FMDB *Transactions on Sustainable Computing Systems*

Performance Evaluation of Vision Transformers for Diagnosis of Pneumonia

Srishti Lodha[1,*], Harsh Malani[2], Arvind Kumar Bhardwaj[3]

[1,2,] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.
[3]Department of Information Technology, Capgemini, Houston, Texas, United States of America.
shrishti2k1@gmail.com[1], harsh13092001@gmail.com[2], arvind.bhardwaj@capgemini.com[3]

Abstract: Pneumonia, a severe and life-threatening result of bacterial or viral infection, can inflate air sacs in the lungs. The disease, which can target young and old people, is common in several countries. Generally, blood tests, pulse oximetry, sputum tests, CT scans, and chest X-rays are used to diagnose Pneumonia. Deep Learning (DL) models can be excessively helpful in analyzing the results of these tests. Over the past few years, several studies have suggested the implementation of different DL architectures for Pneumonia detection. However, these incorporate many trainable parameters for feature extraction from images, leading to a significantly high training time and resource consumption. Moreover, convolutions become monotonous after a certain number of layers, making it extremely difficult to improve the accuracy. In this research, we use Vision Transformers (ViT) for Pneumonia detection, an image classification architecture developed by modifying transformers in 2021. To our knowledge, ViT has only been implemented in one study before this research for Pneumonia diagnosis. Our approach outperformed all existing research and state-of-the-art architectures in this domain regarding all performance metrics and training time and recorded a validation accuracy of 98.18%. We also compare our model's performance with other tuned DL models (CNN) and analyze the performance gap.

Keywords: Performance Evaluation; Vision Transformers; Diagnosis of Pneumonia; CT Scan and Chest X-rays; Deep Learning (DL) Models; Architectures for Pneumonia; Blood Tests.

Received on: 04/11/2022, **Revised on:** 03/01/2023, **Accepted on:** 09/02/2023, **Published on:** 25/02/2023

Cited by: S. Lodha, H. Malani, A. K. Bhardwaj, "Performance Evaluation of Vision Transformers for Diagnosis of Pneumonia," *FMDB Transactions on Sustainable Computing Systems*., vol. 1, no. 1, pp. 21–31, 2023.

1. Introduction

Community-acquired Pneumonia [3] [4] is an infectious respiratory disease with several symptoms, the most common of which are chest pain, chills and fever. It may even cause death. People with a weak immune system, like Asthma patients, are at a greater risk of getting Pneumonia. Moreover, the severity has increased amidst the COVID-19 pandemic, infection from which can easily cause COVID-19 Pneumonia. Mycoplasma, Bacterial, Nosocomial, and Chlamydial are other types of Pneumonia. Deep Learning (DL) plays a crucial role in the medical field because it helps improve and speed up the process and assists. Quick diagnostic measures with DL are extremely useful in case of a serious yet common disease like Pneumonia. This has been an area of interest for researchers for many years, and several Pneumonia detection models using VCG, ResNet, Mask-RCNN, and dozens of other architectures have been proposed. Detecting if a person is infected based on X-RAY scans (easily obtainable) is an image classification task that conventionally demands the use of a Convolutional Neural Network. However, these have a very high training time, with many trainable parameters for feature extraction from the input images. With increasing complexity and values for the hyperparameters, challenges like vanishing gradient can also be caused. The resource

[*]Corresponding author.

consumption is also significantly increased in the process. Moreover, once the convolutions become monotonous (after a certain number of layers) and hit a plateau, improving the model's accuracy becomes extremely difficult.

A highly efficient alternative for image recognition problems, Vision Transformers (ViT) [2], emerged in 2021. It was modified from Transformers [1] (tremendously in demand for Natural Language Processing), a new model introduced by Google in 2017. ViT outperformed existing CNN architectures and state-of-the-art results, quickly gaining popularity. It is as much as 4 times more computationally efficient than CNN and trains much faster while maintaining a high training and validation accuracy. Hence, we decided to use this extremely new and promising architecture for the important classification agenda.

The rest of the paper begins with a literature survey of closely related studies to highlight the contributions of existing research in the domain (section 2). We then move to the proposed methodology in section 3, where we introduce ViT and clearly explain the steps involved in data acquisition, processing, and model development. The results obtained (section 4) are compared with existing research and state of state-of-the-art, and with a tuned custom CNN model, we build to demonstrate the performance and efficiency gap between the architectures. Finally, we move to the conclusion in section 5, followed by the references.

2. Literature Survey

Studies have used different CNN architectures for Pneumonia detection from X-ray images. ViT was used for the same in only 1 prior research: Tyaggi et al. [5]. Here, chest X-ray images were used to obtain and compare the results from 3 different architectures: CNN, VGG16 and ViT. ViT achieved the highest validation accuracy but was only 86.38% (with a training accuracy of 96.45%). Hasan et al. [11] and Hammoudi et al. [17] focused on Covid-19 induced Pneumonia detection. The first study followed a classical CNN approach to train a tuned VGG16 architecture. An 80-20 train-test split on the dataset of chest X-rays yielded a validation accuracy of 91.69% through the proposed tuned model using the Adam optimizer. The latter trained several deep-learning architectures to detect and classify chest X-ray images as bacterial, viral or normal. While InceptionResNetV2 showed the least false positive rate, DenseNet169 achieved the highest overall classification accuracy of 95.72%.

To classify if a chest X-ray has a normal, Bacterial or Viral Pneumonia-infected lung, several CNN architectures were explored by Jain et al. [12]. SoftMax activation, pooling, flattening, dropout techniques and Adam optimizer were employed in training 2 CNNs (2 and 3 layers, respectively), VGG16 and VGG19, Inception-v3, and ResNet50. They conclude that VGG19 outperformed the other transfer learning models with the highest accuracy (88.46% for validation) and least overfitting. However, training and validation losses were still very high in this study. In a similar study, Asnaoui et al. [13] tuned as many as 9 CNN architectures using chest X-ray and CT datasets with 6000 images. These include baseline CNN, VGG16 and VGG19, Xception, Resnet50, etc. Resnet50 showed the best performance with an accuracy exceeding 96%.

Hashmi et al. [14] adopted a unique weighted classifier-based supervised learning approach which combined the weighted predictions from 5 deep learning models, including Resnet18 and Xception. For a balanced improvement in the number of samples for training, data augmentation (partial) was applied. Guangzhou Women and Children's Medical Center pneumonia dataset was used to test the hybrid model, which outperformed the individual-tuned models with a test accuracy of 98.43%. Wu et al. [15] also developed a hybrid model using ACNN (CNN with adaptive median filter recognition), based on a classical Machine Learning model, Random Forest (RF) Classifier. Using ACNN helped clean the data and achieve the required activation in each X-ray image. RF, tuned with GridSearchCV, was then applied. The proposed model detected Pneumonia from chest X-ray scans with an accuracy of 97%.

Ibrahim et al. proposed the AlexNet model-based deep learning approach [18] to classify chest X-ray scans as normal or COVID19/non-COVID-19 (viral/bacterial) infected. For each of the types, segregated datasets were used. Scans were collected from various online sources, including GitHub and Kaggle. 70-30 train test split was used, and an overall classification accuracy of 95.54% was observed. However, using different datasets with highly varied sizes in this study introduces a bias in the model and makes it difficult to generalize the model. The model proposed in Jaiswal [19] is Mask-RCNN based. Local and global features are incorporated for pixel-wise segmentation and a processing step that merges bounding boxes from several models. Here, symptoms are detected from chest radiographs. The model's output is negative or positive with predicted bounding boxes around lung opacities.

Ayan and Unver [16] also use transfer learning to diagnose Pneumonia from chest X-rays. VGG16 and Xception have been used in their study, which acquired maximum accuracies of 87% and 82%, respectively. To maintain unbiasedness in the data, they also use standard augmentation techniques like rotation, shifting, zooming, etc. The final model had 144M trainable parameters with categorical cross entropy as the loss function and RMSprop as the optimizer. Zech et al. [20] believe CNNs perform poorly when generalizing new data. Their study revolves around performance measures of CNNs for generalizing new data collected from 3 hospitals for pneumonia screening. They used a total of 158,323 images for evaluation. The dataset was

fairly balanced. They concluded that the hospital-based system had an AUC score of 0.86, and CNNs trained on this pooled data had an AUC of 0.93. CNNs achieved better performance than hospital-based systems. Vats et al. [21] also use VGG16 and VGG19, along with InceptionV3 and MobileNet, to detect Pneumonia. They designed a new architecture, DenseNet, to overcome certain limitations of traditional transfer learning techniques [22]. They achieve a maximum validation accuracy of 92.6% and compare results with existing state-of-the-art networks [23]. Table 1 presents a summary of the related works to their year of publication, models employed and best-achieved validation accuracy. The comparison also includes the current research study.

Table 1: Summary of related works

Paper	Year of publication	Models employed	Highest validation accuracy (%)	Most accurate model
Tyaggi et al. [5]	2021	CNN, VGG16, ViT	86.38	ViT
Hasan et al. [11]	2021	CNN, VGG16	91.69	VGG16
Jain et al. [12]	2020	2- and 3- layer CNN, VGG16, VGG19, Inception-v3, ResNet50	88.46	VGG19
Asnaoui et al. [13]	2021	CNN, DenseNet201, Xception, Inception_ResNet_V2, VGG16, MobileNet_V2, VGG19, Resnet50, Inception_V3	96	Resnet50
Hashmi et al. [14]	2020	ResNet18, DenseNet121, Xception, MobileNetV3, InceptionV3	98.43	Proposed hybrid weighted classifier
Wu et al. [15]	2020	ACNN, RF	97	Proposed hybrid ACNN-RF
Hammoudi et al., [17]	2021	ResNet50, DenseNet169, VGG19, Inception ResNetV2, RNN	95.72%.	DenseNet169
Ibrahim et al. [18]	2020	AlexNet	95.54%	AlexNet
Jaiswal [19]	2019	Mask RCNN	-	Mask RCNN
Ayan and Unver [16]	2019	VGG16, Xception	87%	VGG16
Zech et al. [20]	2018	CNN	-	CNN
Vats et al. [21]	2022	VGG16, VGG19, InceptionV3, MobileNet, DenseNet	92.6%	DenseNet
Our work	**-**	**ViT, CNN**	**98.18**	**ViT**

3. Proposed Methodology

This section explains the entire methodology (Fig. 1) followed for this research, starting from data acquisition to data processing and, finally, model training and validation, in-depth.

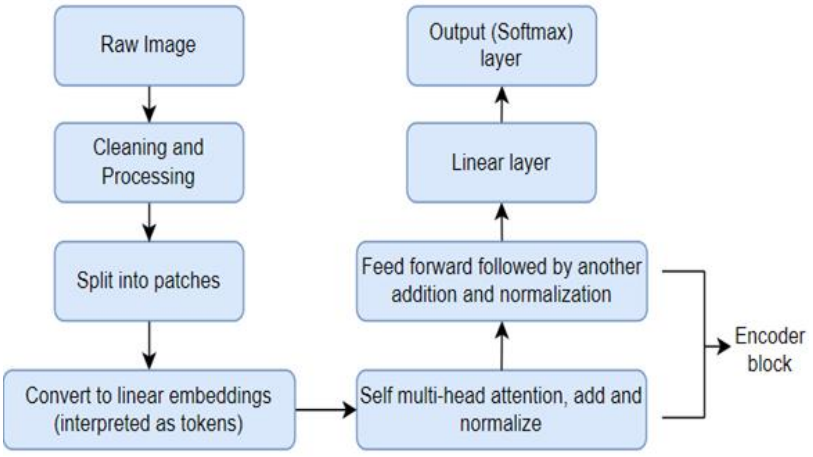

Figure 1: Proposed approach

3.1. Data Acquisition

We used chest X-ray data from Guangzhou Women and Children's Medical Center, Guangzhou, by Kermany et al. [6]. This data wasn't explicitly collected for the sake of the experiment; rather, the imaging was done as part of a regular scan of the patient. To ensure the data had no unreadable or low-quality scans, all X-rays were passed through radiologists to clear them for the model [24]. Any non-suitable scan was discarded.

The total number of clean scans in this dataset is 5863. This is split into 3 categories: train, test and validation. The dataset is divided as follows:

- 5216 scans for training
- 624 scans for testing
- 16 unseen data for final validation

Each category has 2 classes: "normal" for normal lung and "pneumonia" for Pneumonia suspected/diagnosed lung. To avoid any potential error in the dataset, the radiologists and the dataset authors maintained a constant consultation [25].

3.2. Data Processing

Pneumonia is detected and diagnosed via certain clouding effects observed in the lung X-ray scans [26]. If a lung is infected, it appears opaque compared to a normal scan. Since the data has subtle features distributed across the entire scan, it leaves limited scope for segmentation or other image processing techniques. Hence, we use raw X-ray scan data to train our model, and no pre-processing steps are required [27].

Nevertheless, input data has to be processed in some or the other way. There are also certain processing steps when it comes to Vision Transformers [28]. Like any other architecture, the input images must be resized into a fixed specific shape. We resize the input images to size 144x144 [29]. Moreover, the scans were 3-channel images converted to 1-channel images, as X-rays are essentially grayscale images [30]. This reduces the redundant channels and computational resources required along with it.

The images were also augmented before the model training. This was embedded with the model itself, and a separate layer was created for data augmentation. For augmenting, we used 3 criteria:

- Image flipping: A few images were randomly flipped along the vertical axis to generate a new image
- Image rotation: A few images were randomly rotated along both directions, and the amount of rotation was fixed at 0.015
- Image zoom: few images were zoomed with the factor of 0.2 along both height and width
- This augmenting pipeline was applied to every image in the training sample.

3.3. Methodology

Transformers were first introduced in attention is all you need [1] as state-of-the-art architecture for natural language processing [31]. The proposed new architecture was based on an attention mechanism instead of having an encoder-decoder configuration connected by an attention mechanism. Transformers can easily outperform other recurrent mechanisms on large and limited training data [32]. Vision Transformer (ViT) [2] uses a similar image recognition and classification mechanism. The essence of ViT is self-managed attention. To get into ViT, we must understand the basic units and working of a transformer and its attention mechanism [33].

Transformers are also a sequence-to-sequence (seq2seq) [7] model based on an encoder-decoder mechanism without having recurrent units [8]. A transformer has an encoder map which maps the input symbol sequence, let's say, (x1,x2, ..xn) to a continuous sequence (z1,z2, ..zn). The decoder uses this continuous sequence to generate the output sequence (y1,y2,..ym) [43]. The model also considers the previously generated symbols as extra input for the current symbol, which makes it capable of retaining context and hence makes the model auto-regressive. The internal architecture of a transformer encoder is shown in Fig. 2.

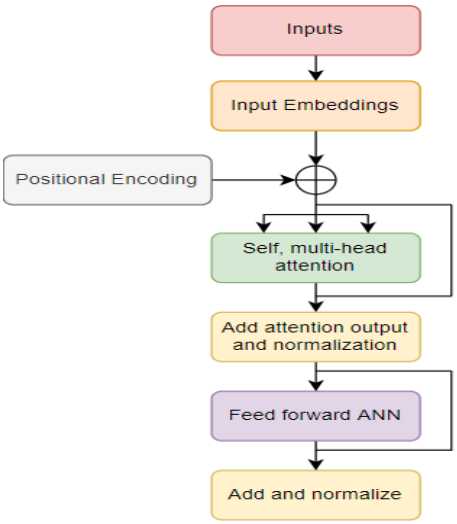

Figure 2: Internal architecture of a transformer encoder

There are 2 major parts associated with the encoder block:

- Self multi-head attention
- A feed-forward neural network

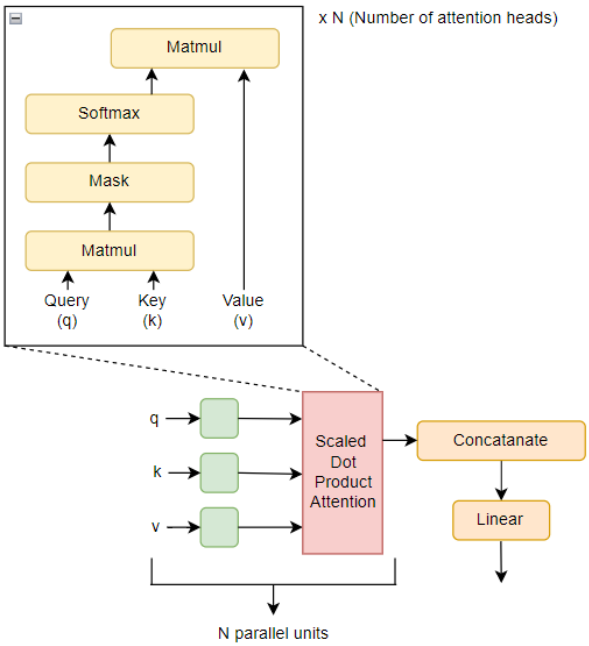

Figure 3: Self multi-head attention block

In multi-head attention, instead of a single attention unit, the attention module has multiple attention units or heads working parallelly, which are later on merged (refer fig. 3). The attention module takes in Q (Query), K (Key) and V (Value) and splits it into N pieces. This N is the number of attention heads the model will use. Each of these splits is independently passed through a separate head, and the attention module runs computations on each head parallelly. These attention scores are then combined to get a final score, passed to further layers.

The reason for this is that with multi-head attention, each input split can learn differently about every other split, which boosts the performance of the model by enabling the transformer to interpret the input sequence better. In the case of ViT, instead of input symbols or tokens, the image is split into various sections, and these splits are used for multi-head attention (explained later in this section). To summarize, having these splits essentially allows the transformer model to have different interpretations

of all other segments viewed via our particular attention head's segment. Hence, better interpretation of the input image and better feature extraction.

Self-attention has the same number of output vectors as input sequence vectors. For example, if the input sequence is (x1, x2, .. xm), the output context sequence will be (c1, c2, .. cm). The context vector ci will always be in the position of the input sequence vector xi, but it depends on all other xis. To simplify, with self-attention, each input sequence vector can interact with every other sequence vector via certain computations to decide which input vector should get the highest attention or weightage.

In our case of ViT, each of the individual input sequences from the multi-head attention module can interact to decide which input image segment should get maximum attention and weightage. This can be compared to feature extraction done in CNN via convolutions. Instead of convoluting the entire image, we break the image into segments to determine which part of the image has some useful features significant enough for classification. Another thing to note here is that in the case of multi-head attention, one head's context vector is not affected by any other attention head. Each head has its output context vector, which is concatenated later.

An example of attention to images is shown in Fig. 4, as illustrated by the original authors of ViT.

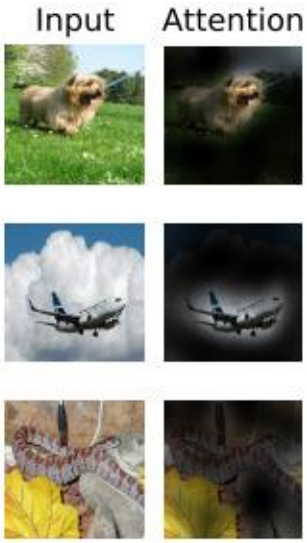

Figure 4: Output of attention block

Although transformers were introduced for NLP problems, they also got into computer vision and image recognition. Recently introduced, Vision Transformer has outperformed most popular benchmarked CNN architectures, including ResNet (BiT) [9]. ViT also acquires accuracy just a few points below ResNet on small datasets without significant regularization. When trained on large datasets (>14M images), this situation drastically changes.

The first step in training a ViT is to split the input image into smaller patches of fixed size. These individual patches are then linearly embedded. As commonly done in NLP, these patches can be considered as tokens for model input. These patches or tokens are fed into an encoder with self-managed attention. Finally, the context vector output is passed via a feed-forward network or a multi-layer perceptron for classification.

In this research, the size of the processed image was kept at (144,144), and the patch size was kept at (6,6). This split renders a total of 576 segments per image. The rate for patch extraction was fixed at [1,1,1,1], with size and strides as [1,6,6,1]. Fig. 5 describes how the raw image is broken into patches and passed through an encoder.

Now, these patches are linearly embedded row by row to form a sequential input vector or token layer. This operation can be considered similar to the "flatten" operation usually done after the last convolution layer in the case of a CNN. The operation starts from the first row and continues sequentially till the last row.

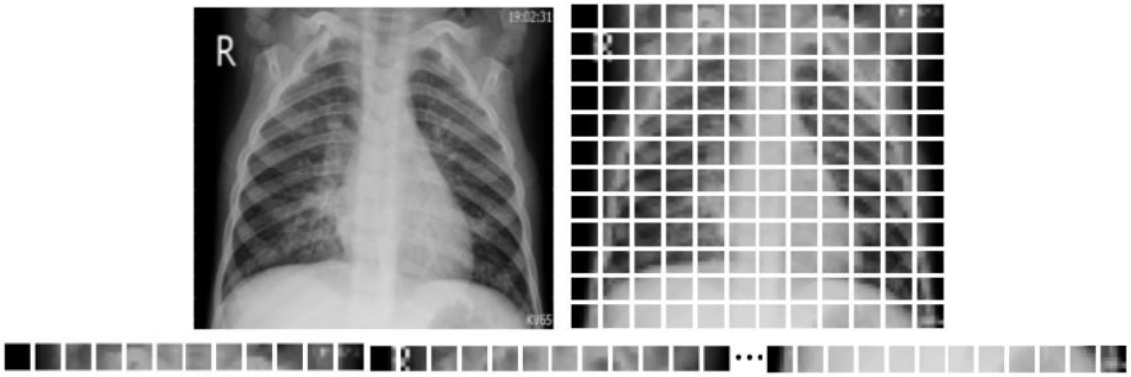

Figure 5: Preparing the model input by splitting the X-ray image into smaller patches, followed by flattening

In NLP, when the input string is split into smaller parts, which can be a collection of words or even characters, the process is called tokenization, and these individual splits are called tokens. In our case, the individual flattened patches are also considered tokens and are passed through an encoder. For the ViT model, we used 8 attention heads and 8 transformer layers. Before the transformer layer, the data augmentation and patch encoder layers were initialized (refer to the model structure in Fig. 6).

input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[]
data_augmentation (Sequential)	(None, 72, 72, 3)	7	['input_1[0][0]']
patches_1 (Patches)	(None, None, 108)	0	['data_augmentation[0][0]']
patch_encoder (PatchEncoder)	(None, 144, 64)	16192	['patches_1[0][0]']

Figure 6: Data augmentation and patch encoder layers

This, as mentioned, was followed by 8 transformer layers. Each layer had a fixed predefined layout used in a recurring fashion. A normalization layer was added at the start of every transformer block. Layer normalization [10] essentially normalizes the input along the features, unlike batch normalization, which normalizes features across the batch dimensions. This layer is followed by a multi-headed self-attention layer with 8 heads. Each multi-head attention layer was also given a drop rate of 0.2 to avoid any possible overfitting. Later, the patch encoder and attention layers were concatenated and followed by another normalization layer. Finally, a simple feed-forward network with 2 dense and 2 dropout layers was added. It consisted of 128 and 64 nodes in the dense layers, respectively, with a dropout rate of 0.1. Fig. 7 summarizes the layout of each transformer layer or block.

layer_normalization (LayerNorm alization)	(None, 144, 64)	128	['patch_encoder[0][0]']
multi_head_attention (MultiHea dAttention)	(None, 144, 64)	132672	['layer_normalization[0][0]', 'layer_normalization[0][0]']
add (Add)	(None, 144, 64)	0	['multi_head_attention[0][0]', 'patch_encoder[0][0]']
layer_normalization_1 (LayerNo rmalization)	(None, 144, 64)	128	['add[0][0]']
dense_1 (Dense)	(None, 144, 128)	8320	['layer_normalization_1[0][0]']
dropout (Dropout)	(None, 144, 128)	0	['dense_1[0][0]']
dense_2 (Dense)	(None, 144, 64)	8256	['dropout[0][0]']
dropout_1 (Dropout)	(None, 144, 64)	0	['dense_2[0][0]']
add_1 (Add)	(None, 144, 64)	0	['dropout_1[0][0]', 'add[0][0]']

Figure 7: Layout of each transformer block

This particular layout was repeated until all defined transformer layers were covered. Finally, once all the transformer layers were done, one final layer normalization layer was added, followed by a flattened layer to be fed into another series of feed-forward layers. The MLP layer has 2 dense layers with 2048 and 1024 nodes, followed by the classification layers at the end with 2 nodes. A dropout layer was also added after each dense layer with a dropout rate 0.2. Fig. 8 summarizes the above-described layers and the total number of trainable features and parameters.

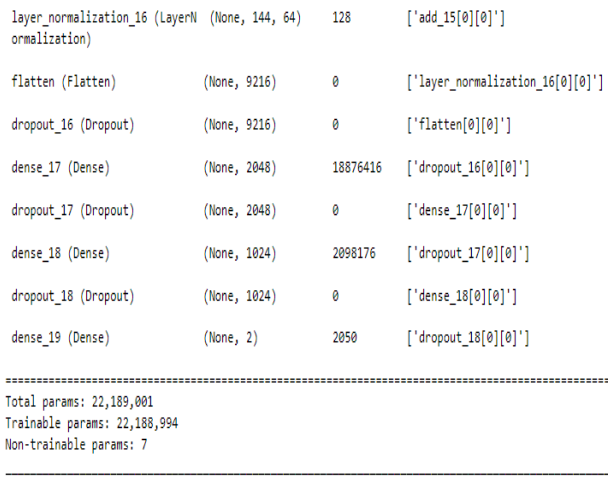

Figure 8: Feed-forward layer

Like any other deep learning architecture and millions of trainable parameters, there is always a risk of overfitting. We use model checkpointing, early stopping, and reducing learning rate on plateauing to prevent overfitting. The configuration for each of the following is described below:

- Checkpointing
- monitor='val_loss',
- mode='min',
- save_best_only=True
- Early stopping
- monitor='val_accuracy',
- min_delta=1e-5,
- patience=15
- Reducing LR on plateauing
- monitor='val_loss',
- factor=0.2,
- patience=10

For the final hyperparameters, we used the Adam optimizer with a learning rate of 0.001 and weight decay of 0.0001. The loss function was sparse categorical cross-entropy, and logits were retained. For validation and evaluation, sparse categorical and top-5 accuracy were used. Finally, the model was trained on a batch size of 16 and a total epoch 50.

For comparison, we also built a custom convolution network with similar hyperparameters. This CNN had a series of convolution layers with increasing features. A batch normalization and max pooling layer followed each layer. After all the feature extraction, the feature map was flattened, followed by a dense layer with 128 nodes and an output layer with 2 nodes. A sample convolution block is shown in Fig. 9.

```
conv2d_2 (Conv2D)           (None, 60, 60, 128)       73856

batch_normalization_1 (Batc  (None, 60, 60, 128)      512
hNormalization)

max_pooling2d_2 (MaxPooling  (None, 30, 30, 128)      0
2D)

dropout_1 (Dropout)         (None, 30, 30, 128)       0
```

Figure 9: Sample convolution block in CNN

4. Results and Analysis

As expected, ViT outperformed the CNN model in terms of overall model performance. ViT yielded a maximum validation accuracy of 98.18% and a training accuracy of 94.65%. CNN yielded a maximum training accuracy of 98.75% but performed very poorly on validation data, with a validation accuracy of only 82.37%. The graphs in Fig. 10 visualize the trend of model accuracy with the number of epochs for ViT and the model loss with epochs.

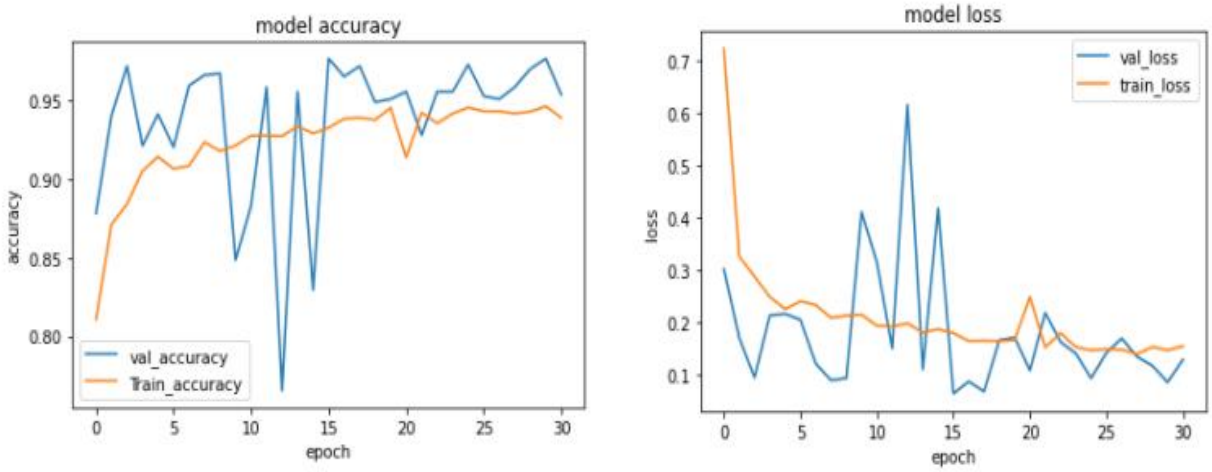

Figure 10: Performance evaluation of ViT

We can see some underfitting for ViT. This is possible when the model has few parameters or isn't complex enough to fit the training data correctly. This can be improved by having more attention heads or transformer blocks and further tuning it. Although the number of parameters for ViT already seems too much (22M), the computation and resources required to train a transformer are much less than a CNN. Hence, further increasing the parameters shouldn't drastically affect the training time and computation resources required. Similar metrics for the CNN model are visualized in Figure 11.

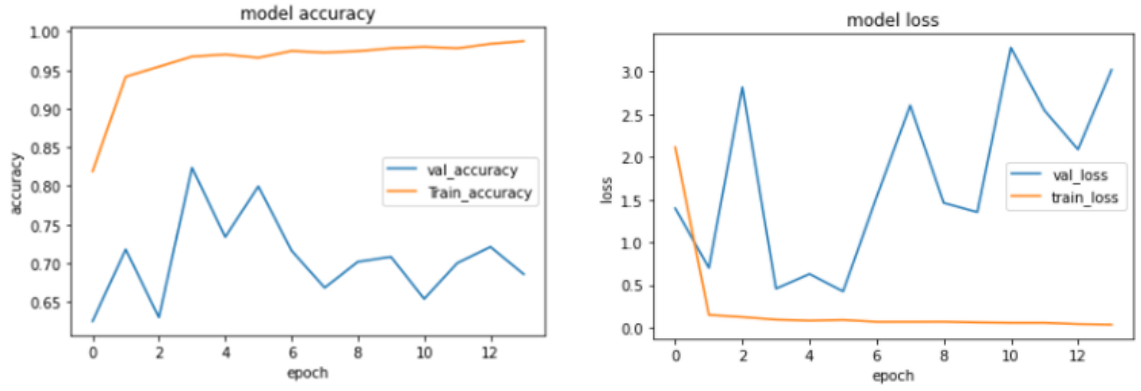

Figure 11: Performance evaluation of CNN

For the CNN model, we can see a large amount of overfitting. This happens because despite having one-third of the trainable parameters compared to the ViT model, the CNN model was too complex for the training data. Vision transformer performs better than the conventional CNN due to reliance on only attention instead of convolutions. The ability of the self-attention module to identify the segments of the input image which require the highest attention outperforms the traditional feature extraction via maintaining a features map using convolutions.

5. Conclusion

Pneumonia has several types and effects and is a common yet severe disease. Improvement in its detection mechanism will always be helpful. In this study, ViT has been efficiently implemented to detect Pneumonia from chest X-rays, scans of which were collected from various sources (like Kaggle and Github). The pipeline follows cleaning, processing, and splitting raw images into patches, which are then converted to linear embeddings for the encoder block. The linear and output layers proceeded with this block (consisting of the self-multi-head attention, addition, normalization, and feed-forward mechanism). Our approach outperformed all existing research and recorded a validation accuracy of 98.18% for Pneumonia detection. We also analyzed this model's performance against tuned CNN. ViT can eventually completely replace vanilla CNN due to its extremely high efficiency and low resource and time consumption. The scope of this paper can be expanded to enhance the model's performance further in terms of even better accuracy and less computation. For instance, specialized neural networks such as recurrent neural networks can be used in the sequential part instead of just vanilla-dense layers.

Acknowledgement: We are grateful to everyone who helped me write this.

Data Availability Statement: This study used online benchmark data in its investigation. This data is totally fresh as displayed here.

Funding Statement: No funding has been obtained to help prepare this manuscript and research work.

Conflicts of Interest Statement: The writers have not disclosed potential bias (s). This is brand new writing from the authors. The information used is cited and referenced appropriately.

Ethics and Consent Statement: All data collection was conducted after receiving approval from an institutional review board and the agreement of all participants.

References

1. A. Vaswani et al., attention is all you need, 31st Conference on Neural Information Processing Systems. Long Beach, CA, USA, 2017.
2. A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv [cs.CV], 2020.
3. E. Prina, O.T. Ranzani, & A. Torres, Community-acquired Pneumonia. The Lancet, Vol.386, no.9998, pp. 1097-1108, 2015.
4. A. Prayle, M. Atkinson, and A. Smyth, "Pneumonia in the developed world," Paediatr. Respir. Rev., vol. 12, no. 1, pp. 60–69, 2011.
5. K. Tyagi, G. Pathak, R. Nijhawan, and A. Mittal, "Detecting Pneumonia using Vision Transformer and comparing with other techniques," in 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, pp. 12–16, 2021.
6. D. Kermany, K. Zhang, and M. Goldbaum, "Labeled optical coherence tomography (oct) and chest x-ray images for classification," Mendeley data, vol. 2,no.2, 2018.
7. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with Neural Networks," arXiv [cs.CL], 2014.
8. A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," Physica D, vol. 404, no. 132306, p. 132306, 2020.
9. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
10. R. Xiong et al., "On Layer Normalization in the Transformer Architecture," arXiv [cs.LG], 2020.
11. M. D. K. Hasan et al., "Deep learning approaches for detecting pneumonia in COVID-19 patients by analyzing chest X-ray images," Math. Probl. Eng., vol. 2021, pp. 1–8, 2021.

12. R. Jain, P. Nagrath, G. Kataria, V. Sirish Kaushik, and D. Jude Hemanth, “Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning,” Measurement (Lond.), vol. 165, no. 108046, p. 108046, 2020.
13. K. El Asnaoui, Y. Chawki, and A. Idri, “Automated methods for detection and classification pneumonia based on X-ray images using deep learning,” in Studies in Big Data, Cham: Springer International Publishing, pp. 257–284, 2021.
14. M. F. Hashmi, S. Katiyar, A. G. Keskar, N. D. Bokde, and Z. W. Geem, “Efficient pneumonia detection in chest x-ray images using deep transfer learning,” Diagnostics (Basel), vol. 10, no. 6, p. 417, 2020.
15. H. Wu, P. Xie, H. Zhang, D. Li, and M. Cheng, “Predict pneumonia with chest X-ray images based on convolutional deep neural learning networks,” J. Intell. Fuzzy Syst., vol. 39, no. 3, pp. 2893–2907, 2020.
16. E. Ayan and H. M. Unver, “Diagnosis of pneumonia from chest X-ray images using deep learning,” in 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 2019.
17. K. Hammoudi et al., “Deep learning on chest X-ray images to detect and evaluate pneumonia cases at the era of COVID-19,” J. Med. Syst., vol. 45, no. 7, p. 75, 2021.
18. A. U. Ibrahim, M. Ozsoz, S. Serte, F. Al-Turjman, and P. S. Yakoi, “Pneumonia classification using deep learning from chest X-ray images during COVID-19,” Cognit. Comput., pp. 1–13, 2021.
19. A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, “Identifying pneumonia in chest X-rays: A deep learning approach,” Measurement (Lond.), vol. 145, pp. 511–518, 2019.
20. J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, “Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study,” PLoS Med., vol. 15, no. 11, p. e1002683, 2018.
21. A. Vats, R. Singh, R. K. Khurana, and S. Jain, “A robust system for detection of pneumonia using transfer learning,” in Mobile Radio Communications and 5G Networks, Singapore: Springer Nature Singapore, pp. 667–678, 2022.
22. A. A. F. Alshadidi et al., “Investigation on the application of artificial intelligence in prosthodontics,” Appl. Sci. (Basel), vol. 13, no. 8, p. 5004, 2023.
23. E. Vashishtha, L. Sherman, T. Sajjad, N. Mehmood, Use of anti-viral therapy in treatment of Covid 19, Journal of Advanced Medical and Dental Sciences Research, Volume 8, Issue 11, Pages 273-276, 2020.
24. J. Solanki, “Prevalence of osteosclerosis among patients visiting dental institute in rural area of western India,” J. Clin. Diagn. Res., 2015.
25. K. Kaur et al., “Comparison between restorative materials for pulpotomised deciduous molars: A randomized clinical study,” Children (Basel), vol. 10, no. 2, p. 284, 2023.
26. M. J. Saadh et al., “Advances in mesenchymal stem/stromal cell-based therapy and their extracellular vesicles for skin wound healing,” Hum. Cell, vol. 36, no. 4, pp. 1253–1264, 2023.
27. M. Munshi, K. N. Tumu, M. N. Hasan, and M. Z. Amin, "Biochemical effects of commercial feedstuffs on the fry of climbing perch (Anabas testudineus) and its impact on Swiss albino mice as an animal model," Toxicology Reports, vol. 5, pp. 521-530, 2018.
28. M. Munshi, M. H. Sohrab, M. N. Begum, S. R. Rony, M. A. Karim, F. Afroz, and M. N. Hasan, "Evaluation of bioactivity and phytochemical screening of endophytic fungi isolated from Ceriops decandra (Griff.) W. Theob, a mangrove plant in Bangladesh," Clinical Phytoscience, vol. 7, article no. 81, 2021.
29. M. Munshi, M. N. H. Zilani, M. A. Islam, P. Biswas, A. Das, F. Afroz, and M. N. Hasan, "Novel compounds from endophytic fungi of Ceriops decandra inhibit breast cancer cell growth through estrogen receptor alpha in in-silico study," Informatics in Medicine Unlocked, vol. 32, p. 101046, 2022.
30. M. S. Valli and G. T. Arasu, An Efficient Feature Selection Technique of Unsupervised Learning Approach for Analyzing Web Opinions. 2016.
31. M. Senbagavalli and G. T. Arasu, “Opinion Mining for Cardiovascular Disease using Decision Tree based Feature Selection,” Asian J. Res. Soc. Sci. Humanit., vol. 6, no. 8, p. 891, 2016.
32. M. Senbagavalli and S. K. Singh, “Improving patient health in smart healthcare monitoring systems using IoT,” in 2022 International Conference on Futuristic Technologies (INCOFT), 2022.
33. M. Z. Amin, K. N. Tumu, M. Munshi, Y. N. Jolly, and M. M. Rahman, "Assessment of Heavy Metal Contents in Commercial Feedstuffs and Broiler (Gallus domesticus) Meat and Its Impact on Swiss Albino Mice as an Animal Model," Agricultural Science Digest - A Research Journal, vol. 39, no. 2, pp. 149-155, 2019.
34. Md. N. Hasan, M. Munshi, M. H. Rahman, S. M. N. Alam, and A. Hirashima, "Evaluation of antihyperglycemic activity of Lasia spinosa leaf extracts in Swiss albino mice," World Journal of Pharmacy and Pharmaceutical Sciences, vol. 3, no. 10, pp. 118-124, 2014.