CROWD DETECTION AND SOCIAL DISTANCE TRACKER

A Project Report

Submitted by:

AJAY VENU V S- LMC20MCA2003

in partial fulfillment of the requirements for the award of

the degree in

MASTER OF COMPUTER APPLICATIONS

at

DEPARTMENT OF COMPUTER APPLICATIONS

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY
KUTTICHAL, THIRUVANANTHAPURAM

JULY 2022

LOURDES MATHA COLLEGE OF SCIENCE AND TECHNOLOGY

(Managed by Archdiocese of Changanacherry) (Affiliated to APJ Abdul KalamTechnological University, Kerala) KUTTICHAL, THIRUVANANTHAPURAM-695574

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project work entitled "**CROWN DETECTION AND SOCIAL DISTANCE TRACKER**" is a bonafide record of the work done by **AJAY VENU V S**, Reg No **LMC20MCA2003**, student of Department of Computer Applications, Lourdes Matha College of Science & Technology, Kuttichal, Thiruvananthapuram, affiliated to the APJ AbdulKalam Technological University, Kerala from April 2022 to july 2022 in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications from APJ Abdul Kalam Technological University, Kerala.

Prof. Bismi K Charleys
(Internal Guide)

Prof. Bismi K Charleys
(Head of the Department)

Internal Examiner **Date:** **External Examiner**

DECLARATION

I undersigned here by declared that the project report" **CROWD DETECTION AND SOCIAL DISTANCE TRACKER"** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala. This submission represents my idea in my own words and, I have adequately and accurately cited and referenced the original sources .I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact of source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University.

Place: Trivandrum AJAY VENU V S

Date :__/0__/2022

ACKNOWLEDGEMENT

An endeavor over a long can be successful only with advice and support of many well – wishers. I wish to place on record my profound indebtedness and gratitude to all those who have contributed directly or indirectly to make this project work a success.

At the very onset, I express my gratitude to God Almighty who sheltered me under his protective wings and showered on innumerable blessings throughout the period of this Master of Computer Application.

It is a great pleasure to express my sincere gratitude to **Rev.Fr.Bejoy Arackal**, Director and **Dr.BeshibaWilson**, Principal Lourdes Matha College of Science and Technology for permitting to do this project with the fullest spirit.

I am highly obliged to **Prof.Bismi K charleys** ,Head of the Department of Computer Applications of Lourdes Matha College of Science and Technology, for being the source of inspiration throughout the course and for her valuable guidance.

With heart full of thanks , I would like to take up this opportunity to wish my Internal guide **Prof.Bismi K charleys**,Assistant Professor and all staffs of Department of Computer Applications for their endless support, encouragements and suggestions in various stages of the development of this project. Finally, I wish to express my sincere gratitude to all our friends who directly or indirectly contributed in this venture.

CONTENTS

Content Page No

ABSTRACT……………………………………………………………………………. 1

CHAPTER 1……………………………………………………………….....................2

INRODUCTION………………………….....…………...…………………….……………...2

1.1 General Introduction…………………………………………………………3

1.2 Goal of the project………………………………………………………...3

CHAPTER 2……………………………………………………………………………4

LITERATURE SURVEY…………………………………………………………4

2.1 Study of Similar Work……………………………………………………5

2.2 Existing System……………………………………………......................5

2.2.1 Drawback of Existing System……………………………………..5

CHAPTER 3……………………………………………………………………………6

OVERALL DESCRIPTION………………………………………………………6

3.1 Proposed System……………………………………………………………7

3.2 Features of proposed System……………………………………………7

3.3 Functions of Proposed System………………………………………….….8

3.4 Requirement Specification…………………………………………………10

3.5 Feasibility Analysis……………………………………………………...10

3.5.1 Technical Feasibility………………………………………………..10

3.5.2 Operational Feasibility……………………………………………11

3.5.3 Economical Feasibility…………………………………………...11

3.5.4 Behavioral Feasibility……………………………………….………11

CHAPTER 4……………………………………………………………..………………12

OPERATING ENVIRONMENT………………………………………….............12

4.1 Hardware Requirement……………………………………………………....13

4.2 Software Requirement…………………………………………………..…13

4.3 Tools and Platforms……………………………………………………...…13

4.3.1 Visual Studio……………………………………….……..……13

4.3.2 Python……………………………………………..……...……..14

4.3.3 Deep Learning………………………………………….………..14

4.3.4 CNN…………………..……………………………………….....14

4.3.5 Open CV…………………………..…………………………......15

4.3.6 YOLO V3………………………………………………………...…16

4.3.6.1 Working of YOLO V3………………………….…………….…16

4.3.6.2 Proposed Architecture Description………………………………..17

4.3.6.3 Proposed Architecture Characteristics…………………………......17

4.3.6.4 Block Diagram……………………………………………………..18

4.3.6.5 Layers Details………………………………………………...…18

4.3.6.6 Convolution Layers in YOLO V3………………………….......18

4.3.6.7 How YOLO Detects………………………………………………..19

4.3.7 Tensor Flow…………………………………………………...….20

4.3.8 COCO Dataset………………………………………………………20

4.3.8 Features of the COCO Dataset……………………………….……..20

4.3.8.2 List of Coco objects and classes…………………………….…...21

4.3.8.3 Annotated COCO Images…………………………………….…22

4.3.8.4 What makes COCO Unique……………………………………….23

4.3.8.5 Feature List and Class of the dataset………………………………23

4.3.9 Image Processing……………………………………………………...24

CHAPTER 5………………………………………………………………………..26

DESIGN……………………………………………………………………………...26

5.1 System Design……………………………………………………………...27

5.2 Program Design…………………………………………………………….....27

5.3 Use Case Diagram……………………………………………………….….28

5.4 Activity Diagram……………………………………………………………28

5.5 Proposed Project Pipeline……………………………………………...…...29

5.6 Input Design…………………………………………………………….....…30

5.7 Output design……………………………………………………………….30

CHAPTER 6…………………………………………………………………………31

FUNCTIONAL AND NON FUNTIONAL REQUIREMENT…………………………….31

6.1 Functional requirement……………………………………………….......32

6.2 Non Functional Requirements……………………………………………...32

CHAPTER 7………………………………………………………………………...33

TESTING……………………………………………………………………….…33

7.1 Testing Process……………………………………………………………..…34

7.1.1 Unit Testing……………………………………………………………….35

7.1.2 Alpha Testing……………………………………………………………..35

7.1.3 Acceptance Testing……………………………………………………….35

7.1.4 Beta Testing………………………………………………………………35

7.2 Validation Checks…………………………………………………………...36

7.3 Test cases…..……………………………………………………………..36

CHAPTER 8……………………………………………………………………….37

RESULTS AND DISCUSSION…………..…………………………………….37

8.1 Results……………………………………………………………….…38

8.2 Screen Shots……………………………………………………………39

CHAPTER 9…………………………………………………………….……...42

CONCLUSION……………………………………………………………...…41

9.1 System Implementation…………………………………………….…..42

9.2 Conclusion………………………………………..………………...43

9.3 Feature Scope……………………………………………………………..43

REFERENCES…………………………………………………………….…44

APPENDICES……..…………………………………………………...…45

GIT HISTORY……………………………………………………………….……46

ABSTRACT

Due to the growing population, Crowd management has become an important issue and challenge for security agencies across the world where effective crowd management can prevent serious accidents and in some cases mortalities. Researchers provide effective solutions for social distance measuring using surveillance videos along with computer vision, machine learning, and deep learning-based approaches. We will use the dataset to build a social distance detector with computer vision using Python, Open CV, and Tensor Flow and Yolov3. In our proposed system we will use live video stream and finally in output it shows the number of violations when someone not maintaining social distance. Our goal is to identify whether the person on image/video stream is maintaining social distance or not with the help of computer vision, machine learning and deep learning.

The proposed method presented in this project is mainly for crowd surveillance and security maintenance. This system can be utilized for events, private property or places that have massive streams of people entering and leaving the area, which necessitates vigilant tracking and identification of all individuals within the premises. With deep learning, particularly CNN, the model is able to train and learn to identify human beings using the database customized.

Reference: 1) People Detection System Using YOLOv3 Algorithm" (2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE2020), 21– 22 August 2020, Penang, Malaysia

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

The corona virus COVID-19 pandemic is causing a global health crisis so the effective protection methods is maintain social distance in public areas according to the World Health Organization (WHO). The COVID-19 pandemic forced governments across the world to impose lockdowns to prevent virus transmissions. Reports indicate that maintaining social distance at work clearly reduces the risk of transmission.

We will use the dataset to build a COVID-19 social distance detector with computer vision using Python, OpenCV, and Tensor Flow and Yolo. In our proposed system we will use live video stream and finally in output it shows the number of violations when someone not maintaining social distance. Our goal is to identify whether the person on image/video stream is maintaining social distance or not with the help of computer vision, machine learning and deep learning. The proposed method presented in this paper is mainly for crowd surveillance and security maintenance.

1.2 GOAL OF THE PROJECT

This system can be utilized for events, private property or places that have massive streams of people entering and leaving the area, which necessitates vigilant tracking and identification of all individuals within the premises. With deep learning, particularly CNN, the model is able to train and learn to identify human beings using the database customized.

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF SIMILAR WORK

In recent times, the COVID-19 pandemic has affected billions of people worldwide and has resulted in the slowing down of the economy, industry shutdown, job losses, etc. Every country has taken appropriate measures to fight against pandemic by keeping in mind that health is the primary concern for human beings. This work introduces the COVID-19 pandemic and discusses its types, influence over mankind, prevention methods, and latest observations.

2.2 EXSITING SYSTEM

Further, this study has designed drone-based case studies for pandemic monitoring, social distance measurements, the necessity of the control room, etc. The simulation is designed to have a single-layer drone movement strategy with a fixed distance. The simulation experimentation is derived from real-time drone movement and area coverage for sanitization. The drone movement and collision avoidance strategy are pre-emptive in nature, i.e., drones are derived to move to a fixed location and execute its functionality. At the ground level, service is designed for which people make queues and maintain social distance before being served.

2.1.2 DRAWBACK OF EXSITING SYSTEM

- Due to growing population crowd management has become a challenge
- Effective crowd management can prevent serious accidents etc..
- Currently social distancing was being measured with drones.
- Requires more effort.
- Time consuming
- Less Accurate

CHAPTER 3

OVERALL DESCRIPTION

3.1 PROPOSED SYSTEM

The Research paper entitled “People Detection System Using YOLOv3 Algorithm”(2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE2020), 21–22 August 2020, Penang, Malaysia) presents a technical approach related to video computer analysis, to detect people and control the crowd . Controlling the behavior of people in public places can be a benefit for understanding the share of overall traffic your area is attracting. Find out what, encourage customers to buy products are most crucial for big companies to increase the sales rate and to improve the quality of customer service. The paper discusses two methods, the first method for detecting people in a closed space and the second method finding density areas which people spend more time to visit. The YOLO model makes predictions with a single network evaluation. Systems like R-CNN and Faster R-CNN, on the other hand, make multiple assessments for a single image, making YOLO extremely fast, running in real-time with a capable GPU. YOLOv3 algorithm has high accuracy to identify people, also we compared the proposed method with other detectors, HOG, SSD and YOLO-tiny which shows the proposed algorithm has better performance at this point. And for finding density areas, We utilized a background subtraction with Gaussian Mixture algorithms and heatmap colour technique to analysis each frame and figure out, where are the density areas which shows people like to spend more time to visit. The experimental results have shown that the accuracy and the performance of both algorithms are quite good.

The proposed method presented in this paper is mainly for crowd surveillance and security maintenance. This system can be utilized for events, private property or places that have massive streams of people entering and leaving the area, which necessitates vigilant tracking and identification of all individuals within the premises. With deep learning, particularly CNN, the model is able to train and learn to identify human beings using the database customized.

3.2 FEATURES OF PROPOSED SYSTEM

The Problem is to Identify crowds and maintain social distancing between people from real time videos and images.

Due to the growing population, Crowd management has become an important issue and challenge for security agencies across the world where effective crowd management can prevent serious accidents and in some cases mortalities.

Researchers provide effective solutions for social distance measuring using surveillance videos along with computer vision, machine learning, and deep learning-based approaches. proposed a framework using the YOLOv3 model to detect humans and the Deepsort approach to track the detected people using bounding boxes and assigned IDs information. They used an open image data set (OID) repository, a frontal view data set.

The authors also compared results with faster-RCNN and SSD developed an autonomous drone-based model for social distance monitoring. They trained the YOLOv3 model with the custom data set. The data set is composed of frontal and side view images of limited people. From the literature, we concluded that the researcher had done a considerable amount of work for monitoring of social distance in public environments. But, most of the work is focused on the frontal or side view camera perspective.

Therefore, in this work, we presented an overhead view social distance monitoring framework that offers a better field of view and overcomes the issues of occlusion, thereby playing a key role in social distance monitoring to compute the distance between peoples

3.3 FUNCTIONS OF PROPOSED SYSTEM

Image processing methods used here:

i) Mean subtraction
ii) Scaling
iii) Optionally channel swapping

i) Mean subtraction

Mean subtraction is used to help combat illumination changes in the input images in our dataset. We can therefore view mean subtraction as a technique used to aid our Convolutional Neural Networks. Before we even begin training our deep neural network, we first compute the average pixel intensity across all images in the training set for each of the Red, Green, and Blue channels. This implies that we end up with three variables: \mu_R, \mu_G, and \mu_B Typically the resulting values are a 3-tuple consisting of the mean of the Red, Green, and Blue channels, respectively.

ii) Scaling

However, in some cases the mean Red, Green, and Blue values may be computed channel-wise rather than pixel-wise, resulting in an MxN matrix. In this case the MxN matrix for each channel is then subtracted from the input image during training/testing. Both methods are perfectly valid forms of mean subtraction; however, we tend to see the pixel-wise version used more often, especially for larger datasets. When we are ready to pass an image through our network (whether for training or testing), we subtract the mean, \mu, from each input channel of the input image:

R = R - \mu_R

G = G - \mu_G

B = B - \mu_B

We may also have a scaling factor, \sigma, which adds in a normalization:

R = (R - \mu_R) / \sigma

G = (G - \mu_G) / \sigma

B = (B - \mu_B) / \sigma

The value of \sigma may be the standard deviation across the training set (thereby turning the preprocessing step into a standard score/z-score). However,\sigma may also be manually set (versus calculated) to scale the input image space into a particular range.

Implementation

cv2.dnn.blobFromImage :

blob From Image creates a 4-dimensional blob from image. Optionally resizes and crops image from center, subtract mean values, scales values by scale factor, swap Blue and Red channels.

3.4 REQUIREMENT SPECIFICATION

- Accuracy: Accuracy in functioning and the nature of user-friendly should be maintained by the system

- Speed: The system must be capable of offering speed.

3.5 FEASIBILITY ANALYSIS

Feasibility study is a test of system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use of resources. Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development. A feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that is spend on it. Feasibility study lets the developer foresee the future of the project and its usefulness. All the projects are feasible given unlimited resources and infinite time. Unfortunately, the development of the computer-based system is more likely to be played by a security of resources and difficulty delivery dates. Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility of producing the quality software is reduced.

3.5.1 TECHNICAL FEASIBILITY

Technical Feasibility centers on the existing computer system (hardware, software, etc) and to what extend it can support the proposed addition. For example, if the current computer is operating at 80 percent capacity, an arbitrary ceiling, then running another application could over load the system or require additional hardware. This involves financial considerations to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible.

3.5.2 OPERATINAL FEASIBILITY

The main problem faced during development of a new system is getting acceptance from the user. People are inherently resistant to changes and computers have been known to facilitate change. It is mainly related to human organizational and political aspects.

The points to be considered are:

- What changes will be brought with the system?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained due course of time?

Generally, project will not be rejected simply because of operational feasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations. This feasibility study is carried out by a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and are skilled in skilled analysis and design process.

3.5.3 ECONOMICAL FEASIBILTY

Economical Feasibility is the most frequently used method for evaluating the effectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justifications or alterations in the proposed system will have to be made if it is having a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle

3.5.4 BEHAVIORAL FEASIBILITY

It is behavioral feasibility because our system can be accessed and used even for a normal user. The user can easily understand menus used on the system. The proposed system is planned in such a way that every user can easily operate the system without difficulty. Hence it is behaviorally feasible. Behavioral Feasibility mainly includes how strong the reaction of staff will be towards the development of the proposed system that involves computer’s use in their daily work. So resistant to change is identified

CHAPTER 4

OPERATING ENVIRONMENT

4.1 HARDWARE REQUIREMENT

1. Processor: Dual Core 1.60 GHz or higher

2. Hard disk: 500 GB

3. RAM: 4GB

4. Monitor: 17" Color Monitor

5. Mouse: Microsoft

6. Keyboard: Microsoft multimedia keyboard

4.2 SOFTWARE REQUIREMENT

1. Operating System: Windows 8 or higher

2. Framework: Microsoft .Net Framework

3. Environment: Visual Studio 2012

4. Language: Python, Open CV, and Tensor Flow and Yolov3.

5. Documentation: Microsoft Word 2010 or higher

4.3 TOOLS AND PLATFORMS

4.3.1 VISUAL STUDIO

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silver light. It can produce both native code and managed code. A fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services. Visual Studio does not support any programming language, solution or tool intrinsically; instead, it allows the plugging of functionality coded as a VS Package. When installed, the functionality is available as a Service. The IDE provides three services: SVs Solution, which provides the ability to enumerate projects and solutions; SVs UI Shell, which provides windowing and UI functionality (including tabs, toolbars, and tool windows); and SVs Shell, which deals with registration of VS Packages. In addition, the IDE is also responsible for coordinating and enabling communication between services. All editors, designers, project types and other tools are implemented as VS Packages. Visual Studio uses COM to access the VS Packages. The Visual Studio SDK also includes the Managed Package Framework (MPF), which is a set of managed wrappers around the COM-interfaces that allow the Packages to be written in any CLI compliant language. However, MPF does not provide all the functionality exposed by the Visual Studio COM interfaces. The services can then be consumed for creation of other packages, which add functionality to the Visual Studio IDE.

4.3.2 PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

4.3.3 DEEP LEARNING

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It’s achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

4.3.4 CNN

In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

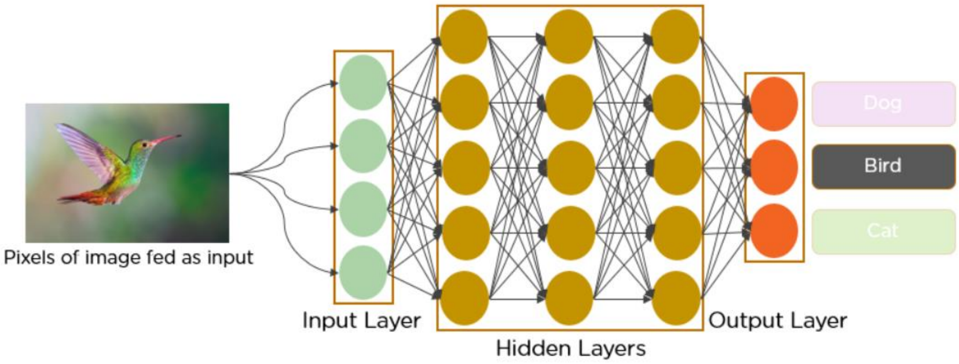


Figure 1: CNN

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

4.3.5 OPEN CV

Open CV (*Open Source Computer Vision Library*) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

4.3.6 YOLO V3

YOLO is a Convolutional Neural Network (CNN) for performing object detection in real-time. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy.

It allows the model to look at the whole image at test time, so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms "score" regions based on their similarities to predefined classes.

High-scoring regions are noted as positive detections of whatever class they most closely identify with. For example, in a live feed of traffic, YOLO can be used to detect different kinds of vehicles depending on which regions of the video score highly in comparison to predefined classes of vehicles.

4.3.6.1 Working of YOLO v3

YOLO is a Convolutional Neural Network (CNN) for performing object detection in real-time. CNNs are classifier-based systems that can process input images as structured arrays of data and identify patterns between them (view image below). YOLO has the advantage of being much faster than other networks and still maintains accuracy.

It allows the model to look at the whole image at test time, so its predictions are informed by the global context in the image. YOLO and other convolutional neural network algorithms "score" regions based on their similarities to predefined classes.

High-scoring regions are noted as positive detections of whatever class they most closely identify with. For example, in a live feed of traffic, YOLO can be used to detect different kinds of vehicles depending on which regions of the video score highly in comparison to predefined classes of vehicles.

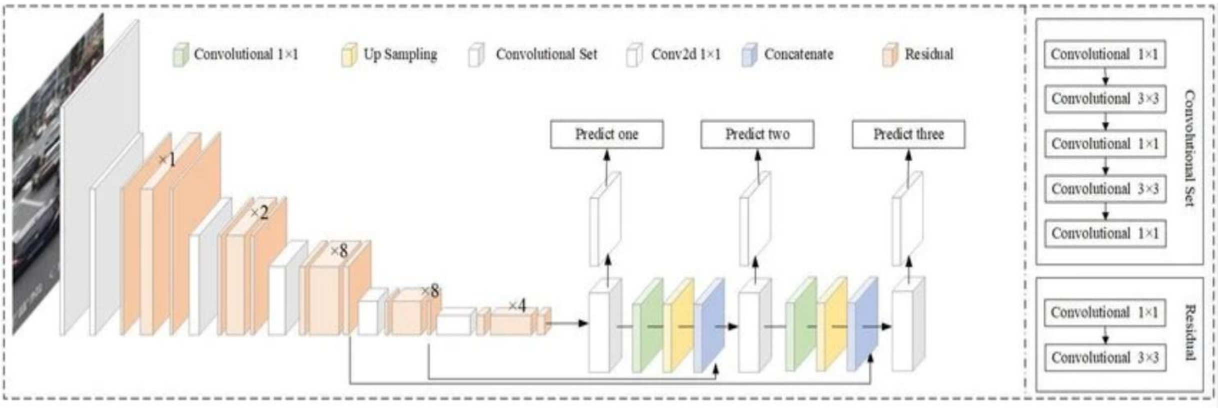


Figure 2:YOLO Working

4.3.6.2 Proposed Architecture Description:

YOLOv3 uses a new feature extractor known as Darknet-53 which, as the name suggests, uses 53 convolutional layers while the overall algorithm consists of 75 convolutional layers and 31 other layers making it a total of 106 layers. Pooling layers have been removed from the architecture and replaced by another convolutional layer with stride ‘2’, for the purpose of down-sampling. This key change has been made to prevent the loss of features during the process of pooling.

4.3.6.3 Proposed Architecture Characteristics:

● YOLO v3 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.

● In YOLO v3, the detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network.

● The shape of detection kernel is 1 x 1 x (B x (5 + C)). Here B is the number of bounding boxes a cell on the feature map can predict, '5' is for the 4 bounding box attributes and one object confidence and C is the no. of classes.

● YOLO v3 uses binary cross-entropy for calculating the classification loss for each label while object confidence and class predictions are predicted through logistic regression

4.3.6.4 Block Diagram

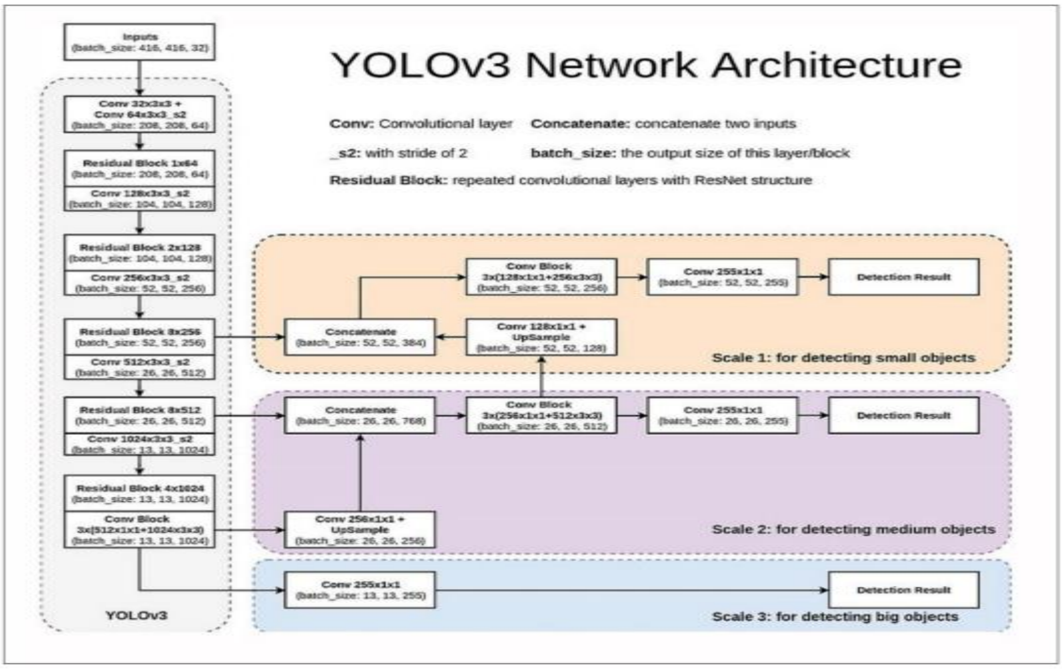

YOLOv3 Network Architecture
Conv: Convolutional layer
Concatenate: concatenate two inputs
_s2: with stride of 2
batch_size: the output size of this layer/block
Residual Block: repeated convolutional layers with ResNet structure
Inputs (batch_size: 416, 416, 32)
Conv 32x3x3 + Conv 64x3x3_s2 (batch_size: 208, 208, 64)
Residual Block 1x64 (batch_size: 208, 208, 64)
Conv 128x3x3_s2 (batch_size: 104, 104, 128)
Residual Block 2x128 (batch_size: 104, 104, 128)
Conv 256x3x3_s2 (batch_size: 52, 52, 256)
Residual Block 8x256 (batch_size: 52, 52, 256)
Conv 512x3x3_s2 (batch_size: 26, 26, 512)
Residual Block 8x512 (batch_size: 26, 26, 512)
Conv 1024x3x3_s2 (batch_size: 13, 13, 1024)
Residual Block 4x1024 (batch_size: 13, 13, 1024)
Conv Block 3x(512x1x1+1024x3x3) (batch_size: 13, 13, 1024)
YOLOv3
Conv Block 3x(128x1x1+256x3x3) (batch_size: 52, 52, 256)
Conv 255x1x1 (batch_size: 52, 52, 255)
Detection Result
Concatenate (batch_size: 52, 52, 384)
Conv 128x1x1 + UpSample (batch_size: 52, 52, 128)
Scale 1: for detecting small objects
Concatenate (batch_size: 26, 26, 768)
Conv Block 3x(256x1x1+512x3x3) (batch_size: 26, 26, 512)
Conv 255x1x1 (batch_size: 26, 26, 255)
Detection Result
Conv 256x1x1 + UpSample (batch_size: 26, 26, 256)
Scale 2: for detecting medium objects
Conv 255x1x1 (batch_size: 13, 13, 255)
Detection Result
Scale 3: for detecting big objects

Figure3:YOLO Block diagram

4.3.6.5 Layers Details

● YOLO makes use of only convolutional layers, making it a fully convolutional network (FCN)

● In YOLOv3 a deeper architecture of feature extractor called Darknet-53 is used.

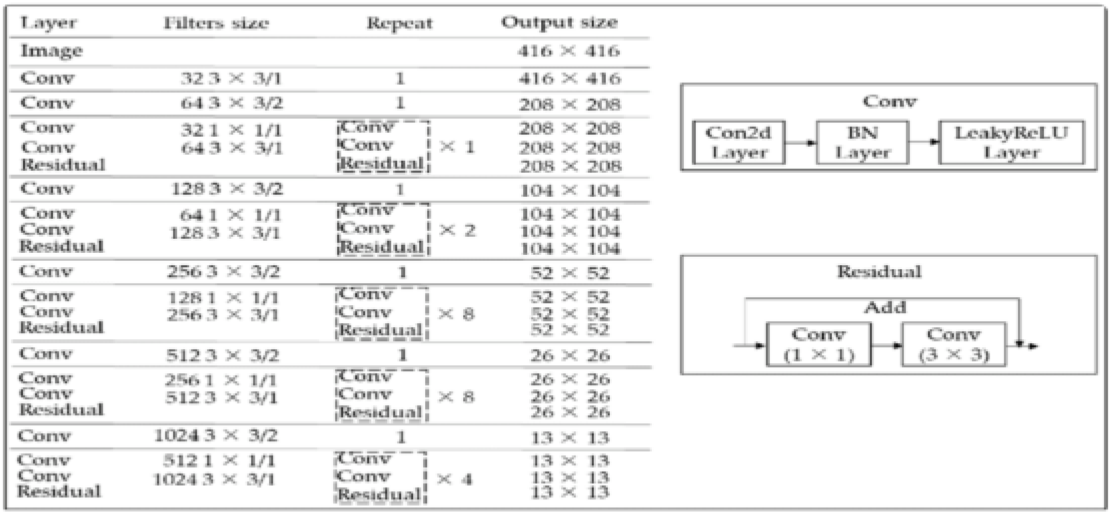
4.3.6.6 Convolution layers in YOLOv3

● It contains 53 convolutional layers which have been, each followed by batch normalization layer and Leaky ReLU activation.

● Convolution layer is used to convolve multiple filters on the images and produces multiple feature maps

● No form of pooling is used and a convolutional layer with stride 2 is used to downsample the feature maps.

● It helps in preventing loss of low-level features often attributed to pooling.



Layer	Filters size	Repeat	Output size
Image			416 × 416
Conv	32 3 × 3/1	1	416 × 416
Conv	64 3 × 3/2	1	208 × 208
Conv Conv Residual	32 1 × 1/1 64 3 × 3/1	Conv Conv × 1 Residual	208 × 208 208 × 208 208 × 208
Conv	128 3 × 3/2	1	104 × 104
Conv Conv Residual	64 1 × 1/1 128 3 × 3/1	Conv Conv × 2 Residual	104 × 104 104 × 104 104 × 104
Conv	256 3 × 3/2	1	52 × 52
Conv Conv Residual	128 1 × 1/1 256 3 × 3/1	Conv Conv × 8 Residual	52 × 52 52 × 52 52 × 52
Conv	512 3 × 3/2	1	26 × 26
Conv Conv Residual	256 1 × 1/1 512 3 × 3/1	Conv Conv × 8 Residual	26 × 26 26 × 26 26 × 26
Conv	1024 3 × 3/2	1	13 × 13
Conv Conv Residual	512 1 × 1/1 1024 3 × 3/1	Conv Conv × 4 Residual	13 × 13 13 × 13 13 × 13

Figure 4:Dimension Table

4.3.6.7 How YOLO detects the objects in a given image.

First, it divides the image into a 13×13 grid of cells. The size of these 169 cells vary depending on the size of the input. For a 416×416 input size that we used in our experiments, the cell size was 32×32. Each cell is then responsible for predicting a number of boxes in the image.

For each bounding box, the network also predicts the confidence that the bounding box actually encloses an object, and the probability of the enclosed object being a particular class.

Most of these bounding boxes are eliminated because their confidence is low or because they are enclosing the same object as another bounding box with very high confidence score. This technique is called **non-maximum suppression**.

The authors of YOLOv3 Joseph Redmon and Ali Farhadi, have made YOLOv3 faster and more accurate than their previous work YOLOv2. YOLOv3 handles multiple scales better. They have also improved the network by making it bigger and taking it towards residual networks by adding shortcut connections.

4.3.7 TENSOR FLOW

Tensor Flow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

Tensor Flow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of Tensor Flow, named Tensor Flow 2.0, in September 2019.

Tensor Flow can be used in a wide variety of programming languages, most notably Python, as well as Java script, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.

4.3.8 COCO DATASET

COCO stands for Common Objects in Context, as the image dataset was created with the goal of advancing image recognition. The COCO dataset contains challenging, high-quality visual datasets for computer vision, mostly state-of-the-art neural networks.

For example, COCO is often used to benchmark algorithms to compare the performance of real-time object detection. The format of the COCO dataset is automatically interpreted by advanced neural network libraries.

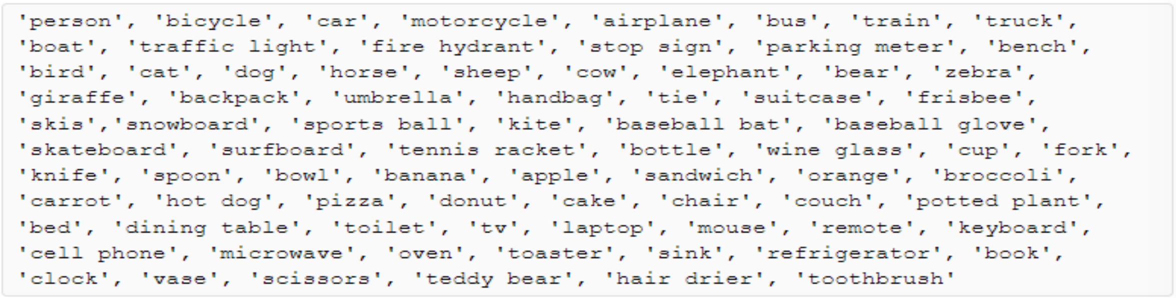
4.3.8.1 Features of the COCO dataset

- Object segmentation with detailed instance annotations
- Recognition in context
- Superpixel stuff segmentation
- Over 200’000 images of the total 330’000 images are labelled
- 1.5 Mio object instances
- 80 object categories, the “COCO classes”, which include “things” for which individual instances may be easily labelled (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 keypoints, popularly used for Pose Estimation

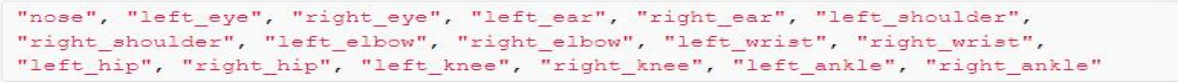
4.3.8.2 List of the COCO Object Classes and KeyPoints

The COCO dataset classes for object detection and tracking include the following pre-trained 80 objects:



```
'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck',
'boat', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench',
'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra',
'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
'skis','snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove',
'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork',
'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli',
'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant',
'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard',
'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book',
'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
```

The COCO keypoints include 17 different pre-trained keypoints (classes) that are annotated with three values (x,y,v). The x and y values mark the coordinates, and v indicates the visibility of the key point (visible, not visible).



```
"nose", "left_eye", "right_eye", "left_ear", "right_ear", "left_shoulder",
"right_shoulder", "left_elbow", "right_elbow", "left_wrist", "right_wrist",
"left_hip", "right_hip", "left_knee", "right_knee", "left_ankle", "right_ankle"
```

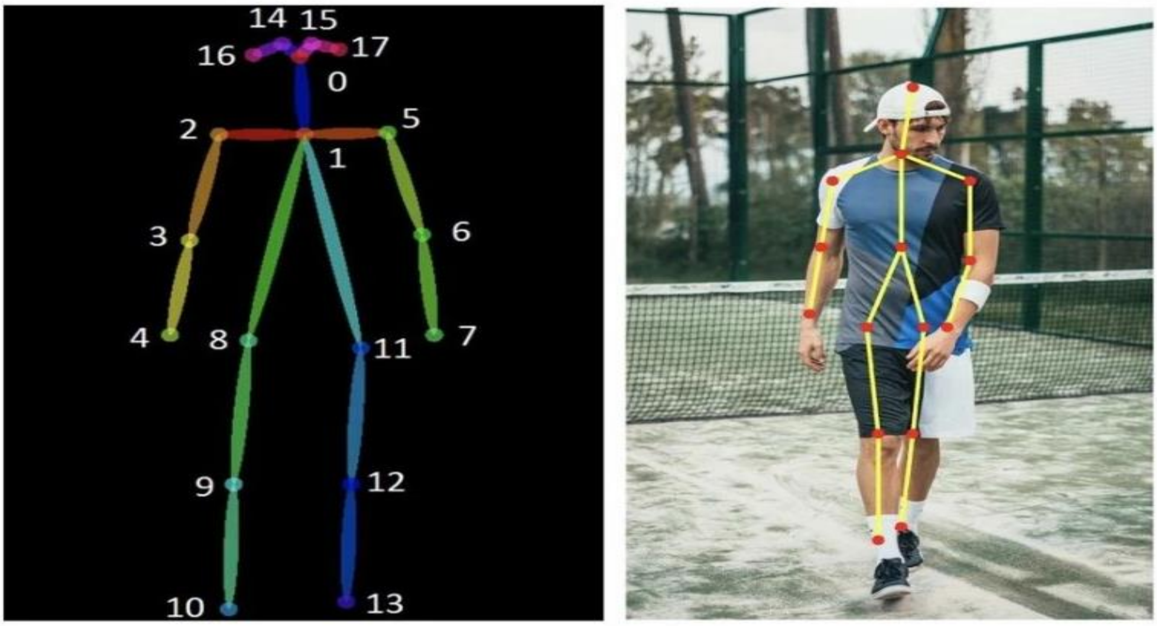


Figure 5: COCO Keypoints

4.3.8.3 Annotated COCO images

The large dataset comprises annotated photos of everyday scenes of common objects in their natural context. Those objects are labeled using pre-defined classes such as “chair” or “banana”. The process of labeling, also named image annotation and is a very popular technique in computer vision.

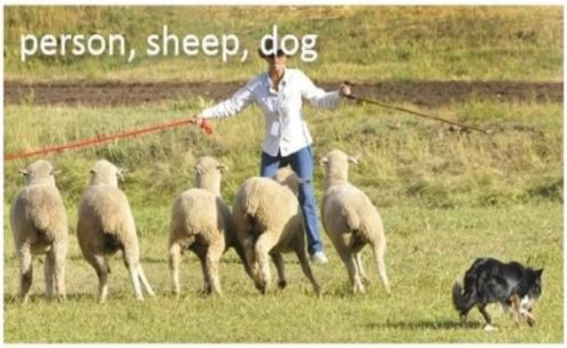


Figure 6: Image Classification

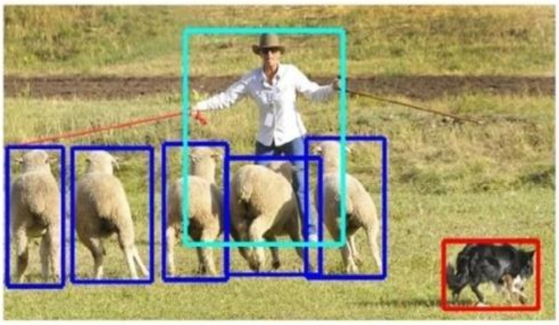


Figure 7 Object Localization

4.3.8.4 What makes COCO UNIQUE

With COCO, Microsoft introduced a visual dataset that contains a massive number of photos depicting common objects in complex everyday scenes. This sets COCO apart from other object recognition datasets that may be specifically specific sectors of artificial intelligence. Such sectors include image classification, object bounding box localization, or semantic pixel-level segmentation.

Meanwhile, the annotations of COCO are mainly focused on the segmentation of multiple, individual object instances. This broader focus allows COCO to be used in more instances than other popular datasets like CIFAR-10 and CIFAR-100. However, compared to the OID dataset, COCO does not stand out too much and in most cases, both could be used.

Data. With 2.5 million labelled instances in 328k images, COCO is a very large and expansive dataset that allows many uses. However, this amount does not compare to Google's OID, which contains a whopping 9 million annotated images.

Google's 9 million annotated images were manually annotated, while OID discloses that it generated object bounding boxes and segmentation masks using automated and computerized methods. Both COCO and OID have not disclosed bounding box accuracy, so it remains up to the user whether they assume automated bounding boxes would be more precise than manually made ones.

4.3.8.5 Feature List & Class of the dataset

a) Class of the data set: Person in COCO Dataset.

b) Properties of image :

“id": int,

"width": int,

"height": int,

"file_name": str,

"license": int,

"flickr_url": str,

"coco_url": str,

"date_captured": datetime

4.3.9 IMAGE PROCESSING

Image processing methods used here:

i) Mean subtraction

ii) Scaling

iii) Optionally channel swapping

i)Mean subtraction

Mean subtraction is used to help combat illumination changes in the inputimages in our dataset. We can therefore view mean subtraction as atechnique used to aid our

Convolutional Neural Networks. Before we even begin training our deep neural network, we first computethe average pixel intensity across all images in the training set for each ofthe Red, Green, and Blue channels. This implies that we end up with three variables:

\mu_R, \mu_G, and \mu_B

Typically the resulting values are a 3-tuple consisting of the mean of theRed, Green, and Blue channels, respectively.

ii) Scaling

However, in some cases the mean Red, Green, and Blue values may becomputed channel-wise rather than pixel-wise, resulting in an MxNmatrix. In this case the MxN matrix for each channel is then subtractedfrom the input image during training/testing. Both methods are perfectly valid forms of mean subtraction; however, wetend to see the pixel-wise version used more often, especially for larger datasets. When we are ready to pass an image through our network (whether fortraining or testing), we subtract the mean, \mu, from each input channelof the input image:

R = R - \mu_R

G = G - \mu_G

B = B - \mu_B

We may also have a scaling factor, \sigma, which adds in a normalization:

R = (R - \mu_R) / \sigma

G = (G - \mu_G) / \sigma

B = (B - \mu_B) / \sigma

The value of \sigma may be the standard deviation across the training set(thereby turning the preprocessing step into a standard score/z-score).However,\sigma may also be manually set (versus calculated) to scale theinput image space into a particular range

CHAPTER 5

DESIGN

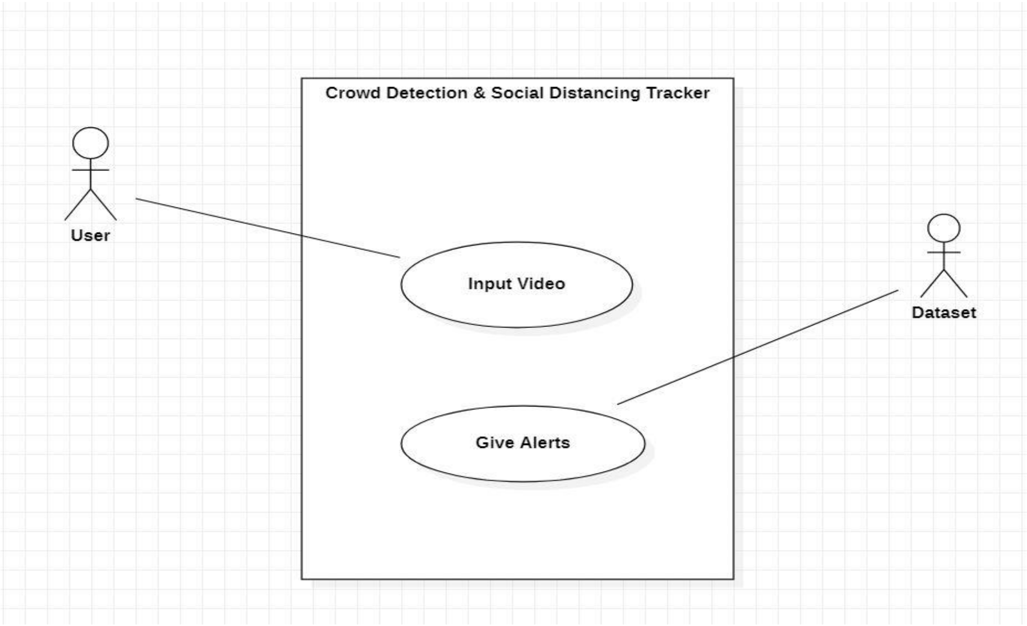
5.1 SYSTEM DESIGN

System design is the process of defining the architecture, modules, and data for a system to satisfy specified requirements. It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate. The purpose of design phase is to plan a solution for problem specified by the requirements .System design aims to identify the modules that should be in the system, the specification of those modules and how they interact with each other to produce the result. The goal of the design process is to produce a model for or representation of a system can be used later to build .The produced model is called design of the system.

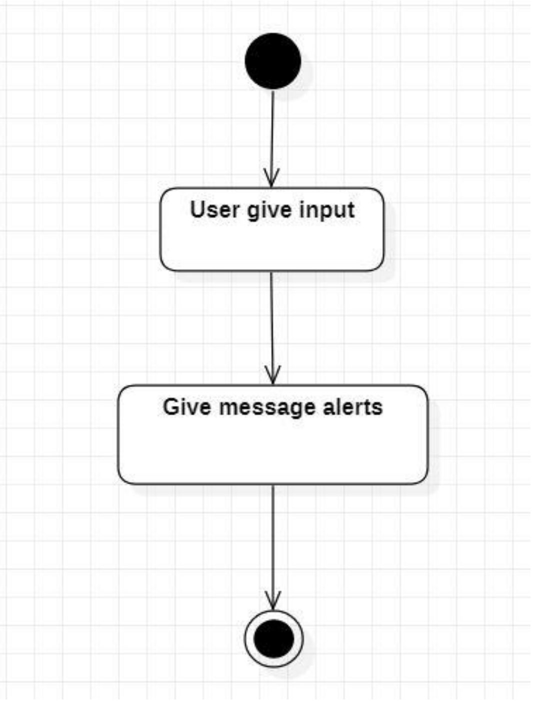
5.2 PROGRAM DESIGN

- As a user, I want to explore the dataset, so that I can use it for object detection.
- As a user, I want to learn CNN architecture, so that I can explore the layers and description
- As a user, I want to learn dimention table, so that I can know about the filter size, output size etc Of the image
- As a user, I want to learn the YOLO architecture, so that object detection becomes easier
- As a user, I want to implement the algorithm, So that I can obtain the person class from the set.
- As a user, I want to obtain, the partitioned dataset, so that I can get the test images and train images
- As a user, I want to train the model using the Dataset, So that I can start testing
- As a user, I want to implement the model, So That I can detect object.
- As a user, I want to detect human from recorded video or live streaming video, So that I can estimate the human count
- As a user, I want to send alert messages to Concerned, S o that they can identify if any violation is there

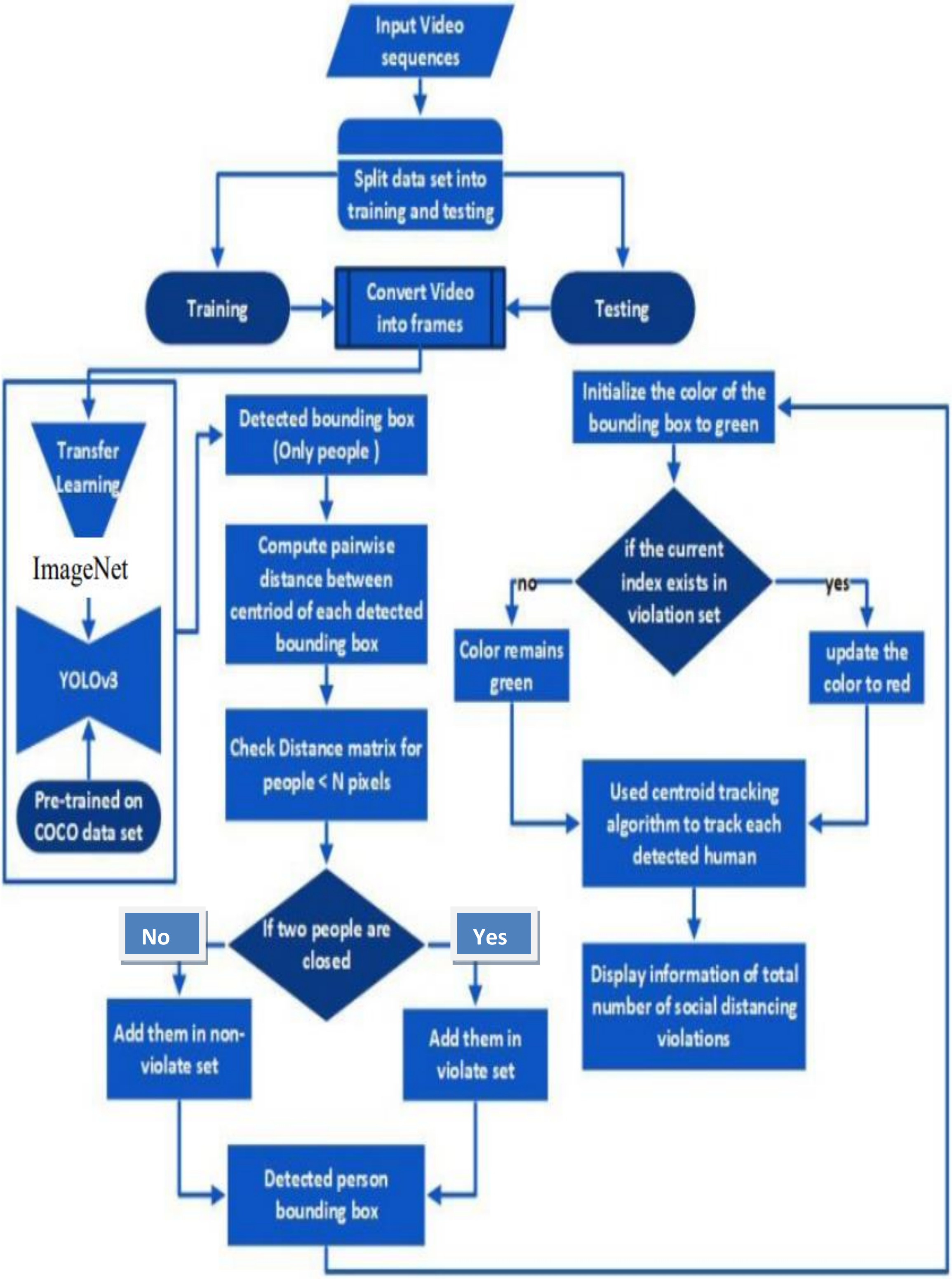
5.3 USECASE DIAGRAM



5.4 **ACTIVITY DIAGRAM**



5.5 PROPOSED PROJECT PIPELINE


Input Video sequences
Split data set into training and testing
Training
Convert Video into frames
Testing
Transfer Learning
ImageNet
YOLOv3
Pre-trained on COCO data set
Detected bounding box (Only people)
Compute pairwise distance between centriod of each detected bounding box
Check Distance matrix for people < N pixels
If two people are closed
No
Yes
Add them in non-violate set
Add them in violate set
Detected person bounding box
Initialize the color of the bounding box to green
if the current index exists in violation set
no
yes
Color remains green
update the color to red
Used centroid tracking algorithm to track each detected human
Display information of total number of social distancing violations

5.6 INPUT DESIGN

Input is the process of converting user inputs computer-based format. The project requires a set of information from the user to prepare a report. In the order, when organized input data are needed. Input data is collected and organized into groups of similar data. The goal behind designing input data is to make the data entry easy and make it free from logical error. So, the input screens in the system should be really flexible and faster to use. The user need not to input any data manually, we are using the video data directly from the web camera.

Objectives: -

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understandable
- To make clutter free screens
- The prevention of irrelevant data entry
- To make a user-friendly input screen

Here in our system, ‘Real-Time Social Distancing Detector’, input screens ensure the reliability and accuracy of the system. The input design determines whether the user can interact directly with the computer. With input design, we can say that it is more user friendly as compared to the existing manual system containing paper operations

5.7 OUTPUT DESIGN

Outputs are the most important direct source of information to the user and to the management. Efficient and eligible output design should improve the system’s relationship with the user and help in decision making, Output design generally deals with the results generated by the system i.e., reports. These reports can be generated from stored or calculated values. Reports are displayed either as screen window preview. Most end users will not actually operate the information system or enter data through workstation, but they will use the output from the system. The system provides a comprehensive output screen with graphical representations for user to better understand the provided data

CHAPTER 6

FUNCTIONAL AND NON FUNCTIONAL REQUIREMENT

6.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally functional requirements are expressed in the form of "system must do requirement".

6.2 NON-FUNCTIONAL REQUIREMENTS

A non functional requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors .Some of the non-functional requirements are mentioned below.

Usability: The system shall have a clean interface with only needed features, clear terminology and tools tips where necessary. Warnings or alerts shall be specified in clear way.

Efficiency: The system shall respond to different searches being conducted like searching particular product, search quantity, etc. in a very fast way.

Interoperability: The system shall be able to interact with other systems. The system should able to be supported at least one software which has a relationship with Payment process

Portability: The system shall be independent of the specific technological platform used to implement it.

Reliability: Reliability defined as a measure of the time between failures occurring in system, so that the system shall operate without any failures for a particular period of time.

Availability: Availability measures the percentage of time the system is in its operational state so that the system be available for use 24 hours per day and 365 days per year.

CHAPTER 7

TESTING

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to Do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases.

Testing Objectives Include:

- Testing is a process of executing a program with the intent offending an error.

- A good test case is one that has a probability of finding an as yet

undiscovered error.

Testing Principles:

- All tests should be traceable to end user requirements

- Tests should be planned long before testing begin

- Testing should begin on a small scale and progress towards testing in large

- Exhaustive testing is not possible.

- To be most effective testing should be conducted by an independent third party.

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department.If the implementation is not carefully planned and controlled it can cause chaos and confusion.

7.1TESTING PROCESS

Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program. Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding,Testing present interesting anomaly for the software engineer.

7.1.1 UNIT TESTING

This is the first of testing.In this different modules are tested against the specification produces during the design of the modules. It refers to the verification of single program module in an isolated environment. Unit testing focuses on the modules independently of one another to locate errors.

In our project we test each module and each forms individually. Each forms may tested using appropriate values. The input screens need to be designed very carefully and logically.While entering data in the input forms, proper validation checks are done and messages will be generated by the system if incorrect data has been entered.

7.1.2 ALPHA TESTING

It is the most common type of testing used in the Software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user. Alpha testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing. Alpha testing is conducted at the developer's site. In-house virtual user environment can be created for this type of testing.

7.1.3 ACCEPTANCE TESTING

An acceptance test is performed by the client and veries whether the end to end the ow of the system is as per the business requirements or not and if it is as per the needs of the end user. Client accepts the software only when all the features and functionalities work as expected. It is the last phase of the testing, after which the software goes into production. This is also called as User Acceptance Testing (UAT).

7.1.4 BETA TESTING

Beta Testing is a formal type of software testing which is carried out by the customer. It is performed in Real Environment before releasing the product to the market for the actual end users. Beta testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta testing is successful when the customer accepts the software. Usually, this testing is typically done by endusers or others. It is the testing done before releasing an application forcommercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area. So end user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

7.2 VALIDATION CHECKS

As a web application developer, form validation is a crucial part of your work, and it should not be underrated as it could lead to security flaws in your application. You should consider it a must if you're striving to provide a professional end user experience.

- Basic form validation

- Custom error messages

7.3 TEST CASES

TEST CASE ID	TEST CASE NAME	TEST CASE STEP	EXPECTED RESULT	STATUS	DEFECTS
1	Opening the Application	Install and opening the application on pc	Camera ON Real time monitoring Start	PASS	Nill
2	Detecting humans	By using real time camera streaming humans are detected	Humans are detected by Bounding boxes	PASS	Nill
3	Calculating Social distance	By using the bounding box and pixel distance of the video frames distance calculated	Distance between the Humans are calculated	PASS	Nill
4	Give Alert message	By checking the threshold limit valuing using the distance alert will give	Alert message will display	PASS	Nill

CHAPTER 8

RESULTS AND DISCUSSION

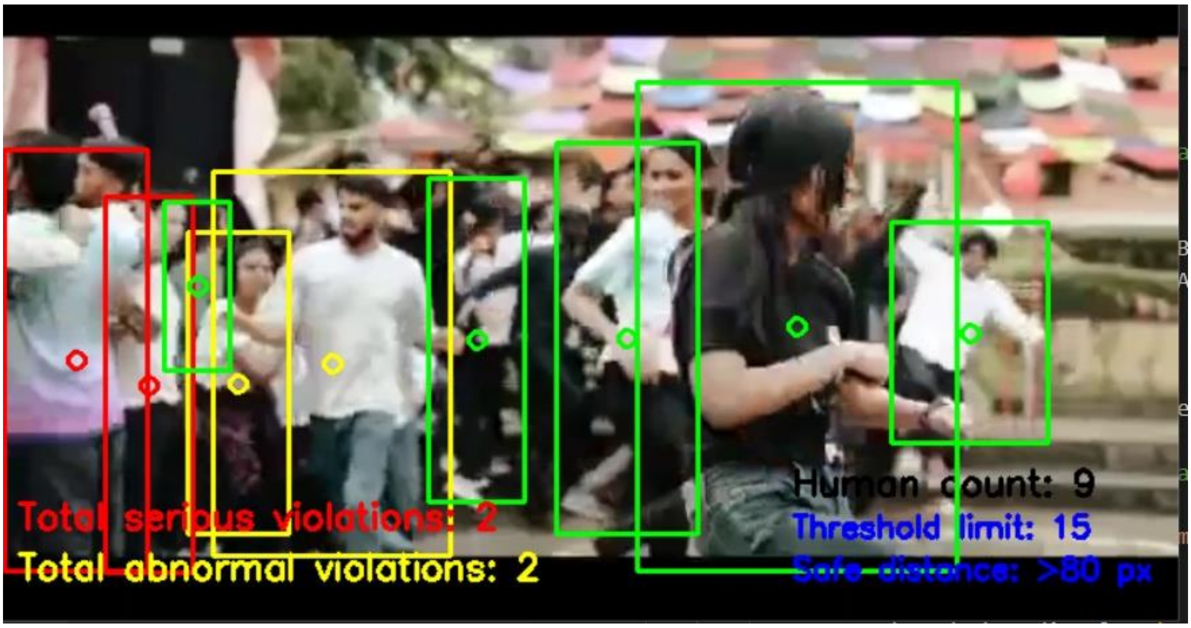
8.1 RESULTS(Salient Features)

The objective of the proposed system is to identify whether a person on image/video stream is maintaining social distance or not with the help of computer vision, machine learning and deep learning. We make this possible with the help of Python libraries like OpenCV, TensorFlow, Kerasetc.. Then we can use the model to classify images. At first, we have to detect the human from the frame using haar cascade human detection algorithm.

User interface is designed such that they are very user friendly
and the user is not required to input data manually,

- Not much training required.
- Easy analysis of data and statistical view.
- There are no needs of experts.
- The new system is more user friendly.
- This system is much faster and efficient than the old system
- Lower points failure than the old system.
- It is cheaper than the existing system
- No need for expensive equipment.
- System provides various information’s through report quickly and accurately in easily understandable formats.
- It aimed to be paperless software. There is no unnecessary printing of particular documents.
- Taking into the speed of computer access, large data in less time and facilities provided by the access.

8.2 SCREENSHOTS



Total serious violations: 2

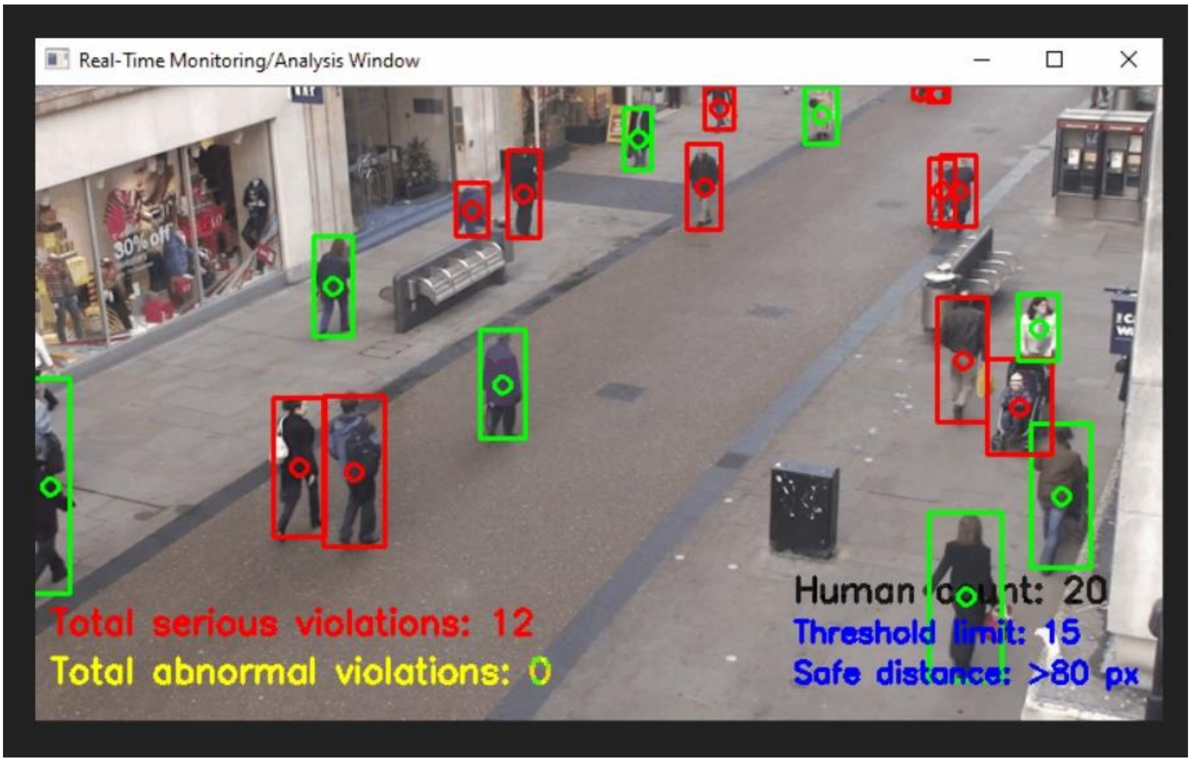
Total abnormal violations: 2

Human count: 9

Threshold limit: 15

Safe distance: >80 px

Figure 8:Output Screen



Total serious violations: 12

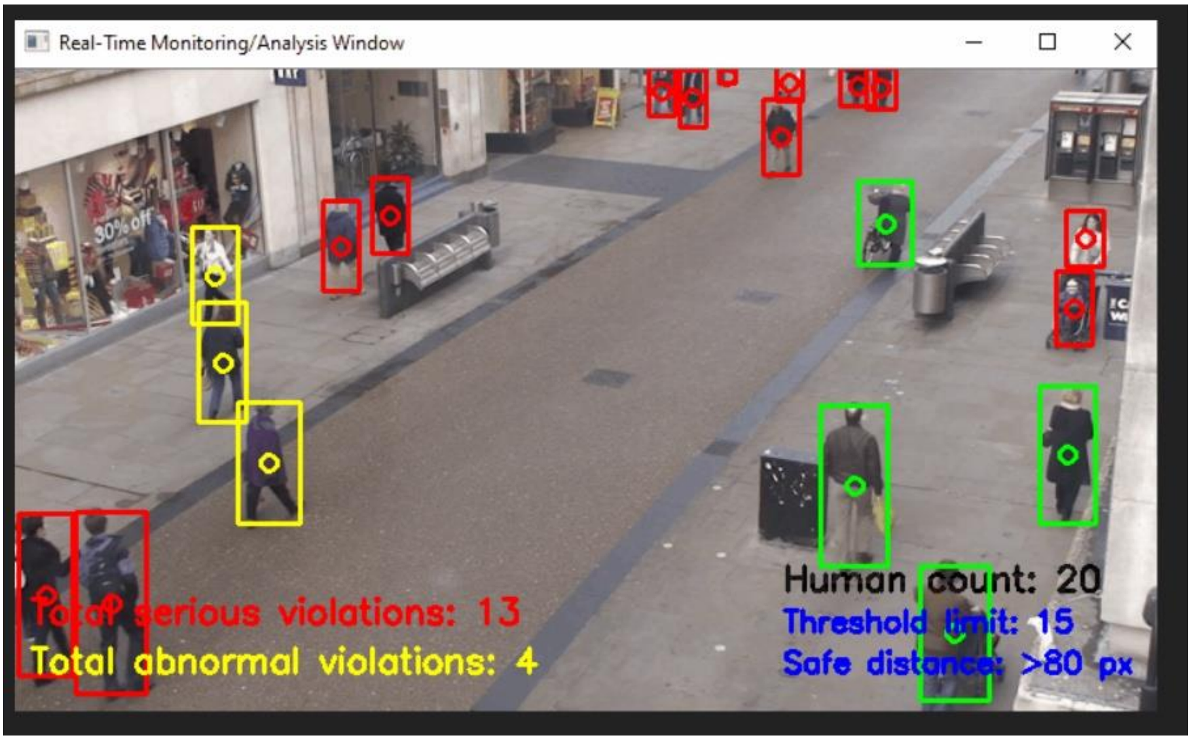
Total abnormal violations: 0

Human count: 20

Threshold limit: 15

Safe distance: >80 px

Figure 9:Output Screen



Total serious violations: 13

Total abnormal violations: 4

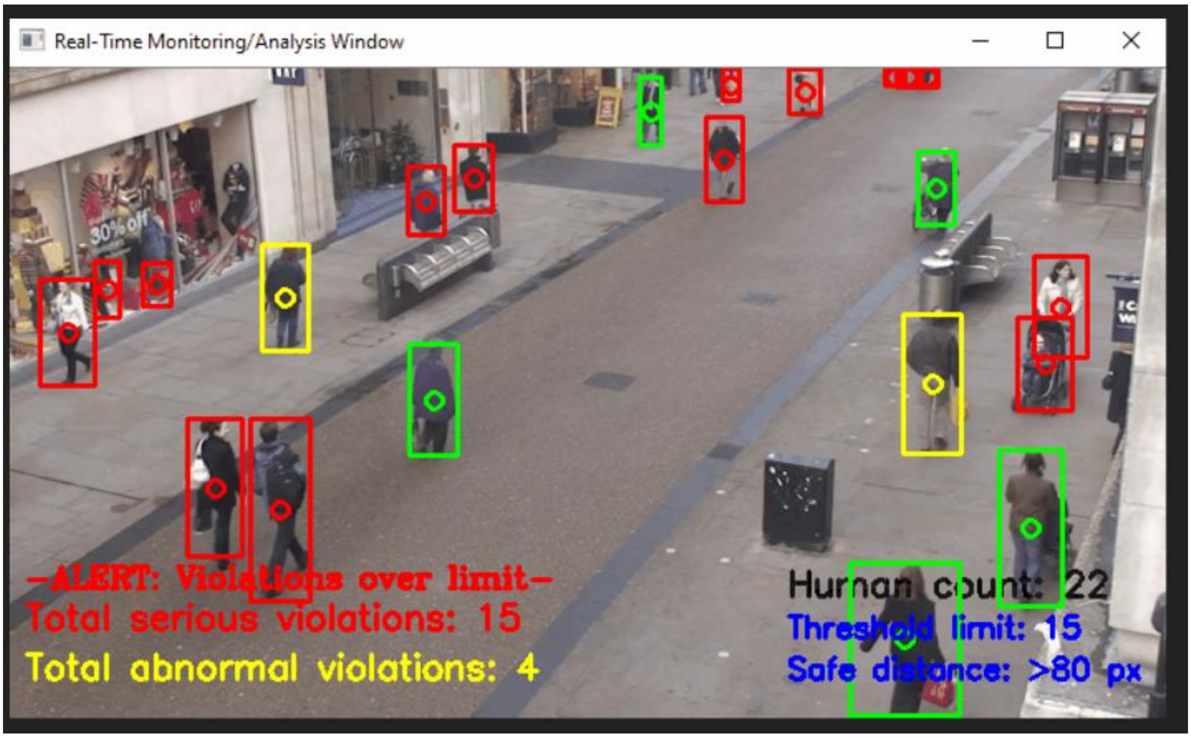
Human count: 20

Threshold limit: 15

Safe distance: >80 px

Figure10:Output Screen

Alert Message when count exceeds threshold value



-ALERT: Violations over limit-

Total serious violations: 15

Total abnormal violations: 4

Human count: 22

Threshold limit: 15

Safe distance: >80 px

Figure11:Alert Message

CHAPTER 9

CONCLUSION

9.1 SYSTEM IMPLEMENTATION

After the system has been tested, the implementation type or the change overtechnique from the existing system to the new system is a step-by-step process. In the system at first only a module of the system is implemented and checked for suitability and efficiency. When the end user related to the particular module is satisfied with the performance, the next step of implementation is preceded.

Backups are necessary since any time unexpected events may happen. And so during the program execution, the records are stored in the workspace. This helps to recover the original status of the records from any accidental updating or intentional deletion of records.

An Implementation plan is a management tool for a specific policy measure, or package of measures, designed to assist agencies to manage and monitor implementation effectively. Implementation plans are intended to be scalable and flexible; reflecting the degree of urgency, innovation, complexity and or sensitivity associated with the particular policy measure. Agencies are expected to exercise judgment in this area; however, the level of detail should be sufficient to enable the agency to effectively manage the implementation of a policy measure. At a minimum, plans should reflect the standards outlined in the Guide to Preparing Implementation Plans.

The implementation stage involves following tasks:

➢ Careful planning

➢ Investigation of system and constraints

➢ Design of method to achieve the changeover phase

Here blobFromImage creates a 4-dimensional blob from image. Optionally resizes and crops image from centre, subtract mean values, scales values by scalefactor, swap Blue and Red channels

9.2 CONCLUSION

Deep learning helps computers to derive meaningful links from a plethora of data and make sense of unstructured data. Here, the mathematical algorithms are combined with a lot of data and strong hardware to get qualified information. With this method, information from digital data can be automatically extracted, classified and analyzed.

Although deep learning has been around for several years, the trend has only really picked up in the last three to four years. The reason for this was among other things better hardware resources, more sophisticated algorithms and optimized neural networks. Deep learning is not a new approach but a development of the older approach of artificial neural networks.

The proposed method presented in this project is mainly for crowd surveillance and security maintenance. This system can be utilized for events, private property or places that have massive streams of people, entering and leaving the area, which necessitates vigilant tracking and identification of all individuals within the premises. With deep learning, particularly CNN, the model is able to train and learn to identify human beings using the database customized.

9.3 FUTURE SCOPE

In future we can expect the modified version of ”CROWD DETECTION AND SOCIAL DISTANCE TRACKING”. The system is very flexible for further up gradiation with additional requirement. Since it uses COCO dataset which contains 80 classes it can be used to any kind of objects in addition to humans. So it can be used at any place where crowd management is necessary. And also another advantage is that it can be used in traffic signals where signal timing can be adjusted based on the number of vehicles present at the moment.

In addition to that here we have used alert messages and instead of that in future it can be modified to voice alerts such as super markets etc.

The information can be updated to the latest coming versions, there are also possibilities for enhancing and further developing the project with the latest information and needs of this Crowd Detection and Safe Distance tracking.

REFERENCES

[1]People Detection System Using YOLOv3 Algorithm" (2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE2020), 21– 22 August 2020, Penang, Malaysia

[2]https://viso.ai/deep-learning/yolov3-overview/
[3]https://www.zenesys.com/blog/python-coding-standards-best-practices-for-python-code-quality

[4]https://www.mathworks.com/help/vision/ug/object-detection-using-yolo-v3-deep-learning.html
[5]https://viso.ai/computer-vision/coco-dataset/
[6]https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
[7]https://learnopencv.com/deep-learning-based-object-detection-using-yolov3-with-opencv-python-c/
[8]https://learnopencv.com/category/application/
[9] https://cocodataset.org/#home

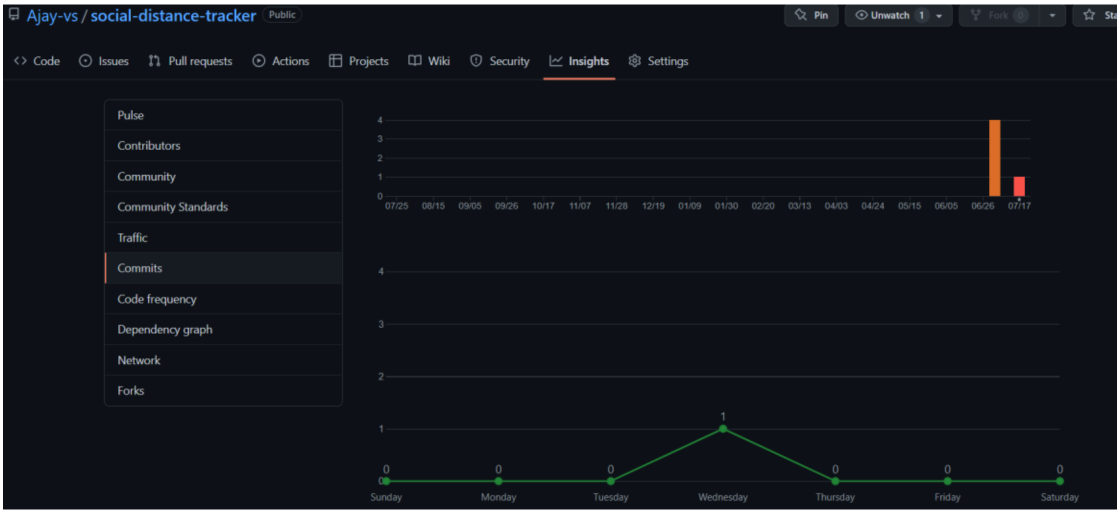
[10]https://towardsdatascience.com/deep-learning-with-python-neural-networks-complete-tutorial-6b53c0b06af0

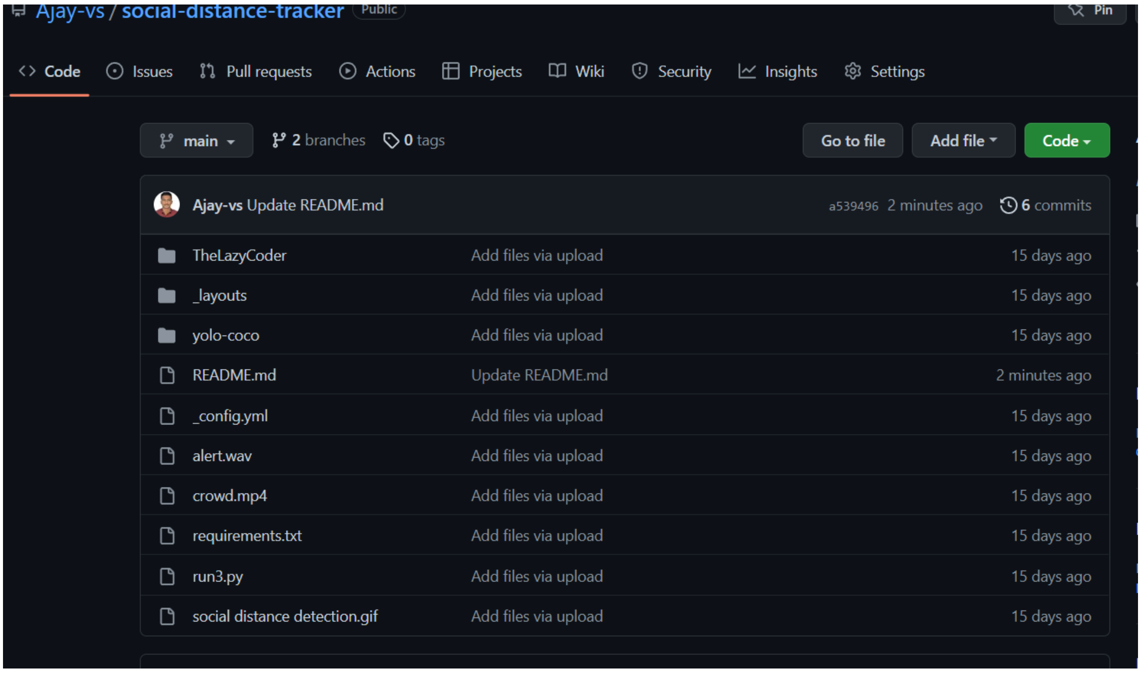
APPENDICES

- List of figures

FigureNo	Name	PageNo
1	CNN	15
2	YOLO working	17
3	YOLO blockdiagram	18
4	Dimension table	19
5	COCO Keypoints	22
6	Image classiffication	22
7	Object Localization	22
8	Output Screen	39
9	Output Screen	39
10	Output Screen	40
11	Alert message	40

GIT HISTORY





Ajay-vs Update README.md a539496 2 minutes ago 6 commits

TheLazyCoder	Add files via upload	15 days ago
_layouts	Add files via upload	15 days ago
yolo-coco	Add files via upload	15 days ago
README.md	Update README.md	2 minutes ago
_config.yml	Add files via upload	15 days ago
alert.wav	Add files via upload	15 days ago
crowd.mp4	Add files via upload	15 days ago
requirements.txt	Add files via upload	15 days ago
run3.py	Add files via upload	15 days ago
social distance detection.gif	Add files via upload	15 days ago