Configuration Manual

MSc Research Project
Data Analytics

Karan veer Singh
Student ID: x20146248

School of Computing
National College of Ireland

Supervisor: Aaloka Anant

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Karan veer Singh
Student ID:	x20146248
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Aaloka Anant
Submission Due Date:	31/01/2022
Project Title:	Configuration Manual
Word Count:	1108
Page Count:	7

Signature:	Karan veer Singh
Date:	30th January 2022

Configuration Manual

Karan veer Singh
x20146248

1 Introduction

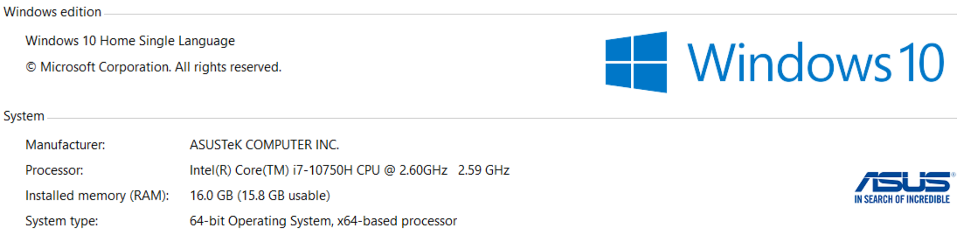
This configuration manual explains the hardware and the software used in project to identify the social distancing from Video Surveillance Cameras using Deep Learning Architectures. The steps mentioned in this manual can be followed to reproduce the results and build a Social Distancing Monitoring model.

2 System Specification

In this section we will discuss Hardware and Software specifications that are used while implementing this project

2.1 Hardware Requirement

The system specification of the system in which all the experiments are implemented are discussed below:



Windows edition

Windows 10 Home Single Language
© Microsoft Corporation. All rights reserved.

System

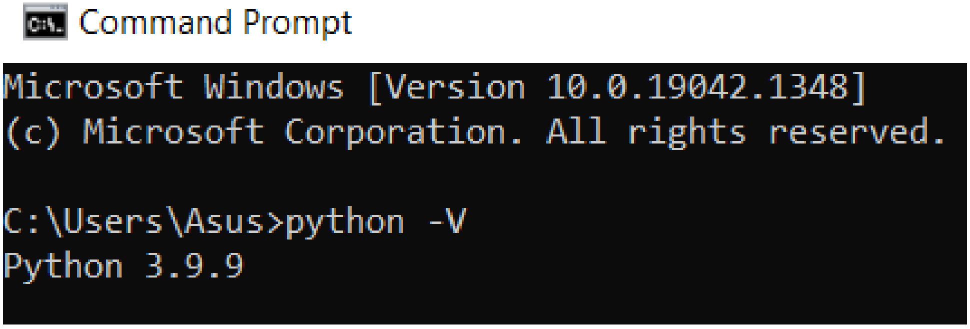
Manufacturer:	ASUSTeK COMPUTER INC.
Processor:	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
Installed memory (RAM):	16.0 GB (15.8 GB usable)
System type:	64-bit Operating System, x64-based processor

Figure 1: System Specification

- **Processor:** Intel Core i7.
- **System Memory:** 1TB Hard disk
- **RAM:** 16GB

2.2 Software Requirement

- **Windows Edition** Windows 10 Enterprise.
- **Scripting Language:** Python 3.9.9



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>python -V
Python 3.9.9
```

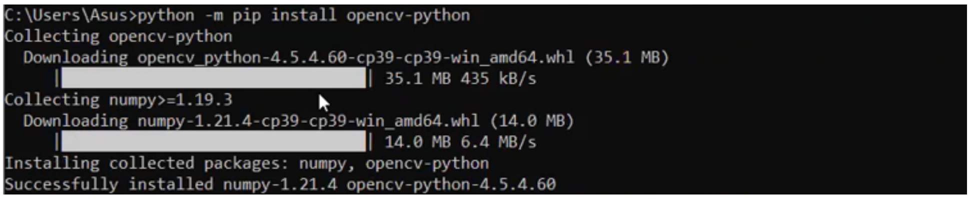
Figure 2: System Specification

- **RAM:** 16GB
- **Cloud Storage:** Google Drive
- **Libraries:** TensorFlow, Matplotlib, OpenCV, Numpy, Keras.

3 Implementation of social distancing monitoring tool

The main task of person detection with bounding boxes has been achieved using YOLOv3 model along with 2 other algorithm plot.py and utills.py and finally importing these two algorithum to main main.ipynb.

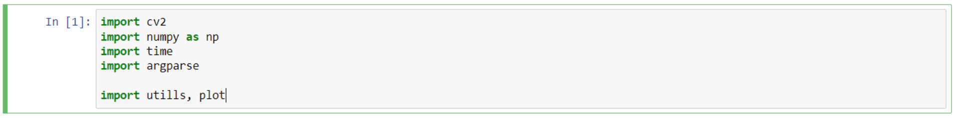
Step 1 is to install important libraries before executing main.ipynb *opencv* (2021)



```
C:\Users\Asus>python -m pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.5.4.60-cp39-cp39-win_amd64.whl (35.1 MB)
     |████████████████████████████████| 35.1 MB 435 kB/s
Collecting numpy>=1.19.3
  Downloading numpy-1.21.4-cp39-cp39-win_amd64.whl (14.0 MB)
     |████████████████████████████████| 14.0 MB 6.4 MB/s
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.21.4 opencv-python-4.5.4.60
```

Figure 3: Library's installation step 1

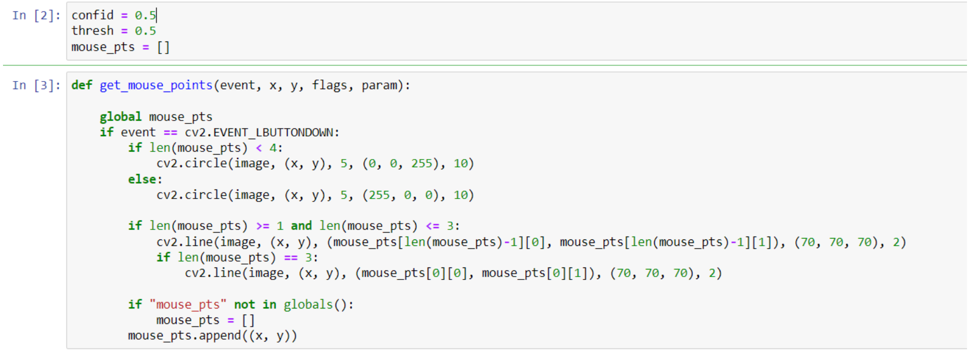
Step 2 is to import required libraries and 2 other algorithm plot.py and utills.py in main.ipynb.



```
In [1]: import cv2
        import numpy as np
        import time
        import argparse

        import utills, plot
```

Figure 4: Import required libraries and 2 other algorithm

Step 3 is to run cell 2 having default configuration for confidence, threshold and cell 3 having function to get mouse point.

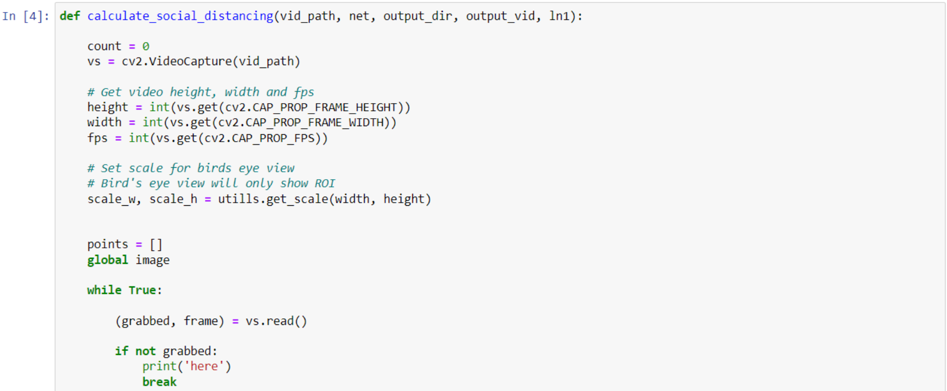


```
In [2]: confid = 0.5
        thresh = 0.5
        mouse_pts = []
```

```
In [3]: def get_mouse_points(event, x, y, flags, param):

            global mouse_pts
            if event == cv2.EVENT_LBUTTONDOWN:
                if len(mouse_pts) < 4:
                    cv2.circle(image, (x, y), 5, (0, 0, 255), 10)
                else:
                    cv2.circle(image, (x, y), 5, (255, 0, 0), 10)

                if len(mouse_pts) >= 1 and len(mouse_pts) <= 3:
                    cv2.line(image, (x, y), (mouse_pts[len(mouse_pts)-1][0], mouse_pts[len(mouse_pts)-1][1]), (70, 70, 70), 2)
                    if len(mouse_pts) == 3:
                        cv2.line(image, (x, y), (mouse_pts[0][0], mouse_pts[0][1]), (70, 70, 70), 2)

                if "mouse_pts" not in globals():
                    mouse_pts = []
                mouse_pts.append((x, y))
```

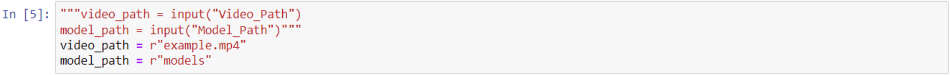
Figure 5: Default values

Step 4, Defining calculate_social_distancing function and giving Yolo model Path and sample video path.



```
In [4]: def calculate_social_distancing(vid_path, net, output_dir, output_vid, ln1):

            count = 0
            vs = cv2.VideoCapture(vid_path)

            # Get video height, width and fps
            height = int(vs.get(cv2.CAP_PROP_FRAME_HEIGHT))
            width = int(vs.get(cv2.CAP_PROP_FRAME_WIDTH))
            fps = int(vs.get(cv2.CAP_PROP_FPS))

            # Set scale for birds eye view
            # Bird's eye view will only show ROI
            scale_w, scale_h = utills.get_scale(width, height)


            points = []
            global image

            while True:

                (grabbed, frame) = vs.read()

                if not grabbed:
                    print('here')
                    break
```

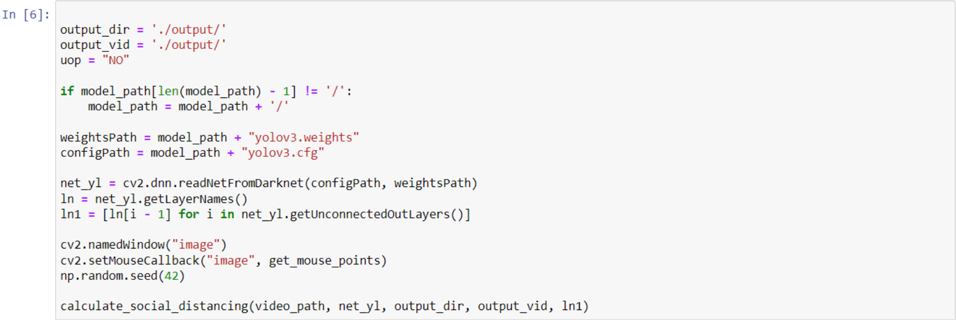
Figure 6: calculate_social_distancing function



```
In [5]: """video_path = input("Video_Path")
        model_path = input("Model_Path")"""
        video_path = r"example.mp4"
        model_path = r"models"
```

Figure 7: Yolo model Path and sample video path

Final step is run last cell of main.ipynb to run complete framework and generate bounding box around sample video.



In [6]:

```python
output_dir = './output/'
output_vid = './output/'
uop = "NO"

if model_path[len(model_path) - 1] != '/':
    model_path = model_path + '/'

weightsPath = model_path + "yolov3.weights"
configPath = model_path + "yolov3.cfg"

net_yl = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
ln = net_yl.getLayerNames()
ln1 = [ln[i - 1] for i in net_yl.getUnconnectedOutLayers()]

cv2.namedWindow("image")
cv2.setMouseCallback("image", get_mouse_points)
np.random.seed(42)

calculate_social_distancing(video_path, net_yl, output_dir, output_vid, ln1)
```

Figure 8: Complete framework

After completing this step, a window will appear as shown in fig 9, where mouse points must be picked to convert the frame into a Bird's-eye view, and distance between individual bounding boxes must be determined using Euclidean Distance as shown in fig 10.

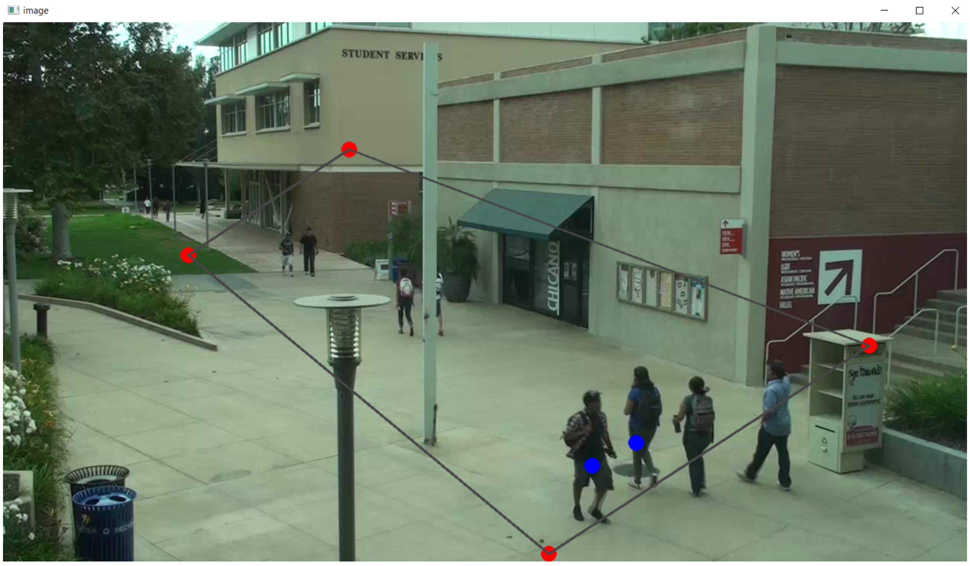
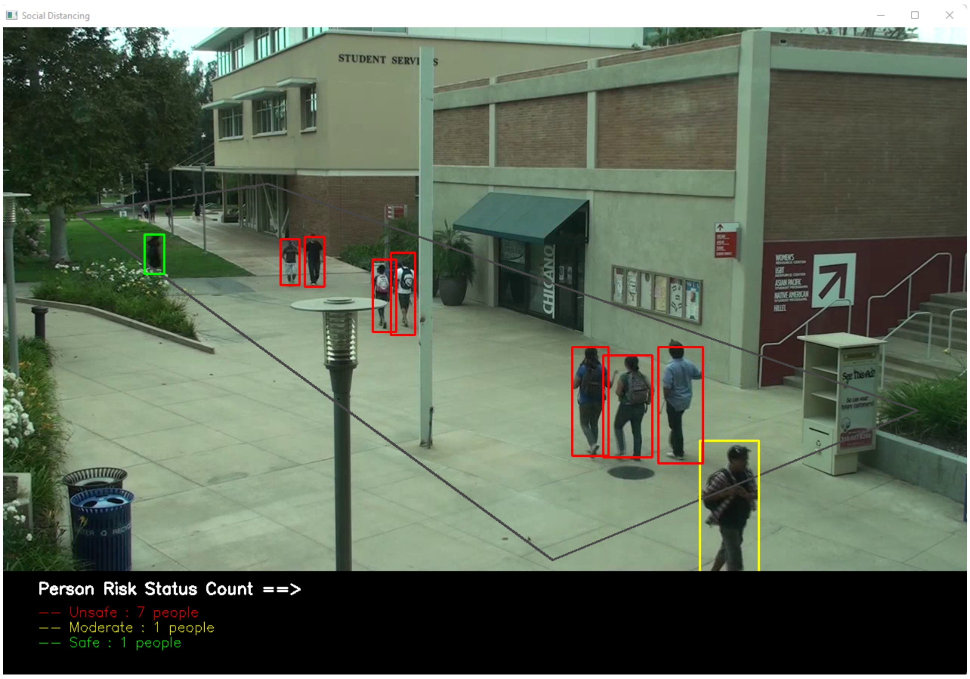


Figure 9: Mouse points selection to convert the frame into Bird-eye view



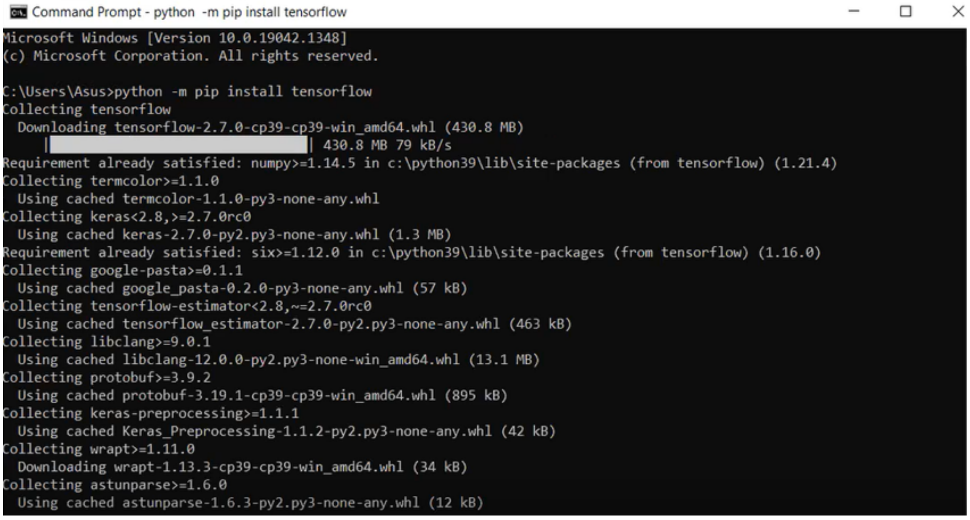
```
Person Risk Status Count ==>
-- Unsafe : 7 people
-- Moderate : 1 people
-- Safe : 1 people
```

Figure 10: Final output calculated using Euclidean Distance

4 Results and evaluation recreation using tensorboard

In this paper, we used three different object detection methods to identify the person class. A performance analysis of each algorithm was carried out to understand the best model.

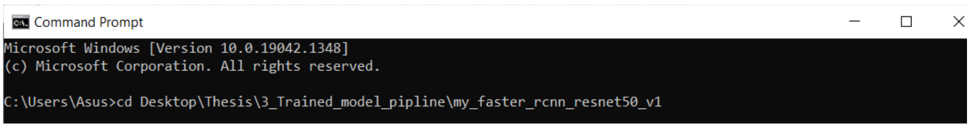
Step 1 install important libraries tenserflow *tensorflow* (2021)



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>python -m pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.7.0-cp39-cp39-win_amd64.whl (430.8 MB)
     |████████████████████████████████| 430.8 MB 79 kB/s
Requirement already satisfied: numpy>=1.14.5 in c:\python39\lib\site-packages (from tensorflow) (1.21.4)
Collecting termcolor>=1.1.0
  Using cached termcolor-1.1.0-py3-none-any.whl
Collecting keras<2.8,>=2.7.0rc0
  Using cached keras-2.7.0-py2.py3-none-any.whl (1.3 MB)
Requirement already satisfied: six>=1.12.0 in c:\python39\lib\site-packages (from tensorflow) (1.16.0)
Collecting google-pasta>=0.1.1
  Using cached google_pasta-0.2.0-py3-none-any.whl (57 kB)
Collecting tensorflow-estimator<2.8,~=2.7.0rc0
  Using cached tensorflow_estimator-2.7.0-py2.py3-none-any.whl (463 kB)
Collecting libclang>=9.0.1
  Using cached libclang-12.0.0-py2.py3-none-win_amd64.whl (13.1 MB)
Collecting protobuf>=3.9.2
  Using cached protobuf-3.19.1-cp39-cp39-win_amd64.whl (895 kB)
Collecting keras-preprocessing>=1.1.1
  Using cached Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Collecting wrapt>=1.11.0
  Downloading wrapt-1.13.3-cp39-cp39-win_amd64.whl (34 kB)
Collecting astunparse>=1.6.0
  Using cached astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
```

Figure 11: Tenserflow libraries installation

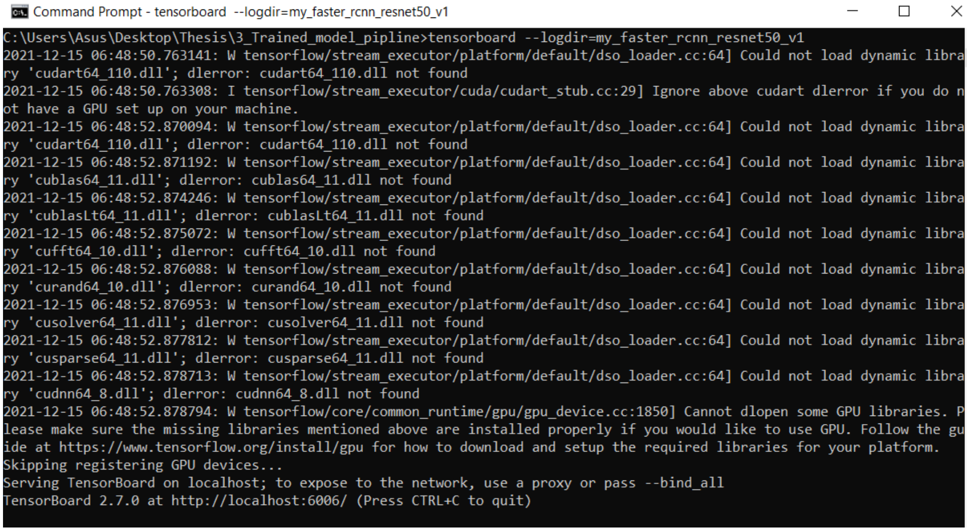
Step 2 changing current directory to directory containing trained models file as shown in fig 12.



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd Desktop\Thesis\3_Trained_model_pipline\my_faster_rcnn_resnet50_v1
```

Figure 12: Changing current directory to directory containing trained models filel

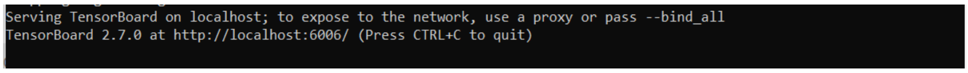
Step 3 Ruining tensorboard command to show the output of selected model as shown in figure 13
tensorboard –logdir=my_faster_rcnn_resnet50_v1


Command Prompt - tensorboard --logdir=my_faster_rcnn_resnet50_v1

```
C:\Users\Asus\Desktop\Thesis\3_Trained_model_pipline>tensorboard --logdir=my_faster_rcnn_resnet50_v1
2021-12-15 06:48:50.763141: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2021-12-15 06:48:50.763308: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do n
ot have a GPU set up on your machine.
2021-12-15 06:48:52.870094: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2021-12-15 06:48:52.871192: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2021-12-15 06:48:52.874246: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2021-12-15 06:48:52.875072: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2021-12-15 06:48:52.876088: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'curand64_10.dll'; dlerror: curand64_10.dll not found
2021-12-15 06:48:52.876953: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2021-12-15 06:48:52.877812: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2021-12-15 06:48:52.878713: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2021-12-15 06:48:52.878794: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries. P
lease make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the gu
ide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.7.0 at http://localhost:6006/ (Press CTRL+C to quit)
```

Figure 13: Tensorboard command to get total/$_{l}oss$

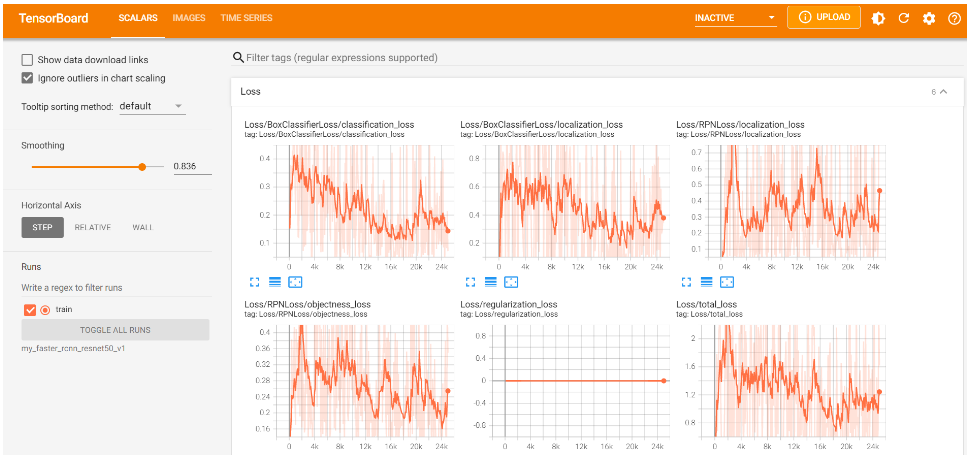
Step 4 Opening http://localhost:6006/ to get the total tensorboard output



```
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.7.0 at http://localhost:6006/ (Press CTRL+C to quit)
```

Figure 14: output link

Loss is a statistic that must be calculated in order to assess the model's effectiveness. In terms of performance, the model with the lowest Total loss is regarded the best. The mAP score, training duration, and FPS were observed and graphically depicted during the model training.


TensorBoard
SCALARS
IMAGES
TIME SERIES
INACTIVE
UPLOAD
Filter tags (regular expressions supported)
Loss
6
Loss/BoxClassifierLoss/classification_loss
tag: Loss/BoxClassifierLoss/classification_loss
Loss/BoxClassifierLoss/localization_loss
tag: Loss/BoxClassifierLoss/localization_loss
Loss/RPNLoss/localization_loss
tag: Loss/RPNLoss/localization_loss
Loss/RPNLoss/objectness_loss
tag: Loss/RPNLoss/objectness_loss
Loss/regularization_loss
tag: Loss/regularization_loss
Loss/total_loss
tag: Loss/total_loss
0 4k 8k 12k 16k 20k 24k
0.1 0.2 0.3 0.4
0.2 0.4 0.6 0.8
0.1 0.2 0.3 0.4 0.5 0.6 0.7
0.16 0.2 0.24 0.28 0.32 0.36 0.4
-0.8 -0.4 0 0.4 0.8
0.8 1.2 1.6 2
TOGGLE ALL RUNS

□ Show data download links
☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.836

Horizontal Axis STEP RELATIVE WALL

Runs

Write a regex to filter runs

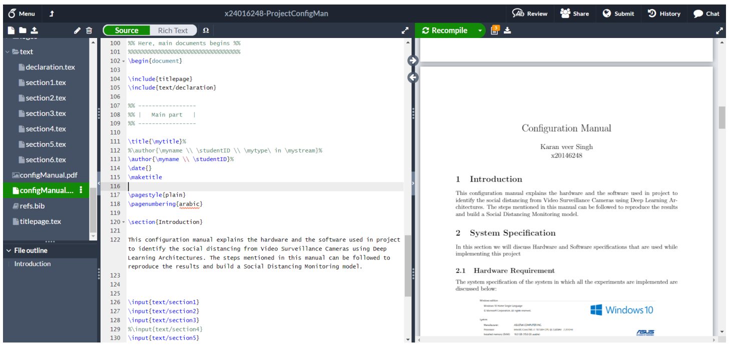
☑ ○ train

my_faster_rcnn_resnet50_v1

Figure 15: Tensorboard output

5 Other Software Used

For the thesis's documentation, we used the overleaf. Figure 16 shows how Overleaf can be used for documentation.*overleaf* (2021).



```
%% Here, main documents begins %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}

\include{titlepage}
\include{text/declaration}

%% ----------------
%% |   Main part   |
%% ----------------

\title{\mytitle}%
%\author{\myname \\ \studentID \\ \mytype\ in \mystream}%
\author{\myname \\ \studentID}%
\date{}
\maketitle

\pagestyle{plain}
\pagenumbering{arabic}

\section{Introduction}

This configuration manual explains the hardware and the software used in project to identify the social distancing from Video Surveillance Cameras using Deep Learning Architectures. The steps mentioned in this manual can be followed to reproduce the results and build a Social Distancing Monitoring model.



\input{text/section1}
\input{text/section2}
\input{text/section3}
%\input{text/section4}
\input{text/section5}
```

Configuration Manual

Karan veer Singh
x20146248

1 Introduction

This configuration manual explains the hardware and the software used in project to identify the social distancing from Video Surveillance Cameras using Deep Learning Architectures. The steps mentioned in this manual can be followed to reproduce the results and build a Social Distancing Monitoring model.

2 System Specification

In this section we will discuss Hardware and Software specifications that are used while implementing this project

2.1 Hardware Requirement

The system specification of the system in which all the experiments are implemented are discussed below:

Figure 16: Overleaf

References

opencv (2021). [Online; accessed 9. Dec. 2021].
URL: *https://pypi.org/project/opencv-python/*

overleaf (2021). [Online; accessed 9. Dec. 2021].
URL: *https://www.overleaf.com/project*

tensorflow (2021). [Online; accessed 10. Dec. 2021].
URL: *https://www.tensorflow.org/*