Mohamed Khider University of Biskra
Faculty of Science and Technology
Department of Electrical Engineering

MASTER THESIS

Electrical Engineering
Telecommunication
Networks and Telecommunication

Ref.: Entrez la reference du document

Submitted and Defended by:
HAMED AICHA & HERABI ITHAR

On: Sunday, June 26, 2022

OpenCV Social Distancing Detector

Board of Examiners:

Dr	OUAMANE ABDEALMALIK	**MCA**	University of Biskra	**President**
Dr	FEDIAS MERIEM	**MCB**	University of Biskra	**Examiner**
Dr	MEDAOUAKH SAADIA	**MCB**	University of Biskra	**Supervisor**

University year: 2021-2022

Mohamed Khider University of Biskra
Faculty of Science and Technology
Department of Electrical Engineering

MASTER THESIS

Electrical Engineering
Telecommunication
Networks and Telecommunication

Réf.:Entrezlaréférencedu document

Submitted and Defended by:
HAMED AICHA & HERABI ITHAR

On: Sunday, June 26, 2022

OpenCV Social Distancing Detector

In:………………………………

Presented by: **Favorable opinion of the supervisor:**

HAMED AICHA Dr. MEDAOUAKH SAADIA

HERABI ITHAR

Favorable opinion of the jury president

DR.OUAMANE ABDEALMALIK

Stump and signature

Resume

Social distancing is a method used to control the spread of contagious diseases and it is the most effective no pharmaceutical way to prevent the spread of disease. Social distancing implies that people should physically distance themselves from one another, reducing close contact, and thereby reducing the spread of a contagious disease (such as coronavirus). The aim of this work is to study and apply an algorithm for detecting social distance using deep learning and computer vision between people to control the spread of covid-19. This application is developed to give alerts to people for maintaining social distance in crowded places. By using pre-recorded video as input and the open-source object detection pretrained model using the YOLO algorithms, we can tell if people are following social distancing or not, and based on that we are creating red or green bounding boxes over it. It is also working on web cameras, CCTV..etc, and can detect people in real-time. In this work, we have used the YOLOv3 method on the tested video and have obtained very satisfactory results, which have been discussed.

Keywords: social distancing detector, deep learning, Yolo, yolov3, object detected.

ملخص:

التباعد الاجتماعي هو طريقة تستخدم للسيطرة على انتشار الأمراض المعدية وهي الطريقة الأكثر فعالية من الأدوية غير الصيدلانية لمنع انتشار المرض. يعني التباعد الاجتماعي أنه يجب على الأشخاص أن يبتعدوا بأنفسهم جسديًا عن بعضهم البعض ، مما يقلل الاتصال الوثيق ، وبالتالي الحد من انتشار الأمراض المعدية (مثل فيروس كورونا). الهدف من هذا العمل هو دراسة وتطبيق خوارزمية لاكتشاف المسافة الاجتماعية باستخدام التعلم العميق والرؤية الحاسوبية بين الناس للسيطرة على انتشار كوفيد -19. تم تطوير هذا التطبيق لإعطاء تنبيهات للأشخاص للحفاظ على مسافة اجتماعية في الأماكن المزدحمة .باستخدام الفيديو المسجل مسبقًا كمدخلات ونموذج الكشف عن الكائنات مفتوح المصدر الذي تم اختباره مسبقًا باستخدام خوارزميةYOLOv3 ، يمكننا معرفة ما إذا كان الأشخاص يتابعون التباعد الاجتماعي أم لا ، وبناءً على ذلك نقوم بإنشاء مربعات حدود حمراء أو خضراء فوقها. كما أنها تعمل على كاميرات الويب ، وكاميرات المراقبة ، وما إلى ذلك ، ويمكنها اكتشاف الأشخاص في الوقت الفعلي. في هذا العمل ، استخدمنا طريقة YOLOv3 على الفيديو الذي تم اختباره وحصلنا على نتائج مرضية للغاية ، والتي تمت مناقشتها.

الكلمات المفتاحية : كاشف التباعد الاجتماعي،التعلم العميق،yolo،yolov3،اكتشاف الكائن

Sommaire

List of figures

Chapter 1

Chapter 2

Chapter 3

List of tables

Chapter 1:

Acronyms

CNN	Convolutional Neural Network
DL	Deep learning
AI	Intelligence Artificially.
ML	Machine Learning
RNN	Recurrent Neural Network.
SSD	Single Shot MultiBox Detector .
SPP	Spatial Pyramid Pooling
RPN	Region proposal network .
R-CNN	Region-based Convolutional Neural Network

Introduction general

When the novel coronavirus (Covid-19) pandemic emerges, the spread of the virus has left public keep anxiety if they do not have any effective cure. The World Health Organization (WHO) has declared Covid-19 as a pandemic due to the increase in the number of cases reported around the world **[1].** To contain the pandemic, many countries have implemented a lockdown where the government enforced that the citizens to stay at home during this critical period. The public health bodies such as the Centers for Disease Control and Prevention (CDC) had to make it clear that the most effective way to slow down the spread of Covid-19 is by avoiding close contact with other people. To flatten the curve on the Covid-19 pandemic, the citizens around the world are practicing physical distancing.

To implement social distancing, group activities and congregations such as travel, meetings, gatherings, workshops, praying had been banned during the quarantine period. The people are encouraged to use phone and email to manage and conduct events as much as possible to minimize the person-to-person contact. To further contain the spread of the virus, people are also informed to perform hygiene measures such as frequently washing hands, wearing mask and avoiding close contact with people who are ill. However, there is a difference between knowing what to do to reduce the transmission of the virus and putting them into practice, Individuals, communities, businesses, and health care organizations are all part of a community with their responsibility to mitigate the spread of the Covid-19 disease **[1].** Hence, this work aims to facilitate the enforcement of social distancing by providing automated detection of social distance violation in workplaces and public areas using a deep learning model. In the area of machine learning and computer vision, there are different methods that can be used for object detection. These methods can also be applied to detect the social distance between people. We will explain more in the following points **[2]**:

1. Deep learning has gained more attention in object detection was used for human detection purposes.
2. Develop a social distancing detection tool that can detect the distance between people to keep safe.
3. Evaluation of the classification results by analyzing real-time video streams from the camera.

The YOLOv3 object detection framework uses to identify people in a given frame. After computing a pairwise distance between the centroids of the detected bounding boxes of people, this value is compared to a predefined minimum pixel threshold. By mapping these pixels to measurable units, detection of violations in social distancing norms is identified. Most of the research projects concerning an automated social distancing monitoring system used use Deep Learning, Image processing, and CNN[3](Convolutional Neural Networks) to develop the said system. The most popular "State of The Art" CNN model is YOLOv1-5, RCNN (Region-Based CNN), Fast RCNN, Faster RCNN, and SSD (Single-shot Detector), These are a few CNN models that are mostly used is object detection because of their high accuracy rate and Frames rate or FPS.

The main objective of this work is to support the reduction of the coronavirus spread and its economic costs by providing an AI-based solution to automatically detect violations of social distancing among individuals [4]. In this work, we study how to detect where each person is by the YOLOv3 method, which deals with taking the entire image in one case and predicting bounding box coordinates and class probabilities for these boxes. The created bounding boxes help to identify groups of people who satisfy the distance threshold calculated with the help of calculating the Euclidean distance between pedestrians and return a bounding box that turns red if the distance between two people is dangerously close. The number of violations is confirmed by counting the red rectangles that appeared on nearby people.

We have chosen to organize our study around three main chapters as follows:

- The first chapter, we will start with General Information about Social Distancing then we will get to know the deep learning and his methods.
- The second chapter, we will see the Object Detection using YOLOv3 and Social distancing detector steps.
- The third chapter, we will see Implementation and experimental results, we are going to learn how to perform social distancing detection in video streams using: OpenCV, Python, and Deep Learning.

We will end this dissertation with a general conclusion and the perspectives

Chapter 1

General Information on Social Distancing and Object Detection

1.1. Introduction

Social distancing, as is considered the best solution to address the coronavirus, by maintaining a safe distance between people, a detection tool was developed to alert people to maintain a safe distance between them, By providing automated detection of the violation of social distance in public, using a model deep learning in the field of machine learning and computer vision.

In the first chapter, we'll briefly discuss what social distancing is! an overview of object detection, and how deep learning can be used to implement a social distancing detector.

1.2. What is social distancing?

Social distancing, also called "physical distancing," means keeping a safe space (**stay at least 6 feet**) between yourself and other people who are not from your household in both indoor and outdoor spaces. The objective is to reduce transmission, reducing the size of the epidemic peak, and spreading cases over a longer time to relieve pressure on the healthcare system. Social distancing is arguably the most effective non pharmaceutical way to prevent the spread of a disease by definition, if people are not close together, they cannot spread germs. **[5]**

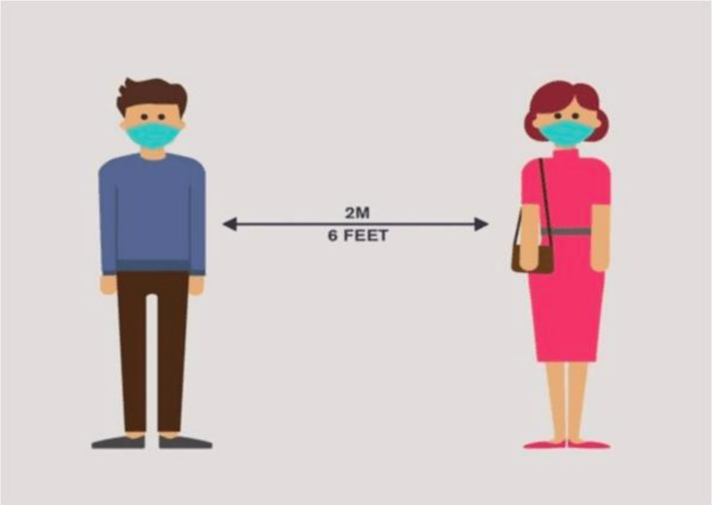


Figure 1.1: keeping a safe social distance. [3]

1.3. Why would social distancing measures be used

Public health and social measures contribute to preventing outbreaks, and are therefore critical in limiting further spread of COVID-19.Today, social distancing measures are most often thought about as a way to slow the spread of pandemic influenza, some examples of social distancing measures that would be conducted during a pandemic include **[6]:**

- Public and private colleges suspending classes, going to Web-based learning and canceling all large campus meetings and gatherings;
- Public and private libraries modifying their operations and restricting people from gathering by allowing people to come in only to pick up materials that have been reserved or requested on-line or by telephone;
- Business changing company practices, having employees telecommute and canceling any large meetings or conferences.

This information can help us to take basic measures and take them when the government announces the distancing law into effect.

1.4. Overview of object detection

Object detection is a fundamental research field in computer, the concept of computer vision is based on teaching computers to process an image or video at a pixel level and understand it.

Object detection aims to locate and identify the interested regions in an image at the same time **[7],** it is widely used and has strong practical value in real life such as video fire detection ,security monitoring .Well-researched domains of object detection include include face detection, pedestrian detection and vehicle detection **[8]** .The goal of object detection is to recognize instances of a object classes (e.g. {people, cars, bikes, animals}) and describe the locations of each detected object in the image using a*bounding box*. Two examples are shown below.

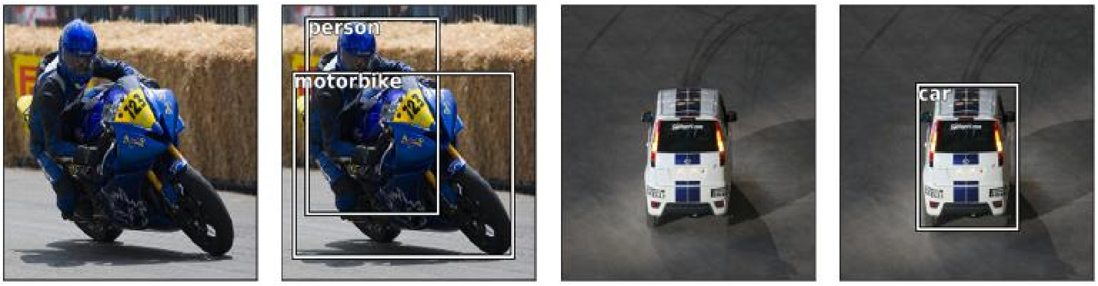


Figure 1.2: Image using abounding box [*8/*

We'll use rectangles to describe the locations of each object, which may lead to imperfect localizations due to the shapes of objects. An option approach would be segmentation which provides localization at the pixel-level. It takes 4 variables to uniquely identify a rectangle**. [9]**

Just like multi-label image classification, we can have multi-class object detection where we detect multiple kinds of objects in a single image as shown in figure below

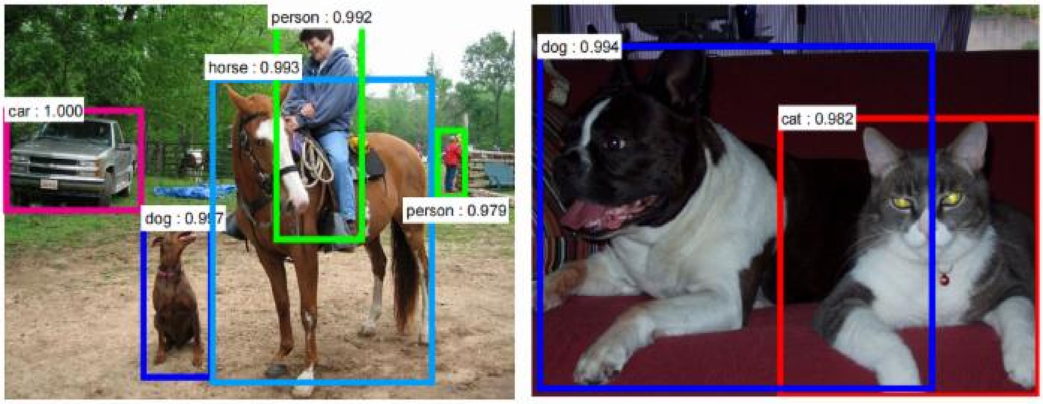


Figure1.3: Result of Objection detection task by using different algorithms in deep learning [8].

1.4.1. Principal of Object detection

The principle of object detection is based on stages, where the image is examined if it contains any object, then we classify the object and then locate the object in the image, and to have a good object detection method, it is necessary to have a solid region detection algorithm like (SSD, SSD-MobileNetv3,Yolo...) and a good image classification algorithm. We will explain these stages as follows:

→1/ classification: aims at assigning an image to one of a number of different categories (e.g. car, dog, cat, human, etc.), essentially answering the question “What is in this picture?” One image has only one category assigned to it. **[10]**

→2/ localization: allows us to locate our object in the image, so our question changes to “What is it and where it is*?”***[10]**

→3/Object detection: object detection identifies the object*s* well as its location in an image. Object detection is a computer technology that determines the location and size of the object in the digital image. It detects the characteristics of objects and ignores everything else. The selection of objects is carried out by classification using a database model. [9]

Here’s a classic example to understand this difference:

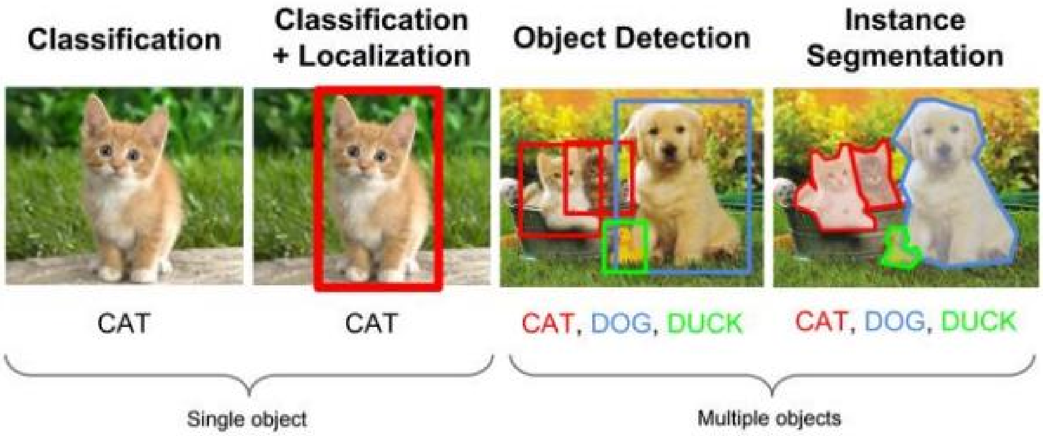


Figure1.4:Exemple de classification d'image en utilisant la détection d'objet [11].

1.5. Deep Learning

Deep learning-based object detection is an important element of data science, It is a field that is based on learning and improving on its own by examining computer algorithms, we present a three primary DL methods you'll encounter:

- R-CNN and their variants, including the original R-CNN, Fast R- CNN, and Faster R-CNN;
- Single Shot Detector (SSDs);
- YOLO.

1.5.1. Definition of deep learning

Deep learning is a form of machine learning, deep learning has recently seen a surge in popularity as a way to accelerate the solution of certain types of difficult computer problems, most notably in the computer vision and natural language processing (NLP) fields.**[13]**

Deep Learning is based on the idea of artificial neural networks and is designed to handle large amounts of data by adding layers to the network. A deep learning model can extract features from raw data thanks to multiple layers of processing and learn about these features incrementally through each layer with minimal human intervention**. [12]**

1.5.2. How Deep Learning Works

Computer programs that use deep learning go through much the same process as the toddler learning to identify the dog. Each algorithm in the hierarchy applies a nonlinear transformation to its input and uses what it learns to create a statistical model as output.

Iterations continue until the output has reached an acceptable level of accuracy. The number of processing layers through which data must pass is what inspired the label deep**. [14]**

1.5.3. The relation between Machine Learning and Deep Learning and AI?

1.5.3.1 Artificial intelligence or AI :

Artificial Intelligence is one of the technological innovations that happened, to replace the manual work that is done by human in various fields. Artificial Intelligence is a branch of science and technology that creates intelligent machines and computer programs to perform various tasks which requires human intelligence. It is systems that mimic various functions which a human can do**. [15]**

AI uses external data like the big data in order to achieve excellent performance for the given tasks**.[15]**

1.5.3.2 Machine learning:

Machine learning is the tool used for large-scale data processing. It is well-suited to complex datasets that have huge numbers of variables and features. One of the strengths of many machine learning techniques, and deep learning, in particular, is that they perform best when used on large datasets, this improving their analytic and predictive power. In other words, machine learning techniques, and deep learning neural networks, in particular, learn best when they can access large datasets where they can discover patterns and regularities hidden in the data **[16]**.

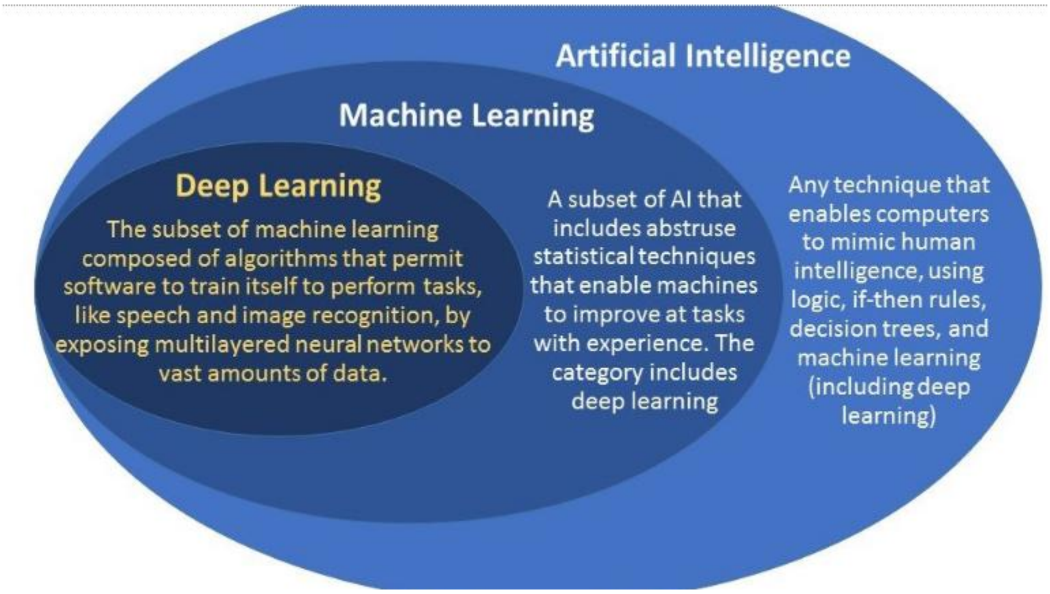
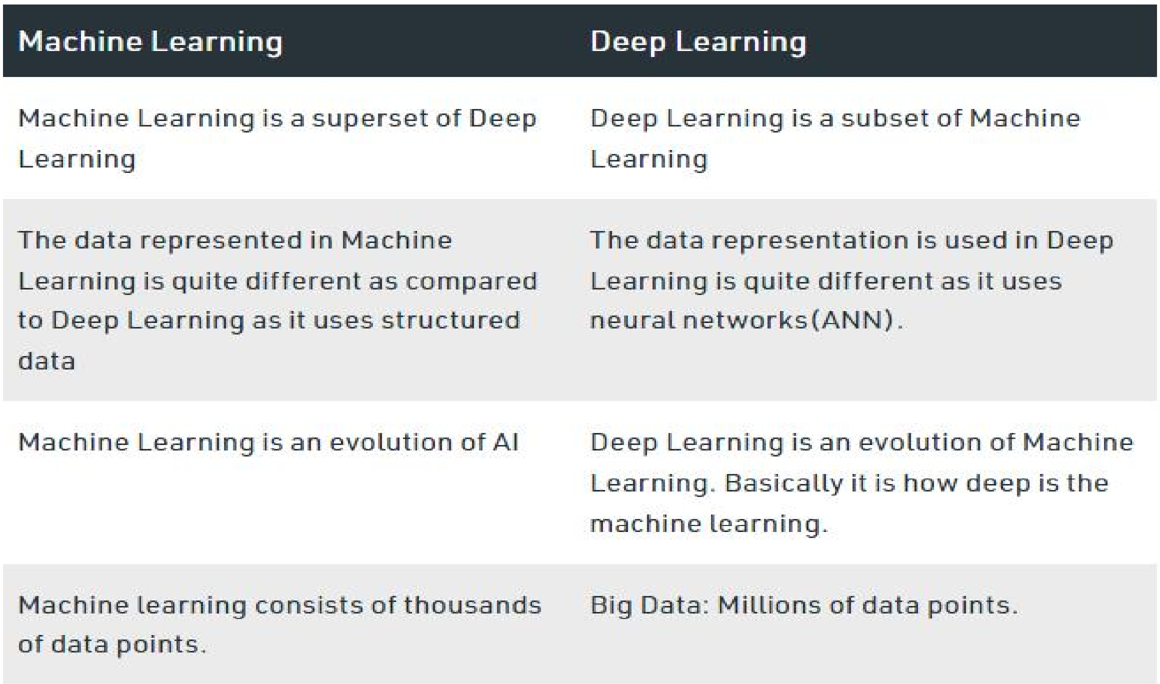

Artificial Intelligence
Machine Learning
Deep Learning
The subset of machine learning composed of algorithms that permit software to train itself to perform tasks, like speech and image recognition, by exposing multilayered neural networks to vast amounts of data.
A subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience. The category includes deep learning
Any technique that enables computers to mimic human intelligence, using logic, if-then rules, decision trees, and machine learning (including deep learning)

Figure 1.5: The Relation between Machine Learning and DL and AI [17].



Machine Learning	Deep Learning
Machine Learning is a superset of Deep Learning	Deep Learning is a subset of Machine Learning
The data represented in Machine Learning is quite different as compared to Deep Learning as it uses structured data	The data representation is used in Deep Learning is quite different as it uses neural networks(ANN).
Machine Learning is an evolution of AI	Deep Learning is an evolution of Machine Learning. Basically it is how deep is the machine learning.
Machine learning consists of thousands of data points.	Big Data: Millions of data points.

Table 1.1: The Difference between DL and Machine learning [10]

Machine learning, deep learning, and artificial intelligence all have relatively specific meanings but are often broadly used to refer to any sort of modern, big-data-related processing approach. [**13]**

1.5.4 Methods of Deep Learning:

1.5.4.1 R-CNN

Object detection presents two challenges: finding objects and classifying them. This is the goal of R-CNN: to divide the difficult task of object detection into two easier tasks**[18]**:Objects Proposal (find objects) and Region Classification. Region proposals have a higher probability of containing an object.

The R-CNN algorithm was proposed by Girshick **[19]**, which is the first real target detection model based on convolutional neural networks. The improved R-CNN model achieves 66% mAP. As shown in figure (1.6), the model first uses the Selective Search to extract approximately 2000 region proposals of each image to be detected. Then the size of each extracted proposals is uniformly scaled to a fixed-length feature vector and these extracted image features are input into the SVM classifier for classification. Finally, a linear regression model is trained to perform the regression operation of the bounding box. Compared with the traditional detection method, the accuracy of the R-CNN does improve a lot, but the amount of calculation is very large, and the calculation efficiency is too low.

Secondly, directly scaling the region proposal to a fixed-length feature vector may cause object distortion **[20]**

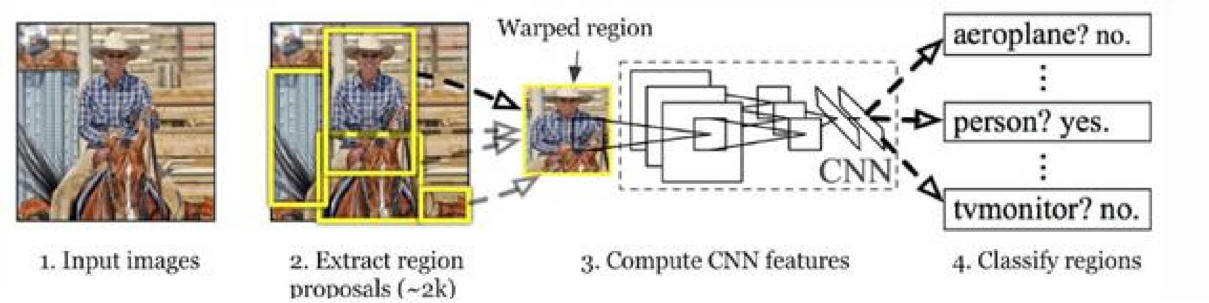


Figure 1.6 : the architecture of the R-CNN algorithm [11].

1.5.4.2 Fast R-CNN:

Fast R-CNN has also been proposed by Girshick to improve the speed of training and testing and to improve detection accuracy. In fast R-CNN, instead of processing region proposals by the CNN Network, the input image is processed and a convolutional feature map is produced. Regions of interest (ROI) are generated from the feature maps and passed to the fully connected layer. It should be noted that the R-CNN fast is nine times faster in training and 213 times faster in inference than the R-CNN **[21].**

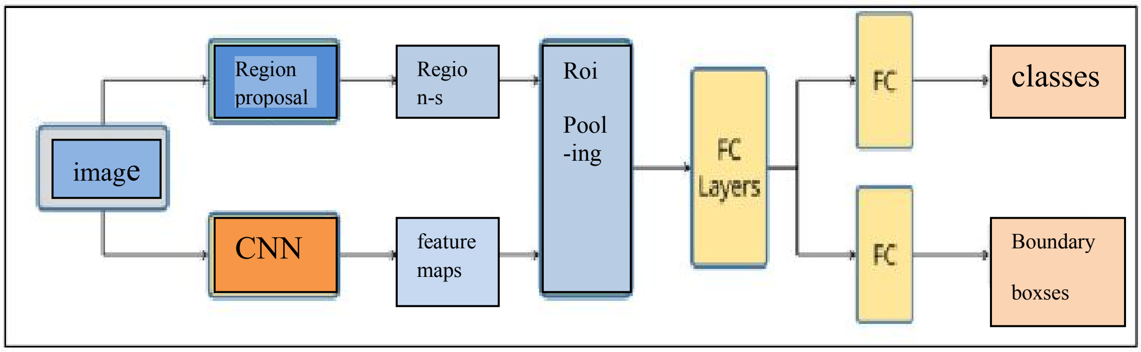


Figure 1.7: Fast R-CNN architecture [21]

1.5.4.3 Faster R-CNN

Even though the inference time was reduced from 47 s in R-CNN to 2.3 s in fast R-CNN, this last algorithm determines the regions and their bounding boxes using an algorithm of selective research, which itself leads to a considerable delay in the process. In 2015, Redmon et al . proposed the region proposal network **(RPN),** which is a neural network distinct used to predict bounding boxes. This network is merged with R-CNN, which shares convolution features. The new algorithm is named R-CNN faster. Faster R-CNN won first place in the ILSVRC competition.**[11]**

Faster R-CNN = Fast R-CNN + RPN **[2]**

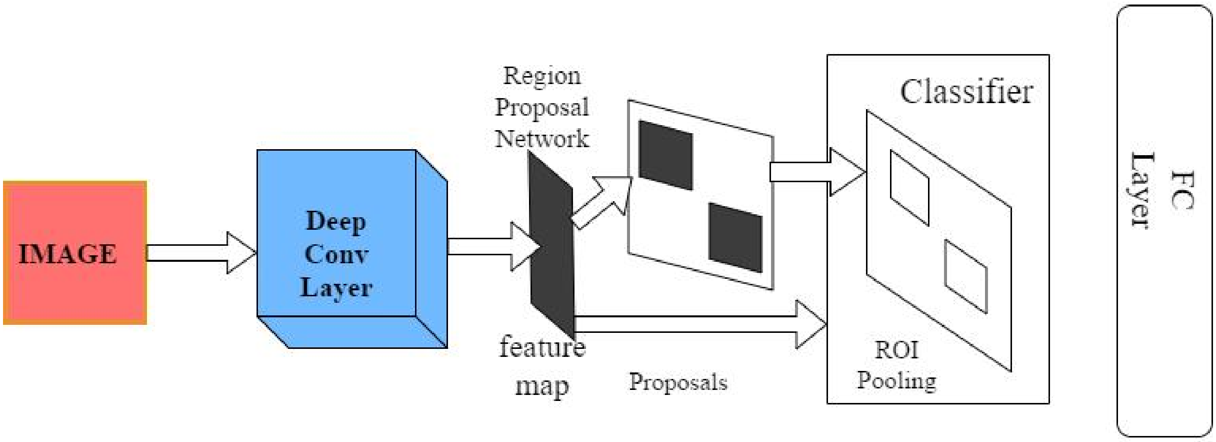


Figure 1.8: faster R-CNN architecture [11].

1.5.4.4 SSD

SSD is a real-time, efficient and high-precision detection algorithm proposed by Liu W and al. in 2016**[22]**,Single Shot MultiBox Detector (SSD) was the first single stage detector that matched accuracy of contemporary two stage detectors like Faster R-CNN , while maintaining real time speed. SSD detects smaller objects earlier in the network when the image features are not too crude, while the deeper layers were responsible for offset of the default boxes and aspect ratio.

During training, SSD match each ground truth box with the default boxes with the best jaccard overlap and train the network accordingly, similar to Multibox . They also used hard negative mining and heavy data augmentation. Similar to DPM, it utilized weighted sum of the localization and confidence loss to train the model. Final output is obtained by performing non maximum suppression. Even though SSD was significantly faster and more accurate than both state-of-art networks like YOLO and Faster R-CNN, it had difficulty in detecting small objects. This issue was later solved by using better backbone architectures like ResNet and other small fixes. **[23]**

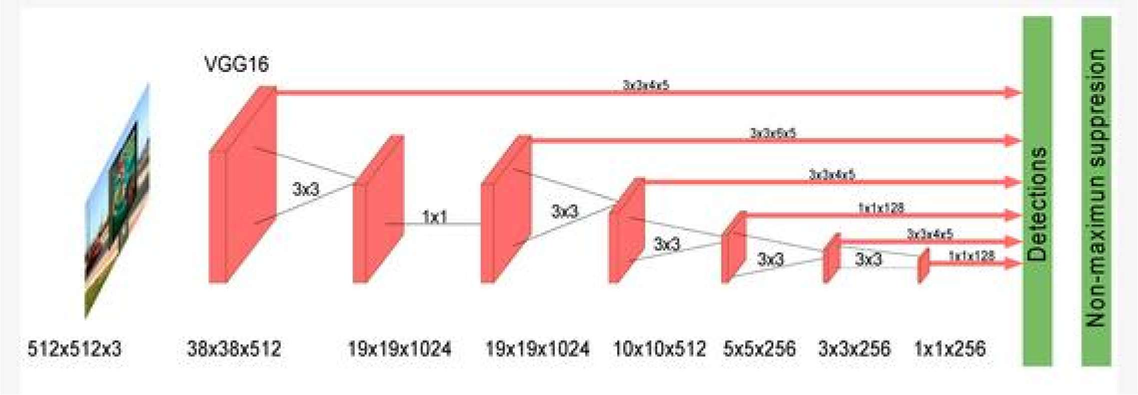


Figure 1.9: Network structure of SSD[22]

1.5.4.5 Yolo (YOU ONLY LOOK ONCE)

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev **[25],** he's a family of neural networks that improved the speed and accuracy of object detection with deep learning.

The main improvement in YOLO is the integration of the entire object detection and classification process in a single network. Instead of extracting features and regions separately, YOLO performs everything in a single pass through a single network, hence the name "You Only Look Once." **[26].** The object might have different spatial locations and aspect ratios within the image. Hence, the real-time algorithms of object detection using the CNN model such as R-CNN and YOLO had further developed to detect multi-classes in a different region in images had been developed. The illustration for the YOLO model is shown in **Figure (1. 10)**

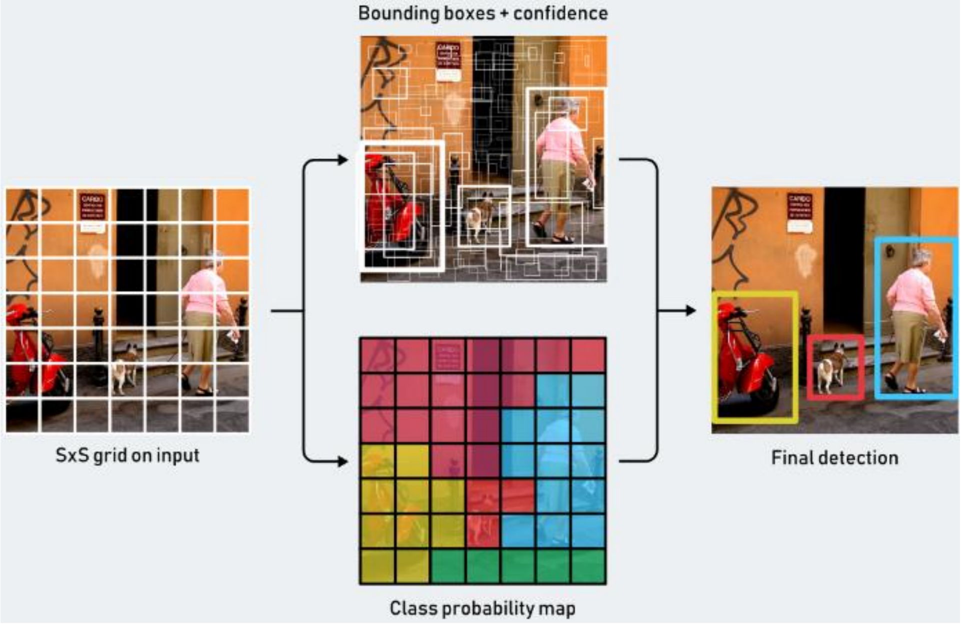


Figure 1.10: A simplified illustration of the Yolo object detector pipeline [26].

We use a totally different approach. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

YOLO is a great example of a single-stage detector. We'll be using YOLOv3 in this project, in particular, YOLOv3 is significantly larger than previous models, YOLOv3 had "incremental improvements" from the previous YOLO versions [10], but is, in my opinion, the best one yet out of the YOLO family of object detectors. [13]YOLO trained on the COCO dataset. The COCO dataset consists of 80 objects, including people, bicycles, cars and trucks, Airplanes, stop signs and fire hydrants, animals, including cats, dogs, birds, horses, cows, and sheep, to name a few, kitchen and dining objects, such as wine glasses, cups, forks, knives, spoons, etc.

1.5.5 Convolution neural network (CNN)

Convolutional Neural Networks (CNNs) are Artificial Intelligence algorithms based on multi-layer neural networks that learn relevant features from images, being capable of performing several tasks like object classification, detection, and segmentation. The advantage of CNN over others classification algorithms (SVM, K-NN, Random-Forest, and others) is that the CNNs learn the best features to represent the objects in the images and have a high generalization capacity, being able to precisely classify new examples with just a few examples in the training set **[28]**

1.6 Object detection based on Deep Learning

At present, with the development of deep learning technology and image recognition technology, object detection algorithms based on deep learning have become the mainstream direction in the field of image recognition; and deep learning detection algorithms such as the Fast R-CNN, the R-FCN, the Faster R-CNN, SSD, and others have endlessly emerged.[24]

Object detection represents a significant focus of research in the field of computer vision that can be applied in driverless cars, robotics, video surveillance, and pedestrian detection. Recently, deep learning techniques have revolutionized the object detection field by improving object detection accuracy and robustness, Because deep neural networks can automatically learn different features, object detection based on deep learning is characterized by more abundant features and stronger feature representation capabilities than are possible with traditional hand-crafted features**[29]**(such as SIFT, Scale Invariant Feature Transform, HOG, Histogram of Oriented Gradients and Haar features) **[30].**

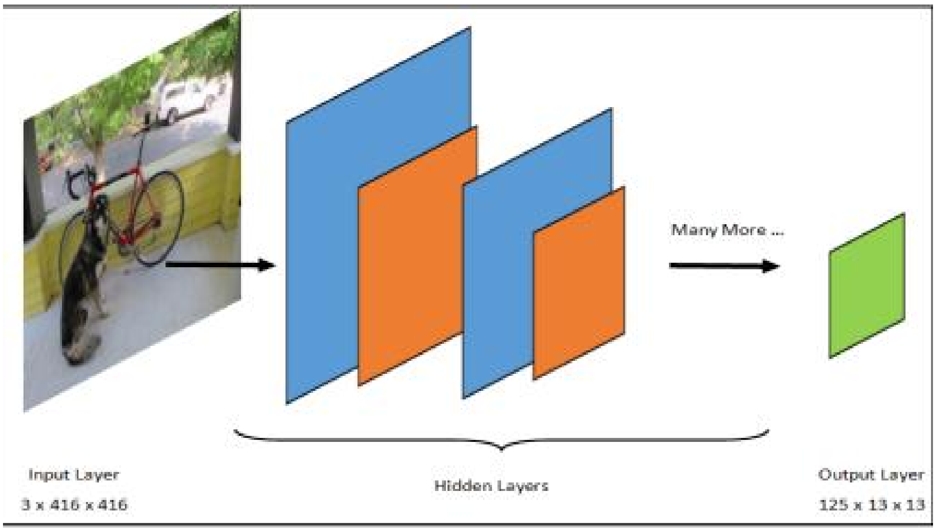


Figure 1.11: Example of object detection based on DL [11].

1.7 Conclusion

Nowadays, the world is getting smarter, and image-processing technique is provided all over the world, such as camera surveillance, object tracking, object detection, classification...etc. However, object detection is the most crucial and special part of image processing. In this chapter, we have presented the meaning of social distancing, then an overview of object detection and its principle. Then, we presented a state of the art on object detection aiming to show the diversity of approaches proposed in this field. These stages are divided into two categories: stage based on automatic learning (Machine Learning) and stage based on deep learning (Deep Learning). Finally, we discussed the principle and the different stages of the most common detection method such as; CNN.R-CNN, FAST R-CNN, FASTER, SSD and YOLO.

Chapter 2

social distancing detection using YOLOv3

2.1 Introduction

In this chapter, we present the main steps to build a social distancing detection. At first, we describe the importance of the Yolov3 method to detect objects and how it works; we review how we detect the distance in the second part. Specifically, we focus on the centroid algorithm to track the person and take the center point of the bounding box by assign identifies .And with these center points we calculate the pairwise distance on the video stream. At the end, we check if the distance is in safe or unsafe condition between people.

2.2 Object Detection using YOLO

Object detection algorithms based on deep learning have become the mainstream direction in the field of image recognition; and deep learning detection algorithms such as Yolo have endlessly emerged, in this section we are going to see how the Yolo algorithm works and show the version of it.

2.2.1 Definition of Yolo

YOLO stands for "You Only Look Once" which is one of the fastest object detection algorithms that use a variant of **darknet** originally trained on **Imagenet** consisting of 53 layers. It was developed by Joseph Redmon for the task of detection and computer vision**.[31]**

- **Darknet53** is the basic feature extraction network of target detection network YOLOv3,which is used to extract the characteristics of training images.**[32]**
- **Imagenet** is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web .**[33]**

The YOLO network consists of three main pieces **[34]:**

1) **Backbone**: A convolutional neural network that aggregates and forms image features at different granularities.

2) **Neck**: A series of layers to mix and combine image features to pass them forward to prediction.

3) **Head**: Consumes features from the neck and takes box and class prediction steps

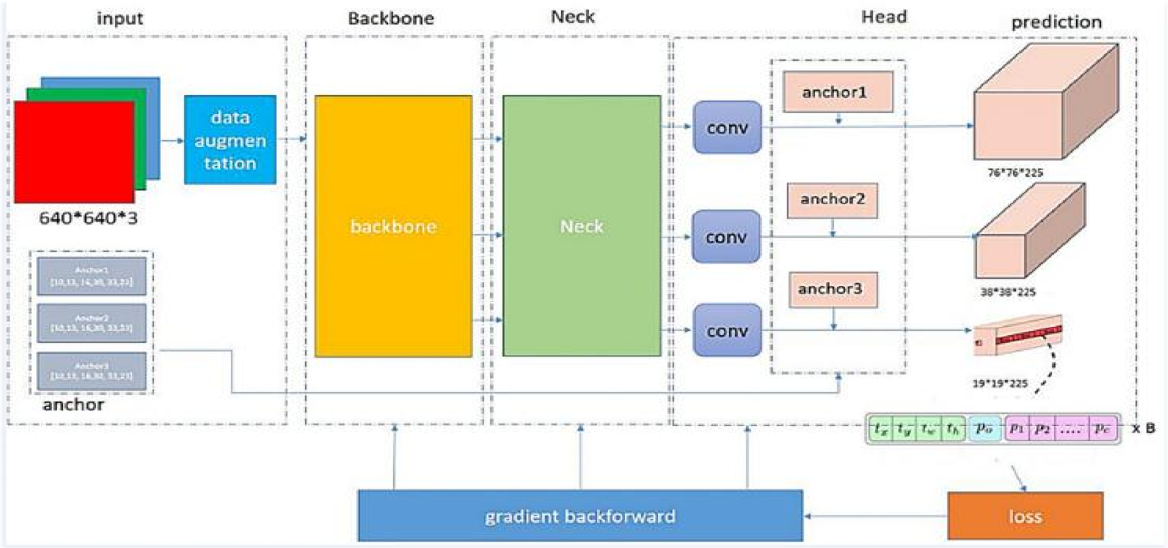


Figure2.1: Network structure of the Yolo algorithm [35].

Yolo is a state-of-art; Real-time detection system that predicts and classifies bounding boxes for objects within a given image simultaneously. the image data input To Yolo is divided into SxS grid cells according to resolution, features are extracted through CNN, and the predicted tensore is obtained through a fully connected layer, as shown in the figure(2,2). The predicted tensor has a size of (S x S) and a length of (B x 5+c).Here, SxS is the number of grid zones, B is the number of candidate boundary boxes whose center point is included in the grid zone, and C is the number of objects that can be classified . Each lattice zone is represented by a vector having a length of (Bx5+C), and the set of SxS lattice zones constitutes a predicted tensor of S x S x (B x 5 + C). **[34]**

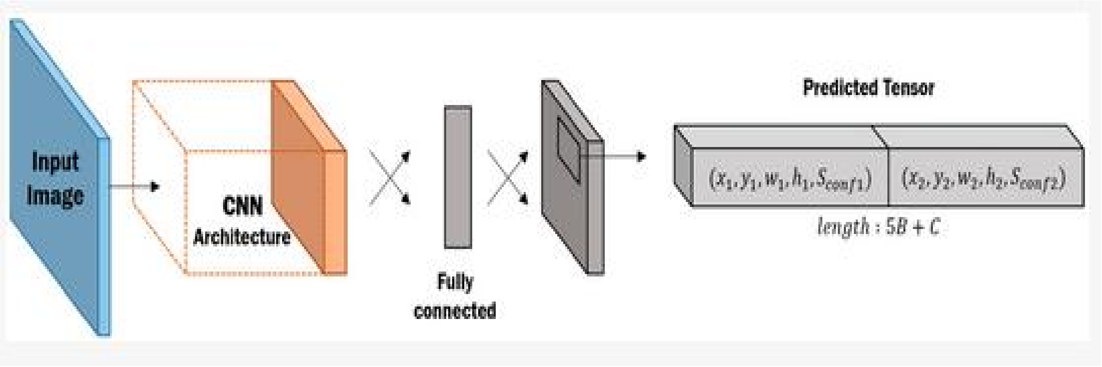


Figure2.2: structure of Yolo [34].

2.2.2. Versions of Yolo

There are five versions of YOLO: v1, v2, v3, v4, and v5. YOLO v1 is inspired by **GoogleNet** (Inception network) which is designed for object classification in an image. This network consists of 24 convolutional layers and 2 fully connected layers. **[36]**

The Main improvement measures of YOLO network from V1 to V5 **[37]:**

- YOLOv1: The grid division is responsible for detection, and confidence loss.
- YOLOv2: Anchor with K-means added, two-stage training, and fully convolutional network.
- •YOLOv3:Multi-scaledetection by using FPN.
- YOLOv4: SPP, MISH activation function, data enhancement Mosaic/Mix-up, GIOU (Generalized Intersection over Union) loss function.
- YOLOv5: Flexible control of model size, application of Hardswish activation function, and data enhancement.

2.2.2.1. YOLOv1

Has two full-connected layers and twenty-four convolutional layers. The model size of YOLOv1 has reached 1 GB, which occupies a very large storage space and requires the running platform with high performance.

2.2.2.2. YOLOv2

Removes the fully-connected layers and introduces anchor boxes to predict bounding boxes, making the YOLO detector faster and more robust than YOLOv1**. [38]**

2.2.2.3. YOLOv3

YOLOv3 performs multi-label classification with the help of logistic classifiers instead of using softmax as in the case of YOLOv1 and v2. In YOLOv3 Redmon et al.proposed Darknet-53 as a backbone architecture that extracts features maps for classification. **[36]**

2.2.2.4. YOLOv4

Compared with YOLOv3, YOLOv4 introduced mosaic data enhancement in the data processing. In addition, the backbone, net-work training, activation function, and loss function were optimized, which made YOLOv4 faster and achieved the best balance between the accuracy and speed in these real-time object detection algorithms**. [39]**

2.2.2.5. YOLOv5

Despite being released a month after YOLO v4, the start of research for YOLOv4 and YOLO v5 was quite close (March – April 2020). For avoiding collision, Glenn decided to name his version of YOLO, YOLOv5. The architecture of YOLOv4 and YOLOv5 are very similar **[40]**

2.2.3. YOLOv3 principle

In this work, we will use YOLOv3. YOLOv3 is an improved version of YOLOv2. It uses multi-scale prediction to detect the final target, and its network structure is more complex than YOLOv2. YOLOv3 predicts bounding boxes on different scales, and multi-scale prediction makes YOLOv3 more effective for detecting small targets than YOLOv2**. [41]**

The YOLOv3 architecture has residual skip connections and an up sampling layer **[23]**. The key novelty is this algorithm is that it makes its detections at three different scales. The Yolo algorithm is a fully connected layer and the detection is done by using a 1*1 kernel on the feature maps to make the detections at three different locations using three different scales, as can we seen in the Architecture below (figure 2.3). As mentioned, the shape of the kernel for detection is a 1*1*(B*(A+C)), where B indicates the number of bounding boxes, C refers to the number of classes and the A refers to the 4 bounding box attributes (height, width, x and y coordinates). The Yolov3 algorithm was trained on a dataset known as the coco dataset, which has 80 classes and the bounding box attributes summing to 3, so in effect, the kernel size becomes 1*1*255.

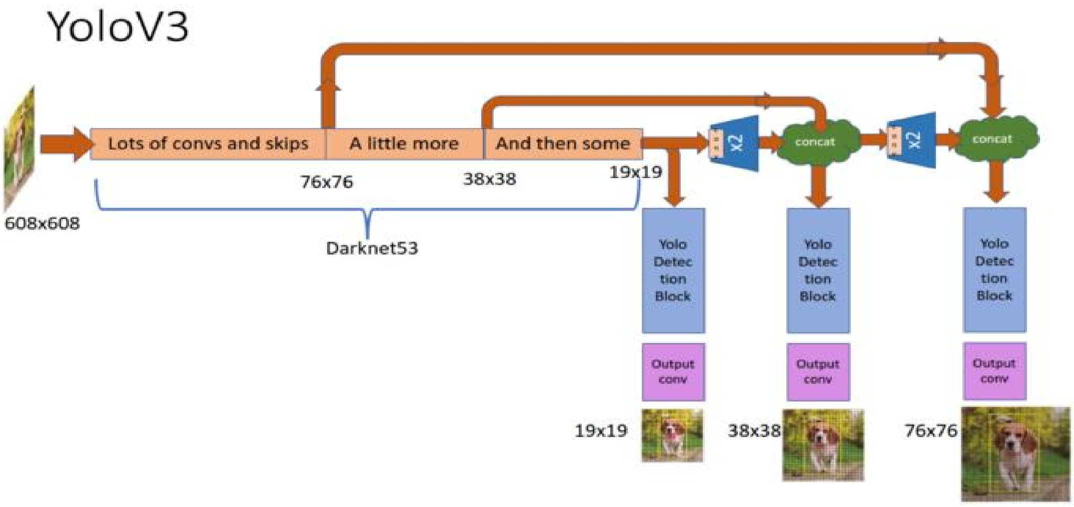


Figure2.3: Architecture of Yolov3 [42].

2.2.4. YOLO algorithm steps

The main steps of the YOLO algorithm are **[43]**:

- The input image is divided into a grid of boxes eg. 13 × 13. This way image got divided into a set of smaller images and it increases the possibilities of object detection in each cell.
- Perform the image classification and localization on each grid cell. This step produces a probability of an object detected, dimensions of the bounding box and class of the detected object.
- Perform thresh holding to remove the multiple instances detected. It only picks those cells which are having highest probabilities.
- Perform the non-max suppression to further refine the detected bounding boxes using the technique of Intersection of Union.

2.2.5. How YOLOv3 Works

How YOLO works: is that we take an image and split it into an **SxS** grid, within each of the grids we take b bounding boxes. For each of the bounding boxes, the network outputs a class, probability and offset values for the bounding box. The bounding boxes having the class probability above a threshold value are selected and used to locate the object within the image. [**44] .**In order to implement object detection, each grid cell predicts B bounding boxes with their parameters and confidence scores for those boxes.

This confidence score reflects the presence or absence of an object in the bounding box,

The confidence score is defined as **[40]:**

$$\text{confidence score } = \text{p(Object)} * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Evaluating the YOLO model on the famous COCO dataset which contains 80 classes and sets each cell to predict 2 bounding boxes, the total output parameters are 7 × 7 × (5 ∗ 2 + 80).

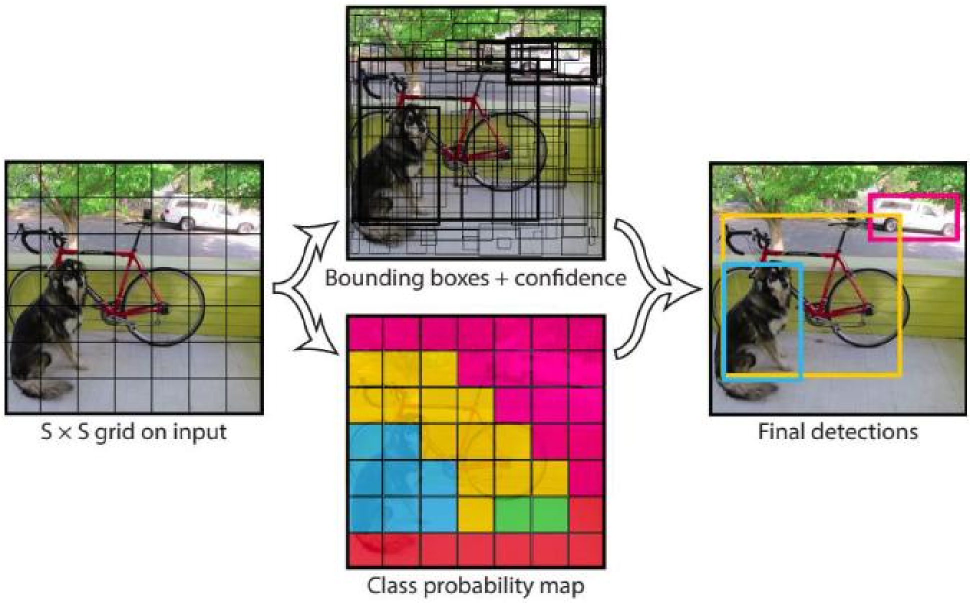


Figure2.4:Yolo model with 7×7 grid cell applied on input image [40].

2.2.5.1. Bounding box

YOLOv3 predicts boxes at different scales using the concept of FPN. It mainly utilizes a certain amount of convolutional layers and residual layers to complete the detection process, and uses the features of the entire image to predict each bounding box. [45]

After dividing the picture into a certain number of squares, each square checks the picture if it contains an object or not.

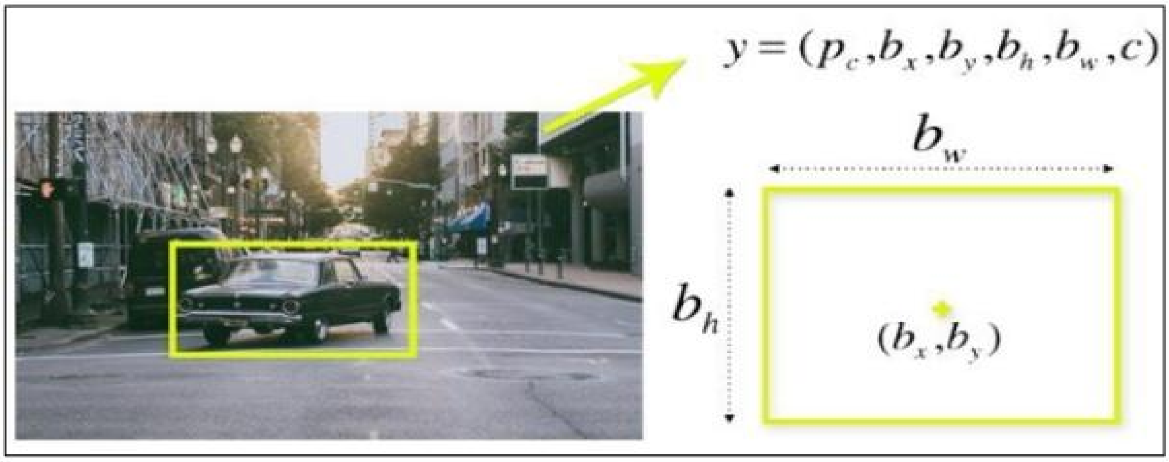


Figure2.5: An Example of bounding box [46].

The grid cell gives the bounding squares predictions in addition to the conditional class c.

The grid indicates 4 coordinates for each bounding square, bx, by, bw, bh **[47]:**

$$bx = \sigma(tx) + cx$$
$$by = \sigma(ty) + cy$$

$$bw = pwe^{tw}$$
$$bh = ph\, e^{th}$$

bx, by: Determine the center of the square.

bh : height.

bw : width .

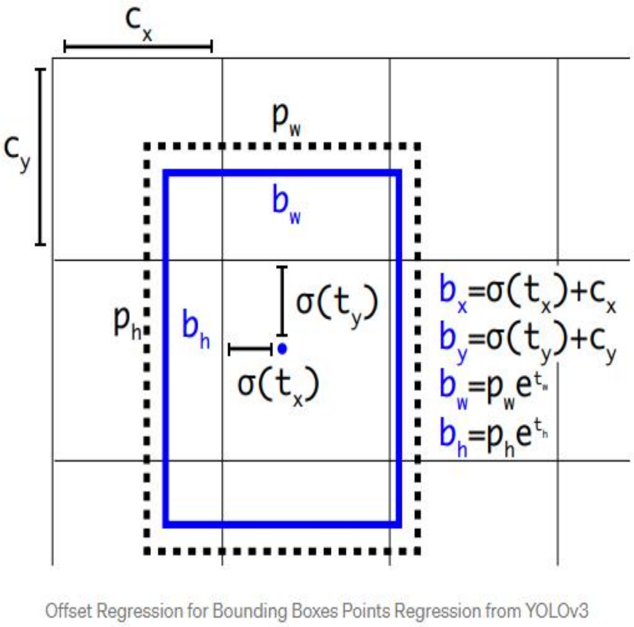


Figure2.6: Bonding box with tension priors and location prediction [47].

Each square in figure (2,7) represents an array with 8 values : pc , by,bx,bw,bh,c1,c2,c3 .

PC represents the class probability, if the presence of an object is registered, then the value "pc = 1"

Let's take the example of the car in the figure below:

- We mentioned before that the value "pc = 1" represents the presence of an object in the image, while "pc = 0" means there is no object. In this example, we find the value c2 = 1, because the object represents a car

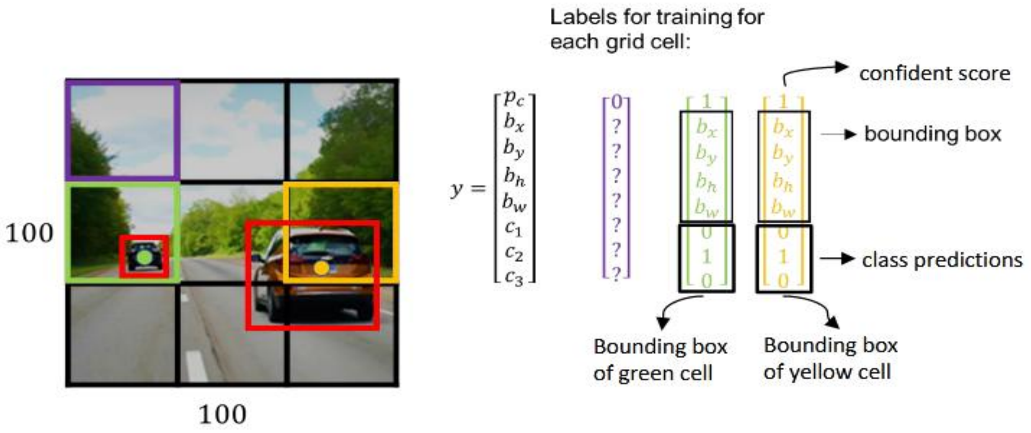


Figure2.7: specifying label vector y in a Yolo model has 3×3 grid cells and predict object for 3 classes [48].

He the picture is divided by grid 3×3 (9 squares), each square represents a matrix and the depth of the matrix is the number of b and c values, the result of checking the square is at the top left of the square.

As shown in figure (2,8), the picture contains a lot of objects, the number of squares must be doubled, because the greater the division, the greater the accuracy of determining the object

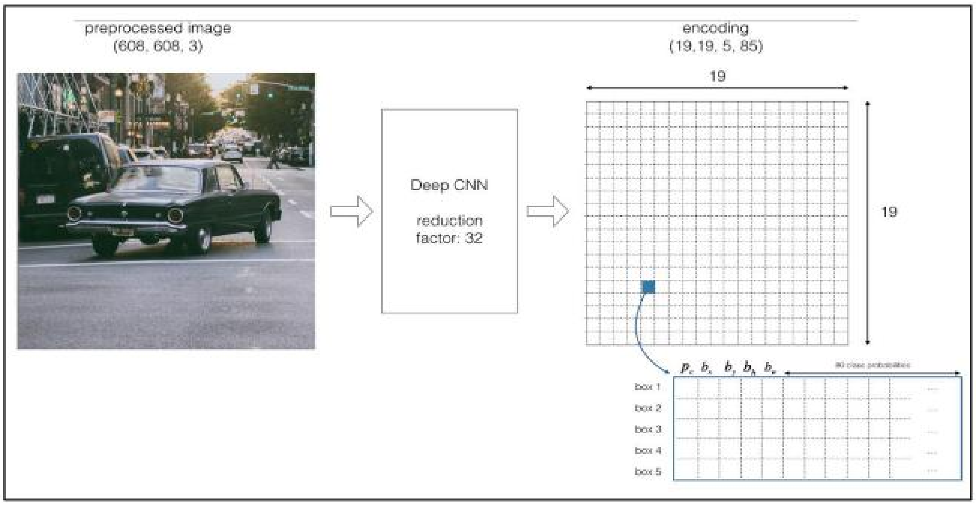


Figure2.8: the process of englobate box division

2.2.5.2. Non-max-suppression:

A technique to filter the predictions of object detectors.NMS is integrated into OD algorithms to filter detection boxes.NMS makes selections based on the Intersection over Union (IOU) among detection boxes. IOU measures the ratio of the overlapped area over the union area between two boxes. The NMS works in two steps **[49**]:

For a given object category, all of the detected bounding boxes in this category (i.e., candidate set) are sorted based on their box confidence scores from high to low.

NMS selects the box which has the highest confidence score as the detection result, and then it discards other candidate boxes whose IOU value with the selected box is beyond the threshold. Then, within the remaining boxes, NMS repeats the above two steps until there is no remaining box in the candidate set.

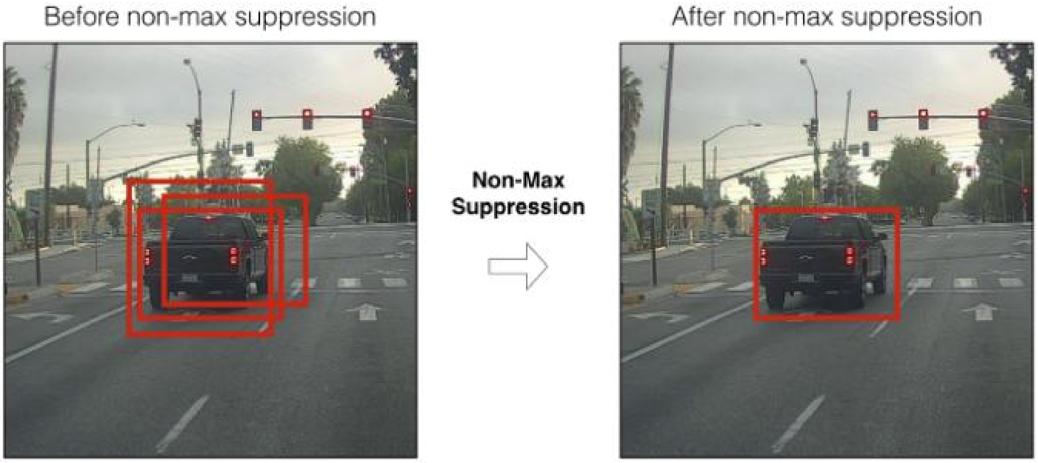


Figure2.9:Non-Max suppression on Algorithme [50].

This means that we will not put several bounding boxes on the same object. As we see in the (figure 2.9), we have one car in the picture surrounded by several boxes, in this case, a defect will occur in the evaluation, so one must choose a suitable bounding box for the car. The appropriate bounding square for the object is chosen according to accuracy, by comparing squares according to **IOU.**

2.3. Social distancing detector steps

This section discusses the essential steps to prepare a workflow for social distancing on images as seen in Fig.(2.10) **[51]**:

- Prepare an input images or streaming a video from a camera which contains people.
- Applying the deep learning object detector to detect people in images or video streams.
- Check the number of persons that are in the images or video stream.

- Compute the distance between the centroid of the bounding boxes which are enclosed to the detected people.

Finally, the algorithm will decide for safe or unsafe social distancing based on the number of persons and the measured distance between the centroid of bounding boxes.

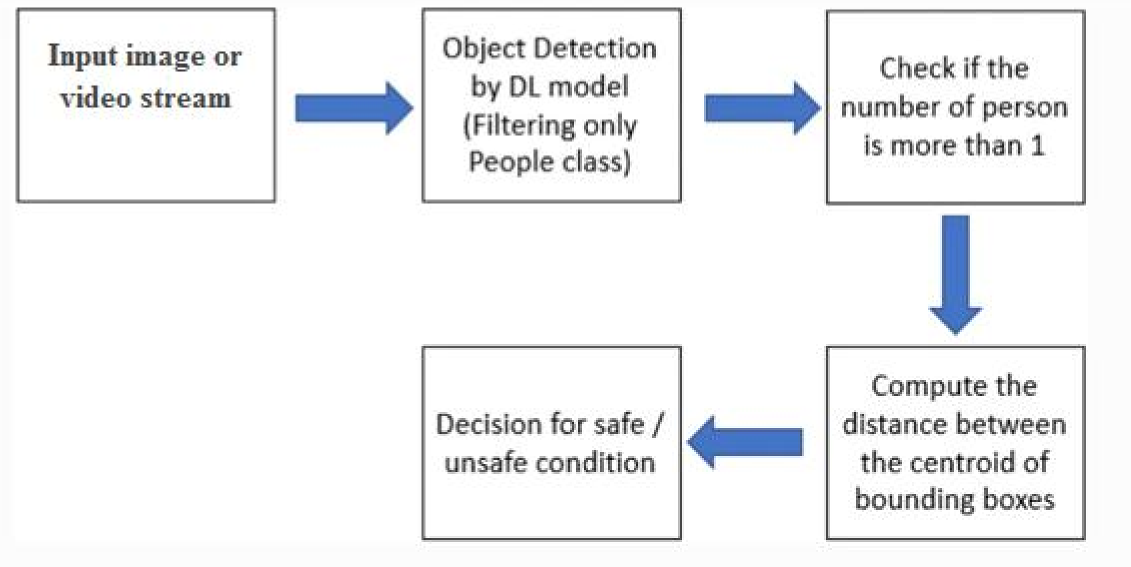

Input image or video stream
Object Detection by DL model (Filtering only People class)
Check if the number of person is more than 1
Compute the distance between the centroid of bounding boxes
Decision for safe / unsafe condition

Figure2.10: The steps for people detection and social distancing classification on image [51].

There are two main components to the program: the setup, which only occurs once in the beginning, and the operation, which is a loop that occurs once for each frame of the input video.

2.3.1. The setup

The first step of the setup loop is to input an image consisting of objects, then we detect a person, which is accomplished using an object detection program called YOLO. This program recognizes a wide variety of objects, but this work includes a filter that only keeps the person known.

2.3.1.1. People detection

This part presents how we used a deep learning model along with some knowledge in computer vision to build a social distancing detector.

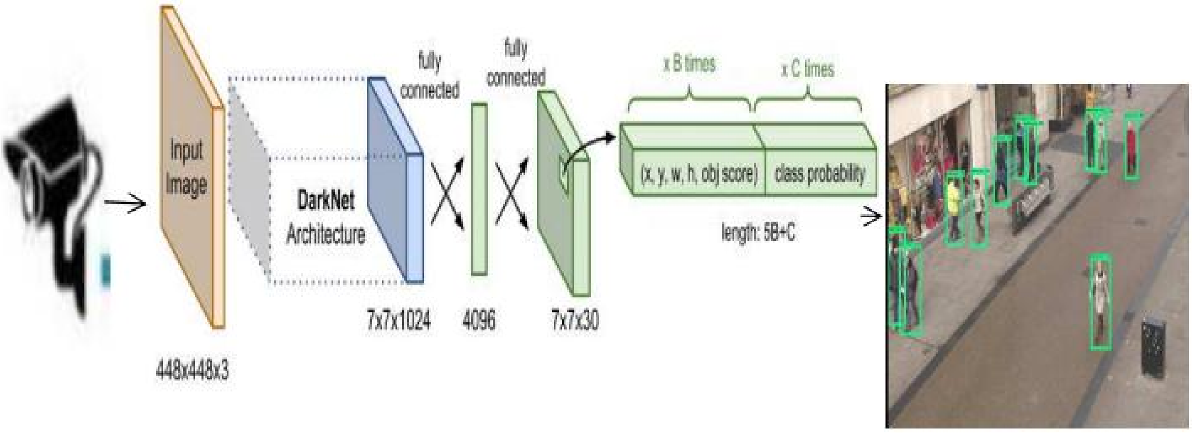


Figure2.11: Stage-1 the overall structure of the people detection module. [30]

In order to detect persons, we need to define some steps. **[52]**

- **Pass the frames to Yolo algorithm**

For every frame that needs processing. Some parameters have to be specified when doing so **[32].** These include the type of input that the model requires and which outputs we want to get back from it. In our case the outputs needed are the following:

- Bounding boxes coordinates of each object
- The confidence of each prediction (0 to 1)
- Class of the prediction (0 to 90)

The YOLO framework deals with object detection in a different way, It takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes, There are 2 main conventions followed when representing bounding boxes **[53]:**

- Specifying the box with respect to the coordinates of its top left, and the bottom right point.
- Specifying the box with respect to its center, and its width and height**.**

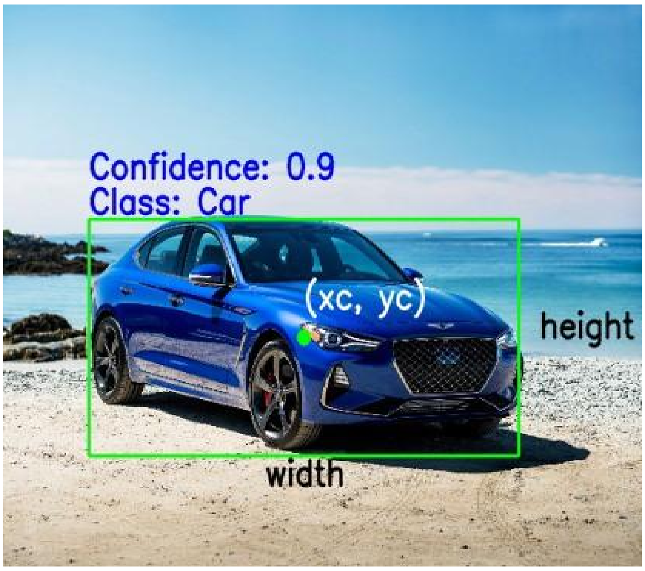


Figure2.12: Object Detection with coordinates [53].

As we see in **figure** (2, 12) the bonding box specifying also with:

1. Class: What is in the object inside the box. Eg car, truck, person etc

2. Confidence: Indicates how likely the object is actually present in that box. Eg a confidence of 0.9 would indicate that there is a 90% chance that the object actually exists in that box.

- **Filter out weak predictions and non-relevant objects**

The objective is to exclude both weak predictions and all other classes of objects except person , and The class associated to a person is 1. **[52]**

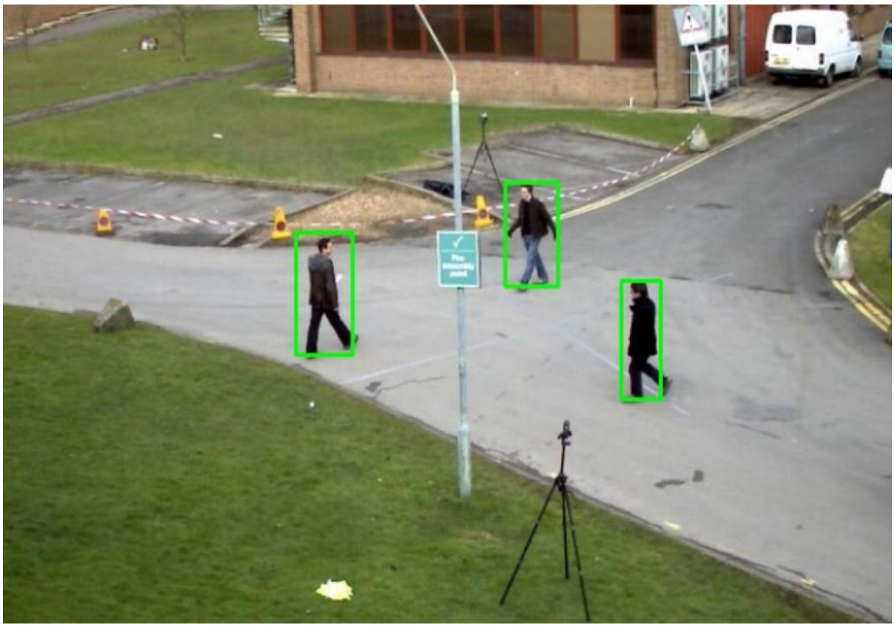


Figure2.13: Filtering only the people detection [52].

2.3.1.2. Person Tracking

The idea of object tracking is to analyze video frames and find the location of moving objects in every single video frame. Object tracking techniques use algorithms like deep sort, centroid tracker, csrt, kcf..Etc, if the object has the same metric throughout the frame then it will track the same object throughout the sequence of frames and keep the same object ID for that object **[52].**Object tracking is the operation of **[54]**:

- Taking an initial set of object detections (such as an input set of bounding box coordinates)
- Creating a unique ID for each of the initial detections
- And then tracking each of the objects as they move around frames in a video, maintaining the assignment of unique IDS.

2.3.1.2.1. Centroid tracking algorithm

To track our object we use Centroid Tracking algorithm. This algorithm works by calculating the Euclidean distance between the centroid of identified people over frames **[55]**.there are five steps wich are:

- **Step 1: compute the centroid of bonding box**

To build a simple object tracking algorithm using centroid tracking, the first step is to accept bounding box coordinates from an object detector and use them to compute centroid. These bounding boxes can be produced by any type of object detector you would like (color thresholding + contour extraction, Haar cascades, HOG + Linear SVM, SSDs, Faster R-CNNs, Yolo ….). In this work, we use YOLOv3 as an object detector.

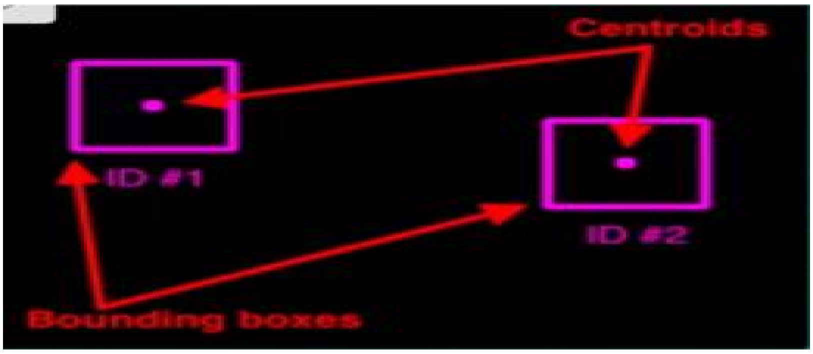


Figure 2.14: Shows that it set box coordinates and computes the centroid [54]

- ***Calculate the center point of a bounding box***

To measure the center point C(x, y); of the bounding box for the detected person, midpoint equation is used as in (2.1)**[56] .**

$$\mathbf{C(x,y)= (\frac{Xmin+Xmax}{2}, \frac{Ymin+Ymax}{2})} \qquad \mathbf{(2.1)}$$

Width= {Xmin and Xmax}.

Height= { Ymin and Ymax} .

- **Step 2: Compute Euclidean distance centroid**

For every frame in our video stream we apply **Step 1**of computing object centroids**[57]**, however, instead of assigning a new unique ID to each detected object , we first need to determine if we can, associate the new object centroids (yellow) with the old object centroids (purple).

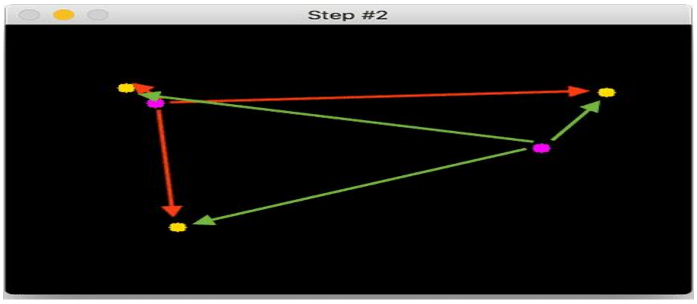


Figure2.15: Three objects are present in this image [54].

We compute the Euclidean distances between each pair of original centroids (purple) and new centroids (yellow). But how do we use the Euclidean distances between these points to actually match them and associate them?

- ***Calculate the Euclidean distance:***

To measure the Euclidean distance, C1 (X1, Y1) and C2 (X2, Y2) between two pedestrian, distance equation is used as in (2.2) **[56]**.

$$D(C1, C2) = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2} \qquad \mathbf{(2.2)}$$

- **Step 3: Update(x, y)-coordinates of existing objects**

We assume that the same object will be moved the minimum distance compared to other centroid, which means the two pairs of centroid having minimum distance in following frames are considered to be the same object .that means we tacking the same person.

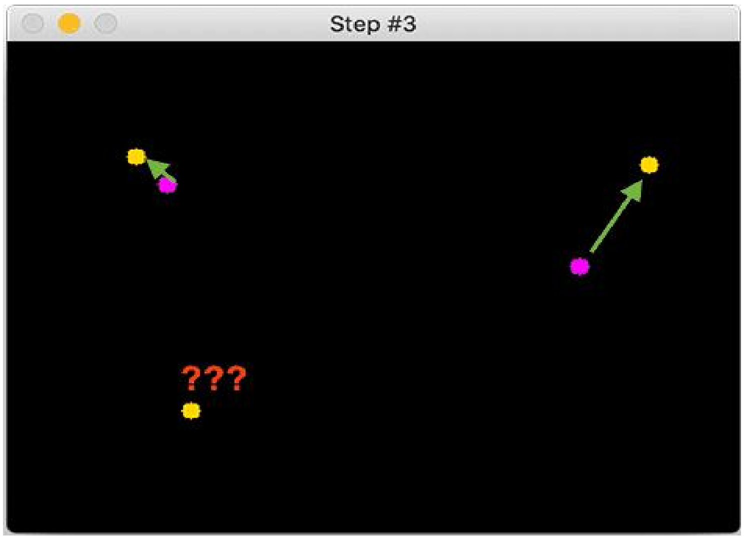


Figure2.16: Now it's time to assign the IDs to the moved centroid that will indicate the same object [54].

But what about the lonely point in the bottom-left?

- **Step 4: Register new objects**

In the event that there are more input detections than existing objects being tracked, we need to register the new object. “Registering” simply means that we are adding the new object to our list of tracked objects by **[57]**:

- Assigning it a new object ID
- Storing the centroid of the bounding box coordinates for that object

We can then go back to **Step 2**and repeat steps for every frame in our video stream.

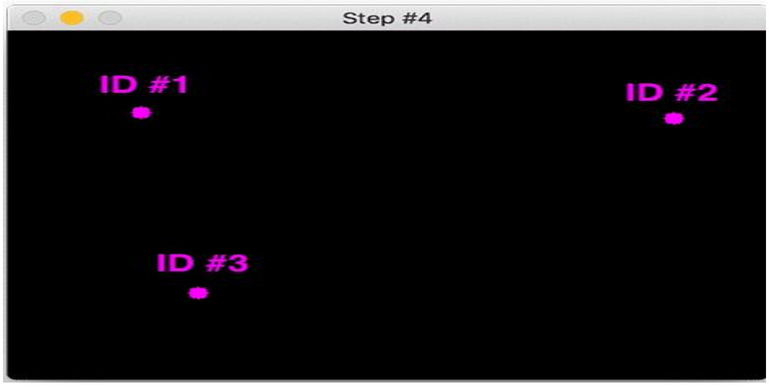


Figure2.17: We have a new object that wasn't matched with an existing object, so it is registered as object ID*3 [54].

➢ Step 5: Deregister old objects

Any reasonable object tracking algorithm needs to be able to handle when an object has been lost, disappeared, or left the field of view **[57]**. For this implementation, we will deregister old objects when they cannot be matched to any existing objects.

2.3.1.2.2. Disadvantages of object tracking algorithm

While our centroid tracker worked great in this example, there are two primary drawbacks of this object tracking algorithm:

- The first is that it requires that object detection step to be run on every frame of the input video.

- If two or more objects overlap each other to the point where their centroids intersect , the algorithm may (unknowingly) swap the object ID.**[57]** It’s important to understand that the overlapping object problem is not specific to centroid tracking it happens for many other object trackers as well, including advanced ones.

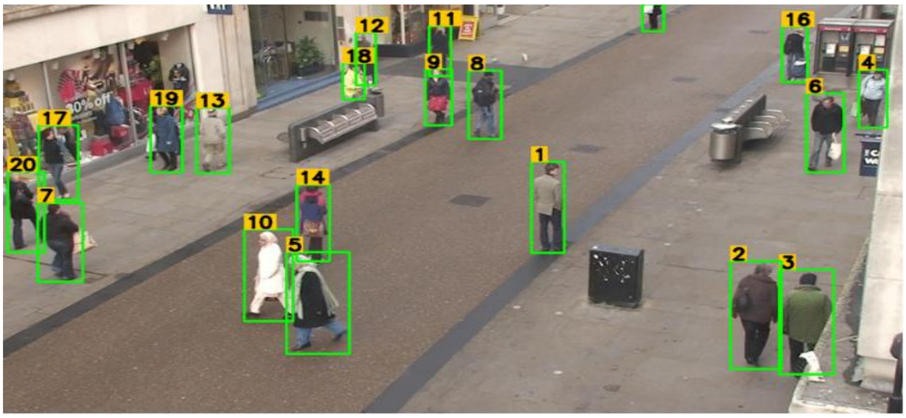


Figure2. 18: Simple online and real time tracking technique [60]

2.3.2. The operation

The first step from the operation is to see the scene by camera calibration to get better results and enable you to compute actual measurable units (rather than pixels).

2.3.2.1. Bird eye view transformation

Camera view calibration is applied which works by computing the transformation of the perspective view into a top-down view, which means he gives us a top view of the scene thankfully to OpenCV cause has great built-in functions to apply this method **[58]**, The actual distance between pedestrians corresponds to the number of pixels in the top-down view can be estimated..**[59]**

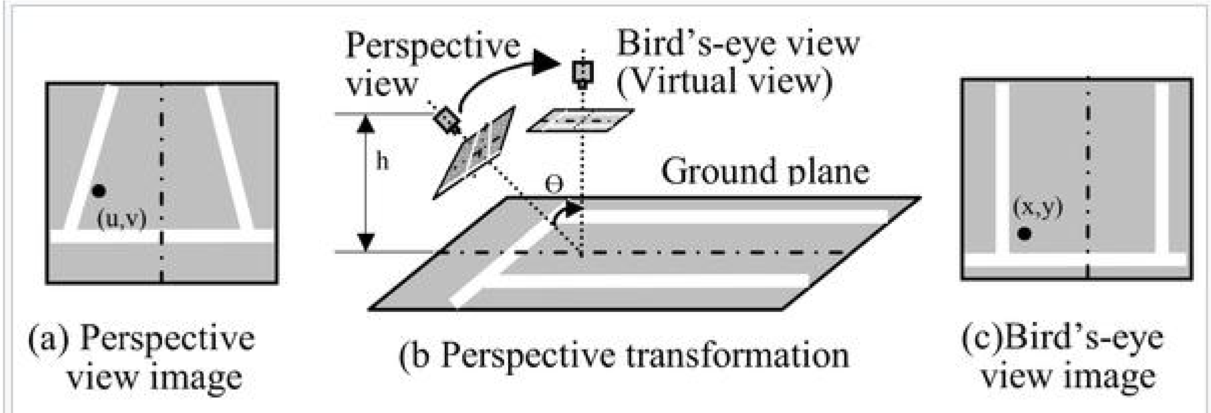

Perspective view
Bird's-eye view (Virtual view)
h
Θ
Ground plane
(u,v)
(x,y)
(a) Perspective view image
(b Perspective transformation
(c)Bird's-eye view image

Figure2. 19: illustration of perspective transformation in a parking lot scene [60].

The first step involves selecting 4 points on the original image that are going to be the corner points of the plan which is going to be transformed. These points have to form a rectangle with at least 2 opposite sides being parallel. If this is not done, the proportions will not be the same when the transformation happens. For each person detected, the 2 points that are needed to build a bounding box a returned. The points are the top left corner of the box and the bottom right corner. From these, we computed the centroid of the box by getting the middle point between them. Using this result, we calculated the coordinates of the point located at the bottom center of the box. **[60]**

2.3.2.2. Distance measuring

To compute the pairwise distance we have 2 steps: calibration and testing by calculating the distancing between two pairs.

In calibration, there are 2 steps **[61]:** The user is required to input the focal length, then

The user is required to position these 2 individuals at the minimum social distance that is to be maintained.

1 / To calculate the distance between two persons first the distance of the person from the camera then we calculate the perceived focal length of the camera, we assumed person distance D from the camera and person's actual height H=165cms and with person detection pixel height P of the person is identified using the bounding box coordinates. Using these values, the focal length of the camera can be calculated using the formula below **[55]**:

$$F = (D \times P)/ H \qquad (2.3)$$

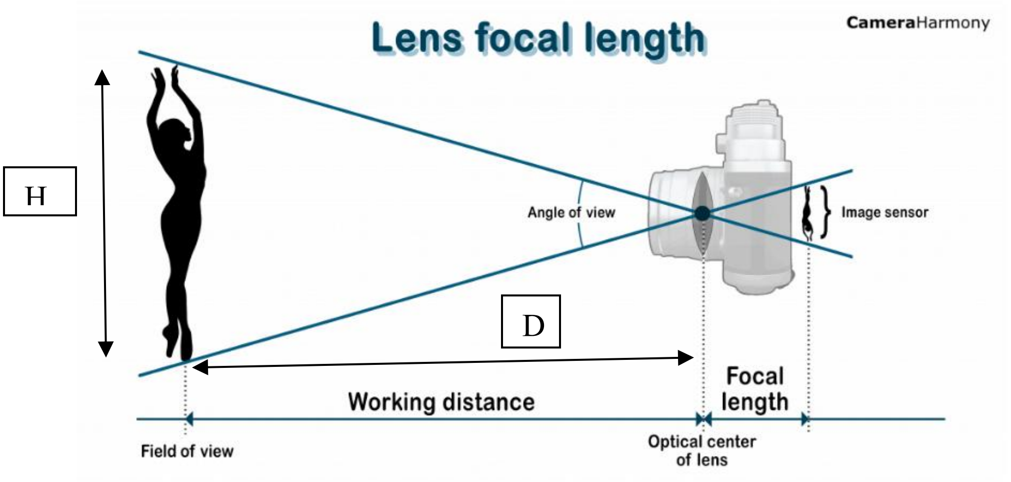


Figure2.20: focal length of camera

Then we use the real person's height H, the person's pixel height P, and the camera's focal length F to measure the person's distance from the camera. The distance from the camera can be determined using the following**[55]**:

$$\mathbf{D1= (H\times F)/ P} \qquad \mathbf{(2.4)}$$

In testing we need: to calculate the Euclidian distance between two pairs of pedestrian as we show in eq(**2.2.)**

2.3.2.3. Decision for safe /unsafe condition:

The social distance, which is first calculated in calibration mode, will be used as the reference social distance. In the testing mode, the social distance between 2 individuals will be calculated using the equations shown above. If the calculated social distance is lesser than the reference social distance, the pair of individuals will be identified as violations.

The pair of pedestrians, whose distance is below the minimum acceptable distance, t., is marked in red, and the rest is marked in green. A red line is also drawn between the pair of individuals whose distance is below the pre-defined threshold. The bounding box's color threshold operation, c, can be defined as**[59]** :

$$C = \begin{cases} \boldsymbol{red} & D < t \\ \boldsymbol{green} & D \geq t \end{cases} \qquad (2.5)$$

2.3.2.4. Results :

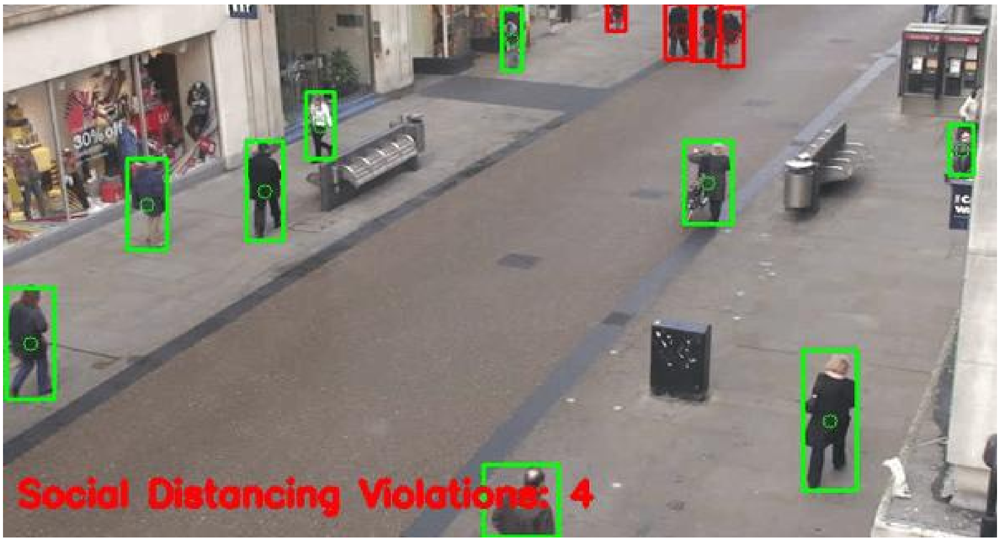


Figure2.21: Output image

Here we see 4 violations (4 bounding box in red) means that people in there are in close distance and the others in green box means they are keeping a safe distance .

2.4. Conclusion

in this chapter we presented the yolov3 as well as his function on our project, which is a fast algorithm for detecting people, we use the centroid algorithm for tracking people to compute the distance between the pedestrian . we compute the Euclidian distance and based on that we check how many people are following the social distance.

Chapter3

Implementing a social distancing detector with OpenCV and deep learning

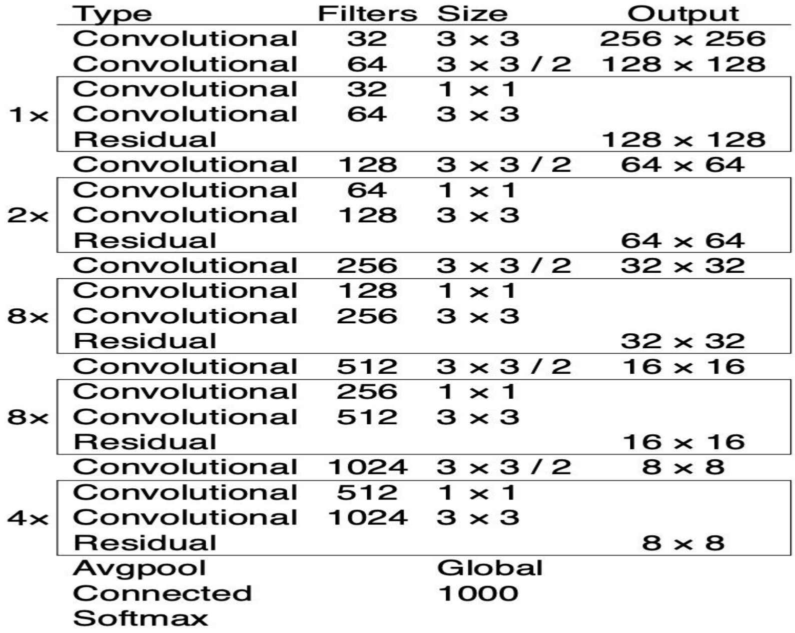
3.1 Introduction

The social distancing detector is usually applied to images or a video sequence. We start this chapter by choosing the network for our tool, then explaining how to load video sequences with Python 3.7 and OpenCV and get frames with DNN model we detect our object in our work we detect only people. We typically see about object detection a darknet network, we have seen how to use deep learning to detect objects in an image. For this, we used a YOLOv3 neural network, pre-trained to detect and identify objects in 80 categories (person, car, truck, potted plant, giraffe ...). YOLOv3 is the state-of-the-art object detection algorithm: It is very accurate and fast when evaluated on powerful GPUs, compared to other algorithms, basic on that we will create our structure to build a social distancing detection.

3.2 Darknet

Darknet-53 is a convolutional neural network that acts as a backbone for the YOLOv3object detection approach. Darknet is mainly for Object Detection, and has a different architecture, features than other deep learning frameworks. It is faster than many other NN (like CNN,RCNN, Fast RCNN,Faster RCNN,..). Architectures, We would say Darknet architecture & YOLO is a specialized framework. YOLO can run on CPU but you get 500 times more speed on GPU as it leverages CUDA.

- YOLOv3 uses a variant of Darknet, which originally has 53 layer network trained on Imagenet.
- For the task of detection, 53 more layers are stacked onto it, giving us a 106-layer fully convolutional underlying architecture for YOLO v3.
- YOLO model is compatible with OpenCV DNN module.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure3.1: Darknet architecture

3.2.1 Cuda

CUDA is a parallel computing architecture developed by NVIDIA, which makes it possible to use the resources of the graphics processor (GPU). CUDA is used in a large number of applications: machine learning, scientific computing, video games, etc.

3.3 Working environment

Now, we will see the (hardware and software) requirements that are used to achieve distance detection with deep learning which the goal of this project is.

3.3.1 Hardware

In order to carry out this work project, a set PC TOSHIBA with the following features:

- Processor: Intel(R) Core(TM) i5-2410M CPU @ 2.30 GHz.
- RAM: 4, 0 Go.
- Laptop’s in-built camera to build a real time social distance detector .
- System type: 64-bit exploitation System.
- Windows edition: Windows 7 Pro.

3.3.2 Software

For this project, we used a set of development tools such as: The programming language Python 3.7, the computer vision library OpenCV and PyCharm IDE.

3.3.2.1 *Python*

Python is the most used programming language in the field of Machine Learning, Big Data, and Data Science. Python is a general-purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability.

3.3.2.1.1 *Why we choose python*

This programming language has many interesting features such as[2]:

- It is cross-platform (portable). That is to say, it works on many operating systems: Windows, Mac OS X, Linux, Android and iOS, from Raspberry Pi mini-computers to supercomputers.
- It is free. You can install it on as many computers as you want (even on your phone!).
- It is a high-level language. Just you need a little knowledge of how a computer works to be used.
- It is an interpreted language. A Python script does not need to be compiled to be executed, unlike languages like C or C++. It is relatively easy to handle.
- All these characteristics mean that Python is now taught in many training courses, from secondary education to higher education.

Figure3.2: Logo of python

3.3.2.1.2 *Python Users:*

Many corporations have used and been using this tool for different functions. Some of them are:

- Google is one of the Python users that included this language in its web search system and employed Python's creator, too.
- YouTube video sharing service makes extensive use of Python.
- NASA, Los Alamos, Fermilab, JPL, and others use Python for scientific
- Programming tasks.
- IRobot uses Python to develop commercial robotic vacuum cleaners.
- Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm, and IBM use Python for hardware testing.

3.3.2.2 PyCharm

PyCharm is the most popular IDE (Integrated Development Environment) for Python scripting language, and includes great features such as excellent code completion and inspection with an advanced debugger and support for web programming and various frameworks such as Django and Flask. All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

3.3.3 The installation of software and the packages

3.3.3.1 OpenCV

OpenCV-Python (for Open Computer Vision) is a free library of Python bindings designed to solve computer vision problems. Developing by Intel, specializing in real-time image processing, to install OpenCV via PIP give the following command:

Pip install OpenCV-python

To Test OpenCV Installation use the following command in python**: import cv2**.

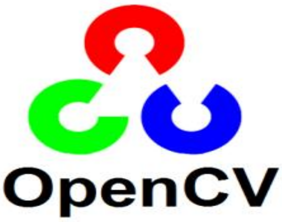


Figure 3.3: Logo of OpenCV

3.3.3.2 NumPy

Is an extension of the Python programming language, this free and open source software library provides multiple functions allowing in particular to directly create an array from a file or on the contrary to save an array in a file, and to manipulate vectors, matrices and polynomials. It can also be integrated with C/C++.

To install the package, we need to use the command;

```
Pip install numpy
```

3.3.3.3 Argparse

The argparse module makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and argparse will figure out how to parse those out of sys.argv. The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

3.3.3.4 ImUtils

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, detecting edges, and much easier with OpenCV and both Python 2.7 and Python

3.3.3.5 *Video Stream*

A Video Stream is an attached video file.

3.4 Distance detection in video with OpenCV

In this section, we discuss the steps taken to train our human detection model and the investigated datasets to train the model, followed by experimental results on people detection, social distancing measures.

3.4.1 Data base

As with any deep learning task, the first most important task is to prepare the dataset ,with the realized of OpenCV , Aleksandr Rybnikov add new module called “dnn” deep learning-based distancing detector which is based on the Yolo .For this project the dataset used for this model is COCO dataset which is Common objects in context. Models contained in the Yolo Object detection model zoo are trained to detect over 80 different kinds of objects labeled in this COCO dataset. This list of objects like a car, a toothbrush, a banana, a person etc. Efficiency of the system depends on the speed of the model.

3.4.1.1 Features of the COCO dataset

- Object segmentation .
- Superpixel stuff segmentatio
- Over 200’000 images of the total 330’000 images are labeled
- 1.5 Million object instances
- 80 object categories, the “COCO classes”, which include “things” for which individual instances may be easily labeled (person, car, chair, etc.)
- 91 stuff categories, where “COCO stuff” includes materials and objects with no clear
- Boundaries (sky, street, grass, etc.) that provide significant contextual information.
- 5 captions per image
- 250’000 people with 17 different key points

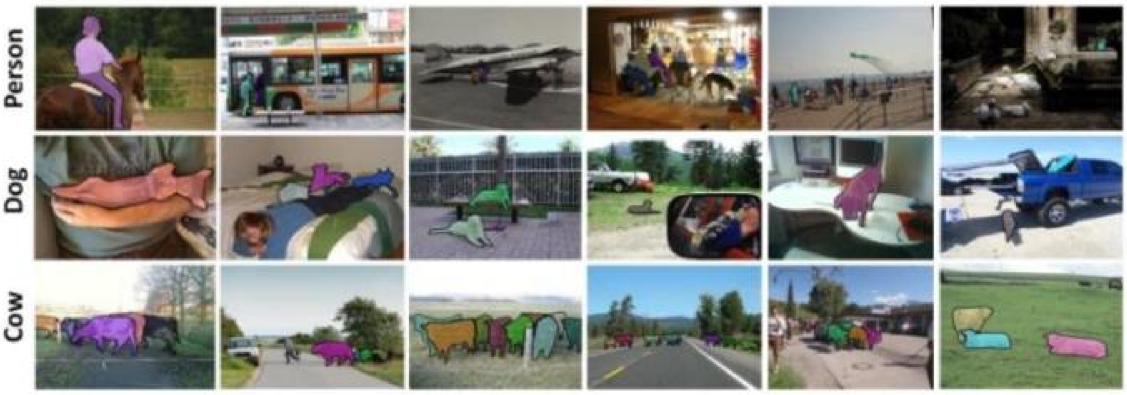


Figure 3.4: Samples from COCO Dataset

3.4.1.2 Get The COCO Data

To train YOLO you will need all of the COCO data and labels. Here we are loading our model in .net variable using cv2.dnn , as we mention before , the network we use it is Darknet .

- **Net**: Many layers’ computes gradients via forward / backward

- **Dnn:** Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks. it have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, medical image analysis…, where they have produced results comparable to and in some cases surpassing human expert performance

- Configpath : file includes yolov3.config .
- weights Path: file include yolov3.weights.

3.4.1.3 Modify cfg for COCO

Before starting to train, you must prepare your data for object detection. To prepare your data, you can use LabelImg tool. In this tool, you can prepare your data in two different format; xml and txt. For yolov3, txt format must be.

- YOLOv3 needs certain specific files to know how and what to train. You must create these three files (.data, .names, and .cfg).

Classes = “80”, number of class in your data set
train = train file path, <path-to-coco>/trainvalno5k.txt
valid = <path-to-coco>/5k.txt
names = class names file,” data/coco.names”
backup = where you want to store the Yolo weights file “backup”

- You should replace <path-to-coco> with the directory where you put the COCO data.
- Every new category should be on a new line, its line number should match the category number in the*.txt*label files we created earlier.

3.4.2 diagram of social distancing detector

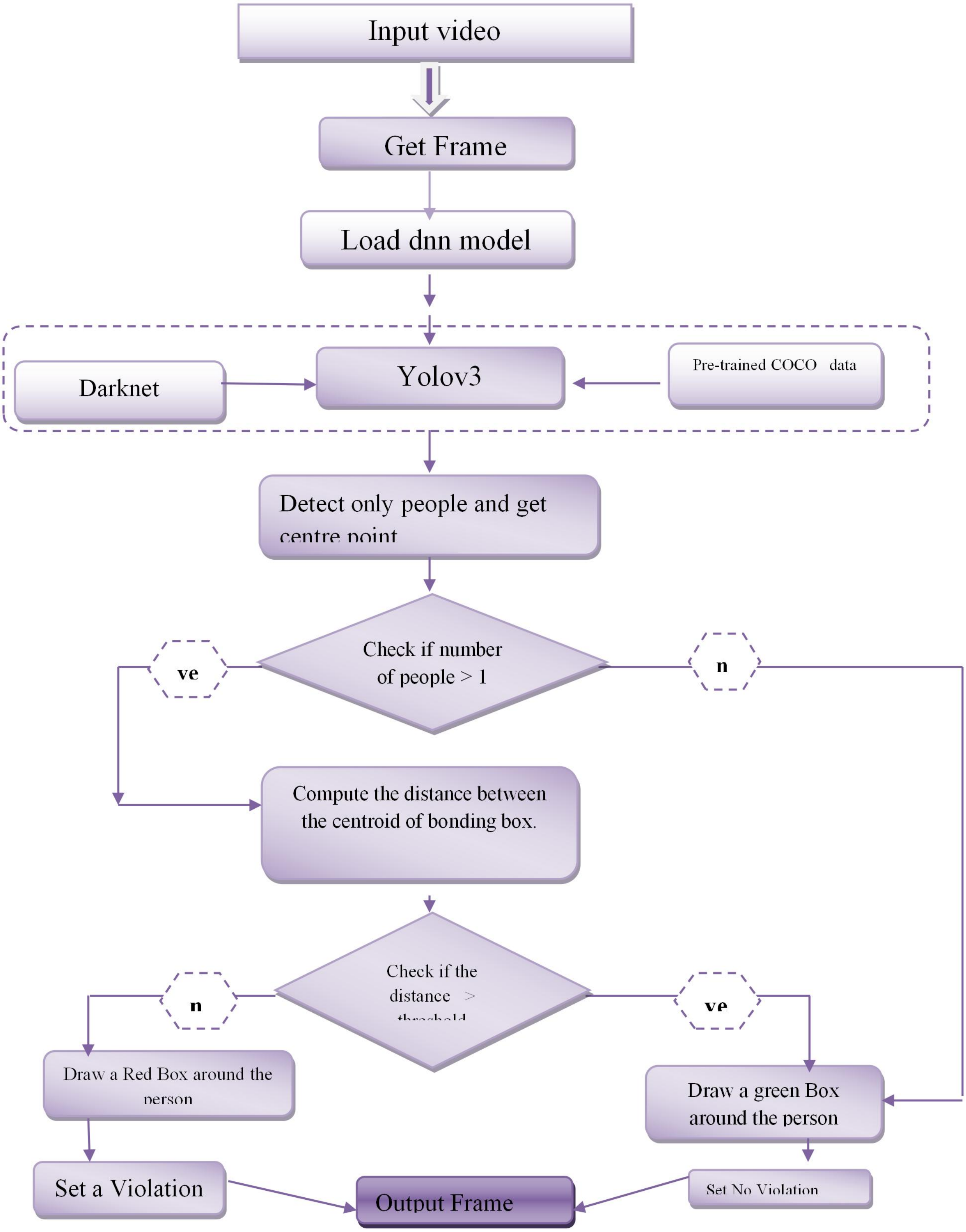

Input video
Get Frame
Load dnn model
Darknet
Yolov3
Pre-trained COCO data
Detect only people and get centre point
Check if number of people > 1
ve
n
Compute the distance between the centroid of bonding box.
Check if the distance > threshold
n
ve
Draw a Red Box around the person
Draw a green Box around the person
Set a Violation
Output Frame
Set No Violation

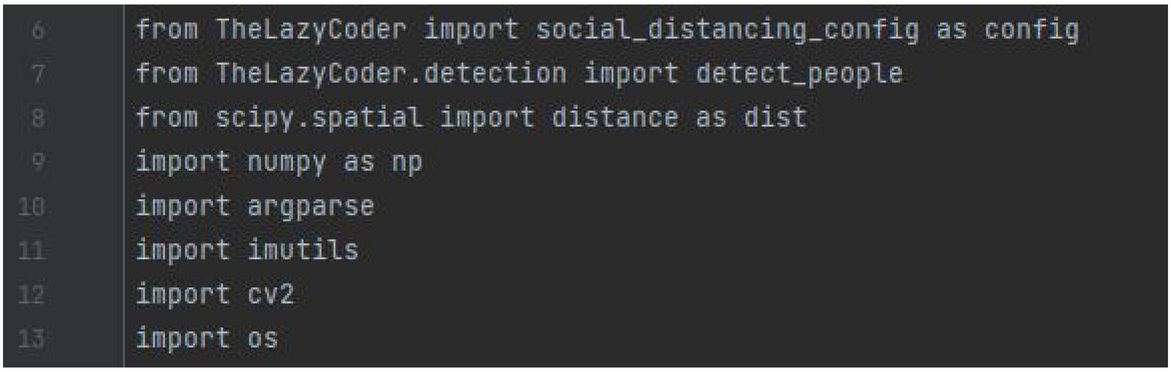
Figure3.5: Diagram of social distancing

3.4.3 Distance detection in video stream with OpenCV and deep learning

The first task of detecting social distancing is to prepare the data set which is called coco-dataset,which is a new unit with the realization of OpenCV.

The deep learning social distancing detector is called dnn which is based on the yolov3 framework with a darknet core which will allow us to detect social distancing with fast performance using OpenCV.

At the first of this stage, we need to import the necessary packages so that our model could run:

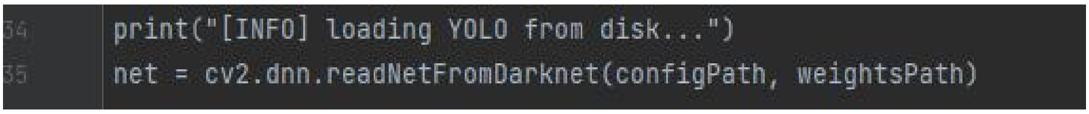


```
from TheLazyCoder import social_distancing_config as config
from TheLazyCoder.detection import detect_people
from scipy.spatial import distance as dist
import numpy as np
import argparse
import imutils
import cv2
import os
```

Figure3.6: Importing the package of libraries

After importing the libraries, and Using OpenCV DNN module, we load our YOLO net into the memory.

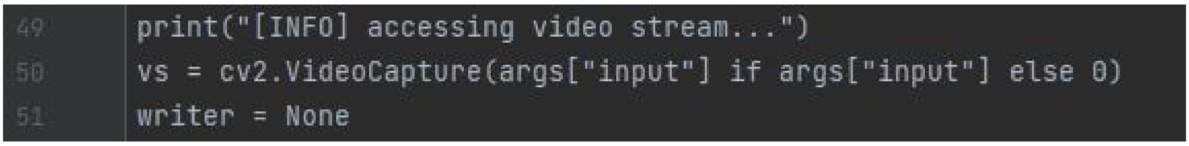
Here we are loading our model in "net" variable using cv2.dnn.readNetFromDarKnet() function.



```
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
```

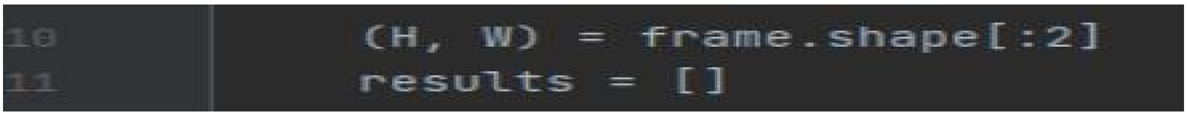
Figure3.7:loading Yolo.network

After that we load the video with cv2.VideoCapture () function and args "input"(which is the video input is entered by the user) and then extracting height and weight dimensions for every frame with frame. Shape[:2] (line10 and 11)where we extract the first and the second dimension.



```
print("[INFO] accessing video stream...")
vs = cv2.VideoCapture(args["input"] if args["input"] else 0)
writer = None
```

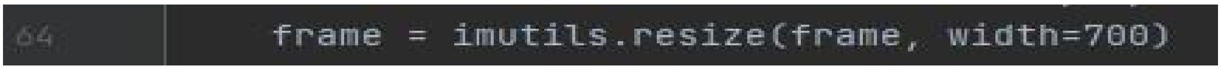
Figure3.8:input video



```
    (H, W) = frame.shape[:2]
    results = []
```

Figure3.9: Dimension for frame of video

We need to resize the frame by using theimutils.resize () Function, to 416×416 dimension, The dimensions of our input video for testing are quite large, so we resize each frame while maintaining aspect ratio. at the end function returns a “blob” which is our input frame .

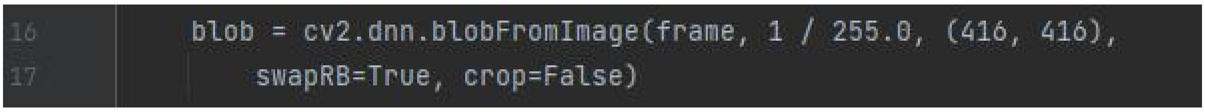


```
    frame = imutils.resize(frame, width=700)
```

Figure3.10:Resizing the frame

The input to the network is called blob object, Pre-processing our frame requires that we construct a blob. It has the following parameters:

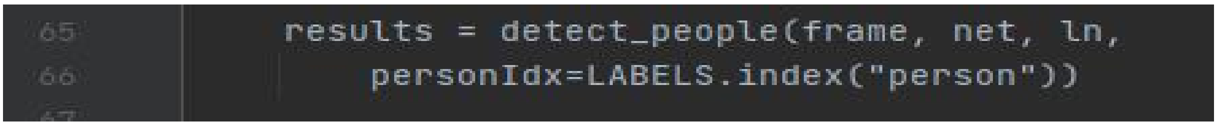
- The image to transform.
- The scale factor (1/255 to scale the pixel values to [0..1])
- The size ,here a 416×416 square image
- The option swapRB=true (since Opencv uses BGR).



```
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
        swapRB=True, crop=False)
```

Figure3.11:preparing the blob

After that we need to detect person, we gather the output layer names from YOLO; we’ll need them in order to process our results, the function we use it to detect only person:



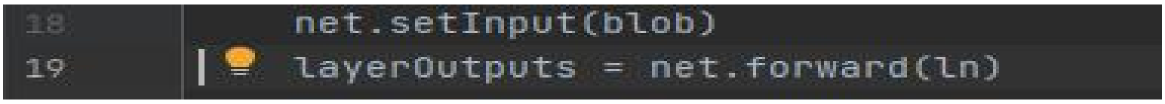
```
    results = detect_people(frame, net, ln,
        personIdx=LABELS.index("person"))
```

Figure3.12: Detect Person

The function accepts four parameters:

- **frame**: The frame from our video file or directly from our webcam
- **Net:** The pre-initialized and pre-trained YOLO object detection model
- **Ln**: The YOLO CNN output layer names
- **personIdx**: The YOLO model can detect many types of objects; this index is specifically for the person class, as we won't be considering other objects.

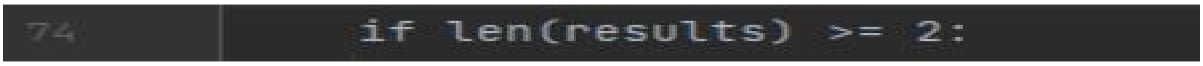
With setInput() we are setting new input value for the network which is our "blob" and with forward() we are running forward pass to compute output of layer. We'll need them in order to process our results.



```
    net.setInput(blob)
    layerOutputs = net.forward(ln)
```

Figure3.13: Setting new input

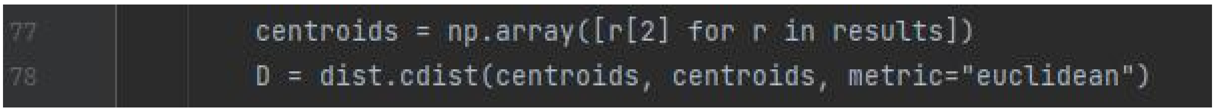
Next, using our detect people function implemented that we mentioned before , we grab results of YOLO object detection. We then check if there are more then 2 person in line 74.



```
74        if len(results) >= 2:
```

Figure3.14: Check the number of people.

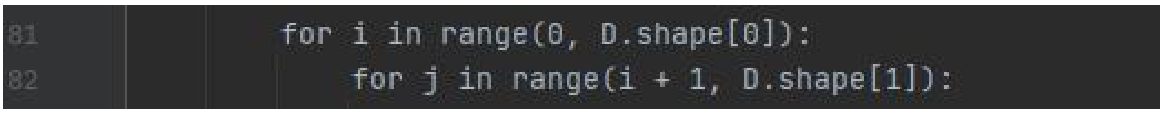
Next, we compute the Euclidean distance after getting the centroid of bonding box, we use the function below:



```
        centroids = np.array([r[2] for r in results])
        D = dist.cdist(centroids, centroids, metric="euclidean")
```

Figure3.15: Compute the distance

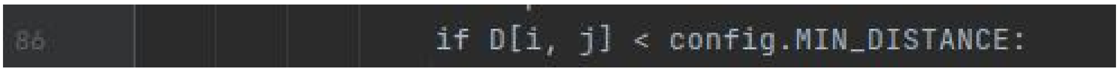
Then loops over the upper triangular of the distance matrix using the commands in figure(3.16);



```
        for i in range(0, D.shape[0]):
            for j in range(i + 1, D.shape[1]):
```

Figure3.16:Looping the upper triangular

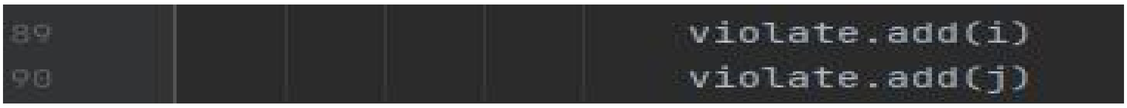
We now check to see if the distance between any two centroid pairs is less than the threshold:



```
if D[i, j] < config.MIN_DISTANCE:
```

Figure3.17: Check the distance

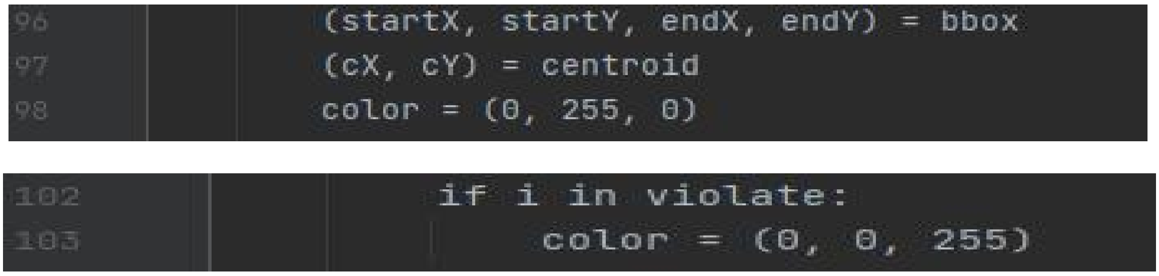
Then we set violate, this set contains a listing of people who violate social distance regulations (which are set forth by public health professionals)



```
violate.add(i)
violate.add(j)
```

Figure3.18:Set violation

After getting the results, the algorithm determines the color of the box in green or red, if the people are keeping safe distance between themthebonding box will be in green [line98],else if the violation occurred then the color of the box willbe in red [line 103].

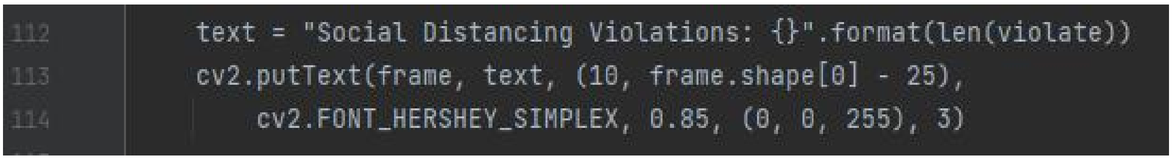


```
96        (startX, startY, endX, endY) = bbox
97        (cX, cY) = centroid
98        color = (0, 255, 0)
```

```
102        if i in violate:
103            color = (0, 0, 255)
```

Figure3.19: determine the color of box.

At the end, we display the total number of violations on the output frame.



```
text = "Social Distancing Violations: {}".format(len(violate))
cv2.putText(frame, text, (10, frame.shape[0] - 25),
    cv2.FONT_HERSHEY_SIMPLEX, 0.85, (0, 0, 255), 3)
```

Figure3.20: setting the number of the violations.

3.4.4 OpenCV social distancing detector results:

In this section, we present the comparison between different outputs of the 3 videos (the calibration of the cameras is varies in this videos,) framework to get a good view and to evaluate the performance of the distance detection method by deep learning. Then we analyzed and annotated it.

3.4.4.1 Results and discussion

We will see different results in figures (3.21),(3.22),(3.23);

- video 1: You can clearly see a good detection result**,** But we note in the third image that there is an error in the result of the detection, as the correct distance was not estimated due to the distance of people from the camera.

- video 2: The process of detecting social distancing in this video is relatively correct, although the camera covered a large area, we see that the detection result is excellent for the area closest to the camera, the more people move away from the camera, the more errors in the detection result.

- video 3: Despite the location of the upper camera is suitable for calculating the distance, there are many errors in the detection result due to a large overlap in the bounding boxes and the inability of the program to estimate the given reference distance.

- **Video 1**

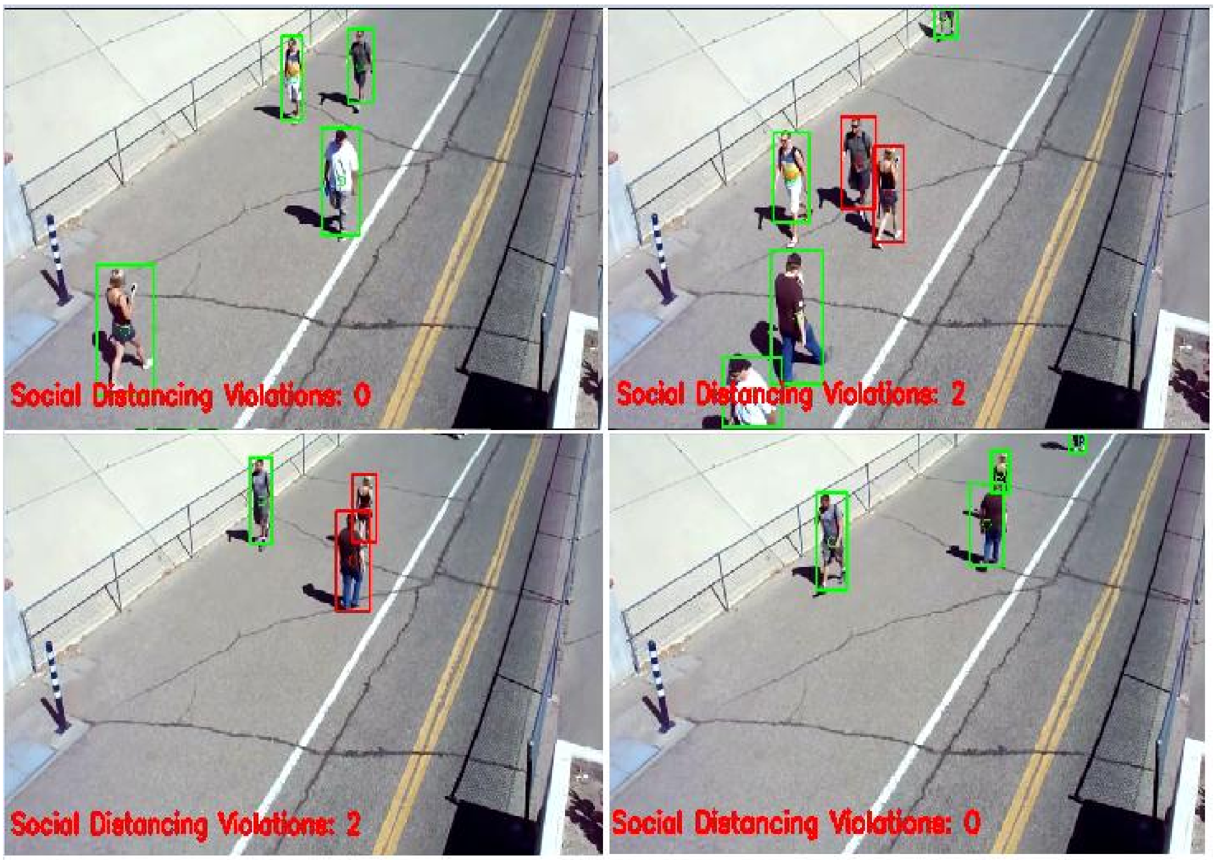

Social Distancing Violations: 0
Social Distancing Violations: 2
Social Distancing Violations: 2
Social Distancing Violations: 0

Figure3.21:Results of social distancing detector on video1

- **Video 2**

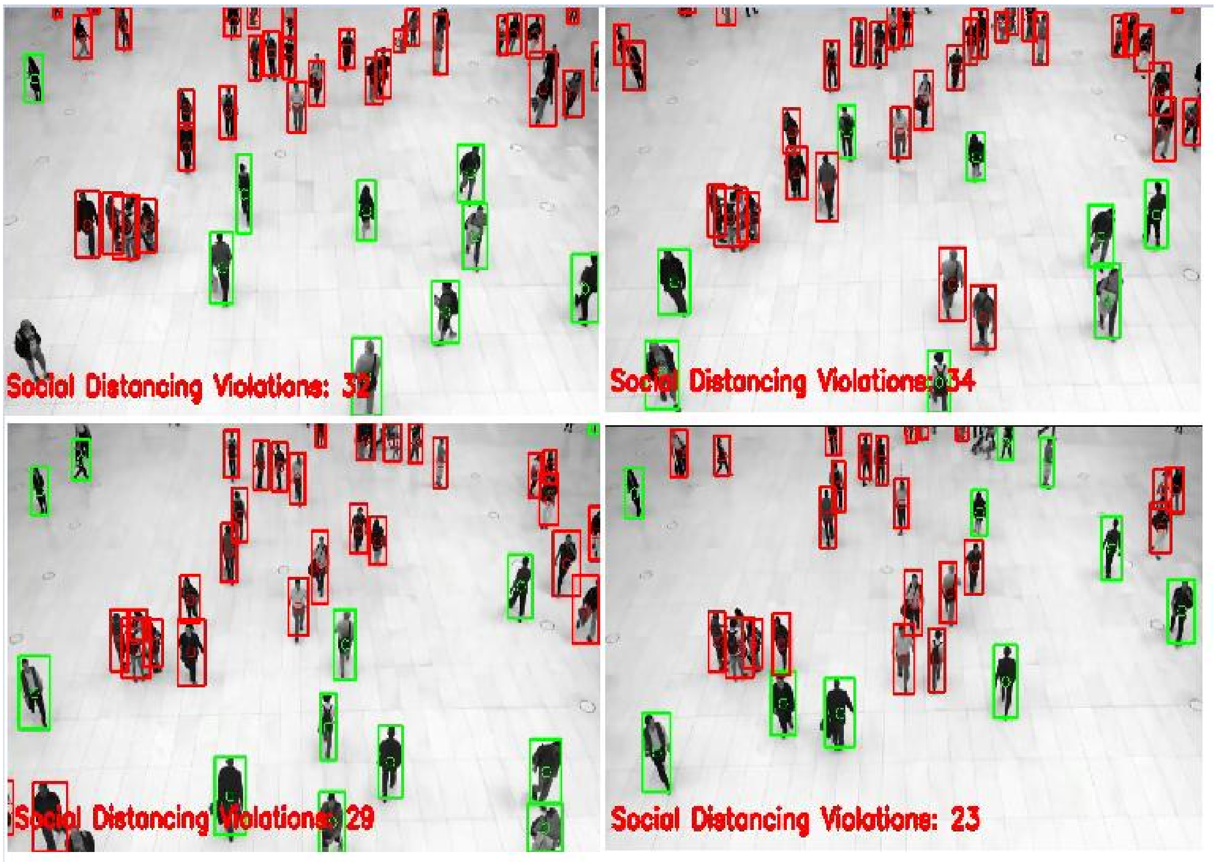

Social Distancing Violations: 32
Social Distancing Violations: 34
Social Distancing Violations: 29
Social Distancing Violations: 23

Figure3.22:Results of social distancing detector on video2

- **Video 3**

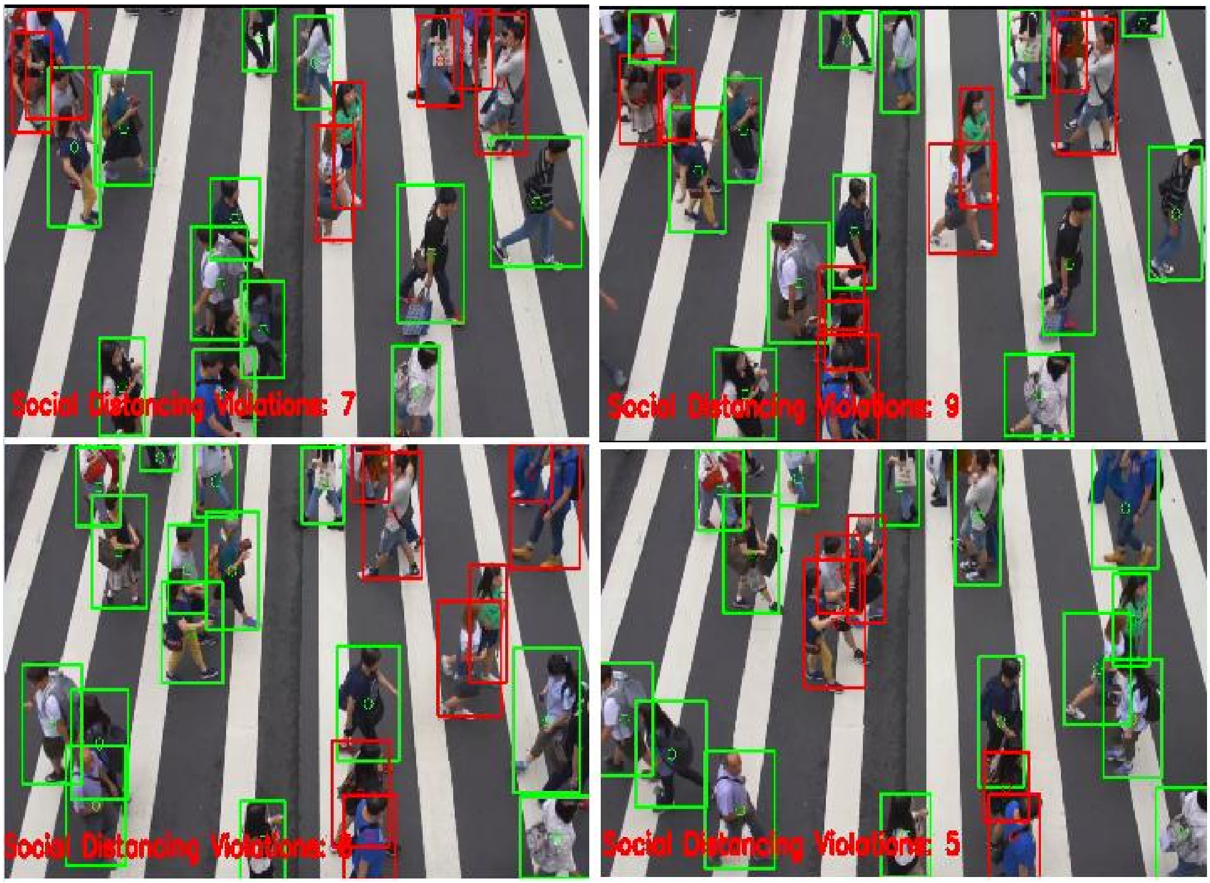

Social Distancing Violations: 7
Social Distancing Violations: 9
Social Distancing Violations: 4
Social Distancing Violations: 5

Figure3.23:Results of social distancing detector on video3

3.5 Conclusion

In this work, we studied an algorithm for detecting social violations using an in-deep learning model. Using computer vision, the distance between people can be measured and any people who do not obey the law was shown in a red Frame. The used model was confirmed using a video showing pedestrians walking down the street. The visual results showed that the used method it is able to determine measures of social isolation that can be developed for use in other environments such as offices, restaurants, and schools. In addition, the work can be improved by expanding the pedestrian detection algorithm, integrating other detection algorithms such as mask detection and human body temperature detection, improving computer hardware power, and measuring camera view.

General Conclusion

The learning of computer vision through its important applications to improve public safety has benefited us greatly since the emergence of COVID-19, Emerging developments and the availability of smart technology allow us to create new models that help meet the needs of the world. So here, we have created a social distance detector that could potentially apply to public health care. In this era of COVID-19, social distancing is very important to reduce a deadly virus that is transmitted from one person to another. The Public Health Organization [1] announced a strict decision to keep a distance of at least 2 meters between two people until urgent medicine is found for that. However, many of us avoid implementing this, as it is difficult, especially in our generation. Despite that, the government focused on finding an effective solution to control the massive crowd, in our project, we focused mainly on that.

In human terms, it is very difficult to monitor whether or not people are following social distancing in crowded places. However, it can be done easily with the help of computers, cameras, and some advanced object detection techniques. All these things gave us the idea to study our project. To overcome this problem, we have studied a social distancing detection program that detects whether a person leaves a safe distance of at least 2 meters or not. The installation of this software module in public places, in the work environment and in various crowded areas will make it easier to monitor the detection of spacing than to monitor by human power.

The aim of our work was to see how the process of monitoring social distancing via object detection and tracking methods, where each individual is identified with the help of bounding boxes. The created bounding boxes help to identify groups of people who satisfy the distance threshold calculated with the help of calculating the Euclidean distance between pedestrians. The number of violations is confirmed by counting the red rectangles that appeared on nearby people. In this work, we used the latest object detection model YOLO v3, where we show that YOLO v3 deals with object detection in a different way, by taking the entire image in one case and predicting bounding box coordinates and class probabilities for these boxes. We use in the simulation part, the widely used high-level programming language Python, the computer vision library OpenCV, and the PyCharm IDE. We used the yolov3 detector, which uses the pre-trained coco, and OpenCV's darknet network of the DNN module and the algorithm detector to perform its function. Deep neural networks have always shown very successful results when it comes to discovering things.

The results obtained show that the Yolov3 detector is the best at detection. It is one of the fastest object detection algorithms that use a variant of Darknet originally trained on Imagenet consisting of 53 layers, and also the model runs in real-time on the CPU. However, the detector had a problem detecting the real distance when there is a large distance from the camera to the person and also when there is an overlap in boxes. The social distancing detector did not leverage a proper camera calibration, meaning that we could not (easily) map distances in pixels to actual measurable units (i.e., meters, feet, etc.). Therefore, the first step to improving our social distancing detector is to utilize proper camera calibration. Doing so will yield better results and enable you to compute actual measurable units (rather than pixels).The problem with this current implementation is *speed* Our CPU-based social distancing detector is obtaining**~2.3 FPS**, which is *far too slow* for real-time processing.

References

[1] *W.H.Organization,"WHOcorona-viruses(COVID-19),"emergenciesdiseases/novel-corona-virus-2019,* Online; accessed May 02, 2020.

[2]Mohd Zafri Baharuddin;Salman Yussof;Sumayyah Dzulkifly, '' Social Distancing Detectionwith Deep LearningModel'' ,Publisher:IEEE,DOI:10.1109/ICIMU49871.2020.9243478,2020 8th International Conference on Information Technology and Multimedia (ICIMU) **.**

[3]Kieran O'Shea and Ryan Nash, 'An Introduction to Convolutional Neural Networks'', 2015.

[4]ANIL DNYANDEV SAPKAL, CHAITANYA RATHOD, AJMAL SHAIKH, ''OpenCV Social Distancing Detector'', JETIR, Volume 8, Issue, January 2021

[5]International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056, Volume: 08 Issue: 01, available=www.irjet.net, p-ISSN:2395-0072 , Jan 2021.

[6]Santa Clara Valley Health & Hospital System, «Information About Social Distancing", www.sccphd.org ,185_factsheet_social_distancing.pdf .

[7] D. Zhang, al, "Local–Global Attentive Adaptation for Object Detection, «Engineering Applications of Artificial Intelligence,vol. 100, pp. 104208, 2021.

[8]Xuejiao Wang, Min Zhi, "Summary of object detection based on convolutional neural network," Proc. SPIE 11373, Eleventh International Conference on Graphics and Image Processing (ICGIP 2019), 113730L (3 January 2020) .

[9]Trad Houssem-eddin,''Detection d'objet avec opencv et deep learning **'**',Université Mohamed Khider de Biskra ,Faculté des Sciences et de la Technologie

Département de Génie Electrique ,30 septembre 2020

.

[10]Posted by Jędrzej Świeżewski," Introduction to YOLO Algorithm and YOLO Object Detection " Ph.D. ,22 May, 2020 .

[11]BENMEHAOAUI Zied,"Object detection with YOLO and Python on images and video_Master Thése ",Université Mohamed Khider de Biskra -2020 – 2021 .

[12] Moualek.djeloul .youcef, " deeplearning pour la classification des image"- Master Thése. Universite abou bakr belkaid-telemcen- 2016/2017 .

[13]Hans-Dieter,"wehle -Machine Learning, Deep Learning, and AI: What's the Difference?", 2017.

[14]Ed Burns:Executive Editor,Kate Brush:Technical Write ,"deep-learning-deep-neural-network", TechTarget, 2018 - 2022,availible= https://www.techtarget.com/ .

[15]Dr. J. KARTHIKEYAN, Dr.TING SU HIE, Dr. NGYU JIN ,''LEARNING OUTCOMES OF CLASSROOM RESEARCH'', December 2021.

[16]Ivan Vasilev,DanielSlater,Gianmario Spacagna,Peter Roelants,Valentino Zocca,"Python Deep Learning", 2019 Packt Publishing.

[17]Arne Wolfewicz,"Deep learning vs. machine learning ,"What's the difference ",November 9, 2021,availible=https://levity.ai/.

[18]saagie,''qu'est-ce-que-la-detection-dobjet'',13/11/2020, availible=https://www.saagie.com/ .

[19] Girshick, R., Donahue, J., Darrel, T.,Malik, J. Rich Feature Hierarchies for Accurate ''Object Detection and Semantic Segmentation'', pp. 580-587 ,In: Computer Vision and Pattern Recognition. Columbus.2014,.

[20]Jun Deng,Xiaojing Xuan,Weifeng Wang,Zhao Li,Hanwen Yao,Zhiqiang Wang, "A review of research on object detection based on deep learning", To cite this article: Jun Deng et al 2020 J. Phys.: Conf. Ser. 1684 012028 .

[21] Fayyad, Jamil, Mohammad A. Jaradat, Dominique Gruyer, and Homayoun Najjaran, "Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review",2020,Sensors20, no. 15: 4220,availible=https://doi.org/10.3390/s20154220 .

[22]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed,'' SSD: Single shot multibox detector'', European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, Oct. 10-16, 2016; Leibe B., Matas J., Sebe N., Welling M., Eds; Publisher: Cham, Switzerland, 2016. Doi: 10.1007/978-3-319-46448-0_2.

[23]Models Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee, "A Survey of Modern Deep Learning based Object Detection ", 2104.11892.pdf,arXiv:2104.11892v2 [cs.CV] 12 May 2021.

[24]G. Bai,al,"An intelligent water level monitoring method based on SSD algorithm,"Measurement : Journal of the International Measurement Confederation,vol: 185, pp. 110047, 2021.

[25]Redmon, Joseph and Farhadi, Ali ,''YOLOv3: An Incremental Improvement'',journal {arXiv}, 2018.

[26]Ben Dickson,'' An introduction to object detection with deep learning thése '' , June 21, 2021

[27]Adrian Rosebrock, pyimagesearch,"YOLO object detection with OpenCV",onNovember 12, 2018.

[28]Eduardo Todt,Bruno Alexandre Krinski ,"Convolutional Neural Network – CNN",VRI Group -,Vision Robotic and ImagesFederal University of Parana-2019.

[29]Tang, Cong, Yongshun Ling, Xing Yang, Wei Jin, and Chao Zheng. "Multi-view object detection based on deep learning."Applied Sciences8, no. 9 (2018): 1423.

[30] Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards accurate region proposal generation and joint object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 845–853, Las Vegas, NV, USA, 26 June–1 July 2016.

[31] Radhika Kumari1, Ms. Upasna Joshi2,’’Deep Learning and Computer Vision-based Social Distancing Detection System’’,(IRJET).

[32]Hui Wang, Fan Zhang, Li Wang -Fruit ,’’Classification Model Based on Improved Darknet53 Convolutional NeuralNetwork’’,College of Information Engineering, Minzu University of China, Beijing 100081

[33]Alex Krizhevsky,lya Sutskever-Geoffrey E, Hinton»ImageNet Classification with Deep ConvolutionalNeural Networks»University of Toronto.

[34]Jacob Solawetz,”YOLOv5 New Version - Improvements And Evaluation”,Jun 29, 2020.

[35]Wentong Wu, Han LiuID, Lingling Li, Yilin Long, Xiaodong Wang, Zhuohua Wang, Jinglun Li, Yi Chang,’’Application of local fully Convolutional NeuralNetwork combined with YOLO v5 algorithm insmall target detection of remote sensingimage’’,October 29, 2021.

[36]Narinder Singh Punn, Sanjay Kumar Sonbhadra, Sonali Agarwal and Gaurav Rai,”Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques”.

[37] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma, “A Review of Yolo Algorithm Development “,The8th International Conference on Information Technology and Quantitative Management,(ITQM 2020 & 2021).

[38] Wuxi 214122, , Wuxi 214122,ChinaTinier, ”YOLO: A Real-Time Object DetectionMethod for Constrained Environments/ date of current version” , 1School of IoT Engineering, Jiangnan University, China2Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, January 6, 2020.

[39]Dihua Wu, Shuaichao Lv, Mei Jiang, Huaibo Song,”Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments”,Computers and Electronics in Agriculture 178 ,2020.

[40] Do Thuan ,” EVOLUTION OF YOLO ALGORITHM AND YOLOV5: THE STATE-OF-THE-ART OBJECT DETECTION ALGORITHM “, Oulu University of Applied Sciences.

[41]Yunong Tian, Guodong Yang, Zhe Wang, Hao Wang, En Li, Zize Liang-,“Apple detection during different growth stages in orchards using the improved YOLO-V3 model”,Computers and Electronics in Agriculture.

[42]Ayoosh Kathuria, “What’snewin YOLO v3”, availible=”https://towardsdatascience.com”

[43] Dhiraj and D. K. Jain, "An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery," Pattern Recognition Letters, vol. 120, pp. 112-119, 2019.

[44]Rohith Gandhi,“R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms”,availible=”https://towardsdatascience.com” .

[45]HAIPENG CHEN , ZHENTAO HE , BOWEN SHI2, AND TIE ZHONG,”Research on Recognition Method of Electrical Components Based on YOLO V3”, publication October 29, 2019.

[46]JędrzejŚwieżewski, ”Ph.D.-YOLO Algorithm and YOLO Object Detection”, availible=”https://appsilon.com” .

[47] Joseph Redmon ,Ali Farhadi,” YOLOv3: An Incremental Improvement”, University of Washington, 8, Apr 2018.

[48]https://datahacker.rs/deep-learning-bounding-boxes/

[49] Derui Wang, Chaoran Li, Sheng Wen, Qing-Long Han, Fellow, IEEE, Surya Nepal, Xiangyu Zhang,and Yang Xiang, Fellow, IEEE-Daedalus:” Breaking Non-Maximum Suppression in Object Detection via Adversarial Examples”,1 Jun 2020.

[50] https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c.

[51] Saponara, Sergio, Abdussalam Elhanashi, and Alessio Gagliardi. "Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19.",Journal of Real-Time Image Processing18, no. 6 (2021): 1937-1947.

[52] basil roth,“A social distancing detector using a Tensorflow object detection model, Python and OpenCV “,doi =281704bd-67a3-4631-b28f-2d989f188fe5,Jun 17, 2020 , availible=”https://deepnote.com/ ”.

[53]Wang, Y., Jia, K., & Liu, P.,’’Impolite pedestrian detection by using enhanced yolov3-tiny. Journal on Artificial Intelligence’’, 2(3), 113-124,2020.

[54] Levon Khachatryan, “CENTROID BASED OBJECT TRACKING”,availible =” https://github.com/lev1khachatryan/Centroid-Based_Object_Tracking ”

[55] J. C. Nascimento, A. J. Abrantes, and J. S. Marques, “An algorithm for centroid-based tracking of moving objects,” in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 6, March 1999, pp. 3305–3308.

[56] Afiq Harith Ahamad, Norliza Zaini, Mohd Fuad Abdul Latip ,Faculty of Electrical Engineering Universiti Teknologi MARA (UiTM) Shah Alam Selangor, Malaysia,” Person Detection for Social Distancing and Safety Violation Alert based on Segmented ROI”, 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE2020), 21–22 August 2020, Penang, Malaysia.

[57]AqeelAnwar,’’monitoring-social-distancing-usingai”,Apr20,2020,availible=https://towardsdatascience.com/monitoring-social-distancing-using-ai-c5b81da44c9f.

[58] Chaitanya rathod , “opencv social distancing detector “,JETIR “Journal of Emerging Technologies and Innovative Research ” , Volume 8, Issue 1 , Published2021.

[59]Dawei Zhang,Published under licence by IOP Publishing Ltd , Journal of Physics: Conference Series,Volume 1881,The 2nd International Conference on Computing and Data Science (CONF-CDS) 2021 28-30 January 2021, Stanford, United States

[60]Jun wang ,“Implementation of bird's-eye view system for camera-on-vehicle “,Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on ,DOI:10.1109/ICCE.2010.5418845 ,IEEE Xplore , February 2010.

[61]Avadhut Joshi1 , Chetan Narkhede2 , Omkar Jadhav3 , Jayashree Mohadikar4,”social distancing detector using open cv”, IJARCCE.2021.10627.pdf, Vol. 10, Issue 6, June 2021 DOI 10.17148/IJARCCE.2021.10627.