Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Development of smart camera systems based on artificial intelligence network for social distance detection to fight against COVID-19

Onur Karaman\textsuperscript{a,1}, Adi Alhudhaif\textsuperscript{b,1}, Kemal Polat\textsuperscript{c,*,1}

\textsuperscript{a} Akdeniz University, Vocational School of Health Services, Department of Medical Imaging Techniques, Antalya, 07070, Turkey
\textsuperscript{b} Department of Computer Science, College of Computer Engineering and Sciences in Al-kharj, Prince Sattam bin Abdulaziz University, P.O. Box 151, Al-Kharj 11942, Saudi Arabia
\textsuperscript{c} Department of Electrical and Electronics Engineering, Bolu Abant Izzet Baysal University, Bolu, Turkey

\textbf{ARTICLE INFO}

\textbf{Article history:}
Received 27 February 2021
Received in revised form 17 May 2021
Accepted 14 June 2021
Available online 24 June 2021

\textbf{Keywords:}
Corona virus (COVID-19)
Deep learning
Convolutional neural network (CNN)
Transfer learning

\textbf{ABSTRACT}

In this work, an artificial intelligence network-based smart camera system prototype, which tracks social distance using a bird’s-eye perspective, has been developed. “MobileNet SSD-v3”, “Faster-R-CNN Inception-v2”, “Faster-R-CNN ResNet-50” models have been utilized to identify people in video sequences. The final prototype based on the Faster R-CNN model is an integrated embedded system that detects social distance with the camera. The software developed using the “Nvidia Jetson Nano” development kit and Raspberry Pi camera module calculates all necessary actions in itself, detects social distance violations, makes audible and light warnings, and reports the results to the server. It is predicted that the developed smart camera prototype can be integrated into public spaces within the “sustainable smart cities,” the scope that the world is on the verge of a change.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

The Novel Coronavirus disease (COVID-19), first identified in Wuhan, China, is a disease caused by the SARS-CoV-2 virus, which causes people to have an acute respiratory infection, and has affected many countries worldwide since December 12, 2019 [1,2]. Increasing rapidly, COVID-19 was declared a “pandemic” by the World Health Organization (WHO) on March 11, 2020 [3].

General symptoms may include dry cough, fever, difficulty breathing, fatigue, and loss of smell and taste. At the same time, in some severe cases, dyspnea and/or hypoxemia occur one week after onset, followed by septic shock, acute respiratory distress syndrome (ARDS), and coagulation dysfunction. Besides, COVID-19 patients have been reported to have gastrointestinal symptoms such as sore throat, runny nose, diarrhea, nausea, and vomiting [4–6]. Although COVID-19 cases are generally reported from all age groups, the most affected ones are those over 30, and the symptoms of the disease progress more seriously in older age groups while the infection rate is lower in children, and young adults and cases are generally reported to be asymptomatic [7,8].

The SARS-CoV-2 virus is generally transmitted through small droplets during the speech, coughing, sneezing, and mostly by inhalation between individuals in close contact, especially in closed and poorly ventilated areas. Droplets are transported to the lungs via the respiratory system, where they begin to infect lung cells [9]. Since the virus can be transmitted through direct or indirect contact with mucous membranes in the eyes, mouth, or nose, individuals can also become infected through contact with contaminated objects [10].

The rapid spread of the disease, the increasing number of cases, and the death rate, as well as the insufficient treatment methods and vaccine options, have encouraged many governments and health authorities to take strict measures to combat the pandemic, such as quarantine, travel and movement restrictions, cancellation of non-mandatory meetings and gathering events [11]. The data obtained as a result of the researches show that countries implementing decisive and early interventions have been able to reduce and control the spread of the disease [12,13]. The strategies proposed by WHO to stop/slow down the spread of the COVID-19 virus primarily consist of hygiene, mask, and social distance.

Social distance (physical distance) measures include several critical activities aimed at slowing/stopping the virus's spread by maintaining the physical distance between individuals and reducing the number of people coming into contact with each other [14].

The countries adopted different strategies and set different social distance thresholds as part of the fight against COVID-19.
According to the World Health Organization, the social distance to be kept is “at least 1 m” [15]. In contrast, the social distance limit recommended by some countries, such as South Korea, the United States, Canada, and the United Kingdom, is about 2 m (6 ft). These limitations are based on the fact that the droplets formed when talking cannot spread to a distance of more than 1 meter. However, this distance will not be enough in case of coughing or sneezing. In this case, it is clear that other measures, such as hygiene and masks, should be applied along with them. In their studies, Jones et al. (2020) showed that there might be a difference in risk for conditions such as ambient ventilation, open/closed area, speech volume/silence, mask use, and period of contact [16]. According to this study, the social distance should not be reduced to less than 2 m in high-risk situations (especially in poorly ventilated and confined areas), while at least 1 m will be sufficient in low-risk scenarios (in open areas and cases of mask use).

However, it should not be ignored that infected people should isolate themselves in all cases. The works in literature have shown that a number of non-drug interventions implemented in China, including measures of social distance, have significantly reduced the number of those infected when performed 1–3 weeks earlier and decreased the number of cases by 66%–86%. It is suggested that if social distance measures are applied in the early stages of disease detection, they may play a crucial role in fighting the virus and preventing the peak of the pandemic [17].

The strategy to be implemented must be comprehensively determined first and followed, recorded, analyzed, evaluated to understand whether it is effective or not in order to minimize the challenges that may be encountered and to provide the most effective solution in the struggle, which is at least as difficult as its implementation. Applying and making these rules adopted by society is a difficult prerequisite, and it is a rule that employees have difficulty in implementing, especially in businesses that continue to work during the quarantine period. In such cases, it becomes important to develop new technologies such as “smart camera systems” that can be easily integrated into security cameras in order to control the social distance rule in the working environment. In recent years, computer vision, machine learning, and deep learning have presented promising results in many daily life problems. Recent advances in deep learning make object detection tasks more effective [18]. In technological ecosystems, the data on traffic, noise, air quality, energy consumption, and movement are collected for improved, evidence-based sustainable decision-making processes. In this regard, the importance of developing and implementing advanced technologies that can help the health systems, the governments, and the public in various aspects during the fight against COVID-19 is growing rapidly [19].

Various studies have been presented in the literature to determine whether the “social distance” rule applied in the fight against the pandemic has been violated and to help reduce the risk of transmission or control the outbreak. The study proposed by Bian et al. (2020) was conducted using sensors that do not use machine learning algorithms. In this proposed study, a wearable, magnetic field-based proximity sensing system was developed to monitor the distance. The wearable system has a detection range of over 2 m and can efficiently track an individual’s social distance in real-time [20]. Punn et al. (2020) [21] used the YOLOv3 model to identify people and proposed a method that uses the Deepsort approach to enclose identified people in bounding boxes and track them using the identity information of these people. They used the Open Image Dataset (OID) repository, which is a front view dataset. They also compared their results with R-CNN and SSD. In their study, Ramadass et al. (2020) proposed a drone-based model for automatic social distance monitoring by training the YOLOv3 model with their own data set. The data set they used consists of front and side views of a limited number of people. They also expanded their study to track the use of masks. Using the drone camera and the YOLOv3 algorithm, they tried to determine social distance and mask used with the side or front views [22].

Sathyamoorthy et al. (2020) have developed a mobile robot that includes commodity sensors, an RGB camera, and a 2D lidar to perform collision-free navigation and estimate the distance between all detected people. In this system, they used the YOLOv3 model to track ~2 m (6 ft) distance between people with the model they developed for automatic detection of human pairs in crowded environments. In the related study, they transformed the angled view of the ground plane of the camera by applying a homography transformation to four manually selected points on the ground plane in the angle view. Thus they were able to estimate the distances between people [23]. In the literature, there are similar studies in which homography transformation is used to estimate the 3-dimensional (3D) coordinates of humans. Yang et al. (2020) used YOLOv4 and Faster R-CNN object detectors to detect people’s images and compared their performances. The concerned study applied an inverse homography transformation called bird’s eye view (BEV) after identifying the bounding boxes on individuals [24]. Khandelwal et al. (2020) developed an algorithm to send real-time voice alerts to employees who violate the social distance rule, using a similar approach in CCTV cameras that monitor the workplace [25].

As can be seen from all these examples, the artificial intelligence applications in smart city services aimed at improving and facilitating the quality of life are of great importance in combating the COVID-19 pandemic. It is assumed that this issue comes to the forefront in the design of smart cities to monitor the emergence and spread of infectious diseases more closely and develop an early warning system. Bearing all in mind, it has been concluded that a significant amount of studies has been conducted for tracking social distance in public environments. However, most researchers focused on a front or side perspective for social distance tracking. Therefore, the aim of this study is to conduct deep learning-based social distance tracking using a birds-eye perspective, which offers a better view and will play a key role in calculating the distance between the individuals.

2. Materials and methods

2.1. Detection of human coordinates from artificial intelligence-assisted images

Identifying the coordinates of the people in the camera images is the first step of this research. At this stage, convolutional neural networks (CNN), which have become popular in the last decade, have been used to define the artificial intelligence-assisted human detection model to be applied.

Human detection from the image can be done with artificial intelligence models trained in advance with large databases such as COCO and PASCAL VOC 2012 [28]. In this study, it was decided to use the models “MobileNet SSD v3”, “Faster RCNN Inception v2” and “Faster RCNN ResNet 50” trained by OpenCV community with COCO database, taking into account the mean average precision (mAP) scores and GPU computation times of the models [27–29]. When selecting the models, the security camera images found in the open-source “Oxford Town Centre Dataset” were used [30]. Artificial neural network models in “.pb” and “.pbtxt” file formats, previously trained by the OpenCV community with TensorFlow, were included in the deep neural networks (DNN) module of the OpenCV library using the readNetFromTensorFlow method contained in the DNN module [31].
2.2. Detection of distance between people

At this stage of work, the coordinates of the people on the image, which are the outputs of the artificial intelligence-assisted human identification software layer, were used as the input of the distance detection software layer, and the distance between people was estimated. Since the distance between people cannot be calculated by measuring pixels directly due to distortion caused by perspective, the perspective in images has been transformed into a bird’s eye view by calculating the transformation matrix (TM) using the “getPerspectiveTransform” method found in the OpenCV library [32]. Thus, the distances between the detected persons were determined by the pixel/meter scale. The transformation matrix (TM) is a 3 × 3 matrix and the equation is defined in Eqs. (1) and (2) [33].

\[
\begin{bmatrix}
    x'_{i} \\
    y'_{i} \\
    1
\end{bmatrix}
= \text{TM} \cdot
\begin{bmatrix}
    x_{i} \\
    y_{i} \\
    1
\end{bmatrix}
\]

(1)

\[
dst (i) = (x', y'), \ src (i) = (x_{i}, y_{i}), i
\]

(2)

Here, “dst” is the target matrix obtained after transformation, and “src” is the source matrix that is transformed. It is computed to convert each element of the source matrix into each element of the target matrix.

During the computation, it is expected that the user who will use the developed prototype will specify the source and target points. The user selects a rectangle on the source image with two opposite sides parallel to each other on the camera image, then the perspective transformation matrix is calculated for the coordinates indicating the lowest middle positions of people using the coordinates of the selected rectangle (Eqs. (1) and (2)).

After calculating TM, a bird’s-eye view image matrix is obtained by multiplying each element of the source matrix (source video frame). The Eq. (3), and the “warpperspective” method of OpenCV library is used while performing this process [34].

\[
\text{src} (x, y) = \begin{bmatrix}
    \text{TM}_{11}x, \text{TM}_{12}y, \text{TM}_{13} \\
    \text{TM}_{21}x, \text{TM}_{22}y, \text{TM}_{23} \\
    \text{TM}_{31}x, \text{TM}_{32}y, \text{TM}_{33}
\end{bmatrix}
\]

(3)

The users who violated the social distance rule are identified using 125 pixels/m, the reference length per pixel determined in the open-source “Oxford Town Centre Dataset” from which the camera images are taken based on the human coordinates and bird’s eye view images obtained using the transformation matrix.

2.3. Embedded system studies

Artificial intelligence-assisted social distance detection software was run on an embedded system on a chip called Nvidia Jetson Nano as it has advantages such as a quad-core ARM A57 processor, 4 GB LPDDR4 RAM, 2 MPI CSI-2 DPHY camera connections, and GPIO pins.

Using the Jetson Nano development card, it was intended to perform computations at a rate of at least one frame per second. The artificial intelligence model must be accelerated by the GPU contained within the Jetson Nano. Therefore, it is necessary to use the model and pre-trained weights together with the most appropriate software to work with the GPU. Studies were conducted on software environments such as TensorFlow, TensorFlow Light, NVIDIA TensorRT, and OpenCV DNN Module [35]. These types of software frameworks allow real-time operation of models with 8-bit integer and 16-bit floating-point optimizations. A choice has been made between described software environments that will achieve minimal losses in mAP values and also take into account the defined computation time.

2.4. Web-based control panel

The analysis of the computations made on the camera images is as important as the sudden feedback to the users. Therefore, a web-based control panel has been developed, and instant camera images and analysis, transformation key points, sensor information, calibration information, a heat map, and social distance violation metrics are included on this panel.

In order to control the camera over the internet, the Python programming language and the Flask web roof were used [36]. All functions of the camera can be monitored and adjusted on the control panel. The web server coming with the Flask web roof is installed on top of the camera. This server can distribute all camera functions online to the internet or to the intranet according to the network configuration. The camera’s IP address can be accessed via port 9091 with an HTML5 supported browser.

2.5. Electronic design and software

In the prototype of the camera system, an extension circuit was developed that connects to the 40-pin GPIO connectors and receives its power from these pins included in the Jetson Nano development kit so that the camera can generate visual and audio warnings, read and report temperature, humidity and acceleration data. In this extension circuit, MPU6050 3 axis accelerometer and three-axis gyroscope integrated module, HDC1080 temperature and humidity sensor module, MAX98357 A PCM Class-D amplifier integrated module, 8 Ω-1 W speaker, 3.3 V/5 V logic level converter circuit, and WS2812B programmable 5 × 5 RGB LED matrix were used. The simplified schematic of this electronic extension circuit connected to the Jetson Nano development board is given in Fig. 1.

2.6. Mechanical design

The mechanical design process of the prototype was carried out with the SOLIDWORKS software. The design consists of a box to contain the Jetson Nano development board and adjustable attachments to hold the camera and LED matrix, and it was made in a form to be printed from a 3-dimension (3D) printer (AnyCubik brand). During the design, the product box was printed on a 3D printer using white polyactic acid (PLA) filament, and metal fasteners were preferred. The prototype was placed on the roof of the Akdeniz University Dining Hall, which was designated as the pilot application area, so as not to be exposed to rain/water.

3. Results

3.1. Artificial intelligence-assisted human identification layer

In this layer, the raw sensor data obtained through the camera sensor is processed, and sections containing people in the image are identified in rectangular bounding boxes. For these operations, different models can be used with classical machine learning methods, especially HOG (Histogram of Gradients) attributes, and linear support vector machine regression models. The use of deep learning, which is another machine learning technique, has increased rapidly in recent years and today forms the basis of computer vision together with convolutional neural networks (CNN). Although the achievements of convolutional neural networks in the field of computer vision have proven themselves in many areas, they are now known as the most powerful tool in object detection.

In this study, the convolutional neural networks were preferred as the object (human) detection layer to detect the coordinates of people from the camera. In object detection applications,
three outstanding meta-architectures are featured in the literature, including Faster RCNN [26], R-FCN [37], and SSD [38]. These meta-architectures are combined with different types of convolution layers used for attribute extraction, thus creating hybrid structures used for object detection. In this work, there are two basic parameters that are given importance. One of them is the mean average precision (mAP) value, and the other is the computation time. When the mAP is higher and the computation time is lower, the performance of the system that we will use on the embedded system will increase. The graph presented in Fig. 2, shows a scatter graph of GPU computation times versus mean average precision.

Object detection models trained with the COCO database are capable of detecting the position of a total of 80 types of objects on the image, such as people, dogs, cows, trains, cars, motorcycles, chairs, sofas. Since this study was intended to detect only humans, the remaining 79 classes detected by the models were ignored, and the results of only humans were examined.

In the reviews of video sequences, a threshold of 0.25 was determined for the estimates made by the models, and the estimated probabilities below this threshold were ignored. In Fig. 3, the results of the “Faster RCNN Inception v2”, “MobileNet SSD v3”...
and “Faster RCNN ResNet 50” models are presented for comparison of object identification models. In the reviews of video sequences, it was realized that the results were in accordance with the graph given in Fig. 2. It was obvious that “Faster RCNN Inception v2” was able to detect people’s positions more accurately with high performance compared to other models.

3.2. Bird’s-eye view transformation

In a system composed of lenses with different focal lengths and working with a single camera, it is not possible to measure the dimensions of the features to be extracted from the image by using an uncontrollably selected camera height and angle. In order to determine the social distance from the camera, the height of all people can be determined as 1.60 m, which is the average human height for Turkey, and the length of the vertical components of the rectangles that detect people can be considered as 1.60 m \(^{[40]}\), and therefore the distance between the nearest another person can be calculated by preserving this measure. But this method depends on the camera angle and will lose its precision as the object moves away from the camera. The intersection of the rectangles detecting humans near the camera and far from the camera on the horizontal axis will cause the algorithm to perform a miscalculation. The wide-angle cameras will further increase the perspective effect so that the distance between people who are away will appear smaller than the distance between people who are close by. In order to eliminate this problem, the user will need to perform calibration after the camera is positioned in the desired spot for measurement.

As a result of the examination of the images obtained with the camera, it was found that the boxes identifying nearby people were larger, and the boxes identifying distant people were smaller due to the perspective effect. Because of the distortion caused by perspective, the distance between people cannot be calculated by measuring pixels directly. Therefore, at the next stage, it was decided to convert the perspective view into the bird’s eye view. In this way, the distances between the detected persons can be determined with the pixel/meter scale. In order to make a bird’s eye view transformation, the necessary matrix that allows transformation to a bird’s eye view must be calculated. The transformation matrix (TM) was calculated using the “getPerspectiveTransform” method found in the OpenCV library (Eqs. (1) and (2)). When calculating, it is expected that the person using the prototype will specify the source and target points. Therefore, in the tests carried out, the task of detecting the coordinate points determined on the source image was given to the user who would use the prototype. After calculating the Transformation Matrix, a bird’s-eye view image matrix was obtained by multiplying each element of the source matrix (source video frame). The Eq. (3), and the “warpPerspective” method of OpenCV library were used while performing this process.

The perspective image is presented in Fig. 4a, and the region to be transformed on the source image is visualized with the Blue rectangle. In Fig. 4b, the bird’s-eye view obtained as a result of the transformation is visualized with a black-and-white color scale.

In order to identify the coordinates where people are located on the bird’s eye view, the coordinates of the rectangles formed by the object detection layer meter must be taken as the reference. When these coordinates are taken as references, the center point of the lower edges of the rectangles that describe people is considered the reference point where people come into contact with the plane they walk on. In this way, the points where people’s feet step on the ground can be visualized on a bird’s-eye view transformation, and the positions of people can be defined. In Fig. 5a, the mentioned center points are given, and in Fig. 5b, the version of the coordinates of these center points transformed into a bird’s-eye view using the transformation matrix given in Eq. (1) and Eq. (2) is visualized. By dimming the bird’s eye view and visualizing it in black and white, it was intended to make the points easier to understand visually.
3.3. Determination of distance violations on images

After determining the coordinates of the people on the bird’s eye view, the pixels in the bird’s eye view must be expressed in meters in order to calculate the distance between these obtained coordinates. In this work, each time the camera angle is determined, a reference length should be used to calibrate the bird’s-eye view. Distance violations are determined using the specified ratio of reference length per pixel. In images taken from the “Oxford Town Centre” database used for testing purposes, this value is approximately 125 pixels/m.

While the World Health Organization recommends a social distance of at least 1 meter, the Turkish Ministry of Health has set the social distance limit at 1.5 m. In the UK, the social distance was reduced from 2 m to 1 m by last July. Taking into account all these explanations, two levels of violations were determined as red and yellow. Violations of the 1-m limit are called the “Red Alert” while the violations of the 2-m limit are called the “Yellow Alert”. The number of instant violations for the image taken from the “Oxford Town Centre” database is visualized in Fig. 6.

3.4. Embedded system studies

The camera system is run on the Nvidia Jetson Nano Linux development kit. It has a GPU with Jetson 128-core Maxwell architecture, an ARM A57 4-core CPU running at 1.43 GHz, and...
4 GB 64 bit LPDDR4 memory. Due to its performance in real-time computer vision applications, it is often used in research in this field. In Fig. S1, a digital image of the Jetson development kit used as part of the work is presented. It is intended to compute the data from the camera buffer of the developed software by attaching a camera module called “RaspberryPi Camera Module v2” to the CSI connectors on the Jetson Nano development kit. The connection was made to the camera using the CSI-Camera [41] library, and input was provided to the software.

The software is configured to run on the device in real-time, give an alarm in case of violation and share results with the remote database. The OpenCV DNN module could not exceed the average FPS 0.3 because it only works on the CPU. Therefore, after the object detection model was tested in the TensorFlow environment to use GPU acceleration, the model was converted to formats suitable for the TensorFlow Light and TensorFlow environment and then tested. The model was also compiled again to ensure CUDA compatibility with the appropriate configuration parameters in this study so that the model could be tested with the CUDA-supported OpenCV DNN [42] module, which was unofficially developed by the OpenCV community and then tested on the CUDA-compatible OpenCV library.

In Table 1, the response time of the model was examined in different environments, and it was found that the use of the OpenCV DNN module on the CPU was not suitable with Jetson Nano. The OpenCV DNN module is a customized module for Intel processors and performs highly on Intel processors and graphics chips. This module, which is very useful in the server environment, is not suitable for device groups called “EDGE Computing” such as Jetson. Although the high performance of the TensorRT environment is very suitable for Jetson, it is aimed that the camera software developed will have an infrastructure that can work not only in the camera but also in the server environment in order to be able to track social distance through security cameras in the future. For this reason, the model was run with CUDA support in the OpenCV DNN module, and an FPS refresh value of 2.1 was deemed more appropriate, including other operations other than object detection. Here, areas marked with red are environments that calculate in more than 1 s, and greens are environments that calculate in less than 1 s.

| Environment   | Mean FPS |
|---------------|----------|
| OpenCV DNN (CPU) | 0.3      |
| OpenCV DNN (CUDA) | 2.1      |
| TensorFlow   | 1.3      |
| TensorFlow Light | 4.1      |
| TensorFlowRT  | 9.5      |

The device does not include a built-in WiFi module and only offers an Ethernet connection. Therefore, a USB-WiFi-Modem with 802.11 n IEEE standard was connected to the device.

For developing the Pilot application and web-based control panel with real data, the camera prototype was established on the roof of the Akdeniz University Dining Hall to detect and analyze people entering/leaving the dining hall. After all, additional electronic and mechanical components were included in the system. For these records, the relevant permissions were obtained from Akdeniz University.

3.5. Web-based control panel

A web-based control panel has been developed to detect people using an object detection model and convert the determined coordinates into a bird’s-eye view, and then calibrate the distances between these coordinates to track the results of the camera system that finds violations of social distance and works on the embedded system, to examine the reports and to adjust the camera settings. At this stage of the work, the entire software chain analyzes and processes the images coming from the camera on a live basis.

3.5.1. Snapshots and analysis

This section, which allows instant tracking of images coming from the camera, the real-time human detection of the camera, perspective transformation, and live number and location of red-and-yellow violations, can be tracked on the page “Snapshot”. The software, which can process about two frames per second with GPU support, composites images in JPEG format and saves them on memory in base64 format after completing coloring, transformation, and violation operations with the CPU. Clients entering the page “Snapshot” can follow base64 format files by updating the processed image on the server with renewed requests at time intervals determined by the JavaScript code block found in the Flask application. Since the camera software uses GPU support, it completes the operations of the previous frame and server operations on the new “threads” during the GPU’s processing. Because GPU operations are not waited for, the interface and additional operations work smoothly.

3.5.2. Transformation key points

In this section of the control panel, it is provided that the user determines the key points necessary for perspective transformation. A block of code written in JavaScript works in this section. This code block controls a group of SVG objects consisting of 1 rectangle and 4 circles. These key points, which can be held and dragged with a computer mouse, save the X and Y coordinates of the key points in the settings document located on the “Mongo DB” database when the button “Save Settings” is pressed. It is possible to zoom in and out with the mouse wheel. In this way, the user determines how far the key points can be selected outside the image boundaries. When the button “restart camera” is pressed, the camera software starts again to read and use the specified coordinates and perform the perspective transformation process.

3.5.3. Sensor information

On the sensors page of the control panel, data from the accelerometer sensor and temperature sensor are read and presented to the user. By analyzing the data from the acceleration sensor, it is determined whether the orientation of the device has changed. For this, the software layer that reads and analyzes the data from the accelerometer sensor checks whether the data from the sensors exceeds the set threshold value after the camera application is launched. If the set threshold value has been exceeded, it is assumed that the camera moves with an external force, and its orientation has changed. In this way, the user will be able to understand whether the position of the camera has been changed, so they will also be able to predict whether they will need to correct the perspective transformation key points again. Acceleration data is recorded in the database at intervals of 30 s.

The data coming from the HDC1080 temperature and humidity sensors in the device are also visualized on this page. Instant temperature and humidity information of the device can be tracked here in real-time. Temperature and humidity values are recorded in the database at intervals of 30 s together with the date top data. Finally, sensor data is presented on this page. In future versions of the software, it is planned to visualize the change of sensor data over time on real-time graphs on this page.
3.5.4. Calibration

After perspective transformation, the length per pixel must be measured in order to detect actual social distance violations on the resulting bird’s-eye image. This setting, which must be manually performed by the user depending on the camera position, can be made from the calibration page of the control panel. The numeric value entered in the “pixel/meter” text field can be saved in the Settings section on the database by pressing the “Save Settings” button, and the camera software can be restarted using the corresponding setting in the database by pressing the “restart camera” button.

3.5.5. Heat map and violation metrics

On this page, the densities of people in the relevant region in the selected time interval are visualized with the heat map. The image resulting from this calculation is decoded in base64 format and converted to the format that Flask will display on the web server and visualized in the user interface.

In addition to the heat map, a metric called “number of violations x duration of violations” has been created to eliminate erroneous object detection and obtain more usable statistical information in terms of transmission. This metric is recorded for the selected time interval, similar to the heat map. In order for this metric to give more accurate results, the FPS value of the camera is also taken into account. In the analysis, the time between two frames from the camera for each violation in the image is multiplied by the total number of violations. This “number of violations x duration of violations” value obtained for one frame is calculated individually for each frame, and the “total number of violations x duration of violations” value is calculated by summing it over all frames in the corresponding time interval.

The “reporting” page presented shows the violation-second metric and heat map calculated for the time interval selected through the pilot application system installed in the rooftop of the Aldeniz University Dining Hall.

3.6. Electronic design and software

This work aims to provide instant visual/audio warnings to the people violating social distance and raise awareness in this regard. The Jetson Nano development card has the ability to react quickly to its operations with built-in digital I2S audio output and GPIO ports. In Figure S2, the digital image on the left shows an electronic extension circuit and the speaker used to connect the Jetson Nano development card to the 40-pin GPIO port. The additional circuit includes a temperature–humidity sensor, an acceleration sensor, a sound amplifier, and a logical level converter circuit. In Figure S2., the digital image on the right shows the extension circuit connected to the Jetson Nano development kit. WS2812B LED Matrix is fixed by soldering to logical level converter circuit output, 5 V power pin, and GND pins just before boxing.

3.6.1. Sensors

Acceleration, temperature, and humidity sensors connected to the Jetson Development Board communicate with Jetson via the I2C bus and transfer data to Jetson. A library called “CircuitPython” [43] was used to connect with sensors.

Integrated MPU6050 is used as the accelerometer sensor. MPU6050 is both a 3-axis accelerometer and a 3-axis gyroscope integrated. However, only the acceleration measurement feature was used in this work. The I2C drivers of this integrated circuit were developed within CircuitPython. Using the developed software layer, the sensor’s X, Y, and Z acceleration values are measured at 30-s intervals, and the values are recorded in the database. If any of the measured acceleration values of X, Y and Z exceeds ±100 mg, the camera is assumed to have been moved and this is recorded in the database as “threshold value passed”.

The Jetson Development Board is easily heated due to power consumption and has to expel the thermal energy it produces. Therefore, an additional fan is installed in the passive cooler located on the Jetson Nano development board. For this reason, it is planned to measure the temperature and humidity values in the box with a temperature and humidity sensor called HDC1080 installed on the device. As with the acceleration sensor, Jetson Nano’s I2C pins and CircuitPython library were used for this. Temperature and humidity values read at intervals of 30 s through the sensor are recorded in the database along with date top data.

3.6.2. Audio and visual warning system

When the developed software layers determine a violation, this situation is turned into an audible and visual alarm. The determined critical condition is triggered if the social distance violation continues for 2 s. After triggering, the warning sound and warning light work for 2 s, and then if the situation is repeated, the warning continues by repeating in the same way.

MAX98357 A [44] PCM Class D amplifier integration is used for audio warnings. Class D amplifiers are preferred because they consume little power due to their structure. This integrated Jetson Nano development kit converts the digital audio information received through the I2S pins into an analog signal and amplifies the converted signal with the built-in Class-D amplifier. The amplified signal is connected to a speaker with a coil resistance of 8 Ω and can withstand a maximum power of 1 W. Jetson Nano ASOC drivers [45] are used to enable MAX98357 A to work together with Jetson Nano.

A 5 × 5 Matrix consisting of RGB LEDs called WS2812B was used to create visual alerts. The most important feature of WS2812B integrated LEDs is that the drive circuits are located in the LED package and can be programmed numerically using the single-wire data communication protocol. The output of each element of the LED matrix used depends on the input of the next element. Thanks to this chain link, which exists both along the row and between the columns, the indices of LEDs can be controlled numerically. In LED software developed using the CircuitPython library, the LED is illuminated in the event of a violation by assigning appropriate color values to each LED (Fig. 7).

Since the LED matrix is designed to meet the 5 V digital voltage level and Jetson Nano’s GPIO pins are suitable for 3.3 V, the voltage levels in the two environments have been converted. The circuit that performs the conversion is presented in Fig. 8. The circuit, prepared using two 10k resistors and one BSS138 N-channel MOSFET transistor, converts the 3.3 V signal received from the Jetson Nano development board to the 5 V voltage level and sends the signal at the 5 V logical level to the 5 × 5 RGB LED matrix input.
3.7. Mechanical design

The mechanical designs of the work were completed with the SolidWorks program. The design consists of a box to cover the Jetson Nano development board and adjustable inserts to hold the camera and LED matrix. The design was made using an AnyCubic 3-dimensional (3D) printer and a white PLA-type filament. The printing stages of the models were given in Figure S3.

After removing the top and bottom covers of the box, camera and LED holders from the 3D printer, the circuit board developed in Jetson Nano was mounted via 40-pin GPIO ports. After that, electronic components and a box made of PLA material were installed. In Figure S4, digital images of the assembly stage are presented.

4. Discussion

One of the main measures that can be implemented in the fight against COVID-19, the “social distance rule”, is that individuals stay at least 1 m away from each other. Artificial intelligence applications are seen as a revolution all over the world and have a wide range of applications. Its application in the health sector has also improved in many ways. This study aims to develop a prototype and control panel of an Artificial intelligence-assisted embedded camera system that will help track, record, and evaluate the movements of individuals in real-time in metropolitan cities where social distance measures are of great importance.

The “MobileNet SSD v3”, “Faster RCNN Inception v2”, “Faster RCNN ResNet 50” models trained by the OpenCV community with COCO database have been tested to identify people in-camera images. Security camera footage from the open-source “Oxford Town Centre Data Set” was used to select the most suitable model. With the developed model, people identified with the help of bounding boxes can be identified. By accepting the lowest points of these bounding rectangles as reference points, bird’s-eye view images were obtained, and using distance calibration, and Euclidean distances between reference points, the binary distances of people were determined.

The as-produced prototype is an integrated embedded system that performs social distance detection with the camera. The system is based on the Nvidia Jetson Nano " development kit. The software developed using the Jetson Nano development kit, and Raspberry Pi Camera module calculates all the necessary operations within itself, detects social distance violations, makes audio and light alerts, and reports the results to the server.

A pilot implementation process was initiated, and the prototype was tested by installing the artificial intelligence-assisted camera prototype on the rooftop of the Akdeniz University Dinning Hall in accordance with the necessary permissions. Social distance monitoring of people entering and leaving the dining hall was carried out using the camera prototype developed in the study.

The length of the paving stones in front of the dining hall was measured as the reference length for calibration before the camera was installed. The stones were square in shape, and the edge of each stone was measured as 40 cm (Figure S5).

In this pilot application, the results were examined through the control panel by connecting to the camera via Wi-Fi. In order to make the results more understandable, an additional code added to the application allowed the video card to save the corresponding frame taken from the camera after calculation without compressing it in PNG file format with date top data.

After the pilot application, each frame of images recorded with a screen refresh rate of about 2 FPS was manually passed through the developed software layer, and all the accounts made were visualized on a video. A sample frame in the selected time interval over this video is presented in Fig. 9. As can be seen from Fig. 9., the developed camera system was capable of detecting all the humans in perspective view by red rectangles and creating heat maps of the distances. Moreover, the violation of the social distance was successfully detected and visualized by the red alerts.

On the online control panel, the monitoring of perspective transformation and calibration settings of the camera prototype, reading sensor data, and, most importantly, social distance tracking has been successfully achieved. The refresh rate of about 2 FPS provided sufficient temporal detail for social distance tracking. Fig. 10 shows a photo from the development stage of the camera prototype. The photo shows a warning light running on a 5 × 5 RGB LED matrix in case of violation.

Thanks to the final developed prototype,

- The awareness of individuals can be increased with the warning system that is activated when individuals violate social distance limits.
- The change of parameters such as the number of people and the number of violations will be able to be analyzed statistically depending on the time.
- It will be statistically determined to what extent people comply with the measures and sanctions taken and to what extent the social distance is effective.
- Thanks to the embedded sensors included in this system, which will be part of the global movement for the development of Smart Cities, real-time data will be provided for the evidence-based decision-making process.
- It will be possible to make improvements in the economy, tourism, and psychological areas where these measures cause serious problems.
- The developed prototype will be able to coordinate within the framework of security cameras to effectively scan the working environment, especially with integration into professional enterprises engaged in strategic production and medical organizations. In addition, data will be provided to help auditors in the strategic manufacturing business reorganize their work areas.

The main limitations of the final prototype can be listed as follows;

- The developed system faces difficulties in detecting people in the dark since the camera does not have night vision.
- It needs calibration after assembling.
- Since it is developed on the embedded system, the FPS values are relatively low.

5. Conclusion

In this work, a deep CNN-based social distance monitoring model was developed using a bird’s-eye perspective. The results confirmed that the developed artificial intelligence model
effectively-identified people who violated social distance and showed the necessary warnings. Thanks to the integration of the developed artificial intelligence-assisted intelligent camera system prototype into various public spaces such as shopping malls, tourism facilities, health institutions, strategic manufacturing enterprises/factories, areas where artistic and sporting events are held, and worshipping areas, it will be possible to monitor the movement dynamics in real-time and slow/prevent the spread of the virus [46–49]. In this algorithm, which can measure the distance between individuals, the individuals who comply with the social distance rule can be marked as green, and the situations in which social distance is significantly violated can also be marked as Red by using the traffic light indicator system. However, it was also determined from the images obtained with the developed prototype that people who were overshadowed were difficult to detect. Therefore, in order for the object detection model to work properly, the camera prototype must be positioned in illuminated locations.

Statistical analysis of the data is of great importance for possible future crisis management, as it will provide detailed information about long-term behavioral changes. Making the proposed system applicable in many public areas such as places of worship, artistic and sporting activity areas, shopping malls, airports, workplaces, tourism facilities, the impact of "social distance" within the scope of the fight against COVID-19 measures on the fields such as social norms, economics, psychology will be minimized. In future studies, it is planned to analyze the data obtained statistically. In this way, regions and/or time zones where social distance cannot be maintained will be identified, and it will be possible to determine how individuals can adapt to restrictions or how the measures taken can be regulated.

Funding

This work is supported by Scientific and Technological Research Council of Turkey (TÜBİTAK) via Grant No. 120E124.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and material

Data are available on request from the author

Acknowledgments

The authors introduce their thanks to Çağın Polat (Notrino Research, ODTU Teknokent, Ankara, Turkey) for his contribution to the prototype development stage, and Serdar İplikçi (Department of Electricity and Electronics Pamukkale University, Denizli, Turkey), for his project supervision. All authors read and approved the final manuscript.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2021.107610.
References

[1] W.G. Carlos, C.S. Dela Cruz, B. Cao, S. Pasnick, S. Jamil, Novel Wuhan (2019-nCoV) coronavirus, Am. J. Respir. Crit. Care Med. 201 (2020) 7–8.

[2] H. Li, S.M. Liu, X.H. Yu, S.L. Tang, C.K. Tang, Coronavirus disease 2019 (COVID-19): current status and future perspective, Int. J. Antimicrob. Agents. 55 (2020) 105951. http://dx.doi.org/10.1016/j.ijantimicag.2020.105951.

[3] X. Wang, P. Chen, J. Wang, F. Feng, H. Zhu, X. Li, et al., Evolution of the novel coronavirus from the ongoing wuhan outbreak and modeling of its spike protein for risk of human transmission, Sci. China Life Sci. 63 (2020) 457–460. http://dx.doi.org/10.1007/s11427-020-1637-5.

[4] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, et al., Clinical characteristics of 1138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China, JAMA 323 (2020) 1061–1069. http://dx.doi.org/10.1001/jama.2020.1585.

[5] M.L. Holshue, C. Debiol, S. Lindquist, K.H. Lofy, J. Wiesman, H. Bruce, et al., First case of 2019 novel coronavirus in the United States, N. Engl. J. Med. 382 (2020) 929–936. http://dx.doi.org/10.1056/NEJMa2001951.

[6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, et al., Speed/accuracy trade-offs for modern convolutional object detectors, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7310–7311.

[7] C.Bulut, Y.Kato, Epidemiology of COVID-19, Turk. J. Med. Sci. 50 (2020) 3223, http://dx.doi.org/10.1542/peds.2020-0702.

[8] Y. Dong, X. Mo, Y. Hu, X. Qi, F. Jiang, Z. Jiang, et al., Epidemiology of 2019-nCoV transmission through the ocular surface must be not be ignored, Lancet 395 (2020) 10224, http://dx.doi.org/10.1016/S0140-6736(20)30113-5.

[9] S.E. Park, Epidemiology, virology, and clinical features of severe acute respiratory syndrome-coronavirus-2 (SARS-CoV-2; Coronavirus Disease-19 outbreak: an observational and modelling study, Nature 585 (2020) 1168–1173, http://dx.doi.org/10.1038/s41586-020-2727-4.

[10] C.W. Lu, D.Anguelov, D.Erhan, C.Szegedy, S.Reed, C.Y.Fu, et al., SSD:Single Shot Multibox Detector, in: European conference on computer vision, 2016, pp. 21–36.

[11] S.H.Ebrahim, Q.A.Ahmed, E.Gozzer, P.Schlagenhauf, Z.A.Memish, Covid-19 transmission with person detection and tracking via fine-tuned YOLO v3 and SiamRPN++, Proc. IEEE Int. Conf. Comput. Vis. 2020, pp. 22–26.

[12] R.M. Elavarasan, R. Khandelwal, S. Agarwal, Using computer vision to enhance safety of workforce in manufacturing in a post COVID world, 2020, arXiv preprint arXiv:2005.05287.

[13] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE PAMI 39 (2016) 1137–1149.

[14] W. Feng, L. Wang, P. Ren, Timer-YOLO: A real-time object detection method for constrained environments, IEEE Access 8 (2019) 1935–1944.

[15] D. Alasmary, M. Fachurrozi, Faster R-CNN with inception v2 for fingertip detection in homogenous background image, J. Phys. Conf. Ser. 1196 (2020) 012017.

[16] H. Wei, N. Kehtarnavaz, Semi-supervised faster RCNN-based person detection and load classification for far field video surveillance, MAKE 1 (2019) 756–767. http://dx.doi.org/10.1039/make1030044.

[17] R.T. Collins, P. Carr, Hybrid stochastic/deterministic optimization for tracking sports players and pedestrians, in: European Conference on Computer Vision, 2014, pp. 298–313.

[18] V.S.K.Parma, S. Adarsh, K.I. Ramachandran, B.B. Nair, Real time detection of speed hump/bump and distance estimation with deep learning using GPU and ZED stereo camera, Procedia Comput. Sci. 143 (2018) 988–997. http://dx.doi.org/10.1016/j.procs.2018.10.335.

[19] H. Singh, Real-time use cases, in: Practical Machine Learning and Image Intelligence approaches, Neural Comput. Appl. 31 (12) (2019) 8661–8680.

[20] Yashas, Summary of the CUDA backend in OpenCV DNN. https://gist.github.com/YashasSamaga/985071dc57885348bec072b4dc23824f, (Accessed February 2021).

[21] P. Bhattacharya, A. Srivastava, A. Khandelwal, S. Agarwal, Using computer vision to enhance safety of workforce in manufacturing in a post COVID world, 2020, arXiv preprint arXiv:2007.03578.

[22] M. Ahmed, A. Sharifi, S. Khalili, Presentation of a developed sub-epidemic model for estimation of the COVID-19 pandemic and assessment of travel-related risks in Iran, Environ. Sci. Pollut. Res. Int. 28 (12) (2021) 14521–14529.

[23] M. Ahmed, M.Q.H. Abadi, A review of using object-orientation properties of C++ for designing expert system in strategic planning, Comput. Sci. Rev. 37 (2020) 100282.

[24] M. Ahmed, S. Jafardzadeh, S. R. Taghibadeh, S. Sharifi, Presentation of a new hybrid approach for forecasting economic growth using artificial intelligence approaches, Neural Comput. App. 31 (12) (2019) 8661–8680.

[25] M. Ahmed, A computational approach to uncovering economic growth factors, Comput. Econ. (2020) 1–26.