

Introduction

Overview :

COVID-19 spread is emotionally challenging for many people, changing day-to-day life in unprecedented ways. In the fight against this virus, social distancing has proven to be a very effective measure to slow down the spread of the disease.

This tool has following features:

- Detect humans in the frame with yolov3.
- Calculates the distance between every human who is detected in the frame.
- Shows how many people are at High(red) and Low(green) risks

Purpose:

While millions of people are staying at home to help flatten the curve, many customers in the manufacturing and pharmaceutical industries are still having to go to work everyday to make sure our basic needs are met. To help ensure social distancing protocol in their workplace, I have developed an AI-enabled social distancing detection tool that can detect if people are keeping a safe distance from each other by analyzing real time video streams from the camera.

Requirements:

1. Numpy
2. Pandas
3. Opencv-python (CV2)
4. Imutils:
5. Argparse
6. Dlib
7. Tensorflow

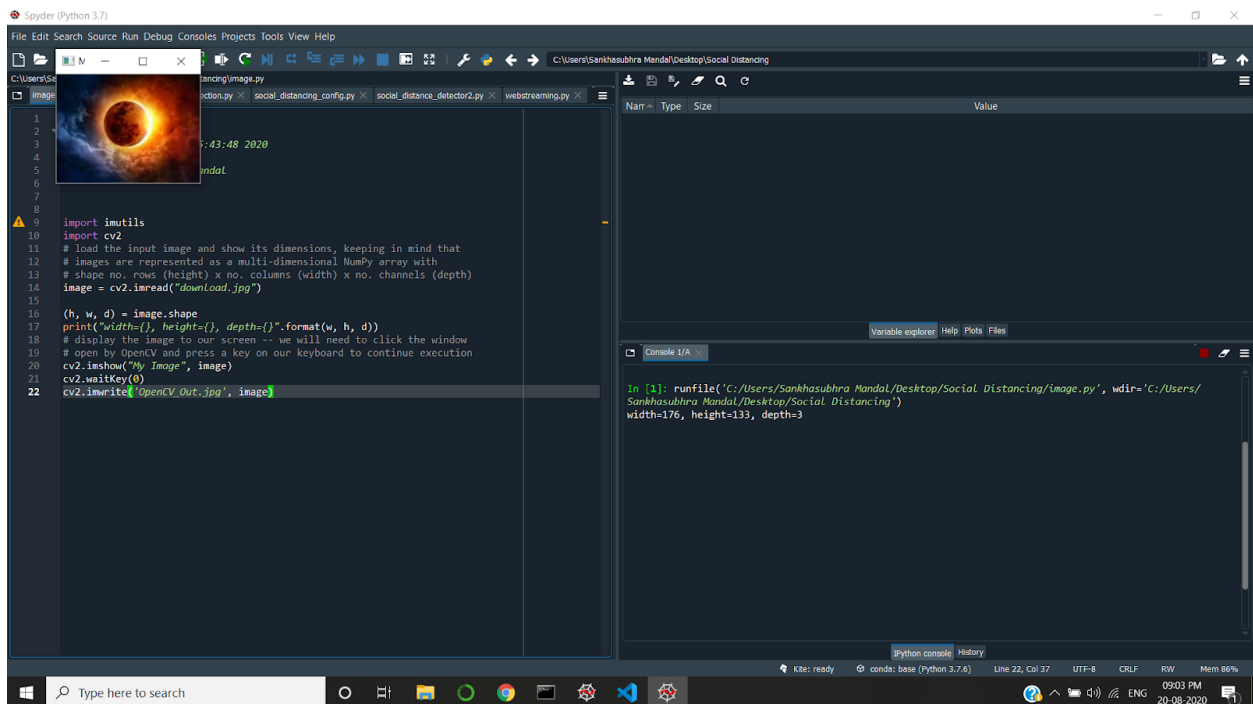
8. Tensorflow GPU
9. Keras

Results

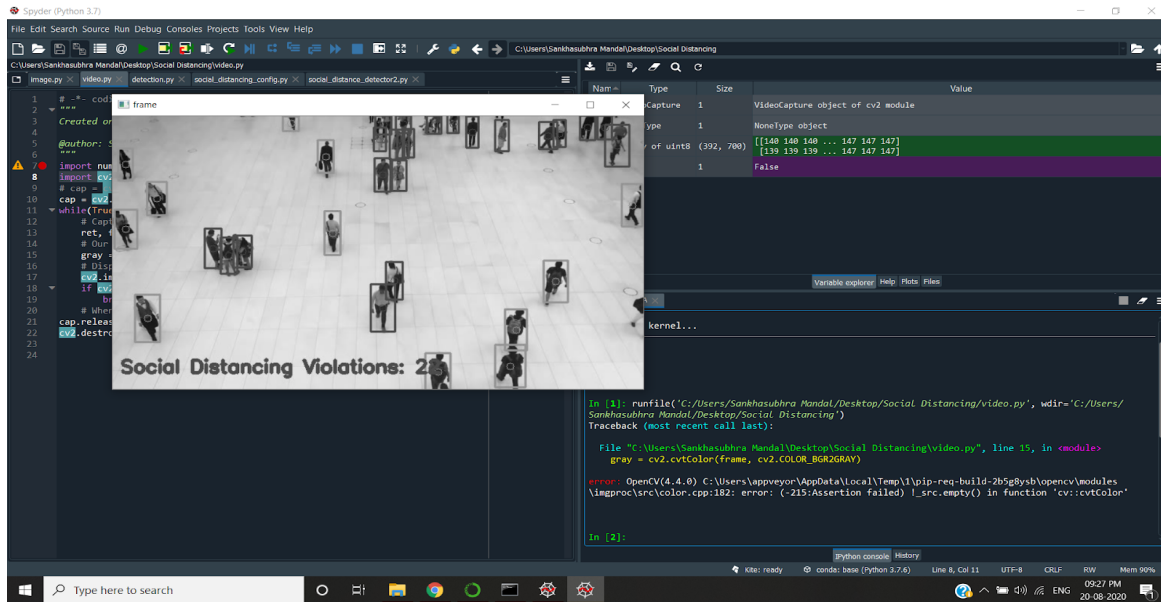
Output:

width=176,
height=133,
depth=3

Running the image.py file to get the shape of download.jpg



Running the video.py file getting this.



Running social_distance_detector2.py(taking inputs passes to arguments over detection.py and the path file also there social_distancing_config.py

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Sankhasubhra Mandal\Desktop\Social Distancing\Yolo\social_distance_detector2.py

```

88 (cX, cY) = centroid
89 color = (0, 255, 0)
90
91 # if the index pair exists within the violation set, then
92 # update the color
93 if i in violate:
94     color = (0, 0, 255)
95 # draw (1) a bounding box around the person and (2) the
96 # centroid coordinates of the person,
97 cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
98 cv2.circle(frame, (cX, cY), 5, color, 1)
99 # draw the total number of social distancing violations on the
100 # output frame
101 text = "Social Distancing Violations: {}".format(len(violate))
102 cv2.putText(frame, text, (10, frame.shape[0] - 25),
103            cv2.FONT_HERSHEY_SIMPLEX, 0.85, (0, 0, 255), 3)
104
105 # check to see if the output frame should be displayed to our
106 # screen
107 # show the output frame
108 cv2.imshow("Frame", frame)
109 key = cv2.waitKey(1) & 0xFF
110 # if the 'q' key was pressed, break from the loop
111 if key == ord("q"):
112     break
113 # if an output file path has been supplied and the
114 # writer has not been initialized, do so now
115 if r"social-distance-detector" != "" and writer is None:
116     # initialize our writer
117     fourcc = cv2.VideoWriter_fourcc(*"MJPG")
118     writer = cv2.VideoWriter(r"output.avi", fourcc, 25,
119                           (frame.shape[1], frame.shape[0]), True)
120 # if the writer is not None, write the frame to the output
121 # file
122 if writer is not None:
123     writer.write(frame)
124 cv2.destroyAllWindows()

```

Frame

Social Distancing Violations: 31

Restarting kernel...

In [1]: runfile('C:/Users/Sankhasubhra Mandal/Desktop/Social Distancing/Yolo/social_distance_detector2.py', wdir='C:/Users/Sankhasubhra Mandal/Desktop/Social Distancing/Yolo')
[INFO] loading YOLO from disk...
[INFO] accessing video stream...

Python console History

Kite: ready conda: base (Python 3.7.6) Line 121, Col 18 UTF-8 CRLF RW Mem 93%

09:35 PM
20-08-2020

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Sankhasubhra Mandal\Desktop\Social Distancing\Yolo\social_distance_detector2.py

```

88 (cX, cY) = centroid
89 color = (0, 255, 0)
90
91 # if the index pair exists within the violation set, then
92 # update the color
93 if i in violate:
94     color = (0, 0, 255)
95 # draw (1) a bounding box around the person and (2) the
96 # centroid coordinates of the person,
97 cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
98 cv2.circle(frame, (cX, cY), 5, color, 1)
99 # draw the total number of social distancing violations on the
100 # output frame
101 text = "Social Distancing Violations: {}".format(len(violate))
102 cv2.putText(frame, text, (10, frame.shape[0] - 25),
103            cv2.FONT_HERSHEY_SIMPLEX, 0.85, (0, 0, 255), 3)
104
105 # check to see if the output frame should be displayed to our
106 # screen
107 # show the output frame
108 cv2.imshow("Frame", frame)
109 key = cv2.waitKey(1) & 0xFF
110 # if the 'q' key was pressed, break from the loop
111 if key == ord("q"):
112     break
113 # if an output file path has been supplied and the
114 # writer has not been initialized, do so now
115 if r"social-distance-detector" != "" and writer is None:
116     # initialize our writer
117     fourcc = cv2.VideoWriter_fourcc(*"MJPG")
118     writer = cv2.VideoWriter(r"output.avi", fourcc, 25,
119                           (frame.shape[1], frame.shape[0]), True)
120 # if the writer is not None, write the frame to the output
121 # file
122 if writer is not None:
123     writer.write(frame)
124 cv2.destroyAllWindows()

```

Name	Type	Size	Value
endY	int	1	66
fourcc	int	1	1196444237
frame	Array of uint8	(393, 700, 3)	[[[123 126 124] [123 126 124]
grabbed	bool	1	True
i	int	1	38
j	int	1	38
key	int	1	113
labelsPath	str	1	yolo-coco\coco.names
ln	list	3	['yolo_82', 'yolo_94', 'yolo_106']
net	dnn_Net	1	dnn_Net object of cv2 module
prob	float	1	0.34840887784957886
results	list	39	[(0.9956908822059631, (...), (...)), (0.995117783546478, (...), (...))
startX	int	1	0
startY	int	1	36
text	str	1	Social Distancing Violations: 28
violate	set	28	{1, 3, 4, 10, 12, 13, 14, 16, 17, 18, ...}
vs	VideoCapture	1	VideoCapture object of cv2 module
weightsPath	str	1	yolo-coco\yolov3.weights
writer	VideoWriter	1	VideoWriter object of cv2 module

Variable explorer Help Plots Files

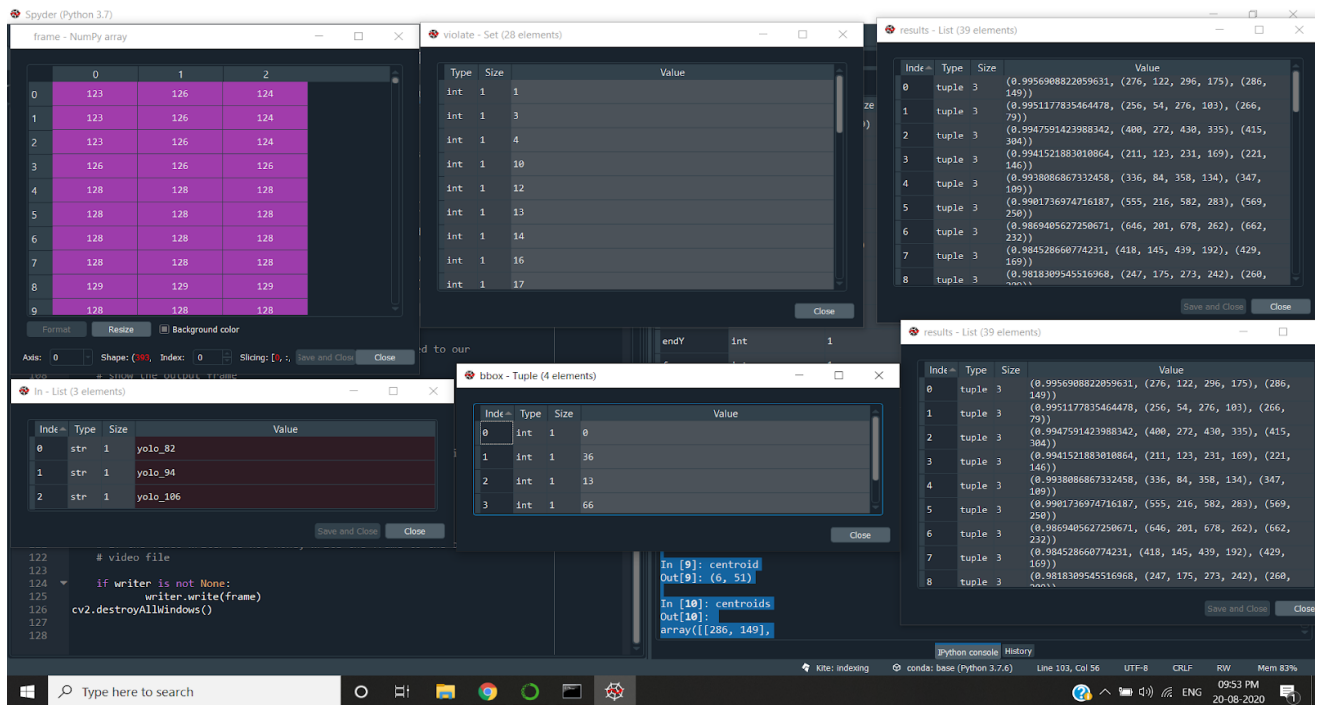
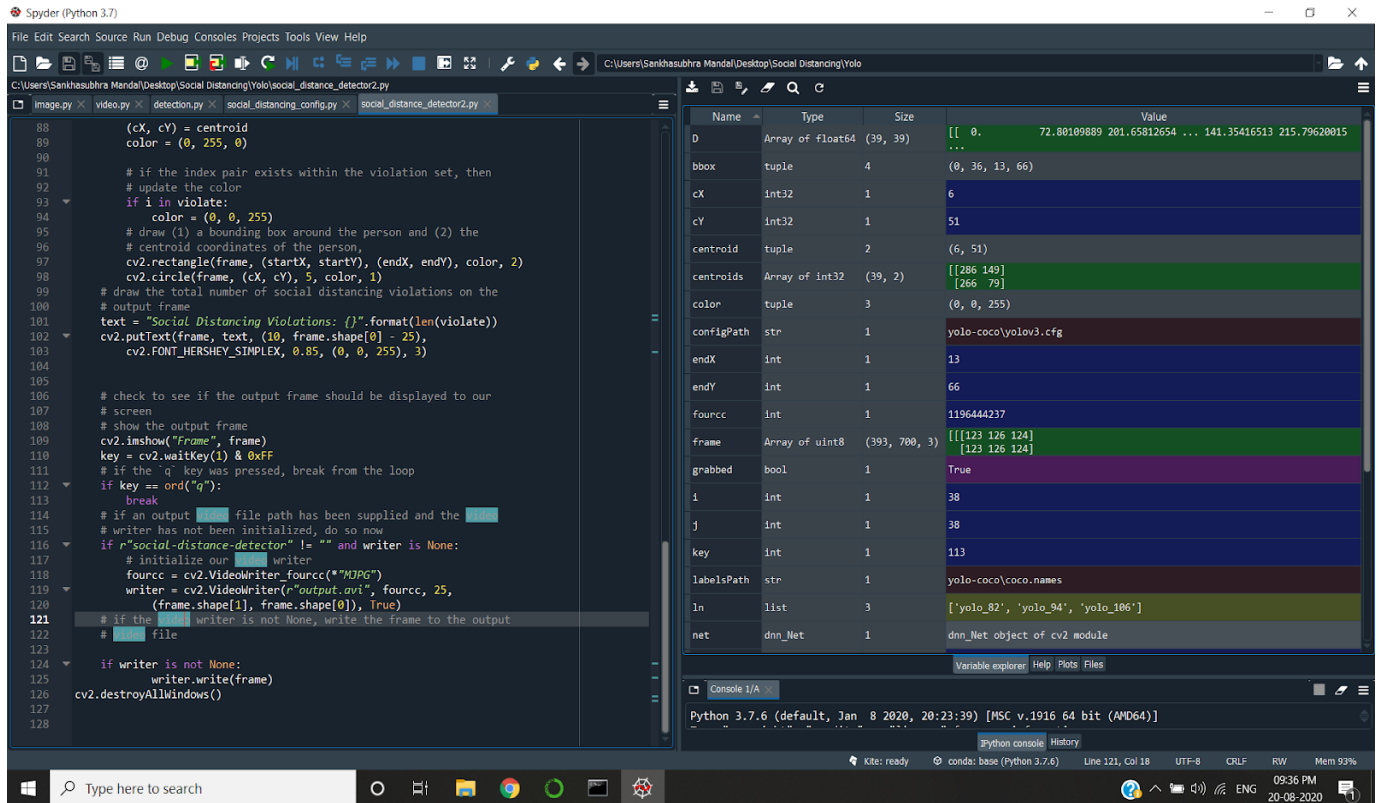
Console 1/A

Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]

Python console History

Kite: ready conda: base (Python 3.7.6) Line 103, Col 36 UTF-8 CRLF RW Mem 94%

09:36 PM
20-08-2020



Output:

bbox: (0, 36, 13, 66)

cX: 6

cY: 51

centroid: (6, 51)

centroids:

```
array([[286, 149],
       [266, 79],
       [415, 304],
       [221, 146],
       [347, 109],
       [569, 250],
       [662, 232],
       [429, 169],
       [260, 209],
       [ 98, 236],
       [408, 51],
       [479, 291],
       [ 39, 65],
       [371, 80],
       [438, 50],
       [564, 187],
       [316, 42],
       [674, 99],
       [119, 18],
       [297, 41],
       [624, 38],
       [157, 13],
       [641, 102],
       [345, 34],
       [167, 204],
       [147, 206],
       [510, 35],
       [598, 28],
       [590, 28],
       [480, 17],
       [667, 6],
       [101, 6],
       [220, 109],
       [ 55, 340],
       [647, 69],
       [693, 63],
       [276, 8],
       [464, 27],
       [ 6, 51]])
```

D:

```

array([[ 0.      , 72.80109889, 201.65812654, ..., 141.35416513,
        215.79620015, 296.65468141],
       [ 72.80109889,  0.      , 269.86292817, ...,  71.70076708,
        204.71443525, 261.50334606],
       [201.65812654, 269.86292817,  0.      , ..., 327.01223219,
        281.30055101, 480.92618976],
       ...,
       [141.35416513,  71.70076708, 327.01223219, ...,  0.      ,
        188.95766722, 273.40263349],
       [215.79620015, 204.71443525, 281.30055101, ..., 188.95766722,
        0.      , 458.62838988],
       [296.65468141, 261.50334606, 480.92618976, ..., 273.40263349,
        458.62838988,  0.      ]])

```

frame:

```

array([[[[123, 126, 124],
         [123, 126, 124],
         [123, 126, 124],
         ...,
         [136, 136, 136],
         [135, 135, 135],
         [134, 134, 134]],

        [[125, 128, 126],
         [125, 128, 126],
         [125, 128, 126],
         ...,
         [139, 139, 139],
         [138, 138, 138],
         [137, 137, 137]],

        [[125, 128, 126],
         [125, 128, 126],
         [125, 128, 126],
         ...,
         [143, 143, 143],
         [142, 142, 142],
         [141, 141, 141]],

        ...,

        [[191, 194, 192],
         [193, 196, 194],
         [198, 201, 199],
         ...,
         [230, 230, 230],
         [230, 230, 230],
         [230, 230, 230]],

        [[189, 192, 190],

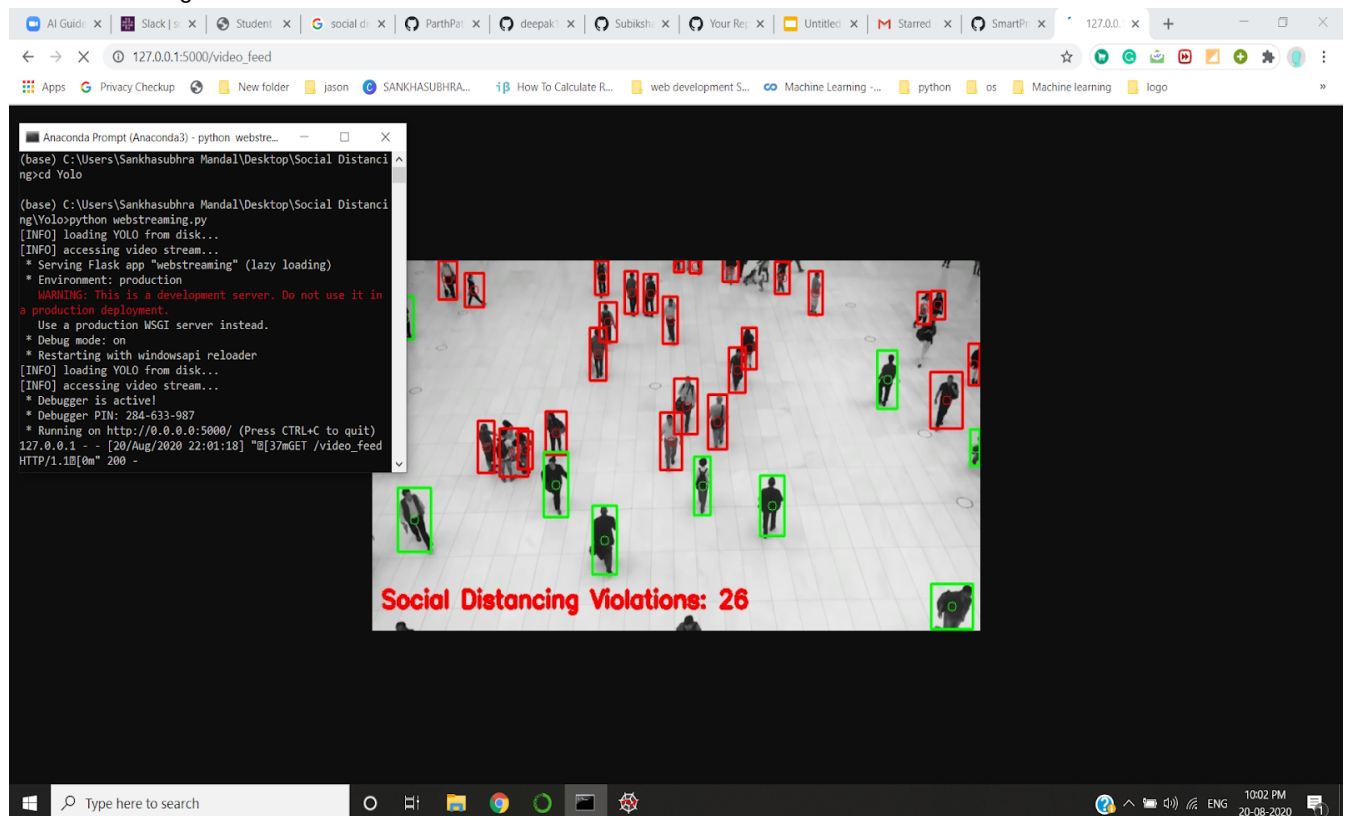
```

```

[196, 199, 197],
[199, 202, 200],
...,
[230, 230, 230],
[230, 230, 230],
[230, 230, 230]],
[[189, 192, 190],
[198, 201, 199],
[199, 202, 200],
...,
[230, 230, 230],
[230, 230, 230],
[230, 230, 230]]], dtype=uint8)

```

Social Distancing Violation : 28



Applications

1. Airports
2. Hospitals
3. Offices

4. Manufacturing plants
5. Retail shops
6. Metro Stations
7. Public libraries
8. Schools
9. Religious Places

Conclusions

This program is a very simple and basic implementation of the Social Distancing Detector. This code can serve as a basis for more complex programs which can be developed from this.

Future Scope

We can extend this by following way:

1.Send Voice Alerts Automatically

Send voice alerts via speakers installed at a specific place when someone violates social distancing. Also, set the rate of detection of faces and sending warnings.

2.Multi-Channel Recognition

Add multiple cameras in a few minutes and allow cameras to access the capability of AI to recognize faces.

3.No new hardware required

You don't need new cameras to enable the platform. Instead, it can work on the existing RTSP camera and connect to your existing smart speakers.