MONITORING SAFE SOCIAL DISTANCING USING DEEP LEARNING

A Project
Submitted in Partial Fulfillment of the Requirements for the
CS 533 - Data Mining/ Big Data Analysis

School of Computing
Southern Illinois University Carbondale
Dec, 2020

Subash Kharel

MS in Computer Science

Southern Illinois University, Carbondale

Suraj Rimal

MS in Computer Science

Southern Illinois University, Carbondale

ROLE DIVISION

SUBASH KHAREL

- Inverse Perspective Manipulation for distance calculation

SURAJ RIMAL

- Deep learning for person detection

ABSTRACT

Deep learning has been used in various areas of computer vision since its first introduction. From image classification to object detection, a lot of research has been done with an aim to achieve great results in various applied fields. In this project we have implemented Yolov5, an algorithm for Deep Learning, to detect person and calculate safe social distancing with an aim to facilitate the spread of COVID19. Yolov5 being a single stage detector is relatively fast and is very applicable in real time application. The downside of Yolov5 is that it cannot perform better with small objects. But, since, we are dealing with person as an object which relatively occupies larger area in an image, we proceeded with Yolov5. The detected bounding boxes are treated with inverse perspective mapping to get the approximate top view of the image and distance is calculated using the coordinates in the top view/bird's eye view.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

With a view of keeping track of people and facilitating criminal chase, China have used 2.58 million cameras covering 15.35 million people [1]. This makes 6 person per camera assigned. The camera are used for tracking peoples faces. This is all possible due to deep learning [2]. Deep learning is the process of learning features in a data with multiple levels of abstraction. Since the proposal of this machine learning technique, it has been used in diverse fields ranging from detecting manufacturing defects to detecting and identifying objects in space which may be time consuming or even impossible with human intelligence.

Year 2020 has been a year of pandemic due to COVID 19. 1.28 million people have died due to this virus and 52.28 million people have suffered from this virus [3]. Having no vaccine detected till date, WHO suggested the use of sanitizer and maintaining safe social distancing to slow down the widespread of virus all over the world.

1.2 AIMS AND OBJECTIVES

The aim of this paper is to implement deep learning technique to detect people in the image or video feed and use those detection to estimate the distance between people in the images. A lot of research have been done in this sector before, but we are exploring this technique once again with the use of new object detection framework i.e. YoloV5 [4]. YoloV5, a state of art object detection methodology, is very powerful and fastest which makes it deploy-able in CCTV cameras.

With a view to monitor the social distancing in public places, this project aims to calculate distances between people after detection. Implementing human detection followed by Inverse Perspective Mapping makes it easier to detect images and calculate the approximate distance between detected human in the image.

1.3 PAPER ORGANIZATION

Chapter II contains few of the similar research that have been done so far. We have defined the methodology involved with proper details and images in chapter III. Chapter IV is for Results and Discussions where we have tried to quantify the accuracy of our experiment with some standard metrics. The paper ends with conclusion containing the results of experiment and suggesting future works that can be done.

CHAPTER 2

RELATED WORKS

I. Hasan et. al found that a general-purpose object detector works fine in direct cross-dataset evaluation compared with state-of-the-art pedestrian detectors. They demonstrated that dataset collected by crawling the internet served to be a useful source of pre-training for pedestrian detection [5].

M. Rezaei, M. Azarmi developed a similar model which is Deep Neural Network based system for person detection.They used CCTV cameras in the crowd for inter-people length estimation. Their suggested model consisted of a YOLOv4 framework for people detection and inverse perspective mapping for accurate people detection and monitoring of social distancing in challenging situations in the different variations of light [6] .

I. Ahmed et. al. aimed to provide a deep learning platform for social distance tracking. For the detection of humans in the video frames, they used the YOLOv3 object detection paradigm. Also, they implemented the transfer learning methods in order to enhance the model accuracy. So, the detection algorithm employs an already trained algorithm.They found out the social distance monitoring using an approximation of physical distance to pixel by applying Euclidean distance [7].

CHAPTER 3

METHODOLOGY

This section involves several steps involved in the research. Each of the steps are sub-classified into multiple smaller steps and are described with proper articulations. Figures and tables are used for demonstration when necessary.

3.1 PEOPLE DETECTION

People detection involves training of deep learning model with images of various people and in various scenario. The detection step has various phases i.e. Data Collection, Data Labelling, Training the model and Testing the model.

3.1.1 Data Collection

Most of the images that we used are from google search. Google image search contains wide variety of images from different sources that makes it easier to accomplish image collection task.

Figure 3.1: Images used in training the model

3.1.2 Data Labelling

Since we are using supervised deep learning method, we need to provide the expected output of the sample data while training the model. In Deep learning for Object detection, bounding boxes are used to label the images. Below are some pictures of how labeling are done in the images. We have used LabelImg [8] for windows to label our training example.

Figure 3.2: Labeling Training Images with Bounding Boxes

3.1.3 Training the model

Data labelling is followed by training the model. Yolov5 was used to train the model for people detection. Training the model involves tuning the hyper-parameters that play a great role in accuracy of detection mechanism. The parameters used for training with their corresponding values are mentioned in table 3.1. The training was ran for 1000 epochs in a machine of following configurations:

- **Operating System**:

  Ubuntu 20.04.1 LTS

- **Processor**:

Intel® Core™ i5-9400 CPU @ 2.90GHz × 6 CPUs

- **GPU**:

  GeForce RTX 2060 SUPER (8 GB)

- **RAM**:

  16 GB DDR4 RAM @ 2666MHz

Hyper-parameters	Value
Initial Learning Rate	0.01
Momentum	0.937
Weight Decay	0.0005
GIoU	0.05
IoU Threshold	0.20
Scale(Augmentation)	0.5

Table 3.1: Table to test captions and labels

3.2 DISTANCE ESTIMATION

3.2.1 Inverse Perspective Mapping

IPM is the process of estimating the top view of an image taken from and angle. OpenCV [9] library provides an easy method for getting the inverse perspective mapping of any image. Using 4 points in image and mapping them to real points in top view, we can get the mapping of complete image to the top view. The figure 4.1 shows the top view of image obtained using this mapping technique.

The four dots in image on both sides represent the reference points used for obtaining

Figure 3.3: Perspective view image and its corresponding top view

top view of the image. This calibration was done once and with that transformation matrix, we proceeded with the calculation of distance. Scale of the map is required in order to get the real distance between the objects. In this project, we used approximate scale to get the distance between objects.

CHAPTER 4

RESULTS AND DISCUSSIONS

The results of the Object Detection algorithm is evaluated using various metrics like Precision, Accuracy and mAP. Below are the graphs displaying the progress of various metrics through different epochs of training.

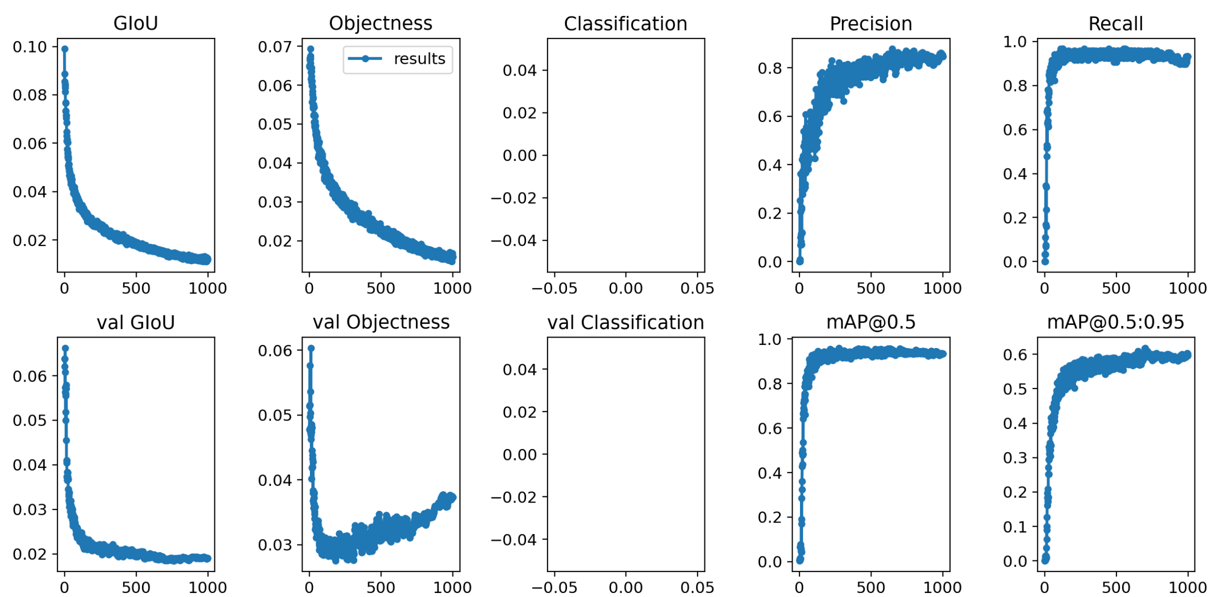

Figure 4.1: Accuracy metrics of training

Running for 1000 epochs, we got the Precision 87% and Recall of 96% with persons dataset. The mAP@0.5 and mAP@0.5-0.95 are 0.96 and 0.59 respectively. The results seemed satisfactory considering the size of dataset we were using.

The demonstration of result can be seen in this video link: `https://www.youtube.com/watch?v=Zp3-rxksG8o`

CHAPTER 5

CONCLUSION

Yolov5 being a single stage model and relatively faster than other 2 stage detectors was able to detect people with relatively good amount of accuracy. Also, using the power of image manipulation, we were able to transform the image into birds eye view and calculate the distance between people approximately.

As a whole, the system was able to detect people in the video feed and able to calculate distance among people. We were also able to warn the people not respecting social distancing using red lines connecting them.

REFERENCES

[1] The Guardian. Big brother is watching. https://www.theguardian.com/cities/2019/dec/02/big-brother-is-watching-chinese-city-with-26m-cameras-is-worlds-most-hea Online; accessed 11 Nov 2020.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[3] Worldometer. Coronavirus update. https://www.worldometers.info/coronavirus/. Online; accessed 11 Nov 2020.

[4] Ultralytics. Yolov5. https://https://github.com/ultralytics. Online; accessed 11 Nov 2020.

[5] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. Pedestrian detection: The elephant in the room. *arXiv preprint arXiv:2003.08799*, 2020.

[6] Mahdi Rezaei and Mohsen Azarmi. Deepsocial: Social distancing monitoring and infection risk assessment in covid-19 pandemic. *medRxiv*, 2020.

[7] Imran Ahmed, Misbah Ahmad, Joel J.P.C. Rodrigues, Gwanggil Jeon, and Sadia Din. A deep learning-based social distance monitoring framework for covid-19. *Sustainable Cities and Society*, page 102571, 2020.

[8] Tzutalin. Labelimg. https://tzutalin.github.io/labelImg/. Online; accessed 11 Nov 2020.

[9] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.