Realtime Face Mask Classification

##### V V Prasanth G, 209309086, Batch-4

**Problem Statement:**

The COVID-19 pandemic has quickly disrupted global trade and travel, affecting our day-to-day lives. Face masks are now commonly worn for protection. Many public service providers will soon need consumers to properly wear masks to use their services. Face mask detection is now an essential responsibility to assist the global civilization. Your safety and the safety of others could perhaps be improved with the usage of the COVID-19 mask detector.

Additionally, categorizing the different types of masks that people use would improve our safety and preventative efforts.

**Meta-data of Data set:**

**-> Dataset is referred from:**

**(**<https://www.kaggle.com/datasets/bahadoreizadkhah/face-mask-types-dataset> **)**

* N-95                   (354-Train ; 50-Test)
* N-95v                  (390-Train ; 49-Test)
* Surgical              (342-Train ; 75-Test)
* Cloth                   (396-Train ; 78-Test)
* No Face Mask    (474-Train ; 78-Test)

**Define Project Objectives:**

Approach:

Using the help of the pretrained model ResNet50 and using machine learning tools like TensorFlow, Keras, and OpenCV is a straightforward method to accomplish this goal. And using Data Augmentation we can resize, rotate, etc. to the dataset so it can train in more real-life scenarios and splitting the dataset into 80% (for train) and 20% (for test). This technique properly identifies the face in the image and determines whether it is covered by a mask or not, as well as the type of mask that is covering it.

Result:

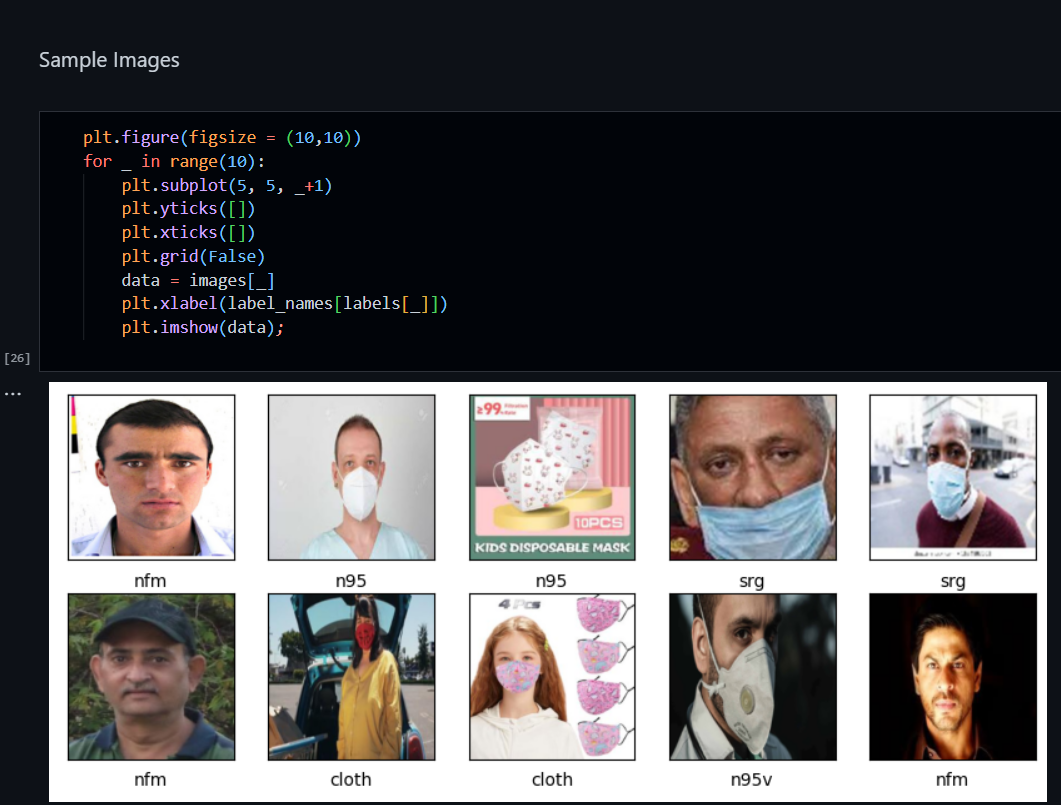
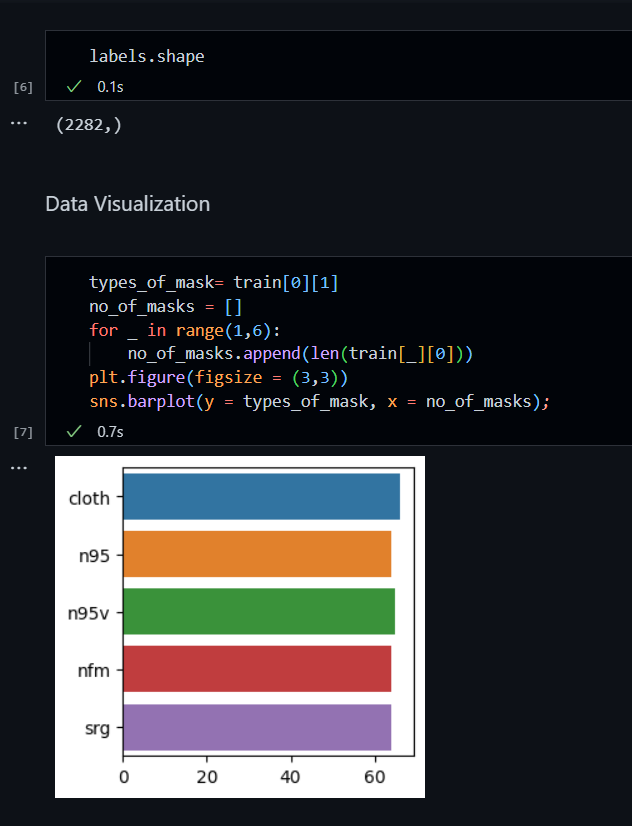
Building a robust solution that should detect and identify whether the person is wearing a mask or not and what type of mask it is. With a varying distance and color combination, it should work for any person.

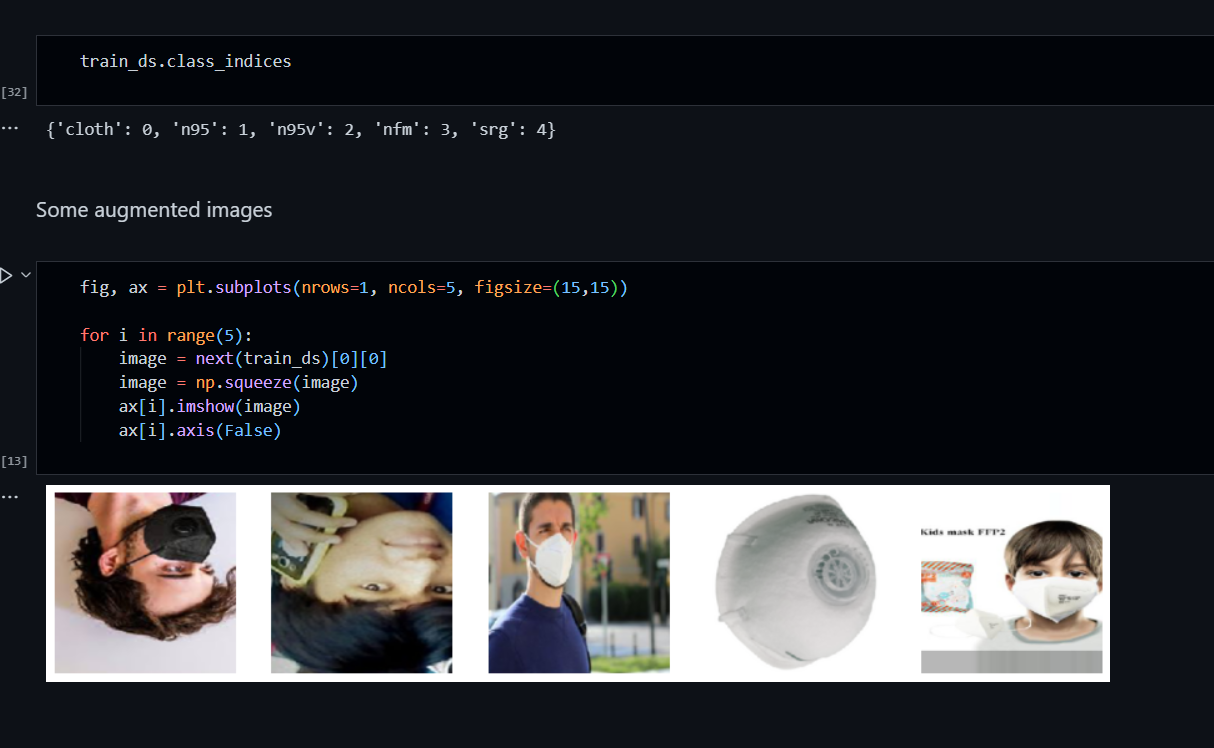
## Phase-1:

Pre-Processing and Data-Augmentation:

Text

Description automatically generated



## Phase-2:

CNN:

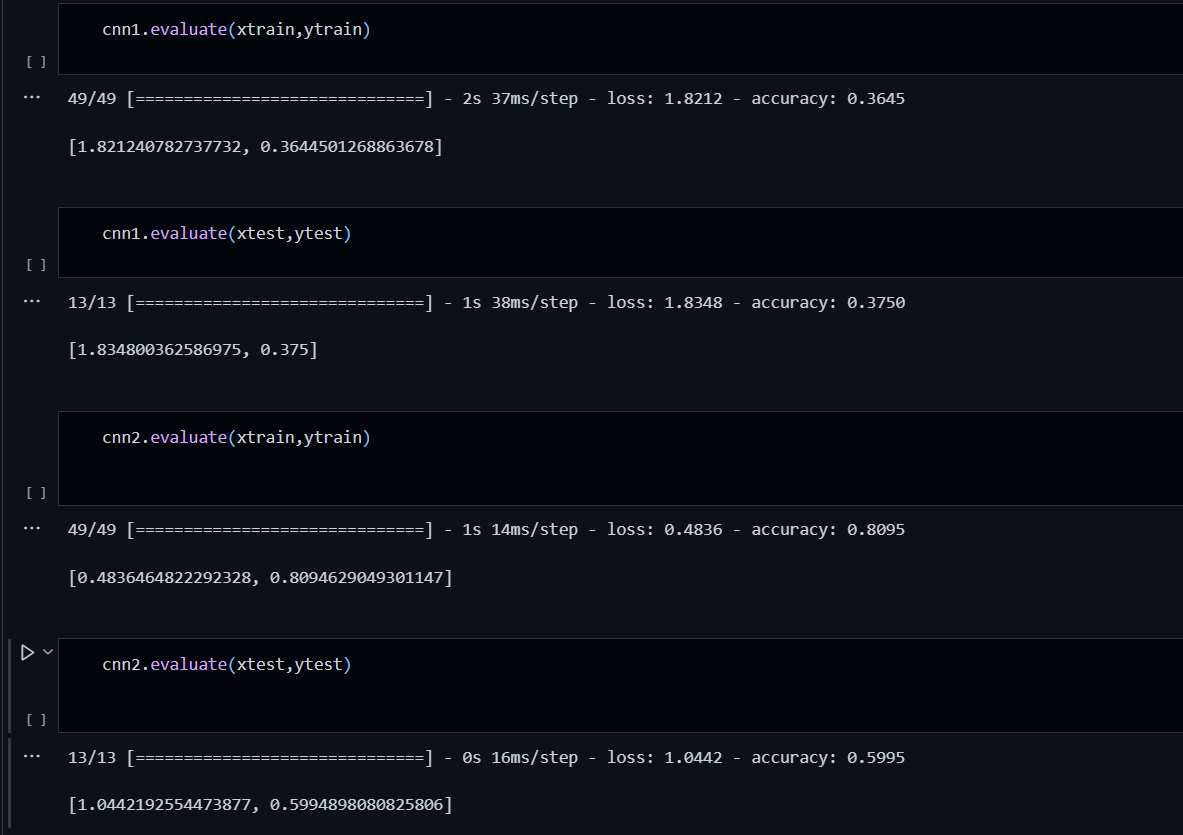
Convolutional neural networks (Conv-Nets or CNNs) are one of the main types of neural networks used to identify and classify pictures. A few applications for CNNs include object identification, face recognition, and others. CNN's image classifiers process an input image and assign it to one of several categories (E.g., Dog, Cat, Tiger, Lion). Depending on the image resolution, computers interpret an input image as a collection of pixels. It will see h x w x d (h = Height, w = Width, d = Dimension) depending on the image resolution.

* Providing an input image to the convolution layer.
* Choose your parameters, apply filters in steps, and add padding if necessary. Apply ReLU activation to the matrix and perform convolution on the image.
* Reduce the size of the dimensionality through pooling.
* Convolutional layers can be added till you're happy.
* Feed the output into a fully linked layer after flattening it (FC Layer).
* Images are classified using an activation function and logistic regression with cost functions.

Since, it is the basic and starting approach for image classification model, I have created 2 different cnn models and one which is fully connected and have many layers; other with fewer layers and the accuracy are as follows:

|  |  |
| --- | --- |
| CNN-1 | CNN-2 |
| Text  Description automatically generated | Text  Description automatically generated with medium confidence |

Accuracies are as follows:



### Transfer Learning:

#### MobileNet-V2:

#### The goal of this model was to optimize accuracy while considering the constrained resources for an on-device or embedded application. It was one of a family of mobile-first computer vision models for TensorFlow. MobileNets are compact, low-latency, and low-power models that may be customized to accommodate different use cases' resource limitations. Like other well-known large-scale models, they can be expanded upon for classification, detection, embedding, and segmentation.

### ResNet50:

Convolutional neural network ResNet-50 has 50 layers total. From the ImageNet database, you can load a network that has already been trained on more than a million photos. A keyboard, mouse, pencil, and numerous animals are among the 1000 object categories that the pre-trained network can identify in photos. Consequently, the network has picked up detailed feature representations for a variety of images.

|  |  |
| --- | --- |
| ResNet50 | MobileNet-V2 |
|  |  |
| Training and Loss Accuracy graphs are as follows | Training and Loss Accuracy graphs are as follows |
|  |  |

* Chosen ResNet-50 on behalf of MobileNet since it performs better in terms of accuracies and even ram management is good in ResNet-50.

## Reasoning for hyperparameters:

* I chose early stopping used to avoid overfitting when training a learner with an iterative method, such as gradient descent.
* Data Augmentation since we have very few images to train and test for.

### Using haarcascade\_frontalface\_default.xml through OpenCV for Realtime

### face detection and classifying the type of mask Text Description automatically generated

### Text Description automatically generatedText Description automatically generated

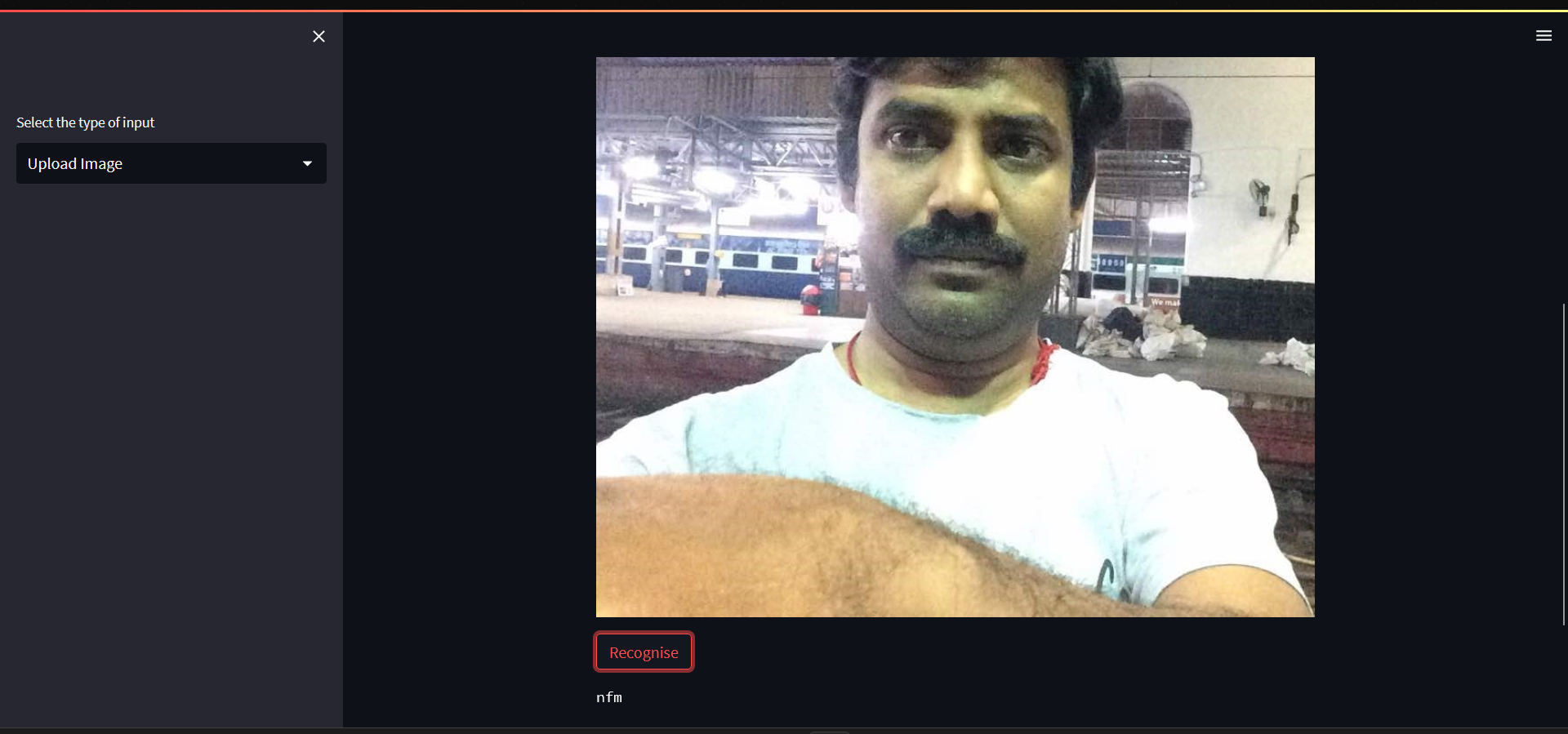
## Results:

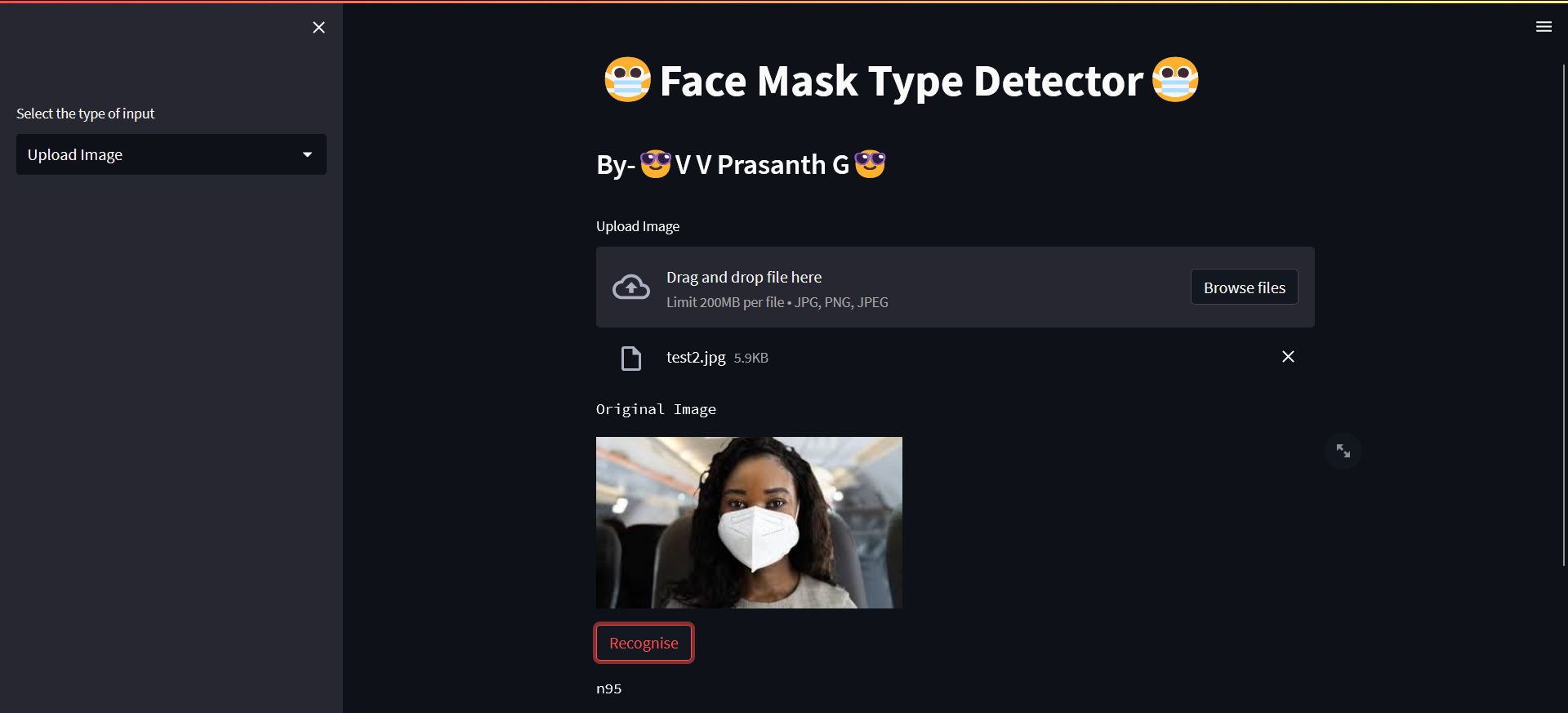
|  |  |  |
| --- | --- | --- |
|  | A person wearing glasses  Description automatically generated with medium confidence |  |

## Used Streamlit for deploying the model:



### Upload Image:





A screenshot of a computer

Description automatically generated with medium confidence

### Take a Shot(Live Image Processing):



