electronics

MDPI

Article

DepTSol: An Improved Deep-Learning- and Time-of-Flight-Based Real-Time Social Distance Monitoring Approach under Various Low-Light Conditions

Adina Rahim [1], Ayesha Maqbool [2], Alina Mirza [1], Farkhanda Afzal [1] and Ikram Asghar [3,*]

[1] MCS, National University of Sciences and Technology, Islamabad 44000, Pakistan; arahim.mscs25mcs@student.nust.edu.pk (A.R.); alina.mirza@mcs.edu.pk (A.M.); farkhanda@mcs.edu.pk (F.A.)
[2] NBC, National University of Sciences and Technology, Islamabad 44000, Pakistan; ayesha.maqool@nbc.nust.edu.pk (A.M.)
[3] Faculty of Computing, Engineering and Science, University of South Wales, Pontypridd CF37 1LP, UK
* Correspondence: ikram.asghar@southwales.ac.uk

Abstract: Social distancing is an utmost reliable practice to minimise the spread of coronavirus disease (COVID-19). As the new variant of COVID-19 is emerging, healthcare organisations are concerned with controlling the death and infection rates. Different COVID-19 vaccines have been developed and administered worldwide. However, presently developed vaccine quantity is not sufficient to fulfil the needs of the world's population. The precautionary measures still rely on personal preventive strategies. The sharp rise in infections has forced governments to reimpose restrictions. Governments are forcing people to maintain at least 6 feet (ft) of safe physical distance to stay safe. With summers, low-light conditions can become challenging. Especially in the cities of underdeveloped countries, where poor ventilated and congested homes cause people to gather in open spaces such as parks, streets, and markets. Besides this, in summer, large friends and family gatherings mostly take place at night. It is necessary to take precautionary measures to avoid more drastic results in such situations. To support the law and order bodies in maintaining social distancing using Social Internet of Things (SIoT), the world is considering automated systems. To address the identification of violations of a social distancing Standard Operating procedure (SOP) in low-light environments via smart, automated cyber-physical solutions, we propose an effective social distance monitoring approach named DepTSol. We propose a low-cost and easy-to-maintain motionless monocular time-of-flight (ToF) camera and deep-learning-based object detection algorithms for real-time social distance monitoring. The proposed approach detects people in low-light environments and calculates their distance in terms of pixels. We convert the predicted pixel distance into real-world units and compare it with the specified safety threshold value. The system highlights people violating the safe distance. The proposed technique is evaluated by COCO evaluation metrics and has achieved a good speed–accuracy trade-off with 51.2 frames per second (fps) and a 99.7% mean average precision (mAP) score. Besides the provision of an effective social distance monitoring approach, we perform a comparative analysis between one-stage object detectors and evaluate their performance in low-light environments. This evaluation will pave the way for researchers to study the field further and will enlighten the efficiency of deep-learning algorithms in timely responsive real-world applications.

Keywords: social distancing; cyber-physical system; IoT towards COVID-19; machine learning; social IoT; computer vision; DepTSol; deep learning; artificial intelligence

Citation: Rahim, A.; Maqbool, A.; Mirza, A.; Afzal, F.; Asghar, I. DepTSol: An Improved Deep-Learning and Time-of-Flight-Based Real-Time Social Distance Monitoring Approach under Various Low-Light Conditions. *Electronics* **2022**, *11*, 437. https://doi.org/10.3390/electronics11030458

Academic Editor: Felipe Jiménez

Received: .30 December 2021
Accepted: 24 January 2022
Published: 3 February 2022

1. Introduction

COVID-19 caused by SARS-CoV2 originated from Wuhan, China, and created a catastrophe in 219 countries [1]. On 23 December 2021, World Health Organization (WHO) declared it a pandemic when it spread in 114 countries with 0.5 million active daily cases [2,3,4]. To date, 274,628,461 cases have been confirmed by WHO worldwide, with a death toll of 5,358,978 [1]. As of December 2021, many different vaccines were approved for public use and 8,387,658,165 doses were administered worldwide [5]. However, with the new drug-resistant variants such as Omicron and evidence of re-infection, vaccines are not sufficient to counter the pandemic. The preventive measures still rely on personal prevention strategies suggested by WHO, e.g., wearing facemasks, avoiding large gatherings and poorly ventilated places, regular handwashing, cleansing and disinfecting touched surfaces daily, and maintaining at least 6 ft of safe physical distance. Even after vaccination, social distancing is still recommended as the best solution for infection avoidance [6].

Social distancing is the means of maintaining a safe distance in both indoor and outdoor environments. As COVID-19 generally transfers between people, especially when an infected person sneezes, talks, coughs, or physically touches another person, the chances of that person becoming infected are increased. At the current stage, as the fifth wave of COVID-19 is emerging in various countries, it is necessary to take precautions to protect ourselves and our families by maintaining a safe physical distance. It has been noticed that physical distancing can reduce the increased number of infected people and help in reducing the burden of healthcare departments, especially in underdeveloped countries where there is a shortage of healthcare resources. In a study, Kylie et al. [7] investigated the correlation between transmissibility and movement based on daily reported cases from Mainland China. They found that the correlation decreases as people's movement decreases within different provinces of China. As a result, China successfully exited its lockdown early. The imposition of a complete lockdown is not a practical solution, as it can lead to an economic crisis. In this situation, a proper strategic plan is needed. Institutes are required to open following a feasible physical distancing strategy. Automated cyber-physical distance monitoring systems can overcome the burden of officials. With the arrival of summer, low-light conditions can become a problem, especially in the cities of underdeveloped countries, where, due to poor ventilated and congested homes, people are often seen in parks, streets, and markets. Besides this, large social gatherings take place at night. In such situations, it is necessary to ensure a safe physical distance between people. By emphasising the same scenario, our main contributions are the following:

- We develop an efficient deep-learning-based physical distance monitoring approach in collaboration with ToF technology to monitor physical distancing under various low-light conditions.
- In comparison to the social distance monitoring solution provided by Adina et al. [8] in the DepTSol model, the limitation of monitoring people at a fixed camera distance in a given environment is addressed by monitoring people at varying camera distances.
- In this article, we evaluate the performance of the newly released, scaled-YOLOv4 algorithm under various low-light environments and perform a comparative analysis between seven different one-stage object detectors in low-light scenarios without applying any image cleansing or visibility enhancement techniques. In the literature, no other studies analyse the performance of deep learning algorithms in the context of low-light scenarios. Based on comparative analysis, in terms of both speed and accuracy, we choose the best algorithm for the implementation of our real-time social distance monitoring framework.
- The proposed technique is not only limited to monitoring social distancing at night, but it is also implementable in generic low-light environments for the detection and tracking of people, as likely violation of safety measures occur at night.

2. Literature Review

The researchers have made remarkable contributions and presented effective solutions to deal with the COVID-19 pandemic. Notable work has been done in the literature on social distance monitoring after it was declared an effective solution for the prevention of disease. Prem et al. [9] used synthetic location-specific contact patterns in Wuhan to monitor the effect of population mixing on outbreaks. They simulated the outbreak trajectory by using the susceptible-exposed-infected-removed (SEIR) model. Their study showed that the mixing of people of different age groups causes different effects in the spread of disease. Young people are found to be less infected than older people, and physical distancing was shown to be an utmost reliable practice to reduce the epidemic peak in Wuhan, China. Adolph et al. [10] analysed the effects of outbreaks in the USA and further evaluated the decision of different politicians and policymakers concerning social distancing. The results were contradictory, which delayed the lockdown and resulted in the spread of COVID-19. The pandemic has triggered a dire need for technology-oriented digital healthcare solutions. To promote social distancing, many governments have utilised a social IoT system comprised of infrared thermometers and self-temperature scanners. To educate the people about the importance of social distancing, the Qatari government [9] employed security robots in various residential and public areas. In Singapore, a temperature screening system was introduced based on the artificial powered thermal scanner SPOTON [11]. The Kuwaiti government [11] introduced an application named 'Shlonik' to monitor people in quarantine. Indonesia launched a robot medical assistance system to limit the contact between patients and medical staff. The robots can carry up to 50 kg of items such as medicine, clothes, and food to a patient's room [11]. Similarly, Iran developed a mobile application for electronic self-tests of COVID infection [11]. Kyrgyzstan created a website for its citizens [11]. The people who need food assistance can register their needs online and obtain food at their doors. The Ministry of Health and Education provided a free program named 'MASK' [11]. This application enables people to see contaminated areas on the map based on the places highly visited by infected people.

In the past few decades, the detection of humanoid forms by using deep learning algorithms have been widely practised. In the literature, different deep-learning-based research studies have been conducted for the automation of social distance monitoring by detecting and monitoring people with high accuracy. Punn et al. [12] presented a deep-learning-based framework and implemented it with surveillance cameras for the automation of physical distance monitoring. The YOLOv3 [13] algorithm is utilised in collaboration with the deep-sort technique for real-time object detection and tracking. In the same background, Sahraoui et al. [14] used Social Internet of Vehicles (SIoV) technology with a Faster RCNN [15] algorithm to monitor physical distancing and alert generation. According to this study, every vehicle is equipped with cameras that capture images, objects in images are detected by Faster RCNN, and notifications regarding violations are sent through an advertisement board. The model's efficiency was evaluated by vehicle-to-infrastructure communication and found very effective. Similarly, Bouhlel et al. [16] introduced two different methods for measuring physical distancing. In the first method, they estimated the crowd's density and classification of ariel frame patches, whereas, in the second method, they used deep learning for detection and tracking. They tested their model on three different datasets and achieved good accuracy. Recently, Adina et al. [8] presented a real-time social distance monitoring strategy in collaboration with deep learning and ToF technology. The authors utilised the YOLOv4 [17] algorithm for real-time people detection and suggested a camera calibration approach for social distance monitoring at a fixed camera distance. The authors mainly focused on low-light scenarios. The model can observe people and show their relevant distance in real-world units with high accuracy and a minimal error rate.

In the drastic situation of COVID-19, Social IoT, deep learning, and computer vision have played a vital role. Researchers have made contributions and provided efficacious, deep-learning-based social distance monitoring solutions, as discussed above, but low-

light conditions are yet to receive due attention. We focused on low-light scenarios and presented an efficient social distance monitoring approach by maintaining a good speed–accuracy trade-off, but the technique was limited to monitoring people at a fixed camera distance in a given environment [8]. By considering this research gap, in this article, a real-time physical distance monitoring approach was introduced by maintaining optimal performance in terms of both speed and accuracy. The proposed approach maintains high privacy standards. Instead of targeting individuals when a safety breach is detected, we propose general voice warnings via speakers.

3. Overview of Scaled-YOLOv4 Algorithm

We have seen a vast number of applications of computer vision and deep-learning-based algorithms in the current era, such as fraud detection [18–20], face recognition [21], theft detection [22,23], pedestrian detection [24–26], traffic monitoring [27–29], and business analytics [30,31]. All of these applications need to be trained on large datasets for effective results. These vast datasets require massive computing capabilities such as GPU, cloud computing facilities, single embedded devices, and large clusters for training. Model scaling plays a vital role in the design of an effective object detector with optimal speed–accuracy features. To make training easier and suitable on different devices, the most common practice is to change the number of convolutional filters, i.e., the width of the backbone, and the number of convolutional layers, i.e., the depth of the backbone, in convolutional neural networks (CNNs). By following the same practice, on 22 February 2021, Wang et al. [32] introduced a scalable-YOLOv4 model, where they showed that a YOLOv4 object detector based on a cross-stage-partial (CSP) framework can be easily scaled up or down and can be easily applied to both small and large networks by maintaining a good speed–accuracy trade-off.

After the successful execution of model scaling, the next phase is to monitor quantitative and qualitative elements that will change. These elements incorporate cost, inference time, and accuracy. The qualitative elements have different effects than quantitative elements depending on the user database or equipment. During the design of effective model scaling strategies, it is ensured that, whether the model is scaled up or down, the quantitative cost can be easily managed accordingly. The authors of the scaled-YOLOv4 model have analyzed different CNN models (ResNet [33], ResNext [34], and Darknet [13]) and monitored their quantitative cost by performing upscaling and downscaling. From the experiments, they found that the change in the number of layers, network size, and width increases the computational cost, whereas their proposed approach of converting CNNs to CSPNet can effectively minimise the floating-point operations per second on ResNet, ResNext, and Darknet by 23.5%, 46.7%, and 50.0%, proving to be the overall best model scaling approach so far. The architecture of the scaled-YOLOv4 model is shown in Figure 1.

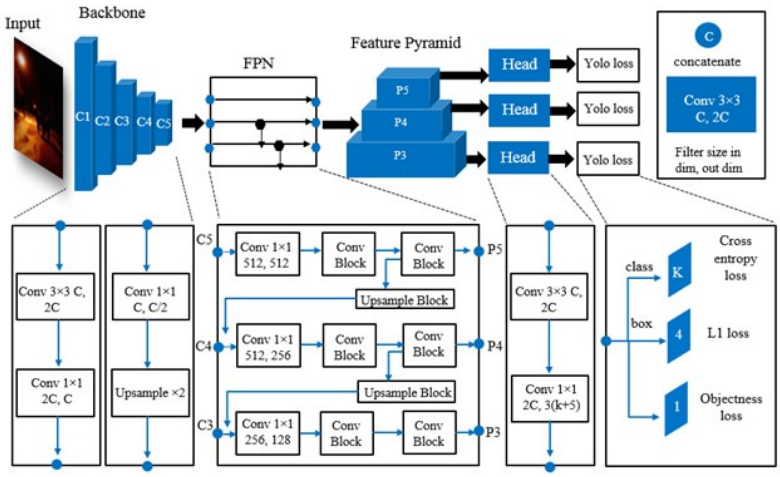


Figure 1. Diagrammatic representation of the scaled-YOLOv4 architecture.

3.1. CSP-ized YOLOv4

Authors of scaled-YOLOv4 have designed scaling algorithms for all general, high-end, and low-end GPUs, as YOLOv4 [17] is the only general GPU-based real-time object detector. The downsampling convolution was not present in the design of the CSPDarknet53, which proved helpful in the reduction of computation in every stage of CSPDarknet by $whb2(9/4 + 3/4 + 5k/2)$. This reduction formula has proved the CSPDarknet to be beneficial over the simple Darknet backbone only when the value of k is greater than 1. Every stage of CSPDarknet has [1-2-8-8-4] residual layers. To attain an optimal performance, the authors placed the first CSP stage into the original Darknet residual layer [17].

The path aggregation network (PAN), a short form of PANet, is used for image segmentation by conserving spatial information, which improves localisation. The PAN in the YOLOv4 is CSP-ized in scaled-YOLOv4 to lessen the computational cost by 40%. In previous object detection algorithms, the Spatial Pyramid Pooling (SPP) is present in the centre of the first computational list of the neck [35]. The designers of scaled-YOLOv4 also added that it is the centre of the first computational list of CSPPAN. The architecture of the proposed computational list is shown in Figure 2.

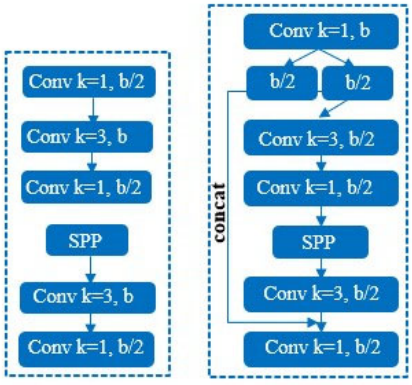


Figure 2. Computational blocks of SPP and CSPPAN.

3.2. YOLOv4-Tiny

Model size affects the inference time and computational cost and requires powerful hardware resources for the best performance. Therefore, during tiny model scaling for

low-end devices, some factors such as memory access cost (MAC), the traffic of dynamic random-access memory (DRAM), and memory bandwidth need to be fully examined.

In lightweight models, to acquire high accuracy with minimal computations, a higher parameter utilisation efficiency is required. The authors analysed the network with the computational load of DenseNet and OSANet with the growth rate (g) and found that OSANet was the best model for tiny model scaling because of its low computational complexity, which is less than $O(whkb2)$. Similarly, to attain the best computing speed, the authors introduced a new concept and performed gradient truncation between the computational layers of CSPOSANet. Power consumption is the most significant factor that is considered when the computational cost of low-end devices is being evaluated. MAC is found to be the biggest factor that affects power consumption, which is calculated by Equation (1).

$$MAC = hw\,(C_{in} + C_{out}) + KC_{in}C_{out} \quad (1)$$

where h represents height, w represents the width of the feature map, C_{in} represents the channel number of inputs, C_{out} represents the channel number of outputs, and K represents the kernel size of the convolutional filter. According to the authors, the smallest MAC value can be derived when $C_{in} = C_{out}$.

By minimising the convolutional input–output (CIO), the DRAM traffic can be minimised. The authors evaluated the CIO of OSA, CSP, and their designed CSPOSANet, as shown in Equation (2), and found that the proposed CSPOSANet can achieve the best CIO results when $kg > b = 2$.

$$kg^2 + (b + kg)^2 = 4 \quad (2)$$

3.3. YOLOv4-Large

While scaling for high-end devices, the accuracy and inference speed can be improved by adjusting the detector's input, backbone, and neck. The prediction capability of the model depends upon the receptive fields of the feature vector. In neural networks, the stage is directly related to the receptive fields, and the feature pyramid network (FPN) indicates that a higher number of stages helps in the prediction of larger objects. YOLOv4-large is designed for the training of large models on distributed cloud-based GPUs. A fully CSP-ized YOLO-P5 is designed and is scaled in YOLOv4-P6 and YOLOv4-P7. The authors performed compound scaling on $\{size^{input}, \#stage\}$, set the depth scale of each stage to 2^{dsi}, set d_s to [1, 3, 15, 15, 7, 7, 7], and found the best results.

4. Materials and Methods

4.1. Data Curation

4.1.1. Training Dataset

To observe people in low-light environments, we utilised the ExDark [36] dataset, which contains images of 12 different low-light scenarios. It is the first dataset available that is entirely based on low-light scenarios. The dataset contains the images of 10 different classes. We extracted the dataset for the person class and trained our models to it.

4.1.2. Testing Dataset

DepTSol was tested on a custom dataset collected from Pakistan at night in the days of COVID-19. Pakistan is one of the most urbanised countries in South Asia. The large population and congested streets make it a riskier place in the growth of COVID-19, and it is very difficult to maintain a safe distance in such narrow places. Hence, the monitoring system needs a high accuracy in terms of the detection and location of people. The test dataset was a collection of 323 RGB frames collected from different low-light conditions and different crowded and less crowded places. In this study, 186 frames were collected

from images depicting a crowd in the market of Rawalpindi Pakistan, which help in assessing the performance of object detectors in low-light conditions; the remaining 134 frames were collected from various outdoor environments. We obtained signed consent forms from the participants of the study, and the identities of those captured in crowded areas have been removed. All frames were captured by a ToF camera of a Samsung Galaxy Note 10+, where Gh is 4.5 ft, and FL is 35 mm [37]. The dataset is publicly available [38].

4.2. Problem Articulation

We defined a scene as five-tuple value $S = \{V_f, G_h, TH_{ud}, A_n, BB_c\}$, where $V_f = height \times width \times 3$ shows the width and height of an RGB video frame and $V_f \in \Re_+$, G_h is the camera height from the ground in feet, TH_{ud} shows the least physical distance that should be maintained to stay safe, A_n is a binary control signal for sending a voice warning if the monitored inter-personal distance is less than TH_{ud}, and BB_c is the colour of the detected bounding boxes. In a given S, we are interested in finding the inter-personal pixel dis- tance $Dpx = \{pd_{(1,2)}, pd_{(1,3)}, ..., pd_{(1,n)}, pd_{(2,3)}, pd_{(2,4)}, ..., pd_{(2,n)}, ..., pd_{(n-1,n)}\}$ at varying CF_D values, where $CF_D \in a$, and a is a multiple of the specified safe physical distance. In our case, it is 180 cm ≈ 6 ft. Therefore, $a = \{180, 360, 540, 720, ..., n\}$. After finding the D_{px}, we converted it into real-world units centimeters (cm) UD_{i+n}. We found TH_{ud} to highlight the safety distance violations ($UD_{i+n} < TH_{ud} \mid UD_{i+n} \geq TH_{ud}$) in the given ROI. In the end, if a safety breach is detected, the BB_c becomes red, and a voice warning is sent to the people violating the safe physical distance by setting the $A_n = 1$; else, in the normal cases, BB_c remains green, and $A_n = 0$.

4.3. Real-Time People Detection

In this study, from the list of scaled-YOLOv4, the CSP-ized YOLOv4 algorithm was utilised for the detection of humans in V_f, as it improves prediction accuracy with a high inference speed. A detailed discussion of the model is presented in the Data Model section. The output of the model is the bounding boxes of detected people $bb_i = \{bb_{(i,1)}, bb_{(i,2)}, bb_{(i,3)}, ..., bb_{(i,n)}\}$, their confidence score bc_i, and the class label bl_i. $bb_{(i,j)} = \{x_{(i,j)}, y_{(i,j)}\}$ gives pixel indices of bounding boxes in V_f, where j shows the associative four corners: bottom-left, bottom-right, top-left, and top-right. The aim was to develop a robust real-time people detection model with minimal localisation and classification errors, capable of delivering high precision by considering various challenges such as variations in clothes, height, poses, and partial visibility. Figure 3 demonstrates the structure of the YOLO-based person detection module.

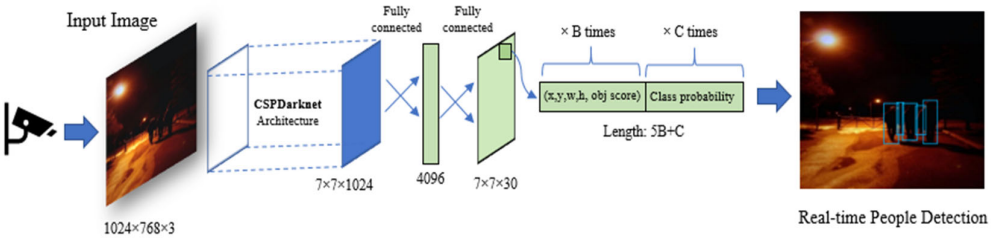


Figure 3. YOLO-based real-time people detection.

4.4. Camera-to-People Distance Estimation

We propose a motionless monocular ToF camera [38] for real-time video surveillance. The built-in accuracy of ToF cameras is good, as it combines the advantage of active sensors and camera-based approaches. Bad lighting conditions and texture mixing are usually noticed in stereo vision cameras, and they are computationally expensive, whereas ToF cameras have proven to be best in such scenarios. In comparison with 3D vision systems, ToF was found to be compact and straightforward, as they have a built-in

illumination ability with no moving parts. It yields efficient results based on low processing power. In contrast to laser scanners, ToF cameras can measure up to 100 fps in one shot, which is much faster than laser technology. ToF technology has a variety of applications, including path-planning for manipulators [39,40], obstacle avoidance [41,42], wheelchair assistance [42], medical respiratory motion detection [43], semantic scene analysis [43], simultaneous localization and mapping (SLAM) [44], and human–machine interaction [45–48].

A ToF camera helps us measure the camera-to-person distance with high accuracy, which allows us obtain optimal performance in our people monitoring approach. In the ToF camera unit, the camera's light blinks, and a modulated light pulse travels from the illumination source to the object. The distance between the camera and the object is calculated by the time taken by the light pulse to return to the source object after striking the target object. The transmitted light faces a delay according to the distance it covers to reach the object and then return to the source, which means that the farther the object, the more time the pulse will take to return to the source. The time delay T_D that the illumination faces is expressed in Equation (3).

$$T_D = 2 \times \frac{D_o}{v_o} \tag{3}$$

where D_o represents the object distance in meters (m), and v_o is the velocity of the light in meters per second (m/s). The maximum range that the camera can cover is determined by the pulse width of the illumination and calculated by Equation (4), whereas the camera-to-object distance is calculated by Equation (5).

$$D_{\max} = \frac{1}{2} \times \mathrm{v_o} \times \mathrm{T_o} \tag{4}$$

The distance between the camera and the object is half of the total distance travelled by the light pulse. Here, T_o shows the length of the pulse.

$$D_{\max} = \frac{1}{2} \times \mathrm{v_o} \times \mathrm{T_o} \times \frac{a_2}{a_1 + a_2} \tag{5}$$

where a_2 is the signal that is generated when the light pulse is emitted, and a_1 represents the signal when no light emission is encountered.

4.5. Threshold Specification and People Inter-Distance Estimation

To initiate the monitoring process, we calibrated the camera in the real-world environment by specifying intrinsic and extrinsic camera parameters. For intrinsic camera parameters, we assumed the fixed focal length (FL) that we set according to the area where the surveillance system was installed, depending on the required field of view (FoV). To specify the extrinsic camera parameters, we divided the *S* into three different camera ranges: CF_D*−near*, CF_D−far, and CF_{DR}. To start the monitoring process, we defined a threshold distance in V_f in the form of pixels. For the specification of threshold distance in V_f and to proceed further, we made arrangements in a real-world environment. We took four target objects, T1, T2, T3, and T4, from which two targets (T1, T2) were placed on the camera-to-frame distance CF_D*−near*, and the other two (T3, T4) were placed at CF_D − far. Two different ranges of frames, i.e., CF_D*−near* and CF_D*−far* with respect to the camera, with four target objects are shown in Figure 4, where CF_D is the distance between the ToF camera and the V_f, (h_m, h_f) is the total height, (l_m, l_f) is the total length of the near and the far V_f, (ixm_{T1}, ixm_{T2}) and (i_{xfT3}, i_{xfT4}) show the length, and (iym_{T1}, iym_{T2}) and (i_{yfT3}, i_{yfT4}) show the

height of the objects in the near and far frames. The pixel size of objects projected in CF_D–*near* is different from the target objects in CF_D–*far* and decreases as the value of CF_D increases. Figure 4 shows that, as long as the frame moves away from the camera, the pixel size of all other parameters increases in addition to the pixel size of the objects present in the frames. We execute Algorithm 1 and 2 to specify the threshold value and monitor people at CF_D–*near* and CF_D–far, whereas Algorithm 3 is executed to monitor people at a distance above the CF_D–*far* up to the specified maximum camera range CF_{DR}, which means that people outside CF_{DR} will not be monitored.

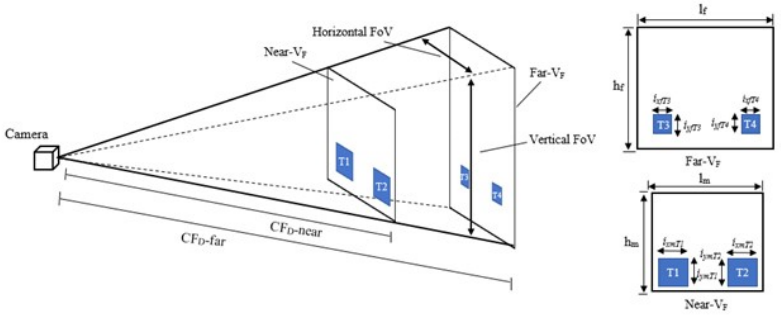


Figure 4. Real-world arrangements of ToF-based video surveillance.

4.5.1. Monitoring People at CF_D–*near*

To initiate the procedure, we should know the threshold distance between the target objects (T1 and T2) in units—in our case, TH_{ud} = 180 cm ≈ 6 ft, the minimum specified safe distance by WHO. We then initialise the camera-to-frame distance CF_D–*near*, which shows how far we start the monitoring process from the camera. E_{px} represents extra pixels that we require because, as CF_D increases, the number of pixels starts to decrease. At the start, E_{px} is initialised at 0 because, at the beginning of the procedure, no pixel loss is encountered. In Step 8, we calculate the Euclidean distance between the centroids of T1 and T2, which yields the threshold distance in terms of pixels TH_{pd}, equivalent to TH_{ud}. In Step 10, to convert pixel distance into units, we find that the proportion of TH_{ud} and TH_{pd} yields the unit points equivalent to pixels, where k represents a constant value that maps pixel distance to the unit distance (cm). In Steps 11 and 12, we calculate the Euclidean distance between the centre points of all detected bb_i at CF_D–*near* and convert it to the unit distance cm. UD_{i+1} shows the distance between all detected persons at CF_D–*near* in terms of cm. In Steps 13 to 18, we compare the monitored unit distance with TH_{ud}. The people violating the TH_{ud} are highlighted by red bounding boxes and are notified by a general voice warning.

Algorithm 1: Monitoring people at CF_D − *near*

Input: CF_{DR}

Output: UD_{i+n} 1

1 ***Start variables:***

c, Global var1

E_{px}, Global var2

A_n, Global var3

BB_c, Global var4

TH_{ud}, Global var5

End variables

2 **Initialization**: $CF_D - near \leftarrow a_1$, $c \leftarrow 1$, $TH_{ud} \leftarrow 180$ cm, $E_{px} \leftarrow 0$

$ED_{(T1T2)} \leftarrow \mathrm{p}(x_{mT2} - x_{mT1})2 + (y_{mT2} - y_{mT1})2$

$TH_{pd} \leftarrow ED_{(T1T2)}$

$k \leftarrow \quad \mathrm{TH}_{\overline{\mathrm{ud}}} / \mathrm{TH}_{\mathrm{pd}}$

$$Dpx^{(i+1)} = \sqrt{\sum_{k=1}^{n} \left(yk_{(i+1)} - xk_{(i+1)} \right)^2}$$

$UD_{i+1} \leftarrow (k \times D_{px(i+1)})$

If $\mathrm{UD}_{(i+1)} < \mathrm{TH}_{\mathrm{ud}}$ **then**

  $A_n \leftarrow 1$

  $BB_c \leftarrow Red$

else

  $A_n \leftarrow 0$

  $BB_c \leftarrow Green$

end

Algorithm 2: Monitoring people at $CF_D - far$

$CF_D - far \leftarrow a_2$

$ED(T3T4) \leftarrow \sqrt{(xfT4 - xfT3)2 + (yfT4 - yfT3)2}$

$Epx \leftarrow ED_{(}T1T2) - ED_{(}T3T4)$

$$Dpx^{(i+2)} = \sqrt{\sum_{k=1}^{n} \left(yk_{(i+2)} - xk_{(i+2)} \right)^2}$$

$UDi+2 \leftarrow (k \times Dpx(i+2)) + ((k \times Epx)) \times c$

If $\mathrm{UD}_{(i+2)} < \mathrm{TH}_{\mathrm{ud}}$ **then**

  $An \leftarrow 1$

  $BB_c \leftarrow Red$

else

  $A_n \leftarrow 0$

  $BB_c \leftarrow Green$

end

4.5.2. Monitoring People at $CF_D - Far$

In Algorithm 2, we change the camera-to-frame distance from $CF_D - near$ to $CF_D - far$. We place T3 and T4 at $CF_D - far$, where their self-distance is the same threshold value TH_{ud} = 180 cm. We then calculate the Euclidean distance between the centre points of T3 and T4. The value of E_{px} is updated this time as objects are now at $CF_D - far$, so the Euclidean distance between the centre points of objects at $CF_D - far$ is not the same as that of objects at $CF_D - near$, but the TH_{ud} between both CF_D values is the same.

To recover the lost pixels at $CF_D - far$, we calculate the difference between the Euclidean distance of T1 and T2 at $CF_D - near$ and that of T3 and T4 at $CF_D - far$ and multiply it by c, where the initial value of c is 1 and increases as long as CF_D increases. After calculating the difference, we update the value of E_{px} and add the lost pixels that are stored in E_{px}

to UD_{i+2} by multiplying E_{px} with k, which converts the recovered pixels into cm, where UD_{i+2} shows the distance between all detected persons at $CF_D - far$ in terms of cm.

4.5.3. Monitoring People Up to CF_{DR}

In Step 1 of Algorithm 3, we start a loop to monitor people above the $CF_D - far$ up to the maximum specified camera range CF_{DR}. We initialise CF_D with a_3, where $a_3 \in a$. In Step 2, we check whether more than one object is present at CF_D. If more than one object is present at CF_D, then Steps 5–17 of the algorithm are executed, where we increment the value of c to recover the lost pixels at each CF_D, convert the monitored Euclidean distance between the centre points of detected objects into cm, and compare it with TH_{ud}. We execute Steps 16 and 17 if a single object or no object is detected at CF_D. The workflow of the proposed DepTSol model is shown in Figure 5.

Algorithm 3: Monitoring people up to CF_{DR}

1 **For** $CF_D = a_3$; $CF_D <= CF_{DR}$; $CF_D += a_1$
2 **If** $bb_i > 1$ **then**
3 c++
4 $E_{px} \leftarrow E_{px} \times c$
5 $Dpx_{(i+n)} = \sqrt{\sum_{k=1}^{n} (yk_{(i+n)} - xk_{(i+n)})^2}$
6 $UD_{i+n} \leftarrow (k \times D_{px(i+n)}) + ((k \times E_{px})) \times c$
7 **If** $UD_{(i+n)} < TH_{ud}$ **then**
8 $A_n \leftarrow 1$
9 $BB_c \leftarrow Red$
10 **else**
11 $A_n \leftarrow 0$
12 $BB_c \leftarrow Green$
13 **end if**
14 **else**
15 c++
16 $E_{px} \leftarrow E_{px} \times c$
17 **end if**
18 **end For**

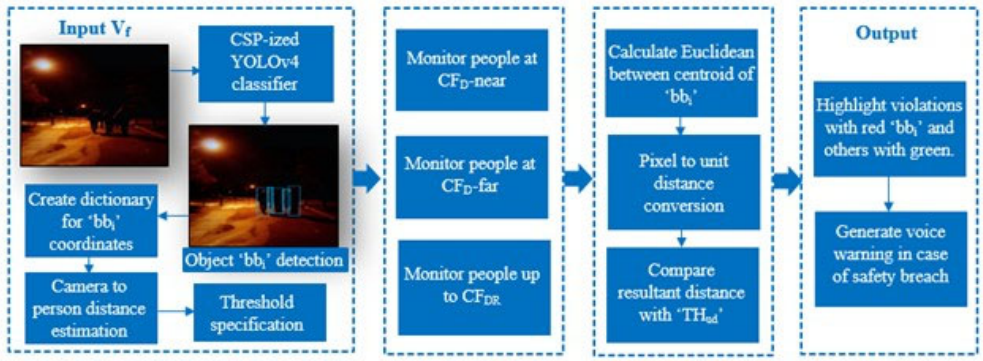


Figure 5. Workflow of the DepTSol model.

5. Experiments & Results

5.1. Experimental Setup

We performed transfer-learning on the MS COCO dataset [49] to train a custom object detector to attain the highest model accuracy. The selection of hyper-parameters for the training of one stage object detectors on the ExDARK dataset were as follows: The network size was 512 × 512. The initial learning rate was 0.01. The initial batch size was 64 with 16 subdivisions. To accelerate gradients vectors in the right directions, stochastic gradient descent (SGD) momentum was used with an initial value of 0.937 and a weight decay of 0.0005. For bounding box regression, generalised intersection over union (GIOU) loss was adopted with an initial value of 0.05. The initial class loss gain was 0.5, and the class binary cross-entropy (BCE) loss positive gain was 1.0. The object loss gain and object BCE loss positive gain was 1.0. The adopted intersection over union (IoU) target-anchor training threshold was 0.2, and the anchor threshold was 4.0. To handle the class imbalance problem by assigning more weights to hard or easily misclassified examples, the focal loss (gamma) was used with an initial value of 0.0.

From data augmentation, the following parameters were adopted: To train the model on varying image colours, the chosen fraction of hue, saturation, and value augmentation were 0.015, 0.7, and 0.4, respectively. To add non-linearity, the mish activation function was used. To make the model localise all people in different portions of the frame, the mosaic data augmentation technique was utilised. All experiments were performed on a Tesla T4 GPU. The utilised PyYAML version was 5.4.1, the torch version was 1.8.0 with cu101, and the mish version was 0.0.3. The architectural configuration of CSP-ized YOLOv4 is shown in Figure 6.

Parameters	Anchors	Backbone	Head
nc: 1 # number of classes depth_multiple: 1.0 # model depth multiple width_multiple: 1.0 # layer channel multiple	anchors: - [12,16, 19,36, 40,28] # P3/8 - [36,75, 76,55, 72,146] # P4/16 - [142,110, 192,243, 459,401] # P5/32	backbone: # [from, number, module, args] [[-1, 1, Conv, [32, 3, 1]], # 0 [-1, 1, Conv, [64, 3, 2]], # 1-P1/2 [-1, 1, Bottleneck, [64]], [-1, 1, Conv, [128, 3, 2]], # 3-P2/4 [-1, 2, BottleneckCSP, [128]], [-1, 1, Conv, [256, 3, 2]], # 5-P3/8 [-1, 8, BottleneckCSP, [256]], [-1, 1, Conv, [512, 3, 2]], # 7-P4/16 [-1, 8, BottleneckCSP, [512]], [-1, 1, Conv, [1024, 3, 2]], # 9-P5/32 [-1, 4, BottleneckCSP, [1024]], # 10]	# na = len(anchors[0]) head: [[-1, 1, SPPCSP, [512]], # 11 [-1, 1, Conv, [256, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [8, 1, Conv, [256, 1, 1]], # route backbone P4 [[-1, -2], 1, Concat, [1]], [-1, 2, BottleneckCSP2, [256]], # 16 [-1, 1, Conv, [128, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [6, 1, Conv, [128, 1, 1]], # route backbone P3 [[-1, -2], 1, Concat, [1]], [-1, 2, BottleneckCSP2, [128]], # 21 [-1, 1, Conv, [256, 3, 1]], [-2, 1, Conv, [256, 3, 2]], [[-1, 16], 1, Concat, [1]], # cat [-1, 2, BottleneckCSP2, [256]], # 25 [-1, 1, Conv, [512, 3, 1]], [-2, 1, Conv, [512, 3, 2]], [[-1, 11], 1, Concat, [1]], # cat [-1, 2, BottleneckCSP2, [512]], # 29 [-1, 1, Conv, [1024, 3, 1]], [[22,26,30], 1, Detect, [nc, anchors]], # Detect (P3, P4, P5)]

Figure 6. Architectural configuration of CSP-ized YOLOv4.

5.2. Evaluation Measures

We used common performance evaluation metrics precision and recall to perform comparative analysis between different one stage object detectors and chose the best for our real-time social distance monitoring solution in terms of performance [50].

Precision is the proportion of the number of true positives (TP) to the total number of positive predictions. In contrast, recall is the proportion of the number of TP to the total number of actual objects. Precision and recall are calculated by Equations (6) and (7). In

the field of object detection, IoU is a threshold value that determines whether the predicted result is TP or true negative (TN).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7}$$

Average precision (AP) depends on the precision–recall (PR) curve and is defined as the precision score averaged over all distinctive recall levels as shown in Equation (8), whereas average recall (AR) is calculated by Equation (9).

$$AP = \sum_{i=1}^{n-1} (\mathrm{r}_{i+1} - \mathrm{r}_i)\, \mathrm{p}_{\mathrm{interp}}(\mathrm{r}_{i+1}) \tag{8}$$

where p_{interp} is interpolated precision at recall levels r_1, r_2, r_3,..., r_n.

$$AR = 2\int_{0.5}^{1} recall(iou)do \tag{9}$$

In this article, the COCO evaluation metric [51] was used for performance evaluation because of its varsity. The standard evaluation metric is Pascal VOC [52], but it defines the mAP score at only 0.5 IoU. However, the COCO evaluation metric contains an mAP score at three different IoU threshold values, including the primary challenge metric that averages the mAP score at 10 different IoU thresholds from 0.50 up to 0.95 with a step size of 0.05. A standard metric is the same as Pascal VOC, which considers only a single threshold value of 0.5, and a strict metric, where the IoU threshold is 0.75. Besides this, COCO provides an mAP score for small ($area < 32^2$), medium ($area > 32^2$ and $< 96^2$), and large size objects ($area > 96^2$). As demonstrated in Equation (10), the mAP score is the mean of all AP values over N number of classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{n} AP_i \tag{10}$$

Similar to the mAP score, the mAR score also has two sets of variations. In the first set, mAR gives the various number of detections per frame; e.g., $mAR^{max=1}$ gives only one detection per frame, $mAR^{max=10}$ gives 10 detections per frame, and $mAR^{max=100}$ gives 100 detections per frame. In the second set, the mAR is calculated based on the size of detected objects such as small ($area < 32^2$), medium ($area > 32^2$ and $< 96^2$), and large objects ($area > 96^2$). The mean average recall (mAR) is the mean of all AR values over N number of classes as shown in Equation (11).

$$\text{mAR} = \frac{1}{N}\sum_{i=1}^{n} AR_i \tag{11}$$

For the final evaluation of the DepTSol model, we used the mean absolute error (MAE) [53] score, which is the mean of the difference of observed and actual distance values, as shown in Equation (12).

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{n} |y_i - y_{i'}| \tag{12}$$

where y_i shows the observed distance, and $y_{i'}$ represents the actual distance.

5.3. Results

We performed various experiments for the evaluation of our social distance monitoring approach DepTSol. Besides evaluating the performance of the CSP-ized YOLOv4, we evaluated the performance of one-stage object detection models on the ExDark dataset and compared the results with CSP-ized YOLOv4 both in terms of speed and accuracy.

As per the literature, low-light environments are not focused on much in the field of object detection. The direct evaluation of object detection models in low-light scenarios will pave the way for researchers to further study the field. The comparative analysis between seven different object detection models, including the Single-Shot Detector (SSD) [54], RetinaNet [55], the Enriched Feature Guided Refinement Network (EFGRNet) [56], YOLOv3, YOLOv3 Spatial Pyramid Pooling (YOLOv3-SPP) [35], YOLOv4, and the CSP-ized YOLOv4, is shown in Tables 1 and 2. From the results, we can analyse that CSP-ized YOLOv4 shows the best performance both in terms of speed and accuracy. The training convergence of CSP-ized YOLOv4 on GIoU loss, objectness loss, classification, precision, recall, and mAP is shown in Figure 7 with a network size of 512 × 512. The SSD has attained the second position in terms of speed but achieved the sixth rank at various mAP scores and remained at the last level in terms of mAP for small area objects. YOLOv4 has achieved the second rank in terms of accuracy and the third rank in terms of speed. YOLOv3-SPP stands at the third level in terms of accuracy, YOLOv3 achieves the fourth rank in terms of speed, and YOLOv3-SPP and YOLOv3 achieve almost the same fps score with a difference of 0.7 fps. EFGRNet has achieved the third rank in terms of fps score and the fourth in terms of mAR and mAP. RetinaNet has reported the lowest speed and mAP score as compared to all other models, while performing better than SSD, EFGRNet, and YOLOv3 for small-size objects.

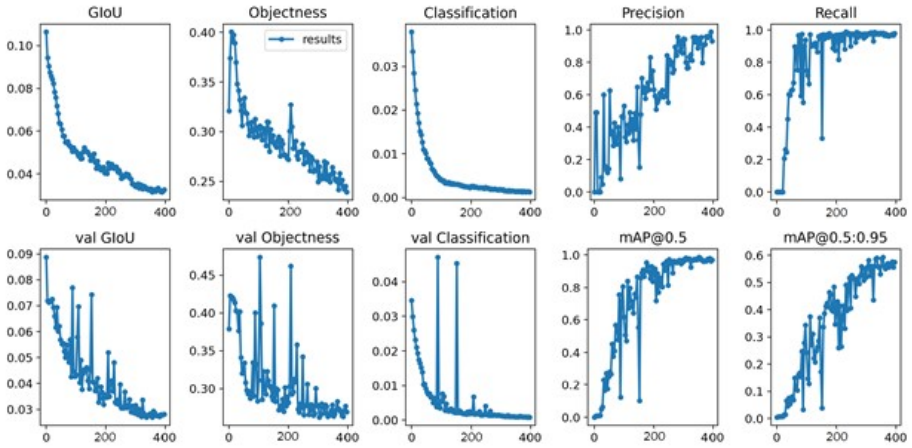


Figure 7. The graphic depiction of CSP-ized YOLOv4 convergence over GIoU, objectness, classification, precision, recall, and mAP score.

Based on the comparative analysis, we have taken the trained model of CSP-ized YOLOv4 for high performance. We obtained the object detection results from CSP-ized YOLOv4 and applied social distance monitoring algorithms on the obtained images coordinates for inter distance estimation. We tested our DepTSol model at 230 different RGB frames. Some of the test results from the qualitative evaluation are shown in Figure 8, and Table 3 shows the quantitative results in terms of the predicted unit distance U_D, the actual unit distance AU_D, and their relevant pixel values. Figure 9 depicts the further qualitative results of the DepTSol model from the testing dataset. To start the monitoring process, we initialised $CF_D - near$ with a_1 $CF_D - far$ with a_2, and the CF_{DR} was a_3, where $(a_1, a_2, a_3) \in a$. We monitored people at each CF_D, calculated the error rate between U_D and AU_D at each level, and summarised it with an MAE score.

Table 1. Comparative analysis of one-stage object detectors on the ExDARK dataset at various thresholds.

Model	Backbone	Size	FPS	$mAP_{[0.50]}$	$mAP_{[0.75]}$	mAP_{small}	mAP_{medium}	mAP_{large}
SSD	VGG-16	512	44.2	73.1%	57.2%	13.6%	31.3%	47.1%
RetinaNet	ResNet-50	512	22.3	70.0%	62.1%	23.28%	28.0%	56.1%
EFGRNet	VGG-16	512	37.9	87.0%	69.1%	17.2%	47.1%	62.8%
YOLOv3	Darknet53	512	33.7	84.5%	55.6%	19.4%	39.8%	61.1%
YOLOv3-SPP	Darknet53	512	33.1	91.1%	64.4%	31.0%	43.6%	74.6%
YOLOv4	CSPDarknet53	512	41.1	98.2%	78.3%	35.3%	54.2%	86.0%
CSP-ized YOLOv4	CSPDarknet53	512	51.2	99.7%	94.0%	55.5%	83.0%	94.3%

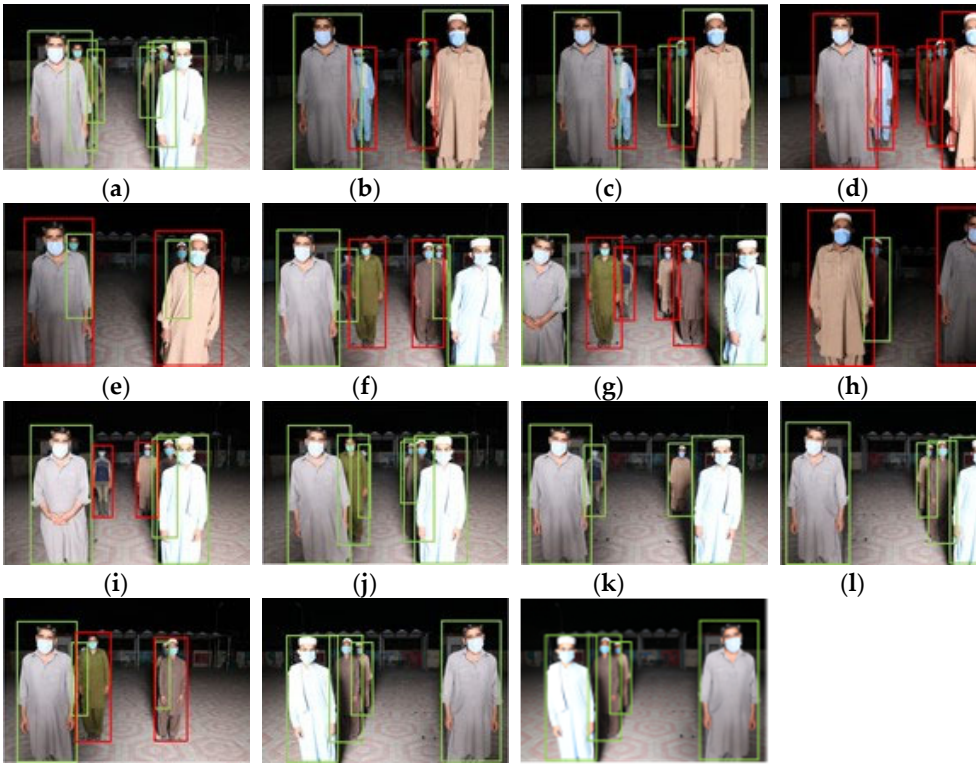
Table 2. Evaluation of one-stage object detection algorithms on the variant mAR score.

Model	$mAR_{max=1}$	$mAR_{max=10}$	$mAR_{max=100}$	mAR_{small}	mAR_{medium}	mAR_{large}
SSD	39.8%	69.4%	65.8%	48.5%	69.8%	77.9%
RetinaNet	74.9%	68.0%	54.2%	41.0%	63.6%	54.7%
EFGRNet	83.9%	71.1%	68.6%	52.1%	80.4%	74.8%
YOLOv3	86.3%	79.6%	75.1%	50.4%	94.2%	89.1%
YOLOv3-SPP	89.0%	88.4%	86.1%	59.0%	94.0%	93.6%
YOLOv4	94.0%	97.2%	95.3%	69.2%	97.7%	97.8%
CSP-ized YOLOv4	96.1%	99.4%	98.0%	73.6%	98.8%	99.5%

Table 3. Quantitative evaluation of the DepTSol model, where TH_{ud} represents the threshold distance in cm, TH_{pd} is the threshold distance in pixels, U_D shows the predicted unit distance at each CF_D level, AU_D is the actual unit distance, AE is the absolute error, and V represents the number of violations per frame.

Frame	CF_D	D_{px}	$D_{px} + (E_{px} \times c)$	TH_{ud}	TH_{pd}	k	U_D (cm)	AU_D (cm)	Error (cm)	FP	TN	V
	CF_D − *near*	308.2	-	180	308.2	0.5842	180	180	0			
(a)	CF_D − far	255.2	308.2	-	-	-	180	180	0	0	0	0
	CFDR	203.1	309.1	-	-	-	180.54	180	0.54			
	MAE = 0.18 cm											
	CF_D − *near*	310.0	-	-	-	0.5842	181.1	180	1.1			
(b)	CF_D − far	151.2	204.2	-	-	-	119.27	120	-0.73	0	0	1
	MAE = 0.92 cm											
(c)	CF_D − *near*	312.3	-	-	-	0.5842	182.41	180	2.41			
	CF_D − far	122.1	175.1	-	-		96.43	100	-3.57	0	0	1
	MAE = 2.99 cm											
	CF_D − *near*	209.0	-	-	-	0.5842	122.0	120	2.0			
(d)	CF_D − far	115.4	168.4	-	-	-	98.36	100	−1.64	0	0	3
	CFDR	67.1	173.1	-	-	-	101.11	100	1.11			
	MAE = 1.58 cm											
	CF_D − *near*	177.3	-	-	-	0.5842	103.56	100	3.56			
(e)	CF_D − far	296.7	349.7	-	-	-	204.20	200	4.2	0	0	1
	MAE = 3.88 cm											
	CF_D − *near*	437.0	-	-	-	0.5842	255.25	250	5.25			
(f)	CF_D − far	156.0	209.0	-	-	-	122.07	120	2.07	0	0	1
	MAE = 3.66 cm											
	CF_D − *near*	436.1	-	-	-	0.5842	254.70	250	4.7			
(g)	CF_D − far	159.0	212.0	-	-	-	123.8	120	3.8	0	0	1
	CFDR	319.0	425.0	-	-	-	248.24	250	−1.76			

					MAE = 3.42 cm							
	CF_D – *near*	518.1	-	-	-	0.5842	302.62	300	2.62			
(h)	CF_D – far	222.0	275.0	-	-	-	160.63	160	0.63	0	0	2
	CFDR	125.11	231.11	-	-	-	129.1	130	−0.9			
					MAE = 1.38 cm							
(i)	CF_D – *near*	246.3	-	-	-	0.5842	143.86	140	3.86	0	0	1
					MAE = 3.86 cm							
(j)	CF_D – *near*	314.3	-	-	-	0.5842	183.58	180	3.58			
	CF_D – far	168.3	*221.3*	-	-	-	129.26	130	-0.8	0	0	1
					MAE = 2.19 cm							
	CF_D – *near*	312.1	-	-	-	0.5842	182.29	180	2.29			
(k)	CF_D – far	259.1	312.1	-	-	-	182.29	180	2.29	0	0	0
	CFDR	244.5	350.5	-	-	-	204.72	200	4.72			
					MAE = 3.1 cm							
(l)	CF_D – *near*	410.4	-	-	-	0.5842	239.71	240	-0.29			
	CF_D – far	197.78	*250.78*	-	-		146.48	150	-3.52	0	0	0
					MAE = 1.90 cm							
(m)	CF_D – *near*	322.4	-	-	-	0.5842	188.31	190	−1.69	0	0	0
					MAE = 1.69 cm							
	CF_D – *near*	202.2	-	-	-	0.5842	118.10	120	−1.9			
(n)	*CFDR*	240.0	346.0	-	-	0.5842	202.13	200	2.13	0	0	1
					MAE = 2.01 cm							
(o)	CF_D – *near*	322.4	-	-	-	0.5842	188.31	190	−1.69	0	0	0
					MAE = 1.69 cm							



(**m**) (**n**) (**o**)

Figure 8. Qualitative evaluation of the DepTSol model. SubFig (a) to (o) represents frames whose quantitative evaluation is depicted in Table 3.

Figure 9. Qualitative visualisations of the DepTSol model on some testing dataset seeds under different low-light conditions.

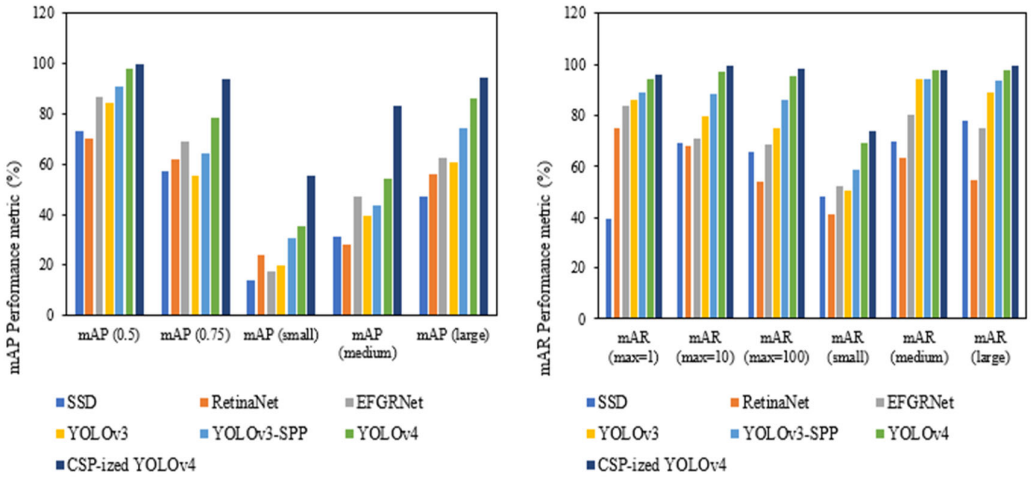


Figure 10. The graphic depiction of the best testing performance by COCO evaluation metrics at varying mAP and mAR scores.

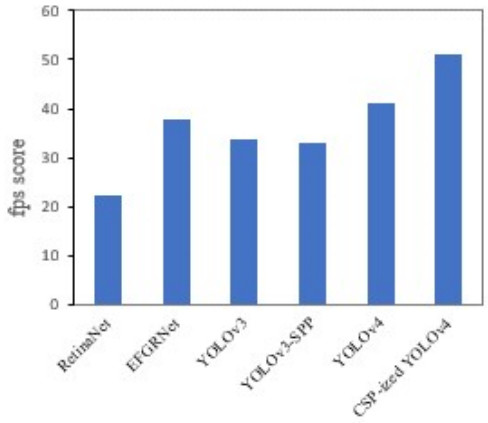


Figure 11. The best speed trade-off in terms of fps score.

6. Limitations and Discussion

Low-light environments play a vital role in the spread of disease. The prvision of effective social distance monitoring approaches is required to serve that motive. The detection of people in low-light environments is itself a challenging task. The application of image processing techniques for the enhancement of dark images and the subsequent application of object detection algorithms results in a slow response time and requires high-power machines to execute multiple tasks. To make the system highly responsive, we directly applied object detection algorithms in low-light scenarios and evaluated the performance. We tested seven different one-stage object detection algorithms on the ExDARK dataset and evaluated the models both in terms of accuracy and speed. From the obtained results, Figures 10 and 11 depict the empirical results of the performed experiments. To summarise the compared models' performance, we explored the testing results of each model by COCO evaluation metrics on a Tesla T4 GPU with a network size of 512 × 512. The CSP-ized YOLOv4 achieved the best performance results as compared to the six other one-stage detectors. Based on COCO evaluation, the CSP-ized YOLOv4 obtained an fps value of 51.2 and an $mAP^{[0.5]}$ of 99.7%. Due to its high performance as compared to other one-stage object detectors, we utilised it for our social distance monitoring task to control FP and TN and to support real-time monitoring.

Analysis shows that the direct application of object detection algorithms in low-light environments for human detection and monitoring purposes is very effective. Additionally, the direct application of deep-learning-based object detection algorithms on low-light datasets promotes the acquisition of effective results at a very low cost. We can save the cost incurred on powerful devices to perform image cleansing and visibility enhancement. Furthermore, the fps score of the models can be further enhanced by utilising the GPUs such as Tesla V100, Volta, and Titan Volta, whereas the training of the models on a higher network size results in a higher mAP score.

The proposed CSP-ized YOLOv4 and ToF-based real-time social distance monitoring approach has shown effective results with an overall MAE of 2.23 cm. Figure 12 presents visualisations of U_D and AU_D. The approach considers individual's privacy concerns. Instead of targeting people individually, we use general voice warnings that alert all people present at the location. The proposed general warning system is highly feasible in outdoor environments, such as night outdoor gatherings. Moreover, for indoor environments such as offices, homes, libraries, and hospitals, we can use non-intrusive audiovisual cues that only target and notify certain people in the environment without distracting others in the surrounding area. The proposed camera calibration technique has addressed the limitations of the previous study of monitoring people at a fixed camera distance C_D in a given environment by dividing the scene into multiple safety threshold distance values (e.g., $CF_D - near$ and $CF_D - far$, up to the maximum specified camera range CF_{DR}). The proposed

approach can effectively monitor people at multiple camera distances in a given environment and generate voice warnings. Moreover, in contrast to the previous study, in the DepTSol model, we improved the mAP score by 1.86%, while no single FP or FN was detected. Besides these numerous improvements, the approach is limited in giving feasible results at CF_D values that lie behind multiple safety thresholds (e.g., if we start the monitoring process 180 cm away from the camera and initialise a $CF_D - near$ of 180 cm and then a $CF_D - far$ of 360 cm, monitoring can be done. The approach does not yield correct results for people between a CF_D of 180 cm and 360 cm). Furthermore, in the proposed camera calibration approach to start the monitoring process, we have to place four target objects in a real-world environment. In addition, the installation of the system in a given environment is dependent on the extrinsic camera parameters (i.e., FL and FoV), which means the pixel threshold value TH_{pd} relevant to the unit threshold value TH_{ud} needs to be calculated every time a change in these parameters is encountered. These limitations can be tackled by introducing a new camera calibration approach that can monitor people apart from specific CF_D values.

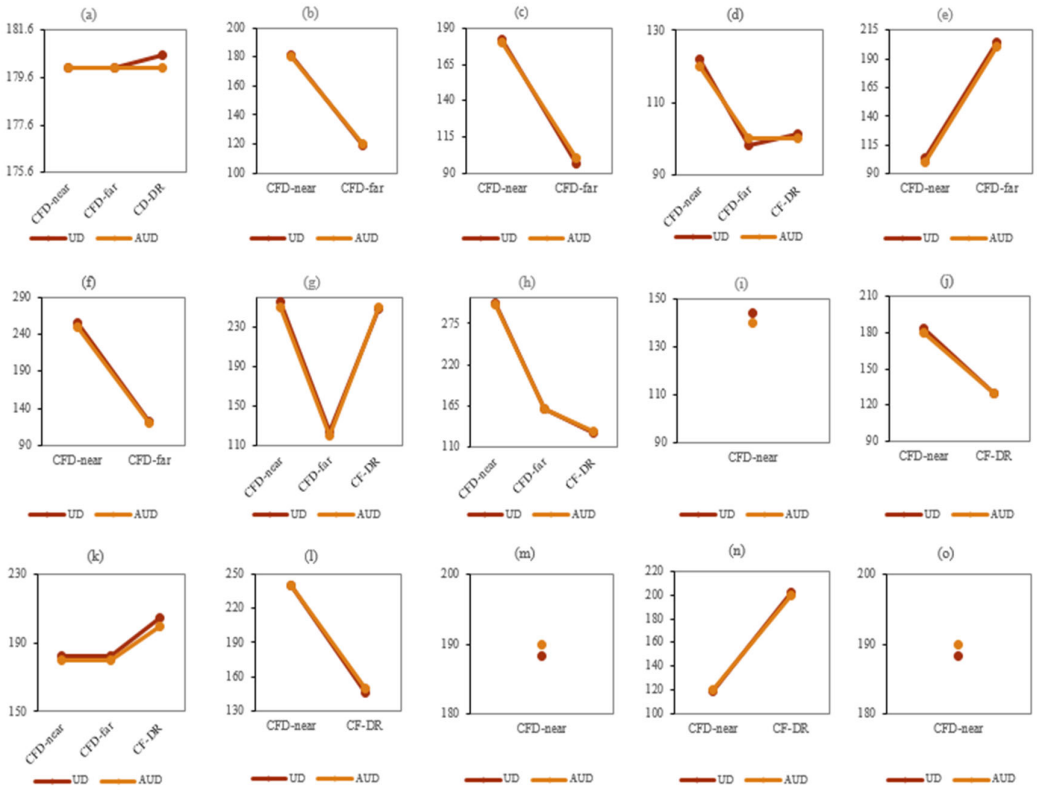


Figure 12. The graphic depiction of DepTSol performance in terms of U_D and AU_D.

7. Conclusions and Future Directions

Social distancing is a highly recommended personal preventive strategy for mitigating the effects of COVID-19. We propose an approach named DepTSol where we mainly focus on low-light scenarios, as such scenarios can play a vital role in the escalation of death and infection rates. We propose a smart implementation of SIoT utilising computer vision and deep learning algorithms with the collaboration of ToF technology and present a cost-efficient and fast, automated social distance monitoring solution. We use a ToF camera to capture people in a real-world environment. People in the images are detected by CSP- ized YOLOv4. In the proposed approach, we calculate the Euclidean distance between the centroids of bounding boxes detected across people and convert distance in cm.

Based on the achieved unit distance, we highlight violations, and a general voice warning is generated to those present in the environment. We evaluated the technique both quantitatively and qualitatively, performed a comparative analysis between different one-stage object detectors, and found that CSP-ized YOLOv4 outperformed all other techniques. Furthermore, the proposed technique achieves outstanding performance in terms of both speed and accuracy, with 51.2 fps and a 99.7% mAP score. The speed and accuracy obtained by DepTSol is higher than those obtained by Adina et al. [8] in their research work, which was 46.2 fps and a 97.84% mAP score, respectively.

In the future, we aim to introduce a new camera calibration technique to resolve the limitations of this study. Furthermore, we aim to extend this approach by adding a facemask detection feature to identify people who are not wearing a mask or who are not wearing a mask correctly at night. Besides this, we will monitor people inside cars and on motorbikes. We will monitor whether the windows of cars are closed or whether people are wearing a facemask, and for bikers, we will ensure that they are wearing a facemask or helmet. In underdeveloped cities, where congested streets similarly play a vital role in the spread of disease, congested roads with minimal distance between traffic can also boost the infection rate.

Author Contributions: Conceptualization, A.R.; methodology, A.R.; Software A.R.; validation, A.R. and A.M. (Ayesha Maqbool); formal analysis, A.R. and A.M. (Ayesha Maqbool); investigation, A.R. and A.M. (Ayesha Maqbool); data curating, A.R. and A.M. (Alina Mirza), F.A. and I.A.; writing original draft, A.R.; writing review and editing, A.R., A.M. (Ayesha Maqbool), and F.A.; visualization; A.M. (Alina Mirza); supervision, A.M. (Ayesha Maqbool); project administration, A.M. (Alina Mirza) and F.A.; funding acquisition, A.M. (Ayesha Maqbool), I.A. and A.M. (Alina Mirza). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is available at Rahim, A. Low-Light-Testing-Dataset-Pakistan. Available online: https://github.com/AdinaRahim/Low-Light-Testing-Dataset-Pakistan-.git (accessed on 1 December 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. WHO. Timeline: WHO's COVID-19 Response. Available online: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/interactive-timeline?gclid=CjwKCAjwgZuDBhBTEiwAXNofRFQ1IcUc8OwIpn7BvGoKmB7P5BoUaxN2DlxxMpc2zXF2pcEXDW6ynBoCaOcQAvD_BwE#event-115 (accessed on 14 April 2021).
2. WHO. WHO Coronavirus (COVID-19) Dashboard. Available online: https://covid19.who.int/ (accessed on 20 April 2021).
3. WHO. COVID-19 Vaccines. Available online: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/covid-19-vaccines (accessed on 4 May 2021).
4. WHO. Coronavirus Disease (COVID-19) Advice for the Public. Available online: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public (accessed on 10 May 2021).
5. Ainslie, K.E.; Walters, C.E.; Fu, H.; Bhatia, S.; Wang, H.; Xi, X.; Baguelin, M.; Bhatt, S.; Boonyasiri, A.; Boyd, O.; et al. Evidence of initial success for China exiting COVID-19 social distancing policy after achieving containment. *Wellcome Open Res.* **2020**, *5*, 81.
6. Rahim, A.; Maqbool, A.; Rana, T. Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera. *PLoS ONE* **2021**, *16*, e0247440.
7. Prem, K.; Liu, Y.; Russell, T.W.; Kucharski, A.J.; Eggo, R.M.; Davies, N.; Jit, M.; Klepac, P.; Flasche, S.; Clifford, S.; et al. The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: A modelling study. *Lancet Public Health* **2020**, *5*, e261–e270.
8. Adolph, C.; Amano, K.; Bang-Jensen, B.; Fullman, N.; Wilkerson, J. Pandemic politics: Timing state-level social distancing responses to COVID-19. *J. Health Politics Policy Law* **2021**, *46*, 211–233.
9. UN. Compendium of Digital Government Initiatives in Response to the COVID-19 Pandemic 2020. Available online: https://publicadministration.un.org/egovkb/Portals/egovkb/Documents/un/2020-Survey/UNDESA%20Compendium%20of%20Digital%20Government%20Initiatives%20in%20Response%20to%20the%20COVID-19%20Pandemic.pdf (accessed on 14 April 2021).

10. Punn, N.S.; Sonbhadra, S.K.; Agarwal, S. Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. *arXiv* **2020**, arXiv:200501385.
11. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:180402767.
12. Sahraoui, Y.; Kerrache, C.A.; Korichi, A.; Nour, B.; Adnane, A.; Hussain, R. DeepDist: A Deep-Learning-Based IoV Framework for Real-Time Objects and Distance Violation Detection. *IEEE Internet Things Mag.* **2020**, *3*, 30–34.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:150601497.
14. Bouhlel, F.; Mliki, H.; Hammami, M. Crowd Behavior Analysis based on Convolutional Neural Network: Social Distancing Control COVID-19. *VISIGRAPP—Proc. Int. Jt. Conf. Comput. Vis. Imaging Comput. Graph. Theory Appl.* **2021**, *5*, 273–280.
15. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:200410934.
16. Bolton, R.J.; Hand, D.J. Statistical fraud detection: A review. *Stat. Sci.* **2002**, *17*, 235–255.
17. Rashidian, A.; Joudaki, H.; Vian, T. No evidence of the effect of the interventions to combat health care fraud and abuse: A systematic review of literature. *PLoS ONE* **2012**, *7*, e41988.
18. Robertson, D.J.; Kramer, R.S.; Burton, A.M. Fraudulent ID using face morphs: Experiments on human and automatic recognition. *PLoS ONE* **2017**, *12*, e0173319.
19. Bruce, V.; Young, A. Understanding face recognition. *Br. J. Psychol.* **1986**, *77*, 305–327.
20. Wang, X.; Jhi, Y.C.; Zhu, S.; Liu, P. Behavior Based Software Theft Detection. In Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; pp. 280–290.
21. Monaro, M.; Gamberini, L.; Sartori, G. The detection of faked identity using unexpected questions and mouse dynamics. *PLoS ONE* **2017**, *12*, e0177851.
22. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 743–761.
23. Alberti, C.F.; Horowitz, T.; Bronstad, P.M.; Bowers, A.R. Visual attention measures predict pedestrian detection in central field loss: A pilot study. *PLoS ONE* **2014**, *9*, e89381.
24. Yao, S.; Wang, T.; Shen, W.; Pan, S.; Chong, Y.; Ding, F. Feature selection and pedestrian detection based on sparse representation. *PLoS ONE* **2015**, *10*, e0134242.
25. Lim, K.; Hong, Y.; Choi, Y.; Byun, H. Real-time traffic sign recognition based on a general purpose GPU and deep-learning. *PLoS ONE* **2017**, *12*, e0173317.
26. Jiang, D.; Huo, L.; Li, Y. Fine-granularity inference and estimations to network traffic for SDN. *PLoS ONE* **2018**, *13*, e0194302.
27. Debashi, M.; Vickers, P. Sonification of network traffic flow for monitoring and situational awareness. *PLoS ONE* **2018**, *13*, e0195948.
28. Kohavi, R.; Rothleder, N.J.; Simoudis, E. Emerging trends in business analytics. *Commun. ACM* **2002**, *45*, 45–48.
29. Wu, C.; Ye, X.; Ren, F.; Wan, Y.; Ning, P.; Du, Q. Spatial and social media data analytics of housing prices in Shenzhen, China. *PLoS ONE* **2016**, *11*, e0164553.
30. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-YOLOv4: Scaling Cross Stage Partial Network. *arXiv* **2020**, arXiv:201108036.
31. Targ, S.; Almeida, D.; Lyman, K. Resnet in resnet: Generalising residual architectures. *arXiv* **2016**, arXiv:160308029.
32. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
33. Li, L. Time-of-Flight Camera–An Introduction. Technical white paper. 2014;(SLOA190B).
34. Weyrich, M.; Klein, P.; Laurowski, M.; Wang, Y. Vision Based Defect Detection on 3D Objects and Path Planning for Processing. In Proceedings of the 9th WSEAS International Conference on ROCOM, 2011.
35. Weingarten, J.W.; Gruener, G.; Siegwart, R. A State-of-the-Art 3D Sensor for Robot Navigation. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; IEEE: 2004 Volume 3, pp. 2155–2160.
36. Yuan, F.; Swadzba, A.; Philippsen, R.; Engin, O.; Hanheide, M.; Wachsmuth, S. Laser-Based Navigation Enhanced with 3d Time-of-Flight Data. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; IEEE: 2009 pp. 2844–2850.
37. Bostelman, R.; Russo, P.; Albus, J.; Hong, T.; Madhavan, R. Applications of a 3D Range Camera towards Healthcare Mobility Aids. In Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, 23–25 April 2006; IEEE: Piscataway, NJ, USA, 2006 pp. 416–421.
38. Penne, J.; Schaller, C.; Hornegger, J.; Kuwert, T. Robust real-time 3D respiratory motion detection using time-of-flight cameras. *Int. J. Comput. Assist. Radiol. Surg.* **2008**, *3*, 427–431.
39. Holz, D.; Schnabel, R.; Droeschel, D.; Stückler, J.; Behnke, S. Towards Semantic Scene Analysis with Time-of-Flight Cameras. In *Robot Soccer World Cup*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 121–132.
40. Castaneda, V.; Mateus, D.; Navab, N. SLAM combining ToF and high-resolution cameras. In Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision (WACV), Kona, HI, USA, 5–7 January 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 672–678.
41. Du, H.; Oggier, T.; Lustenberger, F.; Charbon, E. A Virtual Keyboard Based on True-3D Optical Ranging. In Proceedings of the British Machine Vision Conference, Oxford, UK, 5–8 September 2005; Volume 1, pp. 220–229.

42. Soutschek, S.; Penne, J.; Hornegger, J.; Kornhuber, J. 3-D Gesture-Based Scene Navigation in Medical Imaging Applications Using Time-of-Flight Cameras. In Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008; pp. 1–6.
43. Pycinski, B.; Czajkowska, J.; Badura, P.; Juszczyk, J.; Pietka, E. Time-of-flight camera, optical tracker and computed tomography in pairwise data registration. *PLoS ONE* **2016**, *11*, e0159493.
44. Loh, Y.P.; Chan, C.S. Getting to Know Low-Light Images with the Exclusively Dark Dataset. *Comput. Vis. Image Underst.* **2019**, *178*, 30–42.
45. Rahim, A. Low-Light-Testing-Dataset-Pakistan. Available online: https://github.com/AdinaRahim/Low-Light-Testing-Dataset-Pakistan-.git (accessed on 12 May 2021).
46. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft Coco: Common Objects in Context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
47. Goutte, C.; Gaussier, E. A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation. In *European Conference on Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2005; pp.345–359.
48. Chen, X.; Fang, H.; Lin, T.Y.; Vedantam, R.; Gupta, S.; Dollár, P.; Zitnick, C.L. Microsoft Coco Captions: Data Collection and Evaluation Server. *arXiv* **2015**, arXiv:150400325.
49. Vicente, S.; Carreira, J.; Agapito, L.; Batista, J. Reconstructing Pascal Voc. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 41–48.
50. Willmott, C.J.; Matsuura, K. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Clim. Res.* **2005**, *30*, 79–82.
51. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
52. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
53. Nie, J.; Anwer, R.M.; Cholakkal, H.; Khan, F.S.; Pang, Y.; Shao, L. Enriched Feature Guided Refinement Network for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Kore, 27–28 October 2019; pp. 9537–9546.
54. Huang, Z.; Wang, J.; Fu, X.; Yu, T.; Guo, Y.; Wang, R. DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection. *Inf. Sci.* **2020**, *522*, 241–258.
55. Liu, Y.-C.; Kuo, R.-L.; Shih, S.-R. COVID-19: The first documented coron-avirus pandemic in history. *Biomed. J.* **2020**, *43*, 328–333.
56. Morens, D.M.; Breman, J.G.; Calisher, C.H.; Doherty, P.C.; Hahn, B.H.; Keusch, G.T.; Kramer, L.D.; LeDuc, J.W.; Monath, T.P.; Taubenberger, J.K. The Origin of COVID-19 and Why it Matters.*Am. J. Trop. Med. Hyg.* **2020**, *103*, 955.