Face Mask Detector

Ahmed Eid Abdullah and Ahmed Kamal El Melegy
Zewail city of science and technology
CIE Department

ABSTRACT

At the end of 2019, a new form of viruses has widely spread all over the world, called COVID-19, COVID-19 is a rapidly spreading viral disease that infects both humans and animals, because of this virus, the health of people and economy of many countries are affected. Currently, there are approximately 158 million cases and 3.28 million death cases, and this virus is still spreading rapidly all over the world. The researchers relate this virus with a lung's infections, so the first thing that they do to confirm the positive cases is doing chest x-ray and chest CT to image the patient lungs. Till now, there is not a single country over the world can produce an effective vaccine for COVID-19 virus. The early detection and diagnosis of this new virus with a low cost become the main challenges in current COVID-19 pandemic. Thanks to deep learning and computer vision, we can provide a model that combine computer vision and deep learning together to detect if an individual wear a mask or not. We will use computer vision firstly to detect faces from the live video, then the detected faces will be inputted into the CNNs classifier to detect if wearing face or not. Then again use computer vision to track the face, in case if this face moved left or right by a distance 1 cm, to not repeat the classifying step again, face tracking will be used.

INTRODUCTION

With the outbreak of an unknown virus in late 2019 in China, some people infected with a disease and have a harsh symptom and they do not know that is this. The disease was completely unknown at first, but researchers diagnosed its symptoms as like those who have cold flu. After the laboratory examination and analysis of infected people by polymerase chain reaction (PCR) test, the viral infection was confirmed, and this disease eventually named "COVID-19" by the World Health Organization. Over a short period, The COVID-19 affect many countries economy, until May 9,2021, more than 158 million people are positive cases and more than 3.28 million people died officially due to the disease. The early detection of the disease is very important not only for patient care but also for the surrounding people for public health by isolation of the patient and controlling the pandemic. PCR test is the only test that give good accuracy, but it is very time-consuming, complex, and costly. One of the most important ways to diagnose COVID-19 is to use radiological images (chest X-rays). One way to avoid spreading COVID-19 is there should be social distance between citizens in public places. All citizens should wear mask in public places like school, university, malls, and hospitals. So, implementing a face mask detector is very important. All public organizations should provide a system on the entrance to detect wearing mask or not, if people wear mask, the organization door will be opened, if not the system should give warning to wear mask.

DATASETS

There are two used datasets, one for training the viola jones face detection, and other for training the CNN classifier. The one that used on viola jones called CBCL face dataset that published by MIT's centre for Biological and Computational learning. Each image is 19x19 and greyscale. It divided into training and test sets. The training set composed of 2429 faces and 4548 non-faces, while the test set composed of 472 faces, and 23573 non faces. The other one called Face Mask Detector that from Kaggle. It consists of 7553 RGB images in 2 folders withmask and withoutmask. Images are labelled as withmask and withoutmask. Images of faces with mask are 3725 and images of faces without mask are 3828.
For viola jones dataset: dataset link
For face mask detection dataset: dataset link

PROPOSED APPROACH AND ARCHITECTURE

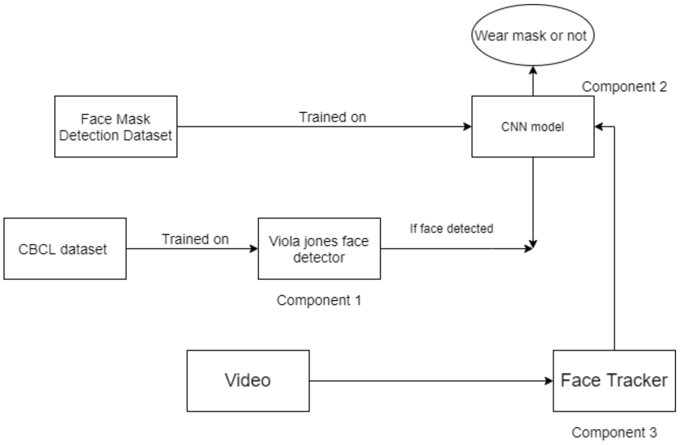


FIGURE (1) OUR MODEL ARCHITECTURE

Our pipeline model consists of 3 main components which are Face detector, CNN classifier model, and face tracker.

FACE DETECTOR VOILA JONES

Image classification is very important task in many fields, and it has been quickly growing filed over the past years. The use of CNNs and deep learning techniques is growing quickly and become the most famous technique to classify images. But, before CNNs became famous, there is another technique used called viola-jones and it still used until now. Viola-jones based on ensemble learning approach, as it uses many different classifiers, each looking at different parts of the image. Each classifier is weak, then the final classifier combines all these. weak classifiers and produce a strong classifier as shown in figure (2)

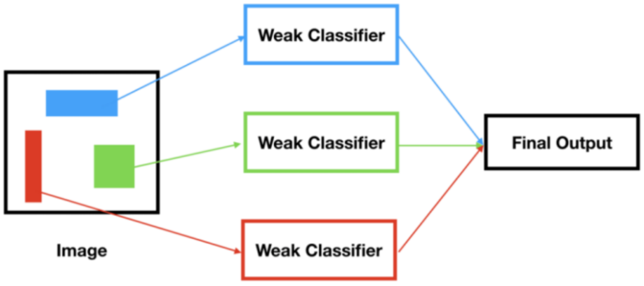


FIGURE (2) VIOLA JONES MAIN IDEA

Advantages of viola jones are: It used in binary classification problem such as object and face detection, it classifies the images quickly because each weak classifier uses small number of parameters, and it has low rate of false positive.

FEATURES AND THE INTEGRAL IMAGE

Viola jones detect faces based on extracting features from these faces, viola jones introduced the following new features that found on figure (3)

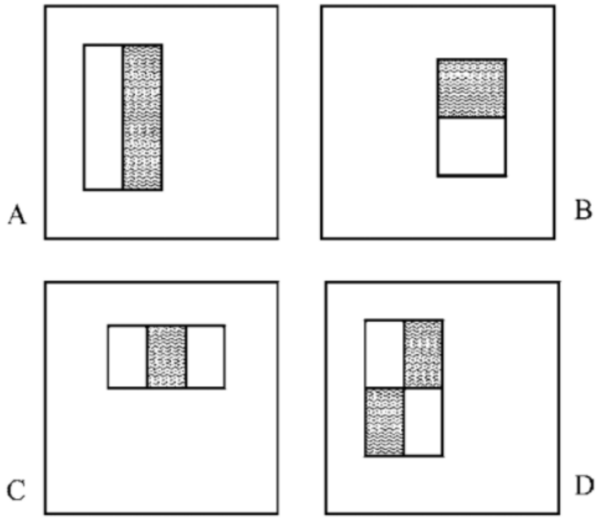


FIGURE (3) A AND B CALLED TWO RECTANGLE FEATURE, C IS THREE RECTANGLE FEATURE AND D IS FOUR RECTANGLE FEATURE.

To extract the features, these rectangle pass through the face, and the sum of the pixels in the unshaded rectangle are subtracted from the sum of pixels in the shaded rectangle. But the problem here is even for small images, there a lot of features and to compute this subtraction, this will lead to computation cost. To reduce the computational steps, Viola jones introduced the integral image. Integral image is defined as a recursive summation relation as found below.

$$\text{ii}(-1, y) = 0$$
$$s(x, -1) = 0$$
$$s(x, y) = s(x, y-1) + i(x, y)$$
$$\text{ii}(x, y) = \text{ii}(x-1, y) + s(x,y)$$

s(x,y) is the cumulative row summation at point (x,y) , while ii(x,y) is the integral image values at (x,y) , and the i(x,y) is pixel value of the input image at (x,y).
integral image is the summation of all pixels above and left of the current pixel value. Integral images make the calculation of features by passing the rectangle over the face easier. So, the viola jones algorithm works on the integral image not the original image.
For training the viola-jones, it uses Adaboost. The idea based on for each subsequent weak classifier, the next classifier corrects the mistakes of the previous classifier. It assigns weight to each training example, then train the classifiers, and choose the best classifier of these weak classifiers and then update its weight according to the error of this classifier and so on. By this method, weights will be updated so that the best classifier is chosen based on the previous classifiers.
Each weak classifier focus on a single feature (f). It has both threshold (θ) and polarity (p) to determine the classification.

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta, \\ 0 & \text{otherwise} \end{cases}$$

```
self.threshold = threshold
self.polarity = polarity
```

FIGURE (4) WEAK CLASSIFIER EQUATION AND PARAMETERS

Polarity can be either 1 or -1, when p = 1, the weak classifier outputs a positive number when feature value is less than the threshold. When p = -1, the weak classifier outputs a positive result when feature value is greater than threshold value.
once training all weak classifiers, selecting the best classifier starts.
The main steps to train by viola jones are 5 steps which are:

- Initialize the training weights
- Normalize these weights
- Select the best weak classifier based on the error
- update the weights based on the chosen classifier error
- Repeat steps 2-4 n times , n is the number of weak classifiers

Viola jones hyper-parameter: The only hyper-parameter is the number of features (number of weak classifiers).

IMPLEMENTATION OF VIOLA JONES FACE DETECTOR

PC SPECS:

- processor: Intel(R) Core (TM) i7-8550U CPU @ 1.80GHZ 1.99GHZ
- RAM: 8.00 GB (7.88 GB)
- Code environment: Python – Jupiter compiler

We implemented 9 functions, image integral function which an image introduced to it as input, and we applied the equations of cumulative summation of pixels on this input image, this function applied recursively then return the integral image.
We also implement a helper class called RectangleRegion to store the rectangle regions to make it easy for computing the values of features later.
We implement viola jones class that contains several functions. Train function whose input is an array of tuples, first element is NumPy array representing the integral image, and the second element is its classification label (1 or 0), another input is the number of positive samples and number of negative samples. In this function we initialize the weights based on below equation.

$$w_i = \begin{cases} \frac{1}{2p} & \text{if } x = 1, \\ \frac{1}{2n} & \text{if } x = 0 \end{cases}$$

where p is the number of positive examples , and n is the number of negitve examples.
Then we implement the build feature function that responsible for building the rectangles found on figure (3). Input to this function is a tuple of the image shape contains the height and width of the image, it returns array of tuples , each tuple's first element is an array of rectange regions which positively contribute to the feature and the second element is the region that negitively contribute to the feature.
As we need to apply each of these features (rectangles) over each training example, we applied the features , before we start training in order to save computations. Apply feature function whose inputs are array of tuples of the features that returned from the build feature functon and the training data as an array of tuples , its first element is the integral images and the seond element is its classification 1 or 0. As each weak classifier has its own variable like polarity and threshold theta. We implement function called classify in the weak classifer class to implement equation found on figure (4). Then we implement the train weak function that train each weak classifer. In order to select the best weak classifer , we implmeneted select best function that select the best classifier.

Time taken to run training on viola jones is: 1 hour and 42 minutes.
We used 50 weak classifiers (50 features)

IMPLEMENTATION OF CNN MODEL

CNN CLASSIFICATION MODEL	
INPUT	Images of size 150x150
OUTPUT	Mask or No Mask
START 1- Input is introduced into Conv2D layer of 32 filter of size 3x3 with RELU activation function 2- Then add MaxPooling2D layer 3- Then add Conv2D layer of 32 filter of size 3x3 with RELU 4-Then add MaxPooling2D layer 5- Then add Conv2D layer of 32 filter of size 3x3 with RELU 6- Then add MaxPooling2D layer 7- Then add flatten layer 8- Add Dense layer of RELU 9- Last layer is Dense with sigmoid activation function **END**	
HYPER-PARAMETERS	- Optimizer - Number of epochs - Number of layers and units
OPTIMIZER	Adam
LOSS FUNCTION	Binary_crossentropy
NUMBER OF EPOCHS	20
TRAINING ACCURACY	98%
TRAINING LOSS	0.0521

Model Testing

As the viola jones trained on 19x19 greyscale images, firstly we divide live video into frames, and each frame we divide it into 19x19 blocks , then introduced these blocks into the viola jones algorithm.
As the CNN model trained on 150x150 RGB images , we resized each frame into 150x150 RGB and introduced it into the CNN model.

Experiment and results

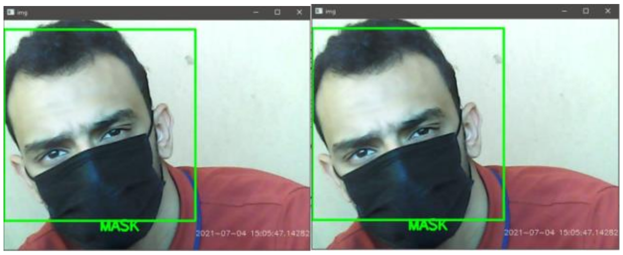


Figure (5) Results of the model (Detect Mask)

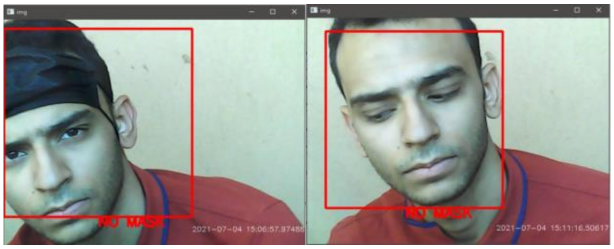


Figure (6) Results of the model (Detect NO Mask)

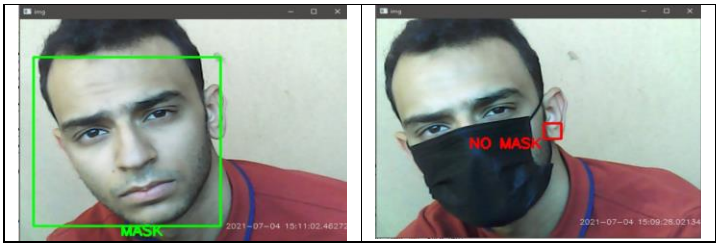


Figure (7) Results of the model (Wrong Detection)

Comparison between our implementation of face detector viola jones with the OpenCV cascadeclassifier module

Opencv cascade classifier can not detect the face if the face wear mask , but in return , it classifies the faces with high stability.
Our implementation detect the faces if it wear masks or not , and it is one of the advantage of our model. Our model suffers from some unstability on face detection , as it can detect face and the box (red or green) appear and dissapear frequently.

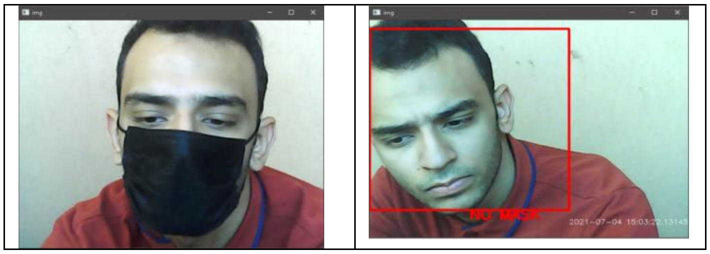


Figure (8) Results of the opencv cascadeclassifiers

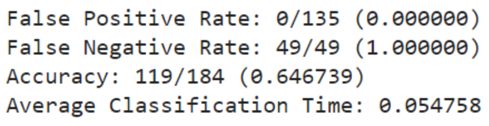
Our model does not work, if it does not detect a face. The red or blue rectangle appears if it detects face.

Performance metrics of the model (all piplines)

- Firstly, we recorded a video of 22 second locally on pc of Ahmed Kamal's face
-specs of the recorded video are: 30 frame per second , resolution640x480
- We load the video by 8 frame per second
- so we have now 8*23 frames = 184 frames
- we did label for these frames , as the first 48 frames ahmed wear mask so that the label is 1
then the next 48 frames , ahmed does not wear the mask , so that the label is 0
the next 48 frames , ahmed wear mask again , so that the label is 1
the last 48 frames , ahmed does not wear the mask , so that the label is 0
- we stored each frames and their label on a pkl file called mine.pkl and this file will be used later to evaluate the performance .
- In evaluate function we did the following: -
For every frame we resized it into 19x19 and we did another copy of this frame for using it later in CNN by resizing it into 150x150 RGB.

- Start timer
- If viola jones output is 1(detect face), then check wearing mask or not.
- Then introduced it into CNN model
- Then compare the output of CNN model with label on the mine.pkl file to evaluate the confusion matrix.
- Else, finding the false positive, false negative values
- End timer and record time taken to classify the frames.

Here our confusion matrix and the time taken to classify the 184 frames which is equal to 0.054758 seconds.



```
False Positive Rate: 0/135 (0.000000)
False Negative Rate: 49/49 (1.000000)
Accuracy: 119/184 (0.646739)
Average Classification Time: 0.054758
```

FIGURE (9) CONFUSION MATRIX AND TIME TAKEN TO CLASSIFY FRAMES

So that the time taken to classify one frame is equal to 0.054758 / 184 = 2.97*10^-4 seconds.

The overall accuracy of our model is 64%

TO CALCULATE THE HOW MANY FRAMES ARE PREDICTED CORRECTLY CONSECUTIVELY:

We made the counter variable for detection stability to make the model detect face if more than specific number (counter) of frame consecutive are classified as face, this it make the model stable but when the counter threshold are above 1 , the model fails to detect any faces in the frame , it only detect face in the frame if this threshold counter is set to 1. This means consecutive true classified variable that we were asked to make from the Eng. Ahmed El Sayed will always zero because the counter threshold is 1to detect any face in the frame.

CONCLUSION AND FUTURE WORK

With the outbreak of an unknown virus in late 2019 in China, all countries put regulations on people on the public area. There should be a face mask detector on public areas. We implemented this detector by combining two pipelines which are viola jones algorithm used to detect the faces, then we implement a CNN model for classification problem (wearing mask or not). Our model achieved 64% accuracy which is not bad. We will work on the model in the future trying to increase its accuracy.

ACKNOWLEDGMENT

Apart from the effort of us, the success of any project depends on the guidelines of many other. We take this opportunity to express our gratitude to the people who have been always guide us in the computer vision course. We would like to show our greatest appreciation to Dr. Elsayed Hemayed. We thank him for his tremendous support and help. We would like also thank Eng. Ahmed El Sayed for his continuous support and help.

REFERENCES

[1] S. Lee, “Understanding Face Detection with the Viola-Jones Object Detection Framework,” *Medium*, 22-May-2020. [Online]. Available: https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14. [Accessed: 04-Jul-2021].

[2] *Cs.ubc.ca*,2021.[Online].Available:https://www.cs.ubc.ca/~lowe/425/slides/13-ViolaJones.pdf. [Accessed: 04- Jul- 2021]

[3] Efendi, M. Zul and W. Yunanto, "Real Time Face Recognition using Eigenface and Viola-Jones Face Detector", *JOIV : International Journal on Informatics Visualization*, vol. 1, no. 4, p. 16, 2017. Available: 10.30630/joiv.1.1.15.