International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

TO DEVELOP AN ASSISTANT TO AUTOMATE HOMES AND OFFICES USING DEEP LEARNING-BASED HUMAN AND FACEMASK DETECTION MODEL IN AN IOT ENVIRONMENT

Francis J Kalliath[1], Nikhil R[2], Murali Krishnan M P[3], Aatish Basavaraj Mundasad[4], Vinayak P Balehittal[5]

[1-5]*Mechanical Dept., New Horizon College of Engineering, Outer Ring Road, Kadubeesanahalli, Bengaluru, Karnataka 560103*

Abstract - *This project aims to take home automation to the next level by bringing in the power of computer vision. Here we develop a deep learning model for Human and Face Mask Detection using Supervised Learning Techniques using SSD, MobileNet along with Image Augmentation on various datasets including coco dataset for training the Prediction Model and then using Raspberry Pi to initiate the home automation tasks to control the room lights, fans, window blinds, water geyser and coffee maker this reduces the surface contact between humans and switches which can help in controlling the spread of viruses. This model also detects the people not wearing masks and provides immediate warning using Face Mask Detection Algorithm providing safety to the people in the room. We are using 12,000 image datasets. Our accuracy is 96.54% on train dataset, 84.14% on test dataset which is split into 20% test and 80% train.*

Key Words: MobileNet ,SSD ,CNN ,IoT , Human Detection

1.INTRODUCTION

With the rapid development of technologies such as smart phones and drones and more embedded devices have been endowed with computer vision function. Artificial Intelligence (AI) and other functions in the smartphone need to accurately locate the object position in the image. Object detection might be the most common one that is adopted as a basic functional module for scene parsing in embedded applications, and hence it has been the area of increasing interest in the current trend.

Object detection is the basic premise for advanced visual tasks such as Video Content Recognition and Image Recognition. Some traditional object detection methods have been eliminated with the advent of neural networks because of their low accuracy and high computational time. Due to the growth of computing power and the dedicated deep learning chips and the availability of large-scale labelled samples (e.g., ImageNet, SSD-MobileNet), Convolutional Neural Network(CNN) has been extensively used due to its fast, scalable and end to-end learning framework. In order to maintain high accuracy and low computational time, these methods rely on powerful GPU computing power. However, such improvement in accuracy with heavy computational cost may not be helpful to face the challenge in many real-world applications that require real-time performance carried out in a computationally limited platform. Recently, fast single-stage object detectors have been proposed such as YOLO, SSD, RetinaNet etc. Among them, SSD MobileNet v3 is extensively used for real world applications as the accuracy and speed are well balanced. Despite that, SSD MobileNet v2 requires heavy computational power and large run-time memory to maintain good performance. By reducing the size of the model, it can run the object detection on embedded systems. The most popular method to reduce the model size and floating point operations (FLOPs) without notably reducing detection accuracy is to reduce the parameters and model size of the network by redesigning a more efficient network. For example, SqueezeNet, MobileNet, etc., these methods can maintain detection accuracy to a great context with significantly reducing the model and TensorFlow-lite, we have redesigned a light weight structure without notably reducing the detection accuracy. Also, by using MobileNet we have reduced the size and FLOPs which helped in building a more efficient network to run the object detection on embedded systems.

1.1 MultiBox Detect

After going through a certain of convolutions for feature extraction, we obtain a feature layer of size m×n (number of locations) with p channels, such as 8×8 or 4×4 above. And a 3×3 conv is applied on this m×n×p feature layer. For each location, we got k bounding boxes. These k bounding boxes have different sizes and aspect ratios. The concept is, maybe a vertical rectangle is more fit for human, and a horizontal rectangle is more fit for car. For each of the bounding box, we will compute c class scores and 4 offsets relative to the original default bounding box shape.
Thus, we got (c+4) kmn outputs.

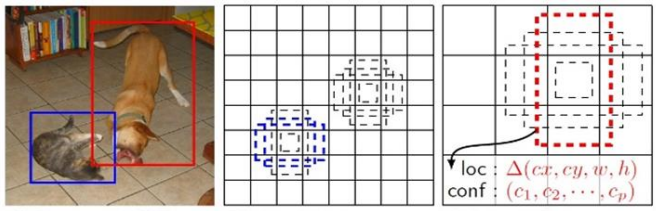

Figure 1. Multibox detect with feature maps

1.2 SSD Network Architecture

To have more accurate detection, different layers of feature maps are also going through a small 3×3 convolution for object detection as shown above.

Say for example, at Conv4_3, it is of size 38×38×512. 3×3 conv is applied. And there are 4 bounding boxes and each bounding box will have (classes + 4) outputs. Thus, at Conv4_3, the output is 38×38×4×(c+4). Suppose there are 20 object classes plus one background class, the output is 38×38×4×(21+4) = 144,400. In terms of number of bounding boxes, there are 38×38×4 = 5776 bounding boxes.

Similarly for other conv layers:

Conv7: 19×19×6 = 2166 boxes (6 boxes for each location)

Conv8_2: 10×10×6 = 600 boxes (6 boxes for each location)

Conv9_2: 5×5×6 = 150 boxes (6 boxes for each location)

Conv10_2: 3×3×4 = 36 boxes (4 boxes for each location)

Conv11_2: 1×1×4 = 4 boxes (4 boxes for each location)

If we sum them up, we got 5776 + 2166 + 600 + 150 + 36 +4 = 8732 boxes in total. For, YOLO, there are 7×7 locations at the end with 2 bounding boxes for each location. YOLO only got 7×7×2 = 98 boxes. Hence, SSD has 8732 bounding boxes which is more than that of YOLO.

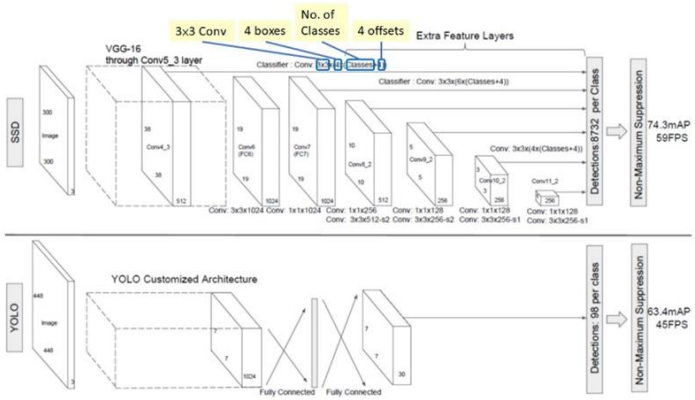

Figure 2. SSD and YOLO architecture comparison

2. Methodology

1. We collected dataset from our college premises that assists us to detect the presence of humans and if people are wearing masks in the frame from live feed cameras. Datasets are collection of data represented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset. It is a curation of data collected in a single file used to train the machine learning model. For our application we chose to use the Mobilenet SSD pretrained model trained on coco dataset for human detection. The basic requirement of the dataset is to contain images which has people and some without them. This can be used to train a detection model to understand if there is a human or not in each given frame. The second model developed by us is a face mask detection model developed from scratch using a dataset we collected for college. This dataset consists of images taken of various people in different lighting condition and various places. This was for training the model to check if a person in given frame is wearing a mask or not.

2. We created a model using convolution neural networks implementing transfer learning along with image augmentation of all forms. MobileNet architecture is being used to obtain transfer learning because of its light-weight nature. Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. Transformations of image include a range of operations from the field of image manipulation, such as shifts, flips, zooms etc. For the architecture of model, we have used Mobilenet since it very light and can easily run-on mobile platforms such as phones and raspberry pi.

3.The model obtained is trained on a mobile platform such as Raspberry Pi. The model is then tested for accuracy in test dataset and train dataset. The test accuracy obtained was 84.14%.

4.If the human is being detected by the highly efficient model, then the lights and fans controlled by Raspberry Pi is turned on. If the human is detected on the left side bounding box by the model running on Raspberry pi, then the green lights representing the area of the room where the human is present is turns ON by sending a high signal to the lights which is connected to the pin 3 on raspberry pi. If none is detected then a low signal is sent to the lights and they are turned off. In the same way the if the human is detected on the right bounding box the red light on the raspberry pi connected to pin 6 is turned on.

5.Obtain datasets for position and posture of humans and identify the activity using highly efficient detection model run on Raspberry Pi. Datasets are obtained for training the model to identify the position and activity of human and run on the Raspberry Pi.

6.This model uses a 5-megapixel camera, a night vision camera, DHT11 sensor, MQ-2 Gas Sensor, Photoresistors, led lights and resistors to assist the model with addition information to predict any anomalies in the surroundings.

7.The model at the end of the week generates a report on the daily activity of individuals.

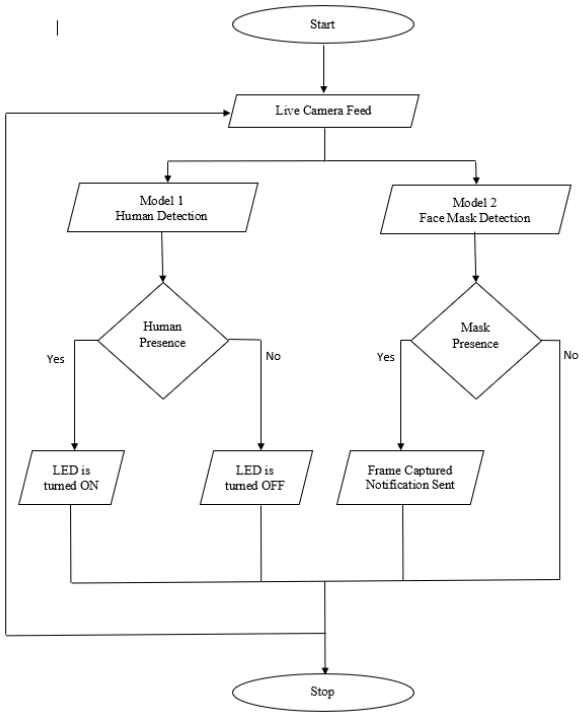

Chart -1: Flow chart for the Project

3. Result

The first model is a Human detection model that has been implemented in raspberry pi hence we have used pretrained model SSD MobilNet v2 model trained on coco dataset which has been converted to tf-lite format so the weights are stored in a light weight format to felicitate the pi to run the model providing a framerate of 2.5 fps.
In the following images the frame has been divided into two parts left box represents the left area of room and right box represents the right area in the room. Is a person is detected on the left box then the green light on the raspberry pi is turned on (Figure 1 and 2) and in case the person is detected on the right side of the room the red light is turned on(Figure 3 and 4).

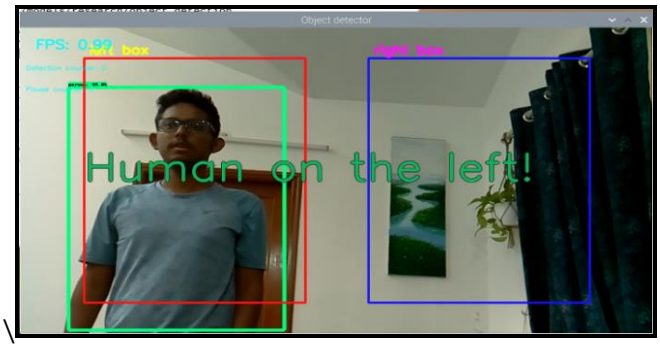

Figure 1. Bounding box with human on left

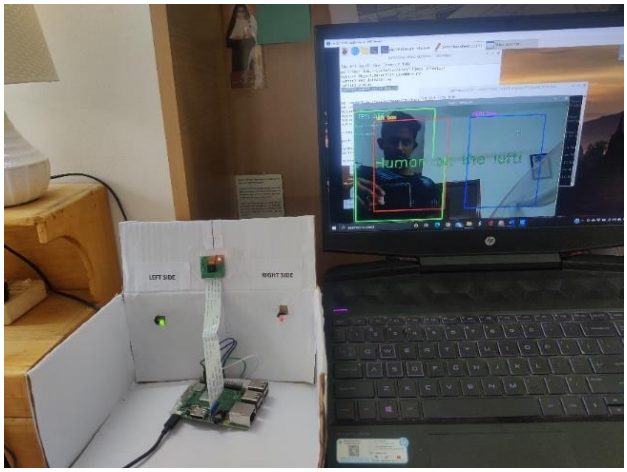

Figure 2. Green LED is turned ON when the human is detected on the left

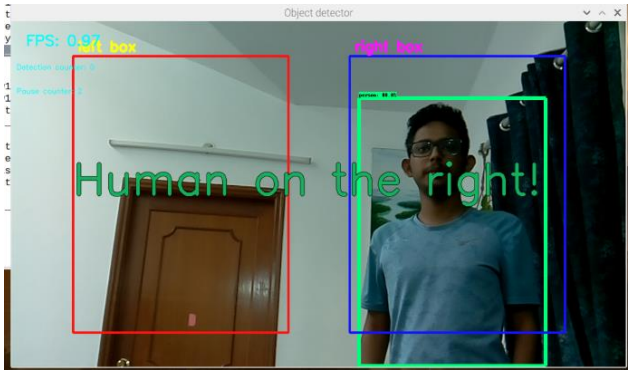

Figure 3. Bounding box with human on right

Figure 4. Red LED is turned ON when the human is detected on the right

The second model is Mobilenet v2 model that was developed from scratch with the dataset we collected having 1000 images (500 with mask and 500 without mask) with image augmentation for mask detection along with transfer learning, we are using the following methods to determine the results:
AP (Average Precision) and mAP (mean Average Precision), which are commonly used in the field of target detection, are

used to evaluate the detection quality of face occlusion by SSD algorithm. The calculation method is as follows:

$AP = \int precision\ (recall)\ D\ recall$

$mAP = \frac{\sum AP}{N(class)}$ (in Table 2).

Among them, precision and recall are commonly used evaluation indexes in dichotomies, and are defined as follows:

$\text{Precision} = \frac{TP}{(TP+FP)}$

$\text{Recall} = \frac{TP}{(TP+FN)}$

TP (True Positive) refers to the positive samples predicted by the model as positive, FP (False Positive) refers to the negative samples predicted by the model as positive, FN (False Negative) refers to the positive samples predicted by the model as negative, TN (True Negative) refers to the negative samples predicted by the model as negative.

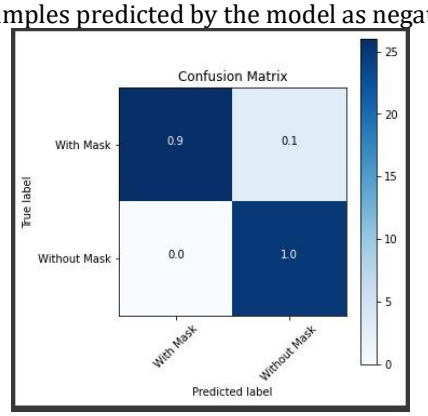

Figure 5. Confusion matrix

```
Confusion Matrix
Normalized confusion matrix
Classification Report
              precision    recall  f1-score   support

   With Mask       1.00      0.79      0.88        29
Without Mask       0.81      1.00      0.89        25

    accuracy                           0.89        54
   macro avg       0.90      0.90      0.89        54
weighted avg       0.91      0.89      0.89        54
```

Table 1. Values of Precision and Recall

The Validation accuracy and the Validation loss of the model obtained over a validation dataset of 500 images is:

Table -1: Model FMDM

Validation accuracy	Validation loss
0.92	0.113

The result graph is as follows:

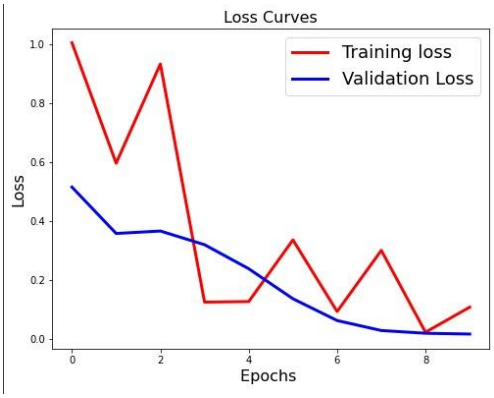

Figure 6. Training Loss and Validation Loss

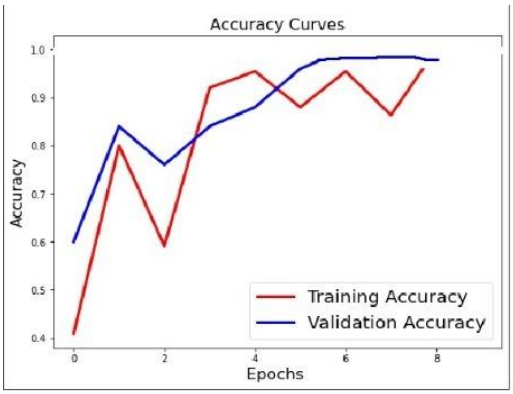

Figure 7. Training Accuracy and Validation Accuracy

Figure 7 shows that the train accuracy and validation accuracy increase with epochs. The validation loss and test loss decrease with epoch and stabilises.

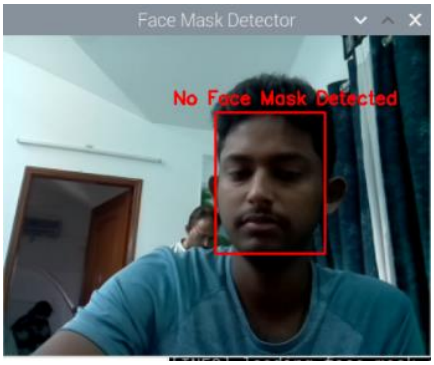

Figure 8. No face mask detected when the mask detection model is run on live stream with help of raspberry pi

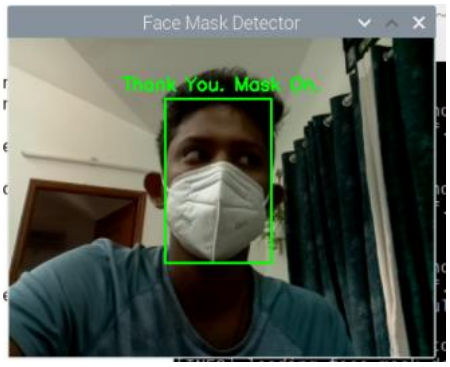

Figure 9. Face mask detected when the mask detection model is run on live stream with help of Raspberry Pi

Figure 8 and 9 shows the performance of the mask detection model on live stream from pi camera of the raspberry pi.

Figure 10. Multiple mask detection with and without mask

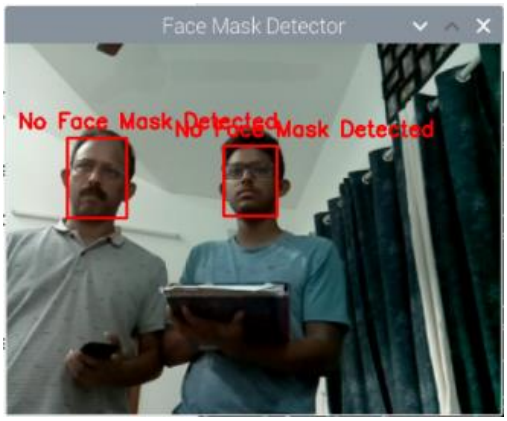

Figure 11. Multiple mask detection with and without mask

Figure 10 and 11 shows the performance of the mask detection model on live stream from pi camera of the raspberry pi when more than one human is present in the frames. The above results prove that using the first model we were able to detect humans and turn on lights in a raspberry. Using the second model we created from scratch, we were able to detect peoples face to detect the presence of mask or not. This model provided us with 84.14% test accuracy.

3. CONCLUSIONS

In this project we have automated the process of turning on and off the lights, fans and other lighting modes for a home or office environment thus saving the electricity and giving complete freedom to an individual while moving in and out of a room. This system will be a mandatory system in the post corona age. As this system will reduce the transmission of virus and other infection caused by touching common switches, thus reducing surface contact between contaminated switches and humans. This model can also detect the presence of face masks on humans and give a customizable response if required. It also can execute custom tasks like turning on the coffee maker, geyser etc. when the model detects your presence in an area and provides one with complete assistance in its presence all of which has been implemented in a raspberry pi environment.

REFERENCES

[1] W.T. Chu, W.W. Li, (June 2017) "Manga Face Net: Face Detection in Manga based on Deep Neural Network," Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval pp. 412-415.

[2] J. Deng, W. Dong, R. Socher, L.J Li, K. Li, L. Fei-Fei, (2009) "Imagenet: A large-scale hierarchical image database," in Computer Vison and Pattern Recognition IEEE Conference on, pp. 248- 255.

[3]] K. Simonyan, A. Zisserman, 2015 "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations.

[4] Szegedy, C., Reed, S., Erhan, D., Anguelov, D (2015) Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441 v3.

[5] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille (2015) Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR.

[6] . Bell, S., Zitnick, C.L., Bala, K., Girshick, R (2016) Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks.

[7]] Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S. Awwal, Vijayan K. Asari (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. .arXiv preprint arXiv: 1803.01164

[8] Francois Chollet (2017). Xception: Deep Learning with Depthwise Separable Convolutions. arXiv preprint arXiv: 1610.02357

[9] S. Ioffe and C. Szegedy. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167

[10] OECD (2015), "Students, Computers and Learning: Making the Connection", PISA, OECD Publishing, Paris.

BIOGRAPHIES

Francis J Kalliath: He is a final year engineering student who is very passionate about deep learning and has completed multiple projects in computer vision. He if focused on solving the real-world problems using technologies like artificial intelligence and IoT

Nikhil R: He is a final-year mechanical engineering student at the New Horizon College Of Engineering. He is interested in the Computer Vision field and its application in various fields such as mechanical, medical, etc. He is an avid enthusiast in implementing the computer vision model on mobile platforms such as raspberry pi, Jetson nano, etc.

Murali Krishnan M P: He is a final year mechanical engineering student at New Horizon College of Engineering. He has done various courses on coursera and have completed various projects based on deep learning and artificial intelligence.

Aatish Basavaraj Mundasad : He final year mechanical engineering student. Curious about Tech and Gizmos of the modern era. Always curious to know and learn the new Tech and work on it.