International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jun-2022)

http://dx.doi.org/10.12785/ijcds/1101115

Smart Surveillance System to Monitor the Committed Violations During the Pandemic

Muhanad Ramzi Mohammed[1], **Amar Idrees Daood** [2]

[1] *Computer Engineering, Mosul, Iraq*
[2] *Computer Engineering, Mosul, Iraq*

Received 05 Apr. 2021, Revised 04 Apr. 2022, Accepted 13 Apr. 2022, Published 15 Jun. 2022

Abstract: The world now faces a medical crisis which needs to be resolved. COVID-19 is a disease which spreads between people mainly when an infected person is in close contact with another. To decrease the virus spreading, World Health Organization suggests some rules to follow such as wearing masks, social distance, and quarantining the infected people. In this work, we propose a surveillance system to monitor public spots and make sure those infected people don't leave quarantine sites and to make sure that people wear a mask and practice social distance. Our proposed Research is designed to detect, track a moving person in video sequence and help to specify his profile when he is entering or leaving a special region. The system detects if they are wearing a mask or not, respecting social distance or not. Additionally, the proposed system informs the persons concerned in the field when the infected people enter in the monitored area or when they are found unmasked by performing a face recognition. The proposed system is used for tracking people in real time. It extracts frames from the video sequence and it performs the detection process using a deep learning pre-trained model and tracks people so we can analyze their behaviour and create profiles of the moving people in the area. We believe that the proposed system can minimize the jeopardy of the pandemic and it is definitely the optimal solution to contain the risk of the virus spreading, especially when manual human monitoring is almost impossible to cover the entire globe. Furthermore, we use transfer learning techniques to train a deep learning model for masked-unmasked face classification. The main novelty in the proposed system is threefold: (1) Adopting an efficient face detector to detect masked faces. (2) Synthesizing masked-face dataset to train masked-unmasked face classifier to decide whether people are wearing masks or not. (3) Adjusting the recognition algorithm to recognize the masked faces. The combination of these aspects gives excellent results. The experimental results show that the proposed system can perform very well in the real time execution.

Keywords: Surveillance system, healthcare, Object detection, face detection

1. Introduction

Video surveillance is an important research area in many disciplines. Many architectures of video surveillance systems are used increasingly in security systems to fulfil different purposes. Generally, with a video surveillance system, we are in the picture of 24 hours a day, 7 days a week. In fact, with the rapid development of hardware and software systems, it is economically and technically feasible to build video surveillance system architecture tailored to specific needs. Video surveillance systems are used widely in many applications, sometimes it is called a visual surveillance system because it is based on the visual appearance of the individual frames in the monitored video. Video surveillance involves the act of observing a scene or scenes and looking for specific behaviours that are improper or that may indicate the emergence or existence of improper behaviour [12]. There are many purposes of video surveillance such as, detecting abnormal events, information gathering for analysis, or even for security reasons. Recently, video surveillance system has been vastly investigated and used effectively in a lot of applications such as, transport systems (railway stations, airports, urban and ruler motorway road networks), government agencies (military base camps, prisons, strategic infrastructures, radar centres, laboratories, and hospitals), industrial environments, automated teller machine (ATM), banks, shopping malls,

E-mail: muhanad.enp85@student.uomosul.edu.iq, amar.daood@uomosul.edu.iq

and public buildings, etc. [8]. Technically, these applications can vary from crime solving or terror attack detection to monitoring shoppers inside the market to bill them automatically. Video surveillance requires mainly two phases of processing. In the first phase, a detection process is needed to detect the object of interest (whether the object is our interest and the action is being done by the object). In the second phase, the tracking process is required to complete the whole-time line so we can explain the scenes of the video properly [9].

Over the last few years, due to globalization a major change has occurred in different sectors worldwide such as business, security, health etc. One of the important sectors which are now a worldwide concern is healthcare crisis management. Currently, the entire globe is facing a very dangerous virus which causes the death of thousands of people, this virus is known as COVID-19. In order to combat COVID-19, the World Health Organization (WHO) recommends wearing a face mask when being in crowded places and to keep social distance between people (at least 1 metre). The cases are rapidly increasing due to infected people not following the rules set by WHO, so the best solution was to quarantine infected people and it is now necessary to observe social distancing and face masks in accordance with recommendations provided by the World Health Organization to reduce the incidence of COVID-19. Due to the emergence of COVID-19 Pandemic, providing surveillance systems to track infected people is one of the most important tasks. In order to accomplish this, we propose a video surveillance system. The proposed surveillance system tends to monitor the behaviour, activity or other information generally of people in a specific area to make sure that they are following World Health Organization's recommendations to wear face masks and practice social distancing.

Researchers and scientists proved the importance of following WHO instruction in reducing the incidence of COVID-19 cases. Several researches were conducted in this field. In [13], the authors proposed a model to detect masked and unmasked faces. The model is based on the LLE-CNN convolutional neural network. The proposed system was tested on the (MAFA) dataset. The system achieved an accuracy of 76.1%. In [14], the authors propose a deep learning model for face mask detection. They transferred the previous knowledge of the ResNet50 and used it for the training process to extract features then they used several classifiers to perform the prediction process. The SVM classifier achieved the highest accuracy of 98%. Even Though, they have achieved a high accuracy, their implementation lacked a complete set of WHO rules. In[15], the authors proposed a model to detect social distance and face mask violations. The authors used MobileNetV2 as their base classifiers for both problems. They solved the problem of social distance by transforming the image into a bird's eye view. They stated that the system can perform well only when people are detected clearly. Additionally, they have not recognized the infected people to check the quarantine rule. According to our knowledge there is no work in the previous research which combines all the rules of WHO in one system.

Hence, we propose a smart surveillance system to check all the rules of WHO. The designed system starts processing the real time feed of the camera by capturing each frame individually. First, we detect the presence of the human bodies in the current scene and compute the distance between the detected people to decide whether they are violating the social distance rule or not. The second step in our work is a face detection which is performed on the region of the detected people, and then classify these detected faces into two categories. The classification process is used to decide whether the detected people are wearing masks or not to check if they are following WHO's mask rule. Finally, we need to identify the infected people by performing a face recognition process. The system searches in a stored database which contains a list of infected people to compare with the detected people in the current scene to recognize the infected persons.

The paper is organized into three sections: section 2 describes the proposed system and methodology. section 3 presents the results. Finally, the conclusions and the findings are illustrated in section 4.

2. Methodology

This research designs and implements a health safety management system to measure the threat or risk level of public areas. The proposed work tends to track and monitor people to build a complete profile for each person according to their obedience of the rules set by (WHO) to decrease the Covid-19 virus spreading. Firstly, the designed system tends to process the scenes of the monitored video to detect the presence of people in a specific region. Secondly, the system measures the distance between the detected humans to decide whether they respect and practice the social distance rule or not.

Then, the face detection process is performed to localize the faces in the scene of detected people. After that, we need to identify the detected faces to check if they are masked or unmasked. This process requires a classifier to recognize the masked faces from the unmasked ones. To perform the training process, we collected a dataset which contains masked and unmasked faces. Data gathering of faces is an easy task due to the availability of this kind of data. On the other hand, providing data with masked faces is a complicated task because it is a new gained-habit to wear masks due to the pandemic. Synthetic image is a feasible solution to provide such a dataset, where augmented reality technique is used to wrap the mask around the unmasked faces in the images.

The final stage of the proposed system uses face recognition techniques to perform a searching process against a stored database which contains a list of known people to add the committed violations (in case it happened) to complete their profile. These violations can be categorized into three classes. First violation is disrespecting the social distance while the second type is not wearing the mask. Finally, the last type of violation is disobeying the quarantine of infected people which needs a pre-knowledge that should be available in our database to label people whether they are infected or not. A personal Computer will be used in this project with a surveillance camera to test the algorithms which will be implemented by the python language. There are two important metrics that we should take into consideration when we choose an algorithm to implement, the computation time and the accuracy. The merit of our research is entirely based on the selection of these algorithms experimentally. For example, there are several algorithms and techniques for Face detection and recognition such as: Haar Cascades filter, PCA analysis, Fisher Faces analysis, Local Binary Patterns Histogram features, Histogram of Oriented Gradients features, and the deep learning techniques.

The designed system consists of four parts that process the input frame successively. The first part is the Social Distance detection where social distance violation is detected through measuring the distance between people appearing on the input frame, YOLO v3-Tiny was used in this part. The second part of our system is the face detection, a pre-trained deep learning model is used to detect front and side faces as well as the masked faces. The third part is the face mask detector where we use the detected faces from the previous stage and feed these detections (face locations) to a mask detector model to classify these faces whether they wear masks or not. Finally, we perform the recognizing process by passing the detected face to the LBPH algorithm, where we can identify infected people based on the database which we have. The flowchart in figure 1 illustrates the flow of the proposed system.

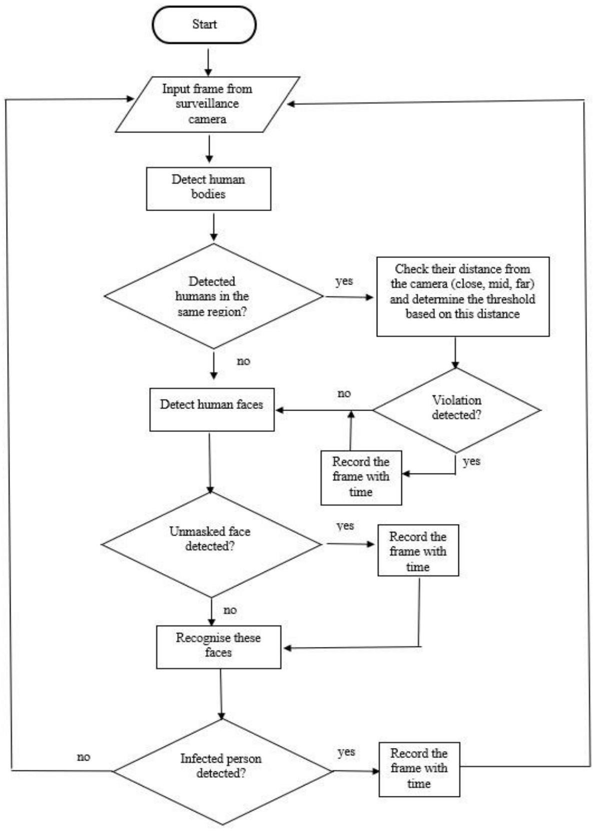

Start
Input frame from surveillance camera
Detect human bodies
Detected humans in the same region?
yes
no
Check their distance from the camera (close, mid, far) and determine the threshold based on this distance
Violation detected?
no
yes
Record the frame with time
Detect human faces
Unmasked face detected?
yes
no
Record the frame with time
Recognise these faces
Infected person detected?
yes
no
Record the frame with time

Figure 1. the system flowchart

2.1. Social Distance

In order to detect social distance violations, we need to perform object detection to detect human bodies (Human Detection). The problem of detecting humans can be simply stated as: given an image or video sequence, localise all subjects that are human [10]. A fast and accurate Human Detection is required because the results of this stage will be the foundation(inputs) of the following stages. Therefore, a deep learning model was used for this task. YOLO v3-tiny [1] was used to detect humans appearing on the surveillance camera. YOLO v3-tiny is a one stage object detector [2], that means it detects and classifies objects in a single step rather than needing to first detect the object and then classify it in another stage. YOLO v3-tiny is a simplified version of YOLO v3 which has a much smaller number of convolution layers than YOLO v3, which means that YOLO v3-tiny does not need a large amount of memory, thus reducing the need for hardware. Hence, it speeds up the detection process, but at the expense of performance [3]. YOLO v3-tiny predicts bounding boxes using dimension clusters as anchor boxes [1]. The network predicts 4 coordinates for each bounding box, x, y, w, h. The x, y represents the upper left corner of the bounding box, while w and h represent the width and height of the bounding box for the detected object.

These coordinates will be used to check whether there is social distance violation or not by measuring the distance between the detected boxes of the human bodies. Assume that the centre coordinates of the box of one person is x1 and y1, and the centre coordinates of the other person is x2 and y2, then the distance between those people can be easily found by measuring the euclidean distance using equation 1:

$$distance = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \qquad (1)$$

Then we compare the distance against a threshold parameter that has been measured experimentally to determine whether these two persons have violated the social distance or not. Adoption of a fixed threshold value may cause a problem of determining the correct number of violations. Therefore, We used Fuzzy Logic rules to overcome this problem. We divide the scene into three regions (Near, Mid, and Far). The system specifies the person's distance from the camera using the width of the bounding box. Then it determines if the person is in the "far", "mid" or "near" region. The main rule is that there is no need to measure the distance between people detected in different regions. On the other hand, the threshold differs based on people's distance from the camera. The threshold will be the highest when people are near to the camera as the distance between the centre of the two bounding boxes appears to be very high. While in reality they are close to each other. We see this in figure 10 and figure 11. However, in figure 2 and figure 3 people are detected in the "mid" region, thus another threshold is used to determine the social distance violations. Let's take an example, figure 2 shows all people in the "mid" region. The coordinates of the person on the right is (x=527,y=94,w=156,h=343) and the bounding box centre is found to be (605,265). While the coordinates of the person in the middle is (x=360,y=54,w=127,h=311) and the bounding box centre is (423,209). By applying equation 1, we find the euclidean distance between the two persons is 190 pixels which is higher than the threshold related to people of the "mid" region, meaning that they are practising the social distance. Algorithm 1 summarizes the steps of the social distance detection component of the system.

Algorithm 1: Social distance detection algorithm.

```
Input:  a frame from a surveillance camera
output: v: number of violations
1-  get the input frame from a surveillance camera, v=0
2-  pass the frame to yolo v3-tiny to detect humans.
3-  save the bounding boxes coordinates in a vector
4-   i = the first bounding box (first person detected)
5-  j= the (i+1)th bounding box
6-  check both persons' distances from the camera
7-  if they are not in the same region
         then goto step 11
8-  determine the threshold based on their distance
9-  compare distance between them with the threshold
10- if violation detected
         then record the time and save the frame,
              v=v+1
11- j=j+1
12- if j <= length of vector
         then goto step 6
13-  i = i+1
14-  if i <=  the length of  vector
         then goto step 5
15- end
```

2.2. Face Detection

Face Detection is a very important component of the designed system. This stage is performed before Face Mask Detection and Face Recognition stages. In this part, we need an efficient face detector because in the time of COVID-19 we expect to see many masked faces and traditional face detectors such as HAAR CASCADE can't detect these masked faces. Partially occluded faces are very difficult to detect using conventional methods. Additionally, we need to localize complicated cases such as side faces and non-frontal faces. Hence, we adopt a pre-trained deep learning model for this task. This model detector is based on the Single Shot Detector (SSD) framework with a ResNet base network. In our work, we used a Cafee version of this network to detect faces in the individual frame. The main advantage of the SSD detectors over the normal two-stage detectors is that the SSD processes the image to extract features, and predict the class of the object all in one stage/step. After we process each frame individually with image normalization, then we feed it to the face detector to determine all the locations of detected faces in that frame. Finally, we need to check the confidence of the model detector for each face to filter out weak detections. Many researchers use HAAR CASCADE classifier to detect faces before proceeding to recognize them. We have found through conducting a massive number of experiments that most conventional detectors can only detect unmasked front faces, they can neither detect masked/unmasked nor side faces. This is the main reason for selecting the pre-trained deep learning face detector in our designed system. After that, we pass these detections to next stages to complete the process of the monitoring, the face mask detector to detect mask/unmasked faces and also to the identification stage to recognize these faces.

2.3. Face mask detection

Face Mask Detection consists of two phases. Firstly, we need to train a face mask detector on a specific dataset, this task is done using Keras and Tensorflow frameworks. Keras is a software library that provides an interface for the Tensorflow framework. Tensorflow can perform many tasks, but the main task is to train deep learning models. Secondly, when the face mask detector is ready, we perform a face detection and then classify each face individually as with_mask or without_mask using the trained classifier.

To prepare the dataset for the mask detector we collected about 1000 face images. Unfortunately, all these faces are not masked. In order to train a binary classifier to differentiate faces into two classes (With_mask, Without_mask), we need to provide masked faces in our dataset. Therefore, we augmented the masks to create synthetic images with masked faces.

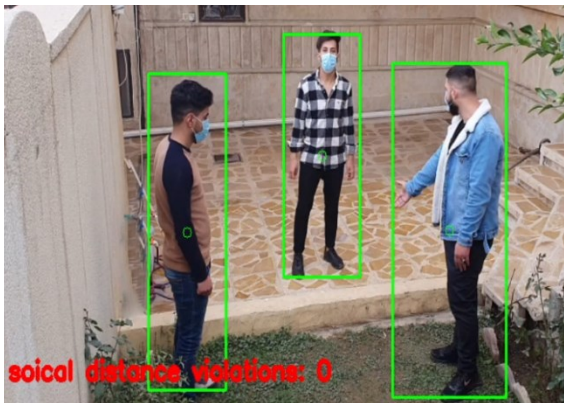


Figure 2. Zero social distance violations are detected

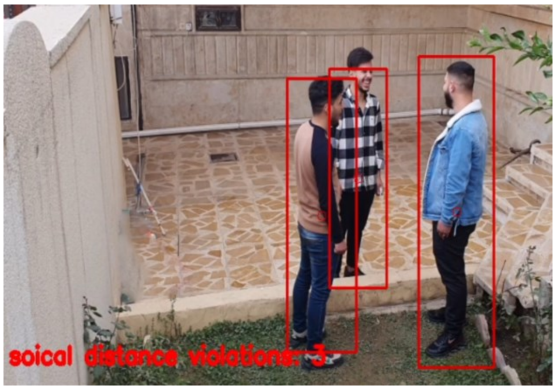


Figure 3. three social distance violations are detected

This is done by using the Facial Landmarks to make it easy to add the mask to the face. To accomplish this, we select 500 images randomly from the collected data. Then, we use a face detector to localize the face. Once we determine where the face is, we apply a facial landmarks detection to localize the landmarks of these faces. We used the method [16] to localize the eyes, nose, mouth, ears, chin, etc. This detector detects 68 key facial features. We used the nose bridge and the chin as main features to add the mask automatically over the face to create the convenient data for the masked faces class. figure 4 shows the results of the landmark facial features. After that, we use 500 images with masks and the other 500 images without masks to train a model for detection. We divide our dataset into two partitions, 800 images for training (400 masked images and 400 unmasked images) and 200 for testing (100 masked images and 100 unmasked images). Then, we use a data Augmentation technique to generate more pictures and increase our dataset to help improve the performance and reduce the effect of the overfitting. Different types of data manipulation are used to synthesize more images such as rotations, scaling (zoom in and zoom out), translation and flipping. In total for the training, we have 700 images with masks, and 700 images without masks.

After data preparation, our goal is to train a deep learning model to detect whether a person is wearing a mask or not. Achieving this particular task requires a fast and accurate model to perform the recognition. Any pre-trained model can deliver a high accuracy rate but we need to select a model that can give us a real time response. Therefore, the MobileNetV2 network is used as our base classifier to perform the training process. We use the transfer learning technique to transfer the previous knowledge of the MobileNetV2 model.

First, we remove the top layer of the MobileNetV2 model and then we flatten the feature of its last layer. After that, we add a fully connected layer with 128 nodes to the model. Additionally, we attach a dropout layer to reduce the effect of the overfitting problem. Finally, we end the model with the output layer with two nodes to perform a binary classification for two classes (with mask and without mask). We start the training process using the prepared dataset by freezing all the top layers except the ones which we add to perform the binary classification. Once the training is done, we can use the trained detector to determine whether the detected faces are masked or not. In order to evaluate the trained detector, we classify the testing data. Figure 5 summarises the steps of this stage. The accuracy of our detector is 98% when it is tested using white and blue face mask. Unfortunately, the detector's performance is not good when a black face mask is used. To check reliability of our detector against different colors of the masks, we collected about 867 images with a blue-masked, white-masked, and black-masked faces. These images were passed to our detector to compute the accuracy. The detector was able to correctly classify 759 frames as "masked face" and the other 108 frames were misclassified of the black-masked faces, that's only 86% accuracy. To boost up the accuracy of our detector, new images with black face-mask were synthesized. We added 25 black-masked faces to our dataset and the model was trained again using the old dataset and the new images. The detector was tested again after the training process. Then we repeat the test to check against black-masked faces. This modification improved the accuracy of predicting to 96%. Figure 6 shows an example of a frame containing 3 colored unmasked/masked faces passed to our detector and classified successfully. Other results of this stage are shown in figure 8 and figure 9.

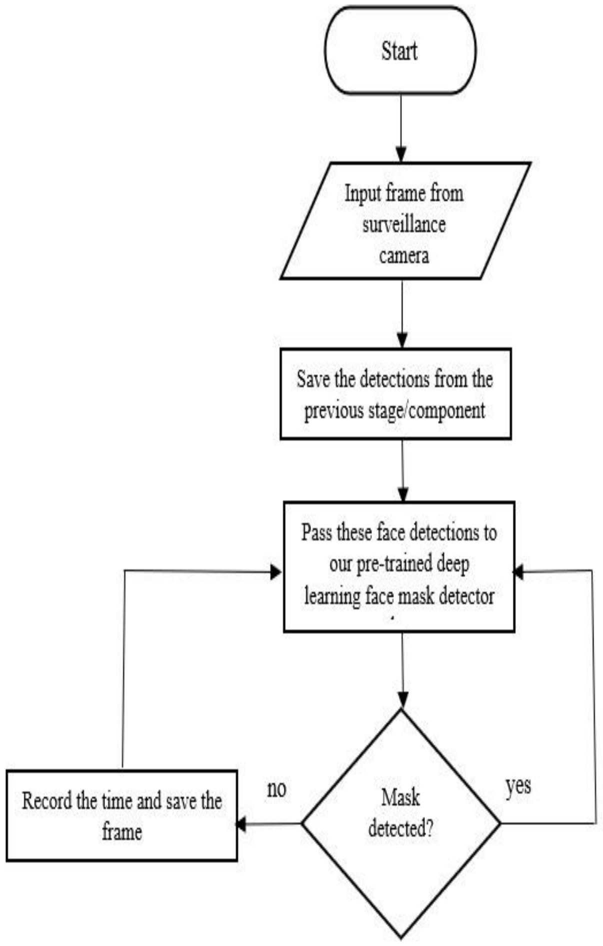

Start
Input frame from surveillance camera
Save the detections from the previous stage/component
Pass these face detections to our pre-trained deep learning face mask detector
Mask detected?
no
yes
Record the time and save the frame

Figure 5. Face mask detector algorithm

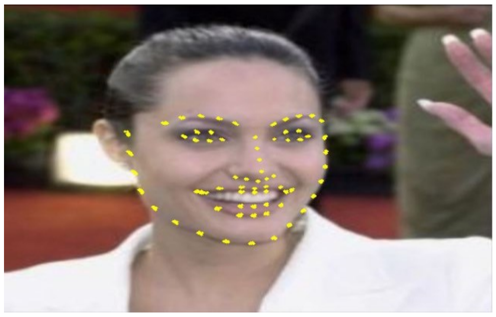


Figure 4. Landmarks facial features (68 points)

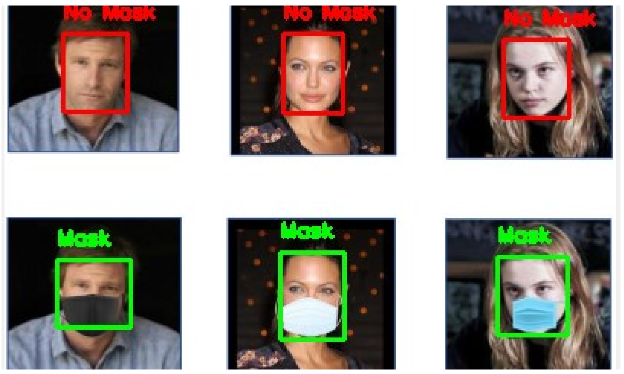


Figure 6. testing the detector using different mask colours

2.4. Face recognition

Now after faces are classified as with_mask or without_mask, we need to recognize faces appearing on the camera and know to whom they belong based on the database we already have. To accomplish this task, we used Local Binary Pattern Histogram (LBPH) Features. LBPH features operates better than other algorithms at the minimum low resolution of 35 pixels, and can identify faces at different angles [4]. Achieving a high accuracy and efficiency for face recognition is a very complicated task. Many algorithms such as Sparse Coding (SC) [5], Local Binary Pattern (LBP) [6], Histogram of Oriented Gradients (HOG) [7] and other techniques provide medium accuracy rate, while LBPH algorithm can recognize front and side faces with higher accuracy rate [4].

The first step in the face recognition process is the data preparation. This task can be considered as a registration phase where we need to collect images for all people that are infected with the Covid-19 virus. In the registration process, we collect a few images with different scenarios for the infected people. Therefore, we capture their faces with different view sides and label them with a unique ID for each individual person. Additionally, we apply a data augmentation technique by placing masks automatically on their faces to increase our dataset. After that we apply the feature extraction process by constructing LBPH for each individual image in our dataset and store these features in the infected people list. By achieving that, we finish the training process and the system is ready to recognize anyone in this list using the nearest neighbour classifier. In the testing stage, LBPH are extracted from the detected faces and then the KNN classifier is used to predict the identity of the tested sample. In the real time execution, the frames are first fed to the face detector which will return face locations, and decide whether they are masked or unmasked, these face locations will be fed to our face recognizer model to return the ID of the person to determine if the person is infected or not. Figure 10 and Figure 11 show how the recognition process is applied on masked and unmasked faces. Figure 7 illustrates the face recognition component.

The three components of the designed system work smoothly one after the other by processing the input frame with reasonable delayed time for a real time execution. The smart selection for the right algorithms and models for each stage in our system makes the performance good enough for a real time execution. For example, yolo v3-tiny is used in the first stage to detect social distance violations which performs fast but at the expense of the accuracy of the detection. Furthermore, the recognition process is performed by applying LBPH features with a simple KNN classifier. The only bottleneck that we have in our system is the Resnet-ssd face detector which is used to detect faces that take time to process the frame. However, it is used due to its ability to detect masked faces as well as side faces. Figure 10 and figure 11 show the system's ability to detect social distance violation as well as mask/unmasked faces and recognise them with infected people being recognized with their unique ID.

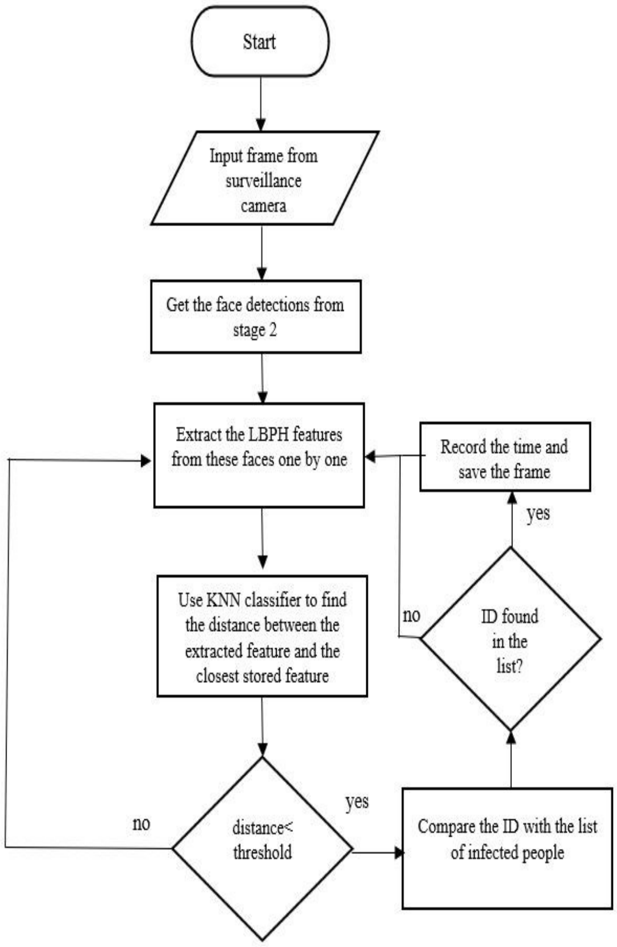

Start
Input frame from surveillance camera
Get the face detections from stage 2
Extract the LBPH features from these faces one by one
Record the time and save the frame
yes
Use KNN classifier to find the distance between the extracted feature and the closest stored feature
no
ID found in the list?
no
distance< threshold
yes
Compare the ID with the list of infected people

Figure 7. Face recognition algorithm

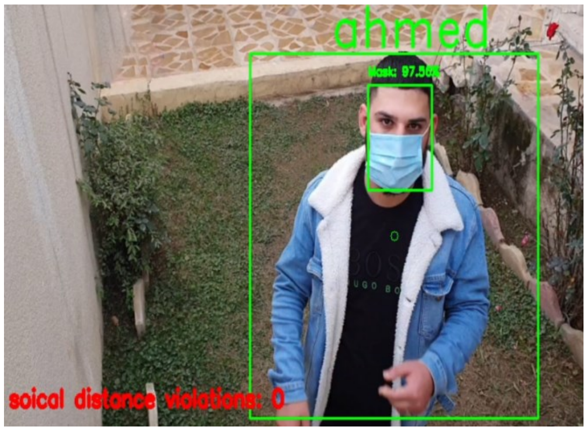


Figure 9. masked face is detected with the identification process

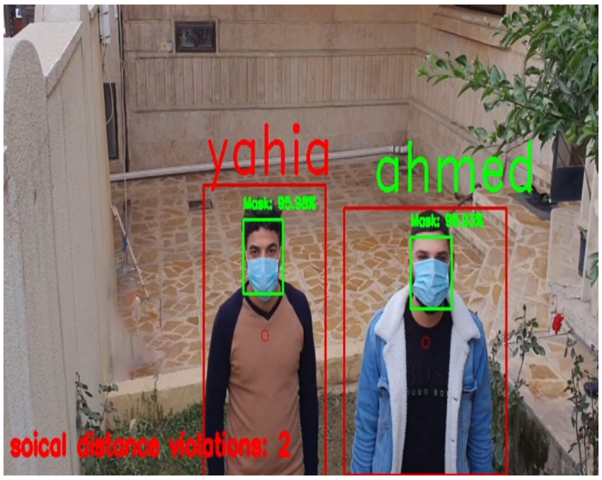


Figure 10. masked faces detected and recognized, and social distance violations are detected.

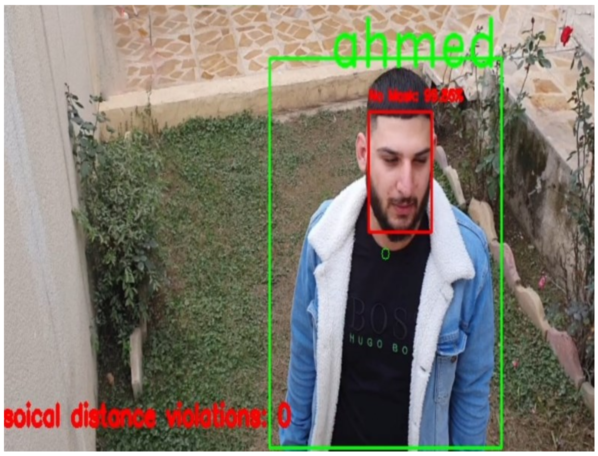


Figure 8. unmasked face is detected with the identification process

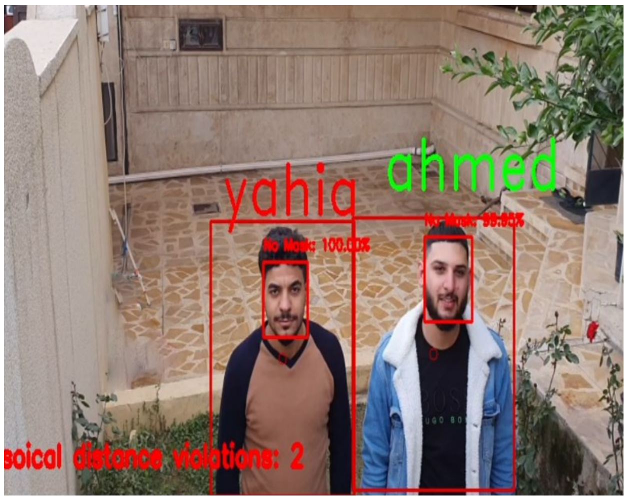


Figure 11. unmasked faces detected and recognized, and social distance violations are detected.

3. Results

The system was tested using some conducted experiments for each component to evaluate the performance and the efficiency. The first step was done by testing each component of the system alone to make sure that they can process the input frame efficiently. Additionally, we increase the load detections to measure the time and the accuracy of each component individually. Labelled Faces In The Wild dataset (LFW) was used to increase the detections load in testing of the face detection, face recognition and the face mask detection. Lastly, the social distancing violation detection was tested using two different scenarios. Two different videos were recorded to achieve this test, the first one is seven minutes long with only 3 people, the second one is 14 minutes with 8 people.

3.1. Face Detection

Face detection component was tested using the Labelled faces in the wild dataset. This dataset contains 13233 images of faces for 1680 people. These faces were passed to the system's pre-trained deep learning face detector, they were all detected successfully and the whole process took 506 seconds to be executed with 0.0434126 second per one face detection with detection rate of 100%. On the other hand, the HAAR CASCADE classifier was tested using the same dataset. It failed to detect 981 faces and the whole process took 405 seconds to be executed with 0.0380065 second per one face detection with detection rate of 92.6%. Same test was repeated on the face detection component, but with the masked faces images. The dataset, which used to perform this test, comprised 700 images of masked faces. The images were passed to the pre-trained deep learning face detector, 17 images were not detected with a detection rate of 97.5%. On the other hand, the same images were used to test the HAAR classifier; it failed to detect 485 images with a detection rate of 30.7%.

Furthermore, we tested the designed system using a stream of frames which were made into a video, each frame has a different number of faces. This experiment is designed to test the system by increasing the load in each individual frame. Additionally, we repeated the same test of load on the HAAR CASCADE classifier. The results of this experiment are shown in Table 1.

Table 1. Load of test is increased to measure the execution time

# Of faces in one frame	execution time with HAAR CASCADE in seconds	execution time with pre-trained deep learning detector in seconds
1	0.0380065	0.0434126
2	0.0385599	0.0401704
3	0.0412968	0.0405717
4	0.0428055	0.0412475
5	0.0445392	0.0419451
6	0.0461054	0.0482149
7	0.0472252	0.0404066
8	0.0476073	0.0401444
9	0.0478005	0.0406794
10	0.0485671	0.0422669
13	0.0603586	0.0444123
15	0.0638809	0.0421526
18	0.0774703	0.0410909
20	0.0822831	0.0413624
25	0.0912612	0.0438564
30	0.1045169	0.0421712

These results show that the HAAR classifier takes less time in detecting one face in any frame. On the other hand, as the number of faces in one frame increases, the execution time gets higher. While the pre-trained deep learning detector gives almost the same time duration of detecting different numbers of faces in one frame. Additionally, its high performance in detecting side/masked faces in comparison with the low performance of the HAAR classifier in detecting them. Figure 12 shows the face detection process using HAAR classifier.

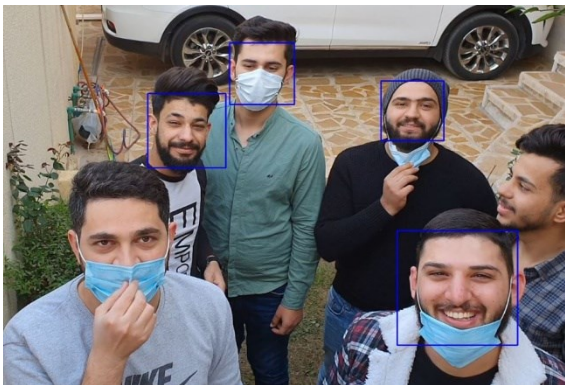


Figure 12. face detection process using HAAR classifier.

3.2. Face Mask Detector

We used the same dataset that was used in the previous section to test the trained mask detector. However, this dataset does not contain masked faces images. Therefore, we augmented some of these images to add the masks by synthesizing extra images with masked faces. The system was able to perfectly detect the mask with no issue to report. We increased the load of the number of faces in each frame by creating a stream of video. We passed the synthetic video to test the mask detector and measure the execution time. The results of this experiment are shown in figure 14. Figure 13 shows some detection results of masked and unmasked faces.

3.3. Face Recognition

Face Recognition component was also tested using the Labelled faces in the wild dataset. As the system needs to recognize masked and unmasked faces, the training must be done with masked faces images. Therefore, all faces were passed an extra component to cover the faces with masks. 14 characters were used to test the face recognition algorithm. Each person with about 20 unmasked images and the same number of synthesized masked images to create a small dataset.

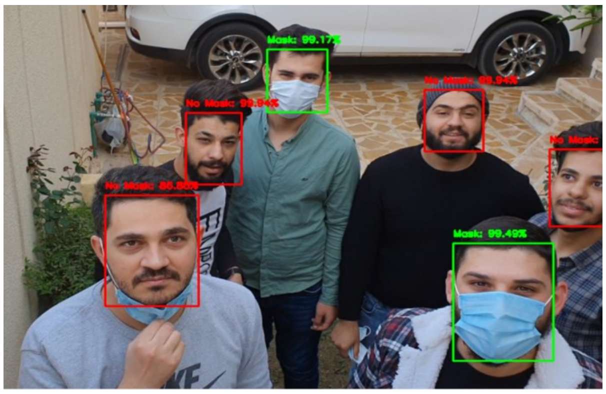


Figure 13. masked and unmasked face detections using pre-trained model.

We divided this dataset into 80% for training and 20% for testing. After we performed the training process, we evaluated the trained model using the testing dataset. The trained model achieved a recognition rate of 91.6%.

Furthermore, we tested the designed system using a stream of frames which was converted into a video, each frame has a different number of faces from the Labelled Faces in the wild dataset (LFW). This experiment was used to test the efficiency by increasing the load of the number of faces. Figure15 shows the execution time for each frame. The results indicate that frames with a higher number of faces take longer to be processed due to the processing time of each individual face in the particular frame. Figure 16 shows some results of the recognition process.

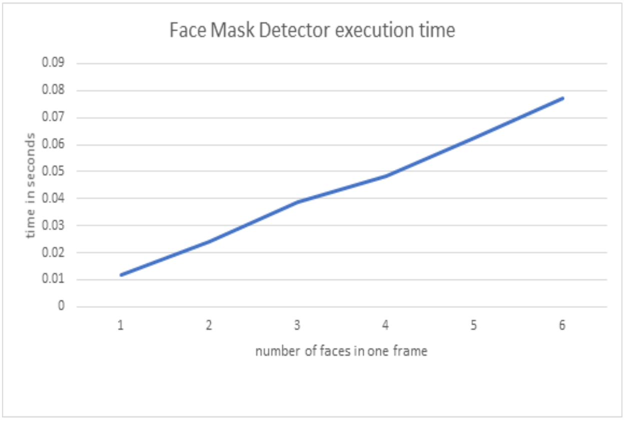


Figure 14. load of test is increased to measure the execution time

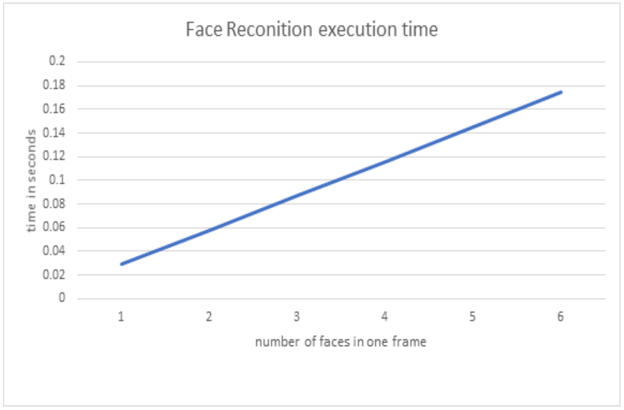
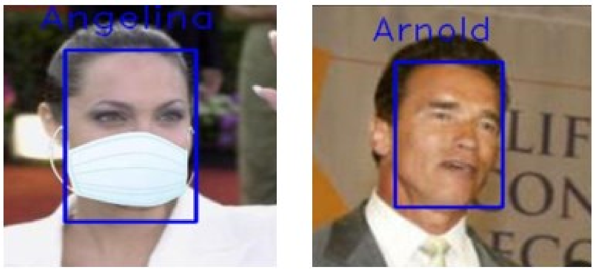


Figure 15. load of test is increased to measure the execution time



(a) (b)

Figure 16. (a): result of the recognition process for a masked face. (b): result of the recognition process for an un-masked face

3.4. Social Distance Detector

In order to test the Social Distance detector, a series of videos of 14 minute were used. Each video was processed and the social distance detection was tested and the results were recorded, the detection rate was 100%. The load of the people's presence was increased in each video from 3 to 8 people. The detection process was done successfully, some of these results are shown in figure 17. Additionally, we measured the time load of this experiment, Figure 18 shows the execution time for processing different frames with different numbers of people.

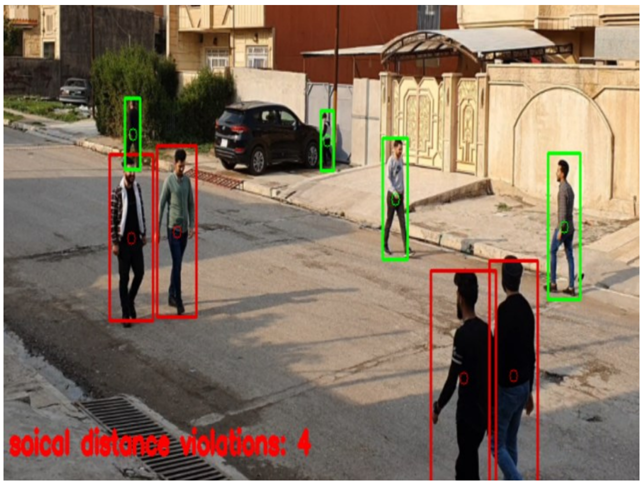


Figure 17. Social Distance detections

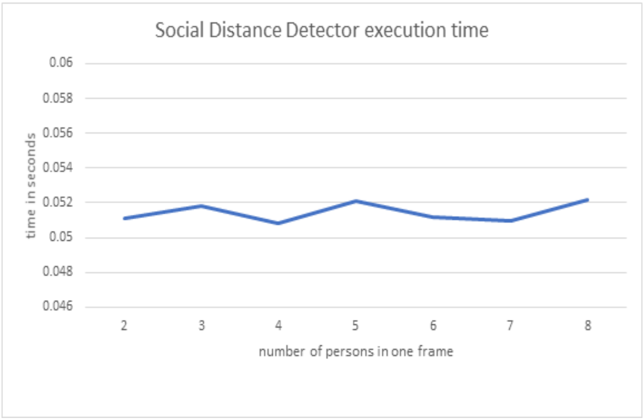


Figure 18. load test is increased to measure the execution time.

4. conclusion

In this paper, we propose a smart surveillance system with an advanced level of tracking and detection that will be useful during the pandemic. As the disease spreads between people, social distancing and wearing masks are mandatory requirements. Hence, violations of one of these rules can be detected through the surveillance system. The proposed system performs a Human Detection algorithm in real-time to test the social distance rule. Furthermore, Face Detection and Recognition algorithms are used to detect and recognize people (based on a previously stored database) to test the violations of these rules set by WHO. Masked face detector is trained using transfer learning techniques to synthesize an efficient model to test the rule of masked-unmasked face. The selected algorithms are chosen carefully to maximize the throughput detections and optimize the system's performance. In this work, we conduct a massive number of experiments to test the designed system. The experimental results show that the system satisfies the design requirements. The system tracks and detects violations and the time of the violations are recorded.

References

[1] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[2] Adarsh, Pranav, Pratibha Rathi, and Manoj Kumar. "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model." 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2020.

[3] Yi, Zhang, Shen Yongliang, and Zhang Jun. "An improved tiny-yolov3 pedestrian detection algorithm." Optik 183 (2019): 17-23.

[4] Ahmed, Aftab, et al. "LBPH based improved face recognition at low resolution." 2018 international conference on Artificial Intelligence and big data (ICAIBD). IEEE, 2018.

[5] Olshausen, Bruno A., and David J. Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." Nature 381.6583 (1996): 607-609.

[6] Chao, Wei-Lun, Jian-Jiun Ding, and Jun-Zuo Liu. "Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection." Signal Processing 117 (2015): 1-10.

[7] Liqiao, J. H. U., and Q. I. U. Runhe. "Face recognition based on adaptive weighted HOG." Computer Enigeering and Applications 53.3 (2017): 164-168.

[8] Gawande, Ujwalla, Kamal Hajari, and Yogesh Golhar. "Pedestrian Detection and Tracking in Video Surveillance System: Issues, Comprehensive Review, and Challenges." Recent Trends in Computational Intelligence (2020).

[9] Mrs. Prajakta Jadhav1, Mrs. Shweta Suryawanshi2, Mr. Devendra Jadhav3, "Automated Video Surveillance", IRJET, 2017.

[10] Nguyen, Duc Thanh, Wanqing Li, and Philip O. Ogunbona. "Human detection from images and videos: A survey." Pattern Recognition 51 (2016): 148-175.

[11] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

[12] Thomas L. Norman CPP, PSP, CSC, in Effective Physical Security (Fifth Edition), 2017

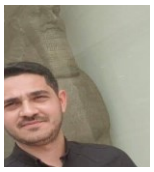
[13] Ge, Shiming, et al. "Detecting masked faces in the wild with lle-cnns." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[14] Loey, Mohamed, et al. "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic." Measurement 167 (2021): 108288.

[15] Khandelwal, Prateek, et al. "Using computer vision to enhance safety of workforce in manufacturing in a post covid world." arXiv preprint arXiv:2005.05287 (2020).

[16] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867-1874.

Dr. Amar Idrees Daood
A faculty member of the Computer Engineering department at Mosul university/Nineveh, Iraq. He received his M.Sc. in computer engineering from Mosul University in 2008. He completed his Ph.D. in 2018 in Computer engineering from Florida institute of technology Florida USA. Dr Daood's current areas of research are Computer Vision, Artificial intelligence, Machine learning, Computer Graphics, Real time system and Parallel Computing.



M uhanad Ramzi Mohammed
My name is Muhanad Ramzi, I graduated from Mosul University in 2018 with a bachelor degree in Computer Engineering.