A Comparison of Computer Vision Models: MobileNet vs The Vision Transformer

Kenneth Hudson

khud1010@gmail.com

Abstract

Computer Vision models have been used for decades to automate image classification tasks that would take a human counterpart much more time and effort to complete. This paper challenges the popular CNN architecture, MobileNet, by comparing it against the Vision Transformer on a medical imaging classification task. Convolutional Neural Networks (CNNs) have been the dominant architecture due to their use of backpropagation, which minimizes error by adjusting connection weights between the many hidden layers. More recently, natural language processing saw the emergence of the transformer architecture that uses an encoder-decoder structure to generate new text or translate languages. In 2021, the Vision Transformer (ViT) was introduced and the decoder was replaced with a classification head that could be used to output label predictions. This challenged both the notion that CNNs are the best, or only, option for computer vision tasks and that transformers could only be used as generation machines. The image content used for training and testing consists of various chest x-rays from the National Institutes of Health's online dataset. These images are very similar in terms of content, so the models will need to understand more complex relationships in order to properly distinguish the classes. The goal of this study is to determine if there is a significant difference between these two architectures by training and evaluating them on a complex image classification task.

Keywords:

CNN, Vision Transformer, Chest X-ray, model architecture

Table of Contents

Introduction and Problem Statement

Medical imaging, like x-rays, can give healthcare providers unseen insight into the proper diagnosis for a patient. However, reviewing these images can be repetitive and humans make mistakes, resulting in missed diagnoses. This paper trains and compares two popular computer vision models, MobileNet and ViT, on chest x-rays in order to find the more appropriate model to use in this medical setting. The CNN architecture has historical significance and has been utilized in practice by many industries. It uses multiple layers of convolution and pooling to grow its receptive field and detect features that become more complex as the data passes through more layers. MobileNet is unique in that it uses depthwise separable convolutions, instead of the standard convolutions, to reduce the number of parameters and overall computational cost during training. The transformer architecture utilizes a different method for training that was originally intended for natural language processing tasks. Images are split into patches and then flattened into embedding vectors that are fed sequentially through the model encoder, which is similar to how text is given to an NLP model. Then, in lieu of a decoder, outputs are passed to a classification head that can predict labels. Both of these models are trained on a dataset of chest x-rays and on the same machine in order to reduce the possibility of confounding variables being introduced. These images all contain very similar content, so understanding the subtle differences between classes will be difficult. Evaluation and performance metrics are recorded and compared to determine which architecture performs better on this medical imaging classification task. The comparison of these models will highlight the key differences between model training algorithms.

Literature Review

The basic CNN architecture has been around for decades as the dominant option in computer vision modeling. Works on CNNs are numerous and go back as far as 1980, when the Neocognitron was introduced (Fukushima 1980). The alternating layers of convolution and pooling paved the way for more teams like John Denker's to develop the first neural network that could be used for hand-writtten digit recognition (Denker et. al 1988). A year later, Yann LeCun and team introduced backpropagation in a paper that changed how model training was done forever. That paper introduced 'weight sharing,' allowing every node in each layer to share a common variable, updated each epoch by calculating the margin of error and applying a small learning rate to favor correct predictions (LeCun et al. 1989). This eliminated the need for engineers to "hard code" weight connections, as models could now update automatically. Despite this breakthrough, CNNs didn't gain popularity until AlexNet won the ImageNet competition in 2012. AlexNet used the rectified linear unit (ReLU) activation function, introducing non-linearity by setting negatives to zero and leaving positives unchanged. This increased

training efficiency and reduced overfitting (Krizhevsky, Sutskever & Hinton 2017). Since AlexNet's win, many new CNN architectures have been introduced to improve validation metrics. In this paper, MobileNet was chosen for its lightweight and efficient design.

MobileNet was originally introduced by Andrew Howard and his team at Google in 2017. The main feature of this new architecture was a depthwise separable convolution, which factorizes the standard (2-D) convolution and splits it into two separate convolutions, a depthwise convolution and a pointwise convolution. The standard convolution effectively filters features based on the kernels (2x2 pixels, 3x3 pixels, etc.) and the combination, or pooling, of those outputs into a new representation of the image. The difference with a depthwise separable convolution is that it separates this filtering and combination step into two separate calculations and then sums the outputs. The depthwise convolutions apply a single filter, in each input channel, and then pointwise convolutions, which are really 1x1 convolutions, are used to combine the output of the depthwise layer. This has the same effect as a standard convolution, but with less calculations needed as the matrix multiplication is reduced. The authors of the MobileNet paper claim that a 3x3 depthwise separable convolution has upwards of eight times less computational demand than a standard 3x3 convolution (Howard et al. 2017). This, applied over many convolutional layers, can see massive reductions in model parameters and an increase in efficiency. CNN architectures, like MobileNet, have reigned supreme the last few decades. However, a new architecture, Transformers, would challenge that idea and attempt to break into the realm of image classification.

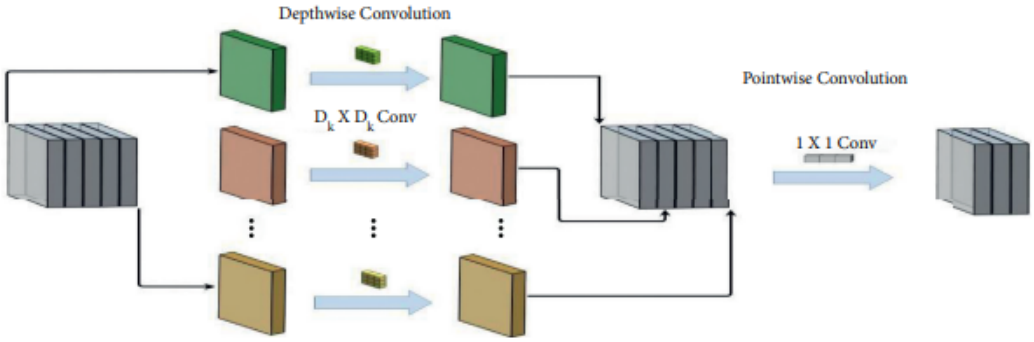


Figure 1: Depthwise Separable Convolution (Kadam, Ahirrao, & Kotecha 2022)

Vaswani and their team released the Transformer architecture in 2017 and it relied on an encoder-decoder structure as well as the attention mechanism to learn relationships across input text data and perform NLP tasks. Recurrent Neural Networks (RNNs) were all the buzz in NLP until the transformer sparked a new generation of models, such as Google's BERT (Bidirectional Encoder Representations from Transformers) and OpenAI's GPT (Generative Pre-trained Transformer). These transformers work by sequentially passing text to an encoder, which

includes a multi-headed attention mechanism and then a fully connected network. The outputs from these layers are representations of the input and are mapped to be used later by the decoder. The decoder differs slightly from the encoder in that it uses these representations to guess the next output in the sequence based on the previous pass through the model. The key to transformers is the self-attention mechanism that allows the current input to look at the prior embeddings in the input sequence and determine a better encoding based on the context to the sequence. The model can then update the input embeddings before feeding the next embedding to the decoder and thus provide more context to the next output (Vaswani et al. 2017). Transformers changed the NLP landscape but didn't break into the computer vision area of deep learning until 2021, when the Vision Transformer was introduced.

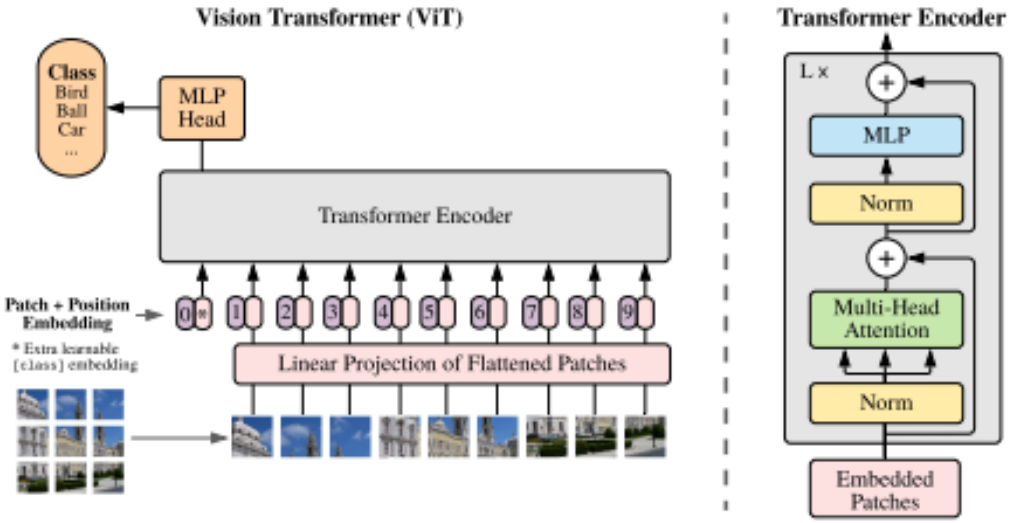


Figure 2: Vision Transformer (Dosovitskiy et al. 2021)

Dosovitskiy and their team at Google Brain (2021) decided to introduce a new transformer that removed the decoder entirely and used image data in order to perform object recognition tasks. They decided to split an image into multiple patches, then flatten each one into a vector and attach a position embedding. These embedded vectors are then passed to the normal transformer encoder as described above. The difference from regular transformers is the replacement of the decoder with a classification head, consisting of a multi-layer perceptron, a softmax function, and a "class" token, similar to BERT (Devlin et al. 2019). This class token is what is used in the final classification head to predict the label based on the probabilities of all possible outputs. By replacing the decoder, the model no longer has its generative capabilities but can now perform discriminative tasks which was a huge shift from the previous usability of transformers. So how do they compare to CNNs on image classification tasks that are not as obvious as those used in the vision transformer paper (ImageNet/CIFAR). This paper tests how

these models stack up against each other when presented with a more subtle task of recognizing differences in chest x-rays.

Previous research on this dataset, detailed by Wang et al. (2017), utilized radiology reports and natural language processing to address multi-label classification with various CNN architectures, including AlexNet (Krizhevsky, Sutskever, & Hinton, 2017). Li Yao's team (2018) experimented with a DenseNet encoder and LSTM decoder, while Kufel et al. (2023) achieved some success with EfficientNet. These studies demonstrate how different architectures can improve model training and validation on this specific dataset. However, they predominantly use variations of the basic CNN architecture or training methods. This paper attempts to differentiate itself by comparing two fundamentally distinct architectures, employing varied techniques to understand complex data relationships.

Data

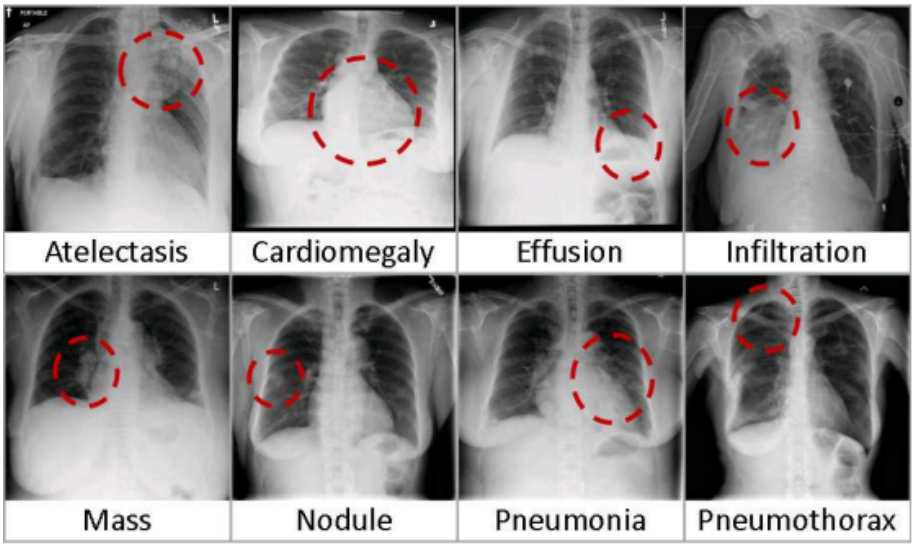


Figure 3: Abnormal Chest X-Rays (NIH Chest X-Ray Dataset)

Chest X-Rays (Figure 3) are a common tool used in the medical field to determine if a patient is suffering from any complications related to some of the body's most vital organs, the heart and lungs. Much of a diagnosis, and the subsequent treatment, relies on the proper interpretation of the x-ray by a physician. The data used to train these models during experimentation were supplied by the National Institutes of Health as an open source dataset (a link can be found in the Data Availability section). Each image in the dataset is originally a 1024 x 1024 pixel .png file in black and white. The dataset also comes with a .csv file that contains the labels for each image. There are 15 total labels in the dataset and some images had multiple labels assigned to it. The label 'No Finding' represents more than 50% of the data (Figure 4),

while infiltrations, atelectasis and effusions make up just over 25% of the images that do have findings associated with them. Due to this class imbalance, the labels were filtered into two classes each representing about half of the entire dataset. By sorting the labels into binary dataset, not only is the data more balanced (Table 1), but the model will train on a simpler classification task, instead of a multi-label classification problem. The new binary task will determine whether or not the x-ray has 'No Findings' or should be flagged as 'Unhealthy'. The images were separated into training and validation folders to be used for their respective tasks based on two text files accompanying the dataset. This split cut the full dataset of 112,120 images into 86,524 (77.17%) training images and 25,596 (22.83%) validation images.

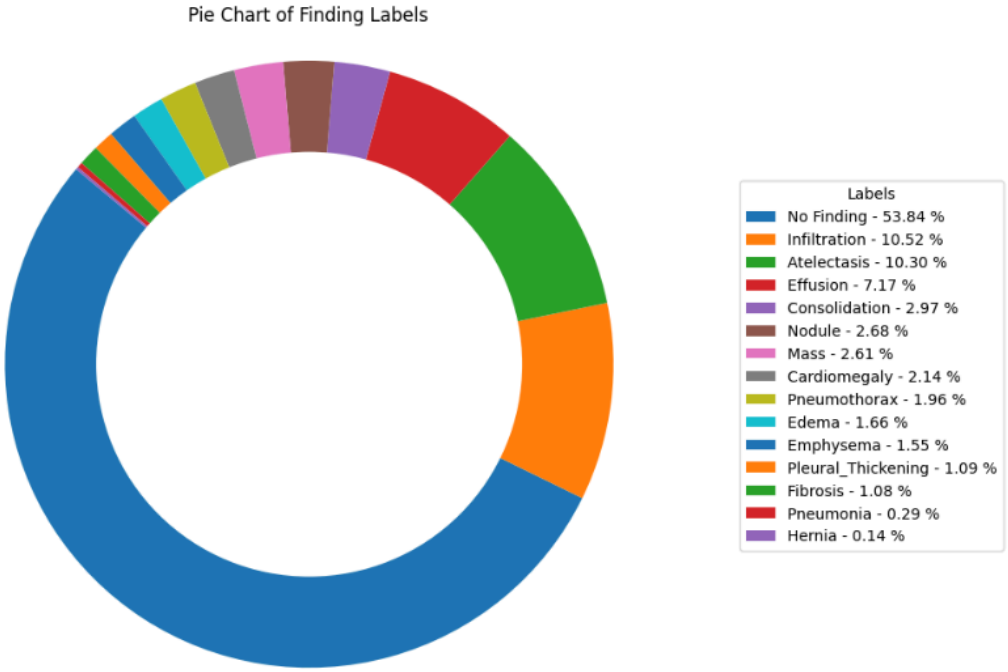


Figure 4: Pie chart showing the distribution of labels (based on percentage) in the dataset

Class	Count	Percentage
No Finding	60361	53.84%
Unhealthy	51759	46.16%
Total	112120	100%

Table 1: Number of images in each class along with the respective percentage of images in the total dataset

Methods

The two models created for this project were built in python using the popular deep learning library, pytorch. Both MobileNet and the Vision Transformer are available in pytorch's model database with pre-trained weights from the ImageNet-1K dataset. The construction of both followed a similar format and all data pre-processing, including the train-test split, was done before training occurred. The image processing pipeline included resizing images to 224 pixels by 224 pixels and normalizing the data, which helps to standardize the distribution of pixel values. Model training was run on a Nvidia TITAN Xp GPU for 10 epochs with the recommended pytorch parameters. Graphics Processing Units (GPUs) are known for accelerating processing speeds through parallelization of tasks across many cores (Gyawali 2023). Machine learning practitioners have found that parallelism is quite useful when training deep learning models, which can require many matrix multiplication calculations. Speeding up training allows more tests to be conducted with various hyperparameters to be tested. Run times for both models are tracked and discussed in the Results sections

The Adam optimizer, short for Adaptive Moment Estimation, with a learning rate of .001, was utilized for both models to minimize the loss function via a stochastic gradient descent algorithm (Kingma and Ba 2014). It is particularly advantageous because it balances the benefits of momentum and adaptive learning rates to make it well-suited for complex modeling. A confusion matrix was used at the end of model training to compare the predictions to the actual labels. Metrics such as accuracy, precision, and f-1 score were all calculated using the confusion matrix and provide insight into the model's predictive power.

Results

The results in Table 2 are calculated using the confusion matrices (A.1 and A.2) for each model. MobileNet outperforms ViT across most metrics, achieving higher accuracy (0.70 vs. 0.58), recall (0.89 vs. 0.56), and f1-score (0.79 vs. 0.62). ViT demonstrates a slightly better performance in specificity (0.60 vs. 0.39), while both models perform equally well in precision (0.70 vs 0.69), indicating they are both effective at identifying positive instances. The f1-score, which balances precision and recall, further highlights MobileNet's superior performance. Initial training on a CPU resulted in extremely long training times—2 hours per epoch for MobileNet and 18 hours per epoch for ViT. This led to a decision to move training to a GPU, where training times were significantly reduced and closer to each other. The GPU epoch runtimes (A.3) were consistently about half for MobileNet compared to ViT, around 33 minutes (~1985 seconds) versus 60 minutes (~3562 seconds), respectively.

	Model	
Metric	**MobileNet**	**ViT**
Accuracy	0.70	0.58
Precision	0.70	0.69
Recall	0.89	0.56
Specificity	0.39	0.60
f1-score	0.79	0.62
Seconds/Epoch	~1985	~3562

Table 2: Evaluation Metrics

The loss plots (Figure 7) show that both models were reducing their loss in a similar trend and were showing no signs of overfitting. MobileNet has a lower starting loss (~0.60) as well a larger decrease in loss over the training period (a change of 0.06). However, ViT starts with a higher loss (~0.69) and finishes with a loss score less than the starting loss of its competitor. Overall, MobileNet exceeds ViT and shows that bigger models with more parameters do not always translate to more predictive power.

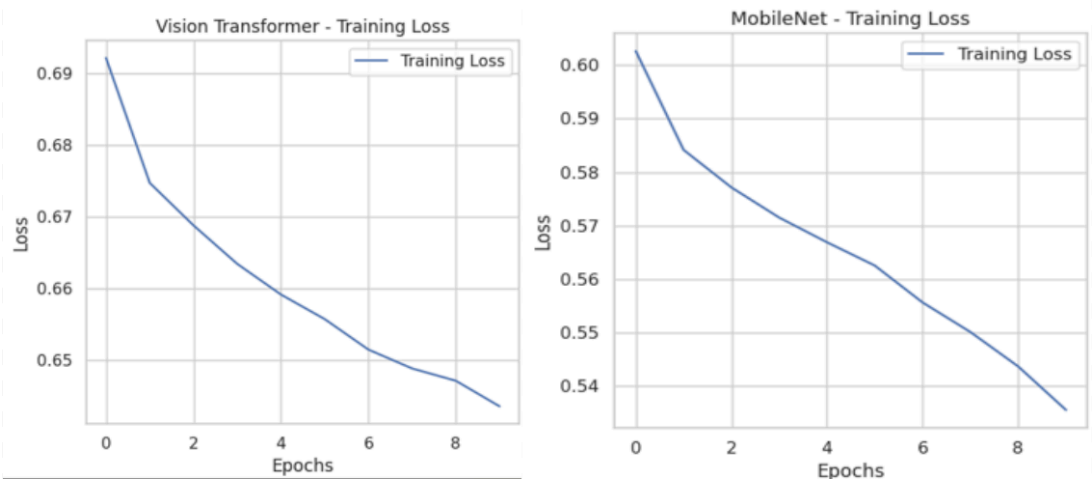


Figure 7: Loss plots

Discussion

The results of this study highlight the differences in performance between MobileNet and ViT when applied to a chest x-ray image classification task. Despite having far fewer parameters and a less complex architecture, MobileNet consistently outperformed ViT in virtually every key evaluation metric. The difference in model performance may be attributed to the nature of the

dataset and the classification problem at hand. MobileNet, with its simpler architecture, is likely better at generalizing from the limited features present in the chest X-ray images. ViT, which was designed to capture more intricate patterns, may struggle with the relatively low-resolution images or the less complex features inherent in this type of medical imaging. Additionally, the binary classification task likely reduced the necessity for the more sophisticated global pattern recognition capabilities of ViT, thereby favoring MobileNet's streamlined approach.

ViT relies on the self-attention mechanism, which scales quadratically with the number of input tokens (in this case, image patches). This mechanism allows ViT to capture long-range dependencies in the data but requires substantial computational power. In contrast, MobileNet uses depthwise separable convolutions, a more efficient type of convolution that significantly reduces the number of parameters and the amount of computation required compared to standard convolutions. This efficiency makes MobileNet faster and less computationally demanding. So, while ViT's complex architecture enables it to model intricate patterns in data, this complexity comes at the cost of increased computational demands, both in terms of memory and processing power. This proved to be unnecessary for this particular classification and shows its limited ability to be used in practice.

An unintended, yet significant, finding is the impact of computational resources on model training. The study showed that both models benefited from GPU acceleration. The training time of each epoch for ViT was substantially changed, reducing it from 19 hours to 1 hour. Compare this to MobileNet, which only saw a reduction from 2 hours to 30 minutes, and it can be seen how impactful a GPU is on certain models. This underscores the importance of considering both model performance and computational efficiency when selecting a model for practical applications, especially in resource-constrained environments.

Conclusions

This study demonstrates that for the task of binary classification of chest X-rays, MobileNet offers a more practical and effective solution compared to the Vision Transformer. MobileNet's superior performance across accuracy, recall, and f1-score, coupled with its faster training time, suggests that it is better suited for this type of medical imaging task. This finding is particularly relevant for real-world applications where computational resources may be limited, and quick, reliable predictions are essential. Additionally, MobileNet's architecture allows for easy integration into existing healthcare systems, providing a scalable solution for hospitals and clinics.

An accidental finding was that utilization of GPU acceleration affected ViT training times far more than MobileNet’s. While powerful, this means expensive resources are needed to handle

the additional burden of that computation and makes it a poor choice for certain use cases. This limitation underscores the importance of considering both performance metrics and resource requirements when selecting models for deployment. MobileNet's lightweight design and efficient convolutions do not sacrifice predictive power for computational burden, proving that CNN architectures are still a reliable option for computer vision tasks on medical imaging.

Directions for Future Work

Further optimization of both models, including fine-tuning hyperparameters and experimenting with different pre-processing techniques, is necessary before seeing the real-life application of a model like this. This would possibly require hardware upgrades, such as a more powerful GPU or more RAM, to handle the increased computational demand effectively. More epochs are also needed as both loss plots still appeared to trend downward and could indicate that both models still had more to offer. This suggests that neither model has fully converged, and further training could enhance their performance. Additionally, it would be beneficial to test these models on a more complex classification task, such as multi-classification of the 15 original labels, to assess their generalizability and robustness. Expanding the scope of the task would provide a clearer understanding of how each model performs under different conditions, potentially highlighting further strengths and weaknesses.

Data Availability

National Health Institutes Chest X-Ray Dataset:

https://nihcc.app.box.com/v/ChestXray-NIHCC

Code Availability

https://github.com/khud1010/ViT_vs_MobileNet

References

Denker, J, Gardner, W., Graf, H., Henderson, D., Howard, R., Hubbard, W., Jackel, L., Baird, H., and Guyon, I. 1988. "Neural Network Recognizer for Hand-Written Zip Code Digits." *Advances in Neural Information Processing Systems*. Morgan-Kaufmann.

Devlin, Jacob, Chang Ming-Wei, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv.Org*. https://doi.org/10.48550/arxiv.1810.04805

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale." *arXiv.Org*. https://doi.org/10.48550/arxiv.2010.11929.

Fukushima, Kunihiko 1980. "Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/bf00344251

Gyawali, Dipesh. 2023. "Comparative Analysis of CPU and GPU Profiling for Deep Learning Models." *arXiv.* https://arxiv.org/pdf/2309.02521.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Hartwig, A. 2017. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1704.04861

Kadam, K. D., Ahirrao, S., and Kotecha, K. 2022. "Efficient Approach towards Detection and Identification of Copy Move and Image Splicing Forgeries Using Mask R-CNN with MobileNet V1". *Computational Intelligence and Neuroscience.* https://doi.org/10.1155/2022/6845326

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv.* https://arxiv.org/abs/1412.6980.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey 2017. "ImageNet classification with deep convolutional neural networks." *Communications of the ACM* (Vol. 60, Issue 6, pp. 84–90). https://doi.org/10.1145/3065386

Kufel, Jakub, Bielówka, Michal, Rojek, Marcin, Mitręga, Adam, Lewandowski, Piotr, Cebula, Maciej, Krawczyk, Dariusz, Bielówka, Marta, Kondoł, Dominika, Bargieł-Łączek,

Katarzyna, Paszkiewicz, Iga, Czogalik, Łukasz, Kaczyńska, Dominika, Wocław, Aleksandra, Gruszczyńska, Katarzyna, and Nawrat, Zbigniew 2023. "Multi-Label Classification of Chest X-ray Abnormalities Using Transfer Learning Techniques." *Journal of Personalized Medicine, 13(10), 1426-*. https://doi.org/10.3390/jpm13101426

LeCun, Y, B Boser, J. S Denker, D Henderson, R. E Howard, W Hubbard, and L. D Jackel. 1989. "Backpropagation Applied to Handwritten Zip Code Recognition." *Neural Computation 1, no. 4. 541–51.* https://doi.org/10.1162/neco.1989.1.4.541.

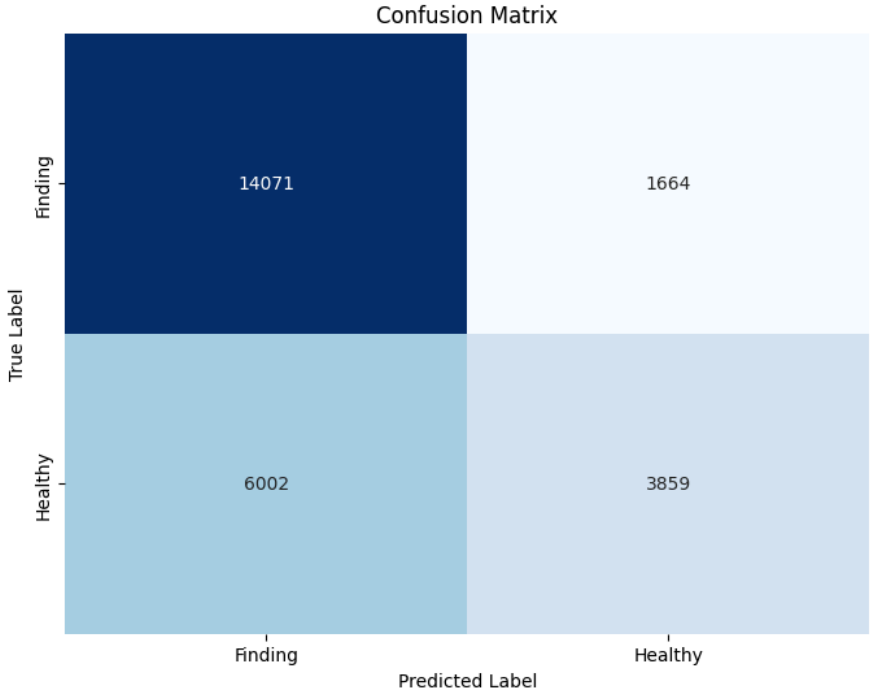
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *arXiv.Org.* https://doi.org/10.48550/arxiv.1706.03762.

Wang, Xiaosong, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. 2017. "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases." *arXiv.Org*. https://doi.org/10.48550/arxiv.1705.02315.

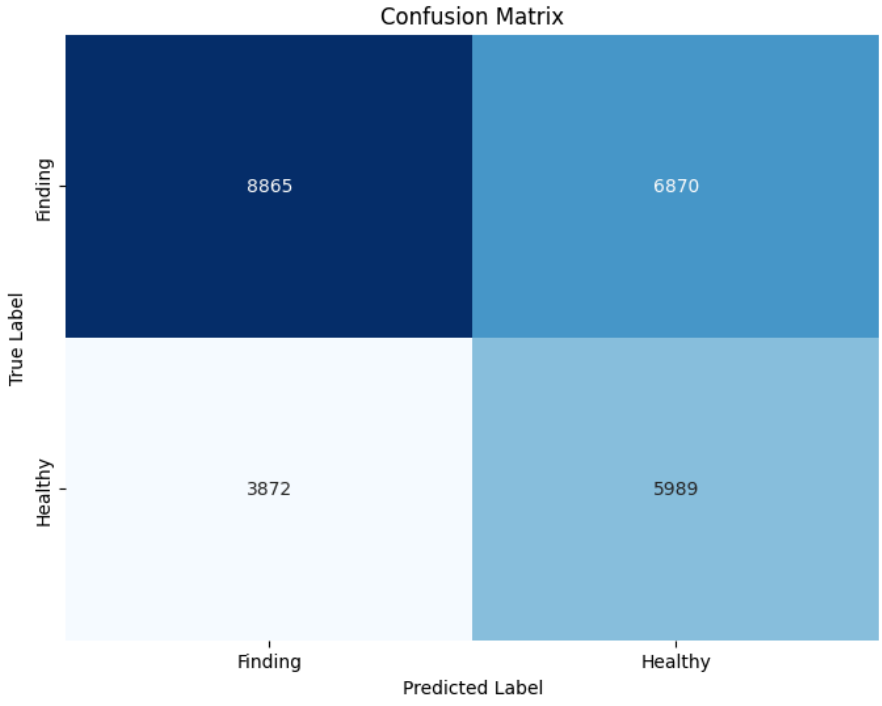
Yao, Li, Poblenz, Eric, Dagunts, Dmitry, Covington, Ben, Bernard, Devon, & Lyman, Kevin 2018. "Learning to diagnose from scratch by exploiting dependencies among labels." *Cornell University.* https://doi.org/10.48550/arxiv.1710.10501

Appendix A

A.1: Mobile Net Confusion Matrix



A.2: Vision Transformer Confusion Matrix



A.3: Epoch Times