



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 4, April 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



Social Distancing Violation Analyzer

Sanjivani Joshi, Piyush Hadge, Sanket Parsewar, Vaibhav Parsewar, Umar Shaikh

Dept. of Information Technology, Sinhgad College of Engineering, Pune, India

ABSTRACT: The Novel Coronavirus outbreak worldwide has made people to consider different ways to perhaps protect themselves from infections. Several safety precautions, including avoiding crowded areas, maintaining hand hygiene, and avoiding touching eyes, mouth, or nose. Researchers, mathematicians, pharmacists, and other professionals have all been challenged by the pandemic to find solutions to the pandemic situation. Algorithms and concepts related to machine learning also find a great place for many scientists. Among all the preventive measures, social distancing is one of the most important protective methods for flattening the COVID-19. In order to identify people who are not maintaining social distance, an effective and persuasive social distancing violation analyzer tool is essential. The system examines frames it receives as input to find humans in the visual frame. The system will then give a warning for anyone who is not maintaining the social distance based on the distance value collected after calculating the pairwise distance between each identified individual. To determine how well the system performs, it is tested using a variety of image input acquisition techniques, including camera, video, and image. The proposed system achieves a greater precision for videos, according to experimental findings. Applications like human tracking, pedestrian recognition, and vehicle tracking can all be added to the system.

I. INTRODUCTION

The SARS-CoV-2 virus is an infectious disease also known as coronavirus disease (COVID-19) [1] [2]. This virus spread throughout the entire world, affecting 213 countries and territories. The number of progressive cases up to February 2023 is estimated to be approximately 67.4 Cr, and the number of deaths is estimated to be around 68.6 L. The World Health Organization (WHO) [3] has declared COVID-19, the SARS-COV2 virus that emerged in Wuhan, China, in December 2019, as a global public health emergency.

To protect against COVID-19, the WHO has advised the population to become acclimated to themselves. They have offered numerous suggestions and preventative steps to be followed to defend oneself against the virus. Some of these include avoiding crowded areas, keeping your hands clean, and refraining from touching your eyes, mouth, or nose. Out of all these preventative measures, social isolation has the biggest impact on the reduction of infections. Social isolation and Social Distancing have the biggest impact on reducing infections out of all these preventative methods. To flatten the curve of disease propagation, social distancing must be enforced. Fig 1 shows a case of social distance. The citizens must adhere to the social distance, according to the higher authority. This work aims to develop a social distancing violation analyzer that can detect and warn people if they are not maintaining social distance. The suggested approach focuses on detecting the individuals that may be seen in picture or video streams and calculating the distance between two people to determine whether they maintain social distancing.

The paper is organized as follows - The relevant work in the identification of social distance is illustrated in Section II. Section III displays the suggested social distance violation analyzer. The methodologies used are explained in Section IV and Software Requirements are provided in Section V.

II. RELATED WORK

Many similar studies on object tracking and identification are being conducted. Various research papers discussed in [4] [5] show how human activity detection can be performed and contact tracing can be done. exploit the YOLO v3 object detection model to detect humans from the background, and the humans are tracked using the Deep Sort approach using bounding boxes. Chakrabarti P. [6] investigated the effect of social distancing while reducing the COVID-19 [8] curve using the Stochastic Simulation Model. The findings indicate that the COVID-19 curve is not flattened by social distance alone. In addition to social distancing, the number of hospital beds also must increase to handle COVID-19. YOLO Object detector [7] is the current prevalent technique of object detection algorithm. The recognized model identifies 80 varieties of images and videos. The main advantage of using this algorithm is its faster computation and efficiency. Gaeta G [5] developed a social distancing detection system by considering four properties:



the developed system never stores the cache data, the individuals should not be targeted, and the high-level human should not be detected.

III. PROPOSED SYSTEM

A. User Interface

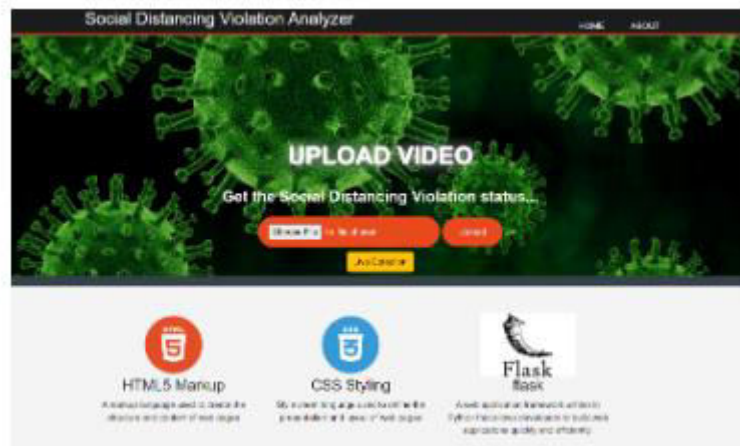


Fig. Home page



Fig. About page

Home: This button would redirect the user to the home page of the website. This page contains the file picker to allow user to select local video files, upload button and the live detection button.

About: This button would redirect the user to the about page of your website. This page contains information about the working of the system and its use cases.

File Picker: This button allows the user to select a local video file from their device. Upon clicking the button, the user may see a popup window that allows them to browse their device and select the video file they want to upload for processing.

Upload: This button allows the user to upload the selected video file for processing. Upon clicking the button, the system starts processing the file and provides the user with an output video file that contains the social detection violation statuses of all the detected people.

Live Detection: This button allows the user to start social distance detection on live video stream coming from an external camera. Upon clicking the button, the user sees a preview of the camera stream and the social distance detection algorithm would analyze the stream in real-time to identify social distancing violations.



B. Recorded Video Processing

There are two ways the user can use the system. One way is to upload a recorded video for processing and the other way is to process a live video stream. In the first way the user uses the file picker to select the desired file that the user wants to process. The user then clicks on the upload button. After that the system processes the video file and presents the processed file to the user. The user can then play or save the video.

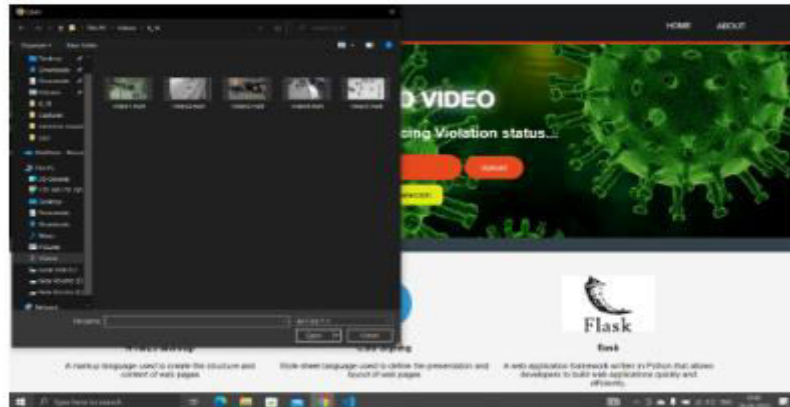


Fig. Selecting recorded video from file system



Fig. Selected video file

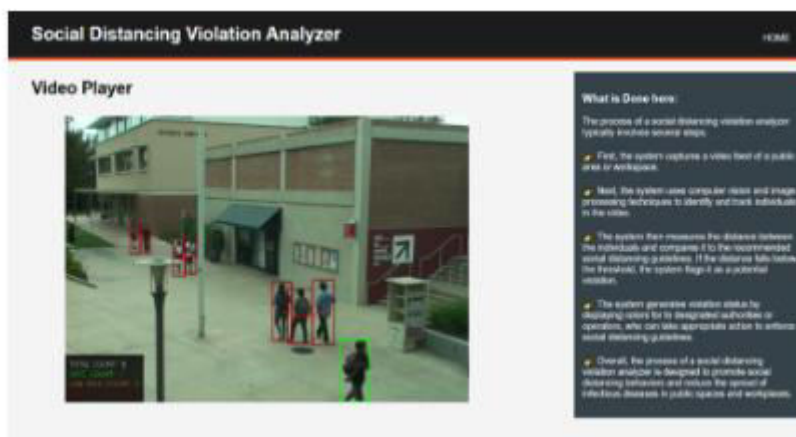


Fig. Detection

C. Live Stream Processing using external camera

The user can also use the system to process live video stream. The user can use the Live Detection button to start live video processing using the attached external camera as video input. The live video stream is shown on a new page. The social distance violation status of detected people is updated in real time.



Fig. Live detection_1



Fig. Live detection_2

D. Integration using Flask

Flask is a micro web framework written in Python that allows developers to build web applications quickly and with minimal overhead. It is classified as a micro-framework because it does not require particular tools or libraries, but it does have support for extensions that can add application features.

Steps involved to integrate Flask -

1. **Create a templates directory:** Create a directory named "templates" in the root directory of your Flask application. This is where your HTML templates will be stored.



2. **Create a static directory:** Create a directory named "static" in the root directory of your Flask application. This is where your static files like CSS, JavaScript, and images will be stored.
3. **Define routes for serving HTML templates:** In your Flask application, define routes for serving HTML templates.
4. **Define routes for serving static files:** In your Flask application, define routes for serving static files.
5. **Link HTML templates to static files:** In your HTML templates, link to static files using the ``url_for()`` function.

IV. WORKING

Description of Architecture

Steps for object Detection using YOLO v3:

- A collection of photos of the shape (m, 416, 416, 3) make up the inputs. This image is sent to a convolutional neural network (CNN) via YOLO v3.
- A list of bounding boxes and the identified classes is the output. Six values are used to represent each bounding box (pc, bx, by, bh, bw, c). Each bounding box is represented by 85 values when c is expanded into an 80-dimensional vector.
- Finally, we do the Non-Max Suppression to avoid selecting overlapping boxes.

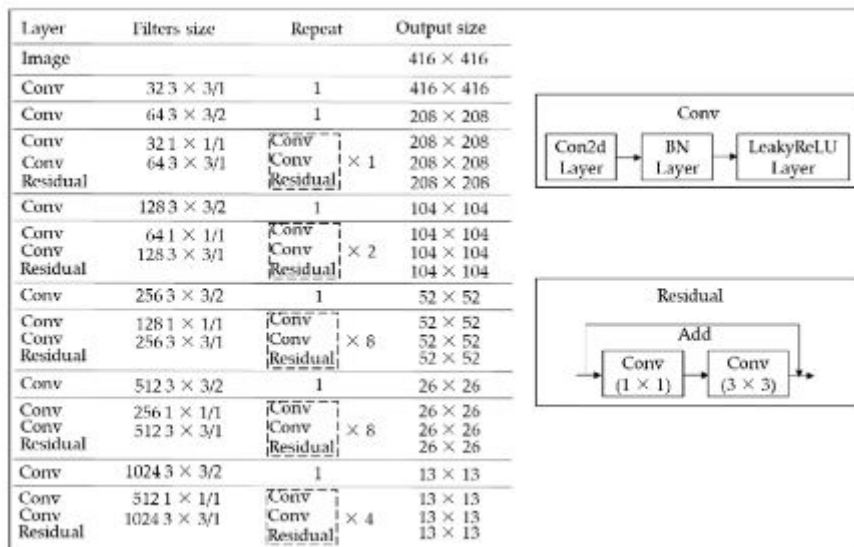


Fig. Architecture of Yolo v3

Layers Details

- YOLO is a fully convolutional network (FCN) because it uses just convolutional layers.
- In YOLOv3, a deeper feature extractor architecture known as Darknet-53 is implemented.

Convolution layers in YOLOv3

- It has 53 convolutional layers, followed by batch normalisation layers and leaky ReLU activation for each layer.
- The convolution layer is used to combine different filters on the images, creating a variety of feature maps.
- No form of pooling is used and a convolutional layer with stride 2 is used to down sample the feature maps.
- It helps in reducing the loss of low-level features that pooling can be responsible for.

A. Person Detection using YOLO v3 pre-trained model

Person detection using YOLO v3 pre-trained model involves using a deep learning algorithm to identify and locate people in an image or video stream. YOLO (You Only Look Once) is a state-of-the-art object detection algorithm that can detect multiple objects in a single frame in real-time. The YOLO v3 model is pre-trained on a large dataset of images and can be fine-tuned on specific object detection tasks, such as person detection. The pre-trained model uses

convolutional neural networks (CNNs) to analyze the input image or video frame at different scales and levels of abstraction. This allows the model to detect objects of different sizes and shapes in the image. To perform person detection using the YOLO v3 pre-trained model, the following steps are typically taken:

1. Load the pre-trained YOLO v3 model and its weights.
2. Process the input image or video frame by resizing it to the required input size of the YOLO v3 model (usually 416x416 pixels) and normalizing the pixel values.
3. Pass the pre-processed input through the YOLO v3 model and obtain a set of bounding boxes, confidence scores, and class predictions for each detected object in the image.
4. Filter the bounding boxes and class predictions to only include those that correspond to people. This can be done using the class prediction scores or a threshold value.
5. Draw the filtered bounding boxes on the original input image or video frame to indicate the location of the detected people.
6. Optionally, perform additional post-processing steps such as non-maximum suppression (NMS) to remove overlapping bounding boxes or tracking the detected people across multiple frames in a video stream.

B. Distance Calculation and Calibration of horizontal and vertical distance

Horizontal and Vertical distance are calculated separately as horizontal distance doesn't require any calibration whereas vertical distance requires calibration. When we calculate the distance between two points in the real world, we need to consider not only the horizontal distance between them but also the vertical distance. The vertical distance is the difference in height between the two points. To calculate the correct physical distance between the two points, we need to adjust the distance formula to consider the vertical distance. The adjustment involves adding a term that is proportional to the average y-coordinate of the two points. This proportional term accounts for the fact that the same difference in y-coordinates can represent a larger physical distance in the real world if the two points are farther apart vertically. So, by including this term, we can calculate the physical distance more accurately.

The process of calibrating the distance formula to account for the vertical distance between points is often called "calibration of distance for slope" or "slope correction". By including a term that is proportional to the average y-coordinate of the points, the formula can more accurately represent the physical distance between the points, considering any vertical separation between them.

The following functions are used to calculate the physical distance and the violation status respectively. The function `calibrated_dist(p1, p2)` takes two points that represent the centres of the two bounding boxes. $p1[0] - p2[0]$ is the horizontal distance and $550 / ((p1[1] + p2[1]) / 2) * (p1[1] - p2[1])$ is the calibrated vertical distance. The `isclose(p1, p2)` function is used to calculate the violation status of the two bounding boxes and it returns the violation status of the two bounding boxes.

```
def calibrated_dist(p1, p2):
    return ((p1[0] - p2[0]) ** 2 + 550 / ((p1[1] + p2[1]) / 2) * (p1[1] - p2[1]) ** 2) ** 0.5

def isclose(p1, p2):
    c_d = calibrated_dist(p1, p2)
    calib = (p1[1] + p2[1]) / 2
    if 0 < c_d < 0.15 * calib:
        return 1
    elif 0 < c_d < 0.2 * calib:
        return 2
    else:
        return 0
```

Fig. Distance calibration code snippet

C. Updating Violation Status based on the distance between two people

There are three violation statuses - green, orange and red. Green is safe while orange and red are unsafe. The system changes the colour of the bounding boxes of people based on their distance from other people. If two people are close to each other their bounding boxes will turn orange, alerting the user. If the people are very close their bounding boxes will be red, and there will also be a line connecting the centres of their bounding boxes.



Fig. violation status – Green (safe)



Fig. violation status – Orange (low risk)



Fig. Violation status – Red (high risk)

D. Provide statistics to the user

At all times the user is shown the number of people detected, the number of people with red, orange and green violation statuses. This is achieved using Graphics functions in OpenCV like putText().



Fig. Statistics

V. REQUIREMENT

A. Software Requirements –

The software requirements outline the characteristics and features of our suggested system. The criteria may be plainly visible or may be hidden to the user,

- YOLO v3

Either known or unknown, and either expected or unexpected. The system is taught using the YOLO deep learning method; therefore, a separate dataset is not required to recognise persons in the picture. For object detection, it makes use of a clever real-time convolutional neural network (CNN).

- Computer vision.

With the use of computer vision, we can learn how to alter and extract data from images and videos as well as understand how they are stored OpenCV is a comprehensive open-source library for image processing, computer vision, and machine learning. Today, it makes a big difference in real-time functioning, which is crucial in modern systems.

- Flask.

For Web Application, our proposed system uses Flask which is a lightweight and flexible web framework for Python that allows developers to easily build web applications. Flask is considered a "micro" framework because it does not require tools or libraries, and it has a minimalistic core with many optional extensions that can be added as needed. Flask is ideal for building medium-sized web applications. Flask provides several features that simplify web development, including routing, templating, and handling of HTTP requests and responses. It also supports integration with various databases and includes built-in tools for testing and debugging

- Python 3.x or above

The coding language used for the implementation of our system is python 3. Because of its many features, versatility, and simplicity, Python is a favourite programming language. Because of its independent platform and widespread use in the programming world, Python is the programming language that lends itself to machine learning.

Other requirements as given below:-

- coco.names file for labels of the model
- Libraries – numpy, cv2,
- IDE - Visual Studio Code



B. Hardware Requirements:-

- Laptop with latest generation i5/i7/i9 processor
- Dedicated Graphics Card is preferred
- External Wide-Angle Camera with minimum resolution of 1280x720

VI. CONCLUSION

This study has proposed a tool to calculate the distance between the two humans and alert them when they did not maintain the social distance. Efficient deep learning to automate the process of monitoring each individual and identifying in the Realtime. Different modules have been developed for human detection and social distancing recognition. Detection and tracking algorithms used to help track the people who are violating the social distancing threshold. In the future, different modern object detectors like R-CNN, Faster R-CNN, SSD, R-FCN, YOLO, etc. may be deployed to increase detection accuracy. The proposed algorithm may be set for different views through many cameras instead of single camera in the future to get more accurate results. This system can be used in live video streams and images also, CCTV for surveillance of people during pandemics. This study has proposed a tool to calculate the distance between the two humans and alert them when they did not maintain the social distance.

REFERENCES

- [1] J Dietrich, Alexander M., Keith Kuester, Gernot J. Muller, & Raphael S. Schoenle (2020) News and Uncertainty about COVID-19: Survey Evidence and Short-Run Economic Impact, Becker Institute for Economic White Paper.
- [2] Qiu, J., Shen, B., Zhao, M., Wang, Z., Xie, B., & Xu, Y. (2020). A nationwide survey of psychological distress among Chinese people in the COVID-19 epidemic, 33(2), e100213. <https://doi.org/10.1136/gpsych2020-100213>
- [3] Cucinotta, D., & Vanelli, M. (2020). WHO Declares COVID-19 a Pandemic. *Acta bio-medica :Atenei Parmensis*, 91(1), 157–160. <https://doi.org/10.23750/abm.v91i1.9397>
- [4] Subetha, T., & Chitrakala, S. (2016, February). A survey on human activity recognition from videos. In 2016 International Conference on Information Communication and Embedded Systems (ICICES) (pp. 1- 7). IEEE.
- [5] Gaeta, G. (2020). Social distancing versus early detection and contacts tracing in epidemic management. *Chaos, Solitons & Fractals*, 140, 110074.
- [6] Ghosh, P., Basheer, S., Paul, S., Chakrabarti, P., & Sarkar, J. (2020). Increased detection coupled with social distancing and health capacity planning reduce the burden of COVID-19 cases.
- [7] YOLO object detection with OpenCV. (URL: <https://www.pyimagesearch.com/2018/11/12/yoloobject-detection-withopencv/>)
- [8] Wu, J., Tang, B., Bragazzi, N. L., Nah, K., & McCarthy, Z. (2020). Quantifying the role of social distancing, personal protection and case detection in mitigating COVID-19 outbreak in Ontario, Canada. *Journal of mathematics in industry*, 10, 1-12
- [9] Social Distancing Violation Analyzer. Sanjivani Joshi, Piyush Hadge, Sanket Parsewar, Vaibhav Parsewar, Umar Shaikh Dept. of Information Technology, Sinhgad College of Engineering, Pune, India. <https://doi.org/10.5281/zenodo.7789382s>



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.423



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 **6381 907 438** **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details