A hybrid LBP-DCNN based feature extraction method in YOLO: An application for masked face and social distance detection

Ismail Oztel1 · Gozde Yolcu Oztel2 · Devrim Akgun2

Received: 5 November 2021 / Revised: 6 October 2022 / Accepted: 10 October 2022 / Published online: 21 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

COVID-19 is an ongoing pandemic and the WHO recommends at least one-meter social distance, and the use of medical face masks to slow the disease’s transmission. This paper proposes an automated approach for detecting social distance and face masks. Thus, it aims to help the reduction of diseases transferred by respiratory droplets such as COVID-19. For this system, a two-cascaded YOLO is used. The first cascade detects humans in the environment and computes the social distance between them. Then, the second cascade detects human faces with or without a mask. Finally, red bounding boxes encircle the people’s images that did not follow the rules. Also, in this paper, we propose a two-part feature extraction approach used with YOLO. The first part of the proposed feature extraction method extracts general features using the transfer learning approach. The second part extracts better features specific to the current task using the LBP layer and classification layers. The best average precision for the human detection task was obtained as 66% using Resnet50 in YOLO. The best average precision for the mask detection was obtained as 95% using Darknet19+LBP with YOLO. Also, another popular object detection network, Faster R-CNN, have been used for comparison purpose. The proposed system performed better than the literature in human and mask detection tasks.

Keywords Covid-19 · Deep learning · Face mask detection · Human detection

Ismail Oztel
ioztel@sakarya.edu.tr

Gozde Yolcu Oztel
gyolcu@sakarya.edu.tr

Devrim Akgun
dakgun@sakarya.edu.tr

1 Computer Engineering Department, Sakarya University, Sakarya, 54050, Turkey
2 Software Engineering Department, Sakarya University, Sakarya, 54050, Turkey
1 Introduction

The novel Coronavirus Disease (COVID-19) was identified by examining a group of patients who displayed symptoms such as fever, cough, and shortness of breath [60]. The disease first appeared in the Wuhan province, then quickly spread to other countries. COVID-19 spreads with small liquid particles from an infected person’s nose and/or mouth. As a result, the disease spreads when an infected person comes into close contact with another person. Touching the mouth, nose, and eyes with dirty hands after handling surfaces contaminated by the patients’ respiratory particles also causes infections [60].

According to the WHO’s report [60], social interventions are critical in reducing the number of infections and saving lives. Social distancing measures help to slow disease spread by preventing the emergence of new ones. WHO recommends at least one-meter distance between people to avoid disease transmission. The WHO also recommends using a medical face mask during home care and in health care settings to prevent the spread of the disease. Furthermore, many countries around the world recommend using a face mask in public places. Some countries even require it.

Many people ignore social distance and do not wear masks for various reasons, including a lack of education, forgetfulness, carelessness, etc. To follow people who do not adhere to the mask and social distancing rules may be difficult for authorities. Furthermore, quantifying whether the social distancing rule is followed can be deceptive to the human observer. To visually determine these people without a measuring device can be boring and difficult. As a result, there is a need to benefit from an automated non-invasive, objective, and quantitative system for social distance and face mask detection.

This study presents a deep learning-based system for automated social distance and face mask detection. In the study, the input is an environmental image that may include people. These people may have a facial mask or not. Also, there may be or may not be enough social distance between them. The output is a marked image. In the output image, the red bounding boxes show the people who do not obey the mask and/or social distancing rules. Green bounding boxes show masked people, and yellow bounding boxes show people who follow social distancing rules.

If there are any red bounding boxes in the final frame, it provides feedback to the authorities. For this system, we used the 2-cascaded You Only Look Once v2 (YOLO-v2) model. The first cascade in the model detects humans, and the second cascade detects masked or unmasked faces in images/videos. In order to improve the results, we proposed a novel two-part feature extraction approach. In the feature extraction model, the first part extracts general features from pre-trained Darknet19, and the second part generates special features using the Local Binary Pattern (LBP) layer combined with two-dimensional convolutional layers. Also, a 2-cascaded Faster R-CNN was used for the same tasks for comparison purposes. For more comparison, various pre-trained networks were used with 2-cascaded YOLO and 2-cascaded Faster R-CNN models. Results have been reported in the Experimental Results section.

The proposed system allows rapidly detecting people who do not obey the rules. As a result, the system may ensure that more people obey the rules. This can be a significant improvement for disease control.

The contributions of the study are summarized in the four items listed below.

1. We propose a novel two-part feature extraction approach. The first section uses pre-trained Darknet19 to extract general features. The second section uses an LBP layer and
2-D convolutional layers to extract better features specific to the current classification problem.

2. We presented a system that can aid in the reduction of the COVID-19 spread rate. When the virus epidemic is uncontrollable, a significant social contribution can be made by implementing this system in public places such as airports, markets, schools, bus stops, etc.

3. Face detection can be used to detect the faces in the front profile. On the other hand, face detection is not possible for faces in the back and back-side profiles. In this case, social distance calculation using faces bounding box points becomes impossible. To avoid this situation, we do not compute social distances based on detected faces. The system first detects humans. Then it calculates social distance using humans’ bounding box coordinates.

4. The system works successfully in indoor and outdoor environments, even in occlusion.

2 Related works

2.1 Human detection

The human detection systems include three major steps [34]: (1) extraction of candidate areas that may contain human objects, (2) human object description, (3) classification of these areas as human vs. non-human, and post-processing. Enclosing each human object in a box is a common method for extracting candidate areas. The candidate areas extract limits of the searching area for human objects. Thus, it improves the performance [34]. The feature extraction methods for human detection can be categorized into three groups: shape-based, appearance-based, and motion-based approaches. In shape-based approaches, edge-based features are used for the human object description. In appearance-based techniques, texture and color information are used for feature extraction. Motion tracking can help distinguish one object from another if motion forms are different. After the human features extraction step, the candidate regions are classified as human vs. non-human. Post-processing can be used to merge the positive areas. For human detection task, many researchers used hybrid methods such as Histogram of Oriented Gradients (HOG) and Support Vector Machines (SVM) [7, 8, 15, 32]; the differential gradient and statistical Tamura features. The authors of [65] used the HOG and Adaboost algorithms. Kinect Sensor depth information was used in [52, 61]. Furthermore, multi-sensors were used for this task in [1, 57]. The HOG feature extractor was used by the authors of [24]. The authors then used a dictionary to represent the extracted features, including positive, negative, and trivial bases. Finally, they detected the object by employing the proposed likelihood measure derived from the distribution of the sparse coefficients.

2.2 Mask detection

Owing to the recent technological developments, face analysis studies have been very popular. 2D and 3D face recognition using machine learning and deep learning methods [49, 50], 3D Face Reconstruction in Deep Learning [51], Deep-rooted learning based micro-facial expression [25], etc. have been studied in computer science literature recently.

Previously, face mask research concentrated on face recognition under occlusion caused by face masks [17, 29]. Following the emphasis on the importance of wearing a mask in preventing the spread of COVID-19 [26, 59], studies on masked face recognition and masked
detection have begun to appear in the literature. The authors of [28] combined deep transfer learning methods with machine learning algorithms for face mask detection. They distinguished between masked and unmasked face images in a database. In addition, the authors of [33] classified face images as mask vs. no-mask. Unlike the previous studies, our proposed model uses face detection with mask/no-mask rather than classification. Therefore, masked and unmasked faces can be detected and marked in photos, videos, and natural environments, including multiple face images. The authors of [42] created a new method for detecting the facemask-wearing condition. They used a face detector before the mask detection step. Different from that study, we trained the proposed system to detect masked and unmasked faces without using an additional face detection step. The authors of [27] created a mask detector using Resnet50 and YOLO-v2. They merged two databases and removed low-quality images and redundancy.

In addition to the differences in mask detection mentioned above, our study determines the social distance in the same system.

2.3 Social distance detection

For reducing the impact of the COVID-19 pandemic, one of the suggestions is social distance conservation. Scientists have been conducted research to support this rule in the literature. The authors of [18] used YOLO for pedestrian detection. They calculated the social distance using the Euclidean distance. In their studies, no quantitative result information was found. The authors of [12] used bird's-eye view images, a Gaussian Mixture Model, and Kalman Filters to detect social distance. In a laboratory setting, their calculation results have an error rate of 1.69%, 2.17%, and 3.45%. The authors of [54] proposed a social distance surveillance system based on YOLO. They conducted their research in a simulation environment, and the result was reported as 90% accurate. Some studies, such as [47], used smartphones, Bluetooth, and/or GPS to calculate safe social distance. Also, social distancing impact on COVID-19 [21], factors affecting social distance [14], etc. have been investigated in the literature.

2.4 COVID-19 studies

Recently, many studies have been presented to help reduce the spread rate of COVID-19 or aid in disease diagnosis. Furthermore, some scientific groups, such as [3] encouraged scientists. The authors of [11, 36] worked on automatic segmentation of COVID-19 lung infected region. The authors of [58] presented a method for classifying COVID-19 based on its appearance on chest X-rays using neural networks and texture features. In [31], the authors presented a multiple ensemble neural network model with fuzzy response aggregation for the COVID-19 time series. Ensemble neural networks are made up of a collection of neural network modules. Under various conditions, these modules generated some predictions. Then, the predictor module responses were aggregated using Fuzzy logic. The authors of [56] collected data from Heilongjiang and trained an ordinary differential equation model to fit the data. They extended the simulation using this trained model to characterize the effect of an imported ‘escaper.’ They suggested that an imported ‘escaper caused the newly confirmed COVID-19 infections in Heilongjiang province.’ The authors of [4] described a hybrid method for forecasting COVID-19 time series using fuzzy logic and fractal theory. The authors of [5] presented a hybrid intelligent method for country classification based on fractal theoretical concepts and fuzzy logic mathematical constructs. The authors of [2] worked on a social distance and mask detection system. They collected information from
two datasets. The created database is not publicly available for comparison. Furthermore, the authors studied social distance and mask detection in [30, 48, 62], but the datasets used in these studies are unavailable. As a result, we only cited studies where the results were obtained using a publicly available dataset.

3 Proposed system

This study presents a system that focuses on face mask and social distance detection to help reduce the spread of COVID-19. Redmon et al. [44] reports that state-of-the-art object detection networks use region proposal algorithms to guess the location of objects. Because of its performance, YOLO is a popular model for object detection. We choose this model due to mentioned motivations. Also, we preferred another popular model, Faster R-CNN, for comparison purposes.

The proposed system detects people in the environmental images, then calculates the physical distances between them. The people who obey social distance rules are encapsulated with yellow bounding boxes. Red bounding boxes encircle people who are unconcerned about social distance. Following that, the system detects human faces with or without a mask. Masked human faces are surrounded by green bounding boxes, while red bounding boxes surround unmasked human faces. The system includes 2-cascade YOLO. In the first cascade, YOLO is used with the Resnet50 network for human detection. In the second cascade, a 2-part novel feature extraction method is integrated into YOLO for the mask detection task. For comparison, the same system has also been developed using 2-cascaded Faster R-CNN. The system flow is illustrated in Fig. 1.

One of the study’s strengths is the use of human detection to determine social distance. Face detection can be used to detect people whose faces are visible in the front profile, and face bounding box coordinates can be used to calculate social distance. On the other hand, face detection is not possible for people in the back and back-side profiles. As a result, we used human detection to determine social distance. Also, the study has the advantage
of proposing a novel feature extraction model. The proposed feature extraction model is
detailed below.

3.1 Proposed feature extraction model

In the literature, various feature extraction methods such as [35, 41, 64, 66] have been used.
In this study a new two-stage feature extraction approach has been presented. The proposed
feature extraction model is divided into two parts, as illustrated with the baseline model in
Fig. 2. The first is the transfer learning component, in which we used a pre-trained network
to extract general features. We choose Darknet19 for this purpose due to its success in other
networks we tested. Darknet19, like many other transfer learning models, is composed of
various sequential convolutional layers prior to classification layers, with some of the layers
closest to the classification layer discarded. We used various architectures to extract better
features specific to the current classification problem following the pre-trained network. In
addition to baseline implementation, we defined a model containing an attention layer in
the comparisons. The attention layer has been included in the feature extraction model’s
custom part, as shown in Fig. 3. Our final model contains an LBP layer combined with
convolutional layers to build a new network to improve training success. The output of LBP
is concatenated with the previous features through convolutions, as shown in Fig. 4. We
gave the details of the LBP-based feature extraction approach in the following section.

3.1.1 Local binary pattern

LBP is an efficient texture operator, and this feature makes it feasible for feature extrac-
tion in machine learning applications [40]. It has a lightweight algorithm that works based
on comparison with neighbors. It is described by (1), where $R$ defines the radius of the
operation area, and $N$ defines the number of pixels within the neighborhood.

$$lbp_{N,R}(i, j) = \sum_{n=0}^{N-1} f(p_n, p_c)2^n$$

$$f(p_n, p_c) = \begin{cases} 1 & p_c < p_n \\ 0 & otherwise \end{cases}$$

(1)
The selected neighbor pixel and the center pixel are defined by $p_c$ and $p_n$, respectively. The comparison of $p_c$ and $p_n$ is defined with the $f(.)$ function, which returns 0 or 1 according to the comparison result. According to the index of neighbor, which is defined with $n$, the comparison result is multiplied by $2^n$. The resulting values are summed together once all the comparisons are made to obtain transformed pixels. The equation is applied to all the pixels of the image to complete the LBP transform. In this study, we utilized LBP operation as a feature extraction layer in the deep learning architecture. For this purpose, we defined the custom layer definition using the most common application of LBP, where $N = 9$.

We implemented the layer using custom layer definitions in Matlab. For this purpose, a class inherits the `nnet.layer.Layer` needs to be defined. Some functions can be implemented within this class, such as predict, forward, and backward. The predict and forward functions take input features through the layer at training and prediction time to produce the output. The backward function has not been implemented because the LBP layer involves no trainable parameters. We used (1) which defines the LBP transform to implement the predict and forward functions based on matrix definitions.

### 3.1.2 Pre-trained networks in 2-cascaded YOLO and 2-cascaded faster R-CNN

Pre-trained networks learned to extract robust features. Thus, applying them to a new task can provide advantages over training a network from scratch. It enables more efficient and faster learning with fewer data points [38]. In the proposed study, various pre-trained networks have been adapted in a 2-cascaded YOLO system, and their results have been compared. For the mask detection cascade, Resnet18, Resnet50, Resnet101 [16], Shufflenet [63], Darknet19, Darknet53 [43], and Xception [6] have been used separately. For human detection cascade, Resnet50 have been used. For comparison, the same networks have also
Table 1 Details of the pre-trained networks used as feature extractor

| network name | size (MB) | input size |
|--------------|-----------|------------|
| darknet19    | 78        | 256x256    |
| darknet53    | 155       | 256x256    |
| resnet18     | 44        | 224x224    |
| resnet50     | 96        | 224x224    |
| resnet101    | 167       | 224x224    |
| shufflenet   | 5.4       | 224x224    |
| xception     | 85        | 299x299    |

been adapted and applied in Faster R-CNN. The pre-trained networks used in this study and their details are given in Table 1.

3.2 YOLO for human and mask detection

We used the 2-cascaded YOLO-v2 model in the proposed system. YOLO model [44, 45] detects objects using a single-stage object detection network. The algorithm uses a CNN on an input image for network predictions. An object detector decodes the predictions. Thus, bounding boxes are produced. The anchor boxes are used to detect classes of objects in an image. The YOLO predicts Intersection over Union (IoU), anchor box offsets, and class probability attributes for each anchor box. With IoU, it predicts the objectness score of each anchor box. With anchor box offsets, it refines the anchor box position. Class probability is the prediction of the class label assigned to each anchor box.

Owing to transfer learning, a pre-trained CNN can be used as the feature extractor in a YOLO network. The pipeline of YOLO is illustrated in Fig. 5.

YOLO works on full images for making predictions. Seeing the larger context is an advantage for correct prediction. It learns generalizable representations of objects. In YOLO, object detection is considered a single regression problem. First, it divides the input image into $S \times S$ grids. Each grid is responsible for finding out whether the object is in the field. Thus, YOLO creates a separate prediction vector for each grid. In this study, different feature extractor networks have been used for YOLO, and results have been compared.

3.3 Faster R-CNN for human and mask detection

2-cascaded Faster R-CNN is developed for comparison purposes in this study. Faster R-CNN is a state-of-the-art region proposal-based object detection algorithm. It is structured with a pre-trained network for feature extraction and two sub-networks. The first one is a
region proposal network (RPN) which is presented in [46]. An RPN uses an image as the input and produces rectangular object proposals. Each proposal has an objectness score. This process is modeled using a fully convolutional network. A small network slides over the feature maps by the last shared convolutional layer for obtaining region proposals. The small network uses $n \times n$ spatial window of the input convolutional feature map as the input. Multiple region proposals are predicted simultaneously at each sliding window location. The Faster R-CNN pipeline is given in Fig. 6.

In the proposed 2-cascaded system, the maximum possible proposal number for each location has been empirically selected as 3. Thus, the regression layer produces 12 outputs for the coordinates of 3 boxes. Also, the classification layer produces six score values for each proposal for the probability of object or non-object. The proposal number is based on three reference boxes named anchors.

In the training phase of RPNs, each anchor is assigned a binary class label indicating whether an object or not. If the anchors with the highest IoU overlap with a ground-truth box, it is assigned as a positive class. If an anchor with an IoU overlaps higher than 0.7 with any ground-truth box, it is assigned as a positive class [46].

In Faster R-CNN, the ultimate goal is sharing computation with a Fast R-CNN object detection network. Thus, both networks are assumed to share common convolutional layers. The loss function of the Faster R-CNN is shown in (2) [46].

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p^*_i) + \frac{1}{N_{reg}} \sum_i p^*_i L_{reg}(t_i, t^*_i)$$ (2)

In the equation; $i$ is the anchor index number, $p_i$ is the predicted probability of the anchor, $p^*_i$ is 1 or 0 according to the if the anchor is positive or negative, respectively. $t_i$ represents the predicted bounding box coordinates and $t^*_i$ represents the ground-truth box. $L_{cls}$ is the loss of the classification. $L_{reg}(t_i, t^*_i)$ is the loss of the regression. The regression loss can be active with just positive anchors, and the situation is performed using $p^*_i L_{reg}$ in the formula. $N_{cls}$ and $N_{reg}$ are are used to normalize the two terms in the formula with a balancing parameter $\lambda$.

In the Faster R-CNN structure, the second sub-network predicts the actual class label of each object proposal. The advantage of the RPN module is telling the second network where to search for the object.
3.4 Social distance calculation

The bounding box coordinates obtained from the human detection step are used for the social distance calculation. Using a constant value as a safe distance can not be reliable in the videos/images. Because the distance between two people far from the camera may be perceived as less than it actually is. Thus, we used ratio calculation instead of distance while considering the safe distance. In the proposed system, if the Euclidean distance between two people is greater than twice the width of one person’s bounding box, this is a safe distance.

Suppose that bounding box corner points of two humans are \((x_1, y_1)\), \((x_2, y_2)\), \((x_3, y_3)\) and \((x_4, y_4)\) that shown in the Fig. 7.

First, the \((\text{mid}_1)\) and \((\text{mid}_2)\) midpoints of these points for each person are calculated using (3) and (4).

\[
\text{mid}_1 = \left(\frac{(x_1 + x_2)}{2}, y_1\right) = \left(\frac{(x_1 + x_2)}{2}, y_2\right) \tag{3}
\]

\[
\text{mid}_2 = \left(\frac{(x_3 + x_4)}{2}, y_3\right) = \left(\frac{(x_3 + x_4)}{2}, y_4\right) \tag{4}
\]

Then, Euclidean distance is calculated for the midpoints (5).

\[
|\text{mid}_1, \text{mid}_2| = \sqrt{\left(x_{\text{mid}_1} - x_{\text{mid}_2}\right)^2 + \left(y_{\text{mid}_1} - y_{\text{mid}_2}\right)^2} \tag{5}
\]

Finally, using (6), it is compared whether the calculated distance is a safe distance or not. According to the computation results, the social distance is relatively determined.

\[
|\text{mid}_1, \text{mid}_2| > 2 \times |x_2 - x_1| \tag{6}
\]

4 Experimental results

The system was developed with Matlab 2020a. The computer features used in training and testing are as follows: Intel Core i7 Processor (2.6 GHz), 32 GB of RAM, Nvidia GTX 970M (6 GB), and Windows 10 system.
Performances of detection systems are usually evaluated using the Average Precision (AP) criteria index, which is defined in (7) [39]. AP uses precision (shown in (8)) and recall (shown in (9)) values in order to evaluate the detector. Also, the log-average miss rate metric is used to evaluate object detection studies. Therefore, these metrics were used in the evaluation of this study. Log-average miss rates are calculated as a mean of nine False Positives Per Image (FPPI) miss rates in the range of $10^{-2}$ to $10^{0}$ to give a stable performance.

$$AP = \sum \frac{\text{precision}}{\text{recall}}$$ (7)

$$\text{Precision} = \frac{TP}{(TP + FP)}$$ (8)

$$\text{Recall} = \frac{TP}{(TP + FN)}$$ (9)

where $TP$ is true positive, $FP$ is false positive and $TN$ is false negative.

### 4.1 Datasets and data preparation

Each module of the system was trained and tested using different datasets because of the different tasks. The human detector was trained and tested using Pascal VOC Dataset [9] and the mask detector was trained and tested using Face Mask Detection Dataset (FMD) [23]. Both datasets have XML annotations. To obtain the Matlab format of the labels, we used xml2struct function [10]. Also, data augmentation was applied to both modules to improve network accuracy. For this purpose, we flipped the input images randomly during the training process. Table 2 shows the sample numbers of the Pascal VOC and Face Mask Detection datasets. In the human detection dataset, faces are not marked as masked or unmasked. Also, in this database, there is not any masked face image. Similarly, in the mask detection dataset, humans are not marked. Therefore, mask detection and human detection work separately in the proposed system.

#### 4.1.1 Pascal VOC dataset

The last version of the Pascal VOC dataset was used for the human detection task. The dataset has four main labels: person, vehicle, animal, and indoor. For the human detection task, the person labels with corresponding coordinate information were selected. The training set has 9000, the validation set has 583, and the testing set has 5138 images.

#### 4.1.2 Face mask detection dataset

Face Mask Detection Dataset was taken from Kaggle. This dataset allows researchers to label a person in a frame with “face with mask” or “face without mask”. The dataset has 853 images in total. 767 of them have been used for training, and the rest have been used for the testing step.

| Dataset                          | Label          | Sample number in training set | Sample number in testing set |
|----------------------------------|----------------|-------------------------------|-----------------------------|
| Pascal VOC                       | person         | 16647                         | 7326                        |
|                                  | with_mask      | 3112                          | 243                         |
| Face Mask Detection (FMD)        | without_mask   | 665                           | 52                          |
4.2 Experiments on human detection

Resnet50 pre-trained network was used to extract the features for the human detection stage. In the author’s previous work [37], a Faster R-CNN based system was developed for the human detection task. In this study, that method has been extended for social distance detection and combined with mask detection using a 2-cascaded Faster R-CNN system. Also, a YOLO-based human detection system was developed, and the results were compared in this paper.

In the training stage, the learning rate is 0.005, the minibatch size is 2, the momentum is 0.9. The training data has been shuffled before each training epoch.

The human detection subsystem has been compared to approaches that used the same Pascal VOC dataset. Table 3 shows comparison results. Both Faster R-CNN and YOLO using Resnet50 overperformed compared to other approaches. Human detection with YOLO using Resnet50 has the best AP score and the human detection subsystem produced promising results. Also, the approximate detection time for each test image has been evaluated and the results have been reported in the table. As can be seen in the table, YOLO is faster than Faster R-CNN and some other detectors for the human detection task.

4.3 Experiments on mask detection

Some popular pre-trained networks were used with the YOLO and Faster R-CNN for the mask detection task. In the experiments, the following parameters were used. The momentum=0.9, mini-batch size=8, the learning rate drop factor=0.1. The training data was shuffled before each training epoch. In order to obtain better training success, different learning rates were used for the pre-trained networks. The learning rate 0.0001 was used for

| Study                      | Feature extractor | Detector              | AP (%) | Time (per frame) |
|----------------------------|-------------------|-----------------------|--------|------------------|
| Jiang and Ma [22]          | HOG III Feature   | Grammar model         | 52.3   | 2s               |
| Jiang and Ma [22]          | HOG Feature       | Grammar + Poselet models | 52.3  | 12s              |
| Jiang and Ma [22]          | HOG III Feature   | Grammar + Poselet models | 55.5  | -                |
| Htet Lin [19]              | Fusion of G and T Feature by JH | Grammar model | 46.8  | -                |
| Htet Lin [19]              | Fusion of G and T Feature by JH | Poselet model | 49.6  | -                |
| Htet Lin [19]              | Fusion of G and T Feature by JH | Smart model | 55.3  | -                |
| Sumit et al. [55]          | fire module +dropout | Tiny YOLO            | 55.7   | -                |
| Sumit et al. [55]          | fire module +dropout+ residual network | ReSTinet | 63.8  | -                |
| Proposed [37]              | Resnet50          | Faster R-CNN          | 65.0   | 0.573s           |
| Proposed                  | Resnet50          | YOLO                  | 66.0   | 0.023s           |
Table 4  Comparison of the mask detection performances based on different feature extraction networks using Faster R-CNN and YOLO without the proposed LBP feature extraction model

| Network   | Depth | Parameters (millions) | Average precision |           |           |
|-----------|-------|-----------------------|-------------------|-----------|-----------|
|           |       |                       | Faster R-CNN      | YOLO      |           |
|           |       |                       | Withmask          | Withoutmask | Withmask | Withoutmask |
| resnet18  | 18    | 11.7                  | 0.68              | 0.40      | 0.85      | 0.68       |
| resnet50  | 50    | 25.6                  | 0.73              | 0.73      | 0.92      | 0.90       |
| resnet101 | 101   | 44.6                  | 0.74              | 0.48      | 0.82      | 0.80       |
| xception  | 71    | 22.9                  | 0.69              | 0.55      | 0.88      | 0.62       |
| shufflenet| 50    | 1.4                   | 0.64              | 0.28      | 0.69      | 0.51       |
| darknet19 | 19    | 20.8                  | 0.78              | 0.70      | 0.91      | 0.73       |
| darknet53 | 53    | 41.6                  | 0.71              | 0.30      | 0.91      | 0.72       |

Darknet19, Darknet53; 0.001 was used for Resnet18, Resnet101, Shufflenet, and 0.005 was used for Resnet50 and Xception. The stochastic gradient descent with momentum (SGDM) optimizer were used for the networks.

Table 4 shows the comparative results of the pre-trained networks for the mask detection task. These pre-trained networks were used in YOLO and Faster RCNN models. As seen in the table, the best performance was obtained as 92.0% for masked face detection using Resnet50 with YOLO. Also, Resnet50 and Darknet19 networks showed better performance than other pre-trained networks. According to the experimental results, usually, masked faces were detected with higher performance than unmasked faces. The difference is mainly due to the number of masked samples being higher than the unmasked ones. In the experiments, YOLO produced better results compared to the Faster R-CNN.

Also, a novel 2-part feature extraction approach has been integrated into the system to improve detection accuracy. The first part of this approach extracts general features using the Darknet19 pre-trained network, and the second part extracts more specific features. The highest performance was obtained using the proposed 2-part feature extraction approach in YOLO as 95% AP for both with mask and without mask face detection. The Precision-Recall curves for the networks of baseline model, model with attention layer, and model with LBP layer are shown in Fig. 8.

Fig. 8  Precision-Recall curves of the a) Baseline model, b) Model with attention layer and, c) Model with LBP layer
Figure 9 illustrates some outputs of the feature extraction network for 16 channels. Note that all the outputs are scaled to the same dimensions for visualization. The output of the first pooling layer with a mask and with no mask produces images that can be visually understood. The output of the third pooling layer produces more specific features that are difficult to interpret visually. Another visualization is given for the output of the LBP layer which is placed after a convolutional layer for manipulating the output of the transfer learning model according to the current problem. In addition, the output of the last layer, which is the relu activation function, is visualized. A close inspection of the same features in the same locations with mask and with no-mask shows visible differences.

The Faster R-CNN and YOLO mask detection subsystems were compared to the other studies that apply face mask detection. Table 5 shows comparison results. In the table, both [13] and [27] reported AP scores just for “with mask” detection. Therefore, the comparison was applied according to the detection of “with mask” results. The proposed YOLO system shows the best AP score compared to the literature. Also, the lowest log average miss rate is

| Study          | Feature extractor | Detector | Dataset          | AP(%) | log average miss rate | Time (per image) |
|----------------|-------------------|----------|------------------|-------|-----------------------|------------------|
| Ge et al. [13] | 2-cascade CNNs    | LLE-CNN  | (MAFA)           | 76.4  | -                     | -                |
| Loey et al. [27]| Resnet50          | YOLO     | (MMD + FMD)      | 81.0  | 0.4                   | -                |
| Ieamsaard et al. [20] | PA-NET       | YOLO     | FMD              | 87.1  | -                     | -                |
| Singhet al. [53] | -                 | YOLO     | Custom           | 55.0  | -                     | 0.045s           |
| Singhet al. [53] | -                 | YOLO     | Custom           | 62.0  | -                     | 0.15s            |
| Proposed       | Resnet50          | Faster R-CNN | FMD            | 78.0  | 0.44                  | 0.399s           |
| Proposed       | Resnet50          | YOLO     | FMD              | 92.0  | 0.14                  | 0.022s           |
| Proposed       | proposed 2-part feature extractor | YOLO | FMD              | 95.0  | 0.07                  | 0.021s           |
observed with the proposed YOLO with a 2-part feature extractor. Moreover, the proposed Faster R-CNN system outperforms [13] and is second behind [27]. In [27], authors used a merged database of FMD (used in this study) and Medical Masks Database (MMD). Also, the authors removed bad-quality images. Although FMD includes bad-quality images, as they mentioned, bad-quality images were not removed in this study for a more reliable comparison with the subsequent studies.

The approximate detection times for the proposed system are also presented in Table 5. As can be seen in the table, the system that includes a 2-part feature extractor YOLO integration works fastest compared to other proposed methods. Since Darknet19 is a smaller network, it is expected to run faster than Resnet50.

### 4.4 System Testing

Figure 10 shows sample frames from FMD dataset (Fig. 10a, b, c, d) and during real-time environment (Fig. 10e, f). As seen in Fig. 10b and c, the proposed system can be applied to a classroom environment. Although some people have not been detected in some frames, the human detector generally produces promising results. Also, the face mask detector performed well for these figures. Figure 10e and f illustrate samples from real-time test environment. Although the camera, used in real-time, has low-resolution and the environment has poor lighting conditions, the performance is satisfactory. The system is not affected by different eye and facial expression recognition.

Poor performance can be expected with lower resolution cameras as camera capacity will affect image quality. On the other hand, the images presented in the Fig. 10e and f were taken with a webcam of 2.0 MP. As can be seen from these images, our system produces satisfactory results even in low resolution cameras. As seen in Fig. 10b and d, people who are further away from the camera sometimes cannot be detected. Similarly, even in the
human visual system, detecting/analyzing objects very far away may perform poorly. Thus, this can be an expected situation for camera systems.

This study’s weakness is determining the social distance along the x and y-axis and not using the z-axis. Using a single camera to measure the relative position between humans along the x and y-axis is possible. One way can be using multiple camera systems to measure this position along the z-axis.

5 Conclusions

This paper presents a deep learning-based social distance and face mask detection system. With this system, it aims to decrease the spread rate of diseases transferred by respiratory droplets. The system first detects people in the environmental images and calculates the physical distance between them. Then, the system uses a mask detection module. We implemented this method using a 2-cascaded YOLO-v2 model. We designed the first cascade for human detection and the second cascade for mask detection. In addition, a novel two-part feature extraction approach is adapted to this system. In this approach, the first part extracts available features using the transfer learning, and the second part extracts specific features for the problem. Moreover, for comparison, a 2-cascaded Faster R-CNN model was developed for the same tasks. We applied different pre-trained networks with YOLO and Faster R-CNN. The system produces the best AP as 66% for human detection using Resnet50 in YOLO and 95% for mask detection using the proposed 2-part feature extractor in YOLO. The system produced better results compared to the literature.

We plan to improve the study for future work with better deep learning models such as different YOLO versions and different datasets. This system can be extended to the z-axis estimation of social distance using multiple camera systems. Alternatively, a fish-eye camera placed on top-view can be used for better distance detection. In either case, high-quality samples can be collected to form public datasets.

Availability of data and materials The authors declare that all data supporting the findings of this study are available within the article.

Declarations

Conflict of interest/Competing interests The authors did not receive support from any organization for the submitted work.

References

1. Bellotto N, Hu H (2009) Multisensor-based human detection and tracking for mobile service robots. IEEE T Syst Man Cy B 39(1):167–181. https://doi.org/10.1109/TSMCB.2008.2004050
2. Bhambani K, Jain T, Sultanpure KA (2020) Real-time face mask and social distancing violation detection system using yolo. In: 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC), pp 1–6. https://doi.org/10.1109/B-HTC50970.2020.9297902
3. Boccaletti S, Ditto W, Mindlin G, Atangana A (2020) Modeling and forecasting of epidemic spreading: The case of Covid-19 and beyond. Chaos, Solitons & Fractals 135:109794. https://doi.org/10.1016/j.chaos.2020.109794
4. Castillo O, Melin P (2020) Forecasting of COVID-19 time series for countries in the world based on a hybrid approach combining the fractal dimension and fuzzy logic. Chaos, Solitons & Fractals 140:110242. https://doi.org/10.1016/j.chaos.2020.110242
5. Castillo O, Melin P (2021) A Novel Method for a COVID-19 Classification of Countries Based on an Intelligent Fuzzy Fractal Approach. Healthcare 9(2):196. https://doi.org/10.3390/healthcare9020196
6. Chollet F (2016) Xception: Deep learning with depthwise separable convolutions. arXiv:1610.02357
7. Cortes C, Vapnik V (1995) Support-vector vectors. Mach Learn 20(3):273–297. https://doi.org/10.1023/A:102267411411
8. Dalal N, Triggs B (2005) Histograms of Oriented Gradients for Human Detection. In: IEEE COMPUT SOC CONF, vol 1, pp 886–893. https://doi.org/10.1109/CVPR.2005.177
9. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2012) The pascal visual object classes challenge. (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
10. Falkena W (2020) xml2struct. https://www.mathworks.com/matlabcentral/fileexchange/28518-xml2struct
11. Fan D-P, Zhou T, Ji G-P, Zhou Y, Chen G, Fu H, Shen J, Shao L (2020) Inf-Net: Automatic COVID-19 Lung Infection Segmentation From CT Images. IEEE T Med Imaging 39(8):2626–2637. https://doi.org/10.1109/TMI.2020.2996645
12. Gad A, ElBary G, Alkhedher M, Ghazal M (2020) Vision-based approach for automated social distance violators detection. In: International conference on innovation and intelligence for informatics, computing and technologies, pp 1–5. https://doi.org/10.1007/978-3-030-50389-5_5
13. Ge S, Li J, Ye Q, Luo Z (2017) Detecting masked faces in the wild with LLE-CNNs. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 426–434. https://doi.org/10.1109/CVPR.2017.53
14. Guo Y, Qin W, Wang Z, Yang F (2021) Factors influencing social distancing to prevent the community spread of COVID-19 among Chinese adults. Prev Med 143:106385. https://doi.org/10.1016/j.ypmed.2020.106385
15. Han F, Shan Y, Cekander R, Sawhney HS, Kumar R (2006) A two-stage approach to people and vehicle detection with HOG-based SVM. Performance Metrics for Intelligent Systems 2006 Workshop, pp 133–140
16. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: PROC CVPR IEEE, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
17. Hong Q, Wang Z, He Z, Wang N, Tian X, Lu T (2020) Masked face recognition with identification association. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), pp 731–735. https://doi.org/10.1109/ICTAI50040.2020.00116
18. Hou YC, Baharuddin MZ, Yussof S, Dzulkifly S (2020) Social distancing detection with deep learning model. In: International conference on information technology and multimedia, pp 334–338. https://doi.org/10.1109/ICIMU49871.2020.9243478
19. Htet Lin H (2020) Smart feature fusion and model for human detection. Rev Comput Eng Res 7(1):38–46. https://doi.org/10.18488/journal.76.2020.71.38.46
20. Ieamsaard J, Charoensook SN, Yammen S (2021) Deep learning-based face mask detection using yolov5, pp 428–431. IEEE. In: 2021 9th International Electrical Engineering Congress (iEECON). https://doi.org/10.1109/iEECON51072.2021.9440346
21. Jason R, Gunawan AAS (2021) Forecasting Social Distancing impact on COVID-19 in Jakarta using SIRD Model. Procedia Comput Sci 179:662–669. https://doi.org/10.1016/j.procs.2021.01.053
22. Jiang Y, Ma J (2015) Combination features and models for human detection. In: PROC CVPR IEEE, pp 240–248. https://doi.org/10.1109/CVPR.2015.7298620
23. Kaggie (2020) Face mask detection. https://www.kaggle.com/andrewmvd/face-mask-detection
24. Krishna Vinay G, Haque SM, Venkatesh Babu R, Ramakrishnan KR (2016) Sparse representation-based human detection: a scale-embedded dictionary approach. Signal Image Video Process 10(3):585–592. https://doi.org/10.1007/s11760-015-0781-5
25. Lalitha SD, Thyagarajan KK (2019) Micro-facial expression recognition based on deep-rooted learning algorithm. Int J Comput Intell Syst 12(2):903. https://doi.org/10.1007/s11760-015-0781-5
26. Li Y, Guo K, Lu Y, Liu L (2021) Cropping and attention based approach for masked face recognition. Appl Intell 51(5):3012–3025. https://doi.org/10.1007/s10489-020-02100-9
27. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2020) Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustainable Cities and Society, p 102600. https://doi.org/10.1016/j.scs.2020.102600
28. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement 167:108288. https://doi.org/10.1016/j.measurement.2020.108288
29. Maghari AYA (2020) Recognition of partially occluded faces using regularized ICA. Inverse Problems in Science and Engineering, pp 1–20. https://doi.org/10.1080/17415977.2020.1845329
30. Meivel S, Indira Devi K, Uma Maheswari S, Vijaya Menaka J (2021) Real time data analysis of face mask detection and social distance measurement using matlab. Materials Today: Proc. https://doi.org/10.1016/j.matpr.2020.12.1042

31. Melin P, Monica JC, Sanchez D, Castillo O (2020) Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: The case of mexico. Healthcare 8(2):181. https://doi.org/10.3390/healthcare82020181

32. Mi C, Zhang Z, He X, Huang Y, Mi W (2015) two-stage classification approach for human detection in camera video in bulk ports. Pol Marit Res 22(s1):163–170. https://doi.org/10.1515/pomr-2015-0049

33. Mohan P, Paul AJ, Chirania A (2020) A Tiny CNN architecture for medical face mask detection for resource-constrained endpoints. arXiv:2011.14858

34. Nguyen DT, Li W, Ogubona PO (2016) Human detection from images and videos: a survey. Pattern Recognit 51:148–175. https://doi.org/10.1016/j.patcog.2015.08.027

35. Nie Q, Zou Y-B, Lin JC-W (2020) Feature extraction for medical CT images of sports tear injury. Mob Netw Appl 26(1):404–414. https://doi.org/10.1007/s11036-020-01675-4

36. Oulefki A, Agaian S, Trongtirakul T, Kassah Laouar A (2021) Automatic COVID-19 lung infected region segmentation and measurement using CT-scans images. Pattern Recognit 114:107747. https://doi.org/10.1016/j.patcog.2020.107747

37. Oztel I (2020) Human detection system using different depths of the resnet-50 in faster r-cnn. In: International symposium on multidisciplinary studies and innovative technologies, pp 1–5. https://doi.org/10.1109/ISMSITS0672.2020.9255109

38. Oztel I, Yolcu G, Oz C (2019) Performance comparison of transfer learning and training from scratch approaches for deep facial expression recognition. In: International conference on computer science and engineering, pp 1–6. https://doi.org/10.1109/UBMK.2019.8907203

39. Padilla R, Netto SL, da Silva EAB (2020) A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), pp 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130

40. Pietikäinen M, Hadid A, Zhao G, Ahonen T (2011) Computer vision using local binary patterns vol 40. Springer, London. https://doi.org/10.1007/978-0-85729-748-8

41. Qiao L, Lin JC-W (2022) Research on standardized feature positioning technology of motion amplitude based on intelligent vision. Mobile Networks and Applications. https://doi.org/10.1007/s11036-021-01883-6

42. Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Sensors (Switzerland) 20(18):1–23. https://doi.org/10.3390/s20185236

43. Redmon J (2016) Darknet: Open source neural networks in C. https://pjreddie.com/darknet/

44. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: PROC CVPR IEEE, pp 779–788. https://doi.org/10.1109/CVPR.2016.01675

45. Redmon J, Farhadi A (2017) YOLO9000: Better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6517–6525, IEEE. https://doi.org/10.1109/CVPR.2017.690

46. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE T Pattern Anal 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

47. Rushi ME, Yussof S, Ali M, Abobakr Hassan AA (2020) Mysd: A smart social distancing monitoring system. In: International conference on information technology and multimedia, pp 399–403. https://doi.org/10.1109/ICIMU49871.2020.9243569

48. Sathyabama B, Devpura A, Maroti M, Rajput RS (2020) Monitoring pandemic precautionary protocols using real-time surveillance and artificial intelligence. In: International conference on intelligent sustainable systems, pp 1036–1041. https://doi.org/10.1109/ICISSS.2020.9315934

49. Sharma S, Kumar V (2020) Low-level features based 2D face recognition using machine learning. Int J Intell Eng Inform 8(4):305. https://doi.org/10.1504/IJIEI.2020.112038

50. Sharma S, Kumar V (2020) Voxel-based 3D occlusion-invariant face recognition using game theory and simulated annealing. Multimed Tools Appl 79(35-36):26517–26547. https://doi.org/10.1007/s11042-020-09331-5

51. Sharma S, Kumar V (2022) 3D face reconstruction in deep learning era: A Survey. Archives of Computational Methods in Engineering. https://doi.org/10.1007/s11831-021-09705-4

52. Shi Y, Dong X, Shi D, Yang Q (2015) Human detection using color and depth information by kinect based on the fusion method of decision template. In: The second seminar on soft computing, vol 1, pp 1–6
53. Singh S, Ahuja U, Kumar M, Kumar K, Sachdeva M (2021) Face mask detection using YOLOv3 and faster r-CNN models: COVID-19 environment. Multimed Tools Appl 80(13):19753–19768. https://doi.org/10.1007/s11042-021-10711-8

54. Somaldo P, Ferdiansyah FA, Jati G, Jatmiko W (2020) Developing smart covid-19 social distancing surveillance drone using yolo implemented in robot operating system simulation environment. In: Humanitarian technology conference, pp 1–6. https://doi.org/10.1109/R10-HTC49770.2020.9357040

55. Sumit SS, Rambli DRA, Mirjalili S, Ejaz MM, Miah MSU (2022) Restinet: On improving the performance of tiny-yolo-based cnn architecture for applications in human detection. Appl Sci 12:9331. https://doi.org/10.3390/app12189331

56. Sun T, Wang Y (2020) Modeling COVID-19 epidemic in Heilongjiang province. China. Chaos, Solitons & Fractals 138:109949. https://doi.org/10.1016/j.chaos.2020.109949

57. Thornton SM, Hoffelder M, Morris DD (2008) Multi-sensor detection and tracking of humans for safe operations with unmanned ground vehicles. IEEE Workshop on Human Detection from Mobile Platforms, pp 1–8

58. Varela-Santos S, Melin P (2021) A new approach for classifying coronavirus COVID-19 based on its manifestation on chest X-rays using texture features and neural networks. Inform Sci 545:403–414. https://doi.org/10.1016/j.ins.2020.09.041

59. Vu HN, Nguyen MH, Pham C (2021) Masked face recognition with convolutional neural networks and local binary patterns. Applied Intelligence. https://doi.org/10.1007/s10489-021-02728-1

60. World Health Organization (2020) Coronavirus disease 2019 (COVID-19) Situation Report – 72. Technical report, WHO. https://apps.who.int/iris/bitstream/handle/10665/331685/nCoVsitrep01Apr2020-eng.pdf

61. Xia L, Chen C-C, Aggarwal JK (2011) Human detection using depth information by Kinect. In: CVPR 2011 WORKSHOPS, pp 15–22. IEEE. https://doi.org/10.1109/CVPRW.2011.5981811

62. Yadav S (2020) Deep learning based safe social distancing and face mask detection in public areas for COVID-19 safety guidelines adherence. Int J Res Appl Sci Eng Technol 8(7):1368–1375. https://doi.org/10.22214/ijraset.2020.30560

63. Zhang X, Zhou X, Lin M, Sun J (2017) ShuffleNet: An extremely efficient convolutional neural network for mobile devices. arXiv: 1707.01083

64. Zheng W, Hou W, Lin JC-W (2022) A deep learning based feature entity relationship extraction method for telemedicine sensing big data. Mobile Networks and Applications. https://doi.org/10.1007/s11036-022-02024-3

65. Zhu Q, Yeh M-C, Cheng K-T, Avidan S (2006) Fast human detection using a cascade of histograms of oriented gradients. In: IEEE COMPUT SOC CONF, vol 2, pp 1491–1498. https://doi.org/10.1109/FRIOO.2006.119

66. Zhu X, Zhang J, Li H, Fournier-Viger P, Lin JC-W, Chang L (2017) FRIOD: A deeply integrated feature-rich interactive system for effective and efficient outlier detection. IEEE Access 5:25682–25695. https://doi.org/10.1109/access.2017.2771237

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.