Vision Transformers on Chest X-Ray Dataset

Aishwarya Raman
Department of Computer Science
NYU Courant
ar6381@nyu.edu

Nidhi Ranjan
Department of Computer Science
NYU Courant
nr2387@nyu.edu

Shreya Gupta
Department of Computer Science
NYU Courant
sg6606@nyu.edu

Abstract

The use of computer vision and deep learning based tools in healthcare and medical imaging has become increasingly frequent in the last couple of years. However, in order to implement a computer-aided diagnosis system in the professional healthcare industry for disease prediction and diagnosis, it is crucial that the system is reliable. Extracting and labeling medical images is strenuous task. Chest X-ray images are one of the most commonly accessible radiological examinations and are used for screening and diagnosis of many lung diseases. By leveraging a sample of the Chest Xray dataset, we are interested in exploring the performance of vision transformers for disease classification. Moreover, we want to explore how models pre-trained on generic, non-healthcare data perform when fine-tuned with highly specialized, yet small datasets like the chest X-Ray images that we use. In our experiments, we analyze Vision Transformers with unsupervised pre-training, supervised pre-training and self-supervised pre-training and compare it with our baseline VGG16 model. We find that the vision transformers with self-supervised pre-training show promising results.

Source code for our experiments can be found at (1)

Keywords
vision transformers, cnn, chest xray, self supervised vision transformer, resnet, vgg, medical imaging, classification

1 Introduction

1.1 Medical imaging and diagnosis

Medical imaging deals with the interaction of all forms of radiation with tissue and the design of technical systems to extract clinically relevant information, which is then represented in image format. Medical images range from the simplest such as a chest X-ray to sophisticated images displaying temporal phenomena such as the functional magnetic resonance imaging (fMRI). (2)

Deep learning and Computer Vision have been widely used in research related to medical imaging for computer-aided diagnosis, radiomics, and medical image analysis. Various architectures of Convolutional Neural Networks like 3D CNNs(3), SecRCNN(4), Multi-Receptive-Field CNNs(5) have been explored extensively. Other techniques and models like like Massive training artificial neural networks(6), synergic deep learning models (7), Opposition-based Crow Search have also been explored. (8) surveys multiple deep learning techniques for medical imaging classification. Self-supervised learning (9), semi supervised learning (10), unsupervised learning (11) and of course supervised learning techniques have employed for medical image segmentation, classification and clustering.

1.2 Vision Transformers

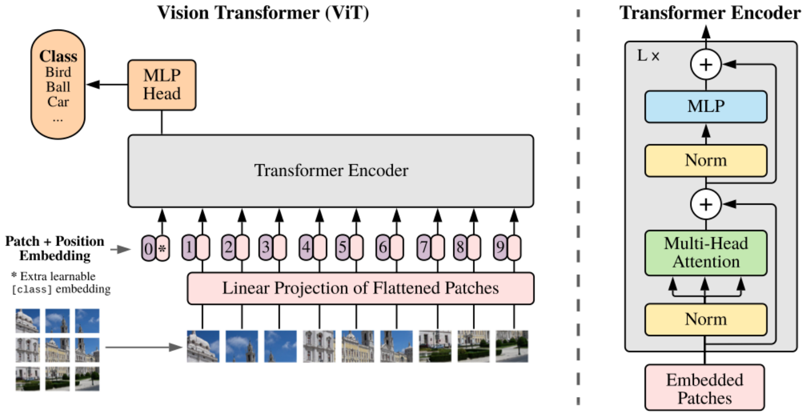
Transformers measure the relationships between pairs of input tokens (words in the case of text strings), termed attention. The cost is exponential with the number of tokens. For images, the basic unit of analysis is the pixel. However, computing relationships for every pixel pair in a typical image is prohibitive in terms of memory and computation. Instead, Vision Transformers (ViT) compute relationships among pixels in various small sections of the image at a drastically reduced cost. These transformers when trained on mid-sized datasets such as ImageNet have a slightly lower performance as compared to state-of-the-art CNN architectures like ResNet. However, when trained on large datasets (14M-300M images) like public ImageNet-21k dataset or the in-house JFT-300M dataset, ViT approaches or beats state of the art on multiple image recognition benchmarks. In particular, the best model reaches the accuracy of 88.55% on ImageNet, 90.72% on ImageNet-ReaL, 94.55% on CIFAR-100, and 77.63% on the VTAB suite of 19 tasks.(12) Further adapting self-supervised pre-training methods to this architecture has been shown to work well .(13). Additionally, self-supervised ViT features contain explicit information about the semantic segmentation of an image, which does not emerge as clearly with supervised ViTs, nor with convnets. Second, these features are also excellent k-NN classifiers achieving a performance of 78.3% top-1 on ImageNet. (13)

1.3 CNN

CNNs learn features from the underlying data. These features are data driven and learnt in an end to end learning mechanism. The strength of CNN is that the error signal obtained by the loss function is used/propagated back to improve the feature (the CNN filters learnt in the initial layers) extraction part and hence, CNN results in better representation. The other advantage is that in the initial layers a CNN captures edges, blobs and local structure, whereas the neurons in the higher layers focus more on different parts of human organs and some of the neurons in the final layers can consider whole organs. (14)

2 Models

2.1 Vision Transformer with no pre-training

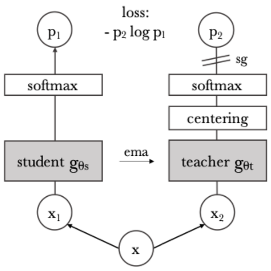


Vision Transformers work by splitting images into patches, and flattening them. It then produces lower-dimensional linear embeddings from the flattened patches. To these patches, positional embeddings are added.We feed this sequence as an input to a standard transformer encoder.The transformer encoder layer is made up of self-attention and feedforward network.We train this model with our dataset in a fully supervised manner.

2.2 Vision Transformer pre-trained on ImageNet by using supervised learning

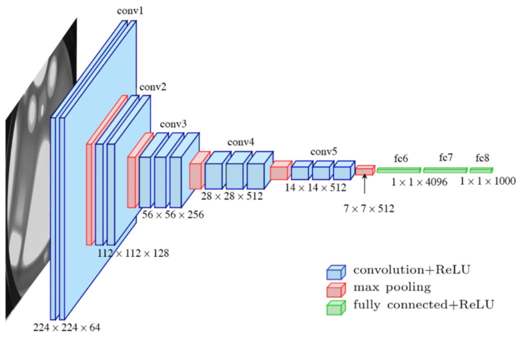
In this model, we use a vision transformer model which was trained on ImageNet, in a supervised fashion. We then fine-tune to use it for our classification tasks. For this, we remove the pre-trained prediction head and attach a zero-initialized D × K feedforward layer, where K is the number of classes to be predicted.

2.3 Self-Supervised Vision Transformers



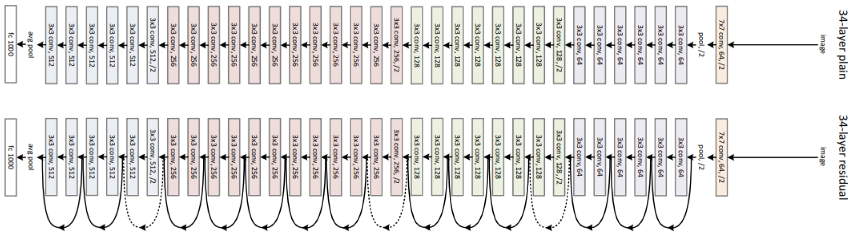
In images, image level supervision often reduces the rich visual information contained in an image to a single concept selected from a predefined set of a few thousand categories of objects.Self-supervised ViT features explicitly contain the scene layout and, in particular, object boundaries. This information is directly accessible in the self-attention modules of the last block. (13) This transformer model uses two different networks - student and teacher. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. (13)

2.4 VGG16



The VGG16 model takes in an input image of size 224×224(×3 color channels), and applies a convolution of size 3×3 (with 64 kernels/output channels) with a stride of 1 and a padding of 1. The output size of the convolution is maintained and then halved by a 2×2 MaxPool layer, whose output size is then maintained until another 2×2 Maxpool layer is applied. The number of channels increases till the input gets to the fully connected layer and then the Softmax layer reveals the class the input belongs to. We use a pretrained version of VGG16, and use the softmax layer to help predict classes

2.5 ResNet34



This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into residual network. The convolutional layers mostly have 3×3 filters and for the same output feature map, the layers have the same number of filters or if the size of the features map is halved, the number of filters is doubled to preserve the time complexity of each layer. It is worth noticing that the ResNet model has fewer filters and lower complexity than VGG Nets. We use a pretrained version of ResNet34 which was trained on ImageNet, we add a fully connected layer, and use the Cross Entropy Loss function for our class prediction.

3 Dataset

3.1 Overview

The complete dataset, ChestXray, is comprised of 112,120 X-ray images with disease labels from 30,805 unique patients. To create these labels, the authors used Natural Language Processing to text-mine disease classifications from the associated radiological reports. The labels are expected to be >90% accurate and suitable for weakly-supervised learning. An image can have multiple labels associated with it. However, due to large image sizes and hardware and time limitations, we used a randomized sample of this dataset. We used of a total of 6978 image-label pairs, 6000 for training, 500 for validation and 478 for testing.

3.2 Preprocessing

We segregated data on the basis of their labels. For our experiments, we resized the images into sizes of 224 x 224. For our experiments to we also set the contrast to 5, and performed normalisation thereafter.

3.3 Class labels and their distribution

This section shows the number of images in each class:

Type	Sample Size
Atelectasis	509
Consolidation	227
Effusion	645
Fibrosis	85
Infiltration	968
Nodule	314
Plueral Thinking	177
Pneumothorax	272
Cardiomegaly	142
Edema	119
Emphysema	128
Hernia	14
Mass	285
No finding	3045
Pneumonia	63

Table 1: Classwise Distribution of Data

3.4 Dataset visualization

This section displays sample images from some classes:

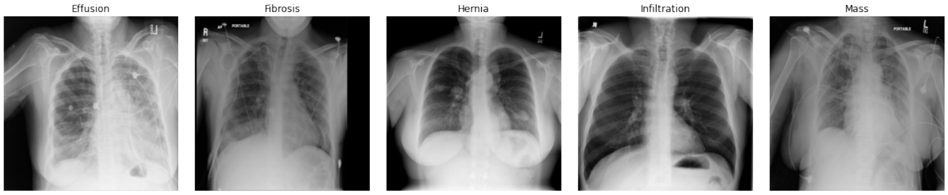


Figure 1: Sample Images from Dataset

3.5 Limitations of our Data

There is an inherent imbalance in the distribution of the original Xray Chest Dataset, as well as the random sample that we choose. This however also replicates the realworld nuances in distribution of the various pathologies that we our dealing with in the real world as well.

4 Experiment

4.1 Software and Hardware

We use Pytorch to train and evaluate our models. We use matplotlib for generating plots. We import the weights of pretrained models(ResNet and VGG-16) from PyTorch Zoo. For Vision Transformers, we use the HuggingFace to get the model's pretrained weights. We use HPC Clusters and CUDA to run and accelerate our experiments. All the code related to our experiments can be found at (1)

4.2 Experimental Setup

We initially trained our models for 100 epochs. However, due to the small size of the dataset, we observed that validation loss starts increasing after about 20 epochs due to overfitting. Therefore, for all our final experiments we trained the model for 15 epochs. We started with larger learning rates 0f 0.01 and 0.001. However, we observed the networks to be unstable and hence we used an initial learning rate 0.0001 with Adam optimiser for back propagation. We evaluate model performance based on the test set accuracy.

4.3 Basline

4.3.1 VGG16

We used a VGG16 model pretrained on ImageNet. We transformed images into sizes of 224 x 224. We choose cross entropy loss function, with Adam Optimiser as our loss function. We trained the model in batches of 32, and for 20 epochs.

4.3.2 ResNet18

ResNet's are a type of deep convolution neural network. ResNet's solved the problem of vanishing gradients, with the help of residual functions. Earlier as the networks went deeper, they started to loose out generalisation capabilities. With ResNet's, the gradients can flow directly through the skip connections backwards from later layers to initial filters. For our experiment, we take the weights of a pretrained ResNet18, and add a fully connected layer to it in order to evaluate our task.

4.4 Vision Transformers

4.4.1 Without Pre-training

We implemented a Vision Transformer using the pytorch library(vit-pytorch). We also implemented efficient attention used by the ViT model using the Linformer library. The attention module has a depth of 12 and 8 attention heads. We implemented a linear layer at the output of the model with 128 hidden neurons. We trained the ViT model on multiple image sizes (224, 256 and 1024) with data preprocessing. We used cross entropy loss along with Adams optimizer and StepLR to decrement the learning rate after every 4 epochs. We trained the model for a total of 20 epochs.

4.4.2 Pre-trained on ImageNet

We implemented a pre-trained vision transformer by importing HuggingFace's transformers ViTFeatureExtractor and ViTForImageClassification. We used ViForImageClassification's from HuggingFace to load Google's "vit-base-patch16-224-in21k" pretrained model (12). This model is pretrained on ImageNet-21k (14 million images, 21,843 classes) at resolution 224x224. A dropout and an output layer with 15 output classes was added to the model. We used cross entropy loss along with Adams optimizer and StepLR to decrement the learning rate after every 4 epochs. We trained the model for a total of 20 epochs.

4.4.3 Self supervised pre-training on ImageNet

We implemented self-supervised vision transformer (DINO) usign the pretarined weights available on HuggingFace models.(13) We used Facebooks's dino-vitb8 pretrained model. The model is pretrained on ImageNet-1k, at a resolution of 224x224 pixels. We added a linear layer at the output of the pretrained model. We used cross entropy loss along with Adams optimizer. We trained the model for a total of 20 epochs.

5 Results

5.1 Model Accuracies

The following table summarises the validation and test accuracies of each model:

Model	**Validation Accuracy**	**Test Accuracy**	**Model Parameters (in millions)**
Baseline model VGG16	45%	43%	138
ResNet34	45%	46.2%	360
Vision Transformer pre-trained using supervised learning	45.3%	44.2%	86
Vision Transformer pre-trained using self-supervised learning	46.5%	48.7%	21
Vision Transformer without any pre-training	45.33%	43.33%	86

Table 2: Result of Experiments

5.2 Training and Validation Loss

The following graphs compare the train and validation loss of each model:

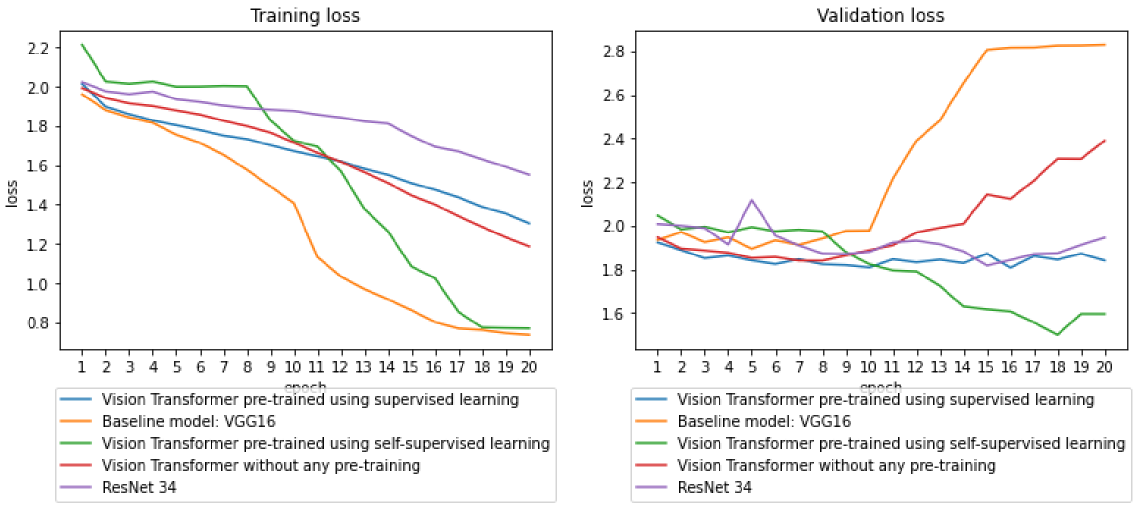


Figure 2: Loss Comparison

6 Discussion

Through our experiments, we found that DINO performed the best with a test set accuracy of 48.7% followed by ResNet34 with an accuracy of 46.2%. However, all the models performed better than the VGG-16 baseline model which has an accuracy of 43%. The SOTA model, a CNN architecture run on this dataset has an accuracy 70.1%. However, the SOTA model was trained on the entire 110K dataset. Since we ran our experiments on a small sample (less than 1/10 of the entire dataset), we believe this may be the reason our accuracy scores are lower. We observed that all model start to overfit after around 15 epochs. This can also be seen from the graph plots. This observation supports our premise of accuracy scores being low due to the dataset size. However, looking at the graphs we can observe that DINO showed the least overfitting closely followed by ResNet34, which could explain the better performance of these models.

The pretraining datasets used do not contain medical image data. Since self-supervised pre-training has been shown to learn features related to semantic segmentation of images, they could generalise better on highly specialised and previously unseen image types. This could also be the reason for lower performance of supervised pretrained ViT. Further, transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data (12). This could explain the better performance of ResNet over other ViTs.

Additionally, we dont see much improvement of pre-trained ViT over not pre-trained one. This may indicate that pretraing on datasets without medical images don't have much impact on medical datasets.

7 Conclusion

Pre-trained vision transformers have shown promising results when fine-tuned with small-medium sized datasets. In (12), Vision Transformer attains excellent results when pre-trained at sufficient scale and transferred to tasks with fewer datapoints. The key idea is increase test accuracy with small sizes of fine-tuning datasets. Based on our experiments, we can conclude that vision transformers have immense potential to achieve this. Further experiments need to be performed to solidify our results. Similar surveys can be done with larger samples of this dataset. Models pretrained on different sizes and different kinds of data, especially medical images should be explored.

References

[1] Code related to experiments. [Online]. Available: https://github.com/aish1795/CV-project

[2] A. Meyer-Baese, A. Meyer-Baese, and V. J. Schmid, *Pattern Recognition and Signal Analysis in Medical Imaging*. Academic Press, 2004.

[3] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, "3d deep learning on medical images: a review," *Sensors*, vol. 20, no. 18, p. 5097, 2020.

[4] Y. Liu, Z. Ma, X. Liu, S. Ma, and K. Ren, "Privacy-preserving object detection for medical images with faster r-cnn," *IEEE Transactions on Information Forensics and Security*, 2019.

[5] L. Liu, F.-X. Wu, Y.-P. Wang, and J. Wang, "Multi-receptive-field cnn for semantic segmentation of medical images," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 11, pp. 3215–3225, 2020.

[6] K. Suzuki, "Overview of deep learning in medical imaging," *Radiological physics and technology*, 2017.

[7] J. Zhang, Y. Xie, Q. Wu, and Y. Xia, "Medical image classification using synergic deep learning," *Medical image analysis*, vol. 54, pp. 10–19, 2019.

[8] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[9] A. Taleb, W. Loetzsch, N. Danz, J. Severin, T. Gaertner, B. Bergner, and C. Lippert, "3d self-supervised methods for medical imaging," *arXiv preprint arXiv:2006.03829*, 2020.

[10] H. Shang, Z. Sun, W. Yang, X. Fu, H. Zheng, J. Chang, and J. Huang, "Leveraging other datasets for medical imaging classification: evaluation of transfer, multi-task and semi-supervised learning," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2019, pp. 431–439.

[11] C. S. Perone, P. Ballester, R. C. Barros, and J. Cohen-Adad, "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling," *NeuroImage*, vol. 194, pp. 1–11, 2019.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[13] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” *arXiv preprint arXiv:2104.14294*, 2021.

[14] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, “Medical image analysis using convolutional neural networks: a review,” *Journal of medical systems*, vol. 42, no. 11, pp. 1–13, 2018.