

Computer Vision

Report 2

submitted to: sir Zeeshan khan

**Section: A2**

Muhammad Sarosh Tahir -F2021376098

**University of management and technology**

## Final Project Report: Comparative Analysis of CNN, ResNet50, and VGG16 Models for COVID-19 Image Classification

**1. Introduction**

The main purpose of this project is to compare the phase1 code with the phase 2 in which I do some fine tuning for increasing the accuracy and for this I add some layers and use some pretrained models to increased the accuracy that help me to predicted the images more accurately, Compare the performance of three different deep learning models—Convolutional Neural Network (CNN), ResNet50, and VGG16—in classifying chest X-ray images into three categories: COVID-19, Normal, and Viral Pneumonia. Additionally, a YOLO (You Only Look Once) model is used to detect regions of interest in images, and these regions are subsequently classified using the VGG16 model. A Flask-based web application is created to provide an interface for uploading and predicting the class of chest X-ray images using the trained models.

**2. Models Overview**

**2.1 Convolutional Neural Network (CNN)**

A custom CNN architecture is designed with the following layers:

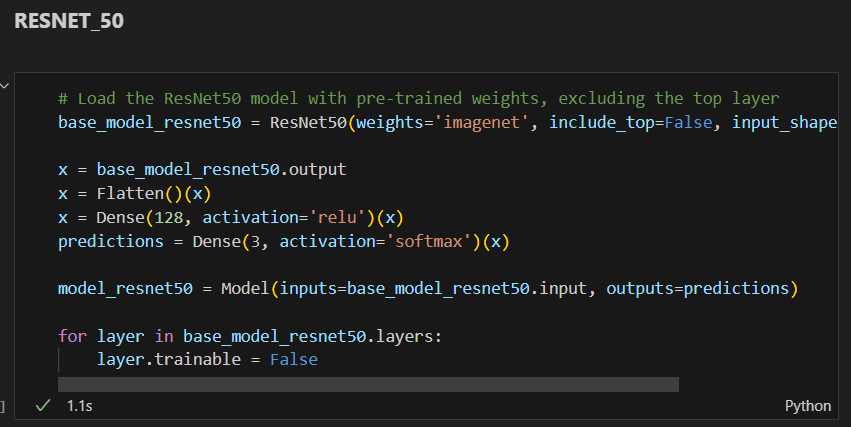
* Three convolutional layers followed by max-pooling layers.
* A flatten layer.
* Two dense layers with the final layer having three output nodes for the three classes.



**2.2 ResNet50**

ResNet50 is a pre-trained deep learning model from the ResNet family, which includes residual connections to help mitigate the vanishing gradient problem in deep networks. The top layer of ResNet50 is replaced with:

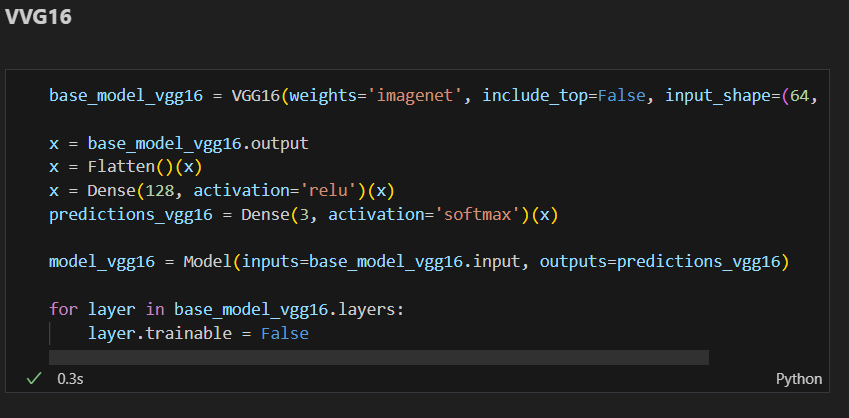
* A flatten layer.
* A dense layer with 128 nodes and ReLU activation.
* A final dense layer with three nodes and softmax activation.



**2.3 VGG16**

VGG16 is a pre-trained deep learning model known for its simplicity and depth, which includes 16 convolutional layers. Similar to ResNet50, the top layer of VGG16 is replaced with:

* A flatten layer.
* A dense layer with 128 nodes and ReLU activation.
* A final dense layer with three nodes and softmax activation.



**3. Methodology**

**3.1 Data Preprocessing**

Images are preprocessed using the ImageDataGenerator from Keras:

* Training images are augmented with rescaling, shear, zoom, and horizontal flip.
* Testing images are rescaled.

**3.2 Training**

Each model is trained on the same dataset of chest X-ray images with the following parameters:

* Optimizer: Adam
* Loss function: Categorical Crossentropy
* Metrics: Accuracy
* Epochs: 20 for CNN, 15 for ResNet50 and VGG16

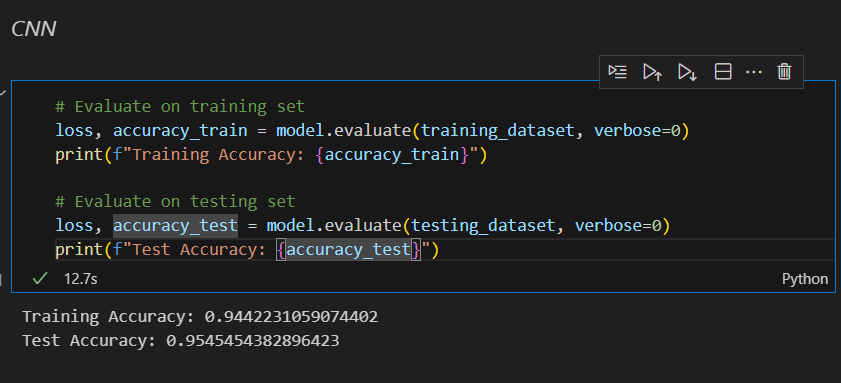
**3.3 Evaluation Metrics**

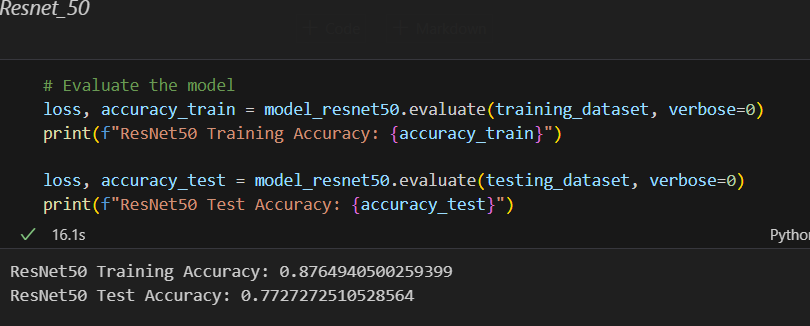
Models are evaluated using:

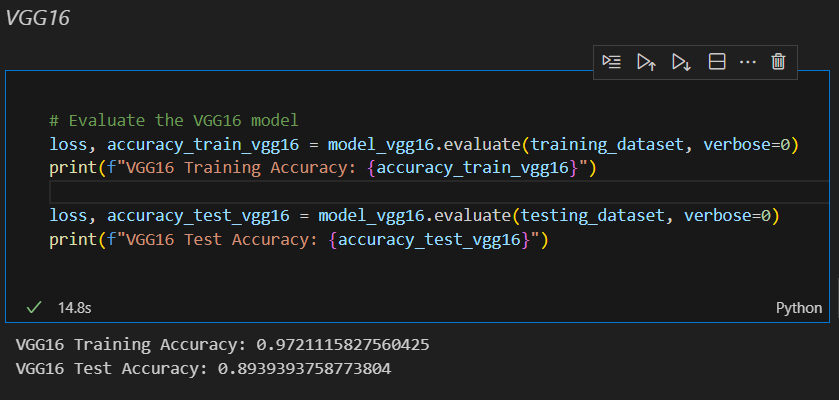
* Accuracy
* Precision
* Recall
* F1-score

**4. Results**

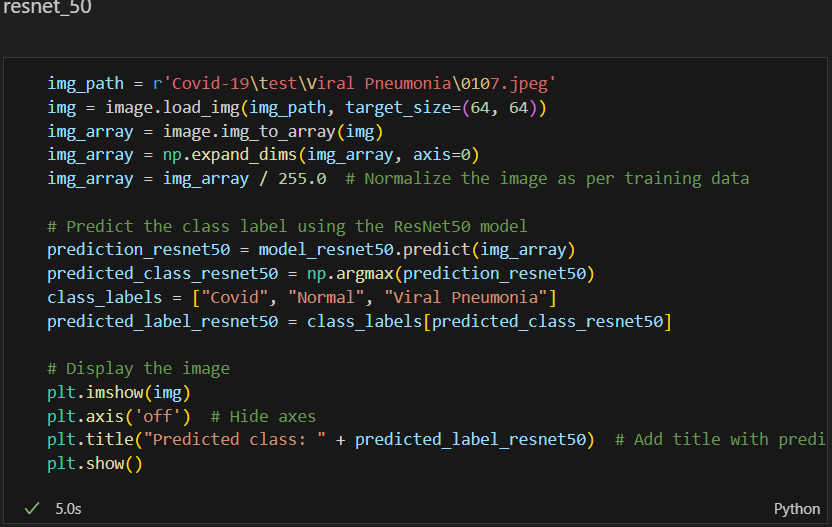
**4.1 Training and Testing Accuracy**

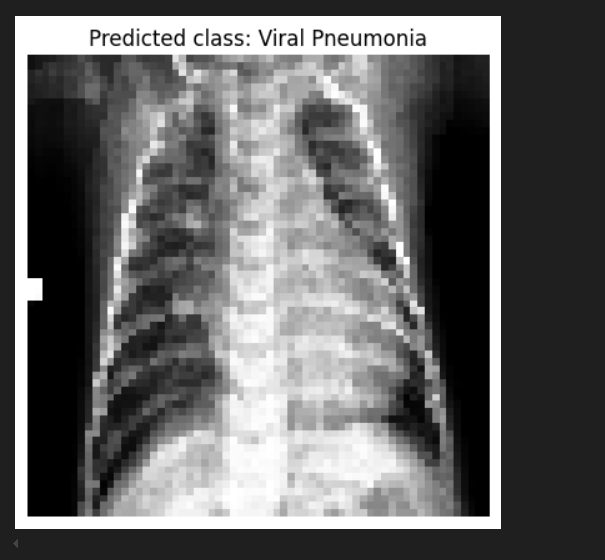
****

****

****

**4.3 Model Predictions**

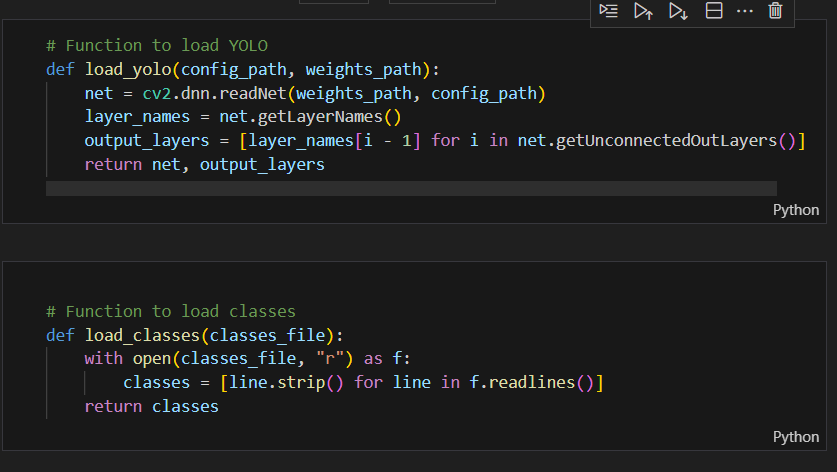
****

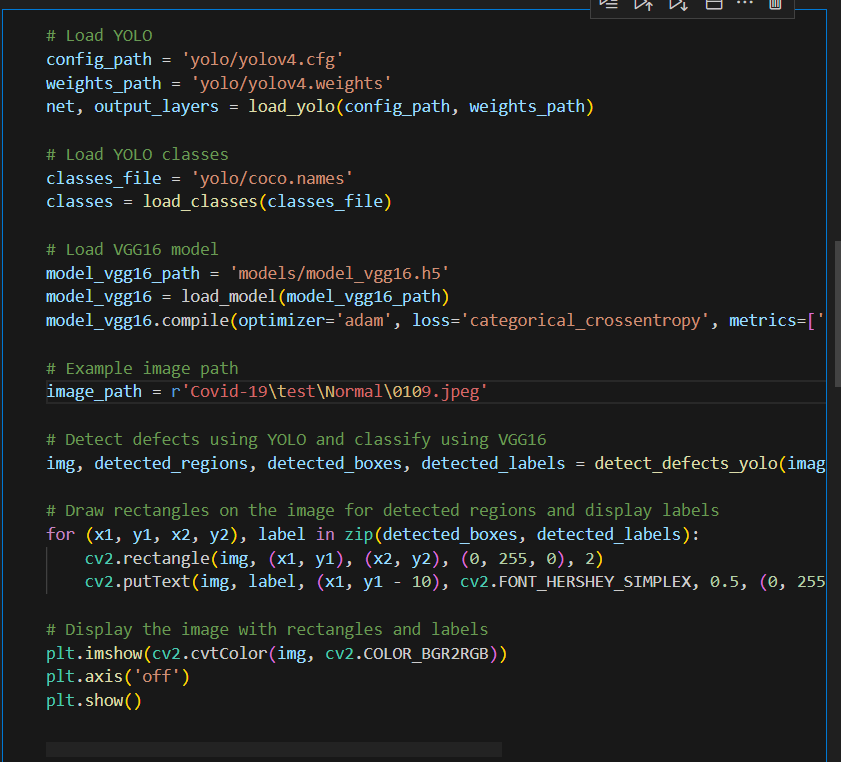
****

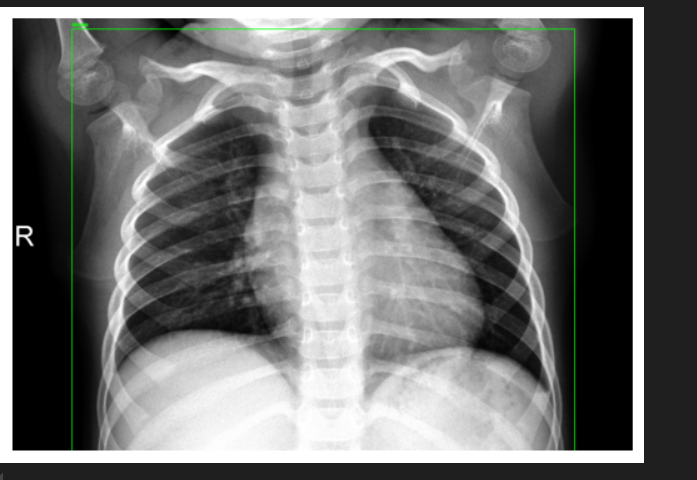
**5. YOLO and VGG16 Integration**

**5.1 YOLO Model**

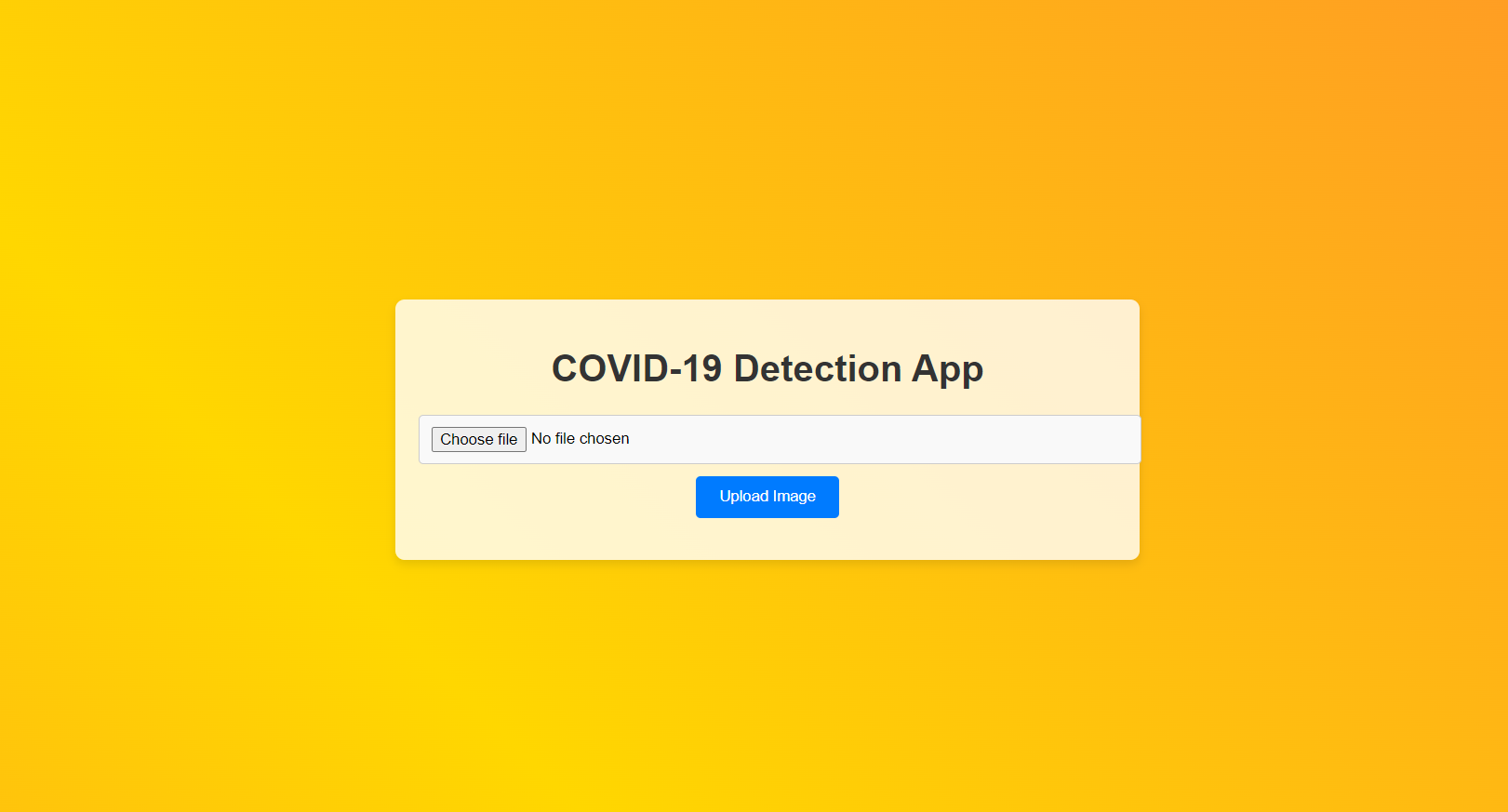
YOLO is used to detect regions of interest in chest X-ray images. The detected regions are then classified using the VGG16 model. The following function demonstrates the process:

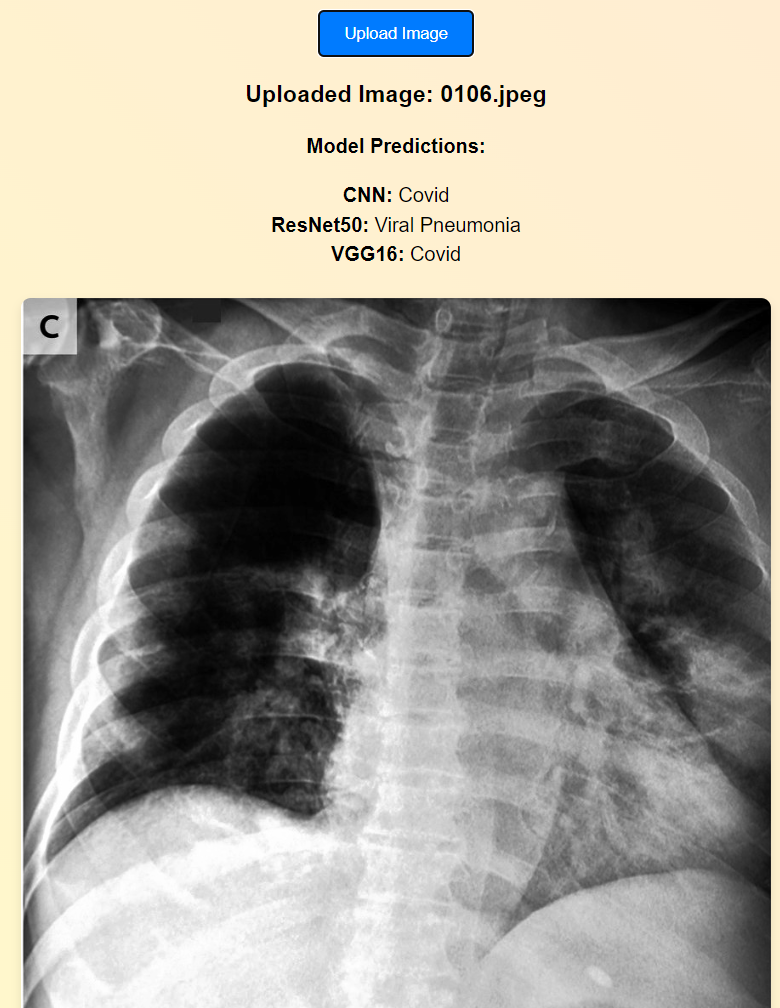






**6. Flask Web Application**

A Flask-based web application is developed to provide an interface for uploading and predicting the class of chest X-ray images. The application allows users to upload an image and get predictions from the three models (CNN, ResNet50, and VGG16). The following code snippet demonstrates the Flask app setup: 



**7. Conclusion**

This project successfully demonstrates the application of three deep learning models—CNN, ResNet50, and VGG16—in classifying chest X-ray images into COVID-19, Normal, and Viral Pneumonia categories. The results indicate that pre-trained models like ResNet50 and VGG16 outperform the custom CNN model. Additionally, the integration of YOLO for defect detection and subsequent classification using VGG16 enhances the robustness of the classification system. The Flask web application provides an easy-to