MINOR PROJECT

FACE MASK DETECTION

Under the guidance and supervision of :-

Dr . Ravi Bhushan Mishra

TEAM MEMBERS

- JAI MADAN (1906109)
- ANOOP DUSADH (1906108)
- AMIT KUMAR (1906091)

Table of Contents

- Introduction
- Objective
- Dataset
- Computing Method
- Software Tool
- Performance/ Accuracy
- Result Analysis
- Reference

INTRODUCTION

- According to the World Health Organization, one of the most effective protective measures against the COVID-19 virus is to wear a face mask when in public areas.
- However, monitoring people for face mask coverage becomes difficult in large groups and gatherings. Thus, Government have turned to the use of AI for detecting incorrect wear/absence of face coverings on individuals in public settings to facilitate contact-tracing efforts to track the spread of COVID-19 and to predict areas where outbreaks occur.

OBJECTIVE

- The focus for our project will be to design several models that determine whether or not an individual is wearing a mask, not wearing a mask, or wearing a mask incorrectly.
- We used the performance metrics gathered on these models (including accuracy, precision, recall, and f1 scores) to determine which model performs the best. The models explored include Decision Trees, CNN, Naive Bayes, and SVM.

DATASET

- We will use a mask (class label 2) v. no mask (class label 0) image dataset provided by Kaggle , which are already separated into two folders, mask and no mask. We will combine this dataset with another one provided by Cornell (00000-05000) incorrect wear of a mask (class label 3).

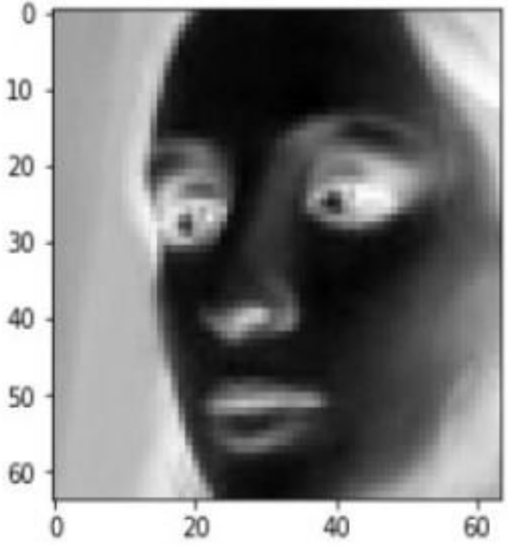
COMPUTING METHOD

- DATA PREPROCESSING
- NAIVE BAYES
- DECISION TREE
- SVM
- CONVOLUTIONAL NEURAL NETWORK

DATA PREPROCESSING

- In order to train our models with our dataset, we first had to normalize all the images before feeding them into our models.
- Normalizing our images meant that all images should be formatted in grayscale and all images should have the same shape.
- We choose grayscale because processing a grayscale image is three-four times faster than processing an RBG image.

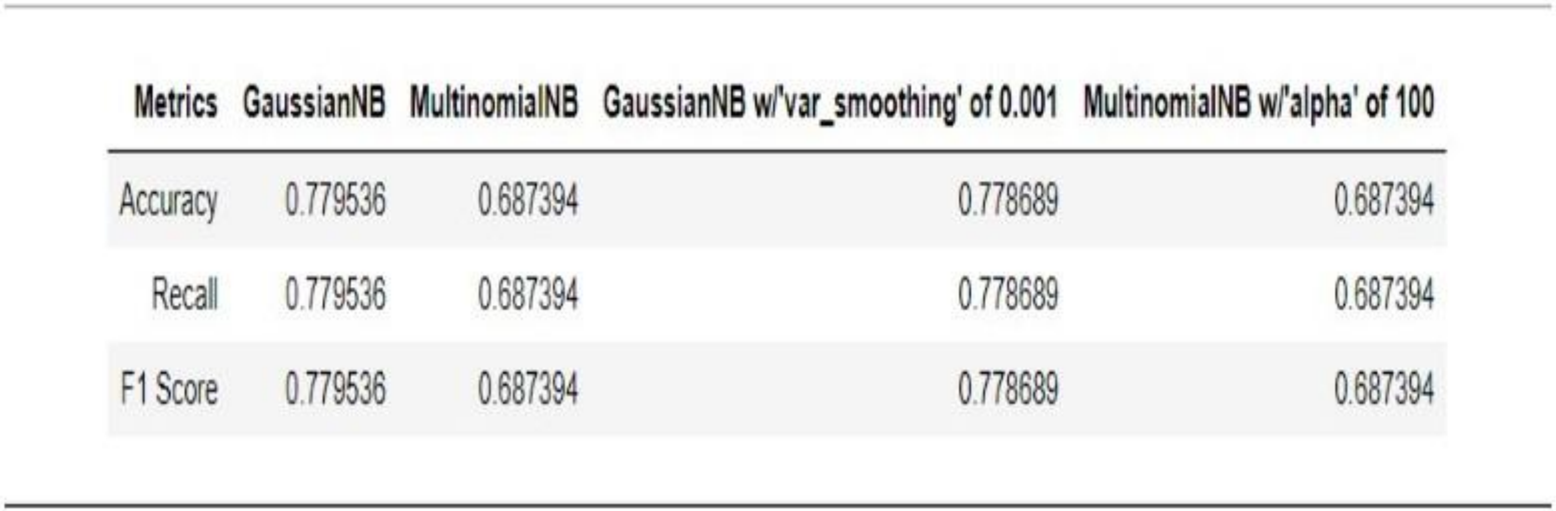
- As for the size of the image, we decided on a 64X64 because we felt the image was still readable and wouldn’t complicate our runtime.



NAIVE BAYES

- It is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- We made a prediction that this type of algorithm would not perform well because the naive bayes has the assumption that all features are conditionally independent.
- When it comes to image classification of any image, the majority of the pixels in the image are correlated to each other; however because the naive bayes algorithm works surprisingly well in natural language processing even when it's assumption doesn't hold, we decide to test it out.

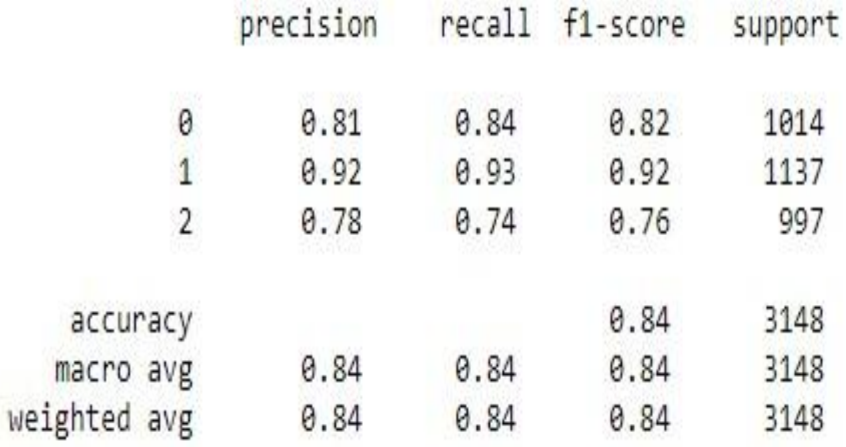
- We used Sklearn Gaussian and Multinomial naive bayes, with and without hypertuning.



Metrics	GaussianNB	MultinomialNB	GaussianNB w/'var_smoothing' of 0.001	MultinomialNB w/'alpha' of 100
Accuracy	0.779536	0.687394	0.778689	0.687394
Recall	0.779536	0.687394	0.778689	0.687394
F1 Score	0.779536	0.687394	0.778689	0.687394

DECISION TREE

- It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees

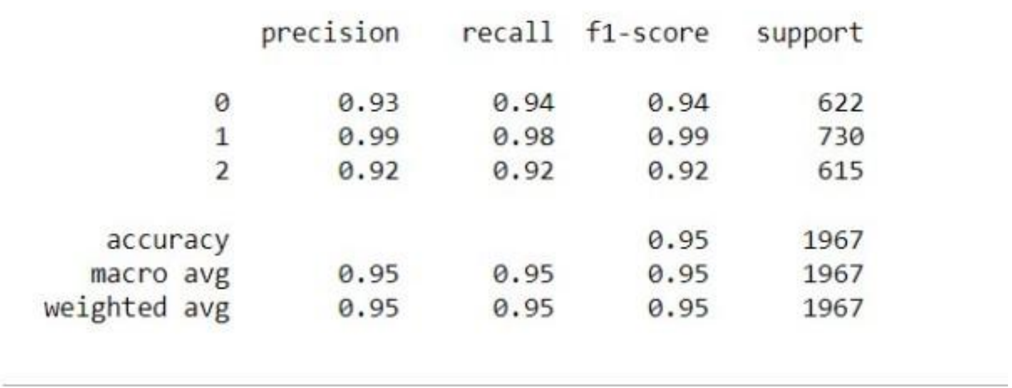


```
              precision    recall  f1-score   support

           0       0.81      0.84      0.82      1014
           1       0.92      0.93      0.92      1137
           2       0.78      0.74      0.76       997

    accuracy                           0.84      3148
   macro avg       0.84      0.84      0.84      3148
weighted avg       0.84      0.84      0.84      3148
```

SUPPORT VECTOR MACHINE

- SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.
- We explored SVM model because they are great for image classification and small to medium dataset
- Instead of using all samples, we used fifty percent of the samples to train our SVM

  model.



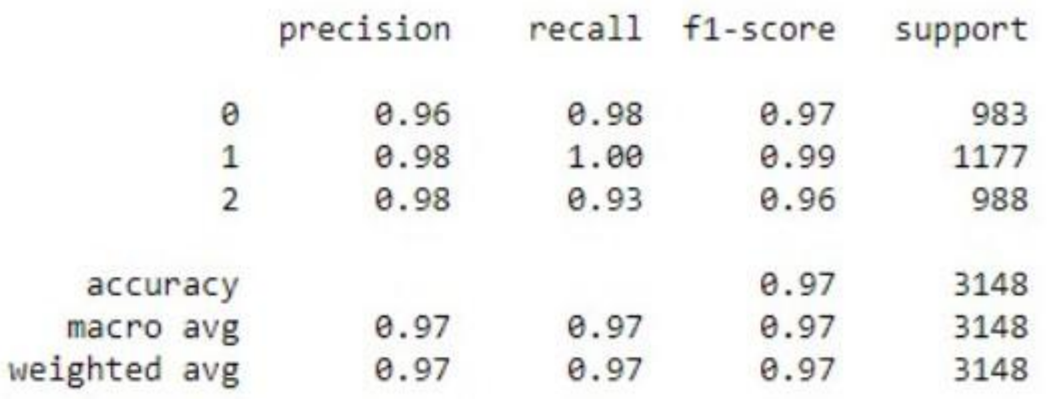
```
              precision    recall  f1-score   support

           0       0.93      0.94      0.94       622
           1       0.99      0.98      0.99       730
           2       0.92      0.92      0.92       615

    accuracy                           0.95      1967
   macro avg       0.95      0.95      0.95      1967
weighted avg       0.95      0.95      0.95      1967
```

CONVOLUTIONAL NEURAL NETWORK

- Convolutional neural networks are very popular in computer vision tasks such as identifying if there is a mask in the image.
- It is a neural network with additional layers in the front to extract certain features in the image.
- We used 2 convolutional layers and 2 max-pooling layers, first create the Sequential model and we will be adding layers into it.

- We are finished with the convolutional part and now we will start the neural network

- We used lower filter numbers in the convolution layers and the dropout layers are for the model to be less prone to overfitting therefore work better for general classification.



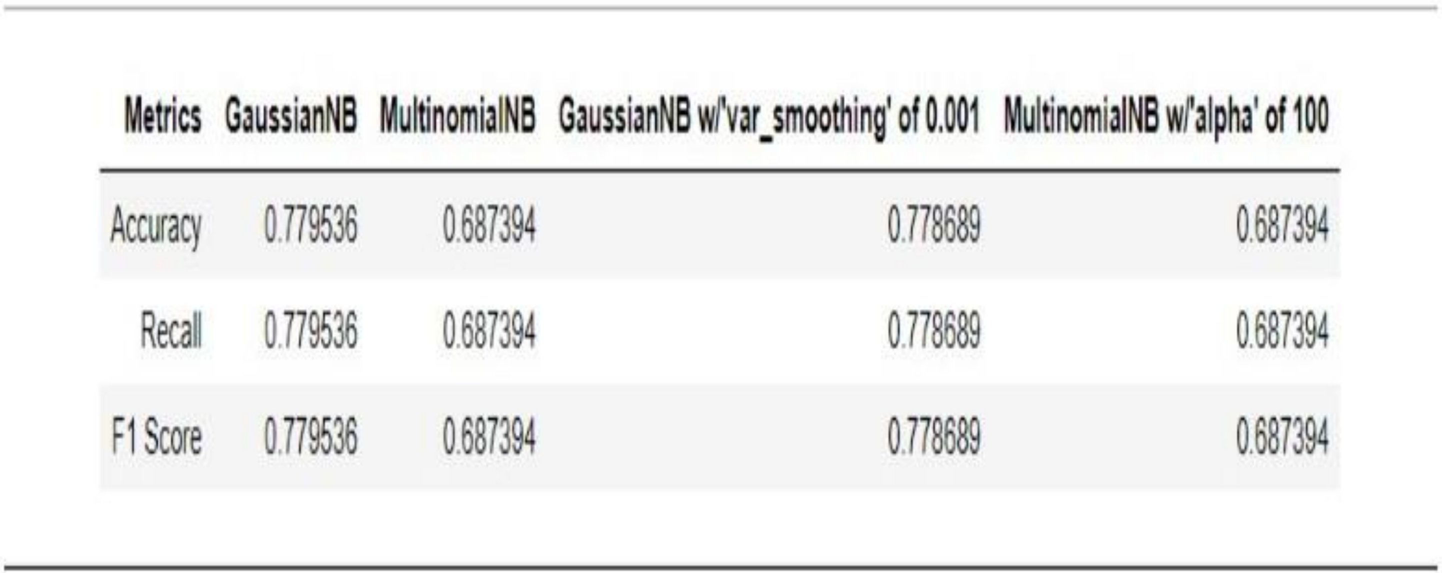
```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       983
           1       0.98      1.00      0.99      1177
           2       0.98      0.93      0.96       988

    accuracy                           0.97      3148
   macro avg       0.97      0.97      0.97      3148
weighted avg       0.97      0.97      0.97      3148
```

SOFTWARE TOOL

- **Jupyter Notebook :** is a web-based interactive computing platform. It's combines live code, equations, narrative text & visualizations.
- **Keras Library :** Keras is an open-source software library that provides a Python interface for artificial neural networks.
- **TensorFlow Library :** TensorFlow is a Python library for fast numerical computing & create deep learning module.
- **Opencv Library :** OpenCV is a tool for image processing and performing computer vision tasks.
- **Numpy Library :** NumPy can be used to perform a wide variety of mathematical operations on arrays.

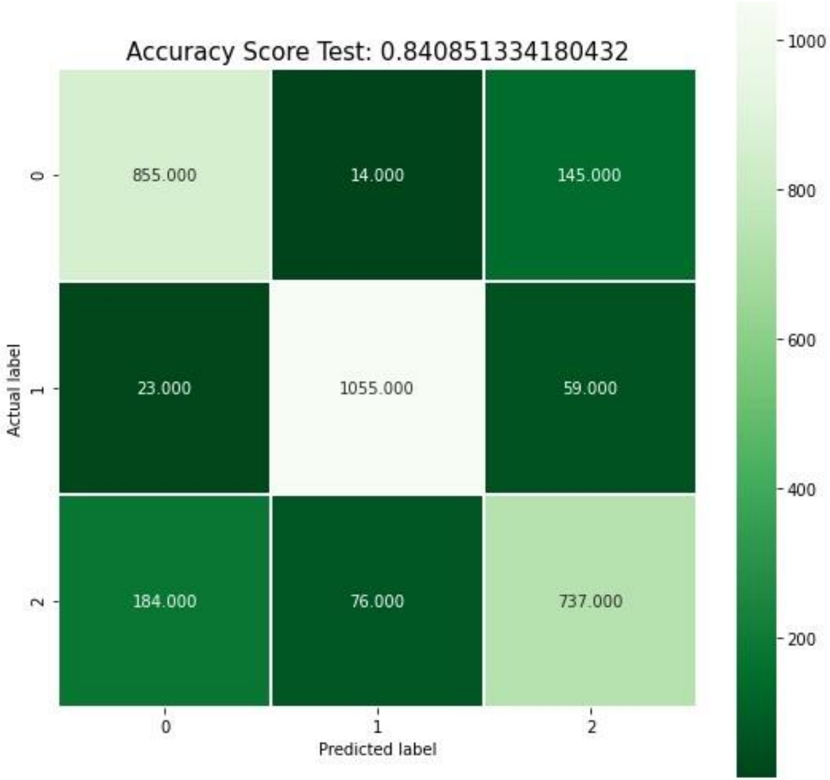
PERFORMANCE / ACCURACY

NAIVE BAYES

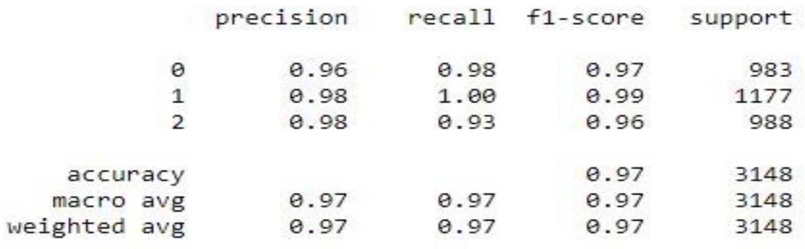


Metrics	GaussianNB	MultinomialNB	GaussianNB w/'var_smoothing' of 0.001	MultinomialNB w/'alpha' of 100
Accuracy	0.779536	0.687394	0.778689	0.687394
Recall	0.779536	0.687394	0.778689	0.687394
F1 Score	0.779536	0.687394	0.778689	0.687394

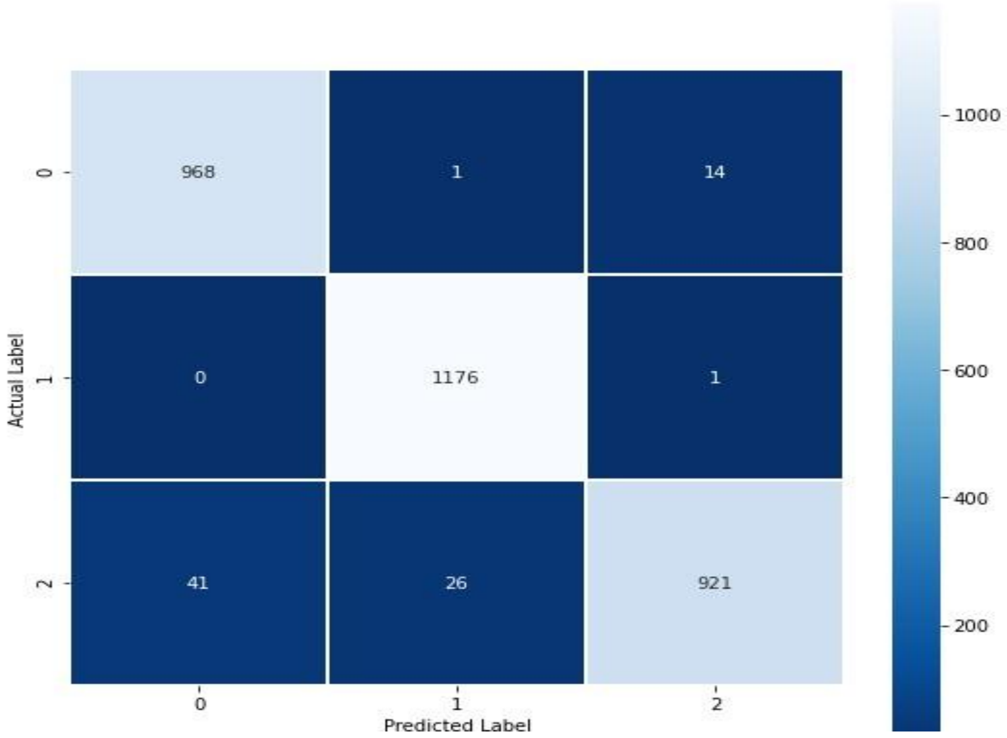
DECISION TREE



CNN



```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       983
           1       0.98      1.00      0.99      1177
           2       0.98      0.93      0.96       988

    accuracy                           0.97      3148
   macro avg       0.97      0.97      0.97      3148
weighted avg       0.97      0.97      0.97      3148
```



RESULT / ANALYSIS

Naive Bayes:-

It performed with 68.7% for multinomial and 77.8% for Gaussian.

Decision Tree :-

The optimal parameters resulted in 83.9% accuracy.

SVM :-

Reached an accuracy of 92.7% with our testing dataset.

CNN :-

An accuracy of 97% with our testing data.

- To sum up, our CNN model performed the best with a testing accuracy of ~97%.

REFERENCES

- **CNN Model** **https://ieeexplore.ieee.org/document/9342585/authors#authors**
- A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement : journal of the International Measurement Confederation , 167 , 108288. https://doi.org/10.1016/j.measurement.2020.108288
- How to Properly Wear a Face Mask: Infographic. Retrieved December 13, 2020, from https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/pro per-mask-wearing-coronavirus-prevention-infographic
- **Ensemble Learning :-https://www.kaggle.com/datasets/andrewmvd/face-mask-detection/code**

THANK YOU