3rd International Conference on
Robotics Automation and Non-Destructive Evaluation,
Chennai, India, 23 April 2022.
https://doi.org/10.13180/RANE.2022.23.04.33

# COVID-19 REAL-TIME PRELIMINARY DETECTION FOR ENTRANCES OF PUBLIC PLACES

RUSHIL PATEL, AMEYA RANADE
*Information Technology, Mumbai University, Bhavans Campus, Munshi Nagar, Andheri West Mumbai, Maharashtra, India*
*E-mail: rushil.patel@spit.ac.in, ameya.ranade@spit.ac.in*
*www.spit.ac.in*

MIHIR NIKAM
*Information Technology, Mumbai University, Bhavans Campus, Munshi Nagar, Andheri West Mumbai, Maharashtra, India*
*E-mail: mihir.nikam@spit.ac.in*

COVID-19 is an unparalleled crisis which completely changed the functioning of our world. The pandemic has also resulted in a host of security issues. People have been advised to follow strict protocols like wearing masks and social distancing to reduce the spread of COVID-19. In the current methods of screening at public places, security personnel are required to measure the body temperature manually using infrared guns. This process can be inefficient due to human error as well as time consumption. Since the face is covered or hidden by a mask at all times, face recognition becomes difficult. This paper suggests systems to implement reliable methods for face recognition as well as body temperature check. This system can be implemented at the entrances of public places such as schools, parks, public transport stations, where people are required to follow COVID-19 protocols. The methodology of screening at public places has changed due to the Covid-19 pandemic. The current system of screening at public entrances entails the presence of at least one person to conduct temperature and vaccination certificate verification in addition to security checks. We thus propose to amalgamate all these checks into one automated system.

## 1. Introduction

This study was held to analyze the screening process at entrances of public places. All public places such as schools, malls, public halls, movie theaters are being checked for preliminary COVID-19 symptoms, mainly high body temperature. Face masks and vaccination certificates are also mandatory for entering these public places. The purpose of this paper is to provide a way in which the screening process can be codified and automated so as to handle the influx of people at public entrances.

This can be achieved by designing a smart system. This system includes 3 important components namely a face mask detection system, a temperature measurement and a vaccination certificate QR code scanner. The first component of the system detects whether the human is wearing a face mask properly. The second component focuses on the human body temperature. The temperature measurement would be carried out using a thermal imaging camera. Security personnel will no longer need to manually measure temperatures by a thermometer gun. A thermal camera can detect temperature immediately compared to an infrared temperature gun as it does not need to send infrared signals. The third component scans the QR code on the vaccination certificate to check the validity of the certificate. If valid, it also checks whether the second dose had been taken 15 days prior. If all the three criteria are fulfilled we can conclude that the person is fit to enter the venue.

## 2. Research Background

### *2.1 Advantages of using FaceNet and MobileNetV2*

A pre-trained deep learning face detector model included with OpenCV can be used to achieve fast and accurate face recognition. The newest version of OpenCV 3.3 contains an improved Deep Nervous Networks (dnn) module.. OpenCV deep learning face recognition uses the Single Shot Detector (SSD) framework with a ResNet based network. It can be utilized with a wide variety of deep learning frameworks, including Caffe, TensorFlow, and Torch / PyTorch.

This algorithm is fast and extracts many features from the image. Next, the most useful features will be nominated via Adaboost. This reduces the number of original features from over 160,000 to 6000 and reduces training time [4]. However, it will still take some time to apply all these features to the sliding window. To remedy this, it introduced a classifier cascade [4] with grouped features. The remaining features of this cascade will not be processed if the first stage window itself fails. If this phase passes, the next features will be tested. This process is repeated several times. If a window can pass through all features, it is classified as a face area. We used the power of OpenCV's deep learning facial recognition model to build a robust and accurate system.

Using MobileNet, a low latency, low power model, resource requirements for different use cases can be adaptably met. This is an effective feature extractor for object detection and segmentation. The MobileNetV2 performance is stronger than MobileNetV1 on a number of benchmarks and tasks across different model sizes. MobileNetV2 [1] is about 35 percent faster than MobileNetV1 on Single Shot Detector Lite detection tasks. The algorithm makes use of depth-wise separable convolutions as a building block so that it is based on MobileNet V1. However, MobileNet V1 is far superior in terms of performance.

The newer version of MobileNet has two new features:

- In experimental research, it has been shown that linear layers are crucial, since they prevent nonlinearities from destroying too much information.
- Shortcut connections between the bottlenecks

### *2.2 Research background for temperature detection module*

Infrared thermometers are traditionally used to scan an area or large group of people with numerous components or objects. The process is tedious and complicated because you must check each one individually. Thermal imaging equipment saves time compared to IR thermometers, and such a system allows operators to inspect a large number of people simultaneously in just seconds. The camera also uses advanced thermal imaging technology to measure temperature from long distances. This facilitates scanning from a safe distance and avoids congestion. Thermal cameras have an efficient solution to monitor the skin temperature of people without direct contact. This is achieved by the thermal imaging camera as the operator not only transmits thousands of temperature values but also converts these measurements into thermal images. These measurements can be analyzed and the accurate temperature of the person can be estimated. Those who are in charge of the device are in different indoor areas, and scientific surveys also demonstrate that the thermal imaging device [7] is usually used to accurately calculate the surface skin temperature.

## 3. Methodology

### *3.1 Detecting the mask*

Internal / external cameras are used to record video input signals to determine if the person is wearing a mask. The frame is extracted from the rendered video and sent as an input to Face-Net (Caffe model). Face-Net Deep Neural network recognizes the human face from the frame that was the original input to the model. After the human face is detected, it is processed (depending on the application with or without mask) and then the Convolutional neural network Mask-Net (MobileNetV2) categorizes whether a person has worn a face mask or not.

### *3.2 Detecting human body temperature*

Thermal imaging cameras are used to collect a limited amount of data only for final system testing, in contrast to internal/external cameras, which cannot provide the datasets needed for the development of this system. We perform this test to measure the accuracy of the program output based on the image captured by the actual infrared camera. We have identified a better camera, however an economic method has been developed to enable prototyping of the proposed method. In order to overcome the above limitations, the next method of developing the system is to simulate the input as if it were captured directly from two different cameras. In this case, the online videos can be used as the input stream instead of streaming the video from the camera, e.g. in the form of RGB videos along with their respective thermal imaging videos. For the purposes of this research, the camera system includes two types of cameras. One is a camera that can capture visible views in RGB representation, and the other is a thermal image camera that can capture thermal images in the form of color images that provide different temperatures values for each colour range. Fig. 9 (a) shows an example of an image captured by a normal visible camera, while Fig. 9 (b) shows a thermal image captured by a thermal camera from the same view. Whereas Fig. 9 (c) shows a manually implemented combination of both types of images that illustrate the same object in different representations.

#### *3.2.1 Image processing over RGB frames in detecting person and other objects*

This subsection describes the operations performed on the received RGB frame rendered from the video. This part achieves the concept of object detection from the frame. Object detection is very important because you want to distinguish between human and non-person objects. This separation allows us to distinguish a person's temperature information from the temperature information of other objects in the same image. For example, if a person has a cup of coffee, we are only interested in the temperature of that person, not the temperature of the coffee cup. In this way, image pixels representing humans can be clearly distinguished from image pixels representing non-humans. The YOLO methods and OpenCV libraries for the Python programming language are used to identify objects in the image. YOLO (You Only Look Once) [5] is a common method of object detection. It is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular for its speed and accuracy. It is used in various applications for detecting people, traffic signs, parking meters and animals. A single forward propagation through a neural network is required to recognize an object. YOLO has excellent learning ability to learn the representation of objects and apply them to object recognition. OpenCV, on the other hand, is a cross-platform library commonly used in the development of real-time computer vision applications. The main focus is on image processing, video recording, and analysis of features such as face and object detection. First, YOLO finds and identifies all objects in the frame. After recognizing the objects, it classifies the objects into

their respective categories. If the identified object is a human, the system calculates the human body temperature from thermal data from a group of pixels representing the human on a thermal frame. This will allow the temperature detection system to function more efficiently and intelligently. Consider a scenario where the detected object is a human and another object, detected very close to the human is hotter than the human. By using the YOLO object detection method, the system distinguishes human body temperature measurements from other objects.

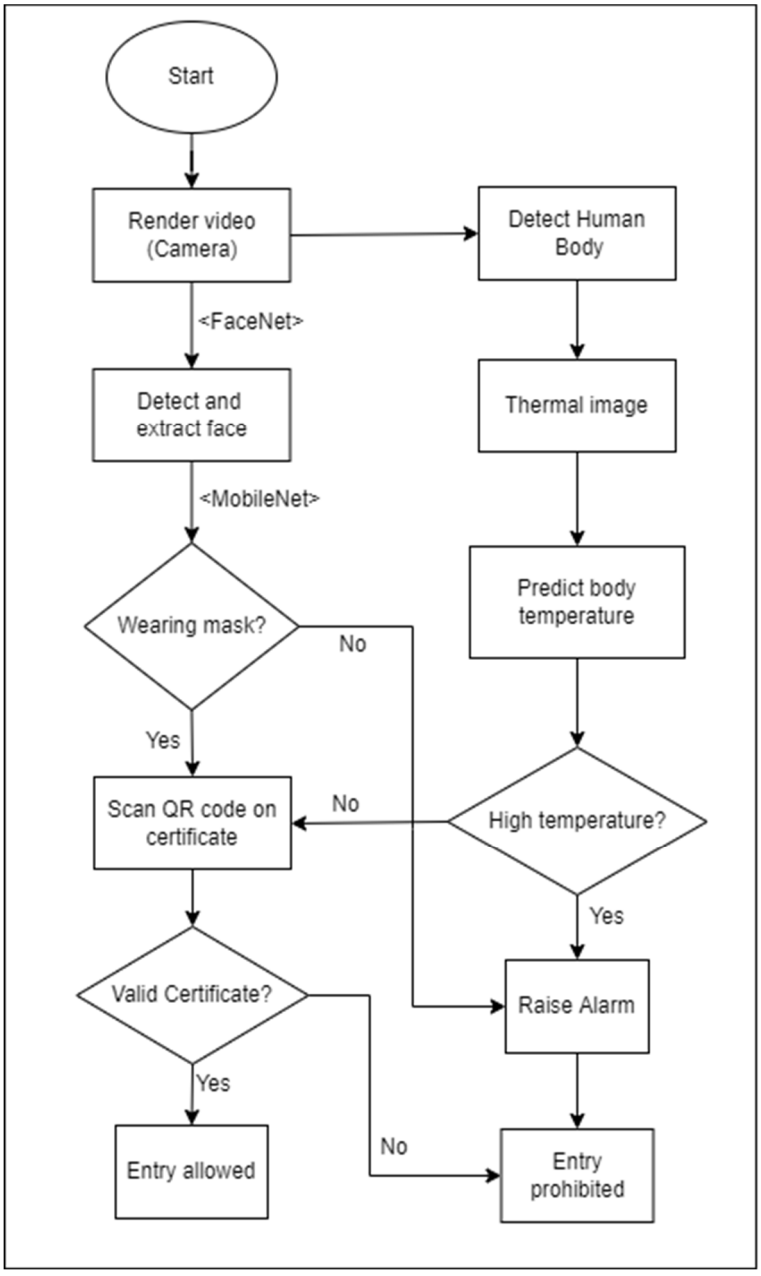



Fig 1. Workflow of the proposed system

### *3.2.2 Processing of images*

On the basis of the YOLO method, this is a method of detecting pixel groups representing people on the visible frame, then on the thermal frame, it is a method of determining the same pixel group with the same coordinates. The main function of the thermal framework is to calculate the actual temperature of the human body, which is done by extracting RGB values from all identifiable pixels in a thermal frame (when rendering a human). Based on Table 1, and using the idea of [10], RGB values are converted into temperature, with r, g, and b values, and temp being in the temperature range. Those four parameters are red, green, and blue values, respectively. The minimum value for r, g, and b is 0, and the maximum value is 255. The conversion values are summarized in the table below. By calculating the average temperature value of all pixels involved, a person's body temperature can be calculated using the conversion method of converting RGB values to temperature values. The program uses this average value to determine if the body temperature of an individual is within normal limits. The guard is immediately notified via an alert that appears on a monitoring screen along with a sound notification if the body temperature is found to be high and outside the normal range. Based on this alert, the guard can then take appropriate action, prohibiting the person from entering the premises.

### *3.2.3 Integration of the two systems*

Overall, by performing image processing on both the visible frame and the thermal frame, the program can detect different objects such as persons, cups, cars, and handbags from the visible frame and in turn will calculate the temperature value on the thermal frame only on the pixels that represent the detected persons only. It will produce output that includes both of these frames as well as bounding boxes that highlight the areas where people are distinguished from other objects in the image. A body temperature alert will be displayed on the output of the display as well, if it detects an abnormal body temperature.

## *3.3 Model Architecture*

### *3.3.1 YOLO*

YOLOv3 (You Only Look Once v3) is discussed in this section of the paper. Deep learning models like YOLO are widely used to classify images, detect objects, and segment them semantically.

Three most prominent types of YOLO [11], they are YOLOv1, YOLOv2 [12], and YOLOv3 [6]. A general architecture has been proposed by YOLOv1. The YOLOv2 improved the bounding box proposal by utilizing predefined anchor boxes. Model architecture and training process were further refined with YOLOv3. So in order to seek out the best results have used YOLOv3 for our proposed system.

Steps for object Detection using YOLO v3 are explained in this section. A square 416 x 416 pixel image in colour is expected as an input from the model. To feed our model, we resize the captured frames to 416 x 416 pixels.After passing this image to the first layer of the YOLO v3 algorithm, this image is then passed to the next layer of the CNN. In order to get an output volume of (19, 19, 425), the last two dimensions of the above output are first flattened. For a grid of 19 by 19, each cell returns 425 numbers. There are 5 number of anchor boxes per grid, 425 = 5 * 85. 5 parameters are *pc, bx, by, bh, bw* respectively. The remaining 80 are the different

classes for detection. Thus, 5 + 80 = 85. So it outputs bounding boxes with class names associated with them. Six numbers represent the bounding boxes, (pc, bx, by, bh, bw, c). The bounding boxes are represented by 85 numbers if c is expanded into an 80-dimensional vector. To avoid selecting overlapping boxes, we apply two algorithms, namely IoU (Intersection over Union) and Non-Max Suppression. In terms of its architecture, YOLO v3 is based on Darknet, a network with 53 layers which was trained on ImageNet. YOLO v3 relies on 106 layers of fully convolutional architecture as its underlying architecture for detection, giving it a total of 106 layers. In YOLO v3, the network is discovered using one by one detection kernels which are applied to feature maps of 3 different sizes on 3 sites. By down-sampling the input image by 32, 16 and 8 respectively, Yolov3 makes predictions at three scales that match the input image. The detection kernel has the following shape: [1 x 1x (B x (5 + C))]. A cell on the feature map is predicted to be able to contain B number of bounding boxes, 5 here means there are 4 bounding box attributes and one object confidence, and C indicates the number of classes to be predicted. he classification loss for each label is calculated based on binary cross-entropy, whereas class predictions and object confidence are predicted by logistic regression.

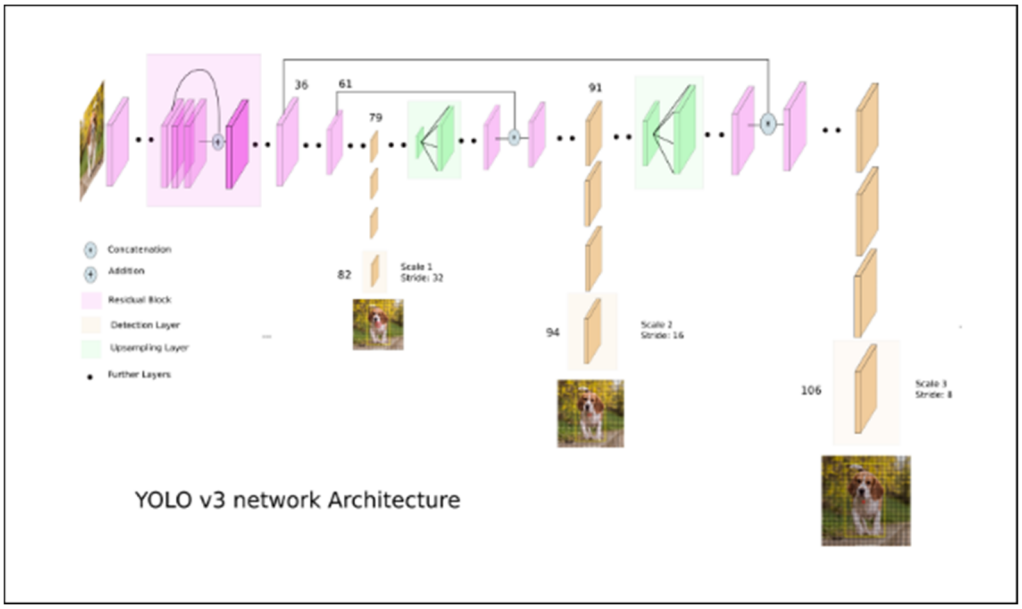



Fig 2. YOLO Model Architecture

### *3.3.2 DARKNET-53*

To detect objects, YOLOv2 incorporated DarkNet-19, a 19-layer network enhanced with 11 more layers to be called DarkNet-19. The detection of small objects in YoLOv2 was poor. By downsampling the input, the layers lose some fine-grained features. Most state-of-the-art algorithms now are incorporating elements that YOLO v2 lacks and this was it's main design flaw. Skip connections, residual blocks and up-sampling were missing in YOLOv2. YOLOv3 incorporates all these seamlessly. A total of 53 additional layers are applied to the task of detection. YOLOv3 is powered by a variant of Darknet built on the Imagenet data set. The architecture contains 106 layers of fully convolutional architecture. While YOLOv3 isn't as quick as YOLOv2 due to this change, this was a necessary compromise. Leaky ReLU Activation and batch normalization is applied following each of the 53 convolutional layers [9]. In the convolution layer, multiple filters are convolutioned on the images. A multiple feature map is

produced, which enables features to be extracted and detected with greater accuracy. This architecture uses only a convolutional layer of stride 2 for down-sampling the images, so there are no pooling layers applied to the input images. By preventing low-level features from being lost, it prevents the loss of useful information often had to do with pooling.

### *3.3.3 BENCHMARKING*

YOLO v3 performs at par with other state of art detectors like RetinaNet while being considerably faster, at COCO mAP 50 benchmarks. Compared to SSD and its variants, this stands out. Here's a comparison of the performances.

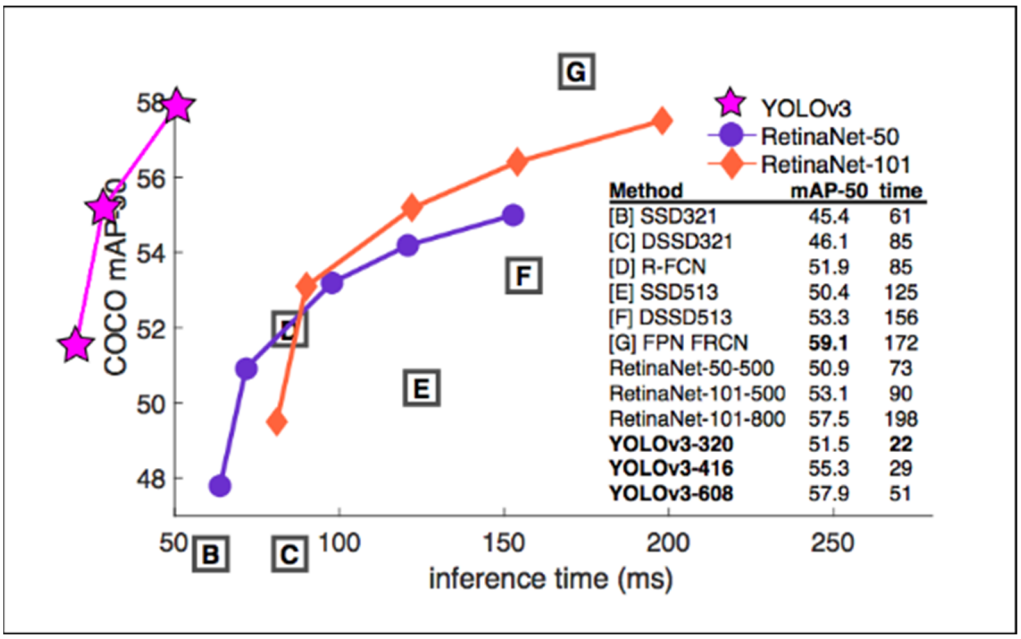



Fig 3. Bench-marking YOLO against other algorithms

## 4. Results and Discussion

This section details the implementation of the project module according to the proposed system workflow. The results produced by the proposed system are then shown and detailed according to various scenarios.

### *4.1 Face Mask Detector*

The application renders video and extracts a frame highlighting the person's face. The two scenarios are tested and analyzed in the module on face mask detection.

1. Without mask

   As seen in Fig 4. (a), the algorithm accurately predicts that the person is not wearing a mask.

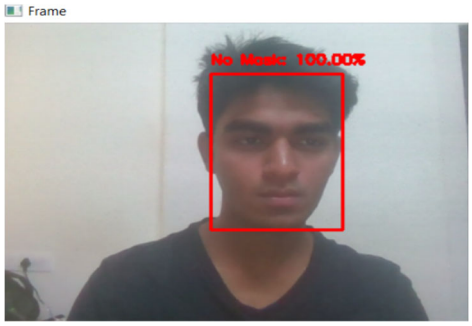



Fig 4 (a) Without mask

2. With mask

   As seen in Fig 4. (b), the algorithm accurately predicts that the person is wearing a mask.

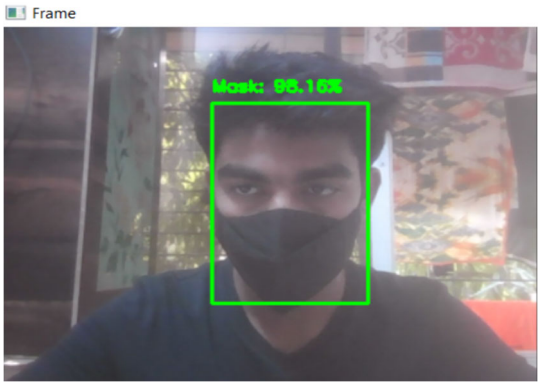



Fig. 4 (b)

**F-score of the face mask detection model:**

Formula-wise, F-score is said to be two times the product of precision and recall divided by the sum of precision and recall. It is also known as the harmonic mean between precision and recall. It is a statistical measure that is used to determine the performance of the model.

$$\boldsymbol{F\ score\ = \frac{2 * precision * recall}{precision\ +\ recall}}$$

In our model, as seen in Fig. 5, we get a F-score of 0.99 which suggests that the rate of performance of this model is extremely good.Loss is a parameter that determines the efficiency of our model. If there are errors in our model, the loss will be high. This means that the model is not accurate. Accuracy, as the name suggests, defines if the predictions done by our model are precise. We mainly calculate this using the elements of the confusion matrix namely, True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

```
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       0.99      1.00      0.99       383
without_mask       1.00      0.99      0.99       384

    accuracy                           0.99       767
   macro avg       0.99      0.99      0.99       767
weighted avg       0.99      0.99      0.99       767

[INFO] saving mask detector model...
C:\Users\Rushil\AppData\Roaming\Python\Python39\site-packages\tensorflo
, the custom mask layer must be passed to the custom_objects argument.
```

Fig 5. f1-score, accuracy and other measurements


Fig 6. Loss/Accuracy Vs Epochs

In Fig. 6, we can see that as the number of epochs increases, our model gets sharper and more accurate.

Overfitting is one of the most common explanations for the poor performance of a predictive model in machine learning. We have created a plot of the model performance on the train and test set can which is being calculated at each point during training epochs. In machine learning, this type of plot is called a learning curve plot, representing one curve for model performance on the training set and the test set for each increment of learning. An accurate analysis of this plot can help us identify overfitting if any. The most common observation suggesting overfitting

that can be made is that the model performance on the training set continues to improve and the performance on the test/validation set is irregular (heavy fluctuations in highs and lows) or is simply degrading after reaching a maximum. But as we can see no such patterns in the plotted graph, it is safe to assume that the model is performing well and it's overfitting. The accuracy of our model is 99\% which suggests that a model gives the right predictions. This ensures the first component of the proposed system works seamlessly.

(a) An example of YOLO detects objects in an RGB frame

(b) Sample image of a thermal frame

Fig 7. Image processing results on RGB frames in detecting people and other objects using the YOLO model

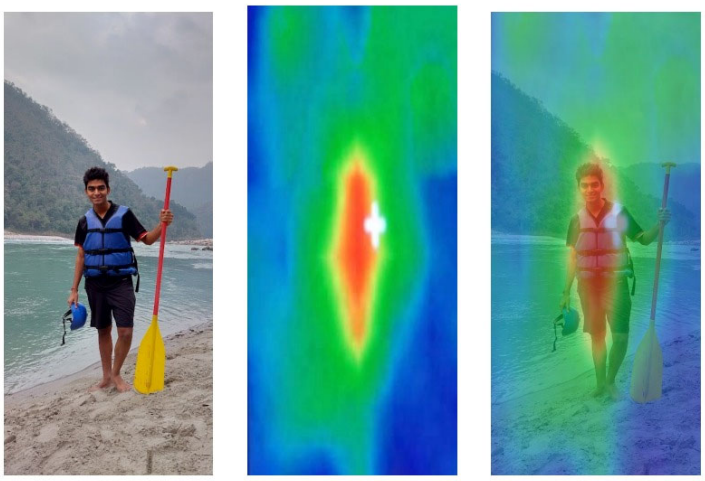
### *4.2 Temperature detection using thermal imaging*

The YOLO model detects all objects present in an image as shown in Fig. 7 and Fig. 8.

A continuous series of images is captured by both cameras and transmitted to a server that runs our programs through a combination of two methods. The image frames are then processed in succession by the program. The first method implemented in the code is to detect objects from the RGB image.

In the second method, temperature is calculated by focusing on the single object detected by the first method. The primary aim of method 2 is to measure the body temperature of the detected person by separating it from the temperature of nearby objects whose pixels may overlap those of the person. After detecting a group of pixels that represents a person using the YOLO method on the visible frame, the next step is mapping to the same group of pixels that represents the person on the thermal frame. A thermodynamic framework is primarily used to measure the temperature of detected human bodies. From all pixels (representing human bodies) of the thermal frame, RGB values are extracted for further calculations. It is composed of four major parameters: red (red), green (green), blue (blue), and temperature (temperature). The conversion values are summarized in Table 1.

To determine a human body's body temperature, the average temperature value of all pixels within an identified frame is used. According to this calculated average value, we predict the human body temperature. The program then determines whether a person's body temperature is within normal acceptable limits.



(a) Example of an image in RGB representation captured by a normal camera

(b) Example of a thermal image captured by a thermal camera

(c) Example of manual merging of visible and thermal image

Fig. 9: Integration of the two systems

Table 1: Conversion of RGB values to temperature in degree Celsius

| Red (R) | Green (G) | Blue (B) | Temperature (temp) °C |
|---|---|---|---|
| >=250 | <=250 | <=250 | 34 |
| >=250 | >=200 | <=200 | 35 |
| >=250 | <=200 | <=150 | 36 |
| >=250 | >=150 | <=100 | 37 |
| >=250 | <=150 | 0 | 38 |

Fig 6. Calculating body temperature from RGB values of thermal image

## 5. Future Scope

- To set up a database to hold information of the people entering any public place premise.
- To codify the entire workflow into a single application.

## 6. Conclusion

The primary objective of our proposed system is to ease the screening processes at entrances of public places. Security personnel are no longer required to hold an infrared temperature gun when measuring people's temperatures as a result of using a thermal camera. By capturing images with a thermal camera, temperatures can be measured. Long queues at public places can be avoided wherein people wait for their temperature to be measured, thus implementing social distancing effectively. Human interaction is thus minimized and the possibility of spreading the

virus decreases. This application also reduces the probability of the security personnel catching the virus, who could in turn act as carriers and infect hundreds of people during the screening process. We thus infer from the implementation that this could be a useful application if implemented at scale. The usage of face detection combined with temperature detecting through thermal images aided us in developing an application that can drastically reduce the spread of the COVID-19 virus. Thermal cameras are becoming more prevalent in companies using facial recognition tools and personal directories. The built-in alarm is triggered when a person is holding a hot beverage, such as coffee, because thermal cameras cannot tell the difference between a human body and other warm objects. Hence, in a case such as this, the security guard on duty must ensure that everything is in order before the person can get his drink back. Due to the camera's inability to detect unwanted objects for temperature measurement, people have to scan their temperature twice. Our model addresses this issue as well.

**References**


1. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", Cornell University, March 2019.
2. T Subhamastan Rao1, S Anjali Devi2, P Dileep3, M Sitha Ram4, "A Novel Approach To Detect Face Mask To Control Covid Using Deep Learning", European Journal of Molecular \& Clinical Medicine, ISSN.
3. D.N. C., G. A., M. R. (2012) Face Detection Using a Boosted Cascade of Features Using OpenCV. In: Venugopal K.R., Patnaik L.M. (eds) Wireless Networks and Computational Intelligence. ICIP 2012. Communications in Computer and Information Science, vol 292. Springer, Berlin, Heidelberg.
4. Vardan Agarwal, "Face Detection Models: Which to Use and Why" July, 2020. [Online]. Available:https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c
5. Manish Chablani,"YOLO, You only look once, real time object detection explained" Aug, 2017. [Online]. Available:https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006
6. Redmon, Joseph and Farhadi, Ali,"YOLOv3: An Incremental Improvement" 2018. [Online]. Available: https://pjreddie.com/darknet/yolo/
7. The United States Food and Drug Administration,"Thermal Imaging Systems (Infrared Thermographic Systems/Thermal Imaging Cameras)" December, 2021. Available: \url{https://www.fda.gov/medical-devices/general-hospital-devices-and-supplies/thermal-imaging-systems-infrared-thermographic-systems-thermal-imaging-cameras}
8. RKirk J. Havens, Edward J. Sharp, Thermal Imaging Techniques to Survey and Monitor Animals in the Wild, Academic Press, 2016. Available: https://doi.org/10.1016/B978-0-12-803384-5.00003-8
9. Sumit Saha,"A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way" December, 2018. [Online]. Available:https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
10. Vendian.org 2021. “What color is a blackbody? - some pixel rgb values”. [Online]. Available at: http://www.vendian.org/mncharity/dir3/blackbody/ [Accessed 10 June 2021].
11. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", Cornell University, May 2016.
12. Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", Cornell University, Dec 2016.