Real-Time Face Mask Detection using Convulutional Neural Networks

Sai Teja Palla
20000325-T097
Department of Computer Science
Blekinge Institute of Technology
sapb20@student.bth.se

I. Introduction

The emergence of the Corona Virus has put the world's health in jeopardy. On January 30, 2020, the World Health Organization labeled it a public health emergency of worldwide concern. [1]. Corona Virus had compelled us to reconsider how we conduct our daily lives while keeping ourselves and others safe. To combat the spread of the new coronavirus, simple and effective tactics are needed, with the most effective safety measure being the use of a face mask in public settings. Face mask detection systems are presently in high demand for ensuring safety in transit, highly crowded places, residential districts, large-scale manufacturers, and other businesses.

Face recognition is a category of biometric security. Face recognition technology is used for protection, surveillance, and authentication. Recognition of face is an essential component of people's daily interactions and lives. Accordingly, there is a high demand for automatic facial recognition for identity verification. A face mask detector can analyze whether someone is wearing a mask. In the discipline of Computer Vision and Pattern Recognition, face detection is a crucial area.

The research of face detection began in 2001, with the help of traditional machine learning algorithms and the design of features to train effective classifiers for detection [2]. However, there occurred a problem with these complex features and low recognition accuracy. Later, face detection methods were created based on deep convolutional neural networks(CNN). For face detection, it is now regarded the most. It provides a high level of performance for detecting face patterns. This strategy can be used in various settings, such as the military, defense, schools, colleges and universities, airplanes, banking, online web apps, gaming, etc. Face masks have become a vital part of our lives. Various Organisations and individuals need support with distinguishing between people who wear masks and who won't wear masks in a given location. The inherent diversity in faces, such as shape, texture, color, having a beard, mustache, glasses, and even masks, is a significant issue in face detection. Hence, the proposed convolutional neural networks and Python algorithm are efficient and accurate for determining individuals' face recognition and detection.

In this project, we build a real-time face mask detector that detects whether someone is wearing a mask. We use Keras to create and train the CNN model to predict face with and without a mask. We use the Opencv library for real-time computer vision; here, we use it to create a face mask detector in our use case. Our CNN model achieves training accuracy as 93.38% and 89.89% as test accuracy. We have the dataset covering 1376 images with 690 images of people wearing masks and 686 images without masks.

II. Related Works

In this study [4] the authors are interested in the face detection process and the role of the human face's interest areas. They suggest the use of horizontal and vertical IPC to precisely pinpoint the facial area (Integral Projection Curves). They have studied the role of significant facial features such as nose and eyes. The characteristic features are built using an efficient approach based on PCA (Principal component analysis) followed by EFM (Enhanced Fisher Model) and then forwarded to the classification step utilizing two methods, Distance Measurements and SVM (Support Vector Machine). Finally, the impact of combining two modalities (2D and 3D) is investigated [4].

YOLO is suitable for target detection in a real-time context because of its quick detection speed. It boasts a higher detection accuracy and a faster detection time when compared to other similar target detection systems. Face detection is the subject of this study, which is based on the YOLO network. The YOLO target detection technique is used to detect faces in this article. The face detection approach based on YOLO has a higher resilience and detection speed, according to the findings of the experiments. Even in a complex setting, good detection accuracy can be guaranteed. Detection speed can match real-time detection requirements at the same time [3].

Due to the particular shape of human faces, existing standards designed for detecting faces with masks on them do not operate effectively. Face recognition is one of the most recent biometric technologies being researched, and it has a wide range of applications. Face identification, on the other hand, is one of the most difficult tasks in image processing. Face detection's main goal is to determine if there is a face in an image and then to pinpoint the position of that face in the image. Face identification is the first step toward developing an automated system that may include additional face processing.

With a training set of faces and non-faces, the neural network is formed and trained. All of the results were created in the MATLAB 2013 environment [5].

The Local Binary Pattern approach is used for filtering the candidate region. LBP reflects the finer points of a person's face, with a focus on texture features. Following the identification of the face using global features, the candidate region is filtered using the local feature recognition LBP approach. LBP reflects the finer points of a person's facial features, with a focus on texture features. Skin color detection and LBP are combined in this study. The facial image is initially identified if the number of pixels representing skin color points exceeds the threshold. Aside from that, it's an image without a face. The candidate window is then identified using the LBP technique. It'll be a facial image if the match is successful. Otherwise, it's a non-human face. This approach, however, is incapable of recognizing faces hidden behind masks or glasses. [6].

III. Methodology

The project's main goal is to create a Convulutional neural network model that can recognize whether or not someone's face is with a mask. We used a convolutional neural network with two pairs of Conv and MaxPool layers to extract features from the face images in the dataset to achieve this goal. Then comes the Flatten and Dropout layer, which converts the data into one dimension. Finally, two dense classification layers which are used for classifying the image based on the output from convulutional layers.

A. CNN Architecture

Convolutional neural networks are a key breakthrough in deep learning. CNN's are a type of neural network commonly used for image recognition and categorization. They are mostly used to recognize patterns in images. We don't feed its features; instead, it recognizes them on its own. Image and video recognition, image classification, medical image analysis, computer vision, and natural language processing are only a few of their uses. Convolution, Pooling or Sub Sampling, Non-Linearity, and Classification are the four fundamental processes of CNN [7].

•**Convolution:** Convnets has got the name from the convolution operator. In the case of a ConvNet, the primary goal of convolution is to extract features from the input image. By learning image attributes with pixels of input data, convolution preserves the spatial relationship between pixels.

•**Pooling:** Pooling's major goal is to reduce the size of the input image while keeping the important information. It can be done in a variety of ways, including Max, Sum, Average, and so on. We define a window and pick the largest element from it in Max Pooling; alternatively, we may take the average (Average Pooling) or the total of all elements (Sum Pooling) within that window.

•**Non Linearity:** A neural network's activation function is a function that is added to help the network understand complex patterns in the input, or to introduce non-linearity. To introduce non-linearity in CNN, the activation function ReLU (Rectified Linear Unit) is used. By conducting an element-wise operation, it converts all negative value pixels in a picture to zero.

• **Fully Connected:** In the output layer, the Fully Connected layer is a standard Multi-Layer Perceptron with a softmax activation function.According to the term Fully Connected, each neuron in the previous layer is connected to each neuron in the following layer. The output of the convolutional and pooling layers represents the input image's high-level features. The Fully Connected layer's goal is to use these features to classify the input image into several groups using the training dataset as a guide.

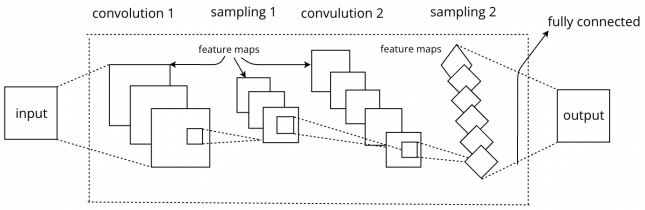

Fig. 1. CNN Architecture

B. Haar Cascade Classifier

The Haar Cascade Classifier is a machine learning-based technique that uses a cascade function to locate an item (positive class) in an image—for instance, finding faces in the image [8]. We can see the demonstration in fig. 6. where my face is detected with the mask.

The Classifier is trained on a collection of pictures that are divided into two categories: positive and negative: images with and without faces. The goal of the model is to extract characteristics from a picture and locate the positive class using Haars filters/kernels.

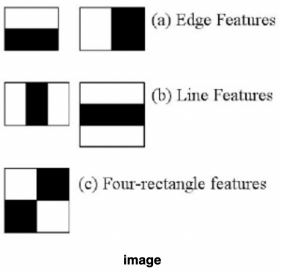

Fig. 2. CNN Architecture

Convolutional kernels are quite similar to Haas filters. Each feature is a single value calculated by subtracting the sum of pixels beneath the white rectangle from the total of pixels beneath the black rectangle. Different features are estimated for different filters, different filter sizes, and different filter positions [8].

Not all portions of an image include the positive class: if a Haar filter feature is unimportant, it is ignored. Adaboost

is used to choose the best features. A threshold value is calculated for each attribute in order to classify the positive and negative examples. In addition, the best characteristics (those that result in the smallest classification error) are chosen. The final Classifier, like an ensemble model, is a weighted sum of the weak classifiers to generate a strong classifier.

C. Dataset

The CNN neural network is trained on a custom dataset that is combined from Kaggle datasets and RMFD Dataset. The RMFD dataset is the world's largest mask dataset, designed to make gathering data resources for intelligent administration and control of similar public events as simple as possible. There are 1376 photos in the Dataset, with two types of people: one with a mask and one without. The image data generation is applied, which is a prerequisite for the deep learning training process.

D. Implementation

In our project, we need to predict whether or not the person is wearing a mask in the webcam. We train the CNN neural network with our training dataset and test the accuracy of the model on the test dataset.

we are training our model on 70% as train dataset, and the 15% is used to validate the model and try to improve the accuracy of the model. the rest 15% test data is used for testing the model based on the metric accuracy score. Python packages used for our model are sklearn,imutils, TensorFlow, OpenCV.

.

•**Pre-processing:** Firstly The entire data(1376 samples) are partitioned into 1100 training samples and 276 testing samples. One of the most critical parts of any machine learning or deep learning training process is the data pipeline. Data pipelines allow you to generate batches of image files and do image augmentation. Our Dataset undergoes image data augmentation where the images are rescaled and modified, which helps our Neural network training process to run smoothly. The images are resized into 150*150 and are divided into batches of 10.

•**Model Creation:** Two convolutional layers are followed by the activation function ReLU and Max Pooling in the CNN model. Dropout is included to prevent over-fitting in neural networks. The final step is to add fully connected layers. Finally, we compiled our model to the loss function, the optimizer, and the metrics. In the learning process, the loss function is utilized to discover error or deviation. During the model compilation phase, Keras requires a loss function, and for the classification applications, we use the binary cross entropy class. Optimization is a key procedure that optimizes the input weights by comparing the prediction and the loss function.

•**Model Training and validation:** We have to split the data before training our model. In our case, there is 70% as a train set,15% as the validation set, and 15% as the test set. The epoch value is set as 10. Models are trained using the fit function. The fit function is basically to evaluate the model on training. The output train generator and validation generator from image augmentation are used to fit the model. After training our model, we save the model for later use.

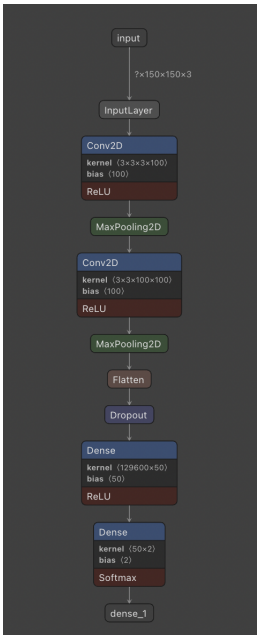

Fig. 3. CNN model structure

•**GUI:** In this GUI code, we load our saved model, which is used for face mask detection, and the haar cascade classifier for face detection. When our GUI code is executed, a webcam window appears, with a green border around the person's face if a mask is worn and a red border around the face if no mask is worn.

IV. Result

The result of the implementation of the CNN based Real-Time Face Mask Detection model is discussed in this section based on the evaluating measure that is accuracy score from validation dataset and test dataset.

A. Accuracy score

The model's training and validation accuracy scores are 93.38 % and 96.39 %, respectively.

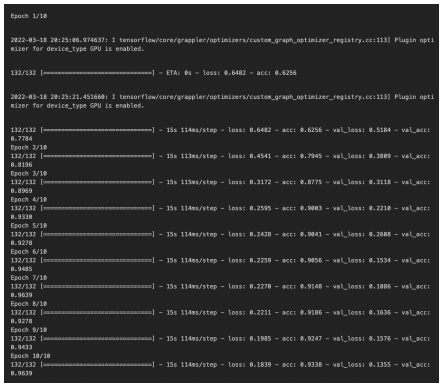

Fig. 4. Training and validation accuracy

For the trained model, the outcome test accuracy is 89.89 percent.%

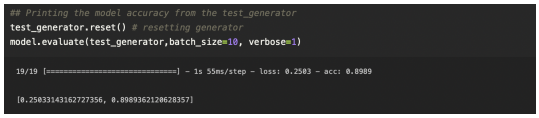

Fig. 5. Test accuracy

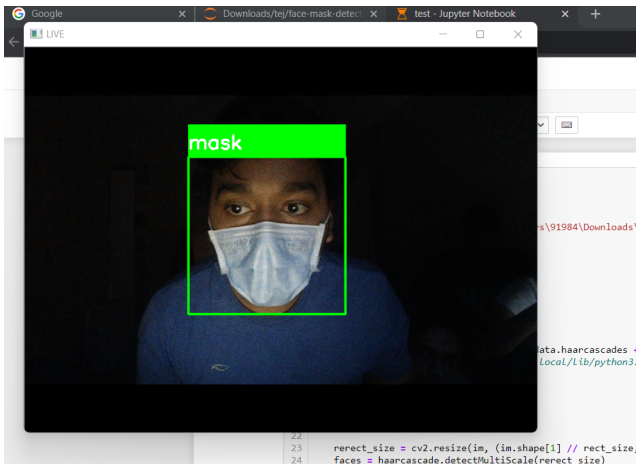

Fig. 6. With Mask

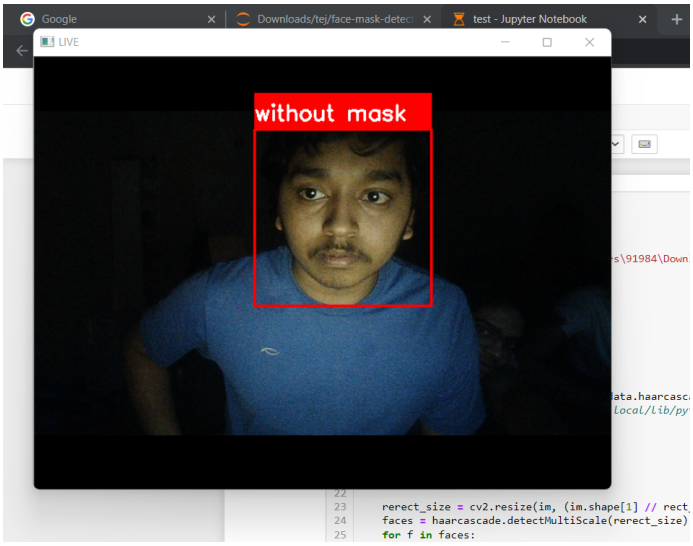

Fig. 7. Without Mask

V. Conclusion

In this project, we have implemented a Real-Time Face Mask Detector using Convolutional neural networks and OpenCV to detect whether or not a person is wearing a mask. Our dataset has various pictures of faces of different shapes, textures, colors, having a beard, mustache, and glasses with and without masks. We have trained the Convolutional neural network model using Keras. The trained CNN model is utilized for classification. The face mask detector detects the face using the haar cascade classifier and uses the saved model to recognize the face and determine whether or not you are wearing a mask. The proposed CNN model shows excellent accuracy and prediction for detecting a human face with and without a mask. The trained CNN model has achieved 93.38% training accuracy and 96.39% validation accuracy and test accuracy as 89.89%.

References

[1] Sohrabi, C., Alsafi, Z., O'neill, N., Khan, M., Kerwan, A., Al-Jabir, A., Iosi- fidis, C. and Agha, R., 2020. World Health Organization declares global emer- gency: A review of the 2019 novel coronavirus (COVID-19). International jour- nal of surgery, 76, pp.71-76.

[2] Nanni L., Ghidoni S., Brahnam S. Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recogn. 2017;71:158–172. doi: 10.1016/j.patcog.2017.05.025. [CrossRef] [Google Scholar] [Ref list]

[3] Wang Yang, Zheng Jiachun "Real-time face detection based on YOLO",1st IEEE International Conference on Knowledge Innovation and Invention 2018

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," IEEE Transactions on pattern analysis and machine intelligence, vol. 23, pp. 681-685, 2001.

[5] S. Saypadith and S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 2018, pp. 1318-1324

[6] Zheng Jun, Hua Jizhao, Tang Zhenglan, Wang Feng "Face detection based on LBP",2017 IEEE 13th International Conference on Electronic Measurement Instruments.

[7] dishashree26, "Architecture of CNN: CNN image recognition," Analytics Vidhya, 24-May-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/architecture-of-convolutional-neural-networks-simplified-demystified/. [Accessed: 19-Mar-2022].
[8] "Cascade classifier," OpenCV. [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. [Accessed: 19-Mar-2022].