International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072

SOCIAL DISTANCING DETECTION SYSTEM WITH ARTIFICIAL INTELLIGENCE USING COMPUTER VISION AND DEEP LEARNING

Vinitha.V[1], Velantina.V[2]

[1]*Student, Department of Master of Computer Application, AMC Engineering College Bangalore, Karnataka, India*
[2]*Student, Department of Computer Science and Engineering (M.Tech), C.B.I.T Kolar, Karnataka, India*

Abstract - *In the fight against the COVID-19, social distancing has proven to be a very effective measure to slow down the spread of the disease. People are asked to limit their interactions with each other, reducing the chances of the virus being spread with physical or close contact. In past also AI/Deep Learning has shown promising results on a number of daily life problems. In this proposed system we will see the detailed explanation of how we can use Python, Computer Vision and Deep learning to monitor social distancing at public places and workplaces. To ensure social distancing protocol in public places and workplace, the social distancing detection tool that can monitor if people are keeping a safe distance from each other by analyzing real time video streams from the camera, Monitoring People at workplaces, factories, shops we can integrate this tool to their security camera systems and can monitor whether people are keeping a safe distance from each other or not.*

Key Words: **Covid-19, Social distancing, Opencv, Python, Computer vision.**

1. INTRODUCTION

To limit the spread of an infectious disease, for instance, Covid-19, is to practice social distancing. This is not a new concept, as most societies have been aware of the value of keeping away from people who are suffering from an infection for many generations. The objective is to reduce transmission, delaying the epidemic peak, reducing the size of the epidemic peak, and spreading cases over a longer time to relieve pressure on the healthcare system. It is an action taken to minimize contact with other individuals. It has been suggested that maintaining a distance of approximately 2 metres from another individual result in a marked reduction in transmission of most flu virus strains, including COVID-19.

In practice, this means that avoiding close proximity to other people will aid in slowing the spread of infectious diseases. Social distancing is one of the non-pharmaceutical infection control actions that can stop or slow down the spread of a highly contagious disease. The virus that causes COVID-19 is currently spreading easily from person-to-person. When a healthy person comes into contact with respiratory droplets from coughs or sneezes of an infected person, they are can catch the infection.

Fig – 1: Social Distancing

The World Health Organization (WHO) states that "COVID-19 is transmitted via droplets and fomites during close unprotected contact between an infector and infectee. A fomite is an object or material which is likely to carry infection, such as clothes, utensils, and furniture. Therefore, transmission of the infection can be avoided by staying away from other people as well as from touching infected fomites. Social distancing aims to decrease or interrupt transmission of COVID-19 in a population[1] by minimizing contact between potentially infected individuals and healthy individuals, or between population groups with high rates of transmission and population groups with no or low levels of transmission.

Methods of Social Distancing - Cancellation of events which involve large numbers of people gathering together, such as

- Closure of Community Facilities
- Closure of non-essential workplaces
- Closure of schools
- Closure of colleges and universities
- Self-Shielding
- Individuals limit face-to-face contacts
- Individuals avoid public places

- Individuals avoid public transport

Social distancing is a term applied to certain actions that are taken by Public Health officials to stop or slow down the spread of a highly contagious disease. The Health Officer has the legal authority to carry out social distancing measures. Since these measures will have considerable impact on our community, any action to start social distancing measures would be coordinated with local agencies such as cities, police departments and schools, as well as with state and federal partners.

1.1 Artificial Intelligence

Artificial intelligence (AI), sometimes called machine intelligence, is intelligence demonstrated by machines, Leading AI define the field as the study of "intelligent agents" any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving". As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI affect "AI is whatever hasn't been done yet. For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology. Modern machine capabilities generally classified as AI include understanding human speech, competing at the highest, autonomously operating cars, intelligent routing in content delivery networks, and military simulations.

Fig – 2: Artificial Intelligence

1.2 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected.

Python is often described a language due to its comprehensive standard library. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution, which binds method and variable names during program execution.

1.3 OpenCV

OpenCV Python is a library of Python bindings designed to solve computer vision problems. Python is a general purpose programming language started by **Guido van Rossum** that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability. Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python makes use of **Numpy**, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib. OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

1.4 TENSORFLOW

Tensor Flow is an open source library for fast numerical computing. It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. Unlike other numerical libraries intended for use in Deep Learning like Theano, Tensor Flow was designed for use both in research and development and in production systems, It can run on single CPU systems, GPUs as well as mobile devices and large scale distributed systems of hundreds of machines. Computation is described in terms of data flow and operations in the structure of a directed graph.

Nodes: Nodes perform computation and have zero or more inputs and outputs. Data that moves between nodes are known as tensors, which are multi-dimensional arrays of real values.

Edges: The graph defines the flow of data, branching, looping and updates to state. Special edges can be used to synchronize behavior within the graph, for example waiting for computation on a number of inputs to complete.

Operation: An operation is a named abstract computation which can take input attributes and produce output attributes. For example, you could define an add or multiply operation.

1.5 YOLOV3

YOLOv3 is the latest variant of a popular object detection algorithm YOLO – You Only Look Once. The published model recognizes 80 different objects in images and videos, but most importantly it is super fast and nearly as accurate as Single Shot MultiBox (SSD). First, it divides the image into a 13×13 grid of cells. The size of these 169 cells varies 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**.**

Easy integration with an OpenCV application: If your application already uses OpenCV and you simply want to use YOLOv3, you don't have to worry about compiling and building the extra Dark net code.

OpenCV CPU version is 9x faster: OpenCV's CPU implementation of the DNN module is astonishingly fast. For example, Dark net when used with OpenMP takes about 2 seconds on a CPU for inference on a single image. In contrast, OpenCV's implementation runs in a mere 0.22 seconds! Check out table below.

Python support: Dark net is written in C, and it does not officially support Python. In contrast, OpenCV does.

2. PROPOSED SYSTEM

The proposed system focuses on how to identify the person on image/video stream whether the social distancing is maintained or not with the help of computer vision and deep learning algorithm by using the OpenCV, Tensor flow library.

Approach

1. Detect humans in the frame with yolov3.
2. Calculates the distance between every human who is detected in the frame.
3. Shows how many people are at High, Low and Not at risk.

Flow Chart

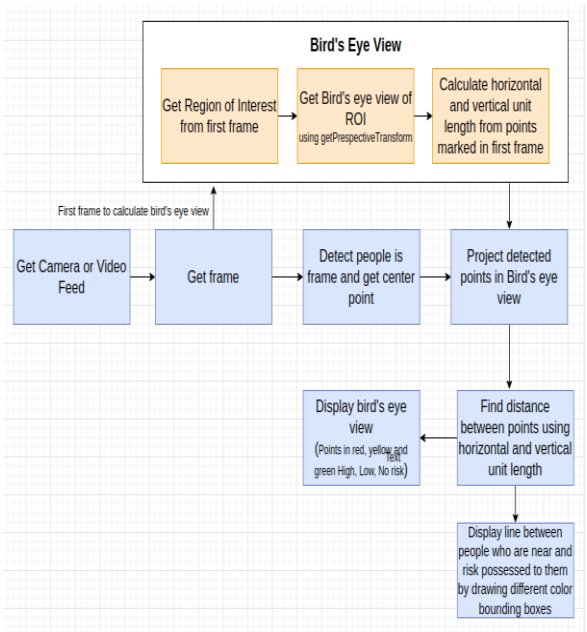

Fig – 3: Flow diagram of social distancing detector model

Camera Perspective Transformation or Camera Calibration:

As the input video may be taken from an arbitrary perspective view, the first step is to transform perspective of view to a bird's-eye (top-down) view. As the input frames are monocular (taken from a single camera), the simplest transformation method involves selecting four points in the perspective view which define ROI where we want to monitor social distancing and mapping them to the corners of a rectangle in the bird's-eye view. Also these points should form parallel lines in real world if seen from above (bird's eye view). This assumes that every person is standing on the same flat ground plane. This top view or bird eye view has the property that points are distributed uniformly horizontally and vertically (scale for horizontal and vertical direction will be different).From this mapping, we can derive a transformation that can be applied to the entire perspective image.

Fig – 4: ROI and scale points selection

Above image shows how we can select Region of Interest (ROI) and this is one time step. We draw 8 points on first frame using mouse click event. First four points will define ROI where we want to monitor social distancing. Next 3 points will define 180 cm(unit length) distance in horizontal and vertical direction and those should form parallel lines with ROI. In above image we can se point 5 and point 6 defines 180 cm in real life in horizontal direction and point 5 and point 7 defines 180 cm in real life in vertical direction. As we can see ROI formed by first 4 points has different length in horizontal and vertical direction, so number of pixels in 180 cm for horizontal and vertical direction will be different in rectangle (bird's eye view) formed after transformation.

So from point 5, 6, 7 we are calculating scale factor in horizontal and vertical direction of the bird's eye view, e.g. how many pixels correspond to 180 cm in real life.

Detection

The second step to detect pedestrians and draw a bounding box around each pedestrian. To clean up the output bounding boxes, we apply minimal post-processing such as non-max suppression (NMS) and various rule-based heuristics, so as to minimize the risk of over fitting.

Distance Calculation

Now we have bounding box for each person in the frame. We need to estimate person location in frame. i.e we can take bottom center point of bounding box as person location in frame. Then we estimate (x,y) location in bird's eye view by applying transformation to the bottom center point of each person's bounding box, resulting in their position in the bird's eye view. Last step is to compute the bird's eye view distance between every pair of people and scale the distances by the scaling factor in horizontal and vertical direction estimated from calibration.

Software and libraries required

Python
Opencv
numpy
argparse

Working

Running the program will give you frame (first frame) where you need to draw ROI and distance scale. To get ROI and distance scale points from first frame Code to transform perspective to Bird's eye view (Top view) and to calculate horizontal and vertical 180 cm distance in Bird's eye view

ROI and Scale points' selection for first frame. The second step to detect pedestrians and draw a bounding box around each pedestrian. To detect humans in video and get bounding box details.

Now we have bounding box for each person in the frame. We need to estimate person location in frame. i.e we can take bottom center point of bounding box as person location in frame. Then we estimate (x,y) location in bird's eye view by applying transformation to the bottom center point of each person's bounding box, resulting in their position in the bird's eye view. To calculate bottom center point for all bounding boxes and projecting those points in Bird's eye view. Last step is to compute the bird's eye view distance between every pair of people (Point) and scale the distances by the scaling factor in horizontal and vertical direction estimated from calibration.

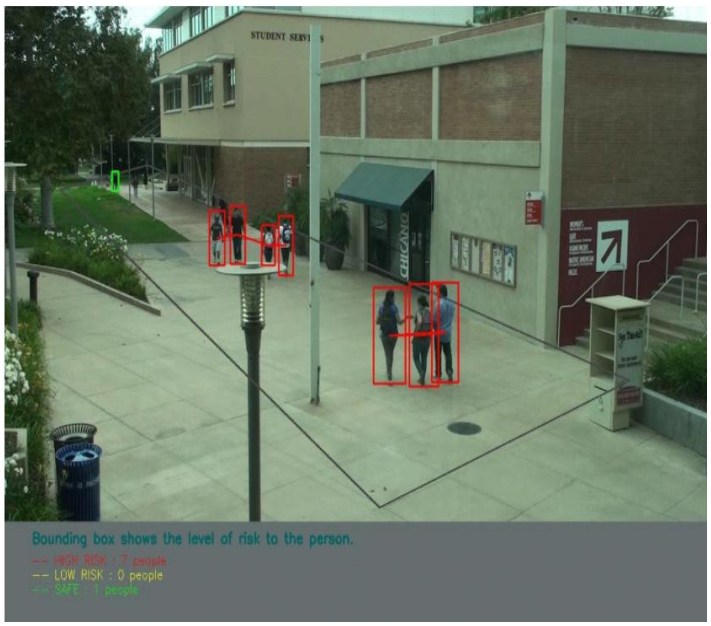

Fig – 5: Social distance detector detecting the distance in a video steam the number of people under low risk, high risk and safe.

Lastly we can draw Bird's Eye View for region of interest (ROI) and draw bounding boxes according to risk factor for humans in a frame and draw lines between boxes according to risk factor between two humans. Red, Yellow, Green points represents risk to human in Bird's eye view. Red: High Risk, Yellow: Low Risk and Green: No Risk. Red, Yellow lines between two humans in output tell they are violating social distancing rules.

3. CONCLUSION

The emerging trends and the availability of intelligent technologies make us to develop new models that help to satisfy the needs of emerging world. So we have developed a novel social distancing detector which can possibly contribute to public healthcare. The model proposes an efficient real-time deep learning based framework to automate the process of monitoring the social distancing via object detection and tracking approaches, where each individual is identified in the real-time with the help of bounding boxes. Identifying the clusters or groups of people satisfying the closeness property computed with the help of Bird's eye view approach. The number of violations is confirmed by computing the number of groups formed and violation index term computed as the ratio of the number of people to the number of groups. The extensive trials were conducted with popular state-of-the-art object detection models Faster RCNN, SSD, and YOLO v3, since this approach is highly sensitive to the spatial location of the camera, the same approach can be fine tuned to better adjust with the corresponding field of view.

This system works very effectively and efficiently in identifying the social distancing between the people and generating the alert that can be handled and monitored.

REFERENCES

[1] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition ´ via sparse spatio-temporal features," in 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. IEEE, 2005, pp. 65–72.

[2] M. Piccardi, "Background subtraction techniques: a review," in 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), vol. 4. IEEE, 2004, pp. 3099–3104.

[3] Y. Xu, J. Dong, B. Zhang, and D. Xu, "Background modeling methods in video analysis: A review and comparative evaluation," CAAI Transactions on Intelligence Technology, vol. 1, no. 1, pp. 43–60, 2016.

[4] H. Tsutsui, J. Miura, and Y. Shirai, "Optical flow-based person tracking by multiple cameras," in Conference Documentation International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI 2001 (Cat. No. 01TH8590). IEEE, 2001, pp. 91–96.

[5] A. Agarwal, S. Gupta, and D. K. Singh, "Review of optical flow technique for moving object detection," in 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I). IEEE, 2016, pp. 409–413.

[6] S. A. Niyogi and E. H. Adelson, "Analyzing gait with spatiotemporal surfaces," in Proceedings of 1994 IEEE Workshop on Motion of Nonrigid and Articulated Objects. IEEE, 1994, pp. 64–69.

[7] Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[10] X. Chen and A. Gupta, "An implementation of faster rcnn with study for region sampling," arXiv preprint arXiv:1702.02138, 2017.

[11] https://blog.usejournal.com/social-distancing-ai-using-python-deep-learning-c26b20c9aa4c.

[12] N. S. Punn and S. Agarwal, "Crowd analysis for congestion control early warning system on foot over bridge," in 2019 Twelfth International Conference on Contemporary Computing (IC3). IEEE, 2019, pp. 1–6.

[13] Pias, "Object detection and distance measurement," https://github.com/ paul-pias/Object-Detection-and-Distance-Measurement, 2020.

[14] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," Neurocomputing, vol. 300, pp. 17–33, 2018.