Low Level Design (LLD)

SDD (Social Distancing Detection)

Revision Number: 1.4
Last date of revision: 08/03/2022

Meenakshi Rao

Document Version Control

Date Issued	Version	Author	Description
01-02-2022	0.1	Meenakshi Rao	First Draft
05-03-2022	0.2	Meenakshi Rao	Added workflow chart, Added overall constraints, KPIs
06-03-2022	0.3	Meenakshi Rao	Added user I/O flowchart
08-03-2022	0.4	Meenakshi Rao	Restructure and reformat LLD

Contents

Abstract

Social distancing detection is very much needed in this pandemic time. This project uses deep learning to evaluate the distance between people so that helps to monitor healthy distance. The detection tool was developed to alert people to maintain a safe distance with each other by evaluating a video feed. The video frame from the camera was used as input, and the open-source object detection pre-trained model.

Main technologies used here are OpenCV, Deep Learning, and Computer Vision. The main objective here is, first human detecting, secondly measure distance between them, if any violation displays that.

1 Introduction

1.1 Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Deep EHR System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The main objective of the project is to detect healthy distance among people. If any violation it has to be intimated.

It uses NumPy, PyCharm, OpenCV, Computer Vision and Deep learning.

- YOLO object detector is applied here for detecting people. It is a real-time object detection algorithm, locates and classifies objects by prioritising speed and recognition instead of locating objects perfectly.
- YOLO object detector files have CNN architecture definitions, pre-trained weights, class names and it is compatible with OpenCV's DNN model.
- Minimum detection confidence or probability and non-maxima suppression threshold are defined first.
- Person-detection module takes 4 arguments: frame, pre-trained YOLO detection model, YOLO CNN output layer names, person-index.
- Results of people-detection module consist of 1. Probability of person detection, 2) bounding Box coordinates for the detection, and 3) centroid of the object.
- Purpose of having non-maxima suppression is to supress weak, overlapping bounding boxes.
- Euclidean metric is to determine the distance between centroids.
- Violate-module manages a list of persons who violate the healthy distance criteria. And set RED colour to boxes.

1.2 Scope

This software can be developed further as Web application. Right now, this is backend application, developed to measure healthy social distance. So that spreading of virus can be brough under control.

1.3 Constraints

Pixel is taken as metrics of distance measurement. Equivalent measurement s like meters, feet also can be used in future implementations.

1.4 Risks

Document specific risks that have been identified or that should be considered.

1.5 Out of Scope

Maintaining database of people violating rules and web development are out of scope of this project.

2 Technical Specifications

2.1 Dataset

	Finalized	**Source**
YOLO Person Detector	Yes	https://pjreddie.com/darknet/yolov1/
Input video	Yes	Local input_files folder

2.2 Overview of YOLO Dataset and Input Files

YOLO has coco-names, coco-weights and coco-config. Sample screen shot of contents are given below:

coco.names

```
person
bicycle
car
motorbike
aeroplane
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
sofa
pottedplant
bed
diningtable
toilet
tvmonitor
laptop
mouse
remote
keyboard
cell phone
microwave
oven
toaster
sink
refrigerator
book
clock
vase
scissors
teddy bear
hair drier
toothbrush
```

coco.names yolov3.cfg

```
[net]
# Testing
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 500200
policy=steps
steps=400000,450000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=255
activation=linear


[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,
classes=80
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

2.3 Logging

We should be able to log every activity done by the user.

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

2.4 Exception Handling

It is mandatory in good coding practice. It avoids unnecessary errors while execution. Unexpected behaviour of code can be easily caught.

2.5 Modular Programming

PyCharm IDE provides good platform for modular approach of implementing code. Reusability, debugging, understanding everything can be archived from this style of coding.

3. Technology stack

Front End	PyCharm IDE
Backend	Python
Database	MongoDB/MySql (could be used for logging instead of files)
Deployment	Not applicable

4. Proposed Solution

Refer:

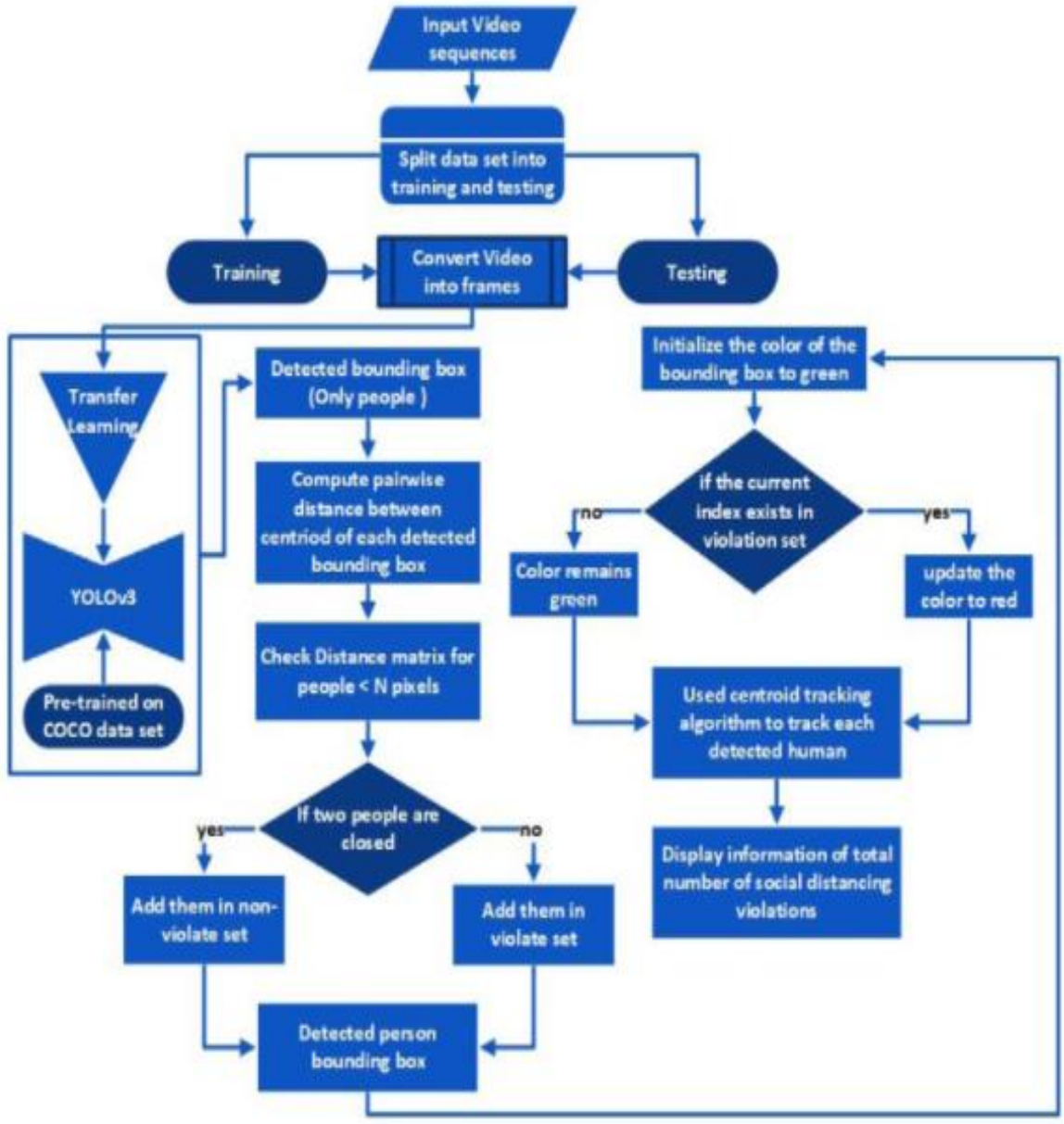
- **https://arxiv.org/abs/2005.01385**
- **https://pyimagesearch.com**
- **https://pjreddie.com/darknet/yolo/**

Object detection is one of the trending fields in Deep Learning. It is required for almost all object-based projects.

Basics of YOLO model, social distance detection concepts are referred from these resources.

5. Model training/validation workflow

Over all flow is given here. Since YOLO provides pretrained model, we are not splitting input frames into train set and test set.



6. Exceptional scenarios

Step	Exception	Mitigation	Moderator
6th March 2022	1.1	First Draft, Added Workflow chart	Meenakshi

7. Test Cases

Test case	Steps to perform test case	Module	Pass/Fail
1	Try with sample data	Detector()	Pass

8. Key performance indicators (KPI)

- Person detection
- Displaying healthy and unhealthy distances among people
- Saving output for future reference
- Successful logging
- Modular coding
- Exception’s handling