Research Project – Pneumonia Detection using Vision Transformers

1. Introduction

Acute pulmonary infection known as pneumonia can be brought on by bacteria, viruses, or fungi and affects the lungs, inflaming the air sacs and leading to pleural effusion, a condition where the lung is flooded with fluid. More than 15% of deaths in children under the age of five are attributed to it. Many cases of pneumonia occur in impoverished and emerging nations, where there is a shortage of medical resources, excessive population, pollution, and unclean environmental conditions. Therefore, preventing the disease from turning fatal can be greatly aided by early diagnosis and care. The diagnosis of lung disorders typically involves radiological evaluation of the lungs using computed tomography (CT), magnetic resonance imaging (MRI), or radiography (X-rays). Pneumonia can be distinguished from a healthy condition by the white patches on the pneumonic X-ray, known as infiltrates. The subjective variability of chest X-ray exams for the identification of pneumonia is however a concern. Therefore, the need for an automated pneumonia detection system exists.

It appears like we are traveling back in time to the 1900s during the Spanish influenza with the recent outbreak of COVID-19, also known as the coronavirus. A lethal virus called the coronavirus has killed hundreds of thousands of people in various parts of the world. The chance of getting more severe complications from the virus appears to be higher in older adults, persons with major underlying medical disorders, and people who have experienced pneumonia in the past.

Medical specialists from all over the world are working around the clock to cure patients and stop the virus from spreading due to increased death rates and a shortage of medical resources. The virus can cause pneumonia in severe forms, increasing the risk of death. In order for patients to obtain treatment in a timely way, especially in underdeveloped areas, it is essential to have rapid and accurate pneumonia detection. With the ongoing development of technology, it is now possible to employ tools built on deep learning frameworks to identify pneumonia from chest x-ray pictures. Here, the challenge would be to support the diagnosing procedure so that therapy can proceed more quickly and with better clinical results.

Deep learning is an essential component of artificial intelligence and is key to resolving many challenging computer vision issues. Convolutional neural networks (CNNs) are widely employed in deep learning models to solve a variety of picture categorization issues. But only when they are given a lot of data can these models work at their best. Such a large volume of labeled data is difficult to obtain for biomedical image classification challenges since it necessitates the expensive and time-consuming job of having professional doctors classify each image.Here, the vision Transformer is being used to classify the images for Pneumonia Detection.

The original Transformer idea has been expanded upon by the concept of Vision Transformer. It is merely the implementation of Transformer in the image domain with a small tweak to handle the various data modalities. A ViT specifically employs several tokenization and embedding techniques. The general architecture is the same, though. A source image is divided into a collection of image patches known as visual tokens. The visual tokens are incorporated into a collection of fixed-dimension encoded vectors. The transformer encoder network, which is essentially the same as the one in charge of processing the text input, is given the encoded vector along with the position of a patch in the image.

2. Vision Transformer

While using fewer computer resources for pre-training, Vision Transformer (ViT) outperforms convolutional neural networks (CNN) in terms of performance. When training on fewer datasets, Vision Transformer (ViT) shows a generally lesser inductive bias compared to convolutional neural networks (CNN), which increases reliance on model regularization or data augmentation (AugReg). The ViT is a visual representation of a transformer whose architecture was initially created for text-based operations. Like the series of word embeddings used when employing transformers to convert text to text, the ViT model represents an input picture as a set of image patches and predicts class labels for the image.

When trained on enough data, ViT displays an exceptional performance that surpasses that of a comparable state-of-the-art CNN while using 4 times fewer computational resources. These transformers are now also used on photos for image recognition tasks and have great success rates with NLP models. In contrast to ViT, CNN divides the photos into visual tokens. By dividing a picture into fixed-size patches, accurately embedding each one, and including positional embedding as an input to the transformer encoder, the visual transformer can change an image. Additionally, ViT models surpass CNNs in terms of accuracy and computing efficiency by roughly four times. ViT's self-attention layer makes it feasible to globally embed information throughout the entire image. In order to recreate the structure of the image, the model also learns from training data to encode the relative locations of the image patches.

The Multi-Head Self Attention Layer (MSP) layer concatenates all the attention outputs in the right dimensions as part of the transformer encoder. The several attention heads aid in the training of an image's local and global dependencies. A two-layer Gaussian Error Linear Unit is included in the Multi-Layer Perceptrons (MLP) Layer (GELU). Since Layer Norm (LN) does not introduce any additional dependencies between the training images, it is applied before each block. As a result, training time and performance are improved. Additionally, after every block, residual connections are added since they enable direct component flow across the network without passing through non-linear activations. The MLP layer implements the classification head in the context of image classification.

One of the fundamental building pieces of machine learning transformers is attention, and more specifically, self-attention. This computational primitive helps a network learn the hierarchies and alignments existing in the input data by quantifying paired entity interactions.

For visual networks to acquire greater resilience, attention has been shown to be a crucial component.

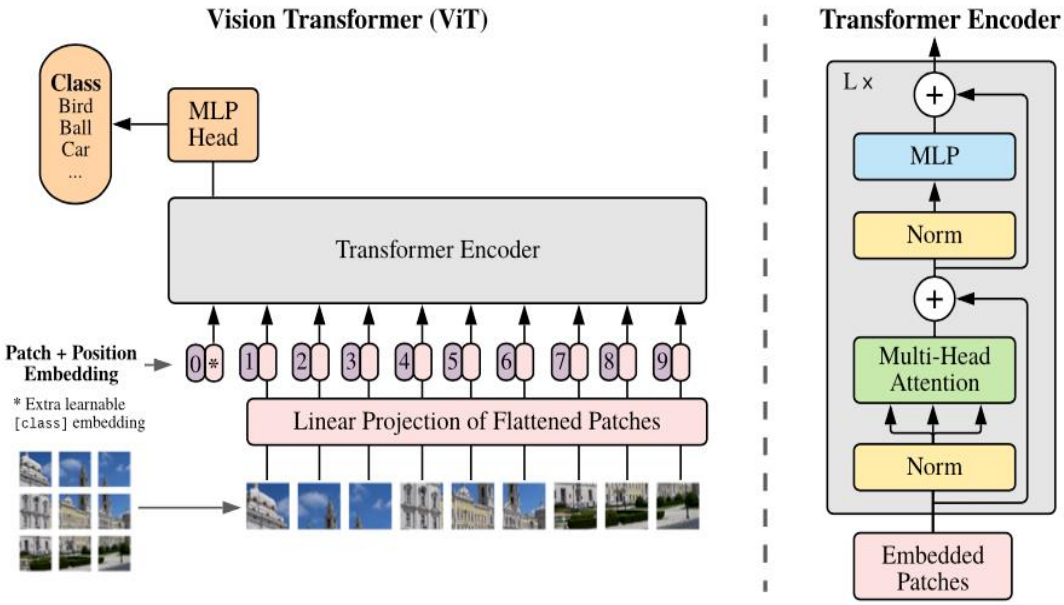

Fig.1. Vision Transformer

The following is a step-by-step breakdown of the vision transformer model's overall architecture:

- Create patches from an image (fixed sizes) the image patches
- Create flattened image patches
- Obtain lower-dimensional linear embeddings from the flattened patches.
- Include positional embeddings
- Input the sequence to a modern transformer encoder as an input.
- Pre-train the ViT model with image labels
- Fine-tune the downstream dataset for image classification

The dataset for Pneumonia detection is obtained for Kaggle. The data preprocessing and the splitting of the dataset into train and test set is done. The feature extraction is done by the ViTFeatureExtractor. The model is run for 30 epochs. Adam optimizer is used.

The Project is being deployed on a Django Framework, a python framework and the model is trained on ODU GPU servers using TensorFlow.

Dataset: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

The Dataset has 6000 Chest X-Ray HD Images. This pneumonia classification dataset has Symptoms include some combination of productive or dry cough, chest pain, fever and difficulty breathing.

Goal: The goal of the project is to identify the Pneumonia disease using chest X-ray image.

Main Steps involved:

1. Data gathering
2. Data Cleaning
3. Data Pre-Processing
4. Splitting Data into Train/Test/Val
5. Training the model
6. Testing the model
7. Deployment
8. Testing the model on Postman using Django Framework

Packages are needed to be used in project development -

1. python = ">=3.10,<3.11"
2. djangorestframework = "^3.14.0"
3. django-filter = "^22.1"
4. transformers = "^4.25.1"
5. numpy = "^1.23.5"
6. opencv-python = "^4.6.0"
7. Pillow = "^9.3.0"
8. tensorflow = "^2.11.0"

3. Results and Snapshots

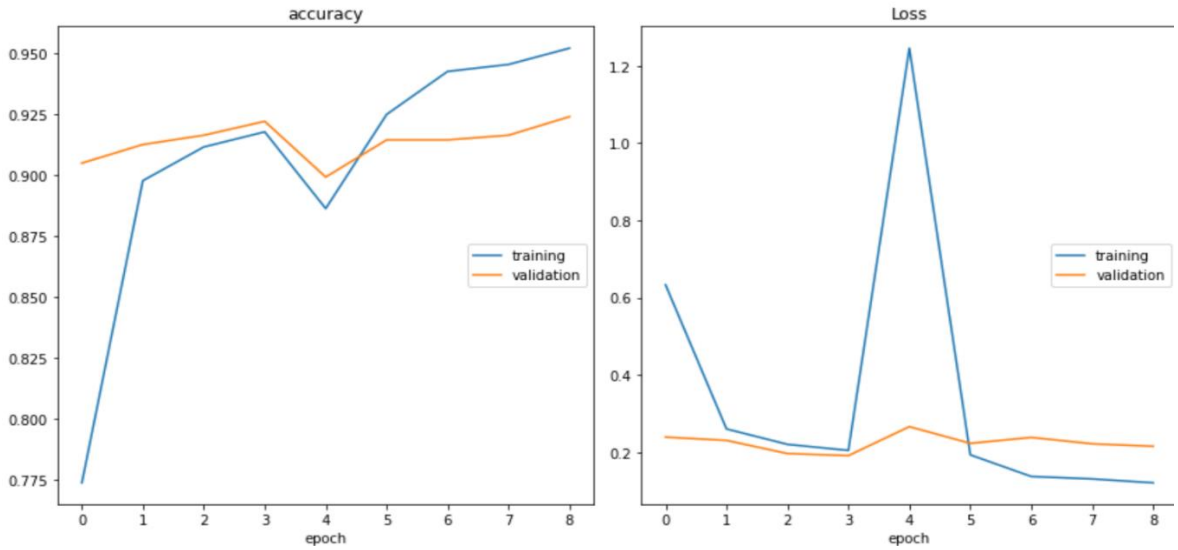

Fig.2. Plots of the accuracy and loss for the training and validation sets

:	Model	Precision	Recall	F1 Score	Accuracy
0	google/vit-base-patch16-224	0.922131	0.910931	0.916497	0.921905

Fig.3. Precision, Recall, F1 Score and accuracy of the Vision Transformer

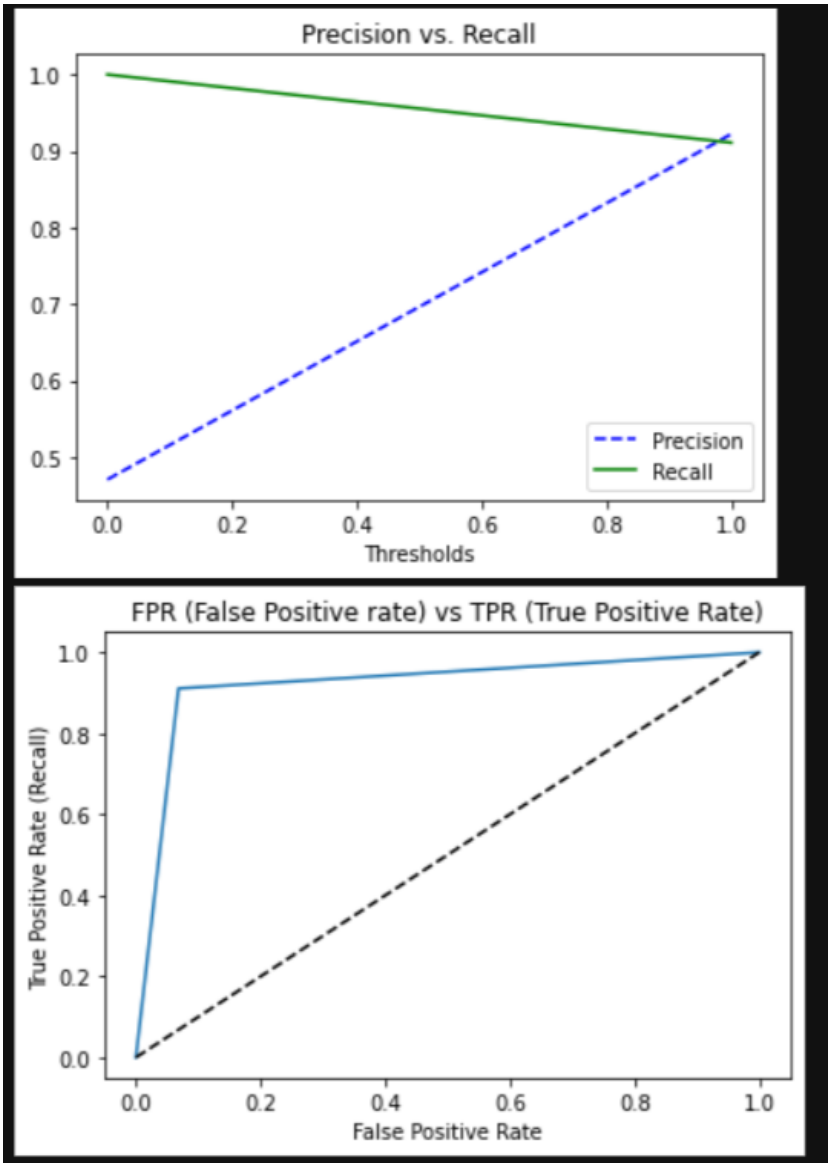

Fig.3. Graph for Precision vs Recall and FPR vs TPR of the Vision Transformer

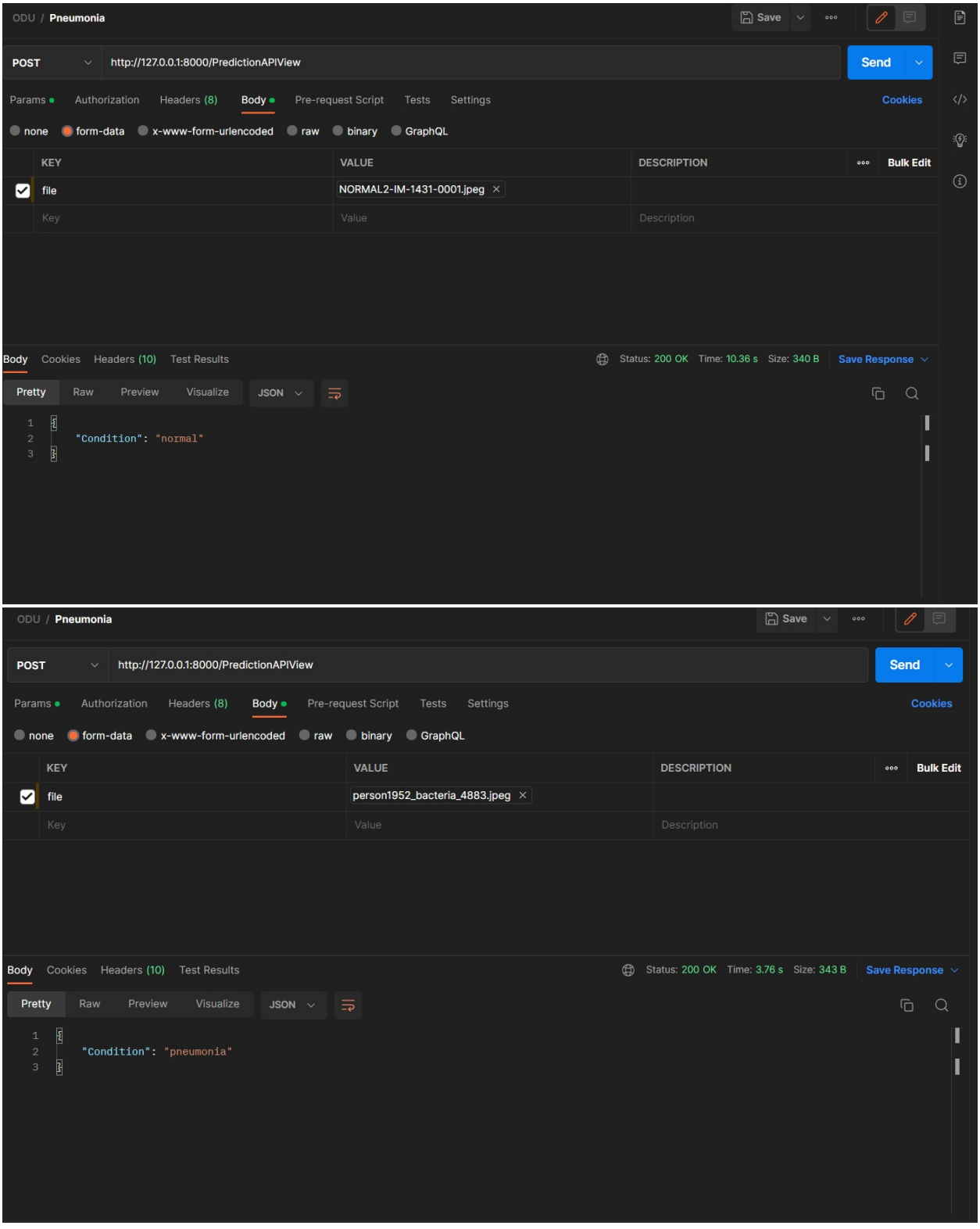

Fig.4. Results from Postman