Crowd Safety Management System

Sreeharish Rajan A R
BE in Electronics and Communication
Velammal College of Engineering and Technology, Madurai
(913117106082)

Senthil Pitchappan V
BE in Electronics and Communication
Velammal College of Engineering and Technology, Madurai
(913117106072)

Servin Joshuva V
BE in Electronics and Communication
Velammal College of Engineering and Technology, Madurai
(913117106073)

Omprakash P
Assistant Professor Department of Electronics and Communication Engineering Velammal College of Engineering and Technology, Madurai, Tamil Nadu India
Affiliated to Anna University, Chennai

Abstract:- Crowd Management is a very important factor especially during a time like this, where the world is collapsing due to a pandemic on rise. Crowds always cause a commotion and difficulty to move through an area, this problem can be solved easily by automation at gateways where the crowd can be moved according to the clearance in the traffic ahead. To make crowd management easy we process image and video in order to count people and manage them. This system also has various uses in similar automated management fields. In our thesis we capture the images and videos of the crowd and we process it to find the emotions and also behavior tagging of individuals. We are also able to find whether the crowd are following social distancing and mask detection mandated and if not, the software provides prompt to the admin. This Crowd Management process would make things very much easy for police men and security guards who often control crowds.

Keywords:- *Image Processing, Video Processing, Flask Web Framework, Webcam, TensorFlow, Keras, Machine Learning, Open CV.*

I. INTRODUCTION

In this bizarre, busy world where we could experience a rapid increase in population, we are forced to face several problems each day due to increasing population. So, in that case Crowd safety management is the need of the day. People now a days fail to follow safety measures and we are at a situation to make them follow the instructions strictly. Due to over population time delay can be experienced by individuals such that this may lead to make a project or plan to get fail. Over crowd can be experienced in places like railway stations, bus depose, ticket counters, shopping malls and so on. Thus, to control the people, crowd management is also the need of the day. Thus, our project provides remedial methods to control issues due to over crowd. Crowd management system, mask detection system, facial expression detection system and social distance detection system are the main theme of our project. Here we can reduce the crowd movement in ticket counters and in entrance gates of big malls. Thus, we can ensure that there isn't more crowd in a specific entrance or exit. In pandemic situations when it is advised to follow certain rules and regulation, we must be aware of the current situation and follow the instructions provided. But many people fail to follow the same. This may lead to spread of disease. Thus, due to unawareness and lethargically attitude of few individuals we all may get into issues. Thus, we are witnessing these issues right now due to the spread of CORONA. Thus, we built a system to carefully monitor people whether people are following certain basic rules and to monitor crowd behavior during pandemic situations. Our system ensures whether people are wearing masks. And if they fail to do so, it will be intimated and those people can be given fine. Our system also helps to maintain social distancing between individuals. So, when individuals get closer to each other than the fixed limit of distance. This will be very much useful to control the situation and the spread of the disease can be stopped. And the final system is facial detection system, in this system facial expression of the crowd is captured and the average rating of the program can be analyzed without asking feedbacks from the audience. This system can be initialized and implemented in auditoriums and meeting halls.

II. CROWD MANAGEMENT

In many places, like auditoriums, conference halls, meetings, etc., people tend to crowd in only one exit and entrance. Which is time consuming and not prescribed in this current Covid situation. Looking into these problems, we developed a solution to manage the crowd in the entrance and exits of such places. Our system is efficient and fast. As our system is built upon the existing CCTV camera setup, taking advantage of that processing power we could efficiently manage crowd.

We have used CSRNet model which uses Convolutional Neutral Networks to map the image to its respective density map. The model does not make use of any fully connected layers and thus the size of the input image is variable. As a result, the model learns from a large amount of varied data and there is no information loss considering the image resolution. There is no need of reshaping/resizing the image while inferencing. The model architecture is such that considering the input image to be (x, y,3), the output is a density map of size (x/8, y/8,1).

The model architecture is divided into two parts, front-end and back-end. The front-end consists of 13 pretrained layers of the VGG16 model (10 Convolution layers and 3 MaxPooling layers). The fully connected layers of the VGG16 are not taken. The back-end comprises of Dilated Convolution layers. The dilation rate at which maximum accuracy was obtained was experimentally found out be 2 as suggested in the CSRNet paper.

Batch Normalisation functionality is also provided in the code. As VGG16 does not have any BN layers, we built a custom VGG16 model and ported pretrained weights of VGG16 to this model.

This whole system will include all the exit cameras in the place. We will calculate the number of people in that specific frame from all the cameras. Find which exit has a smaller number of people and divert the crowd from the crowed exit to the exit where crowd is less.

Fig 1: **-** Input Image

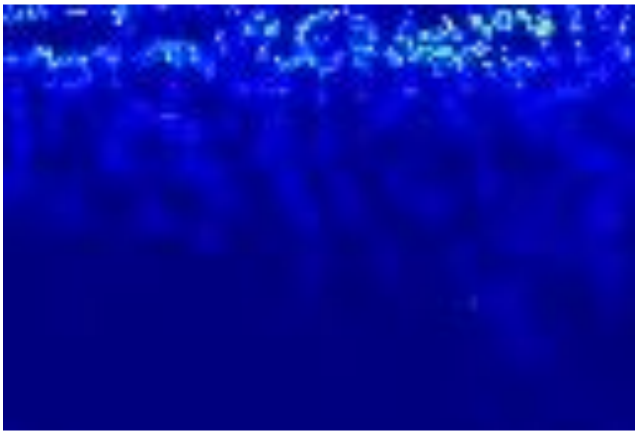


Fig 2: **-** Generated Density Map

Output: Predicted Count: 232

III. MASK DETECTION SYSTEM

Your mask helps protect those around you. Masks are simple barrier to help prevent your respiratory droplets from reaching others. Studies show that masks reduce the spray of droplets when worn over the nose and mouth. You should wear a mask, even if you do not feel sick. Since last year we are in a global pandemic COVID-19. Masks have become a part of our lives even though it's new normal. All theaters, shopping complex, auditoriums, stadiums almost all the places are open and it is mandatory to wear masks in all the places. Looking at the current situation there might me one more lock down. So, we have come up with a solution to detect masks from the people who enter the place where its mandatory to wear masks.

There will be two possibilities (classes / labels):

1. Mask
2. No Mask

If there is Mask, the person is let in by moving the entry rod by 90° and with a green light signal in the system.

If there is No Mask, the person will not be let inside and a red-light alert with a SMS/Email/Call will the sent to the admin along with a buzzer alert.

This system is very helpful for the security team, now they check whether the person is wearing mask, their temperature and sanitize them. With our system in place, human's purpose is eliminated for checking them.

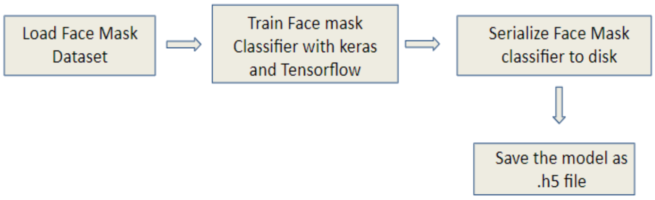


Fig 3: **-** Train Face Mask Detector.

The above block diagram shows how we trained the Face Mask Detection Model. To accomplish this task, we have used MobileNet V2 Architecture, a efficient algorithm that can be applied to devices with less computational power with a RAM of 4GB or 8GB and the devices without GPU. With all the dataset in place, all the images were resized to 224*224 pixels as a part of pre-processing step. Later appending all the pre-processed images and the associated labels to the data and labels lists. From the dataset, 80% of the images were used from training and remaining 20% for testing. Load MobileNet with the lists, leaving the head of the network. We used Adam optimizer and binary cross-entropy. As we have only 2 classes/labels, we have used binary cross-entropy. At last, the serialize our face mask classification model to disk as .h5 model file.

ISSN No:-2456-2165

The below image demonstrates high accuracy and little signs of overfitting on the data. We have obtained ~ 99% accuracy on our test set. Later we trained the same again for more accuracy and efficiency. Moreover, we have also used Google's Teachable Machine, we loaded our dataset and came up with ~99.9% accuracy. The model is also been save in the disk for more accuracy and efficiency.

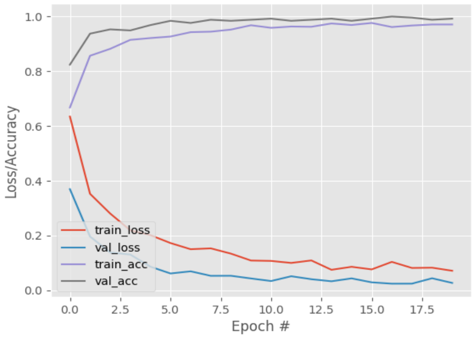


Fig 4: - Training loss and Accuracy Graph

Next step is to implement our face mask detector for images and videos streams. As we have resized all of our image dataset to 224*224 pixels and made our model. We should use make the same configuration size during our detection. So, we took advantage of imutils for its aspect-aware resizing method. We an image and video as an input, we detected the face location and initialized lists for them. These lists include faces, locations, and the prediction. Then we load or trained model, .prototxt file and weights file. Later we used Open CV to draw a rectangle around the face location and put a text over the rectangle the show the predictions.

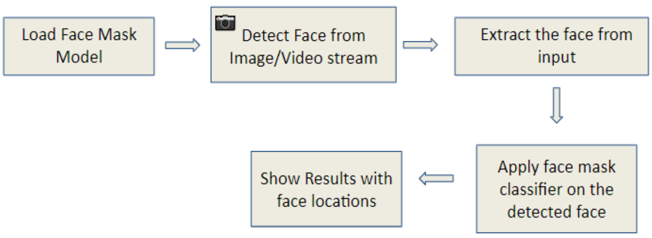


Fig 5: - Detect Face Mask Implementation

IV. RESULTS

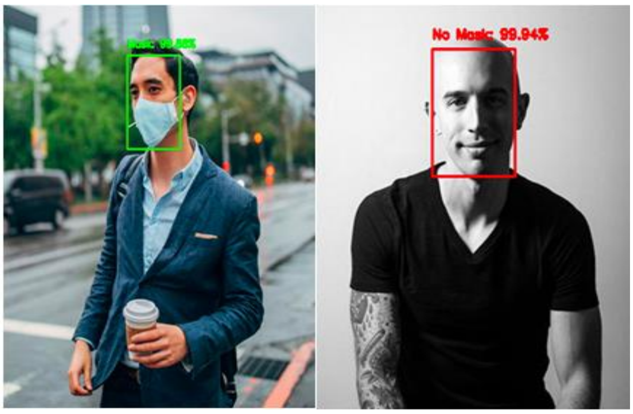


Fig 6: - Face Mask Detection results

V. SOCIAL DISTANCING

As mentioned earlier COVID 19 is nearing and is getting intense in the upcoming days. As masks have become a part of our lives, social distancing is also a very important. It is mandatory to maintain 6 feet (about 2 arm's length) distance between people.

COVID-19 outbreak has created a lot of tension and misery to many families across the globe. During this pandemic, people are advised to not be in close contact with others to reduce the spread of the disease. But there are still many humans who are negligent about this disease by not maintaining social distance. So, I developed this project to monitor if people are maintaining social distance or not.

This can help the Government monitor areas where social distancing is not practiced and enforce stricter laws. Now in most of the place's security or staff of that particular place is in-charge of letting people maintain social distancing. It can either be in a shopping complex, theaters, auditoriums, malls etc. Our solution can reduce the man power to a greater extent.

Our system can effectively find whether people are maintaining social distancing in a particular place. Thus, we used SSD MobileNet (Single-Shot multibox Detection) which needs less processing power compared to TensorFlow or Keras. As TensorFlow is not need in this application.

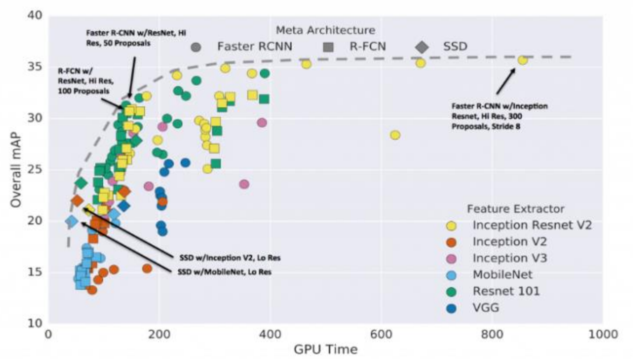


Fig 7: - MobileNet SSD vs others

Single Shot object Detection (SSD) MobileNet, imutils and OpenCV were used to detect people. A bounding box is displayed around every person detected.

Webcam is used to capture the video and detect people in real-time. If people are very close to each other, a red bounding box is displayed around them indicating that they are not maintaining social distance.

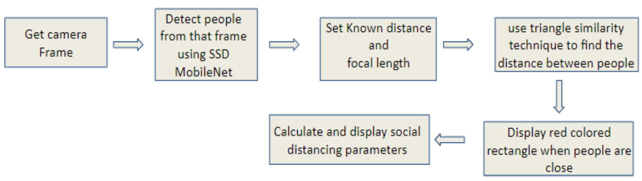


Fig 8: - Social Distancing Block Diagram

To detect the distance of people from camera, triangle similarity technique was used. Let us assume that a person is at a distance D (in centimeters) from camera and the person's actual height is H (I have assumed that the average height of humans in 165 centimeters). Using the object detection code above, we can identify the pixel height P of the person using the bounding box coordinates. Using these values, the focal length of the camera can be calculated using the below formula:

$$F = (P \times D) / H$$

where,

P = width in pixels
D = distance from our camera
H = object with known width
F = focal length

After calculating the focal length of the camera, we can use the actual height H of the person, pixel height P of the person and focal length of camera F to calculate the distance of the person from camera. Distance from camera can be calculated using:

$$D' = (H \times F) / P$$

where,

P = width in pixels
D' = distance from our camera
H = object with known width

Now that we know the depth of the person from camera, we can move on to calculate the distance between two people in a video. There can be n number of people detected in a video. So, the Euclidean distance is calculated between the mid-point of the bounding boxes of all the people detected. By doing this, we have got our x and y values. These pixel values are converted into centimeters using Eq 2.

We have the x, y and z (distance of the person from camera) coordinates for every person in centimeters. The Euclidean distance between every person detected is calculated using the (x, y, z) coordinates. If the distance between two people is less than 2 meters or 200 centimeters, a red bounding box is displayed around them indicating that they are not maintaining social distance. The object's distance from camera was converted to feet for visualization purpose.

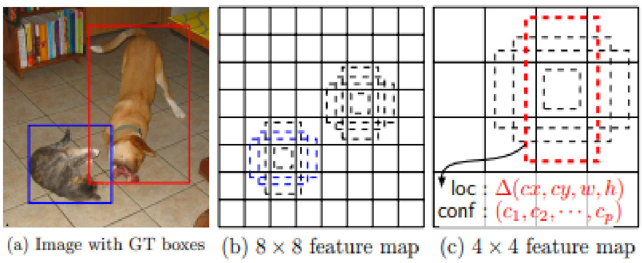


Fig 9: **-** SSD Framework

First, we find people from the input image using SSD MobileNet. To do that we will convert the input frame to grayscale, blur to remove noise and apply edged detection. We find our markers (people) by using cv2.findcounters function with help of SSD MobileNet.

Next, we will use triangle similarity technique, with the help of the above-mentioned formulas. The known distance will be around 200 centimeters (2 meters). And focal length can be changed depending on the detection and the crowd present in that specific area.

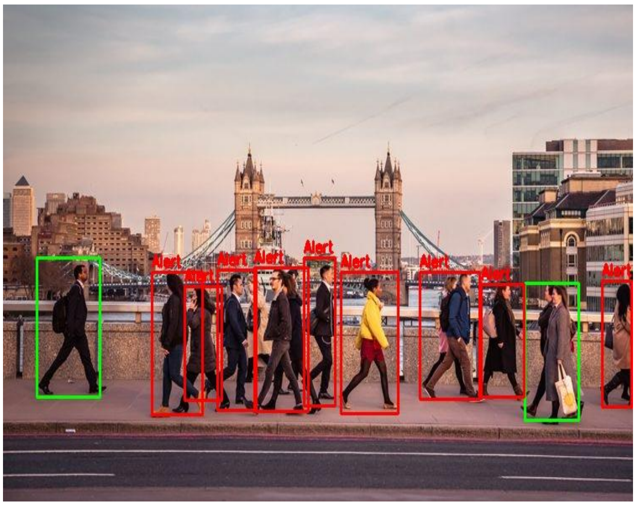


Fig 10: **-** Social Distancing Output

VI. EMOTION DETECTION

It’s no secret that emotions drive behavior. Happy people whistle. Angry drivers crash cars. And now, with the help of emotion detection and analytics, we use it here in our paper to tune into our customers’ feelings and record them for long time use. With the help of this emotion detection technology, we can read the emotion of the people without actually interacting with them face to face.

This Emotion Detection can be used in various places, one of its primary uses include implementing these in International or National Conferences where huge number of people get together to share and network with other people. By using this tech here we don’t basically need to ask people individually about how is their experience and how did they like the conference and which event was way more interesting and which event was funnier and which event was boring. People can lie but their face can’t, so we may be able to find the true original feedback of the conference without any buffer or filter. This detection system is the future of people providing constructive criticism and helps them improving their game.

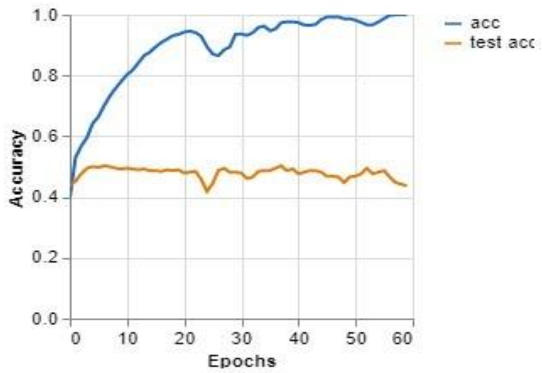


Fig 11: - Training Graph for Emotion Detection

From the testing graph we are able to see the accuracy of our Emotion Detection System and the error level seems to be negligible so, we are able to extract the emotion of the user at a precise level. Even as epochs increase the accuracy doesn't essentially decrease with respect to it. The system is able to provide a constant accuracy without any resilience.

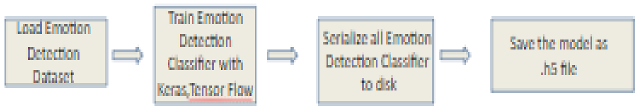


Fig 12: - Training Diagram for Emotion Detection

In the above block diagram, we see that we have trained the system to load Emotion Detection data set in the system. Now by loading we concurrently train Emotion Detection Classifier with Keras and Tensor Flow. Now with the fully trained system we serialize all emotion detection classifier to disk, that is we preload into the system the default emotion types for crossing and checking with real time data. These emotions are made clear and especially all types of human emotions are included explicitly. Finally, these training files are stored in the model as .h5 file for making it accessible by the system.

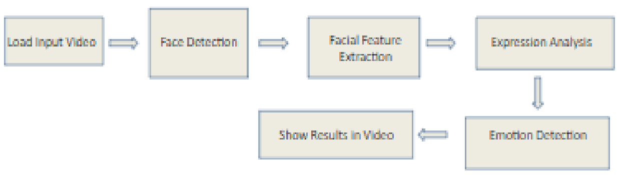


Fig 13: - Implementation Block Diagram for Emotion Detection

Here we use facial recognition tech where we detect the face first then extract the facial feature. Now after the extraction process, we compare the date with, the default data stored in our database where we preload different variations of emotions. Then we find the correct match and detect which emotion is being displayed by the user or person whose video was captured.

We as humans know we have very complex emotions which are difficult to read at certain times. So, we especially use elaborate algorithm in our system to capture the users' emotion and decode it to the data stored in our disk. The following diagram shows the emotions we use in our system as we couldn't show all the emotions by human face in a single diagram, we show a few notable and common ones here.

Fig 14: - Emotion Detection Training Diagram

With this emotion detection software, we have extensive uses where we can even find the review of a movie just by installing CCTV cameras in movie theatres where the movie is screening. By doing so we may able to get an honest feedback from people instead of some immaculate response from people and especially we can avoid biased reviews from fans.

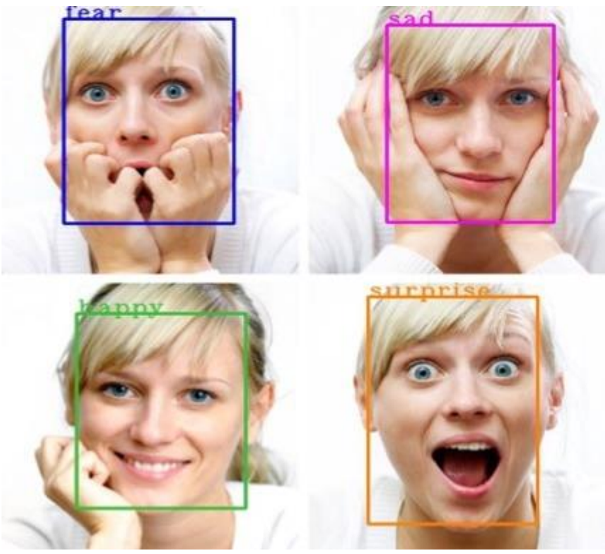


Fig 15: - Output Emotion Detection

This is one of the interesting uses of this emotion detection system, we can implement it in high tension places where there is a probability of terrorism ventures. With enabling this at places such as airports we are able to find the emotions of people and if in any case someone is nervous or showing signs of jitters, the authorities may apprehend and question those people. This allows us to prevent crimes way ahead of time.

It can also be used in Red Corridor Regions in Eastern and Central parts of India to monitor any suspicious activities by the Naxalites. This system for emotion detection can be used in various cases along with the whole complete system or just the Emotion Detection part separately.

VII. PEOPLE COUNT IN AND OUT

Often in many places, the number of people will be in track to analyze the situation, crowd in a particular day and time, and a particular place.

For business like shopping centers, retail chains, museums, restaurants, etc., gathering and analyzing customer traffic data can create value in terms of revenue and business efficiency and generate crucial insights that drive sales, improve conversion rates, and make informed marketing and spending decisions.

Object Tracing is different from Object detection. Object Detection is detecting a specific object from an image and video typically more computationally expensive, and therefore slower, than object tracking algorithm.

People count combines both Object tracking and object detection. We have used centroid tracing algorithm, we used Open CV and dlib. Open CV for vision/image processing along with deep learning for object detection.

We detect the object and set bounding boxes and compute their centroids. From the blow image we can see we have two objects to track at initial stage.

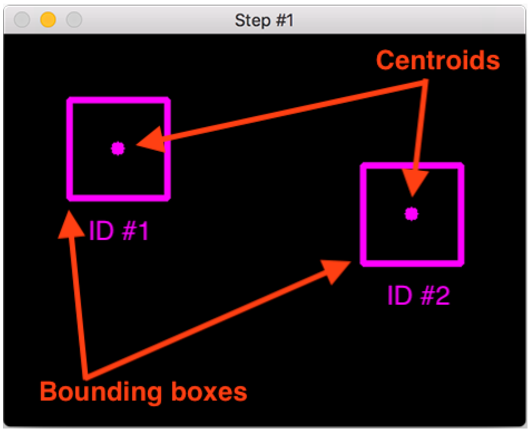


Fig 16: - Bounding box coordinates

Later, the model file, caffe deploy files given in the code along with the input video. We have set a threshold line, where if an object goes up the line it is counted as IN and if the object goes down the line it is counted as OUT. Separate variables are kept in track of the numbers. Checking if the direction is negative, indicating the object is moving up, and the centroid is above the centerline. In this case we increment the total up.

Or checking if the direction is positive, indicating the object is moving down is below the centerline. In this case we increment the total down.

Finally, the total count will be entered in a CSV file along with the current date and time of the tracking. Later, using the data from CSV file, we generate a graph explaining the count at a specific time period.

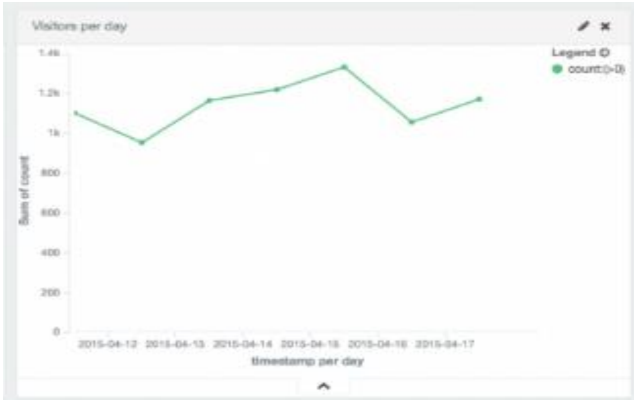


Fig 17: - Final Graph

VIII. API CALLS

Our software is an Open-Source Application, so anyone can use our software from anywhere. Thus, we have developed API calls for some of the features in our application.

1. Mask Detection using Image
2. Emotion Detection using Image
3. Sentiment Analysis using Text

API Endpoint: https://localhost:5000/

Mask Detection:

```
import requests
from pprint import pprint
url = localhost:5000/mask/detect/image/api/v1/KEY
img = {‘image’:open(‘IMAGE_PATH’, ‘rb’)}
r = requests.post(url, files=img)
pprint(r.json())
```

Result:

```
{
'Prediction': ['No-Mask'],
    'data': {'Labels': [['Mask', 0.0], ['No-Mask', 1.0]],
    'Prediction': ['No-Mask']},
    'msg': 'success'
}
```

Emotion Detection:

```
import requests
from pprint import pprint
url = localhost:5000/emotion/detect/image/api/v1/KEY
img = {‘image’:open(‘IMAGE_PATH’, ‘rb’)}
r = requests.post(url, files=img)
pprint(r.json()
```

Result:

```
{
'data': [{'box': [-15, -23, 255, 255],
          'emotions': {'angry': 0.0,
                       'disgust': 0.0,
                       'fear': 0.0,
                       'happy': 0.99,
                       'neutral': 0.0,
                       'sad': 0.0,
                       'surprise': 0.01}}],
          'emotion': 'happy',
          'msg': 'success',
          'score': 0.99
}
```

Emotion Detection - Text:

```
import requests
from pprint import pprint
url = localhost:5000/text/sentiment/api/v1/KEY
data = YOUR_TEXT
r = requests.post(url=url)
pprint(r.json())
```

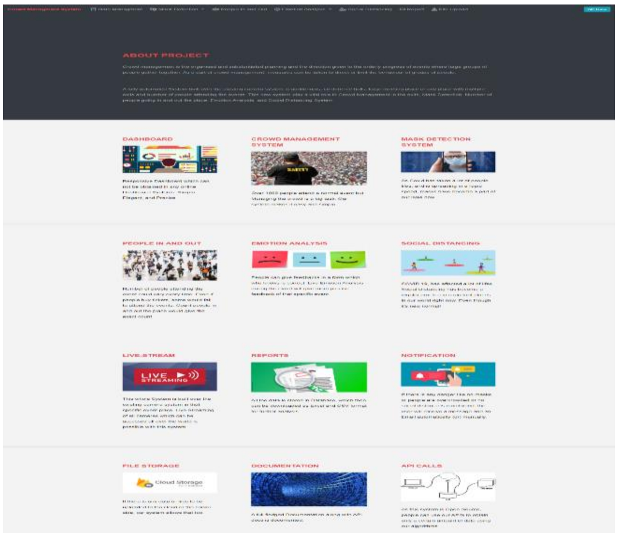
Result:

```
{
    'overall_sentiment': 'Negative',
    'overall_sentiment_score': -0.5423,
   'scores': [{
       'negative': 0.636,
       'neutral': 0.364,
       'positive': 0.0}],
   'sentence': 'you are bad'
}
```

KEY – The API Key given by the admin.

All these above-mentioned codes are POST calls given to the server to obtain the required information. This allows all the users to use these API calls to get information about the image or text they include in the POST call.

IX. SOFTWARE OUTPUT



X. ADVANTAGES

1. Can be implemented into any CCTV camera. Even in mobile camera.
2. Can be used anywhere and everywhere, this system is required.
3. It's not necessary to use all this application together, it also be used separately depending upon the user's needs.
4. API calls one of the biggest advantages of this system.
5. Mask Detection has an accuracy of 98.99%, with only 60 epochs and just 15 batch sets.
6. Emotion Detection has an accuracy of 95% with only 50 epochs with 65 batch sets.
7. Public announcement system.
8. Quick and fast exits.

XI. CONCLUSION

Here Crowd Safety Management System has been discussed and implemented using Machine Learning, Open CV and Flask. This is can implemented in Auditoriums, Conference halls, Colleges, Shopping malls, museums, restaurants, public places, and everywhere and anywhere crowd safety is important. This software will make humans work easier, efficient and secure. As this software has 4 applications, it's not necessary that you have to use all this application. You can use only one application depending on our necessity.

By conclusion we can say that in a digital world this system will work perfectly splendid without any issues. As technology grows the world also grows technologically. A research conducted by a Sweden Lab shows that by the year of 2045 every nook and cranny of the world will be covered by a camera and hooked up to cloud. So, if this is the case we can surely say that our system will play a major role in Crowd Safety Management. We conclude our project by mentioning that it has a great future scope in a modern digital world.

XII. FUTURE SCOPE

In the future, we can implement the same with better technology, which gives more accuracy and more efficient with less computational power. This can then be done with all cloud and give it a potential to access the system all over the world with less latency.

This system has a great future implementation in every field possible. This system can be used either individually or completely based on your need. This seems like one of the excellent features of the system, we can use it wherever there is a camera by just interfacing it with our system.

REFERENCES

[1]. Wei Liu, T.2016. SSD: Single Shot MultiBox Detector. Retrieved from https://arxiv.org/pdf/1512.02325.pdf

[2]. Yuhong Li, Xiaofan Zhang, Deming Chen T.2018. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. Retrieved from https://arxiv.org/abs/1802.10062

[3]. Wafaa Mohib Shalash, Aliaa Al Hazimi T. A Mobile based Crowd Management System. Retrieved from https://www.researchgate.net/publication/318058287_A_Mobile_Based_Crowd_Management_System

[4]. Jie Li, Huib de Ridder T. Designing for Crowd Well-Being: Curent Designs, Strategies and Future Design Suggestions. Retrieved from https://researchgate.net/publication/259308547_Designing_for_Crowd_Well-Being_Current_Designs_Strategies_and_Future_Design_Suggestions

[5]. Huajie Wu, Limin Meng, Jinxue Pan T. Research of people counting based on face detection and area tracking. Retrieved from https://ieeexplore.ieee.org/document/6967168

[6]. H. Hakan Cetinkaya, Muammer Akcay T. People Counting at Campuses. Retrived from https://core.ac.uk/download/pdf/326522302.pdf

[7]. Zhao, X., Delleandrea, E., & Chen,L. (2009). A People Counting System Based on Face Detection and Tracking in a Video. Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 67-72