



# A New UAV-Based Social Distance Detector for COVID-19 Outbreaks Reduction, Using IoT, Computer Vision and Deep Learning Technologies

Nashwan Adnan Othman<sup>1,2\*</sup>, Ilhan Aydin<sup>2</sup>

<sup>1</sup> Department of Computer Engineering, College of Engineering, Knowledge University, Erbil 44001, Kurdistan Region, Iraq

<sup>2</sup> Department of Computer Engineering, Firat University, Elazig 23200, Turkey

Corresponding Author Email: [nashwan.adnan@knu.edu.iq](mailto:nashwan.adnan@knu.edu.iq)

<https://doi.org/10.18280/ts.390607>

## ABSTRACT

**Received:** 22 September 2022

**Accepted:** 10 December 2022

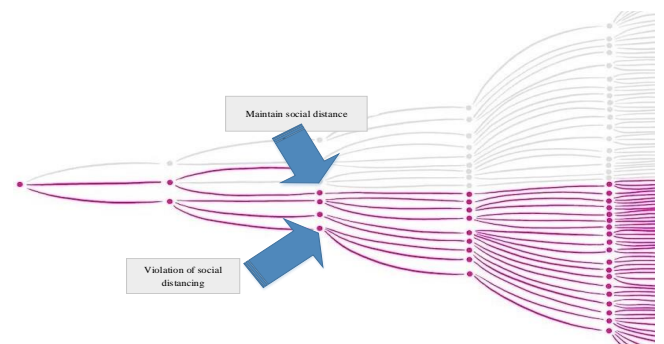
### Keywords:

COVID-19, deep learning, security system, social distance detector, computer vision, embedded system, IoT

Nowadays, we are living in a dangerous environment and our health system is under the threatened causes of Covid19 and other diseases. The people who are close together are more threatened by different viruses, especially Covid19. In addition, limiting the physical distance between people helps minimize the risk of the virus spreading. For this reason, we created a smart system to detect violated social distance in public areas as markets and streets. In the proposed system, the algorithm for people detection uses a pre-existing deep learning model and computer vision techniques to determine the distances between humans. The detection model uses bounding box information to identify persons. The identified bounding box centroid's pairwise distances of people are calculated using the Euclidean distance. Also, we used jetson nano platform to implement a low-cost embedded system and IoT techniques to send the images and notifications to the nearest police station to apply forfeit when it detects people's congestion in a specific area. Lastly, the suggested system has the capability to assist decrease the intensity of the spread of COVID-19 and other diseases by identifying violated social distance measures and notifying the owner of the system. Using the transformation matrix and accurate pedestrian detection, the process of detecting social distances between individuals may be achieved great confidence. Experiments show that CNN-based object detectors with our suggested social distancing algorithm provide reasonable accuracy for monitoring social distancing in public places, as well.

## 1. INTRODUCTION

With rising populations and COVID-19 cases, several countries' health systems are collapsing [1]. Some countries have declared a complete closure and asked their residents to stay at home and avoid public events. More than that, most countries have adopted instructions aimed at decreasing social distances between individuals everywhere [2-4]. Individuals spread germs when they are close to each other and health guidance and social distancing is the most effective non-pharmaceutical method to prevent the spread of most diseases as shown in Figure 1.



**Figure 2.** The importance of social distances

The behavior of physically separating oneself from another in order to prevent the spread of infectious diseases is known as social distancing [5]. Nowadays, it is an important technique to control and prevent the spread of COVID-19 disease as shown in Figure 2. The figure demonstrates the impact of maintaining social distance on the spread of infectious diseases like COVID-19. When individuals maintain distance from one another, it reduces the likelihood of transmission. Conversely, when a person with an infectious disease is in close proximity to others, it increases the risk of disease transmission to those nearby. This chain of transmission can quickly lead to an uncontrolled outbreak if preventive measures are not taken. Also, whenever there is an



**Figure 1.** Guidance social distance between individuals

outbreak of any infectious or almost infectious disease in any environment, it is quickly spread cause of social distances. For this purpose, we suggested a new embedded security system to predict the social distance of people in community areas for example markets or streets in real-time video frames. Deep learning and computer vision technologies allow us to gauge the social distance between individuals.

Currently, cities around the world are improving their smart city infrastructure with state-of-the-art technologies. Smart cities improve people's lives by providing 24/7 security. The basic purpose of intelligent smart cities is to deliver cost-effective infrastructure and services [6, 7]. Unmanned aerial vehicles (UAVs) provide the services required to meet the objectives of intelligent cities. Integration of UAVs with other technologies such as social distance detectors can help to develop safer intelligent city environments [8]. UAVs can help police enforce security in smart cities. Smart cities can be safer with video streaming, abnormal human activity recognition, and social distance monitoring. UAV applications are becoming more popular recently. it can manage traffic, monitor pollution, and deliver packages [9]. UAVs can help police enforce security in smart cities. Smart cities can be safer with video streaming, abnormal human activity recognition, and social distance monitoring [8, 10]. The proposed system uses UAVs to monitor social distance detectors to reduce COVID19 and other diseases.

Deep learning allows us to automate operations that previously required a lot of human intervention [11]. It can help reduce the prevalence of COVID-19 through applications in diagnosis, treatment, and monitoring [12]. As well, computer vision can also improve security in embedded smart systems. It can measure social distances that may be harmful to human health. In this paper, we employed deep learning to detect humans and computer vision to estimate unsafe social distances.

Recently, Internet of Things (IoT) has gained popularity in smart cities, security, healthcare, and transportation [7, 13]. IoT aims to share information and expertise everywhere. Smart cities can benefit from IoT. The IoT automates city functions such as monitoring and risk detection [14]. In this paper, we used IoT to send a notifications and frames to a smartphone when violate social distance detected.

This paper aims to implement an embedded powerful technique for monitoring and detecting risky social distances, which is run with a high FPS on an embedded platform and an IoT-based UAV. The goal is to reduce the intensity of COVID-19 disease spread by recognizing violated measures of social distance and notifying the system owner. The proposed embedded social distance detection system uses the Jetson nano platform, which controls peripherals. The system uses a Tello drone to capture video frames, then applies a deep learning model to detect humans. Next, computer vision functions determine all detected humans' centroids in a video frame. After that, pairwise distances are calculated. Any pairwise distance less than N pixels apart is considered a violation by the system. Subsequently, the system sends notifications and social distance violation frames to smartphones through the Internet. IoT-based Telegram is used to view activity, photos, and notifications. Healthcare organizations could benefit from the use of computer vision and deep learning algorithms to monitor, recognize, and report on violating people indoors and outdoors. Humanitarian workers and law enforcement authorities can use the suggested system to monitor public health guidelines [15].

The main contributions are summarized as the following:

1. A UAV-based smart security system to assist healthcare organizations in the case of pandemics, such as COVID19, was proposed in this work and relies on computer vision, deep learning and the internet of things.

2. Pre-trained deep convolutional neural network (CNN) model for human detection has been suggested, and this model has been improved by applying video stabilization.

3. Using IoT, this system is connected to the internet and may send notifications and photographs to connected smart devices when it detects social distances that have been violated.

4. Developing a new and fast approach by using embedded platform and compare the speed of 3 platforms.

The following is a summary of the paper: Section 1 begins with an introduction about the proposed system relevant technologies. In Section 2, the literature review is presented. Also, the proposed methodology for embedded social distance detector is explained in Section 3. In section 4, the obtained results are displayed. Then, the challenges and limitations with recommendations for future Improvements are described in Section 5. Finally, the paper is concluded with a brief discussion in Section 6.

## 2. LITERATURE REVIEW

The COVID-19 epidemic has impacted negatively on common social and economic patterns, forcing governments all around the world to take drastic measures to limit the disease's spread. Quarantine, border closures, and social distance are among the most commonly used techniques available recently. Drones also offer a low-cost and adaptable alternative for vector-borne illness eradication campaigns. Drones are being used to monitor huge epidemic contagious disease spreading, logistics and cargo delivery, and situational monitoring after COVID-19.

In order to better understand, address, and reduce the prevalence of the COVID-19 epidemic, the use of artificial intelligence and computer vision in diagnostic, therapy, and monitoring is essential [12]. There are many applications of artificial intelligence that can be used to fight against COVID-19. Further, there are many studies has designed UAV-based case studies for pandemic monitoring and social distance measurements. While fighting COVID-19, organizations have used machine learning expertise in numerous areas, including scaling customer communications, understanding how COVID-19 spreads, and accelerating research and treatments.

Several studies have employed smart technologies in the effort to combat COVID-19 and have recommended various approaches to be taken in this battle..There are different digital technologies that have been implemented to combat COVID19 and recommends adjusted ways for better results. Ardakani et al. [16] demonstrate the potential of machine learning and artificial intelligence techniques by proposing a new model that includes a rapid and valid method for diagnosing SARS-CoV-2 disease using deep CNN. Using an expert system that incorporates machine learning and artificial intelligence, the study found that using a CNN (Resnet-101) as an adjuvant tool for radiologist resulting in 86.27%, 83.33% of accuracy and specificity, respectively.

To construct an IoT base system for indoor surveillance to identify social distancing, Petrović and Kocić [17] conducted research utilizing Raspberry Pi, and the Haar cascades method in OpenCV. The system's accuracy ranged from 65% to 73%.

An ultrasonic sensor and a microprocessor are combined proposed by Naqiuddin et al. [18] to create a wearable social distancing detector that alerts users when they are too apart from each other. The system is capable of accurately detecting social distance and assisting in the movement of individuals within a specific area. Another study by Adinanta et al. [19] indicated that KNN was the best algorithm for detecting physical distance. Militante and Dionisio [20] developed a method that is 97% accurate in detecting face masks and physical distances. CNN have been used in the system to identify the use of masks within a physical distance. Yang et al. [21] used a monocular camera for density detection. Deep CNN were used in real time to determine social distance in their system. When a crowd is detected, the system can send a signal to break the crowd. Sahraoui et al. [22] established a framework for avoiding pandemic viral infections. The suggested architecture is based on a notification system for physical distance that makes use of a number of some available futuristic technologies, including as deep learning and the Internet of vehicles. A thermal and visual imaging switching camera system is installed in each vehicle. They've utilized a faster RCNN to measure and detect social distancing, which is more accurate.

Chen et al. [23] conducted research to establish a quadruped robot-based crowd detection system. The robot was able to navigate the downtown area on its own due to the use of a three-dimensional Lidar. The F-RVO motion modeling method was used in conjunction with YOLO to detect pedestrians. Saponara et al. [24], also used YOLOv2 in their research to build a low-cost embedded system. The findings of this study show that the suggested method is suitable for use in smart city surveillance systems that include detection of people, classification of social distancing, and measurement of body temperature. Bian et al. [25] conducted research on the field of study. The proximity sensor was used to create a wearable device. As a belt-mounted sensor, it is able to detect the presence of people behind the user. The sensor was able to precisely measure a distance of up to two meters between persons.

Nguyen et al. [26] explored the use of emerging technologies such as artificial intelligence wireless communication, and networking to facilitate or observe social distancing in their work. They examined the fundamental ideas, methods of assessment, models, and real-world situations of social distancing. A specific method for monitoring social distancing was introduced [27]. The approach involved using YOLO-v3 and Deepsort to find and track humans in surveillance footage, calculating the pair-wise vectorized L2 norm to determine social distancing, and calculating a violation index for instances of non-social distance behavior. Yang et al. [28] suggested an artificial intelligence-based system that uses a single camera to detect and alert individuals in real-time to violations of social distancing. Like the approach described in the study [27], this system relies on a fixed surveillance camera with a limited field of view. In contrast, the use of unmanned aerial vehicles (UAVs) for social distancing monitoring allows for greater flexibility and a wider coverage area.

### 3. THE PROPOSED METHODOLOGY FOR EMBEDDED SOCIAL DISTANCE DETECTOR

Deep learning and computer vision applications have

greatly improved and increased interest in related topics like image processing and pattern recognition. The proposed embedded social distance detector is divided into multiples stages, firstly a camera is used to achieve the video frames by using Jetson TX2 platform. Then, deep leaning models is applied to the taken frames to determine the human's body. Then, computer vision module is applied to each detected people to estimate social distance violation. Finally, if the people violate social distances, the system will send the notification and the frames to the smartphone. If there is no violate social distance detected, the program will not send any notification to the smartphone. In order to prevent the spread of the COVID19 pandemic, it is critical that this system be implemented. Figure 3 depicts the proposed system's flowchart.

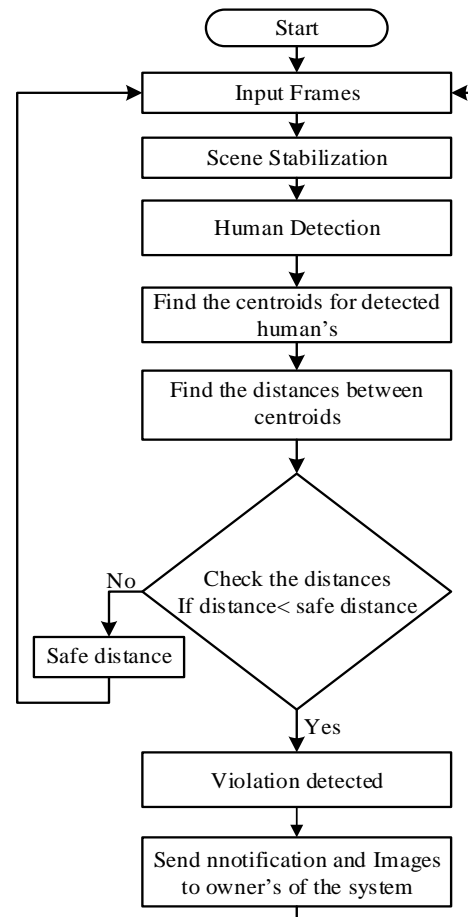


Figure 3. Flowchart for the proposed methodology

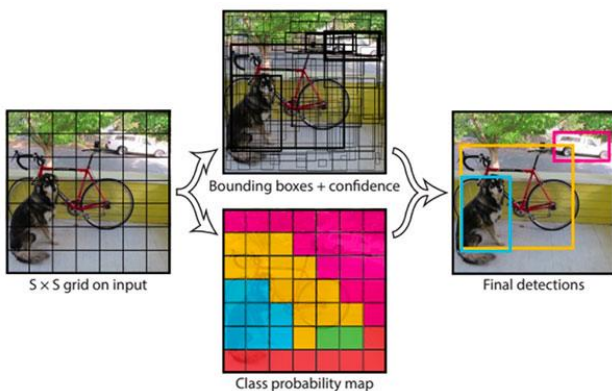
The pre-trained deep learning model detects all people in the frames in the first step. Afterward, the computer vision techniques will estimate the distances between each people. Then, we will check the distances, if the distance between any two people is smaller than N pixels, the system suggests as a violate distances. In this case, immediately the notification and violate frames will be send to the smart device based on IoT. The steps are summarized as the following:

#### 3.1 People detection stage

The initial step of the proposed system is to detect every human in a video frame. Human detection is a type of object detection. There are numerous object detection methods such as Histogram and Oriented Gradients with Linear Support

Vector Machine, Haar cascades, and deep learning-based object detectors. When it comes to object detection based on deep learning, there are three essential object detectors you will come across, which are You Only Look Once (YOLO) [29], Single Shot Detectors (SSDs) [30], and Recurrent Convolutional Neural Network (R-CNNs) [31]. The proposed system uses deep learning-based object detectors for improved human detection accuracy. R-CNNs are an instance of a two-stage detector and tend to be very accurate, but the biggest problem of this algorithm is incredibly slow. Though, both YOLO and SSDs use a one-stage detector approach. These methods handle object detection as a regression issue, using a given image to learn bounding box coordinates and class label probabilities. Generally, YOLO and SSD are significantly faster than two-stage detectors, but they are less accurate.

The proposed system is implemented using an embedded platform, and cause of that, speed is paramount. Also, the proposed implementation needed to be able of running in real time on a standard CPU. For this purpose, we will be using a pre-trained YOLO object detector model to detect all people in the video frames. YOLO is an excellent choice of a single-stage detector trained on the COCO dataset [32]. It splits the image into an  $S \times S$  grid and for every grid cell, guesses  $B$  bounding boxes, confidence for those boxes, and  $C$  class probabilities. These estimates are encoded as an  $S \times S \times (B * 5 + C)$  tensor. Figure 4 illustrates the YOLO object detector pipeline.



**Figure 4.** A simple diagram of how the YOLO object detector works [29]

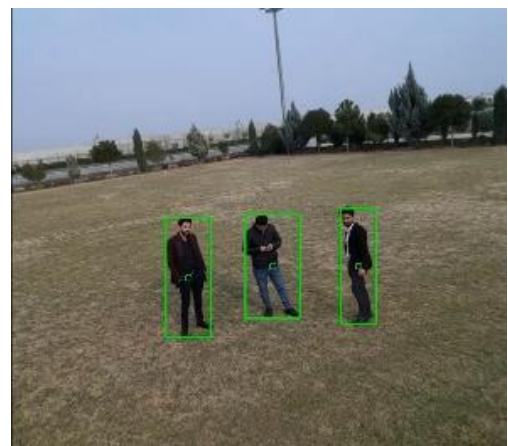
This stage will consider use of YOLOv3 [33], specifically YOLO trained on the COCO dataset, which contains 80 labels. The YOLO model can detect many types of objects, but this index is only for the person class because we won't be looking at anything else. Even though it's significantly larger than previous models, YOLOv3 appears to be the most advanced of the YOLO family of object detectors. Utilizing the pre-trained YOLO object detection model, we get the bounding boxes and associated probabilities.

After applying the object detector model, we most often detect multiple bounding boxes surrounding the object you're looking for. Before returning all the bounding boxes, we should apply Non-Maximum Suppression (NMS) to remove bounding boxes that are too close around each other. The NMS algorithm is often used in object detection tasks to select the best bounding box among multiple detections of the same object. The NMS algorithm considers two factors when selecting the best bounding box for detecting an object: the objectiveness score assigned by the model, and the overlap (also known as the intersection over union or IOU) between

bounding boxes. The objectiveness score indicates how confident the model is that the desired object is present in the bounding box. Among all the bounding boxes containing the object, the NMS algorithm selects the one with the highest objectiveness score as the best bounding box. It then removes any other bounding boxes that have a high overlap with the selected box. The following steps outline the process for using NMS to select the best bounding box:

- Select the bounding box with the highest objectiveness score. The objectiveness score is a measure of how confident the object detection model is that a given bounding box actually contains an object.
- Calculate the overlap (Intersection over Union, or IoU) between the selected bounding box and the other bounding boxes. The IoU is a measure of the overlap between two bounding boxes, and is calculated as the intersection of the boxes divided by the union of the boxes.
- Remove any bounding boxes that have an IoU below the NMS threshold. The NMS threshold is a predetermined value that determines the minimum acceptable overlap between two bounding boxes. If the overlap between two boxes is below the threshold, one of the boxes is considered a false positive and is removed.
- Select the next highest objectiveness score and repeat the process of comparing overlap and removing boxes below the threshold until all bounding boxes have been considered.

The final detection is produced after this process is complete, and includes only the bounding boxes that have passed the NMS threshold and were deemed the most confident detections by the object detection model. Figure 5 shows the people detection stage after applying YOLO object detector model and NMS.

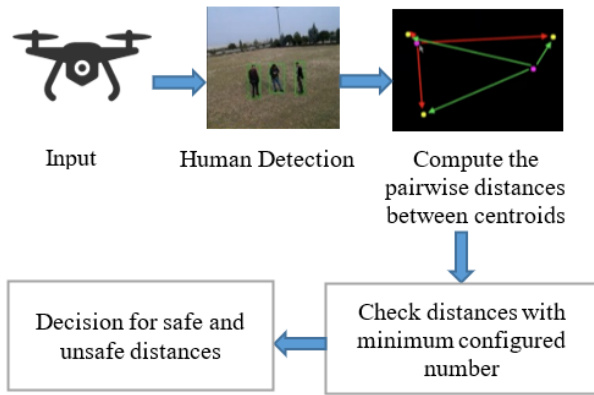


**Figure 5.** Human detection stage

### 3.2 Social distance detector stages

After detecting all people in the input frames, the second step is to build a social distance detector. Before building social distance detector, we will assume that at least two people were detected in the input frames. The social distance detector algorithm is responsible for looping over video frames and ensuring that people are not too close to each other during a pandemic outbreak. Social distance detector stage consists of three steps which are determining the centroids for every detected human's, computing the pairwise distances among each centroid, and then checking if the distances between any two people are smaller than  $N$  pixels or not.

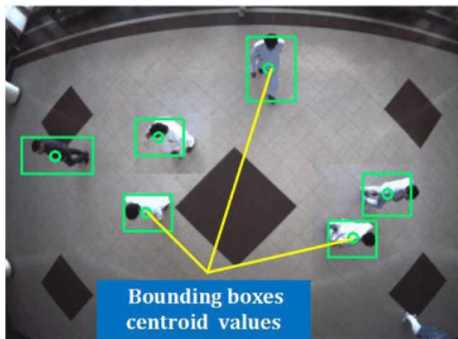
Figure 6 shows the applying steps for social distance detection system.



**Figure 6.** Block diagram of the proposed system

### 3.2.1 Finding the centroids of every detected human

The first stage of the social distance detector is to find the centroids of the detected humans. By using the extracted bounding box coordinates from the previous stage, we can derive the top left coordinates to determine the centroids of the humans. Then, we will assume that at least two people were detected in the input frames. Formerly, we set the green color to draw the bounding box over detected each person and determine the centroids on input frames. Figure 7 demonstrated the detected human's after determining centroids of the detected humans.



**Figure 7.** Determine the centroids for each detected human

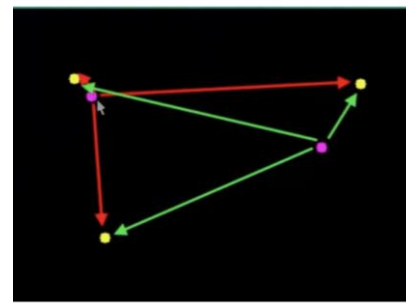
### 3.2.2 Computing the pairwise distances between all centroids

After ensuring there are at least two people detections and determining the centroids for all detected persons, we will be able to analyze the pairwise distances between humans in the input images. Since the matrix is symmetrical, the upper triangle of the distance matrix is looped over. The pairwise distances between the people will be computed by using the Euclidian distance metric as shown in Eq. (1).

$$D[i, j] = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (1)$$

where,  $q_i$  and  $p_i$  are the vector of two human centroids. In this stage, the Euclidian distance equation calculates the distance between the centroid of a human and the centroid of another human detected in the same frame. The distances between humans detected in the frame are used in the next step to

identify whether there are any violations of the required social distance. However, the accuracy of this method may be affected by factors such as the altitude and angle of the UAV camera. Lower distances also result in superior violations. Euclidian distance equation was utilized because it is fast to compute, easy to implement, and works well for social distance detector algorithms. Figure 8 displays an example about pairwise distances among each pair of the centroids.



**Figure 8.** Calculating the pairwise distances among centroids

### 3.2.3 Check Distance Matrix with minimum configured number

After calculating the pairwise distances among every centroid, we will define the minimum pixel distance that must be kept between people in order to maintain social distance. Check to see if any two people are less than  $N$  pixels apart based on these distances. We extract the confidence of the current detected object after looping over each of the detections. After this, we ensure that the current detection is a person and the minimum confidence is met or exceeded by utilizing the following equation:

$$D[i, j] < MCD \quad (2)$$

where,  $D[i, j]$  is the distance between each centroid points and  $MCD$  is the minimum configured distance. Our violation set is then initialized after determining whether or not people are keeping a safe social distance from one other. This set keeps a list of those who are violating the public health workers' guidelines for social distance. To see whether any centroid pairs violate our minimum social distance set and are less than the configured number of pixels, perform a check on the distance between any two centroid pairs. In the case that two people are too close together, we add them to the violate set. Moreover, our violation set is going to be updated with the indexes of the centroid pairs after we've added them to the list. Then, red color of the annotation will be initialized to draw a red bounding box around the violated persons and draw a red line between violated persons. The output frame should then display the total number of social distance violations. Finally, the violations will be sent to the smartphone based IoT.

### 3.3 IoT API

The Telegram Bot API is an HTTP-based interface for developers that want to build bots for Telegram [34]. Bots are third-party applications that run inside of Telegram. It's possible to turn a bot into a smart newspaper, and any published relevant content will be sent to the client immediately. In addition, a bot can easily enhance Telegram chats with external service content, like Image Bot, Audio Bot, and Video Bot. In order to create a TeleBot account, no phone

number has ever been required. The Newbot command can be used to generate a new bot. The Bot Father will ask for a name and username, and then produce a token for the new bot. For example, a token may be produced as follows: 5040033219:AAFytvWnkjeb5JNxFkX-9SFIMyO 2OIdny4 to authorize the bot and submit requests to the TeleBot API. Figure 9 shows the bot is built and BotFather shows you the token. Python TeleBot API package is required for using the Telegram Bot API on a Jetson nano. We can use pip install python-telegram-bot from the shell to achieve it. It can be used in our scripts after installation. In the proposed system, TeleBot API used to transmit messages from Jetson nano to our smartphone.



Figure 9. BotFather generate token

#### 4. EXPERIMENTAL RESULTS

The recommended system uses computer vision and deep learning to detect social distance in real-time based on UAV. The proposed system utilizes Python libraries as a programming language. The proposed methods in this study were implemented on a low-cost embedded platform using the NVidia Jetson Nano Developer Kit card, which is a low-cost embedded platform. Furthermore, Tello Drone used to grab the frames from above. A low-cost experimental environment has been created to new solutions in this area. The Nvidia Jetson Nano card is not connected to a monitor, wireless keyboard, or mouse in this experiment. Instead of, the proposed system will run on startup automatically. Tello drone has been connected to Jetson Nano via a Wi-Fi that included on its main board. Also, we used power bank to power the Jetson Nano Developer Kit. The established real-time system for social distance detector is given in Figure 10. A smart security system based on social distance detectors has been integrated with the existing system to detect violations of personal area between people via smartphone application. An UAV grab the frames from above and immediately these frames will be accessed through WIFI and processed with a powerful embedded platform.



Figure 10. Experimental setup

Images are successfully captured and sent through IoT to a smartphone utilizing an embedded platform when a distance violation is noticed. All frames have been processed through the algorithm, and the system can detect humans. A deep learning model is used to identify humans, while a computer vision technique estimates social distance. In the proposed system, there are six steps that are done. The steps include capturing frames, human detection, determine the centroids of each human, compute the pairwise distances between all centroids, compare the distances with minimum configured number to predict the violate distances with sending notifications and violation pictures as alert to the user's smartphone. IoT will be activated only if a social distance is violated. When the system detects a violation distance, the Jetson Nano's Wi-Fi wireless allows users to see the activity and show violation distance images immediately on the mobile application "Telegram," as shown in Figure 11.

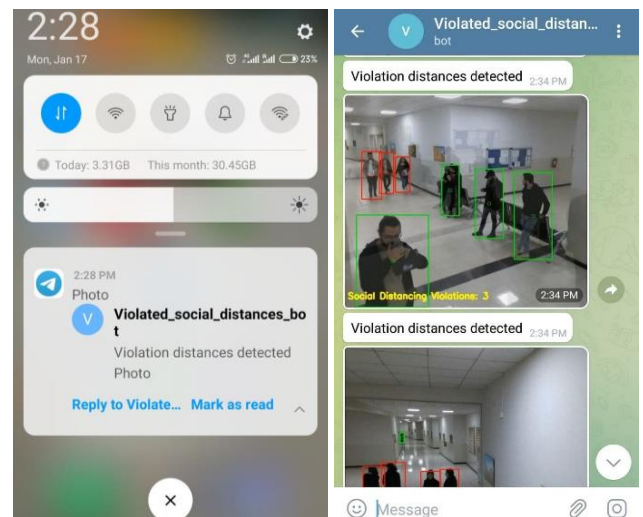


Figure 11. Warning notifications and violated social distance image on a smartphone

The system expects the distances between every people effectively. If the distances are less than the configured threshold, the system will alert violation noticed as shown in Figure 11. As shown in Figure 12, the system also detects and predicts non-violated distances without sending a notification to the IoT API. As displayed in the Figure 13, the proposed system has properly detected social distances on real time data.



Figure 12. A normal-distance result (non-violation)



**Figure 13.** The visualization results of our method

The proposed social distance detector system may not always provide accurate results and has certain limitations. These limitations are caused by a range of factors, which will be examined in more detail in the following chapter. The process of visual UAV-based social distance detection is tough and the actual distances between the people can't be trusted to figure out the measurements. Instead, the centroids among them should be measured. Estimating distances is a challenging problem. Reasons that determine the cause of this, including camera angles, human pose appearance, UAV altitude, distances between the UAV and their targets, speed and flight position, camera movements, view invariance and human actions. These findings show that several social distance detection methods make mistakes due to the abovementioned challenges.

The suggested approach can predict the social distance between people automatically with a good level of accuracy. The video stabilization method performed well and guaranteed improvement over the pre-trained model. This method used to reduce the ego-motion of the acquisition platform that happens

when we take frames using UAV. By this way, we can detect the human body more accurately. Results from UAV applications like human detection and social distance estimation can be improved by the use of video stabilization techniques in a variety of circumstances. Table 1 shows the accuracy comparison for the pre-trained YOLO model with video stabilization.

**Table 1.** Accuracy comparison for proposed system

Proposed Method for Human Detection	Accuracy
With image stabilization	93.50%
Without image stabilization	91.02%

We compared the performance of the suggested embedded system utilizing the Jetson Nano Developer Kit, Raspberry Pi 3, and Jetson TX2. The proposed system comprises of loading one deep learning model, that is the yolo object detector. Table 2 shows the speed comparison of different embedded platforms.

**Table 2.** Speed comparison of embedded platforms

Embedded Platform	Frames Per Second (FPS)	GPU
Jetson TX2	~33.09	Yes
Jetson Nano	~14.6	Yes
Raspberry PI3	~0.5	No

Since we need to implement an embedded real time system with a lowest cost, the main platform for the proposed system is Jetson Nano Developer Kit because it's smaller than Jetson TX2 and cheaper. As well, the Jetson Nano platform can reach a reasonable speed, which is about 14 FPS. Conversely, Raspberry Pi 3 platform can't achieve an acceptable speed to run the suggested system. As showed, a Jetson Nano is 30 times quicker than Raspberry Pi 3. On the other hand, Jetson TX2 can obtain approximately 33 FPS, but the problems in using this platform are it is bigger than Jetson nano and more expensive. Considering these reasons, we preferred using Jetson nano above Jetson TX2 to perform the proposed system at a low cost.

## 5. CHALLENGES, LIMITATIONS AND FUTURE IMPROVEMENTS

Accuracy and precision are essential for this application because it can be used in any working environment. An increase in the number of false positives may cause discomfort among those being observed. There are several limitations to the use of UAVs, including their range, altitude and angle of the camera, the weather conditions they can fly in, regulatory restrictions, safety and privacy concerns, and the cost and availability of the necessary hardware and software. These limitations may affect the deployment and use of UAVs in various application scenarios. Moreover, some potential drawbacks to using UAVs with AI and computer vision include inaccurate or unreliable results, difficulty capturing high-quality images, cost and availability of necessary equipment, and potential legal or regulatory limitations.

There are many other factors that affect the accuracy of the social distance detector system, including camera calibration, camera angles, human pose appearance, UAV altitude, distances between the UAV and their targets, speed and flight position, camera movements, view invariance and human

actions. Because the proposed social distancing detector didn't use proper camera calibration, we couldn't simply convert distances in pixels to actual measurable units. So, the first thing we need to do to improve our social distancing detector is calibrate the camera and the position of the UAV. Doing so will give you better results and let you figure out how many actual, measurable units there are (rather than pixels). As well, we should think about changing your viewing angle from top to bottom. In order to get a more accurate representation of social distance using OpenCV, you may use a perspective transform or stereo computer vision. Even if it is more accurate, such a system is more complicated and is not always required. We can apply distance calculations to the top-down view of pedestrians, resulting in a better approximation of distance. Finally, the factors show that many errors made by social distance detection systems are due to the extreme difficulty in seeing the conditions mentioned above.

Another recommendation to increase the accuracy of the proposed system is to enhance the person detection process. The YOLO implementation in OpenCV is slow not because of the model, but because of the additional pre-processing that is necessary. To further speed up your pipeline, consider using more transfer learning techniques or training another model specifically to detect people running on GPU, which will significantly improve your frame rate. Moreover, the performance of the proposed system can be improved by using embedded platforms like Jetson AGX Xavier which is approximately 55 times faster than Jetson Nano. Furthermore, by utilizing more training examples and utilizing more data augmentation methods during the training phase, we can improve the accuracy of human detection phase. Generally, the proposed system can be optimized for different environments in the future. Various detection and tracking algorithms can help track down the person or persons who are violating the social distancing threshold.

## 6. CONCLUSIONS

This paper presents an UAV-based social distance monitoring framework to reduce COVID-19 outbreaks, using the determination of violating distances that are more affected by the disease in endemic regions. The proposed system uses deep learning, computer vision, and IoT technologies to identify humans, predict social distancing, and send the violation distance to the smartphone. An enhancement social distance approximation technique was developed to improve the results of pre-trained deep learning models by using video stabilization. The pre-trained YOLOv3 model, trained on the COCO dataset, is used for people detection. The detection model generates a bounding box with centroid coordinates, and the distance between the centroids of different bounding boxes is measured utilizing the Euclidean distance equation. An approximation of the physical distance to the pixel is used to check social distance violations between people with a defined threshold, which is used to check whether or not the distance value violates the minimum specified social distance. Afterwards, IoT is used to send pictures of the people that are violating social distancing rules, together with a notification to the smartphone to indicate the violations happening in the specific area. The proposed system was tested, and the result showed a satisfactory performance. Based on the performed test, it can conclude that the framework effectively detects individuals walking too close and violating the rule of safe

social distance. A pre-trained model also achieves a human detection accuracy of approximately 92% when using YOLO v3. In addition, the proposed system was compared with three different embedded platforms. As a proof-of-concept for a low-cost social distancing detector, the proposed method worked well in real-time by using an embedded platform such as the Jetson nano and Tello drone.

## REFERENCES

- [1] WHO, Rolling updates on coronavirus disease (COVID-19). <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen/>, accessed on July 31, 2020.
- [2] Jain, G., Mittal, D., Thakur, D., Mittal, M.K. (2020). A deep learning approach to detect Covid-19 coronavirus with X-ray images. *Biocybernetics and Biomedical Engineering*, 40(4): 1391-1405. <https://doi.org/10.1016/j.bbe.2020.08.008>
- [3] Mahase, E. (2020). Coronavirus: Covid-19 has killed more people than SARS and MERS combined, despite lower case fatality rate. *BMJ*, 368: m641. <https://doi.org/10.1136/bmj.m641>
- [4] Khan, A.I., Shah, J.L., Bhat, M.M. (2020). CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. *Computer Methods and Programs in Biomedicine*, 196: 105581. <https://doi.org/10.1016/j.cmpb.2020.105581>
- [5] Social distancing, (n.d.). [https://en.wikipedia.org/wiki/Social\\_distancing](https://en.wikipedia.org/wiki/Social_distancing).
- [6] Aneedge-basedplatformfordynamicSmartCityapplications.pdf, (n.d.).
- [7] IoT in smart cities\_ A contemporary survey \_ Elsevier Enhanced Reader.pdf, (n.d.).
- [8] Mohamed, N., Al-Jaroodi, J., Jawhar, I., Idries, A., Mohammed, F. (2020). Unmanned aerial vehicles applications in future smart cities. *Technological Forecasting and Social Change*, 153: 119293. <https://doi.org/10.1016/j.techfore.2018.05.004>
- [9] Menouar, H., Guvenc, I., Akkaya, K., Uluagac, A.S., Kadri, A., Tuncer, A. (2017). UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3): 22-28. <https://doi.org/10.1109/MCOM.2017.1600238CM>
- [10] Othman, N.A., Aydin, I. (2021). Challenges and limitations in human action recognition on unmanned aerial vehicles: A comprehensive survey. *Traitement du Signal*, 38(5): 1403-1411. <https://doi.org/10.18280/ts.380515>
- [11] Othman, N.A., Aydin, I. (2018). A new deep learning application based on movidius ncs for embedded object detection and recognition. In 2018 2nd international symposium on multidisciplinary studies and innovative technologies (ISMSIT), pp. 1-5. <https://doi.org/10.1109/ISMSIT.2018.8567306>
- [12] Othman, N.A., Al-Dabagh, M.Z.N., Aydin, I. (2020). A new embedded surveillance system for reducing COVID-19 outbreak in elderly based on deep learning and IoT. In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), pp. 1-6.



- <https://doi.org/10.1109/ICDABI51230.2020.9325651>
- [13] Boursianis, A.D., Papadopoulou, M.S., Diamantoulakis, P., Liopa-Tsakalidi, A., Barouchas, P., Salahas, G., Karagiannidis, G., Wan, S., Goudos, S.K. (2022). Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review. *Internet of Things*, 18: 100187. <https://doi.org/10.1016/j.iot.2020.100187>
- [14] Othman, N.A., Aydin, I. (2018). A face recognition method in the Internet of Things for security applications in smart homes and cities. In 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), pp. 20-24. <https://doi.org/10.1109/SGCF.2018.8408934>
- [15] Deep learning-enabled medical computer vision \_ Enhanced Reader.pdf, (n.d.).
- [16] Ardakani, A.A., Kanafi, A.R., Acharya, U.R., Khadem, N., Mohammadi, A. (2020). Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine*, 121: 103795. <https://doi.org/10.1016/j.compbiomed.2020.103795>
- [17] Petrović, N., Kocić, Đ. (2020). IoT-based system for COVID-19 indoor safety monitoring. *IcETRAN Belgrade*.
- [18] Naqiyuddin, F.A., Mansor, W., Sallehuddin, N.M., Johari, M.M., Shazlan, M.A.S., Bakar, A.N. (2020). Wearable social distancing detection system. In 2020 IEEE International RF and Microwave Conference (RFM), pp. 1-4. <https://ieeexplore.ieee.org/document/9344786>
- [19] Adinanta, H., Kurniawan, E., Prakosa, J.A. (2020). Physical distancing monitoring with background subtraction methods. In 2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), pp. 45-50. <https://doi.org/10.1109/ICRAMET51080.2020.9298687>
- [20] Militante, S.V., Dionisio, N.V. (2020). Deep learning implementation of facemask and physical distancing detection with alarm systems. In 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), pp. 1-5. <https://doi.org/10.1109/ICVEE50212.2020.9243183>
- [21] Yang, D., Yurtsever, E., Renganathan, V., Redmill, K.A., Özgüner, Ü. (2021). A vision-based social distancing and critical density detection system for COVID-19. *Sensors*, 21(13): 4608. <https://doi.org/10.3390/s21134608>
- [22] Sahraoui, Y., Kerrache, C.A., Korichi, A., Nour, B., Adnane, A., Hussain, R. (2020). Deepdist: A deep-learning-based iov framework for real-time objects and distance violation detection. *IEEE Internet of Things Magazine*, 3(3): 30-34.
- [23] Chen, Z., Fan, T., Zhao, X., Liang, J., Shen, C., Chen, H., Manocha, D., Pan, J., Zhang, W. (2021). Autonomous social distancing in urban environments using a quadruped robot. *IEEE Access*, 9: 8392-8403. <https://doi.org/10.1109/ACCESS.2021.3049426>
- [24] Saponara, S., Elhanashi, A., Gagliardi, A. (2021). Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19. *Journal of Real-Time Image Processing*, 18(6): 1937-1947. <https://doi.org/10.1007/s11554-021-01070-6>
- [25] Bian, S., Zhou, B., Bello, H., Lukowicz, P. (2020). A wearable magnetic field based proximity sensing system for monitoring COVID-19 social distancing. In *Proceedings of the 2020 International Symposium on Wearable Computers*, pp. 22-26. <https://doi.org/10.1145/3410531.3414313>
- [26] Nguyen, C.T., Saputra, Y.M., Van Huynh, N., Nguyen, N.T., Khoa, T.V., Tuan, B.M., Nguyen, D.N., Hoang, D.T., Vu, T.X., Dutkiewicz, E., Chatzinotas, S., Ottersten, B. (2020). Enabling and emerging technologies for social distancing: a comprehensive survey and open problems. *arXiv preprint arXiv:2005.02816*. <https://doi.org/10.48550/arXiv.2005.02816>
- [27] Punn, N.S., Sonbhadra, S.K., Agarwal, S., Rai, G. (2020). Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. *arXiv preprint arXiv:2005.01385*. <http://arxiv.org/abs/2005.01385>
- [28] Yang, D., Yurtsever, E., Renganathan, V., Redmill, K. A., Özgüner, Ü. (2021). A vision-based social distancing and critical density detection system for COVID-19. *Sensors*, 21(13): 4608. <https://doi.org/10.3390/s21134608>
- [29] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [30] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21-37. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- [31] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28. <https://doi.org/10.1109/TPAMI.2016.2577031>
- [32] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740-755. [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
- [33] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. 2018. *arXiv preprint arXiv:1804.02767*, 20. <http://arxiv.org/abs/1804.02767>
- [34] Telegram Bot API, (n.d.). <https://core.telegram.org/bots/api/>, accessed December 30, 2021.