A PROJECT REPORT ON

**SOCIAL DISTANCE DETECTION USING DEEP LEARNING**

**By**

**ANURAJ A (GK20CCSR04)**

**AVIN KUMAR (GK20CCSR18)**

**ANANYA S (GK20CCSR19)**

*Project Report submitted to Department of Computer Science,Government College Kasaragod in partial fulfillment of the requirements for the award of Degree of BSc Computer Science*



**GOVERNMENT COLLEGE KASARAGOD**

**(Affiliated to KANNUR UNIVERSITY)**

**Vidyanagar, Kasaragod - 671123**

**DECLARATION**

We hereby declare that the project work titled **“SOCIAL DISTANCE DETECTION USING DEEP LEARNING”** was written and submitted by us is our original work. We also declare that this report has not been submitted to any other Universities or Institutions for the award of any fellowship, degree or diploma.

By,

ANURAJ A

Reg no:GK20CCSR04

AVIN KUMAR

Reg no:GK20CCSR18

ANANYA S

Reg no:GK20CCSR19

**GOVERNMENT COLLEGE KASARAGOD**

**(Affiliated to KANNUR UNIVERSITY)**

**Vidyanagar,Kasaragod-671123**



**CERTIFICATE**

This is to certify that the project **“SOCIAL DISTANCE DETECTION USING DEEP LEARNING”** submitted in partial fulfilment of the requirement of the degree of BSc Computer Science is a result of bonafide work carried out by ANURAJ A, AVIN KUMAR, ANANYA S during the academic year 2022-2023.

Internal Project Guide HEAD OF THE DEPARTMENT

External Examiner

1.

2.

**ACKNOWLEDGEMENT**

We are very happy to submit our project **SOCIAL DISTANCE DETECTION USING DEEP LEARNING**, which detects a group of people who are not keeping social distance. We take this opportunity to express our sincere gratitude to the people who have helped us in our attempt.

We wish to express our heartfelt thanks to our principal Dr. A L Ananthapathmanabha, our HOD and our project guide Ms. Akhila P, Assistant Professor, Department of Computer Science. We are extremely thankful to all the staff of the Computer Science department for their kind cooperation and encouragement to make the project successful.

Finally, we would like to thank the team faculty and our lab instructors who extended their helping hand on various occasions.

Sincerely,

ANURAJ A

Reg no:GK20CCSR04

AVIN KUMAR

Reg no:GK20CCSR18

ANANYA S

Reg no:GK20CCSR19

**ABSTRACT**

Social distance means keeping a safe space between yourself and other people to reduce the spread of coronavirus. The virus is spreading quickly and is dangerous to humans. It is necessary to take certain precautions, of which one is social distancing. Maintaining social distancing during COVID-19 is a must to ensure a slowdown in the growth rate of new cases.

The detection tool was developed to alert people to maintain a safe distance  and focuses on detecting if the people around are maintaining social distancing or not. Our goal is to develop a social distance detection system for detecting the frame of a person and displaying labels, they are marked as safe or unsafe if the distance is less than a certain value. System used for monitoring people via video. In our system YOLO-v3 algorithm was employed for pedestrian detection. The proposed method was validated on a prerecorded video of pedestrians walking on the street. The result shows that the proposed method is able to determine the social distancing measures between multiple people in the video.

**INDEX**

1. **INTRODUCTION**……………………………………………………

1.1.PROJECT OVERVIEW………………………………………………………..

1.2.PROJECT OBJECTIVES………………………………………………………

1.3.PROBLEM STATEMENT……………………………………………………

1. **LITERATURE REVIEW………………………………………..**

2.1 REVIEW………………………..………………..………………………………

1. **METHODOLOGY……………………………………………….**

3.1.CNN……………………………………………………………………………….

3.2.OBJECT DETECTION………………………………………………………..

3.3.SYSTEM ARCHITECTURE………………………………………………….

3.4.YOLO……………………………………………………………………………

1. **IMPLEMENTATION……………………………………………..**

4.1. HARDWARE REQUIREMENT…………………………………………

4.2 SOFTWARE REQUIREMENTS………………………………………….

4.2.1.OPEN CV………………………………..…………………………

4.2.2.PYTHON

4.2.3.TENSORFLOW…………………………………………………

4.2.4.KERAS…………………………………………………………….

4.2.5.NUMPY…………………………………………………………….

4.3.DATABASE USED…………………………………………………………

4.3.1.COCO DATASET…………………………………………

4.4.EXPERIMENTAL SETUP……………………………………………………

1. **EXPERIMENTAL AND RESULT ANALYSIS……………..**
2. **SCREEN SHOTS………………………………………………….**
3. **FUTURE SCOPE AND CONCLUSION……………………**
4. **REFERENCE……………………………………………………….**

**INTRODUCTION**

* 1. **OVERVIEW**

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a virus, the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China, in December 2019. The disease spread worldwide, leading to the COVID-19 pandemic.

COVID‑19 transmits when people breathe in air contaminated by droplets and small airborne particles containing the virus. The rin spreads the virus even if they do not develop symptoms.

According to data obtained by the World Health Organization, the global pandemic of COVID -19 severely impacted the world and has now infected more than eight million people worldwide. Following safe social distancing is the enhanced safety protocols that need to be followed in public places.

As doctors suggest, COVID19 spreading can be slowed down by strictly maintaining a safe distance from others .By implementing social distance detection we can avoid the spread of COVID-19.

* 1. **OBJECTIVES**

To shield ourselves from the COVID-19 Pandemic, everyone should keep a safe social distance.. The objective of the project is to develop a model which can be used to detect individuals who are keeping a safe social distance or not. Suggested by WHO (World Health Organization) to shield ourselves and others through contamination we need to remain at least 6 feet away from each other. But it appears to be aggravating to keep impervious distance in this pandemic. People would possibly be keeping distances and some would perchance not. So, to make it easier, we have designed a social distance detection system. It is a real-time object detection computing device that simplifies the work of preserving social distance. This device is designed to aid the users to retain a secure distance from each other. It helps users by exhibiting the number of violations and alert messages. Hence, users will be capable of holding the social distance.

Following safe social distancing is the enhanced safety protocols that need to be followed in public places. By implementing social distance detection we can avoid the spread of COVID-19.

**1.3.PROBLEM STATEMENT**

We are implementing a social distance detection model that uses deep learning and computer vision. The main goal of the project is to implement a real time system at railway stations, hospital, airport, colleges etc. where chances to spread of COVID-19. This model detects a group of people who are not keeping social distance. The project is implemented using deep learning based architecture YOU ONLY LOOK ONCE (YOLO). Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are. Our system first resizes the input image to 448 × 448, then runs a single convolutional network on the image, and threshold the resulting detections by the model’s confidence.

**LITERATURE REVIEW**

**2.1 REVIEW**

A literature review of social distance detection reveals that there are several different approaches to this problem, including the use of computer vision, wearable devices, and smartphone apps. Overall, the literature suggests that social distance detection can be an effective tool for preventing the spread of infectious diseases like COVID-19. However, further research is needed to fully evaluate the effectiveness of these various approaches, and to develop new and innovative solutions to this important problem.

***Computer Vision:*** The Computer vision method uses cameras and image processing algorithms to detect individuals in a crowd and measure the distance between them. It can be done using deep learning models such as YOLO, RetinaNet, or SSD for object detection, and then using geometric distance calculation or clustering to measure the distance between individuals. The system can be set up to alert individuals if they are too close to each other. Examples of computer vision-based social distance detection systems include Wobot, Cognian, and ProGlove.

***Wearables:*** These are devices that individuals can wear that can alert them if they are too close to other individuals. Some examples of wearables include smartwatches, wristbands, and lanyards. Examples of wearables-based social distance detection systems include Safe Spacer, Aura Aware, and TraceSafe.

***Bluetooth technology:*** This involves using Bluetooth technology to detect when two individuals are too close to each other. The system can be set up to alert individuals if they are too close to each other. A bluetooth device connects to another Bluetooth enabled device, it measures the signal strength as [RSSI](https://www.bluetooth.com/blog/proximity-and-rssi/) ( received signal strength indicator). The connection strength depends on the distance between the devices. The connection will be strong if two devices are nearby. As the device moves apart, the connection weakens.

It is also possible to scan for nearby devices and read the RSSI value without establishing a connection, this makes it very suitable for looking after nearby devices as the majority of people carry a smartphone with Bluetooth on. Examples of Bluetooth-based social distance detection systems include TraceTogether, COVIDSafe, and COVID Alert.

***Ultrasonic sensors:*** These are sensors that can be installed in public spaces to detect the distance between individuals. The system can be set up to alert individuals if they are too close to each other. Examples of ultrasonic sensors-based social distance detection systems include Sonic Notify and Ama XpertEye.

***LIDAR:*** LIDAR,which stands for Light Detection and Ranging, is a remote sensing method that uses light in the form of a pulsed laser to measure ranges(variable distances)to the earth.

**METHODOLOGY**

**PROPOSED SYSTEM**

We proposed a YOLO based system for social distance detection.YOLO is working based on the convolution neural network architecture.

**3.1** **Convolutional Neural Network(CNN)** :

It is a type of artificial neural network commonly used in deep learning for image and video recognition tasks.

CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input data, by applying a series of convolutional and pooling layers. The convolutional layers use filters to extract features from the input image or video frames, while the pooling layers reduce the size of the feature maps and retain the most salient features. The final output is then passed through a series of fully connected layers to generate the final predictions.

CNNs have revolutionized the field of computer vision and have achieved state-of-the-art results on a wide range of image and video recognition tasks, including object detection, image classification, and face recognition. They have also been successfully applied in other domains, such as natural language processing and speech recognition.

Some popular CNN architectures include AlexNet, VGGNet, ResNet, and InceptionNet. These architectures have different numbers of layers, filter sizes, and pooling strategies, and are optimized for different types of image recognition tasks.

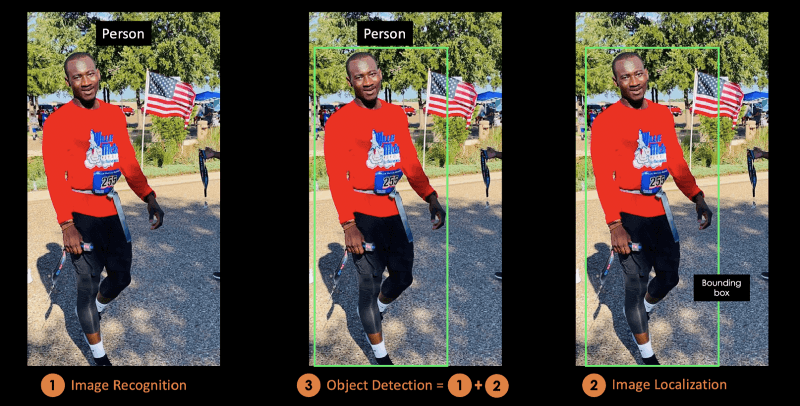
**3.2.OBJECT DETECTION**

Object detection is a technique used in computer vision for the identification and localization of objects within an image or a video. Image Localization is the process of identifying the correct location of one or multiple objects using bounding boxes, which correspond to rectangular shapes around the objects and classify it by its classes. The goal of object detection is to locate the presence of objects and classify them into predefined categories, such as people, animals, vehicles, and so on.

Object detection typically involves two main steps:

1. *Localization*: Determining the location of the object(s) in the image or video, typically by drawing a bounding box around the object(s).
2. *Classification*: Assigning a class label to each object within the bounding box, indicating the type of object that is present.

Object detection has many practical applications, such as in surveillance, autonomous vehicles, robotics, and medical imaging.



**3.3.SYSTEM ARCHITECTURE**

In this section, we will discuss the essential steps that are required to build a sequence design to determine and check if the social distancing rules are respected or not among the individuals on the thermal videos as seen in Figure.

* Streaming the thermal videos, which contains the individuals.
* Extracting the thermal video into frames.
* Applying YOLO architecture to detect only the individuals in thermal videos.
* Verify the number of the individuals that are in the thermal videos.
* Calculate the distance between the center point of the bounding boxes that contains the individuals in the thermal videos.
* Lastly, the algorithm will make the decision for violation or safe conditions for the individuals based on the number of individuals in the thermal videos, and the measured distance between the centroid of bounding boxes. This is to note that we made a level for violation with a threshold set points for the measured distance between the center points of the bounding boxes. The violation level is defined as Risk, which is marked with a red color for the bounding box. We marked the safe condition with a green color for the bounding box.

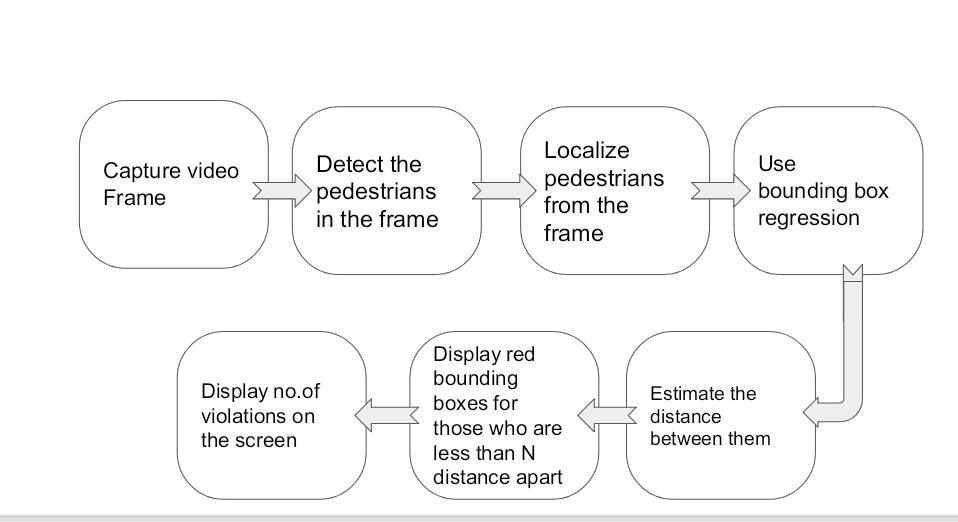
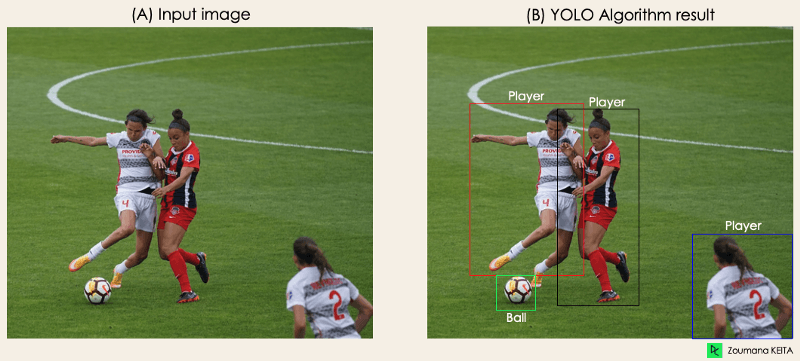


Figure: System architecture

**3.4.YOLO-You Only Look Once**

This is an algorithm that detects and recognizes various objects in a picture (in real-time). The YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. YOLO is generally a real-time object detection algorithm that follows a regression approach which essentially is quite significant. This algorithm helps to discover the bounding boxes as nicely as the class of the object subtly specifying its location. YOLO algorithm first splits the frame into N framework where each framework is having equal dimension region of S\*S. These N frameworks are liable for image localizing and determining. Moreover, the objects are determined in only one execution of the YOLO algorithm in a sort of major way. Hence, this is the reason the YOLO algorithm outperforms all other models.

An example for Yolo Algorithm result:



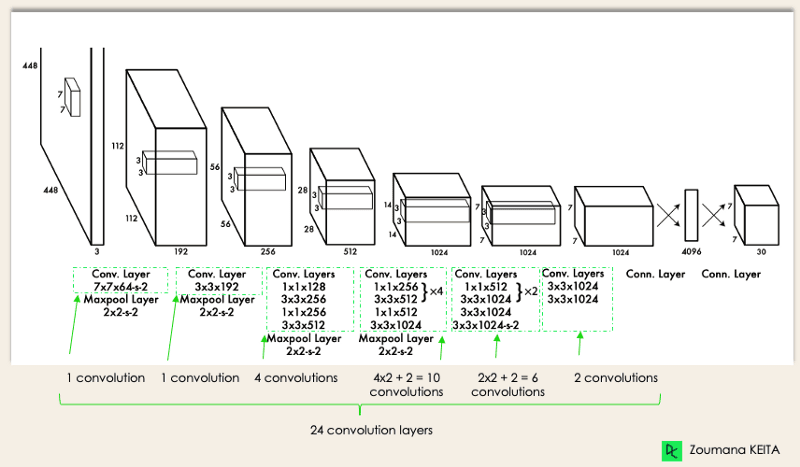
**Importance of YOLO**

YOLO (You Only Look Once) is an important object detection algorithm in computer vision. It is a real-time object detection system that can detect multiple objects in an image or video frame with high ***accuracy and speed****.*

One of the key advantages of YOLO is its***speed***. Traditional object detection algorithms typically involve running a classifier on a large number of candidate regions, which can be slow and computationally expensive. YOLO, on the other hand, uses a single neural network to directly predict the bounding boxes and class probabilities for all objects in the image, making it much faster and more efficient.

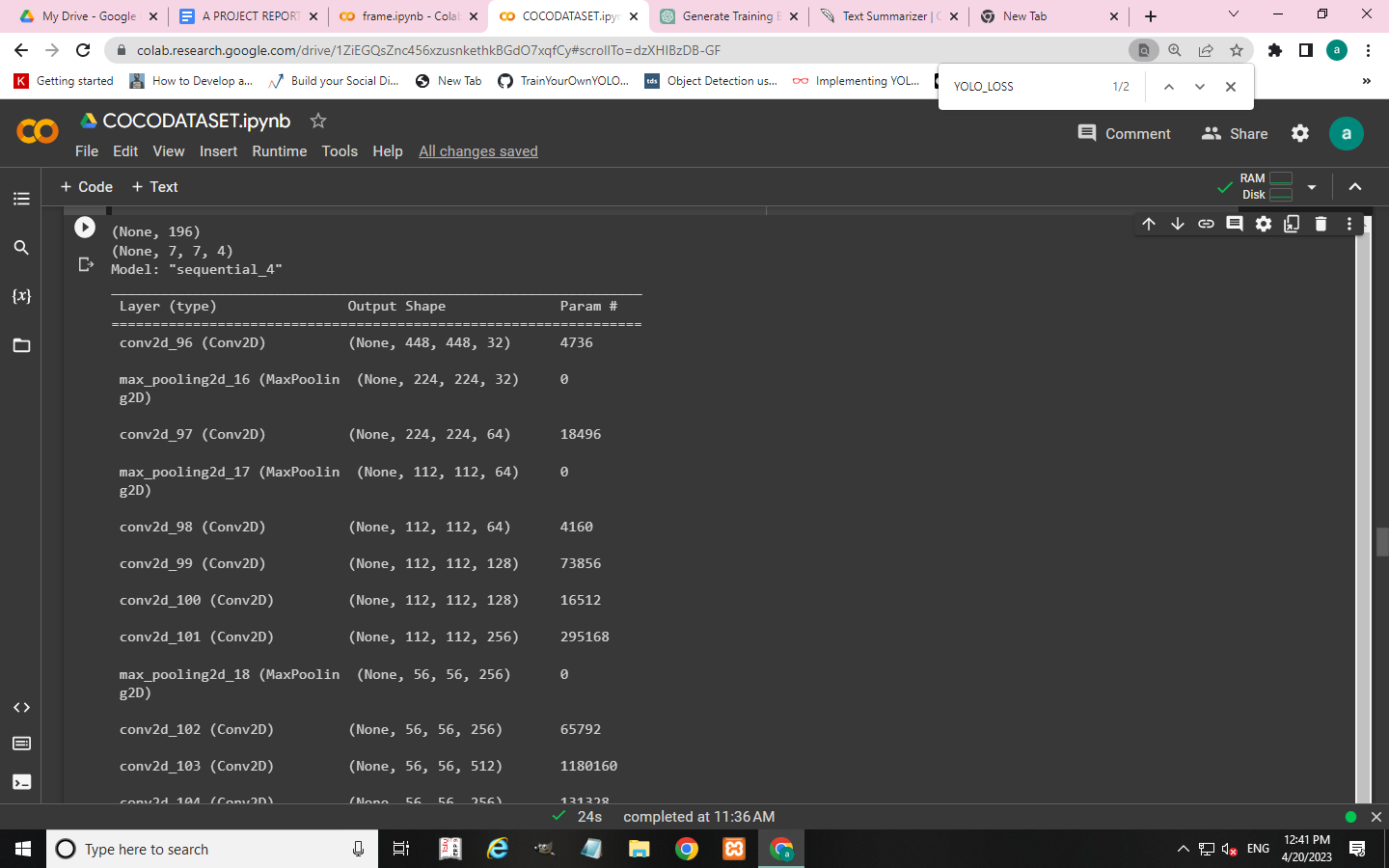
Another advantage of YOLO is its high ***accuracy***. The YOLO algorithm uses a unified architecture to simultaneously detect objects and classify them into different categories. This allows it to achieve high accuracy even in complex scenes with multiple overlapping objects.

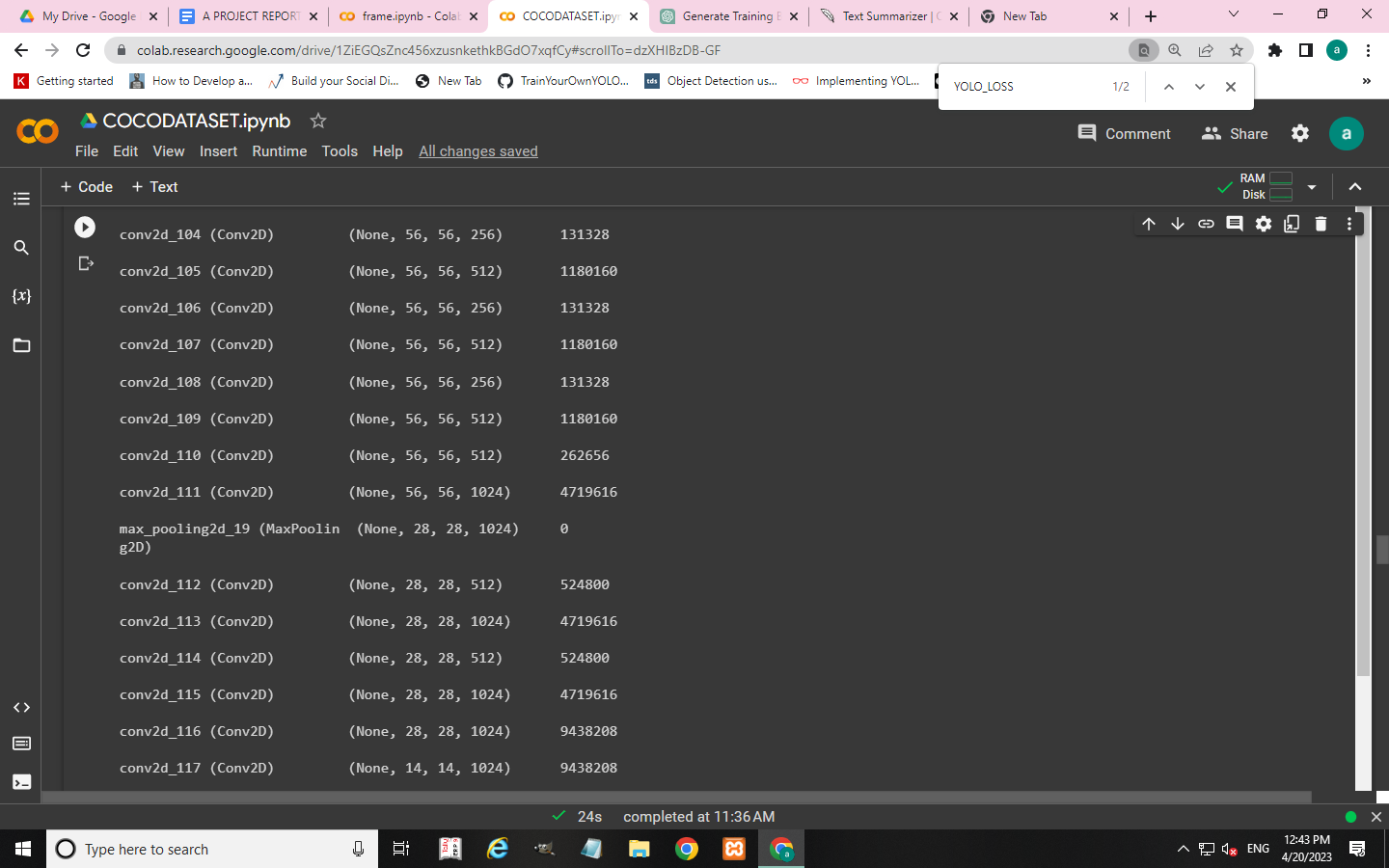
**YOLO Architecture**

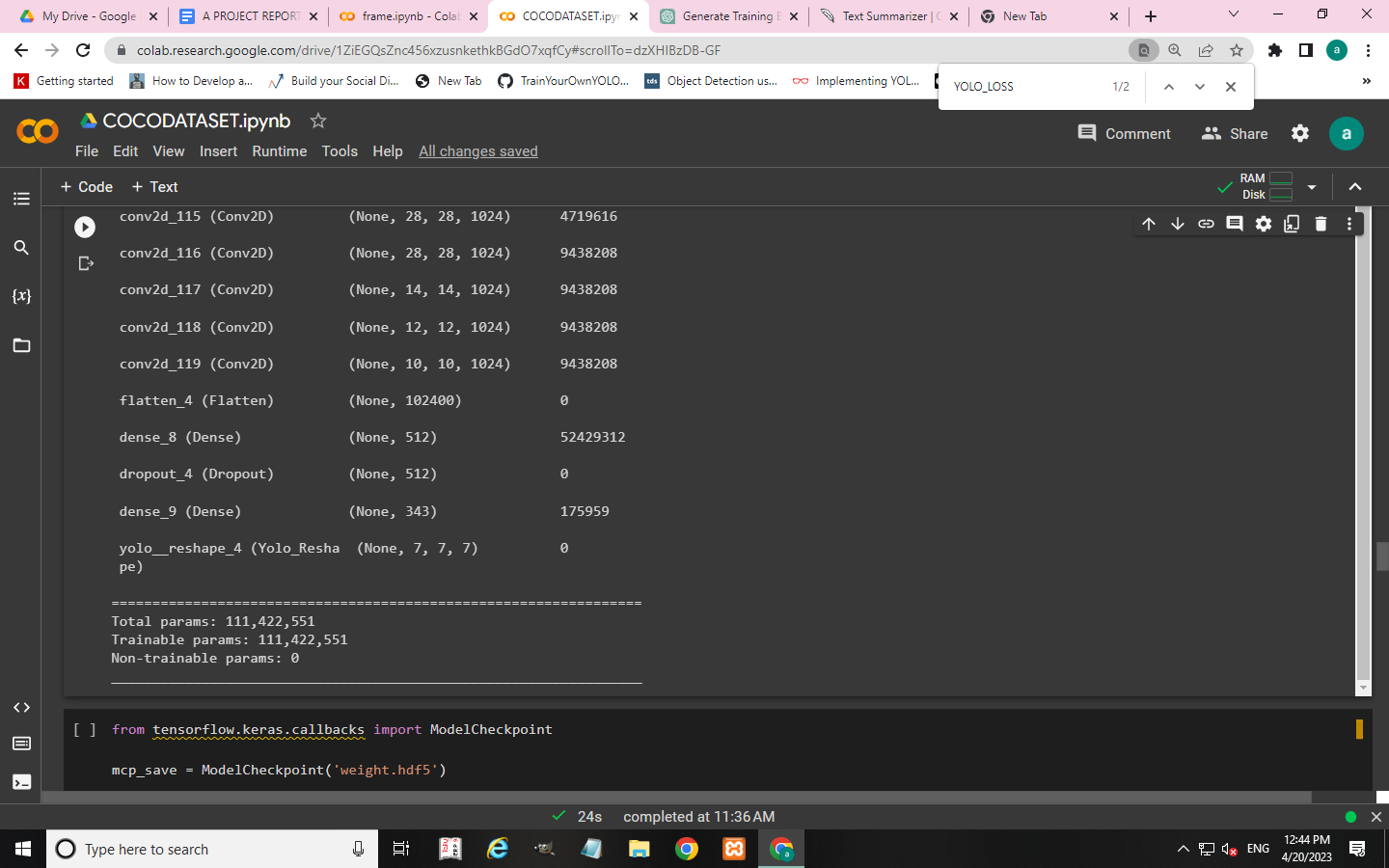


The Sequential model for object detection using the YOLO algorithm is created using the Sequential API. It starts with a Conv2D layer with 32 filters, a kernel size of (7, 7) and a stride of (1, 1). The activation function is a LeakyReLU activation function with an alpha value of 0.1 and a regularization function of L2. The output is then fed to a MaxPooling2D layer with a pool size of (2, 2) and a stride of (2, 2). The final part of the model consists of fully connected layers (Dense layers) and a reshape layer (Yolo\_Reshape). The Flatten layer converts the output of the last Conv2D layer into a 1D vector, which is fed into a fully connected layer with 512 units and a dropout layer with a rate of 0.5. Finally, a dense layer with 343 units and a sigmoid activation function is used to generate the output, which is reshaped using the Yolo\_Reshape.

**Layers of YOLO**

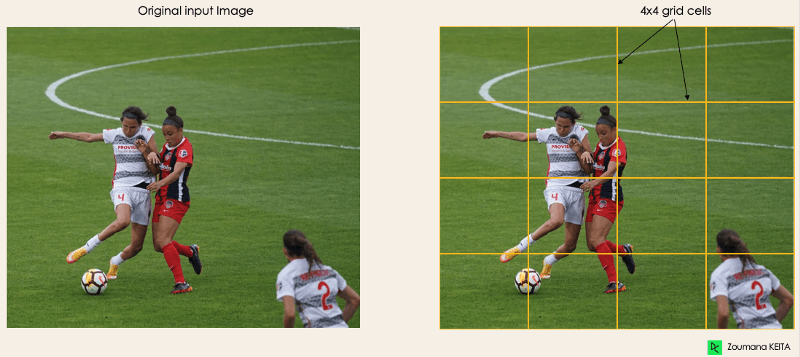




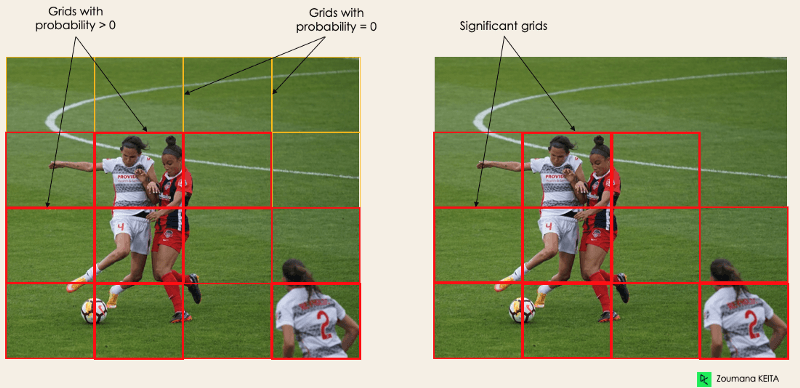


**Working of Yolo**

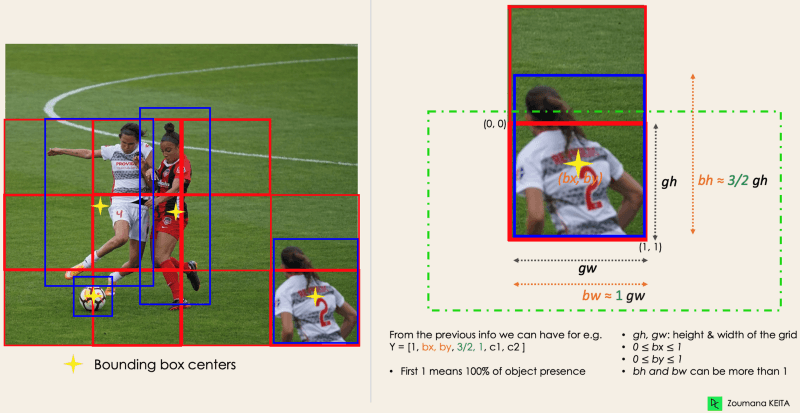
***1. Residual blocks*:** This first step starts by dividing the original image into NxN grid cells of equal shape, Each cell in the grid is responsible for localizing and predicting the class of the object that it covers, along with the probability/confidence value.



***2. Bounding box regression***: The next step is to determine the bounding boxes which correspond to rectangles highlighting all the objects in the image.YOLO determines the attributes of these bounding boxes using a single regression module in the following format, where Y is the final vector representation for each bounding box.

**Y=[pc, bx, by, bh, bw, c1, c2]**

***3. Intersection Over Unions or IOU****:* Most of the time, a single object in an image can have multiple grid box candidates for prediction, even though not all of them are relevant. The goal of the IOU (a value between 0 and 1) is to discard such grid boxes to only keep those that are relevant to only keep those that are relevant.



**4. *Non-Max Suppression or NMS*:** Setting a threshold for the IOU is not always enough because an object can have multiple boxes with IOU beyond the threshold, and leaving all those boxes might include noise. Here is where we can use NMS to keep only the boxes with the highest probability score of detection.

**SOCIAL DISTANCE CALCULATION**

To measure the distance between individuals in social distance detection using YOLO we use the Euclidean distance method. This method involves calculating the distance between the centroids of the bounding boxes around each person detected by YOLO. The centroid of a bounding box is the point at the center of the box, which can be calculated by taking the average of the x and y coordinates of the top left and bottom right corners of the box. To calculate the Euclidean distance between two centroids, use the following formula:

***distance = sqrt((x2 - x1)^2 + (y2 - y1)^2)***

Where (x1, y1) and (x2, y2) are the centroid coordinates of the two bounding boxes.Once we calculate the distances between all pairs of individuals in the scene, we use a threshold distance to determine whether they are maintaining a safe distance from each other. In this project the threshold distance is set to 50 pixels, then any pair of individuals with a distance less than 50 pixels would be considered to be violating social distancing guidelines. Otherwise individuals are not violating the social distance guidelines.

|  |  |
| --- | --- |

**IMPLEMENTATION**

**4.1. HARDWARE REQUIREMENT**

The hardware in which this model is executed are:

● RAM – 12.7 GB

● Operating System – Windows 11

● Disk 78.2GB

**4.2 SOFTWARE REQUIREMENTS**

The programming language used to develop this application is Python and the Google colab

● Programming Language – Python

●Google colab

● Deep Learning Framework - Tensorflow

**4.2.1.OPEN CV**

OpenCV is an Open source library that is used for picture processing, computing device learning, and computer vision. OpenCV has additionally been used in true time operations, the place where it makes the responsibilities greater and easier. Using Opencv we can manage films and snapshots from which the objects, human faces, and even handwriting can be identified. When OpenCV is built in with one of a kind libraries it affords.

**4.2.2.PYTHON**

Python is a high-level, markup language that emphasizes code readability and is dynamically-typed and garbage-collected. It supports multiple programming paradigms and is often described as a "batteries included" language due to its comprehensive standard library.

**4.2.3.TENSORFLOW**

TensorFlow is a free and open-source software library for dataflow and differentiable programming, available for 64-bit Linux, macOS, Windows, and Android and iOS mobile computing platforms.

**4.2.4.KERAS**

Keras is a human-centric API that adheres to industry best practices for reducing psychological feature load. It is a lightweight Python deep learning library that runs on top of Theano or TensorFlow and is available under the Massachusetts Institute of Technology's permissive license. François Chollet created Keras with four guiding principles in mind: modularity, minimalism, and extensibility. It runs on both GPUs and CPUs.

**4.2.5.NUMPY**

Numpy may be a multipurpose library of python which is employed for handling arrays. Numpy essentially stands for Numerical Python which functions in the domain of algebra. It gives array objects quicker than the python list. Since Numpy may be a library of python but it’s partially written in python and for computations, C language is employed.

**4.3.DATABASE USED**

**4.3.1.COCO DATASET**

Common objects in Context or COCO is particularly a large-scale object detection segmentation and a captioning dataset which for all intents and purposes is fairly significant. It consists of a couple of sets of excessive datasets each made for a unique machine learning task. Algorithms for object detection and classification can essentially be pre-trained using the dataset in a major way. The COCO dataset has a specific layout that exactly defines how your notations that are bounding boxes, objects, classes, etc, are saved on disk. The COCO dataset is deliberately biased towards person class, so it determines person class more accurately which is quite significant.

**4.4.Experimental setup**

*model. compile(loss=yolo\_loss,optimizer='adam', metrics=('accuracy')*

**LOSS FUNCTION**

The yolo loss function is used to train the YOLO algorithm, and it aims to minimize the difference between the predicted bounding boxes and the ground truth bounding boxes.YOLO loss function is a combination of losses of bounding box,class and predictions. All of them are Mean-Squared error losses and are modulated by some scalar meta-parameter or IoU score between the prediction and ground truth.The YOLO loss function has two components: localization loss and classification loss. The localization loss measures the error in predicting the coordinates of the bounding box, while the classification loss measures the error in predicting the class probabilities. The localization loss is computed as the sum of the squared differences between the predicted and ground truth values of the bounding box coordinates.The classification loss is computed as the sum of the cross-entropy losses between the predicted and ground truth class probabilities.

*yolo loss=confidence loss+class loss+bounding box loss*

**OPTIMIZER**

In machine learning and deep learning, an optimizer is an algorithm or method used to adjust the parameters of a model in order to minimize the error or loss function. The goal of the optimizer is to find the optimal set of parameters that result in the best possible performance of the model on the training data.Here we use Adam optimizer,Adam (Adaptive Moment Estimation) is a popular optimization algorithm used in deep learning that is based on adaptive estimates of lower-order moments (mean and variance) of the gradients.

**METRICS**

Metrics refer to the evaluation criteria or performance measures used to assess the quality of the model's predictions.Here we use *metrics=['accuracy']* means that the accuracy of the model will be used as the evaluation metric. Accuracy is a common metric used in classification tasks and measures the proportion of correctly classified samples out of the total number of samples.By specifying metrics=['accuracy'], Keras will compute the accuracy of the model on the validation set during training, and report it in the training logs. This allows the user to monitor the performance of the model and make adjustments to the training process if necessary.

**EXPERIMENTAL AND RESULT**

**ANALYSIS**

**RESULT**

The YOLO algorithm works properly in real-time object detection operations.It determines the bounding boxes and class of the image that consists of objects in solely one execution of the algorithm. It gives a lot fairly higher performance in a generally big way.Also,the system will help users by displaying the number of violations and alert messages,so that they can keep a secure distance.It also displays the pedestrians that are violating red bounding boxes and who are not into the green.All this is proven in the images illustrated below.The pedestrians who are close to each other are displayed in to red bounding boxes and the ones who are no longer are displayed into the green. Whole number of violations are displayed on the screen.

**RESULT ANALYSIS**

In social distance detection using YOLO (You Only Look Once) algorithm, min\_threshold and min \_ distance are important parameters used for setting the minimum confidence threshold and minimum distance threshold.

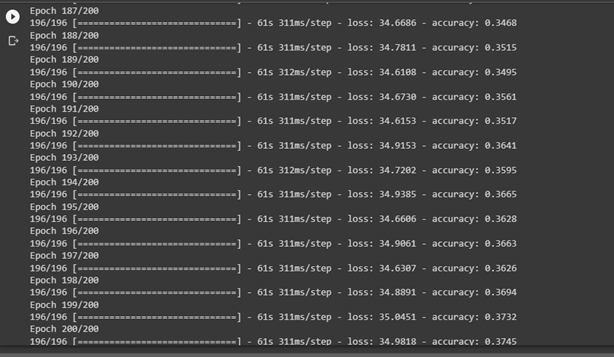
The min\_threshold is used to filter out the predictions made by the YOLO algorithm that have a confidence score below a certain threshold. For instance, the min\_threshold is set to 0.3, which means that only predictions that have a confidence score of 0.3 or higher will be considered as valid detections. Any predictions that have a lower confidence score will be ignored.

The min\_distance is used to define the minimum distance that should be maintained between two individuals in the image. If the distance between two individuals is less than the min\_distance, then the YOLO algorithm considers them as not maintaining social distance.Here we set the minimum safe distance as 50(pixels) that two people can be from each other.

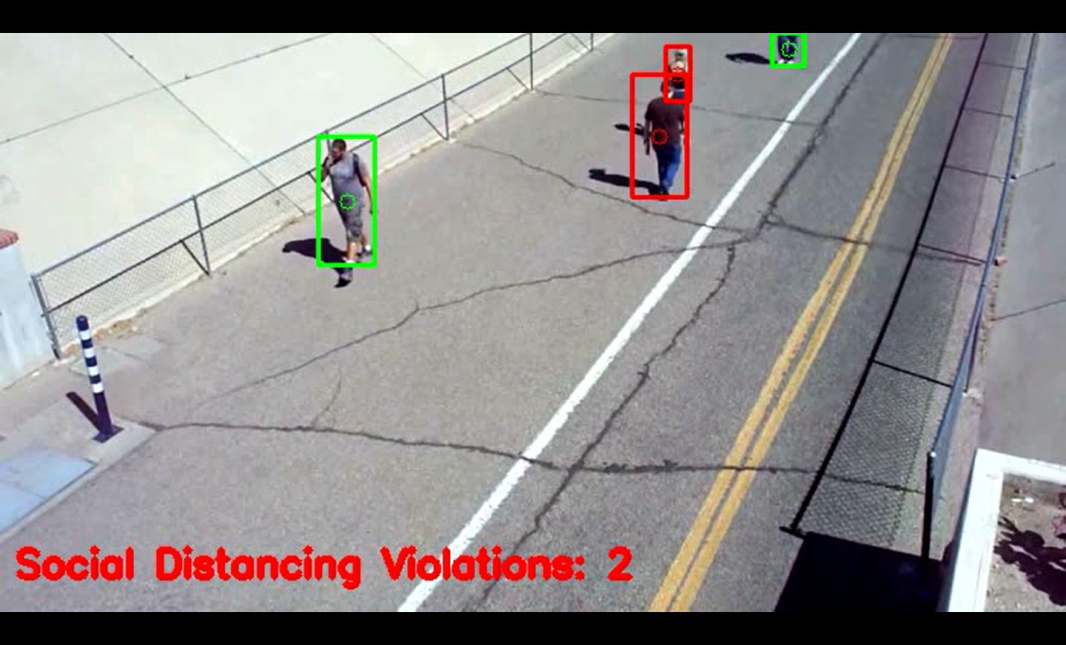
In model epoch refers to a complete iteration through the entire training dataset during the training process. The number of epochs chosen for training a YOLO model for social distance detection can vary based on the size and complexity of the dataset, as well as the hardware and computational resources available.

* Model is trained for 200 epochs

*model. fit(xtrain,ytrain, batch\_size=4, epochs =200, verbose = 1*



**SCREEN SHOTS**



**LIMITATIONS AND FUTURE SCOPE**

The developed system can detect the live video streams but does not keep a record. Unlike the CCTV camera footage, the admin cannot rewind, play or pause it. The proposed system can be integrated with databases of respective organizations to keep a record of the people who were not keeping social distance.

**CONCLUSION**

The system was designed keeping in mind the simplicity of use via the users so that users will particularly be able to use the device in their everyday life in a subtle way. Social distancing detection device that essentially helps users in maintaining a safe distance and alert the ones who are violating by showing the alert message on the screen. This system proposes a very high-quality deep gaining knowledge of the system that simplifies the work of preserving social distance using the YOLO algorithm which was pre-trained through the COCO dataset. The result indicates that the proposed approach is supposed to be used in any working surroundings because of its velocity and accuracy.

**REFERENCE**

# <https://www.maskaravivek.com/post/yolov1/>

# <https://cocodataset.org/>

# <https://www.slideshare.net/irjetjournal/social-distancing-detection>

# <https://github.com/JY-112553/yolov1-keras-voc/blob/master/tiny_yolov1.py>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*