# Project Proposal
# True Face Revealer
# "Put on your mask to show your true face!"

Kiran, L.
IIST

June 16, 2020

# Abstract

Face mask has been a mandatory wear these days due to the pandemic of COVID-19. Manual surveillance and enforcement of use of face mask by the public is a laborious task. The same task can be performed with reduced effort by implementation of a face mask detection system. In case of the availability of database of people's faces, a face recognition system can be augmented to identify the violators and notified them with proof. In this project proposal, we present three methods of developing the system by - Eigen-Jaw method, R-CNN and YOLO. The Eigen-Jaw method is based on PCA and is not pose invariant as well as, has high false positive rate. To improve upon the same R-CNN model is proposed which overcomes the limitations of Eigen-Jaw method, but suffers from high time complexity. To achieve real time performance, YOLO algorithm is proposed for implemention which is only limited by poor detection rate of small sized objects in an image (faces of people, far from camera).

*Keywords*: Face detection, Mask detection, Eigen-Jaw method, Image Procesing, Principal Component Analysis, Python, Open-CV, Neural Network, R-CNN, YOLO.

# Contents

# Chapter 1

# Introduction

During the present pandemic of COVID-19 all over the globe, public safety is the prime concern. As the lockdown period cannot be extended indefinitely, partially for economic reasons, gradually the lockdown was released allowing economic activities. Electronic modes of transaction was advocated to minimize the transmission of disease and without affecting the economy, i.e., usage of touch-free transaction modes[1]. A safe environment is the need of the hour, in order to re-open various facilities. New solutions and policies are being proposed and implemented to achieve the same. One of the primary ways to assure safety in crowded places is installation of real time face mask detection system, which enables the higher ups in enforcing public safety[6].

## 1.1 Problem Statement

A face mask detection system is to be designed. The system should be able to detect faces of people in any crowd, with varying levels of illuminations (through out the day), scale, pose and alignment. It should also be able to detect and identify faces being covered with mask as masked face.

## 1.2 Objective

The primary objective of this project is to design a system which is able to take in static images, locate people's facesin the them and identify each one

of them as wearing or not wearing face mask. The objective can be broken down into two broad sub-objectives as:

1. Face Detection
2. Face Mask Detection

Machine Learning and deep learning algorithms and classifiers are to be used for identifying (classifying) the detected faces in given test image as masked or un-masked faces. The input test image may contain faces of more than one person (i.e., multiple subjects per image and alike).
Further, if possible, a face recognition system is proposed to be augmented to the system to enable notifying the violators.

## 1.3 System Requirements

The proposed project requires the following system and hardware facilities:

1. Jupyter Notebook
2. GPU hardware accelerator for faster training of CNNs and FFNNs (preferable, Nvidia GPU, for CUDA support)
3. Video/Image source - Camera

Most the the project implementation can be completed using Google Colaboratory Platform, except possibly the real time object (face and face mask) detection, which requires offline resources. The model prediction can be completed even without GPU accelerator, hence, a low end system with RAM $\geq 2GB$ with all necessary Python modules installed suffices.

# Chapter 2

# Literature Survey

## 2.1 Survey on Real Life Requirements of Face mask detection systems

The literature survey was performed to get an insight to the real life situations where face mask detection systems can reduce effort and time.

With the increase in the number of CCTV cameras throughout the globe, the time and effort required in monitoring the public and enforcing proper safety measures such as social distancing and wearing of face mask by everyone can be minimized to significant level. Also, this system can be integrated with the automatic door which identifies the arrival or departure of people from shops to ensure that the door opens only when the person is additionally wearing face mask (which ensures safety at the cost of possible constraint of space for passage to allow only one person).

The face mask detection system can be integrated in work places to ensure that all the workers are wearing face mask mandatorily while working. In case on not following the rules, the same can be enforced by an instant reminder service system such as sending a reminder SMS to the person or announcing openly, by the supervisor.

In the above mentioned applications, there arises the need for identification of the person along with detection. In case of the objective of the system being to enforce wearing face mask in places with finite subjects, a face recognition system can be integrated into the system.

Also, this system has been introduced in China, where face pay technology existed previously. In Alipay[1], the system has been upgraded to allow its

users to transact through face pay even while wearing face mask. The same system finds its application in ATMs to validate identity of people transacting in the ATMs, even in case of wearing face mask.

Face mask detection systems also find application in airports, hospitals, shopping marts, railway stations, bus stands, and various other public places where crowd can possibly accumulate.

Similarly, many such real life requirements do exist, where face mask detection system (with or without integrated face recognition system) can play a cruical role.

## 2.2 Literature Review

Various students, research scholars and industrialists working in **Pattern Recognition and Machine Learning**, **Deep Learning** and **Computer Vision** have developed algorithms and deep convolutional neural networks (CNNs) for achieving the tasks of object detection, broadly, and face detection and recognition, face mask detection and alike for myriad applications.

In order to help better the situation due to the pandemic of wide spread COVID-19, **DiDi AI team** has build a mask detection system[2]. The system is based on Depth First Search (DFS) algorithm for face detection (which as will be illustrated, is the most cruical step in the whole pipeline) and the face attributes recognition algorithm used by DiDi. The system so built is robust with high degree of invariance to lighting, face pose, alignment, scales, etc. It can deal with different mask types and uneven mask data during the day and the night. The system includes face recognition system to identify drivers (in its database) not wearing mask with 99.5 % accuracy, and achieves 98 % accuracy during DiDi's spot inspection with in-vehicle cameras. The model, which is pretrained by public ResNet50-caffemodel, was trained on a dataset of 200,000 faces to ensure its robustness. This quick detection system can be widely used in travel scenes, with mobile phone photos, surveillance images etc., and is able to work round the clock.

In a paper titled "**Facial Mask Detection using Semantic Segmentation**" by Toshan Meenpal, Ashutosh Balakrishnan and Amit Verma of National Institute of Technology Raipur[3], an extensive work on building system for detecting faces which are pose, alignment as well as

scale invariant is presented. Beginning from the RGB image of any size, the method uses pretrained (on ImageNet dataset) VGG-16 architecture for feature extraction. The output image from the fully connected layer is processed to remove the unwanted noise and avoid the false predictions, if any, and make bounding box around the faces. Experiments were performed on Multi Parsing Human Dataset obtaining mean pixel level accuracy of 93.884 % for the segmented face masks.

**LeewayHertz:** Software Development for Startup and Enterprises has also developed a fully functional face mask detection system (with recognition) which has been launched in the form of an mobile application (app) and also as a website[4]. The app can be connected to any IP mask detection cameras to detect people without a mask. App users can also add faces and phone numbers to send them an alert in case of not wearing a mask. If the camera captures an unrecognized face, a notification can be sent out to the administrator. AI alerts are sent with the picture of the violator which allows the app to run automatically and enforces the wearing of the mask.

Similar to above, a real time COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning has been made by Adrian Rosebrock[5].

The above list of projects is non-exhaustive, as many research papers of new architectures for object detection with deep learning, CNNs, Region Proposal CNNs, YOLO, etc., are coming up very frequently with a large community or group of teams putting in effort to build real time applicable systems.

# Chapter 3

# Requirement Specifications

The following problem scope for this project was arrived at after reviewing the literature on face mask detection and determining possible real-world situations/places where such systems would be of use. The following system(s) requirements were identified:

1. A system to detect faces in different poses in static images (or frames of video).
2. A system to detect presence or absence of face mask in faces detected.
3. All implemented systems must display a high degree of lighting invariance.
4. All systems must posses near real-time performance.
5. A fully automated face mask detection must be supported
6. The face mask detection system must display a high degree of invariance to scaling, alignment, pose in the detected faces with/without mask.

Unfortunately although we may specify constricting conditions to our problem domain, it may not be possible to strictly adhere to these conditions when implementing a system in the real-world.

# Chapter 4

# Performance Metrics

### Confusion Matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

### Recall:

$Recall = \frac{TP}{TP+FN}$
Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

### Precision:

$Precision = \frac{TP}{TP+FP}$
To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled as positive is indeed positive (a small number of FP).

- **High recall, low precision** This means that most of the positive

examples are correctly recognized (low FN) but there are a lot of false positives

- **Low recall, high precision:** This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP).

**F-measure:**

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more. The F-Measure will always be nearer to the smaller value of Precision or Recall. $F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision}$

**Summary**

We, thus, need to design a system that has both high precision and recall. The purpose of high recall is quite obvious, since the one of the main motives of project is to help public in the current scenario of COVID-19, by enforcing usage of safety face mask.
A high precision is required to avoid disturbing people already following the rules and regulations and wearing safety face mask.

# Chapter 5

# Methodology

This chapter is dedicated to explain the possible methods, algorithms and approachs through which the objective of real time face mask detection can be achieved.
Here, we present the ideas in order of increasing complexity. Where ever possible, the results of the implemented algorithms have been presented.

## 5.1 Method-1: Eigen-Jaw Method

The first method we present here is based on Principal Component Analysis (PCA). We propose to use the method and algorithm similar to that used in face detection and recognition. The only difference in the application is that number of output classes, which is just 2 (with and without face mask) and may be more than two in system built for face recognition.

### 5.1.1 Face Detection

The faces can be detected using Viola Jones Algorithm implemented in OpenCV's haar cascade classifiers. This set of classifiers form a subset of classifier used for general application of object detection. As the region of interest is not the whole face, only the lower half of the face (jaw) may be considered for analysis.

### 5.1.2 Face Alignment

Here, we consider the frontal faces alone for analysis purpose. The frontal faces detected need to be aligned squarely. This can be achieved using eye detection and determination of the angle of tilt of face by triangle method[7]. This detected tilt angle can be used to rotate the face image in opposite sense to orient the face with pair of eyes along a horizontal line and mouth situated on line perpendicular to line joining eyes.

### 5.1.3 Image Processing

As the input images can have varying degrees of illuminations, we need to histogram equalize the input images to reduce the effects due to varying illumination conditions, encountered in daily life.

### 5.1.4 Data Augmentation

As any designed classifier must be robust and be able to detect and identify if a person in the image is or is not wearing face mask, we can augment noisy data to the training data. The noise can be added by including images distorted in the following ways:

1. adding gaussian noise - to account for the variation in camera clarity
2. rotating the faces by arbitrary angle - to improve invariance of the system to face alignment
3. cropping the face images - to aid the classifier to identify the presence or absence of face mask with incomplete images in training data

### 5.1.5 Principal Component Analysis

Through Principal Component Analysis (PCA), the face space image are mapped to eigen-space of lower dimensions. This method is mainly useful when the dataset size is large and memory available is small (limited). Thus extracting the most useful information from the input and using the same further analysis.

As a standard practice, the eigenspace representation of the images can be normalized (using standard scaler) and hence, the test images.

### 5.1.6 Classifier

Once the input data is ready in the desired form, it can be used for classification. Here, various two-class classifiers can be used such as Perceptron, Support Vector Machines, kNN, and like.

From the result of Universal Approximation Theorem, we know that any complex function between the input and output can be approximated by a neural network with finite neurons and hidden layers. Thus, one can build artificial feed forward neural networks for classification of the input images into two classes of interest - With and Without Face Mask.

### 5.1.7 Results

In this subsection, we present the results obtained upon implementation of the above algorithm on a small dataset (collected from internet). The number of principal components considered for analysis is 50.

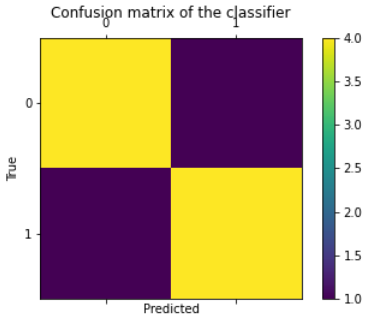


Figure 5.1: Confusion Matrix

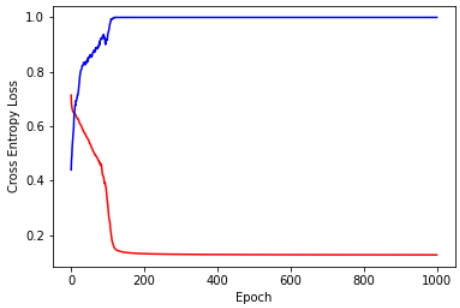


Figure 5.2: Loss (red) and Accuracy (blue) Curves

The following are the set of parameters used to build the neural network:

| Parameter | Value |
|---|---|
| Initialization | Random |
| Activation Function | $tanh(\frac{3}{2}(x-3))$ |
| Neural Network Configuration | [50,45,16,2] |
| Number of Epochs | 1000 |
| Batch Normalization | included for all hidden layers |
| Dropout | not included for any layer |

Figure 5.3: 50 Principal Eigen-Jaws

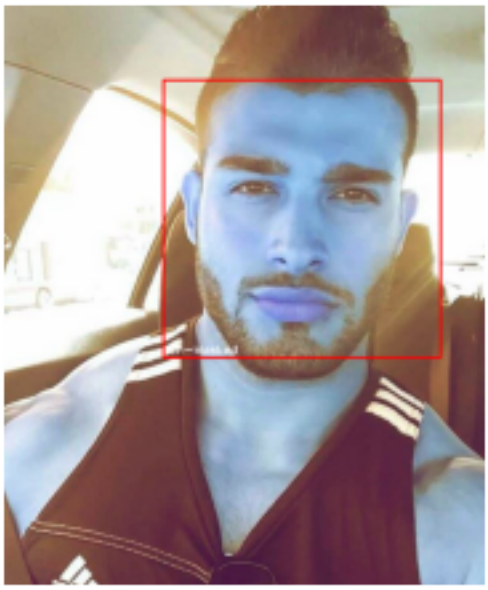

Figure 5.4: Without Mask - Predicted with probability 0.8419

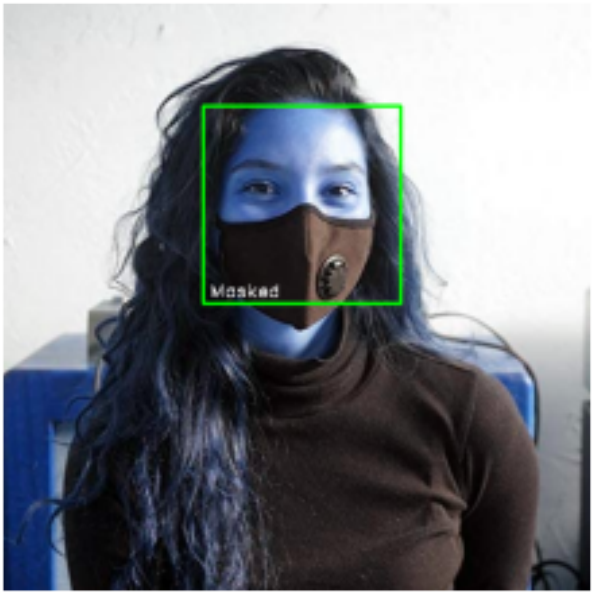


Figure 5.5: With Mask - Predicted with probability 0.8802


Figure 5.6: Test Image - 1 : All correctly identified

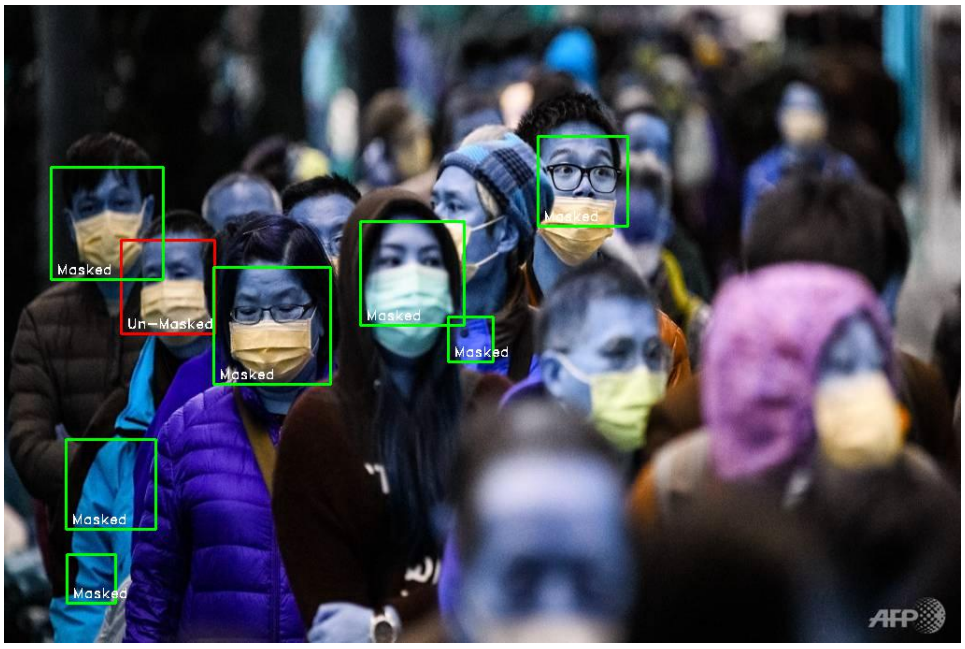


Figure 5.7: Test Image - 2: Some incorrectly identified and some false face detection

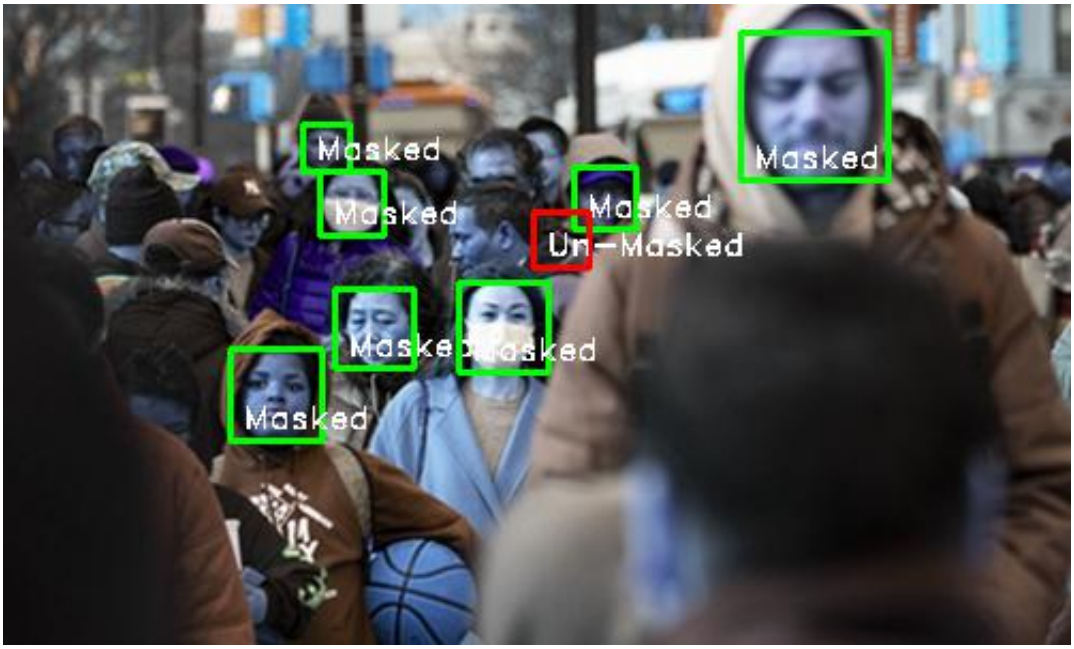


Figure 5.8: Test Image - 3: Some incorrectly identified and some false face detection

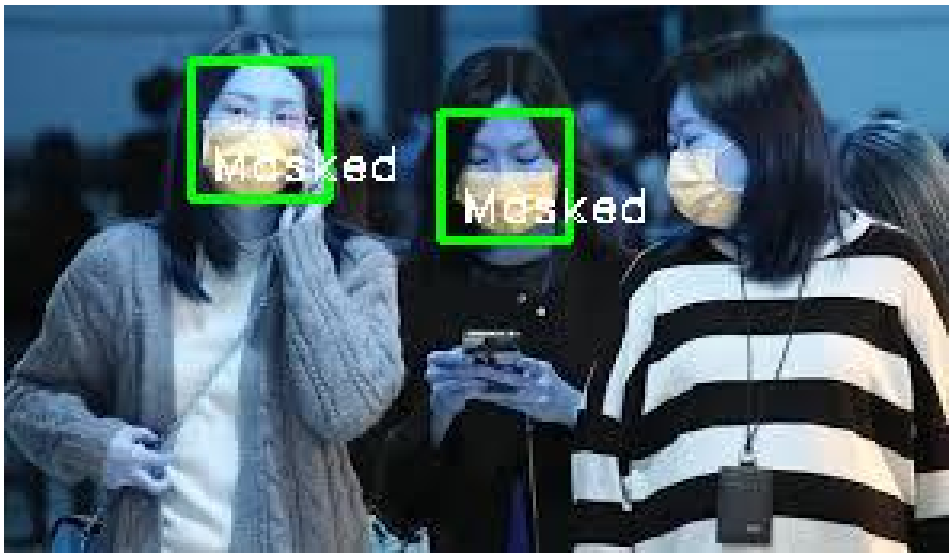


Figure 5.9: Test Image - 4: One not identified

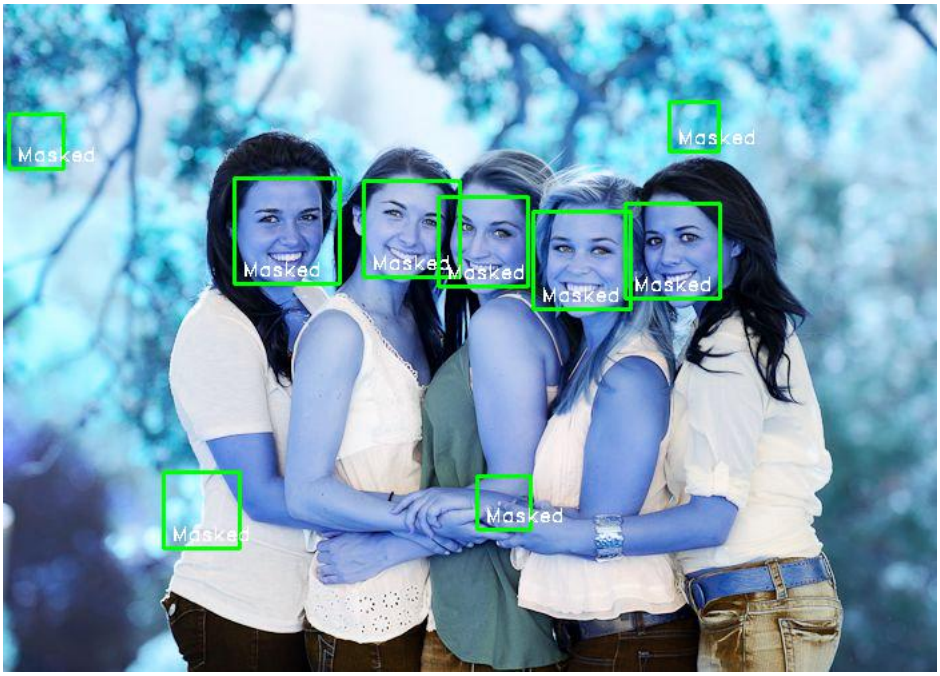


Figure 5.10: Test Image - 5: None correctly identified

| **Requirement** | **Actual** |
|---|---|
| Real Time Performance | $\approx 0.15 - 2s$ per input image (increases with increase in the number of faces detected) |
| Low False Positives | High as seen from the last test image (test image - 5) |

| Advantages | Dis-advantages |
|---|---|
| 1. System able to detect presence or absence of face mask given frontal face image input | 1. Can only work if the input images contain frontal faces and is not pose invariant as seen in Figure 5.9. |
| 2. Near real time performance (time taken $\sim$ 1s) | 2. The time of processing and execution increases with increase in the number of faces detected in the given frame (of video) or static image. |
| 3. Low false positives on images containing single subject per image (as shown in Figure 5.4 and 5.5) | 3. High false positives in images with multiple subjects, as seen in Figure 5.9. |
| 4. The system is invariant to slight degree of face misalignment and performs satisfactorily on low quality images. | 4. The system is not highly robust, since the whole pipeline depends on the efficiency of face detection (bottle neck in the pipeline). |

## 5.2 Method-2: R-CNN (Object Detection)

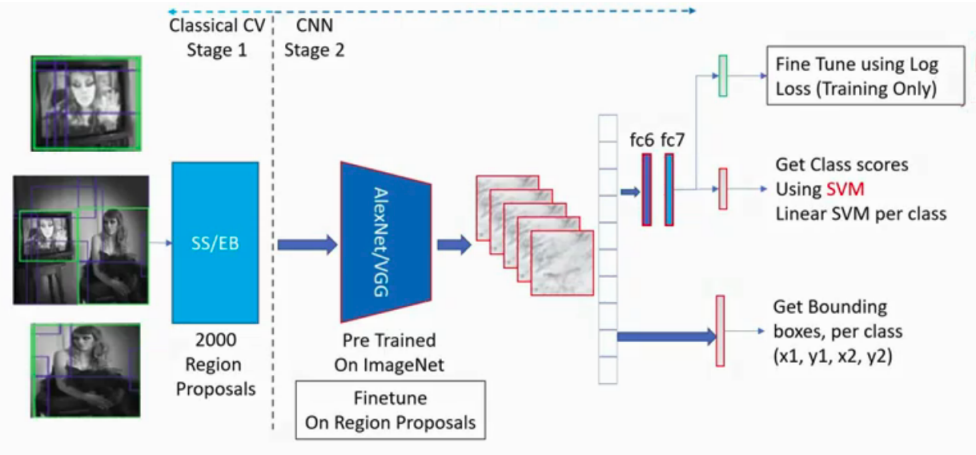


Figure 5.11: **R**egion-based **CNN**s(RCNN)
*Source: https://www.youtube.com/watch?v=7VkJClP9vJg*

The second method we present here is based on Region-proposal CNN[8]. We propose to use the method, algorithm and architecture similar to that

used in any object detection pipeline. Here, the number of output classes are 3 - With Mask, Without Mask and Background.

The primary motive of shifting to CNNs is the advantage of feature engineering which is taken care by the convolutional network. The fully connected layers are used for classification purpose. The conventional procedure of feature extraction is by visually observing the data and using heuristics to derive the best set of features for classification. As we have seen in the previous method, the process of feature engineering is highly laborious in order to build system which are highly fool proof. Thus, the only way around is to give away the task of feature extraction to the CNNs, which learns lower level features in the initial layes of the network and higher level features in the deeper layers of networks.

### 5.2.1 Region Proposal

The first step in the pipeline is the region proposal stage. Various classical computer vision algorithms can be used for the same, such as - Selective search, Edge boxes, Superpixels straddling, Sliding window, and many alike. The most popular ones are Edge boxes and Selective search owing to their high accuracy. The edge boxes detects regions by relying on contours rather than edges. In case of selective search algorithm, various attributes of the images such as color, texture, composition, hierarchy, etc., are used for detecting regions. In case of superpixels straddling, a predefined threshold is defined and used to group the pixels. The higher the threshold for similarity of the pixels, more is the number regions proposed and vice versa.

Thus, the process of image localization and region proposal requires use of heuristics to achieve the same.

Further, the use of region proposal techniques eliminates the need for use of sliding window technique along with image pyramid, which is computationally expensive method.

### 5.2.2 Object Detection and Localization

To improve the accuracy of object localization, post region proposal stage can be achieved using bounding box regression. During the training phase, the bounding box regressor is trained with L2 loss to arrive at the coordinates, width and height of the bounding box. Similarly, the last layer of fully connected network is modified to produce 4 values corresponding to

the x-y coordinates and the width and height of the bounding box. The confidence for the output to belong to one of the classes is obtained as the output of the softmax layer.

#### Confidence Scores

The algorithm outputs the probability values of an object being present in a given region, which is called the confidence score. The threshold for the detection of the object is placed through the confidence score. A very low threshold, then the sensitivity will be high where as the specificity will be low and vice-versa. Thus, there exists a trade-off between specificity and sensitivity.

#### Problem of Multiple Detections

The model can output more than one bounding boxes for representing a detected object. Thus, to filter them out and consider the best among them, the techniques of **Non-Max Suppression** can be used. It computes the ratio of intersection to union (**IoU**) of the bounding boxes and considers the one with highest value as the best possible localization of the object.

### 5.2.3 Feature Extraction

The convolutional network of the model acts like generic feature extraction layer. Any pre-trained architectures for image classification such as AlexNet, VGG-16, GoogleNet and like can be used by fine tuning the parameters. During fine tuning not much of the weights in the convolution layer get updated, instead those in the fully connected layer is the one which gets updated and fine tuned for a different type of classification purpose.

## 5.3 Limitations

The following are the set of limitations for the above described method[9]:

1. One ends up with too many inputs to the localization network.
2. It does not exhibit real time performance.
3. The selective search algorithm is a fixed algorithm with no possible learning. This could lead to bad region proposals.

### 5.3.1 Advantage

Also, since one is not processing the background regions of the image, and only processing those regions of the image where the probability of the object being is high, which in a way is eliminating certain possible false positives from the output.

## 5.4 Method-3: YOLO

The third method we present here, which is also the state of the art technique in object detection, is - You Look Only Once (YOLO). YOLO is a regression based algorithm.

### 5.4.1 Working of YOLO

In YOLO, one split an image into an SxS grid and within each of the grid one takes "m" bounding boxes[9]. For each of the bounding box, the network outputs a class probability and offset values. The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image. Anchor boxes are used to increase the accuracy of object detection.

The output vector of YOLO algorithm for each grid cell is a 5+n dimensional vector, where the first five elements of the vectors correspond to - confidence score (of object being present in that cell) and rest four are bounding box coordinates, width and height, all normalized with respect to the grid cell size. The rest of n elements correspond to the probability distribution over the set of possible classes[10].

If two or more grids contain the same object then the center point of the object is found and the grid which has that point is taken. To get the accurate detection of the object IoU and Non-Max Suppression techniques are applied. If the value of IoU $\geq$ threshold value, then it's considered as a good prediction.

### 5.4.2 Performance of YOLO

YOLO is observed to be orders of magnitude faster than other object detection algorithms and has real performance, even on images with multiple objects. This is because YOLO looks the image completely at once

only, avoiding the problem of multiple inputs to the network. Yet, YOLO's performance in detecting small objects within the image is poor. It makes localization errors but predicts less false positives in the background.

### 5.4.3 Application of YOLO for face mask detection

YOLO algorithm can be applied to achieve face mask detection. Again the same three class classification can be used (with/without mask and background). As mentioned above, the performance of YOLO on small sized objects in an image is poor. This can be overcome by use of high resolution cameras and increasing the input image dimensions to the model. We propose another method to overcome the same:

1. Train a model for human face detection, which is pose, scale, illumination and alignment invariant (i.e., 2 class classification - face and background)

2. Crop out the face images and use the same for detecting face mask separately with use of a separate classifier/model.

3. In case of one employing this method, it is possible to integrate a well trained face recognition model along with the system to aid in notifying the violators.

The major limitation of above proposed method would be high time complexity. Thus, it finds application only when the average time for which a person stays in the field of view of the camera is higher than the model prediction time. YOLO alone suffices for surveillance purpose while the above described pipeline is useful for penalizing the violators at a later point of time (provided the database includes the person in question).

# Chapter 6

# Conclusion

The proposed face mask detection (and face recognition) system, described in terms of algorithms and block diagrams, is designed to incorporate high invariance to illumination, scale, pose, image quality, and alignment. The system so described can be improved to include heuristics to detect face masks based on color and texture. Also, with the invent of jaw-like appearing face masks make the face mask detection process more difficult, which needs to be overcome by designing better architectures, classification network and also applying algorithmic approach. Also, through-out the above discussion the input images is assumed to be an RGB image, which might not be the case when the image source is a Black and White surveillance camera. This can be overcome by conversion of Gray scale images to RGB images using OpenCV's module functions. A method based on Eigen-Jaw method was proposed with the proof of principle implementation results presented. The system so presented is limited for use when the input is frontal face images and hence, is not pose invariant. It suffers from the problem of high false positives. It is expected to be overcome with use of better region selection algorithms and use of RCNN. Since RCNN does not have real time performance, to achieve the same, YOLO algorithm is proposed to be used with three output classes including the background class. The possible methods to address the problem of poor performance of YOLO on small sized objects has also been discussed.

# Bibliography


[1] *https://www.wired.com/story/algorithms-recognize-masked-face/*

[2] *https://github.com/didi/maskdetection*

[3] *https://ieeexplore.ieee.org/document/8888092*

[4] *https://www.leewayhertz.com/face-mask-detection-system/*

[5] *https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/*

[6] *https://issivs.com/facemask/*

[7] *https://sefiks.com/2020/02/23/face-alignment-for-face-recognition-in-python-within-opencv/*

[8] *Evolution Of Object Detection Networks - Cogneethi: Youtube playlist*

[9] *https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e*

[10] *Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam, "Real-Time Object Detection with Yolo", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-3S, February 2019*