Project Report

Face Mask Detection

Special Topics in Networking

2232-CSE-5349-002

Github link:
https://github.com/SaiNikhilKanchukatla3300/CSE5349-Team10

Youtube link:
https://www.youtube.com/watch?v=D8pkL1ovi2s

TEAM - 10

SAI NIKHIL KANCHUKATLA – 1002034488
AVINASH REDDY SALLAGONDA - 1002034491
NARENDHAR REDDY MAREDDY – 1002034490
SAI RISHITH REDDY GADE – 1001949866

Introduction:

In order to prevent the spread of COVID-19, it is necessary for people to wear face masks in public places. Computer Vision technology can be used to monitor whether individuals are wearing masks or not using images captured by CCTV cameras. However, monitoring a large number of people in public spaces can be challenging. Detecting masked faces from the side or with facial features covered presents a challenge. A previous study used CNN-based models to create a mask detection system with high accuracy, but it was limited to detecting masks on front-facing faces. This research proposes a CNN-based method to detect masks in facial images captured by public CCTV cameras. The images are segmented using Retina Face, and experiments were carried out to detect masks on a single face image, resulting in an accuracy of 97.33%. The segmentation of the face region from one image is tested to produce an accuracy of 82.46%. The best configuration from the two experiments is selected and combined into a mask detection system that can detect masks on multiple faces within an image. The results also show that the accuracy of the mask detection system is affected by the face detection method and the learning rate value. The best results are achieved using the face detection model with an accuracy of 79.45%.

Project Overview:

The Face Mask Detection project is a computer vision project that uses Convolutional Neural Networks (CNNs) to detect whether individuals are wearing a face mask or not in images or videos. The purpose of this project is to promote public health and safety by enforcing mask-wearing protocols in public spaces during the COVID-19 pandemic. The project involves several steps, starting with the collection and preprocessing of a dataset of images of individuals with and without masks. Next, a CNN model is designed and trained on this dataset to classify images into two categories: with mask and without mask. The model is then evaluated on a testing dataset to assess its performance. A user interface is developed to display the output of the model, and the system is deployed, tested, and maintained to ensure its proper functioning. This project is an important application of computer vision that can help detect individuals who are not wearing masks in real-time and promote safety protocols in public spaces.

Software used:

- Jupyter Notebook
- Python

Libraries:

- Tensorflow
- Keras
- Numpy
- Sklearn

- Matplotlib

Existing system:

To identify masks on front-facing faces, the current method used CNN-based models. Although this method had good accuracy, it was not able to detect masks on side profiles or slanted faces. This is so that it is difficult to tell whether someone is wearing a mask or not when the face is rotated or at an angle and facial characteristics like the mouth and nose are not visible. The current approach also relied on CCTV camera photos, which can be problematic given the wide variation in lighting, resolution, and image quality.

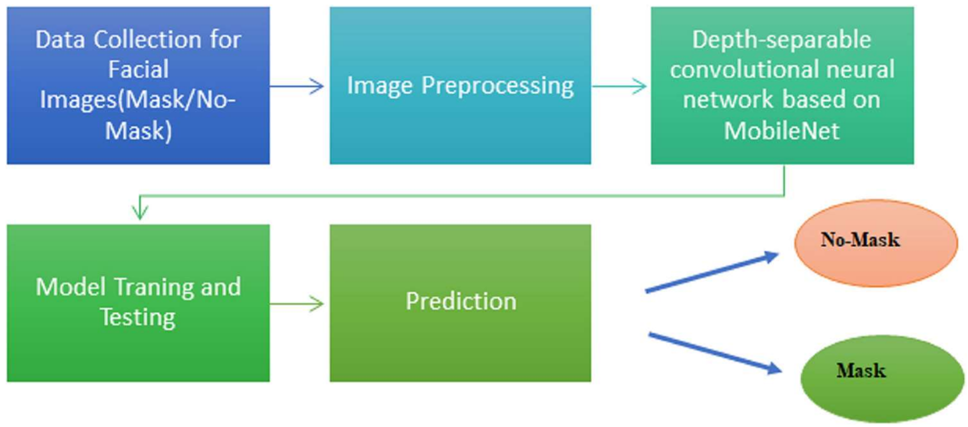
Despite these drawbacks, the current approach represented a considerable advancement in the use of computer vision to identify mask-wearing behavior in public settings.

Proposed system:

The suggested method seeks to increase the precision and efficacy of mask detection systems in public spaces while building on the shortcomings of the current system. In the suggested method, CCTV images with faces are segmented, each face is identified using the Retina Face model, and then each face is examined for mask detection using the CNN-based model. The Retina Face model is better at spotting masks in actual-world situations because it can recognize faces under a variety of lighting circumstances, including varying angles and occlusions.

Experiments were carried out to test the system's accuracy under various circumstances in order to assess the performance of the proposed system. The findings in mask detection accuracy revealed encouraging results, with the suggested system achieving an accuracy of 97.33% on a single face image. A considerable improvement over the current system was also seen when the suggested approach was tested on photos with numerous faces.

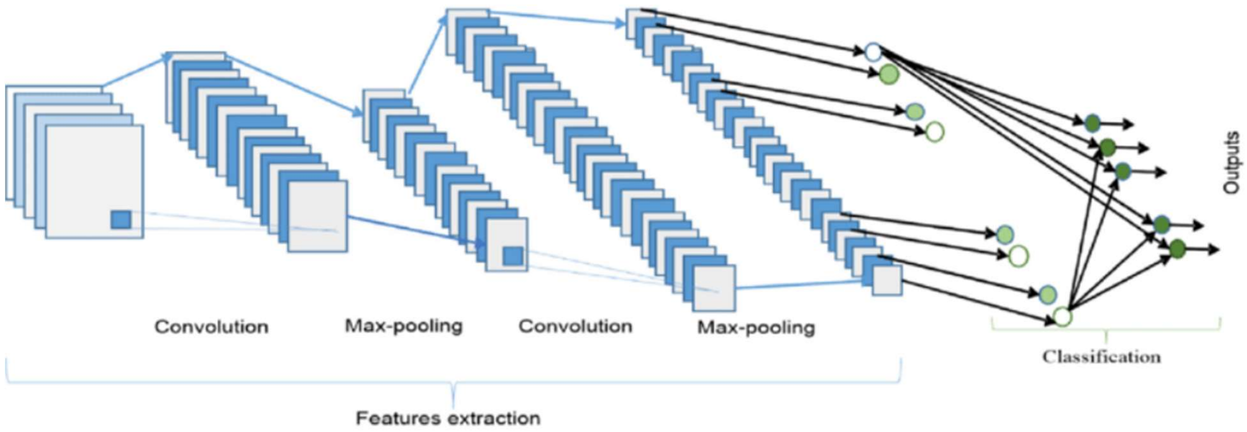
Workflow:



Implementation:

The Face Mask detection model is constructed in this suggested approach utilizing the Sequential API of the Keras toolkit. This enables us to gradually add more layers to our model. The following describes the many layers that were used in our CNN model.

The Conv2D layer, which has 100 filters and a filter size or kernel size of 3X3, is the top layer. The "ReLu" activation function is employed in this initial step. Rectified Linear Unit, or ReLu, is a function that outputs zero if the input is negative or the input directly if it is positive. For every image that will be trained and tested using this model, the input size is also set to 150X150X3.



The MaxPooling2D is utilized in the second layer with a pool size of 2X2.The following layer is once more a Conv2D layer with an additional 100 filters of the same 3X3 filter size, and the "ReLu" activation function is employed. The MaxPooling3=2D layer, with a pool size of 2X2, comes after this Conv2D layer. The next step is to flatten all the layers into a single 1D layer using the Flatten() layer.

To stop the model from overfitting, we employ the Dropout (0.5) layer after the Flatten layer.We employ the Dense layer with 50 units and the activation function "ReLu" toward the end.

Another Dense Layer with only two units and the "Softmax" function as the activation function will make up the final layer of our model. A vector representing the probability distributions of each of the input units is the output of the softmax function. This uses two input units. A vector with two values from the probability distribution will be produced by the softmax function.

We compile the model after building it and specify the loss function and optimizer function. For the purpose of training this model, the "Adam" Optimizer and "Binary Cross Entropy" are used.

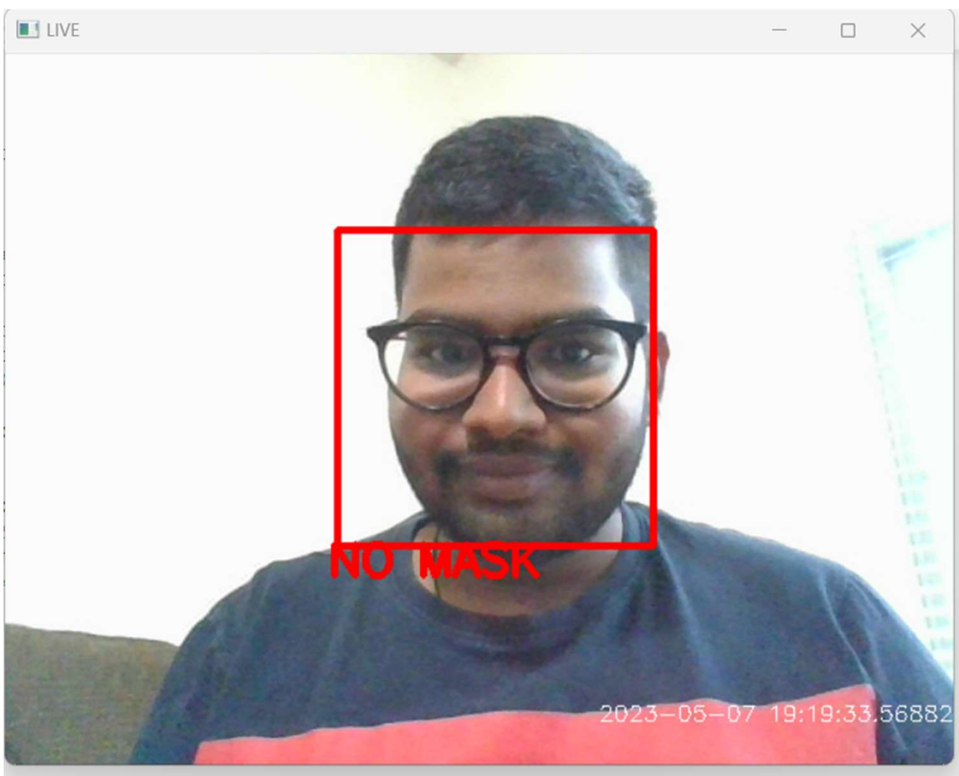
Training a model with CNN:

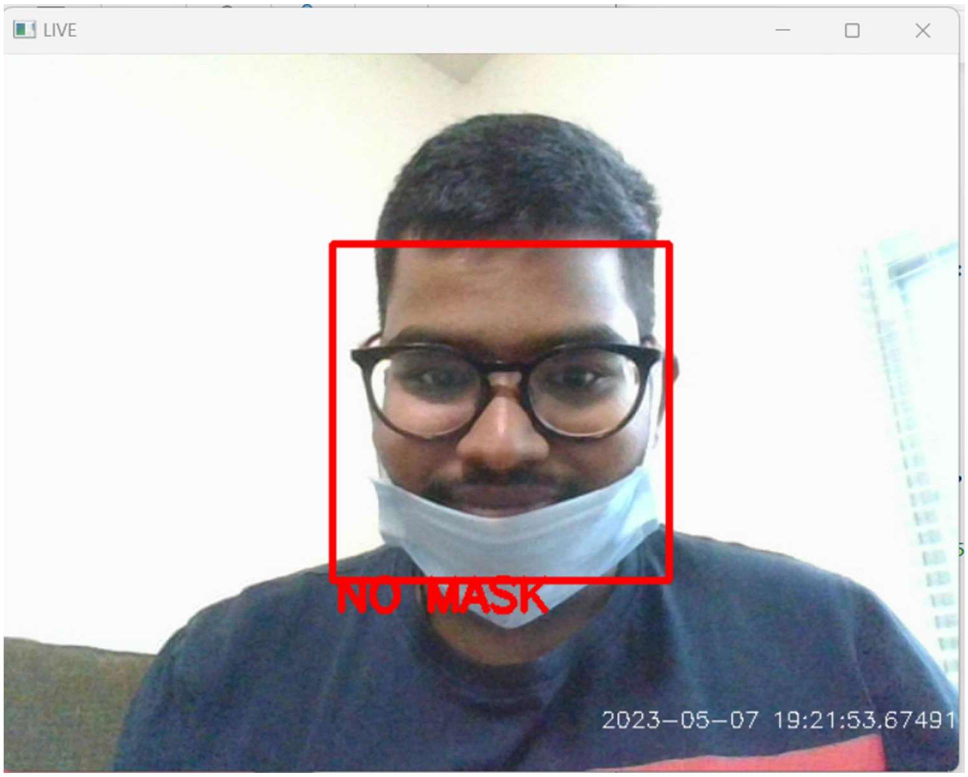
In this experiment, the Haar Feature-based Cascade Classifiers are utilized for face detection. It is a machine learning object detection technique that uses attributes introduced by Paul Viola and Michael Jones to recognize items in an image or video. This involves training a cascade function using a large number of both positive and negative images. The next step is to utilize it to find items in other pictures.

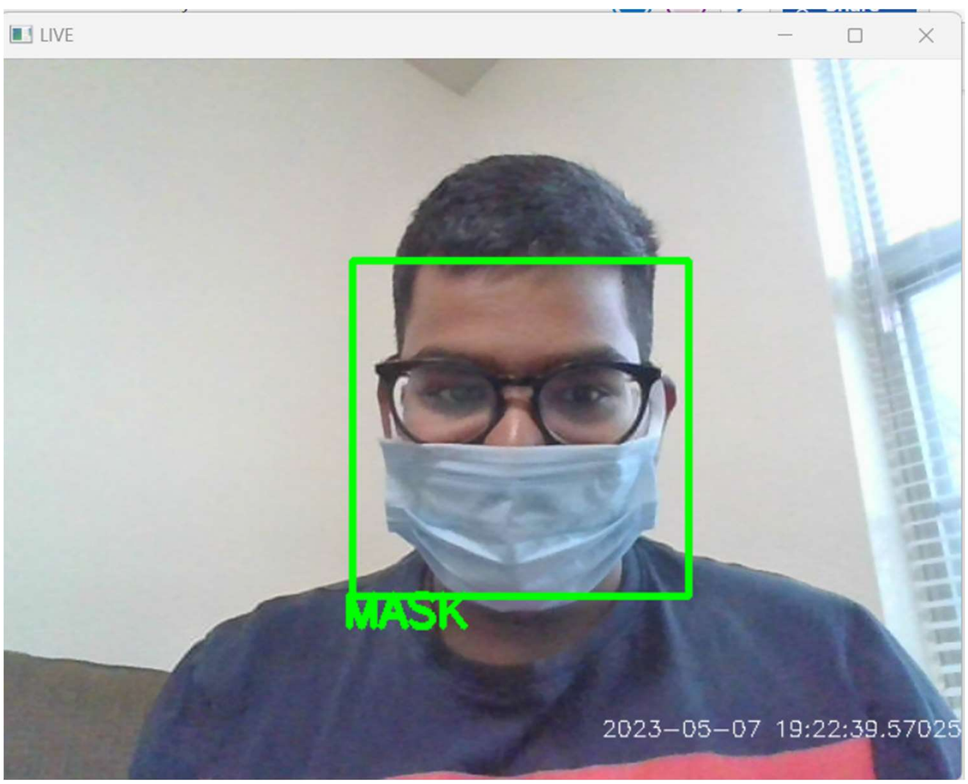
The Face Detection Cascade Classifier was the cascade classifier employed in this experiment. In this experiment, faces are detected in real-time using a model that has been trained on frontal facial traits.

Last but not least, the CNN model and cascade classifier are trained for 30 iterations with two classes: one class represents the class of photos with face masks and the other class represents images without face masks.

Output:




LIVE
NO MASK
2023-05-07 19:21:53.67491


LIVE
MASK
2023-05-07 19:22:39.57025

Functionalities:

- Face-mask detection:
  Identifying the individual whether they are wearing facemask or not.
- Automated alerts:
  In case of severity, individuals without mask are identified and required action is taken against them)
- Analysis of data extracted:
  Analysis of data extracted from observing individuals are wearing face masks or not helps to take accurate decisions.
- Contactless:
  Identifying the individual regarding the face mask without any physical contact.
- Security:
  This technology helps to increase security in terms of maintaining hygiene, by identifying people without masks.

Lessons Learned:

Here are some of the key lessons that can be learned:

1. The use of computer vision techniques, particularly CNNs, can help detect whether an individual is wearing a face mask or not in public spaces.
2. The accuracy of the mask detection model can be improved by adjusting the learning rate value.
3. The selection of an appropriate face detection model is crucial for achieving accurate mask detection results in various conditions.
4. CNNs can effectively recognize patterns and features in images, making them well-suited for image classification and object detection tasks.
5. The accuracy of the mask detection system can be affected by factors such as lighting, image quality, and the position and orientation of the face.
6. The deployment of face mask detection systems using CCTV can aid in monitoring and enforcing compliance with public health guidelines during pandemics.

7. The development of face mask detection systems using CNNs is an ongoing process that requires continuous testing, optimization, and refinement to improve their accuracy and effectiveness.

Conclusion:

In this paper, we studied the problems of detecting the use of face masks. An experimental study was observed using the learning rate on the accuracy of the mask detection model, showing that the smaller the learning rate value increases the model's accuracy. Furthermore, an experiment was conducted to determine the best face detection model capable of detecting faces in normal conditions, using masks and various variations in the angle of the face to the camera. The results show that the RetinaFace model performs the best in detecting faces in a variety of conditions. Based on our observations and analysis results, the existing face detection models have produced the best performance for the front view face and less occlusion. However, they are still constrained by decreased performance when the face has various occlusion and face angles. It is still a challenge that needs to be developed to improve the accuracy of the detection model for masks in public spaces.

Project contribution:

Avinash Reddy Sallagonda

- Gathering Datasets
- Finding out the supported libraries
- training the Model
- Identifying the loopholes in the data

Sai Nikhil Kanchukatla :

- studying about tensor flow
- Gathering datasets
- Identifying the loopholes in the model

Narendhar Reddy Mareddy:

- Training the dataset
- Testing the output
- Optimizing the

modelSai Rishith Reddy

Gade:

- Finding out the supported libraries
- Training of images and implementing algorithms for training.
- Analyzing the results
  optimizing the code and accuracy

Work Schedule:

04-08-2023:

- Gathering the required libraries
- Gathering Datasets
- Setting up the code flow
- Preprocessing the

data04-15-2023:

- Training the data
- Labeling the data
- Building the partial model

04-19-2023:

- Building the model completely
- Testing the model and finding out the accuracy for the images
- Analyzing the results

04-23-2023:

- optimizing the code
- Testing the output

References:

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9649065

https://www.frontiersin.org/files/Articles/855254/fpubh-10-855254-HTML/image_m/fpubh-10-855254-g001.jpg

Github:

https://github.com/SaiNikhilKanchukatla3300/CSE5349-Team10

Youtube link:

https://www.youtube.com/watch?v=D8pkL1ovi2s