VISUAL RECOGNITION GROUP PROJECT -1

Mudit Goyal (IMT2018045) Omkar Lavangad (IMT2018053) Parithimalan A. (IMT2018055)

I. Human Detection

The detection of objects in an image is done using YOLO (You Only Look Once) algorithm with pre-trained weights. YOLO trains and tests on full images and directly optimizes detection performance. The YOLO model has several benefits over other traditional methods of object detection like the following.

- It is extremely fast. Since frame detection in YOLO is a regression problem there is no need for a complex pipeline. We can simply run our neural network on any new image at test time to make predictions.
- It sees the entire image during training and testing unlike other sliding window algorithms which require multiple iterations to process a single image.
- It learns generalizable object representations. When trained on real time images and tested, YOLO outperforms top detection methods like DPM and R-CNN.
- YOLO network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously.This means our network reasons globally about the full image and all the objects in the image. The YOLO design enables end-to-end training and real time speeds while maintaining high average precision.

Working of YOLO

1) First it divides the input image into an S × S grid as shown in fig.1.
2) If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.
3) Each grid cell predicts B bounding boxes and confidence scores for those boxes as shown in fig.2.
4) These confidence scores reflect how confident the model is that the box contains an object. If no object exists in that cell, the confidence scores should be zero.
5) Each grid cell also predicts conditional class probabilities.
6) These probabilities are conditioned on the grid cell containing an object. We only predict one set of class probabilities per grid cell, regardless of the number of B-boxes.
7) Finally, we multiply the conditional class probabilities as shown in fig.3 and the individual box confidence predictions which gives us class-specific confidence scores for each box as shown in fig.4.

Fig. 1. Divide the image into S×S grid

Fig. 2. Calculate bounding boxes and confidence score for each box.

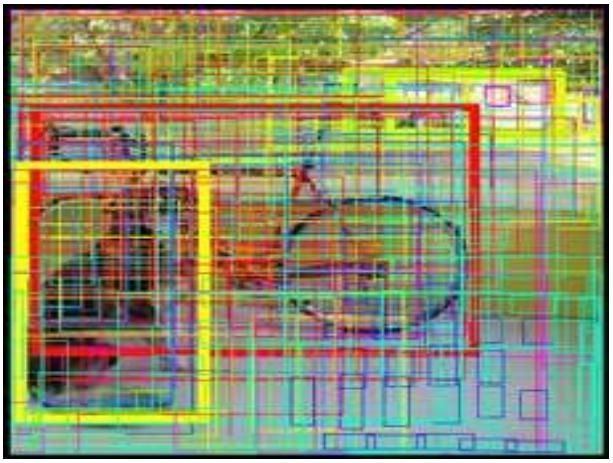

Fig. 3. Multiply probability and confidence scores.

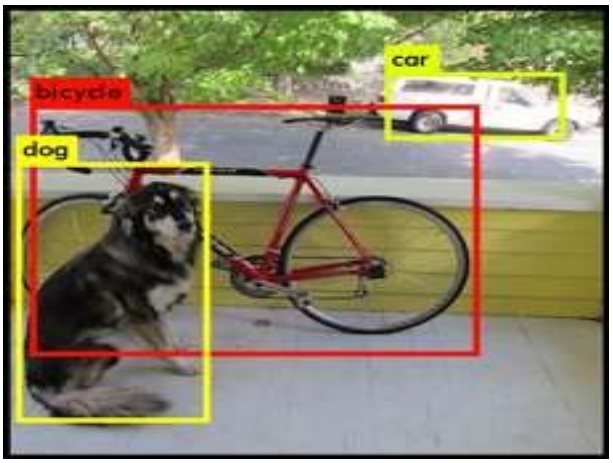

Fig. 4. Final Output.

II. FACE DETECTION

VIOLA JONES

Object Detection using haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones. Here, we use Haar Cascade Classifiers for Face Detection.

Haar Cascade Classifiers: : For face detection, we use a classifier that is trained on a lot of positive images (images of faces) and negative images (images without faces). We use the following 3 Haar-like features to extract points of interest that help in classification. As shown in fig. 5.

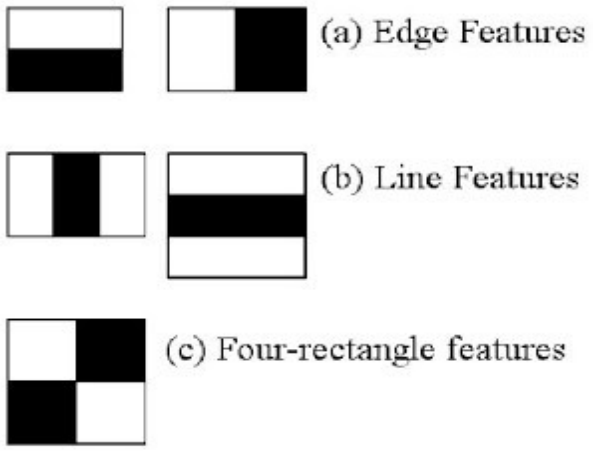

Fig. 5. Haar Features.

Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle. In an image, most parts are non-face regions. For this they have introduced the concept of Cascade of Classifiers.The image is broken down into windows. The features are grouped into different stages of classifiers and applied one-by-one on each window. If a window fails the first stage,it is discarded. The window which passes all stages is a face region. The number of features that are present in the 24×24 detector window is nearly 160,000, but only a few of these features are important to identify a face. So we use the AdaBoost algorithm to identify the best features in the 160,000 features. If all classifiers approve the image, it is finally classified as a human face. In our code we are using the OpenCV's Haar Cascade Classifier. OpenCV's Haar Cascade Classifiers : OpenCV's algorithm is currently using the following Haar-like features which are the input to the basic classifiers. As shown in fig. 6.

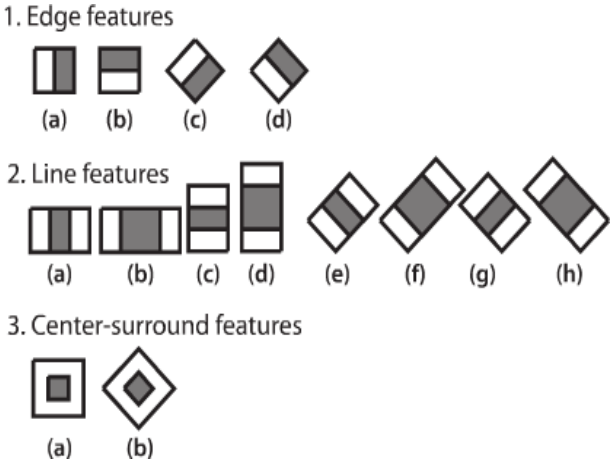

Fig. 6. Cascade Classifier Features.

The CascadeClassifier function is used on images with the following parameters.

- image : Contains image on which detection is done.
- scaleFactor : Determines by how much the image size is reduced at each scale. We are using a scaleFactor of 1.1
- minNeighbors : How many neighbors each candidate rectangle should have to retain it. We are using 2 as the minNeighbors

III. Mask Detection

Transfer learning : It is used in machine learning, is the reuse of a pre-trained model on a new problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another.
We collected a dataset which has images of people with masks and without masks. We used this dataset to train a model that can classify whether the detected face from the cascade classifier is wearing a mask or not. For the model we use Mobilenetv2 as the feature extractor and then train a multilayer neural network using the extracted feature.

Mobilenetv2

It is a convolutional neural network that is 53 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. MobileNet is a streamlined architecture that uses depth wise separable convolutions to construct lightweight deep convolutional neural networks and provides an efficient model for mobile and embedded vision applications. It is lightweight and fast.

Neural network architecture: We use 5 fully connected layers using relu activation and have 2 dropouts in between and use softmax on the last layer.

IV. Results

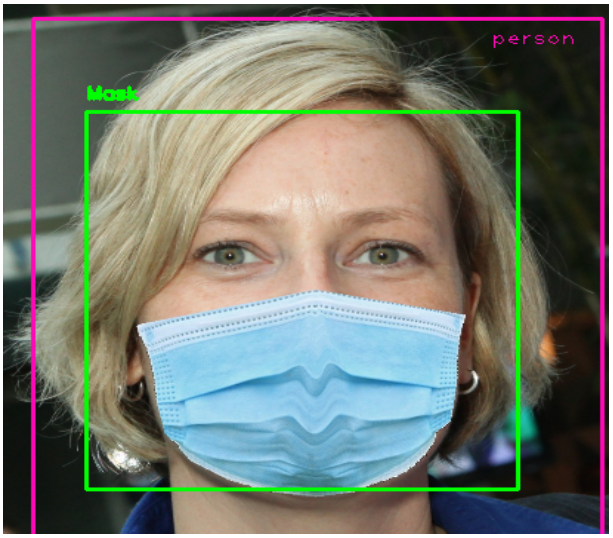

Fig. 7. Detecting a Masked Person.

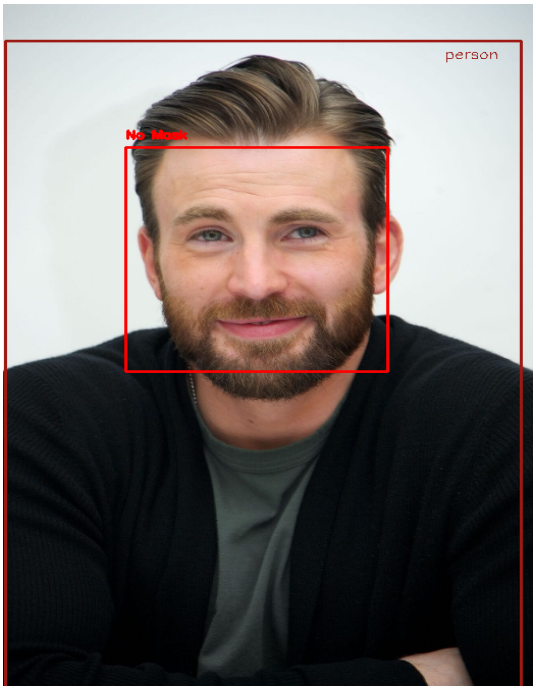

Fig. 8. Detecting a Non- Masked Person.

V. Final System and Performance

The final system uses OpenCV to access camera frames from the device camera. It passes each of the frame through a pipeline of the 3 modules to detect whether a person is wearing a mask or not.
A brief overview of how the system works: First it detects all the humans in each frame of the video using YOLO. Then we detect the faces of the people through Viola Jones and pass this face as an input to a classifier to detect whether the person is wearing a mask or not. Our Classifier consists of MobileNetV2 model as a feature extractor and pass it through a deep neural network. The system will open the door only if a person is detected wearing a mask.

The system is fairly Real Time. The Validation Accuracy of our classifier is 98.62 % . (On a validation dataset of 61 masked faces and 61 non-masked faces.)

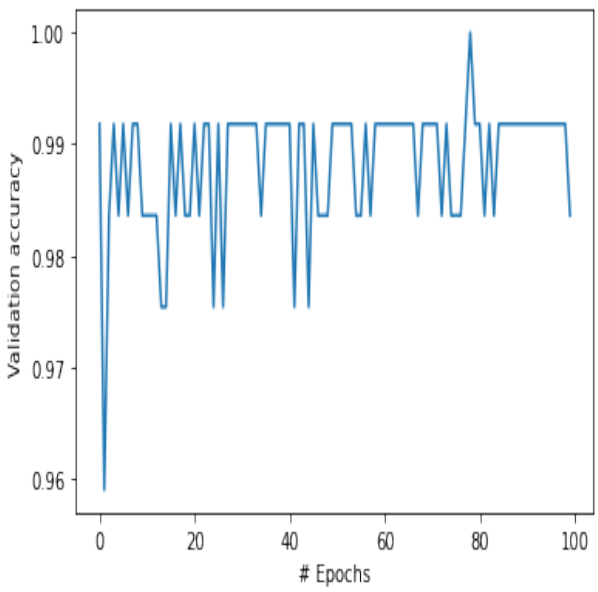

Fig. 9. Validation Accuracy.

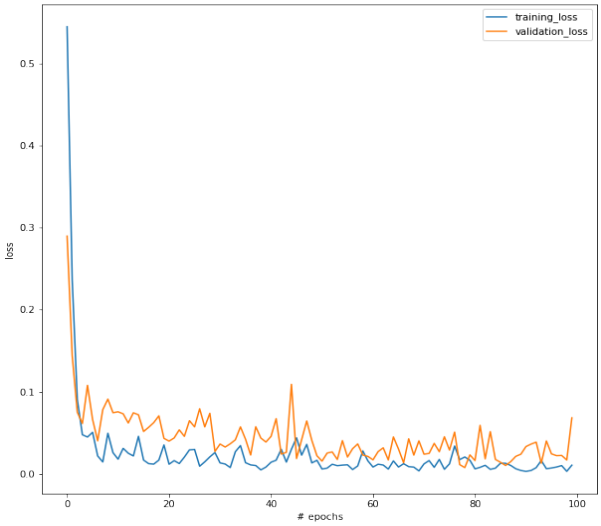

Fig. 10. Training Loss and Validation Loss vs Number of epochs.

VI. Conclusion

We have created a system which detects if a person is wearing a mask or not. It can be used for surveillance such that a person can only enter certain rooms if he/she wears a mask. We can use the video feed of cctv as an input to our system and see if the person who wants to enter a room is wearing a mask and only allow the door to be unlocked if the person is wearing a mask.

VII. Acknowledgment

We would like to thank Professor Dinesh Babu Jayagopi as well as all the Teaching Assistants – Kotha Tejas, Vibhav Agarwal, Sarthak Khoche, Rahil Vijay, Tanmay Jain and Swasti Shreya Mishra .

VIII. References

https://github.com/JadHADDAD92/covid-mask-detector
https://github.com/rakshit087/Face-Mask-Detection-PyTorch
http://www.learningaboutelectronics.com/Articles/How-to-save-a-video-Python-OpenCV.php
https://github.com/nandinib1999/object-detection-yolo-opencv
https://www.kaggle.com/ciplab/real-and-fake-face-detection
https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset/tree/master/RWMFD_part_1/0001