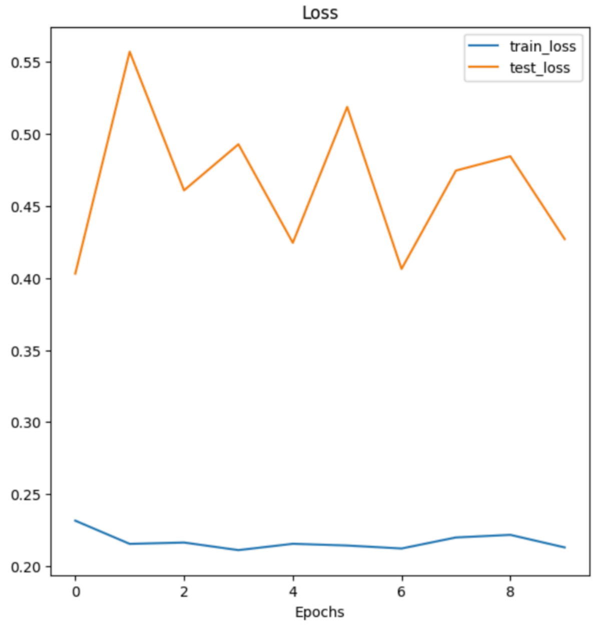
Pretrained-ViT



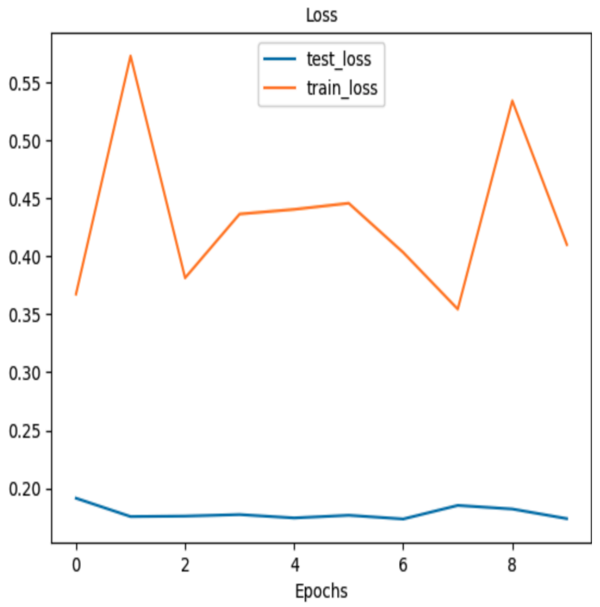
Resnet 50



ConvNeXT

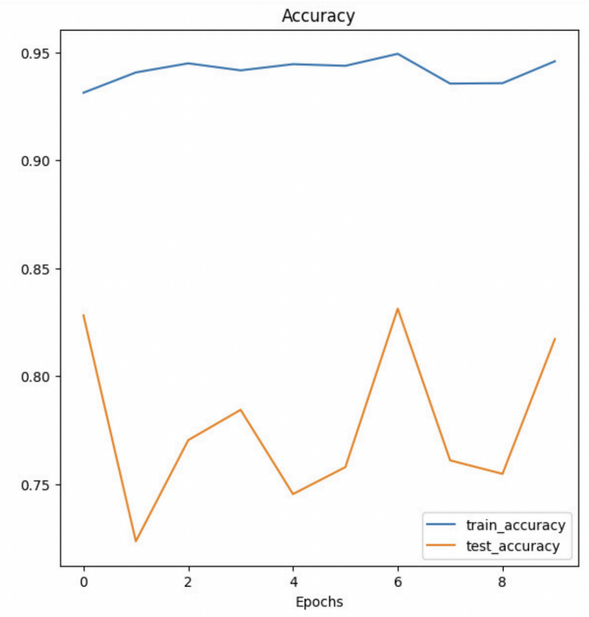


Modified ViT

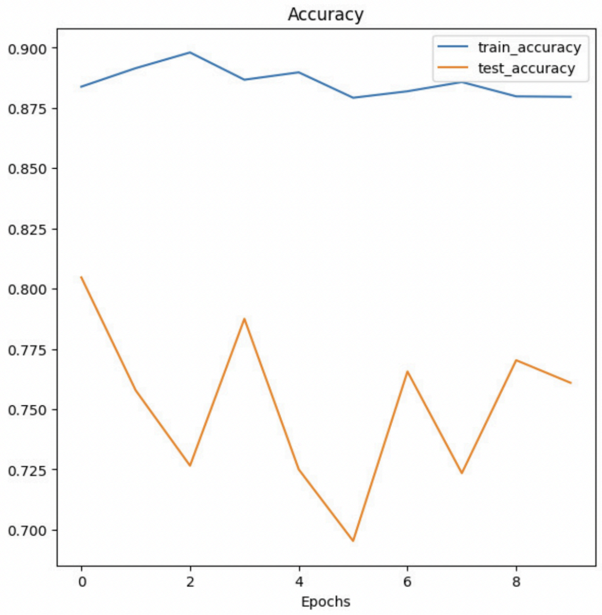


Appendix B - Plots of Accuracy Curves

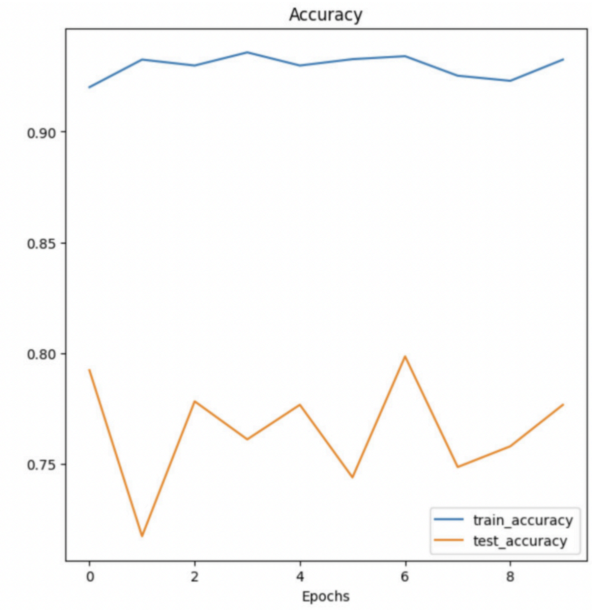
Pretrained-VIT



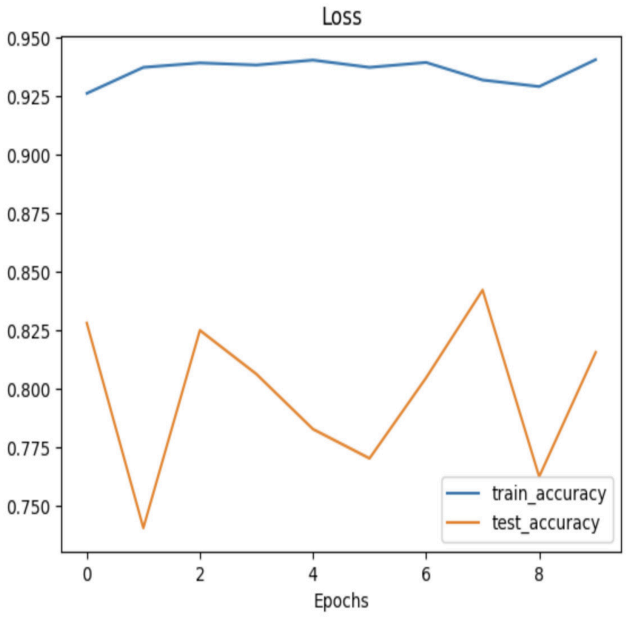
Resnet 50



ConvNeXT



Modified ViT


Loss
Epochs
train_accuracy
test_accuracy