Readme

MONITORING SOCIAL DISTANCING UNDER LOW LIGHT CONDITIONS USING YOLOV5

TABLE OF CONTENTS:

* SIMPLE THEORY
* ALGORITHM USED - YOLOV5
* TECHNOLOGIES USED
* RUNNING INFERENCE

SIMPLE THEORY:

* Mainly focused on monitoring social distancing in low light situations.For this we used the yolo algorithm which gives a good result in object detection. To achieve this we used the dark dataset which has the images of different modules like people, vehicles, buildings, animals etc in low light.
* The dataset used https://github.com/cs-chan/Exclusively-Dark-Image-Dataset/tree/master/Dataset.
* For this project we need to select only people images for monitoring social distancing between people and perform object detection on people images.
* For drawing boundary boxes for people's images we used makesense.ai tool.(https://www.makesense.ai/).

PROCEDURE:

           First we need to get the frame and detect the objects i.e; only people,display the boundary box which is initialised to the green, and calculate the distance if people maintain minimum distance then it displays the green boundary box otherwise change it to red and display the boundary box which is output.

ALGORITHM USED-YOLOV5

YOLOV5 is used for the detection which pre trained on coco dataset.It mainly consists of

i)Back bone

ii)Neck

iii)Head

DISTANCE CALCULATION

To define center\_distance() function for calculating the distance between the coordinates.coordinates are xyxy1 and xyxy2

np.linalg.norm-used to calculate the norm of a vector or matrix,so just to calculate the Frobenius form (x1-x2) and also for distance calculation.

formula:dist=np.linalg.norm([x1-x2,y1-y2]).

TECHNOLOGIES USED:

GOOGLE COLABORATORY as Development environment.

       It is an open-source web application that allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis.

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code.

PYTHON Programming language

Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python’s simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML problem instead of focusing on the technical nuances of the language.

Additionally, Python is appealing to many developers as it’s easy to learn. Python code is understandable by humans, which makes it easier to build models for machine learning.

Python Modules like PANDAS, NUMPY, MatPlotLib , Scikit Learn

Pandas can easily fetch data from different sources like SQL databases, CSV, Excel, JSON files and manipulate the data to perform operations on it.

MatPlotLib is a standard Python library used by every data scientist for creating 2D plots and graphs. It’s pretty low-level, meaning it requires more commands to generate nice-looking graphs and figures than with some advanced libraries.

RUNNING INFERENCE:

* To begin first we need to mount the google colab with google drive.
* Install all the required python requirements i.e; libraries

pip install -r requirements.txt

* Next, we need to train the data.
* yolov5 needs a YAML file, given the one class i.e; person .The weight used :yolo V5x
* Next, we need to validate the dataset and detect the dataset.
* Next, we need to calculate the distance.
* Next, In the detect\_people\_on\_frame() function passed arguments (img, confidence, distance). Next, Pass the frame through the model and get the boxes. Here, we get the xyxy as box coordinates.Next,call the center\_distance() function and got the distance between the two boundary boxes then check the condition between the actual distance  and calculated dist and if (dist < distance) then the boundary box results as red colour otherwise the boundary box remains as green colour.
* Next,In the detect\_people\_on\_video() function pass the argument as(filename,confidence,distance) . From here get the frames from the video and pass the frame to the detect\_people\_on\_frame() function and save the result into output5.mp4. Then call the function detect\_people\_on\_video and pass the input ex.mp4 as filename and confidence=0.5.