IT 351 project report

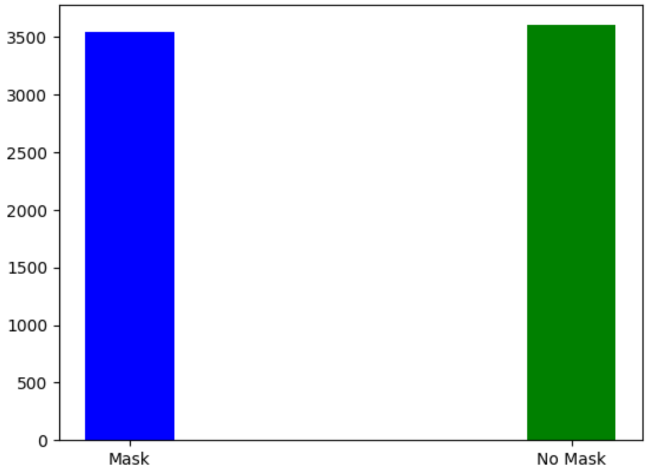
Name: Mohammed faleh almutiri

ID: 381117591

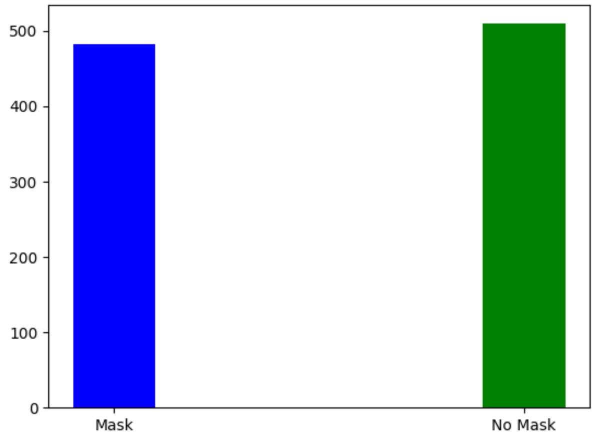
Introduction:

The following project is a computer vision project that aims to identify whether or not the person is wearing a mask, using a pre-trained resnet50 with fine tuning The following are the results.

Dataset introduction :



The dataset is divided into two classes with one being Masked and the other being not Masked (3539, 3589) respectively. Due to issues with the dataset such as being pre augmented I didn't use any part of it for testing as to not pollute the test with some of the train data, so I chose another data for the test, the test was augmented to (1000, 900).

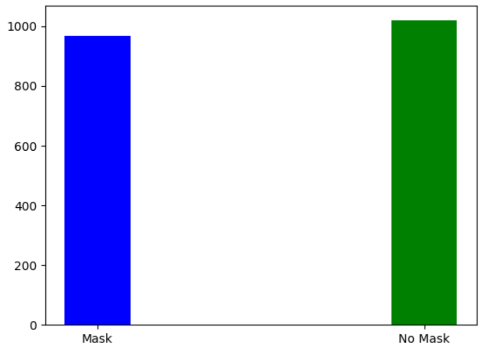


Augmentation strategy :

The train and test were augmented with the following augmentation list

1. image resize to 224,224.
2. random brightness change of 0.5.
3. Random Horizontal Flip with a chance 0.25.
4. Random Grayscale with a chance 0.1.
5. Random color Invert with a chance 0.1.
6. Random Equalize with a chance 0.1.
7. Random Posterize with a chance 0.1 and 8 bits.

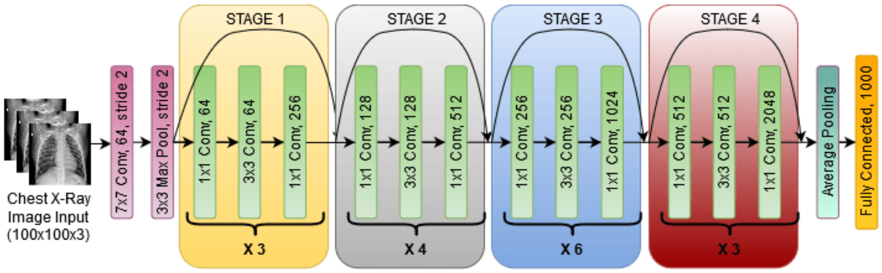
The test was doubled in size while the train size was left unchanged.



the test after augmentation

What is resnet50 and why pre-trained:

Resnet is a convolutional neural network design that has 50 layers. It uses skip connections or residual connections to solve the vanishing gradient issue in deep networks. Deeper networks can be trained more easily because of these connections, which enable information to cross through some levels. Each residual block in ResNet-50 has convolutional layers and short-cut connections. For improved computing performance, the architecture features bottleneck blocks. Global average pooling is used after the residual blocks, then fully linked layers are used for classification. ResNet-50 is frequently employed in a variety of computer vision tasks and has produced state-of-the-art results in picture categorization.



The best strategy for a limited dataset was to employ pretraining with fine-tuning. In order to capture broad features and trends, the model was pretrained on a larger dataset. With the exception of the initial and last layers, most layers were frozen during fine-tuning to preserve learnt representations. The model has five training epochs, which helped it adjust to the limited dataset. After then, all layers were unfrozen and the model was trained for 4 more epochs. With little data, this combined strategy improved the model's performance.

Results :

the model was evaluated using Scikit learn, below are the results which are the macro averages,

type	result
recall	96.679%
f1-score	96.672%
precision	96.666%
accuracy	96.673%

Real Time predictions:

The mask detection model was combined with a cv2 model ("haarcascade_frontalface_default.xml.") This cv2 model was created primarily to identify faces. Using a webcam, the system can now take real-time face images thanks to the integration. Once the face has been located, it is cropped and pre-processed before being sent into the ResNet model to determine whether or not the face is covered. The system can now scan live video streams and make real-time predictions regarding mask-wearing status, and fairly good results.

Libraries needed to run the project:

1. pytorch
2. numpy
3. matlibplot
4. sklearn
5. cv2
6. imblearn

Conclusion:

In this computer vision project, a mask detection system was created using a pre-trained ResNet50 model with fine-tuning. To increase diversity, the dataset's "Masked" and "Not Masked" classes were added. For image classification applications, the ResNet50 model, famous for its skip connections and bottleneck blocks, offered a strong basis. It was implemented to capture generic features and modify them for mask detection through pretraining with fine-tuning. High levels of accuracy, precision, recall, and F1-score were suggested by evaluation measures. The model was used with the facial recognition cv2 model to produce real-time predictions. In this study, mask-wearing identification is addressed using computer vision and pre-trained models, finally the model weights are provided under the name ‘model6-5firsttwo_4all_’ and can be loaded to test the model.