SOCIAL DISTANCING ANALYZER

A PROJECT REPORT

Submitted by

ASHIQUR RAHMAN

ID No. 11708019

TANVEER ISLAM

ID No. 11708035

RAKIBUL HASAN

ID No. 11708051

In partial fulfillment for the award of the degree

Of

BACHELOR OF SCIENCE (ENGG.)

IN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMILLA UNIVERSITY :: CUMILLA-3506

AUGUST,2022

COMILLA UNIVERSITY :: CUMILLA-3506

BONAFIDE CERTIFICATE

Certified that this project report entitled, **“SOCIAL DISTANCING ANALYZER”** is the bonafide work of **“ASHIQUR RAHMAN, TANVEER ISLAM, RAKIBUL HASAN”** who carried out the project work under my supervision.

KHALIL AHAMMAD

CHAIRMAN, EXAM COMMITTEE

Assistant Professor,

Department of Computer Science and Engineering

MD. MOHIBULLAH

SUPERVISOR

Assistant Professor,

Department of Computer Science and Engineering

ABSTRACT

Social distancing has been found to be a useful tool in the fight against the coronavirus in preventing the disease's spread. In order to stop the virus from spreading as quickly as possible, the system is designed to analyze social distance by measuring the space between individuals. To lessen the impact of the epidemic, this technology uses input from camera frames to calculate the distance between people. By analyzing a video stream collected from a security camera, this is accomplished. The video is adjusted for a bird's eye view and sent into the trained object detection model YOLOv3 as input. The Common Object in Context is used to train the YOLOv3 model (COCO). A previously shot video supported the suggested system. The system's results and consequences demonstrate how to assess the space between different people and decide whether or not rules are being broken. Individuals are represented by a red bounding box if the distance is below the minimal threshold value, and a green bounding box otherwise. This technique can be improved to recognize social distance in real-time scenarios.

Keywords: Convolution Neural Network, Computer Vision, Deep learning.

ACKNOWLEDGEMENT

We consider ourselves extremely lucky to have the help of a number of crucial persons during this effort. We want to take this opportunity to thank them all for their assistance during the endeavor.

We'd want to express our heartfelt gratitude to our honorable supervisor, **MD. MOHIBULLAH** for providing us with the excellent chance to work on this fantastic project on **"SOCIAL DISTANCING ANALYZER."** Innumerable ways, his wise counsel, perceptive comments, and patient encouragement benefited the completion of this effort.

We are grateful to him for introducing me to so many new things. Second, we want to express our gratitude to our parents and friends for their assistance in completing this project within the time constraints.

Contents

List of Figures

FIGURES **PAGE NO**

Chapter 1
Introduction

1.1 Background

The widespread coronavirus illness 2019 (COVID-19) has caused a global disaster with its lethal spread to more than 180 nations, and as of March 20, 2022, there were 471,562,771 confirmed cases and 6,102,544 deaths worldwide [1]. The lack of efficacious treatment drugs, as well as a lack of immunity against COVID-19, makes the population more vulnerable. Though vaccines are available, social separation is one of the best effective strategies for combating this epidemic.

The World Health Organization has claimed the spread of coronavirus as a global pandemic because of the increment in the expansion of coronavirus patients detailed over the world [2]. To hamper the pandemic, numerous nations have imposed strict curfews and lockdowns where the public authority authorized that the residents stay safe in their home during this pandemic. Various healthcare organizations needed to clarify that the best method to hinder the spread of the virus is by distancing themselves from others and by reducing close contact. To flatten the curve and to help the healthcare system on this pandemic.

1.2 Motivation

As it is well said, prevention is better than cure, WHO has suggested several safety measures to minimize the transmission of coronavirus. In the present scenario, social distancing [3, 4] has proved to be one of the most exquisite alternative methods as a spread stopper. Social distancing can also be referred to as ‘‘physical distancing,’’ which means maintaining a distance between yourself and the people around you. Social distancing helps to lessen physical contact or interaction between possibly COVID- 19 infected persons and healthy individuals. According to WHO’s standard prescriptions, everyone should keep a distance of at least 6 feet between each other to follow the social distancing. This is a prominent way to break the chain of contagion. Therefore, all the affected countries have adopted social distancing.

Monitoring social distancing in real-time scenarios is a challenging task. It can be possible in two ways: manually and automatically. The manual method requires many physical eyes to watch whether every individual is following social distancing norms strictly. This is an arduous process as one can’t keep their eyes for monitoring continuously. Automated surveillance systems [5, 6] replace many physical eyes with CCTV cameras. CCTV cameras produce video footage, and an automated surveillance system inspects this footage. The system raises alerts when any suspicious event occurs. In view of this alert, security personnel can take relevant actions. There- fore, the automated monitoring system has surpassed several limitations of the manual monitoring method.

1.3 Purpose of the system

This research aims to limit the impact of the coronavirus epidemic with minimal harm to economic artifacts. In this paper, we have proposed an effective automatic surveil- lance system that helps to locate each person and monitors them for the social distancing parameter. This application is suitable for both indoor and outdoor surveillance scenarios. It can be used significantly in various places like railway stations, airports, megastores, malls, streets, etc. The proposed approach can be seen as a combination of two main tasks, mentioned as:

(i) Human detection and tracking

(ii) Monitoring of social distancing among humans

In the first task, this research addresses the problem of human detection and tracking [6, 7, 8] in the surveillance video. Human detection is a two-stage process that involves the localization of an object in the first stage and classification of the localized object in the second stage. This paper has presented a human detection technique based on visual specific learning through deep neural net- works in the video feed. The second task focuses on calculating distance among humans in public areas using our proposed algorithm. The decision is made on social distancing if followed. If not, then the persons who do not follow the social distancing criteria are highlighted with a red rectangle. On seeing this, security personals can take any action related to social distancing rules so that it can be followed strictly.

Chapter 2
Literature Review

2.1 Previous Work

In 2001, a very popular approach for object detection was proposed by Viola and Jones [9]. They used Haar features for features extraction and cascade classifiers with adaboost learning algorithm for classification purposes. This method is 15 times faster than traditional approaches. Fu- Chun Hsu et al. [10] proposed a hybrid approach to detect the head and shoulders by fusing motion and visual characteristics. The authors found that the Histogram of Oriented Optical Flow (HOOF) descriptor is a better choice for segmenting the moving object in video sequences and can handle cluttered and occluded environments efficiently. Vijay and Shashikant [11] proposed a real-time pedestrian detection for advanced driver assistance. This system detects the pedestrian using Edgelet features to improve the accuracy and a classifier based on the k-means clustering algorithm to lessen the system complexity. Suman Kumar Choudhury et al. [12] proposed an advance pedestrian system by incorporating the background subtraction technique to extract moving objects, Silhouette Orientation Histogram, and Golden Ratio Based Partition to extract meaningful information from the moving objects and HIKSVM for object classification. This system can deal with occlusion efficiently and achieved accuracy up to98.36%. Seemanthini and Manjunath [13] deployed the human detection technique for an action recognition system. Singh et al. [14] proposed a human detection frame- work for extensive surveillance in the city through CCTV cameras. They used the background subtraction technique to segment moving objects, HOG descriptor to extract features and SVM for object classification.

Earlier, object detection frameworks implemented the Sliding Window concept [15] for object localization within an image. According to this approach, an image is divided into a particular size of blocks or regions. Further, these blocks are categorized into their respective classes. Various handcrafted feature extraction techniques like HOG [16], SIFT [17], LBP [18], etc. are used to evaluate the attributes or features. Furthermore, these attributes are used to build the classifier to locate the object on the image’s grid. However, this grid-based archetype requires high computational cost and sometimes yields high false-positive rates. Therefore, an effective object classification & localization framework is needed to detect several objects with diverse scales within an image. Additionally, it should reduce the computational cost and false-positive rate. Recently, significant advances have been observed in object detection using deep convolutional neural network (CNN) [19-22].

Convolutional neural networks (CNN) are a class of intensive, feed-forward artificial neural networks that have been used to perform accurately in computer vision tasks, such as image classification and detection. CNN is capable of extracting robust features with the help of the convolution process. Its strong attribute representation capability played a vast role in object detection [23,24]. Aichun, Tian, and Qiao [25] proposed a deep hierarchical model for multiple human upper body detection. This model employs a candidate-region convolutional neural network (CR-CNN) with multiple convolutional features to accommodate the local as well as contextual information from the image and has achieved accuracy up to 86%.

2.2 Related Works

The works that are featured and highlighted in this area relate to object and human detection using deep learning. Deep learning-based work that recently concentrated on classifying items and identifying them is also mentioned. Computer vision is used to detect people, which is regarded as a component of object detection. With the aid of a predetermined model, the detected objects are located and categorized according to their shape [26]. Deep learning and convolutional neural network (CNN) approaches have demonstrated to perform better on visual recognition benchmarks. It is a multiple-layered perceptron neural network with convolutional, sub-sampling, and numerous fully linked layers. It is powerful in detecting different objects from different inputs and it is a supervised feature learning method. Because of the outstanding performance in large datasets such as ImageNet, this model has achieved tremendous success in large-scale image classification tasks [27].

Due to the neural network structure of the object detection and recognition system, which can create things on its own with the aid of descriptors and can learn distinct attributes that are not principally provided in the dataset, the system has had significant success. But in terms of speed and precision, this has its own set of benefits and drawbacks. The real-time object detection algorithms which use the CNN model such as Region-Based Convolutional Neural Networks (R-CNN) [28-30] and You Only Look Once (YOLO) are developed for the detection of multiple classes in various regions. YOLO (You Only Look Once) is a prominent technique as to speed and accuracy in deep CNN based object detection. Figure 1 shows how object detection is done based on the YOLO model.

Transforming the objective and interpretation from the work [31-33], this system which is proposed presents a method for detecting people using computer vision. Instead of using drone technology, the input is a stream of a video sequence from a CCTV camera installed.

The camera's range of view covers the pedestrians passing by in the range of the installed camera. The people in the frame are represented using a bounding box using the deep CNN models. The deep CNN based YOLO algorithm is used to detect the people in the sequence of video streams taken by the CCTV camera. The calculations are done by measuring the centroid distance between the pedestrians, this will represent whether the pedestrians in the video follow sufficient social distance Figure1.

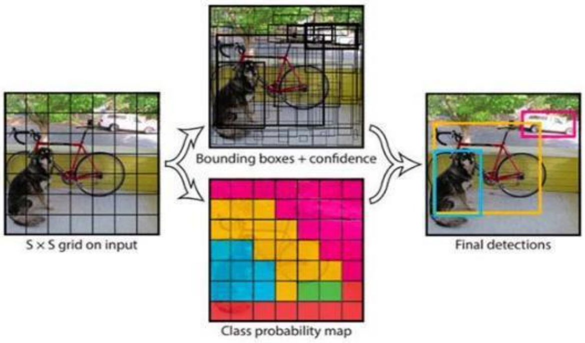


Figure 1. Object Detection using YOLO model.

2.3 Proposed System

To determine the distance between people and preserve safety, the proposed system, the social distancing analyzer tool, was created utilizing computer vision, deep learning, and python. The creation of this work makes use of the YOLOv3 model, which is based on convolution neural networks, computer vision, and deep learning techniques. Initially, an object detection network based on the YOLOv3 algorithm was employed to detect persons in the image or frame [34-36]. From the result obtained, only the "People" class is filtered by ignoring objects of classes. In the frame, the bounding boxes are mapped. The outcome of this operation is used to calculate the distance.

Chapter 3

Methodology and Design

3.1 Approach

The working of the Social Distancing Analyzer is depicted using a flowchart shown in Figure 2.

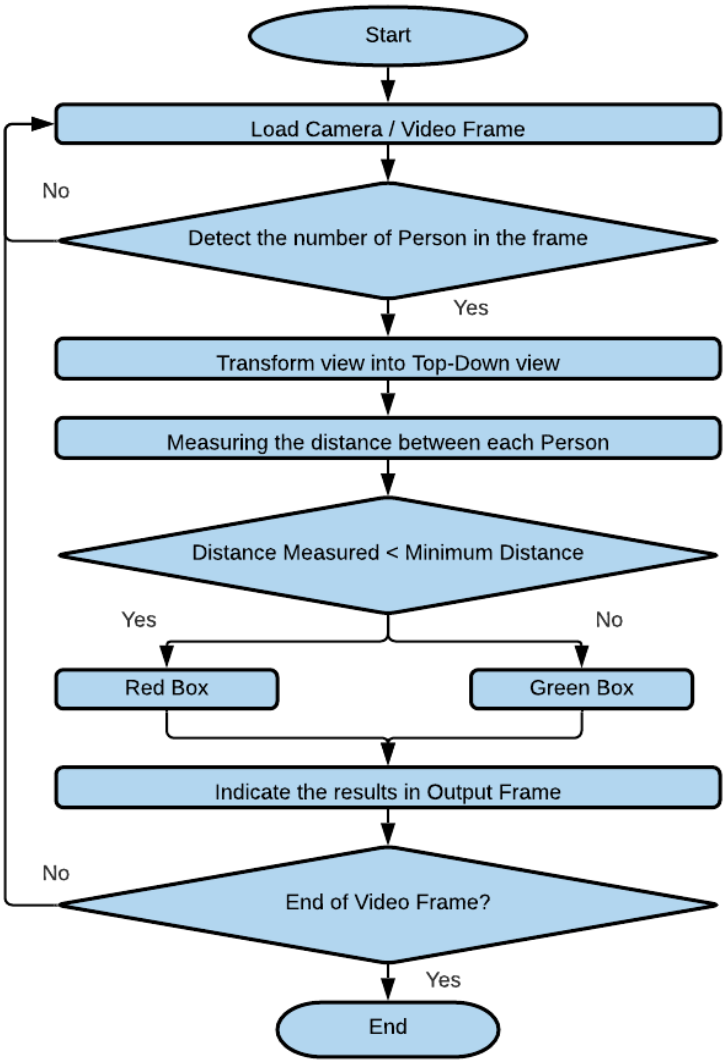

Start
Load Camera / Video Frame
No
Detect the number of Person in the frame
Yes
Transform view into Top-Down view
Measuring the distance between each Person
Distance Measured < Minimum Distance
Yes
No
Red Box
Green Box
Indicate the results in Output Frame
No
End of Video Frame?
Yes
End

Figure 2. The flow chart for the social distancing analyzer model.

3.2 Process Flow Diagram

The process flow pipeline for the Social Distance Analyzer is shown in Figure 3.

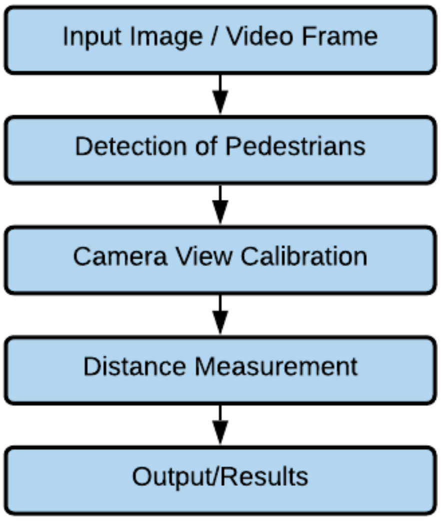


Figure 3. Process flow of the Social Distancing Analyzer model.

3.2 Input Collection

As shown in Figure 4, the CCTV camera's image and video recordings are provided as the input. To precisely estimate the distance between each individual, the camera was set up such that it records at a fixed angle and the video frame's perspective was converted into a 2D bird's view. It is assumed that everyone in the picture is level on the horizontal axis. Then, four locations are selected from the horizontal plane, and the view is altered to a bird's eye perspective. Now that the bird's-eye perspective Figure 3 has been used, it is possible to compute each person's position.

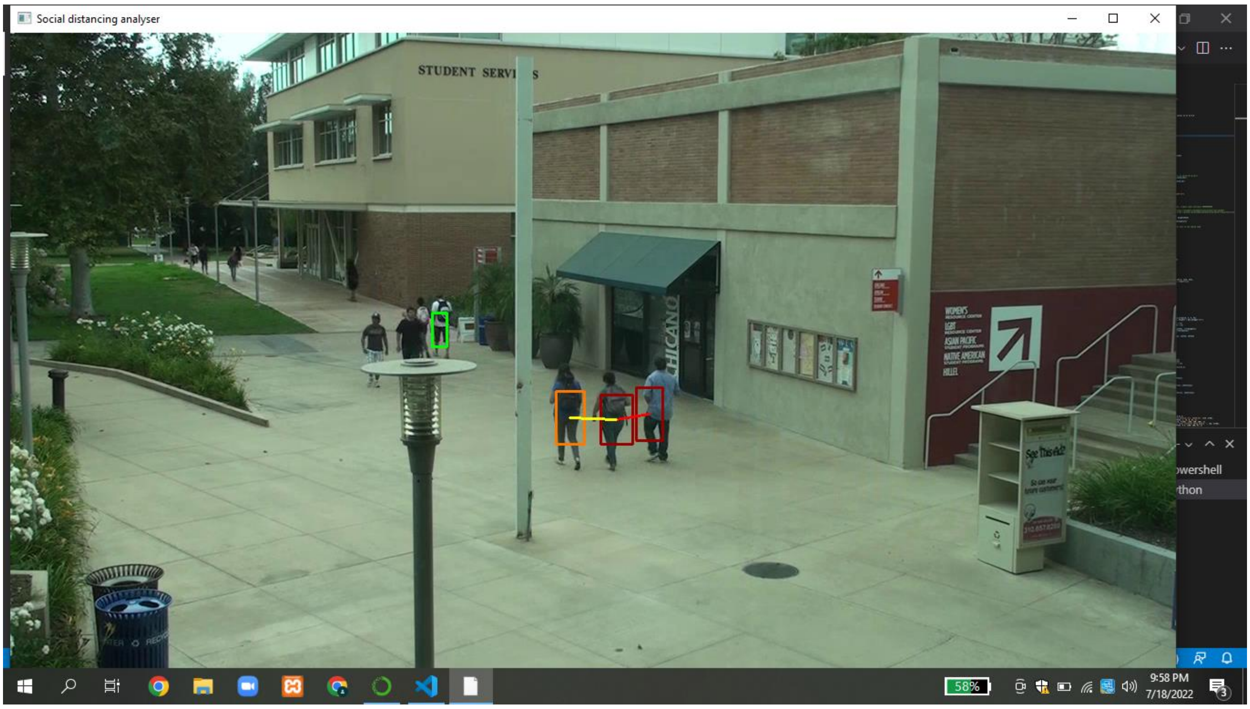


Fig: CCTV footage

Calculating the Euclidean distance between the centroids makes it simple to estimate, scale, and measure the distance between individuals. The distance has a predetermined minimum value or threshold value. Depending on this number, if any distance is discovered to be less than the minimum threshold value, a warning is displayed using bounding boxes that are red in color.

3.3 Calibrating the camera

Using a CCTV camera, the region of interest (ROI) of an image or video frame that was focused on the person walking was collected and then converted into a two-dimensional bird's eye perspective. The size of the modified view is 480 pixels on all sides. By converting the view frame that was collected into a two-dimensional bird's eye perspective, the calibration is carried out. Using OpenCV, the camera calibration is accomplished simply. The calibration function, which chooses four points from the input image/video frame, maps each point to the corners of the rectangle two-dimensional image frame to perform the transformation of vision. Everybody in the image or frame is assumed to be standing on a leveled horizontal plane after applying this modification. Since the total number of pixels between each person in the altered bird's perspective matches to the spacing between each person in the frame, it is now simple to calculate.

3.4 Detection of pedestrians

A quick and effective model for object detection is the Deep Convolutional Neural Networks model. This model solely takes into account regions that only have the "Person" class and ignores regions that are unlikely to have any objects in them. Region Proposals refers to this method of extracting the regions that only contain the items. The size and overlap of the regions indicated by the region proposal are both possibilities. Therefore, depending on the Intersection Over Union (IOU) score, maximal non suppression is utilized in order to ignore the bounding boxes surrounding the overlapping region.

The Social Distancing Analyzer Model's object identification method lessens the problems with computational complexity. It is accomplished by constructing a single regression problem to aid in object detection [5]. Deep learning-based object detection models use the You Only Look model once. This model is appropriate for real-time applications since it is quicker and more precise. Figure 5 displays the YOLOv3 model's pedestrian detection The YOLOv3 is an object detection model that takes an image or a video as an input and can simultaneously learn and draw bounding box coordinates (tx, ty, tw, th), corresponding class label probabilities (P1 to Pc), and object confidence.

On the Common Objects in Context dataset (COCO dataset), the YOLOv3 model has already been trained [4]. There are 80 labels in this dataset, including a pedestrian class for humans. The parameters employed in the Social Distancing Analyzer's pedestrian detection are shown in Figure 5. They are as follows:

- Box Coordinates - tx, ty, tw, th
- Object Confidence - C
- Pedestrians - P1, P2, … Pc

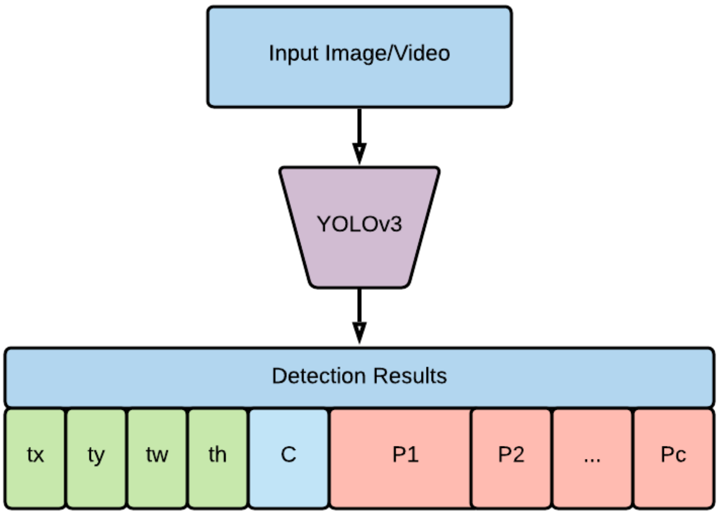


Figure 4. Detection of pedestrians using YOLOv3 model.

There are different objects present in a single frame, the goal is to identify “Only Person” class map bounding boxes related to only the people. The code for drawing the bounding boxes is given below and the output of this code is shown in Figure 6.

```
#To identity “Person Only” class

x = np.where(classes==0)[0]

p=box[x]  count=
len(p)
x1,y1,x2,y2      =
p[0]
print(x1,y1,x2,y2
)
```

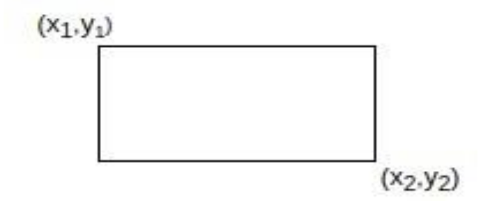


Figure 5. Output obtained from Bounding Box method.

3.5 Measurement of distance

The interval between the set of individuals in an input frame can be easily calculated once the bounding box for each person is mapped. To do so the bottom center of the box mapped to every person within the range is considered. Figure 6 represents the steps followed by the social distancing analyzer model in order to calculate the distance and generate warnings.

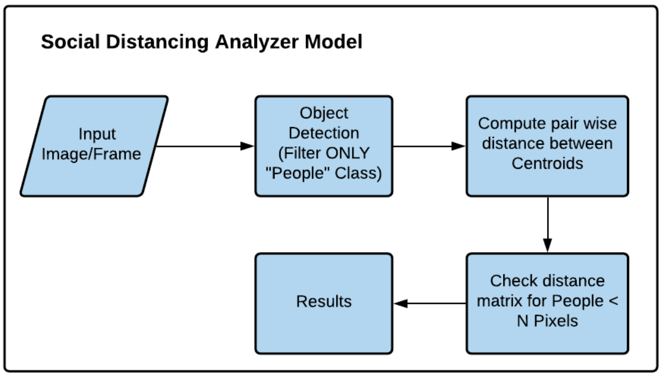

Social Distancing Analyzer Model
Input Image/Frame
Object Detection (Filter ONLY "People" Class)
Compute pair wise distance between Centroids
Check distance matrix for People < N Pixels
Results

Figure 6. The steps involved in the social distancing analyzer model.

The central axis point of each person in the input frame is used to calculate the orientation for each individual for the bird's eye view transformation. By computing the euclidean distance between centroids, it is possible to determine the spacing between each group of people from a bird's-eye perspective. More precise findings can be obtained as the camera is calibrated.

The group of people whose interval falls below the predetermined minimum threshold value is regarded as in violation. A red box is used to indicate those who disobey the rule, and a green box is used to indicate everyone else. The following is the code for calculating the centers of the boxes:

```
#To compute center x_c =
int((x1+x2)/2) y_c = int(y2)

c = (x_c, y_c)

_ = cv2.circle(image, c, 5, (255, 0, 0), -1)


    plt.figure(figsize=(20,10))

    plt.imshow(image)

    def mid(image,p,id):
```

```
 x1,y1,x2,y2 = p[id]

 _ = cv2.rectangle(image, (x1, y1), (x2, y2), (0,0,255), 2)
```

The code to compute the pairwise distances between all detected people in a frame is given below:

```
%%time

from      scipy.spatial      import
distance def dist(midpt,n):

  d = np.zeros((n,n))

  for i in range(n):

    for j in range(i+1,n):

     if i!=j:

        dst = distance.euclidean(midpt[i], midpt[j])

        d[i][j]=ds
  t return d
```

If the result obtained in the previous method is less than the minimum acceptable threshold

value, then the box around the set of people is represented using red color. The code that defines a function to change the color of the closest people to red is given below:

```
def red(image,p,p1,p2):

  unsafe = np.unique(p1+p2)

  for i in unsafe:

    x1,y1,x2,y2 = p[i]

    _ = cv2.rectangle(image, (x1, y1), (x2, y2), (255,0,0), 2)

  return image
```

3.6 Proposed CNN model

Convolutional neural network (CNN) [23,37] has drawn much attention to the research community’s attitude and can be successfully embedded in a broader image classification paradigm. It takes an image as input, assigns significance to different objects within an image based on trainable weights & bias, and effectively differentiate each object.

3.7 Architecture of CNN:

Deep learning is a well-known computer vision approach. To design our model architecture, we used Convolutional Neural Network (CNN) layers. CNNs have been shown to mimic how the human brain functions when interpreting visual data. An input layer, some convolutional layers, some fully connected layers, and an output layer are typical components of a convolutional neural network design. These are layers that have been linearly stacked and are arranged in a certain order. CNN is based on the LeNet architecture with minor modifications. Without taking into input and output, it contains six levels. The following diagram depicts the architecture of the Convolution Neural Network utilized in the research.

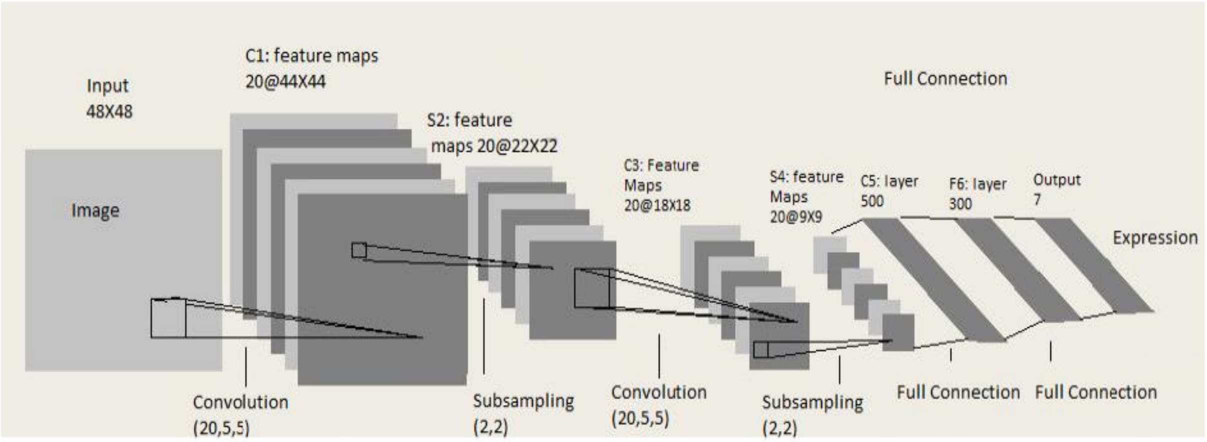


Fig 7. Architecture of CNN

<u>Input Layer</u>

Because the input layer has pre-determined, set dimensions, the picture must be pre-processed before it can be fed into the layer. We utilized OpenCV, a computer vision package, to recognize faces in the image. The haarcascade frontalface default.xml file in OpenCV provides pre-trained filters and utilizes Adaboost to rapidly detect and crop the face. Normalized gray scale photos of size 48 X 48 pixels from the Kaggle dataset are utilized for training, validation, and testing. For testing, laptop webcam photos are also utilized, with the face recognized, cropped, and normalized using the OpenCV Haar Cascade Classifier.

Convolution and Pooling Layers

Convolution and pooling are done in batches. Each batch has N photos, and the CNN filter weights are modified on those batches. Each convolution layer accepts picture batch input with four dimensions: N x Color-Channel x width x height. Four-dimensional feature maps and convolution filters are also available Four-dimensional convolution between image batch and feature mappings is computed in each convolution layer. The only parameters that change after convolution are picture width and height.

New image width = old image width – filter width + 1

New image height = old image height – filter height + 1

Subsampling is used after each convolution layer to reduce dimensionality. This is known as Pooling. Pooling is a dimension reduction method that is often done after one or more convolutional layers. It is a critical phase in the construction of CNNs since adding more convolutional layers can significantly increase processing time. We utilized a common pooling approach called MaxPooling2D, which employs (2, 2) windows over the feature map and only keeps the maximum pixel value.

Max pooling and Average Pooling are two well-known pooling methods. In this project, max pooling is performed after convolution. A pool size of (2x2) is used, which divides the picture into a grid of blocks of size 2x2 and takes a maximum of 4 pixels. Only height and breadth are modified after pooling. The architecture employs two convolution layers and one pooling layer. The size of the first convolution layer in the input picture batch is Nx1x48x48. The picture batch size is N, the number of color channels is 1, and the image height and width are both 48 pixels. The picture batch produced by convolution with a feature map of 1x20x5x5 is Nx20x44x44. Following convolution pooling with a pool size of 2x2, an image batch of size Nx20x22x22 is produced. This is followed by a second convolution layer with a feature map of 20x20x5x5, resulting in a picture batch of Nx20x18x18 pixels. This is followed by a pooling layer with a pool size of 2x2, resulting in a picture batch with the dimensions Nx20x9x9.

Fully Connected Layers

This layer is inspired by how neurons transfer signals throughout the brain. It accepts a large number of input characteristics and transforms them using layers linked by trainable weights. These layers' weights are learned through forward propagation of training data followed by backward propagation of errors. Back propagation begins by calculating the weight adjustment required for each layer before calculating the difference between forecast and real value. By adjusting hyper-parameters like as learning rate and network density, we can regulate the training speed and complexity of the design. As additional data is fed into the network, it may progressively make modifications until mistakes are eliminated. The more layers we add to the network, the better it will be able to detect signals. As appealing as it may sound, the model becomes progressively susceptible to overfitting the training data. Dropout is one technique for avoiding overfitting and generalizing on previously unknown data. During training, Dropout picks a subset of nodes at random and sets their weights to zero. This strategy successfully controls the model's susceptibility to noise during training while keeping the architecture's required complexity. Learning rate, momentum, regularization parameter, and decay are all hyper-parameters for this layer.

The output of the second pooling layer is Nx20x9x9, while the input of the fully-connected layer's first hidden layer is Nx500. As a result, the output of the pooling layer is flattened to Nx1620 dimensions and fed into the first hidden layer. The first hidden layer's output is routed to the second hidden layer. The second hidden layer is Nx300 in size, and its output is supplied to an output layer with the same number of facial expression classes as the number of hidden layers.

Output Layer

The output of the second hidden layer is linked to an output layer with seven unique classifications. At the output layer, we employed softmax instead of the sigmoid activation function. The output is created by utilizing the probabilities for each of the seven classes and the Softmax activation function. This result is shown as a probability for each emotion class. As a result, the model can display the detailed probability composition of the emotions in the face. The projected class is the one with the highest likelihood. As you shall see later, classifying human face expressions as a single emotion is inefficient. Our expressions are typically far more complex, containing a variety of emotions that might be utilized to correctly characterize a certain expression.

3.8 General architecture of YOLOv3

YOLO is a Convolutional Neural Network (CNN) used for real-time object detection. CNNs are classifier-based frameworks that interact with input images as structured arrays of data with the goal of identifying patterns among them. While preserving accuracy, YOLO has the advantage of being much faster than conventional object identification algorithms. Because it enables the model to see the full image at testing, its predictions are influenced by the image's total global context. The regions are "scored" by YOLO and other CNN algorithms based on how closely the images match the predetermined classes.

Other well-known designs like ResNet and FPN served as inspiration for the design of the YOLOv3 feature detector. YOLOv3's feature detector, Darknet-53, contained 52 convolutions with skip connections similar to ResNet and a total of 3 prediction heads similar to FPN, allowing it to process images at various levels of spatial compression.

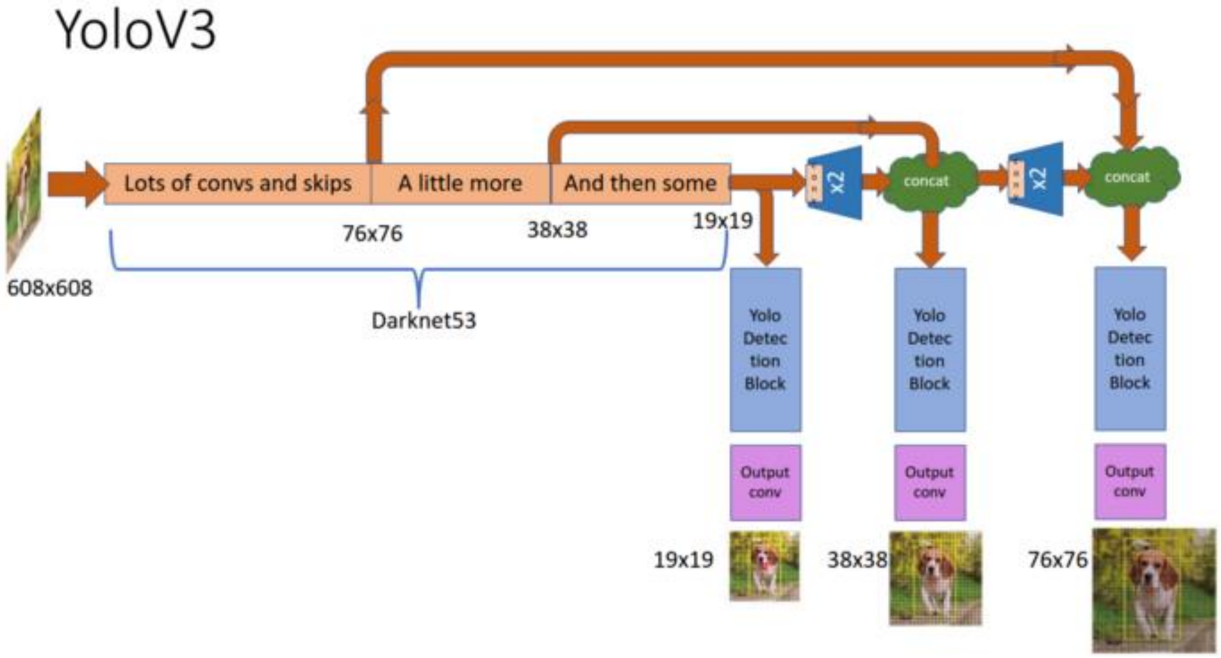


Fig 8: YOLOv3 architecture

Chapter 4
Results and Discussion

4.1 Results of social distance monitoring using pre-trained model

The input is obtained from a pre-shot video of people in a crowded place. The perspective of the recorded video is altered into a two-dimensional bird's view frame by frame in order to precisely calculate the pairwise distances between all detected humans in a frame because the input video is angled. Every individual within the camera's field of vision is identified as the view of the video changes. Circles and points are used to symbolize each person that has been identified in the frame. As demonstrated in Figures 8 and 9, the person whose distance from others is less than the acceptable minimum threshold value is represented by red dots, and the person who maintains a safe distance from others by green points.

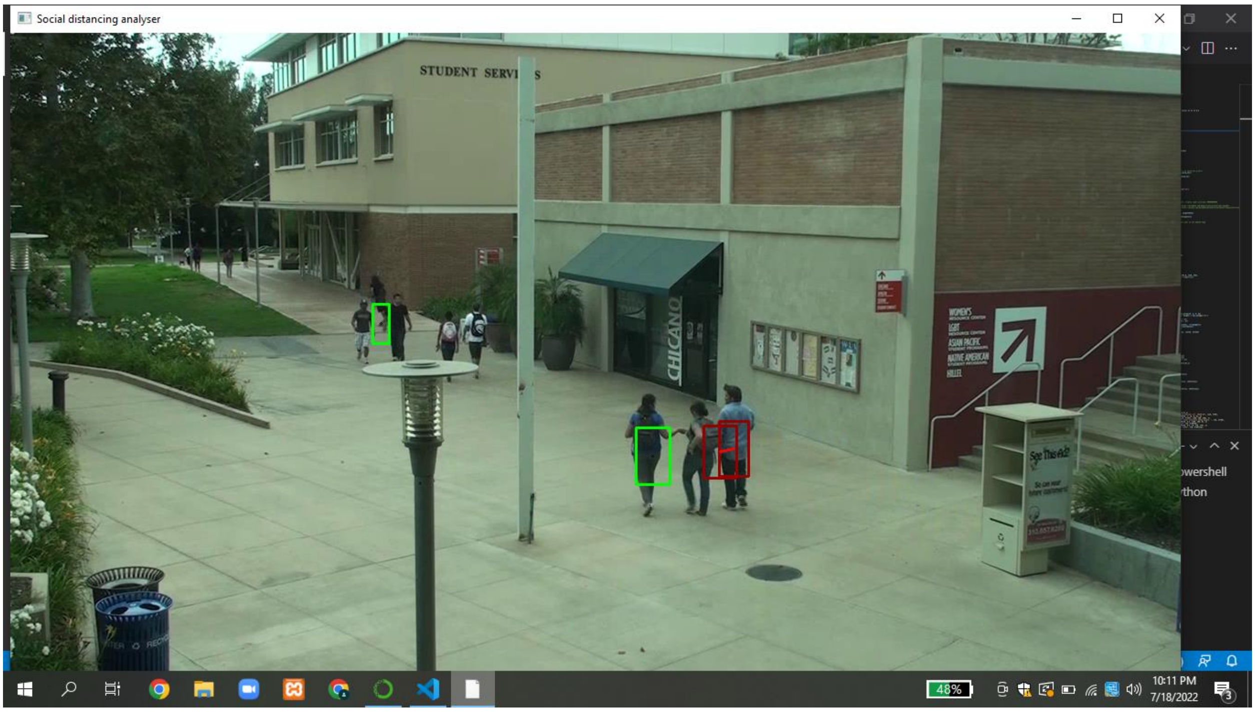


Figure 9. Social distancing analyzer detecting pedestrians in video frame - Unsafe Distance.

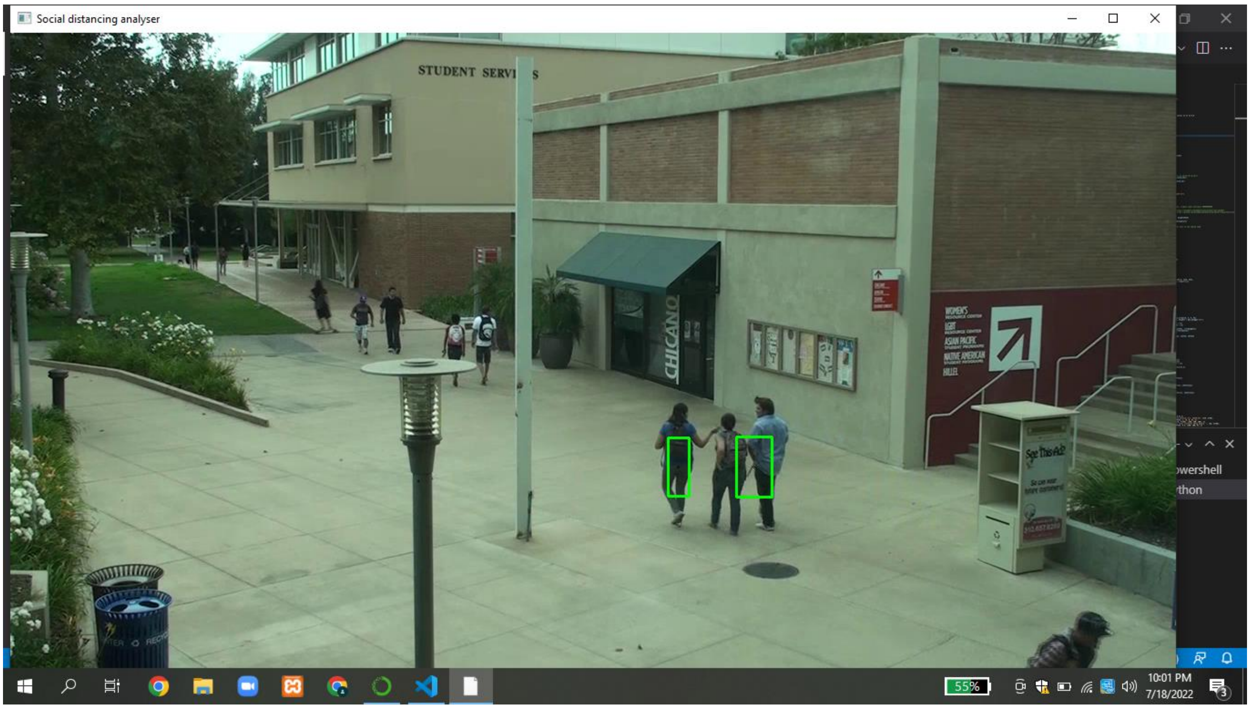


Figure 10. Social distancing analyzer detecting pedestrians in video frame - Safe Distance.

Despite the fact that everyone within the detection range is picked up, there may be some detection errors due to overlapped frames or persons moving too closely together. Figure 10 illustrates this detection issue, when two persons are standing too closely to one another and there are six people inside the detection range yet only four people are spotted.

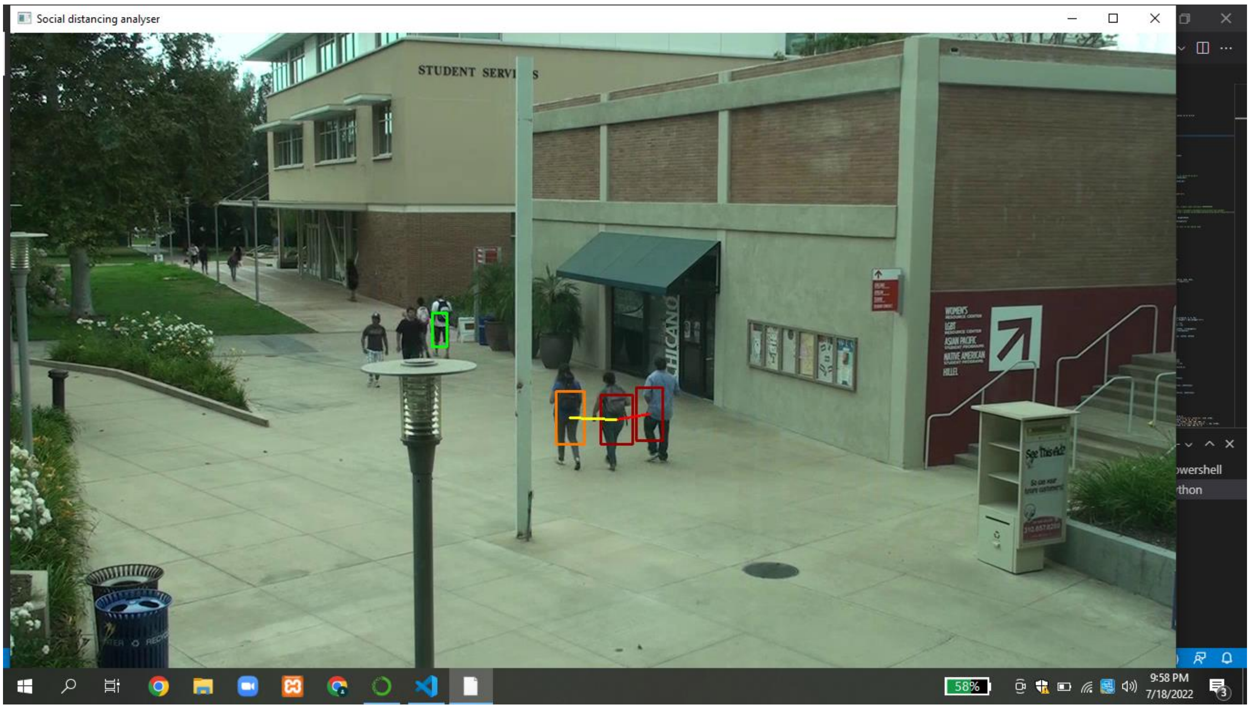


Figure 11. Error in detection people within the range.

The algorithm determines how accurately the distance between each person is estimated. Even if just half of a pedestrian's body is visible, the YOLOv3 algorithm can still identify them as an object because the bounding box will still be mapped to that portion of the body. Comparatively less accuracy can be found in the position of the individual corresponding to the midpoint of the lowermost side of the bounding box. A quadrilateral box is added to symbolize the range in order to eradicate the inaccuracy brought on by the overlapping of frames. Figure 11 depicts the detecting range; only those within this range will be taken into account for calculating distance.

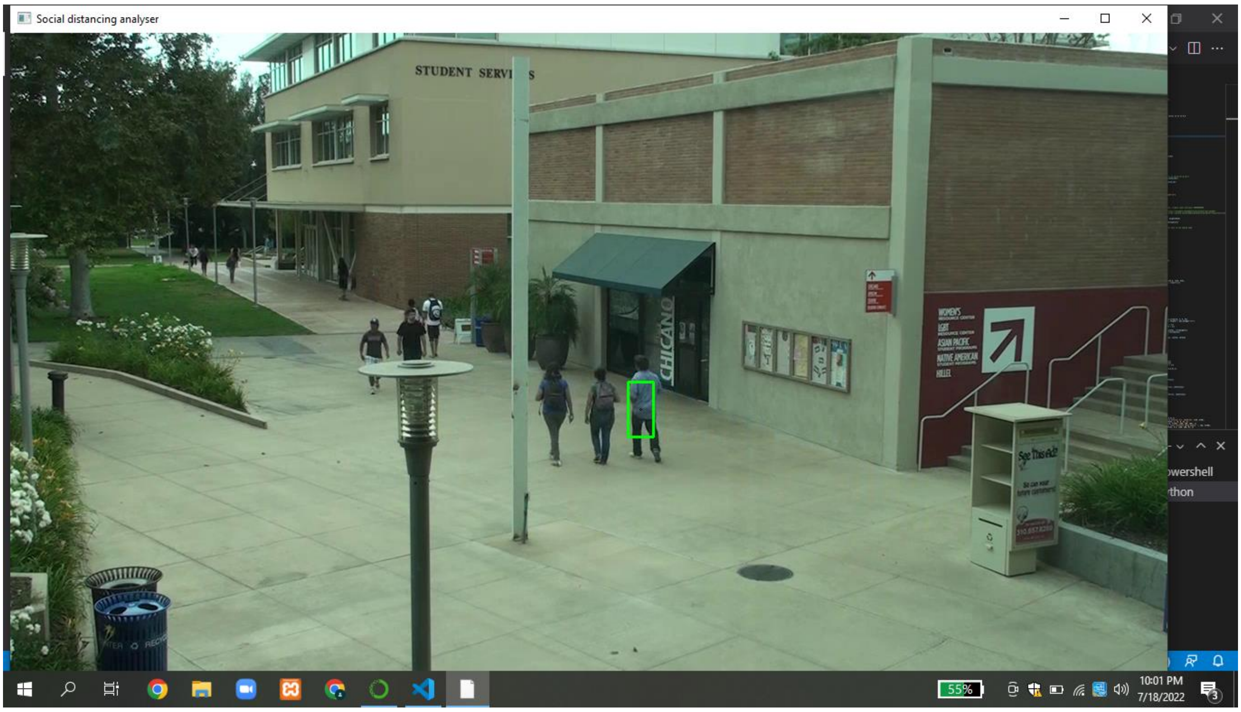


Figure 12. Pedestrians who are out of the specified range are not considered.

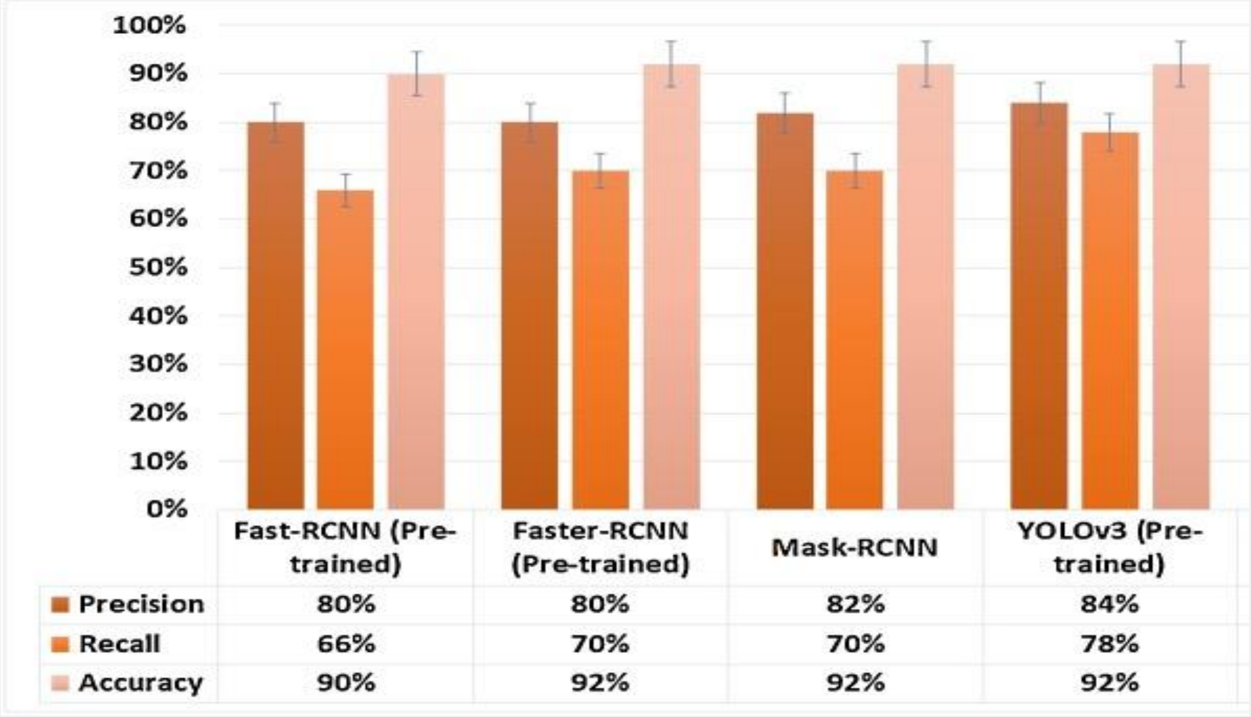
4.2 Performance Evaluation

The newly trained YOLOv3 was compared with other deep learning models. Table 1 shows the True detection and False detection rates for several deep learning models. As can be observed from the outcomes, transfer learning considerably enhanced the outcomes for the overhead view data set. The efficiency of deep learning models is demonstrated by the extremely low false detection rates across several deep learning models, which range from 0.7 to 0.4 percent without any training. On the overhead data set, various pre-trained object detection algorithms are put to the test. Despite being trained on various frontal data sets, the models still perform well, obtaining an accuracy of 90%. The comparative results of various state-of-the-art detection methods are displayed in fig 15.

Table 1

Comparison results of YOLOv3 with other deep learning models.

S. no.	Model	True detection rate	False detection rate
1.	Fast-RCNN (pre-trained)	90%	0.7%
2.	Faster-RCNN (pre-trained)	92%	0.6%
3.	Mask-RCNN (pre-Trained)	92%	0.5%
4.	YOLOv3 (pre-trained)	92%	0. 4%
5	YOLOv3 (trained overhead data set)	95%	0.3%



	Fast-RCNN (Pre-trained)	Faster-RCNN (Pre-trained)	Mask-RCNN	YOLOv3 (Pre-trained)
■ Precision	80%	80%	82%	84%
■ Recall	66%	70%	70%	78%
■ Accuracy	90%	92%	92%	92%

Fig 13: Comparison results of YOLOv3 trained on overhead data set with other methods.

Chapter 5
Conclusion and Future Work

5.1 Conclusion

This work presents an overhead view of a deep learning-based social distance monitoring platform. For human detection, the YOLOv3 pre-trained paradigm is employed. The transfer learning method is used to enhance the pre-trained model's performance because a person's look, visibility, scale, size, shape, and stance differ greatly from an above view. An overhead data set is used to train the model, and the freshly learned layer is then added to the base model. To the best of our knowledge, this study is the first effort to apply transfer learning to a deep learning-based detection paradigm for monitoring social distance from an above perspective. The detection model provides centroid coordinates and bounding box information. The pairwise centroid distances between identified bounding boxes are calculated using the Euclidean distance. A threshold is established and an approximation of the physical distance to the pixel is utilized to check for social distance violations between individuals. To determine whether the distance value breaches the minimal social distance defined, a violation threshold is employed. The people in the scene are also tracked using a centroid tracking method. The framework effectively recognizes individuals who violate social distance and are strolling too closely, according to experimental data. Additionally, the transfer learning methodology improves the detection model's overall effectiveness and accuracy. 95 percent of the model's tracking accuracy. Future iterations of the work could make it more suitable for various indoor and outdoor settings. Different detection and tracking algorithms may be employed to assist in locating the individual or individuals that violate or cross the social distance threshold.

5.2 Future Work

A tool for analyzing social distance is proposed. The system makes use of deep learning and computer vision. The distance between each person can be simply estimated with the aid of computer vision. A red bounding box will be displayed if any group of individuals is discovered to be violating the minimum acceptable threshold value. The created method makes advantage of a previously shot video of people in a busy street. The proposed system has the ability to calculate the separation between individuals. The patterns of social estrangement are the distinction between "Safe" and "Unsafe" distances. Additionally, it shows labels in accordance with the identification and classification of objects. The classifier can be used to create real-time apps and to apply them for live video streams. To monitor people during pandemics, this technology can be combined with CCTV [34]. In congested areas like train stations, bus stops, markets, streets, mall entrances, schools, colleges, workplaces, and restaurants, mass screening is practical and frequently used. We can confirm that a safe distance is maintained between two people by watching the area between them; this can aid in containing the infection.

References

[1] Coronavirus Data; https://www.worldometers.info/coronavirus

[2] WHO (Online). https://www.who.int/dg/speeches/detail/2022

[3] Adlhoch C, et al. (2020) Considerations relating to social dis- tancing measures in response to the COVID-19 epidemic. Euro- pean Centre for Disease Prevention and Control.

[4] Adlhoch C, et al. (2020) Considerations relating to social dis- tancing measures in response to the COVID-19 epidemic. Euro- pean Centre for Disease Prevention and Control.

[5] Singh DK, Kushwaha DS (2016) Tracking Movements of Humans in a Real-Time Surveillance Scene. In: Pant M, Deep K, Bansal J, Nagar A, Das K (eds) Proceedings of Fifth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing. Springer, Singapore

[6] Abughalieh, Karam, Shadi Alawneh (2020) Pedestrian orientation estimation using CNN and depth camera. No. 2020–01– 0700. SAE technical paper.

[7] Karpagavalli P, Ramprasad AV (2017) An adaptive hybrid GMM for multiple human detection in crowd scenario. Multibed Tools Appl 76:14129–14149. https://doi.org/10.1007/s11042-016-3777-4

[8] Bahri H, Chouchene M, Sayadi FE et al (2019) Real-time moving human detection using HOG and Fourier descriptor based on CUDA implementation. J Real-Time Image Proc. https://doi.org/ 10.1007/s11554-019-00935-1.6

[9] Viola P, Jones M (2001) Robust real-time object detection using a boosted cascade of simple features. IJCV 12:18–24

[10] Hsu FC, Gubbi J, Palaniswami M (2013) Human head detection using histograms of oriented optical low in low quality videos with occlusion. In: 2013, 7th International conference on signal processing and communication systems (ICSPCS), IEEE, pp 1–6

[11] Gaikwad V, Lokhande S (2015) Vision based pedestrian detec- tion for advanced driver assistance. Proc Comput Sci 46:321–328

[12] Choudhury SK, Sa PK, Padhy RP, Sharma S, Bakshi S (2018) Improved pedestrian detection using motion segmentation and silhouette orientation. Multimedia Tools Appl 77(11):13075–13114

[13] Seemanthini K, Manjunath S (2018) Human detection and tracking using hog for action recognition. Proc Comput Sci 132:1317–1326

[14] [Singh DK, Paroothi S, Rusia MK, Ansari MA (2020) Human crowd detection for city wide surveillance. Proc Comput Sci 171:350–359. https://doi.org/10.1016/j.procs.2020.04.036

[15] Gajjar, Vandit, Ayesha Gurnani, Yash Khandhediya (2017) Human detection and tracking for video surveillance: A cognitive science approach. In: Proceedings of the IEEE International Conference on Computer Vision Workshops.

[16] Singh DK (2019) Human action recognition in video. In: Luhach A, Singh D, Hsiung PA, Hawari K, Lingras P, Singh P (eds) Advanced informatics for computing research ICAICR 2018 communications in computer and information science. Springer

[17] Najva N, Edet Bijoy K (2016) SIFT and tensor based object detection and classification in videos using deep neural networks. Procedia Comput Sci 93:351–358. https://doi.org/10.1016/j. procs.2016.07.220

[18] R. Jayaswal, J Jha (2017) A hybrid approach for image retrieval using visual descriptors, 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017, pp. 1125–1130, doi: https://doi.org/10.1109/CCAA. 2017.8229965.

[19] Agnes SA, Anitha J, Pandian SIA et al (2020) Classification of mammogram images using multiscale all convolutional neural network (MA-CNN). J Med Syst 44:30. https://doi.org/10.1007/ s10916-019-1494-z

[20] Zhang H, Hong X (2019) Recent progresses on object detection: a brief review. Multimedia Tools Appl 78(19):27809–27847

[21] Zhu A, Wang T, Qiao T (2019) Multiple human upper bodies detection via candidate-region convolutional neural network. Multimedia Tools Appl 78(12):16077–16096

[22] D.T. Nguyen, W. Li, P.O. Ogunbona, Human detection from images and videos: A survey, Pattern Recognition, 51:148-75, 2016.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, In Computer Vision and Pattern Recognition, 2009.

[24] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587. 2014.

[25] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

[26] A. Haldorai and A. Ramu, Security and channel noise management in cognitive radio networks, Computers & Electrical Engineering, vol. 87, p. 106784, Oct. 2020. doi:10.1016/j.compeleceng.2020.106784

[27] A. Haldorai and A. Ramu, Canonical Correlation Analysis Based Hyper Basis Feedforward Neural Network Classification for Urban Sustainability, Neural Processing Letters, Aug. 2020. doi:10.1007/s11063-020-10327-3

[28] Landing AI Creates an AI Tool to Help Customers Monitor Social Distancing in the Workplace [OnLive] (Access on 4 May 2020).

[29] [Ahmed, I., Ahmad, M., Rodrigues, J. J. P. C., Jeon, G., & Din, S. (2020). A deep learning-based social distance monitoring framework for COVID-19. Sustainable Cities and Society, 102571. doi: 10.1016/j.scs.2020.102571

[30] Dhaya, R. CCTV Surveillance for Unprecedented Violence and Traffic Monitoring. Journal of Innovative Image Processing (JIIP) 2, no. 01 (2020): 25-32

[31] Ramadass, Lalitha, Sushanth Arunachalam, and Z. Sagayasree. Applying deep learning algorithms to maintain social distance in public places through drone technology. International Journal of Pervasive Computing and Communications (2020).

[32] Degadwala, Sheshang, et al. Visual Social Distance Alert System Using Computer Vision & Deep Learning. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2020.

[33] Marquez ES, Hare JS, Niranjan M (2018) Deep cascade learning. IEEE Trans Neural Netw Learn Syst 29(11):5475–5485