Lightweight Convolutional Neural Network Model for Human Face Detection in Risk Situations

Michał Wieczorek, Jakub Siłka, Marcin Woźniak, Sahil Garg, *Member, IEEE*, and Mohammad Mehedi Hassan, *Senior Member, IEEE*

***Abstract*—In this article, we propose a model of face detection in risk situations to help rescue teams speed up the search of people who might need help. The proposed lightweight convolutional neural network (CNN) architecture is designed to detect faces of people in mines, avalanches, under water, or other dangerous situations when their face might not be very visible over surrounding background. We have designed a novel light architecture cooperating with the proposed sliding window procedure. The designed model works with maximum simplicity to support mobile devices. An output from processing presents a box on face location in the screen of device. The model was trained by using Adam and tested on various images. Results show that proposed lightweight CNN detects human faces over various textures with accuracy above 99% and precision above 98% what proves the efficiency of our proposed model.**

***Index Terms*—Deep learning, face recognition, lightweight convolutional neural network (CNN).**

I. Introduction

RESCUE systems can benefit from the support of various artificial intelligence models. Modern rescue actions are very often based on the detection of variety of life symptoms to faster help people in need, i.e., face detection is very important to locate survivors. There are several models which provide efficient techniques of face detection by using deep learning. Convolutional neural network (CNN) developed in a form crafted to support device which serves as detector can be a very efficient face detector. In mobile systems CNN model must be devoted to work on minimum requirements. Second, the goal is to make the system detect on various background and texture. Third, such models should be able to detect faces in various conditions and clarity. In [1], the authors presented an idea of using CNN coworking with other algorithms, which helped the network to define regions in the image of potential attention, where faces might appear. The model proposed in [2] was using mixed features to analyze facial gesture on light devices of mobile computing. In [3], the authors discussed how deep learning model can influence subjective learning technique in face detection task. Also CNN models are adjusted to fulfill conditions of system architecture or input images. Light models of CNN are reduced in some aspects just to support faster detection. A proposal of [4] presented faster CNN by reduction of some unimportant architecture aspects. The empowered model discussed in [5] presented CNN applied to malware classification. The model proposed in [6] was oriented on detecting faces in static and video sequences. As a result the system was able to support *ad hoc* action of face detection which can be crucial in risk situations when by symptoms rescue team is able to identify people in need much faster. Liveness detection models from face images are also used in identity verification systems. As proposed in [7], such systems may efficiently use artificial intelligence in real time environments. LiveNet as face liveness oriented CNN architecture was presented in [8]. This model was using generalized features on the way to real time systems. In [9], the authors proposed a weakly supervised model in which faces were detected by using discriminative analysis. In [10], the authors proposed a model of face liveness detection by using sparse dynamic feature maps analysis for CNN architecture. There are also other face detection models. In [11], the authors proposed a fuzzy classifier for face detection, which was evolving during the process by adjusting components to input images. The model discussed in [12] was built on deep learning to detect faces in video sequences.

A. Related Works

There are many interesting models of CNN developed for face detection task in various circumstances. We can learn about construction for traffic systems and car cameras, thermal or infrared vision, mobile computing, security, and many others. A model presented in [13] was oriented on face detection from thermal infrared recordings, where the use of a normal camera is not possible due to various conditions. As a result the system was able to analyze some aspects of faces and provide simple

Manuscript received May 18, 2021; revised September 17, 2021; accepted November 4, 2021. Date of publication November 22, 2021; date of current version April 13, 2022. This work was supported by the King Saud University Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP-2021/18. Paper no. TII-21-2101. *(Corresponding author: Marcin Woźniak.)*

Michał Wieczorek, Jakub Siłka, and Marcin Woźniak are with the Faculty of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland (e-mail: michal_wieczorek@hotmail.com; kubasilka@gmail.com; marcin.wozniak@polsl.pl).

Sahil Garg is with the Electrical Engineering Department, École de Technologie Supérieure,Université du Québec, Montreal, QC H3C 1K3, Canada (e-mail: sahil.garg@ieee.org).

Mohammad Mehedi Hassan is with the College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mmhassan@ksu.edu.sa).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3129629.

Digital Object Identifier 10.1109/TII.2021.3129629

decision support to the operator. The model discussed in [14] was using feature aggregation techniques to work on light model for tiny face detection. We can also learn about efficient models for face detection in traffic. These models help to detect people on the street to avoid accidents, i.e., in autonomous cars or to support the driver with attention alerts. In [15], the authors proposed a model for pixel statistical analysis to detect faces of pedestrians in the traffic. A system proposed in [16] was using a cascade version of CNN devoted for driving systems to support detection of people crossing the highway, while the model proposed in [17] was developed for pedestrians detection by using face detection model based on CNN. In [18], the authors proposed a model for traffic environments, where faces were detected by comparison of candidates generations. In [19], a model of face detection was developed for face detection in nature environments, where deep learning architecture was trained to distinguish face features over extensively changing textures of the nature. The model presented in [20] was oriented on fast detection of faces devoted for mobile devices. As a result the system was searching for special global characteristics which helped in fast detection. An extensive literature overview for face detection models was presented in [21].

Our proposed model is a devoted lightweight CNN architecture developed for fast detection of human faces in risk situations when a rescue team is using light devices like smart glasses or mobiles to scan the surroundings for people. The model is efficient in detection of faces over complex textures or in environments, where human face might be distorted due to coverage or other situations. Proposed model is devoted to searching of human faces in mines, water, avalanches, and other similar situations, where a rescue team may use light devices to speedup the process of helping by searching for areas, where humans are awaiting help. The novelty of our model is in the proposed lightweight CNN construction, where the number of necessary image processing operations was reduced to the minimum. The proposed CNN is evaluating segments returned from our developed sliding window procedure. The whole model is developed in way to support parallel processing on devices with adequate processors. The developed architecture was trained by using Adam algorithm, as this one was the most efficient in our research examinations. We have tested the model on various images. Experimental simulations have shown that accuracy is above 99% and precision is above 98%.

II. Developed System Model

Our system is developed for fast face detection by using mobile devices like goggles or smartphones which need simplified processing for efficient work. The idea of the system is to make it available for any rescue team which may need fast and reliable support during rescue actions. We assume that the team will use monitoring equipment by which the scene of rescue will be scanned in search of human faces. The rescue team will follow direction of the marked face and repeat scanning until people in need are found. We have implemented augmented reality (AR) environment to conduct experiments for our research project. The idea of rescue action by using our proposed system is presented in Fig. 1. We can see sample screens from mountain rescue action simulation, where the team is locating the survivor under crashed construction, and in the second situation we can see the visualization from our model of the rescue action for people after explosion in a block, where we assume that our model is used from a helicopter as a support to rescue team on ground. The presented experiments are using our developed lightweight CNN for detection of survivor faces, which are marked on the image by our proposed procedure. The whole system is trained in an adaptive way to improve future retraining by using images from past actions to retrain the lightweight CNN for constantly improved detection.

A. Lightweight CNN Architecture

CNN are designed to work with images. Construction of CNN makes use of transforming an image into a numerical vector which is classified by the final fully connected output. On the CNN layer, we are using operations of convolution, pooling, and dropout. Convolution and pooling help to extract the most important features of objects from images, while dropout helps to reduce those which are unimportant for detection. As a result of this processing image is transformed into numerical vector containing the most important information necessary to detect desired objects. To make our proposition available for a variety of mobile devices we have designed lightweight CNN architecture. The developed model consists of minimum number of pooling and convolution operations to perform efficient face detection. The developed lightweight CNN performs image processing in three blocks. The first block is doing convolution with kernel size 5 after which $\max$ pooling operation is performed. The second block works with convolution kernel size 10 and $\max$ pooling operation, while in the third block we use convolution kernel size 15 together with $\max$ pooling. The result of this processing is forwarded to flatten layer for detection. Applied activation functions in our architecture are as follows.

1) *Sigmoid*

$$S(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$S'(x) = S(x)(1 - S(x)). \tag{2}$$

2) *ReLu*

$$R(x) = \max(0, x) \tag{3}$$

$$R'(x) = \begin{cases} 0 \text{ when } x < 0 \\ 1 \text{ when } x > 0 \end{cases}. \tag{4}$$

In Fig. 2 we can see a scheme of our proposed lightweight CNN architecture. According to this decision scheme a face is detected on the image batch, from boxing phase.

B. Face Marking Process—Boxing Phase

In our system we are using the developed lightweight CNN in two phases. In the beginning of detection a raw scan from mobile device is forwarded to CNN evaluation. The proposed model decides if this image contains any faces. If the response is

Fig. 1. On the left, sample presentation of a mountain rescue action from our research experiments in the developed system. After an avalanche while using the proposed face detection model on mobile device rescue team search for survivors. When reaching the location of action, the mountain rescue team is using a mobile device with the proposed lightweight CNN to scan the landscape for face and move toward marked direction. Scanning is repeated if the face is not visible and team move forward again. These steps are repeated until the person in need is found. On the right, in a building after explosion while using the proposed face detection model, the rescue team search for survivors from drone or flying assistant. After reaching location of action the rescue team is using a drone or helicopter camera with proposed lightweight CNN to scan the landscape for faces and direct the land team to move toward marked direction. Scanning is repeated until the survivor is found.

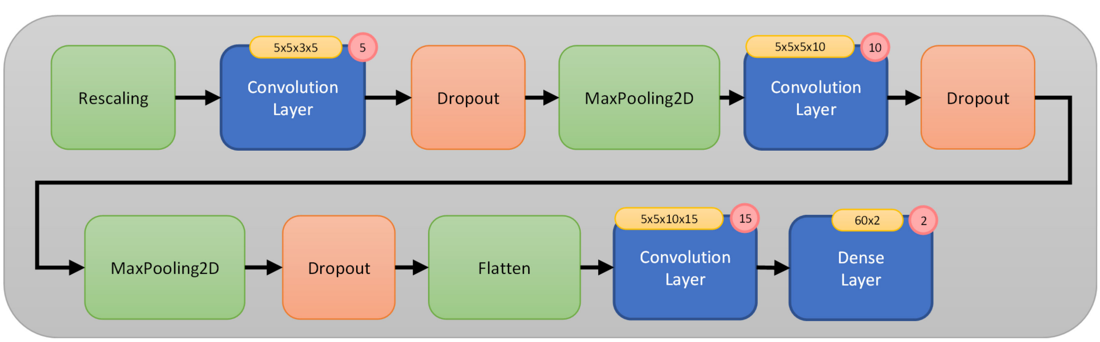


Fig. 2. Developed lightweight CNN architecture in which a minimum number of processing layers is adapted to detect human faces. On each layer yellow number represents kernel features, while red represents applied bias.

negative, the procedure ends. If faces are detected we are using our lightweight CNN in parallel mode. We start by constructing a sample set of batches from the initial image. Each of the batches is forwarded to separate GPU core for detection. Decisions from all cores are collected. The system starts second phase of face detection. All marked batches are compared to reduce the number of similar ones to minimum. Applied boxing phase is presented in Algorithm 1. As a result we have a set of unique batches which show faces of survivors. For each of these batches, dimensions and position over the image are stored in memory. This set is used to mark face over initial image. The result is returned to the mobile device, where an image is presented to the user. These operations enable the rescue team to efficiently scan the horizon of action and find locations of people in need. Therefore the rescue team can directly move toward them with help. Our proposed detection model is as simple as possible

Algorithm 1: Box Selection Process.

Get generated boxes list,
for each box in the list **do**
  Compare box to all other boxes by using positions of box corners, position of center, CNN result,
  if positions of box corners are the same for two different boxes **then**
    Remove one box as duplicate,
  else
    for each box in the list **do**
      if CNN result of box < Detection Threshold **then**
        Remove this box from the list,
      else
        Find the nearest box $B2$ by Euclidean metric result ||center position $B1$ - center position $B2$|| is minimum among all,
        if any borders of boxes $B1$ and $B2$ overlap **then**
          Average the $B1_{x_1}, B1_{x_2}, B1_{y_1}, B1_{y_2}$ and $B2_{x_1}, B2_{x_2}, B2_{y_1}, B2_{y_2}$ positions,
          Average the $B1$ and $B2$ certainty,
        end if
      end if
    end for
  end if
end for
Present final boxes on the image.

to boost mobile devices on faster processing. Modern devices have construction oriented on augmented reality (AR) features and simple operations therefore our simplified construction will boost the system in calculations. Applied face marking model is presented in Fig. 3.

The training of our system is done on external server which only sends the final lightweight CNN architecture components to the mobile client. Processing is run accordingly in adaptive way. We assume that implemented model learns from collected images. The system is using captured images to retrain the composed lightweight CNN, so that each newly classified image also helps to improve overall efficiency. Fig. 4 presents our idea for two phase adaptive work.

C. Adam Training Procedure

In order to optimize training process and improve performance of our model, we have used Adam optimization algorithm. It is widely used in machine learning research due to its huge performance boost with small computational load to the system. Adam formula can be described as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{5}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{6}$$

where β_1 and β_2 parameters are hyperparameters and g is the current gradient value of the error function. Values t_m and m_t are used to calculate the correlations written as $\hat{m}_t$ nd $\hat{v}_t$ according

Algorithm 2: Adam Training Process.

Random weights generation,
while *global error value* $\varepsilon < error\_value$ **do**
  Compute signal flow for *input image*,
  for each image **do**
    for each iteration **do**
      Use activation function on the neuron from (1) to (3),
      Process transformation operations on the image,
    end for
    Forward transformed image,
  end for
  for each image **do**
    for each neuron in layer **do**
      Calculate the error value $\varepsilon_i(t)$ from (11),
      Compute weights change from (5) to (8),
      Update weights w_{i+1} by using (9),
    end for
  end for
  Calculate *global error* ε from (10),
end while

to the following equations:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{7}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}. \tag{8}$$

Using the values computed above, the final formula for changing weights can be defined as a change between current weight w_t and calculated correlations

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{9}$$

where η is a learning rate (in our case 0.0005) and ϵ is a small constant value used for numerical stability. The algorithm is presented in Algorithm 2.

The training algorithm is run on the network as long as we minimize detection error value. In this procedure we have used cross-entropy function

$$\varepsilon = -\sum_{p=0}^{P} (d^p \log y^p + (1-d^p) \log(1-y^p)) \tag{10}$$

where y is the received value for training image, while d is expected value for this image, P is the number of training images. In iterations of applied Adam algorithm, weights of lightweight CNN are recalculated regarding received error on the detection neuron

$$\varepsilon_i(t) = d^p - y^p \tag{11}$$

where we understand this value as a difference between expected and received images. To ensure that this was the best option we have conducted an experiment on timing presented in Table I and the compared results form other algorithms are in Table II which assured us in this choice.

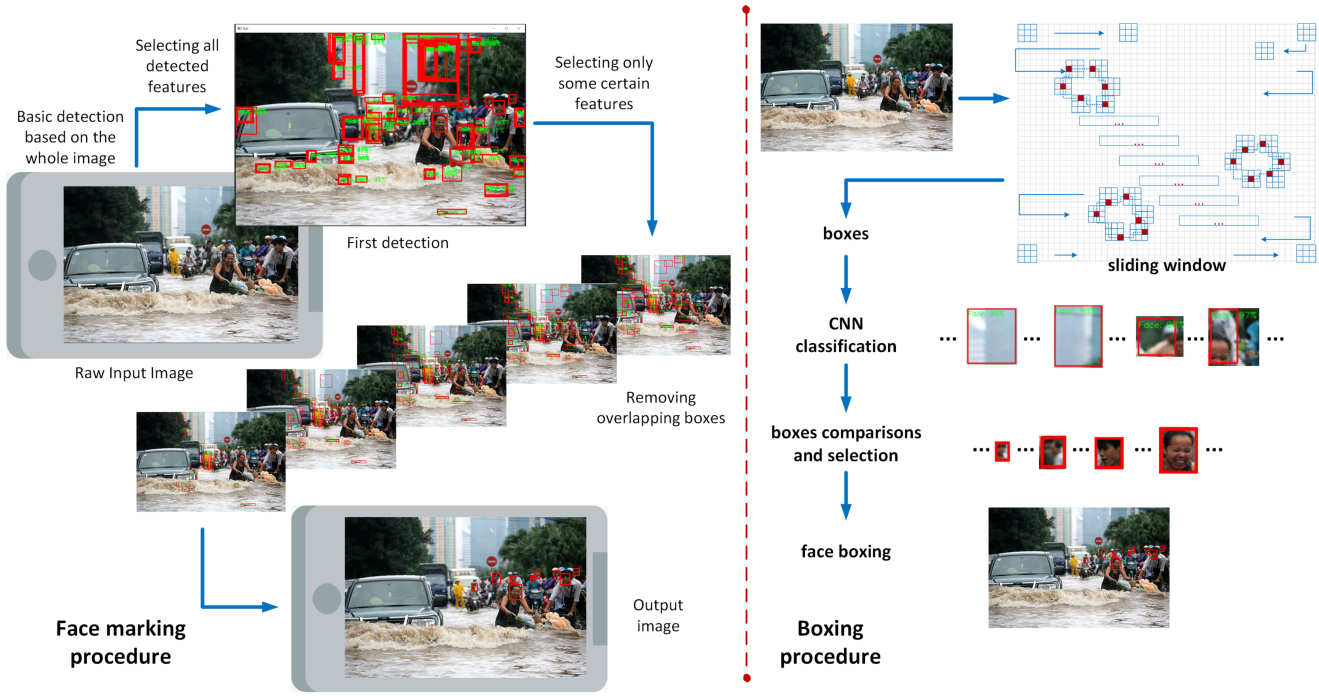


Fig. 3. Face marking procedure starts with scanning action horizon by mobile device camera. The image is evaluated by proposed architecture. If detection is positive, various dimensions of batches are processed by proposed lightweight CNN model in first phase of detection. If any batch image is classified as showing face further processing is done. The image is scanned with those various sizes of batches and each classified batch is stored. Batches are compared and repeated ones are removed. As a result system returns only final detection of faces on the display of mobile device.

TABLE I
COMPARISON OF DEVELOPED LIGHTWEIGHT CNN TRAINING TIMES BY USING DIFFERENT COMPARED METHODS

	Adadelta	Adagrad	Adamax	Adam	Ftrl	NAdam	RMSprop	SGD
time	3 min 56.78s	4 min 35.95s	3 min 55.44s	4 min 22.54s	3 min 44.66s	3 min 52.33s	4 min 12.69s	3 min 47.30s

TABLE II
COMPARISON OF MACHINE LEARNING METRICS FOR TESTING DIFFERENT OPTIMIZATION ALGORITHMS IN 200 ITERATIONS

Algorithm	Accuracy	Precision	Recall	F1	Specificity	FDR	FPR	FNR	FOR	NPV
Adam	99.08%	98.71%	99.11%	0.99	97.75%	99.02%	2.25%	0.34%	0.8%	99.2%
Nadam	98.74%	98.38%	98.64%	0.99	97.46%	98.91%	2.54%	0.7%	1.64%	98.36%
RMSprop	98.69%	98.38%	98.5%	0.98	97.6%	98.97%	2.4%	0.85%	1.98%	98.02%
Adamax	98.37%	97.72%	98.45%	0.98	96.02%	98.25%	3.98%	0.59%	1.36%	98.64%
Adagrad	80.06%	76.4%	78.52%	0.77	64.4%	82.35%	35.6%	11.61%	25.3%	74.7%
SGD	72.96%	67.1%	64.22%	0.66	56.46%	86.01%	43.54%	22.25%	57.57%	42.43%
ftrl	70.05%	35.02%	50.0%	0.41	0.0%	100.0%	0.0%	29.95%	100.0%	0.0%
Adadelta	70.05%	35.02%	50.0%	0.41	0.0%	100.0%	0.0%	29.95%	100.0%	0.0%

III. RESEARCH RESULTS

We have tested our solution in computer simulations, in which we have applied data Flickr-Faces-HQ since it is a high-quality image dataset of human faces. It was originally created as a benchmark for generative adversarial networks as train and test data. This is a collection of 70 000 images, where faces are visible in various situations. Images are of size 1024×1024 and contain considerable variation in terms of age, ethnicity, and image background. We have divided the data 70:30 as training and test data in our experiments using ten topfold cross validation model. In our research experiments we have used computer with hardware parameters CPU: Threadripper 2950x 16core/32threads, GPU: NVIDIA RTX 3090 24 GB, and RAM: 96 GB for all simulations and calculations.

In the evaluation of our model, we calculated the number of faces detected from images. We have used four classes when the face is detected correctly TP as true positive, incorrectly FP as false positive, not face is not detected TN as true negative, and not face is detected as face FN as false negative. For these classes we have calculated metrics, and in Table II we can see numerical values for our applied metrics as follows.

1) *Accuracy* $\frac{TP+TN}{TP+TN+FP+FN}$.
2) *Precision* $\frac{TP}{TP+FP}$.
3) *Sensitivity (Recall)* $\frac{TP}{TP+FN}$.

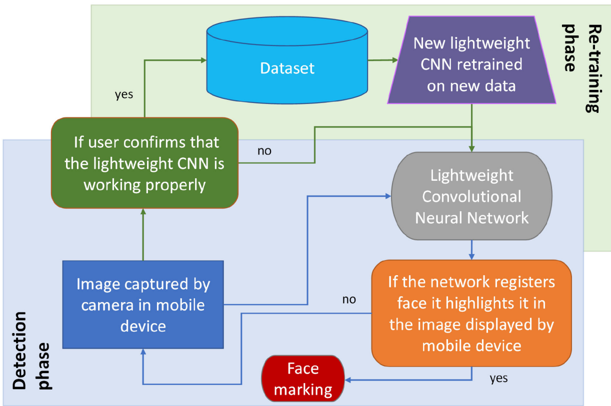


Fig. 4. Applied model of the system training, where each correctly classified image is used to retrain the system for more efficient evaluation in future actions.

4) *F1 Score* $\frac{2 \cdot TP}{2 \cdot TP + FP + FN}$.
5) *Specificity* $\frac{TN}{TN+FP}$.
6) *False Detection Rate* $\frac{FP}{FP+TP}$.
7) *False Positive Rate* $\frac{FP}{FP+TN}$.
8) *False Negative Rate* $\frac{FN}{FN+TP}$.
9) *False Omission Rate* $\frac{FN}{FN+TN}$.
10) *Negative Predictive Value* $\frac{TN}{TN+FN}$.

First we have tested various tested training algorithms to confirm our choice of Adam method as to train our developed Lightweight CNN model. Results of compared training procedures are presented in Fig. 5. Table I presents comparisons of training times by using different methods. We can see that longest times are for Adagrad, Adam, and RMSProp. However, in any case time differs not more than at maximum 1 minu. Therefore, we can assume that all models work in a similar way. Table II presents comparison of standard machine learning metrics for Adam, RMSprop, Nadam, Adamax, Adagrad, SGD, Ftrl, and Adadelta. We can see that among all tested models Adam is reaching the highest accuracy, precision, and recall metrics. We conclude that by using this model our lightweight CNN is reaching the best results. Adam, RMSprop, Nadam, and Adamax are reaching above 99% of each metric but still results of Adam are the best among all. Similarly for other metrics Adam is also reaching the best results. Results of training, both on test and train dataset, are presented in Fig. 5. We can see that training process works without any interruptions for Adam algorithm. When we compare AUC curves presented in Fig. 5 we can notice that also these metrics are the best for Adam, where they reach the highest number of correct detections in the earliest iteration. Therefore we conclude that because of using Adam to train our architecture we can assume that the proposed system will gain the best possible overall efficiency in search of people's faces in risk situations.

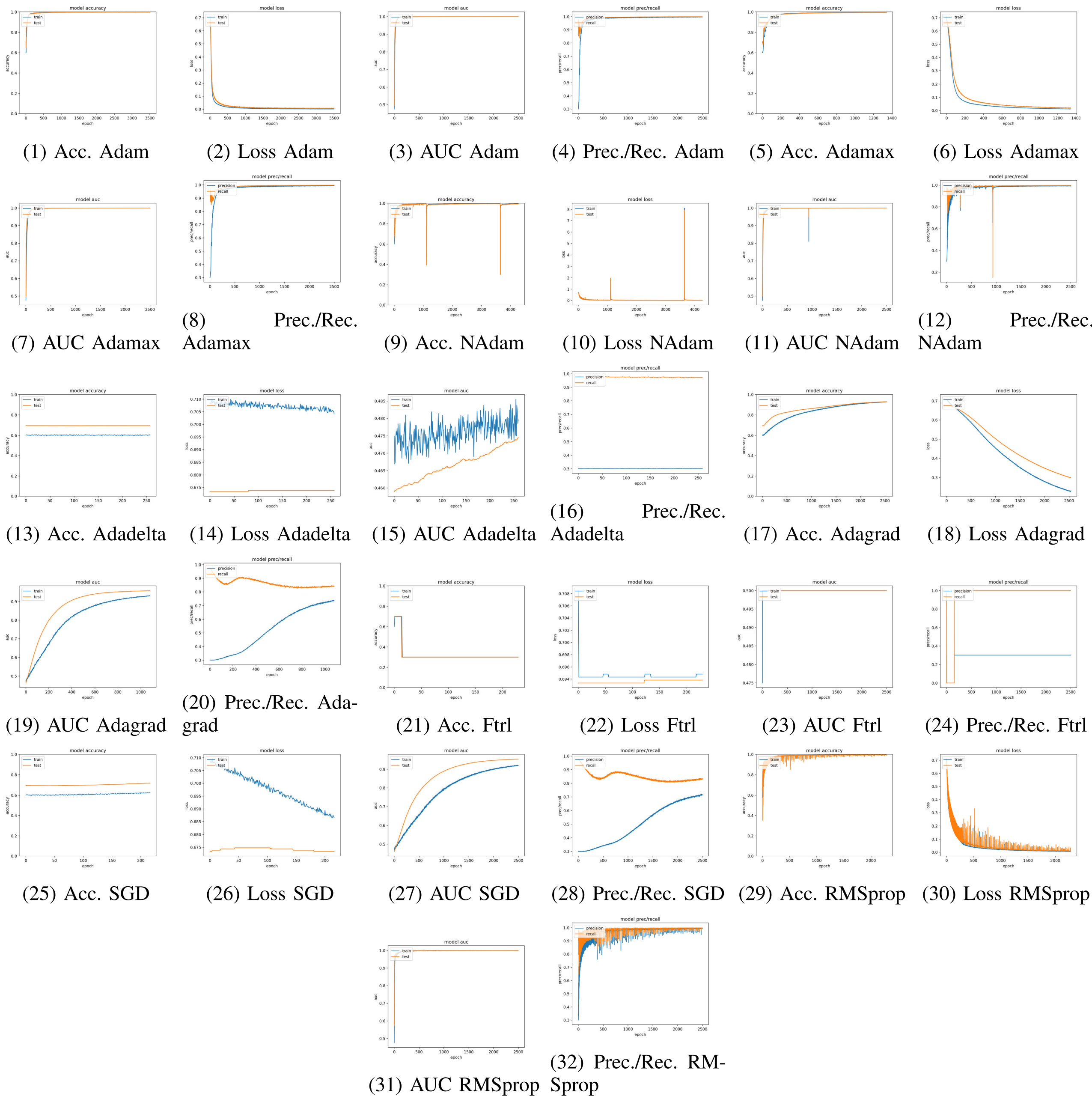
Sample results of face detection in risky situations can be seen in Fig. 6. We have tested our system for floods, avalanches, mine, fire, car accidents, and other situations, where using our proposed model may help the rescue team to find the survivors faster. In Fig. 6, we can see how the system detects faces in images, where there is just one person or more people, respectively, to the situation of rescue action. As we can see the system correctly detects most of the faces in all examples. The system works correctly independent of influence of bad visibility due to low transparency of air or weak light. This is another argument that proposed lightweight CNN is efficient an technique for rescue team equipment. In Fig. 7, we can see detection results on our validation set, in which we have 8 classes (cases of rescue scenario) of 60 samples (rescue scenes) for each class. Results show that our model works well both for samples with faces where detection was necessary, and also for samples without faces where detection was not necessary. Our model was correctly working in both situations. We can see slight disturbances in detection of 1–2 samples in case of fire, avalanche, or earthquake. This is the result that in some images faces were in dense smoke or much covered in mud. We can conclude that if the face is covered in more than half with nontransparent material our system may have some problems in detection, however, in other situations even if the face is not fully visible our system works well.

A. Conclusion

We have selected Adam algorithm for the optimization of our architecture as a result of tests and comparisons. In research experiments we were searching for the highest possible accuracy. However, efficiency of the model cannot be limited to this measure only, therefore, as a second measure we searched for the highest possible precision. This reflects on how many selected elements are important for a given abstract class, which is very important for a system that must draw attention of relevant services to people in dangerous situations. In addition to accuracy and precision, we wanted to pay attention to specificity to limit the number of false results which may distract the potential rescuer. Therefore, when we compared results of all tested algorithms for these three measures: Accuracy, precision, and specificity, we came to a conclusion that Adam would be the best algorithm for our system training.

In Table III we can see a comparison of detection results from literature proposed models. We can see that most of models are based on complex architectures, what results in higher expectations from processing device. Our idea is a lightweight CNN model therefore we can implement it on most of devices and run it without problems. Additionally our model is trained on an external server so computation burden is only related to image processing during scan. If we compare metrics, our proposed architecture is reaching the best results in presented categories. That gives another proof for efficiency of our solution.

The application part of our system would help on faster detection of survivors and people in need in variety of situations. The system works as a mobile app, and in our research simulations we tested it on images. The app will run on device, also a smartphone, so the rescue team will simply use it to scan the horizon and detect people from a distance. In our research simulations, the proposed software was efficient for human faces in dark, foggy, or even partly covered conditions. The proposed architecture is developed as lightweight CNN so that it does not



(1) Acc. Adam (2) Loss Adam (3) AUC Adam (4) Prec./Rec. Adam (5) Acc. Adamax (6) Loss Adamax (7) AUC Adamax (8) Prec./Rec. Adamax (9) Acc. NAdam (10) Loss NAdam (11) AUC NAdam (12) Prec./Rec. NAdam (13) Acc. Adadelta (14) Loss Adadelta (15) AUC Adadelta (16) Prec./Rec. Adadelta (17) Acc. Adagrad (18) Loss Adagrad (19) AUC Adagrad (20) Prec./Rec. Adagrad (21) Acc. Ftrl (22) Loss Ftrl (23) AUC Ftrl (24) Prec./Rec. Ftrl (25) Acc. SGD (26) Loss SGD (27) AUC SGD (28) Prec./Rec. SGD (29) Acc. RMSprop (30) Loss RMSprop (31) AUC RMSprop (32) Prec./Rec. RMSprop

Fig. 5. Results of training process by using various algorithms. In each chart we can see comparisons between training data (blue line) and test data (orange line).

TABLE III
COMPARISON OF OUR DEVELOPED MODEL TO OTHER RESULTS OF FACE DETECTION METHODS FROM LITERATURE

	Year	Method	Accuracy
Our model	2021	Lightweight CNN + face boxing procedure	99.61%
Zhang et al. [22]	2017	Inside Cascaded Structure and Body Part Sensitive Learning	97.43%
Zhang et al. [22]	2017	Inside Cascaded Structure	97.12%
Simbolon et al. [23]	2020	interpolation and histogram equalization	95.52%
Li et al. [24]	2015	multi-threshold AdaBoost	99.47%
Ennehar et al. [25]	2019	Haar-like features with local binary patterns and Support Vector Machine	91.04%
Alafif et al. [26]	2017	Large-Scale Deep Learning	97.40%
Dabhi et al. [27]	2016	system based on Viola-Jones algorithm	95%
Owusu et al. [28]	2019	Feed-forward neural network and Haar features	98.5%
Meena et al. [29]	2016	Viola Jones	90%
Song et al. [30]	2018	Joint Analysis of RGB and Near-Infrared Image	99.4%
Maatta et al. [31]	2011	Micro-Texture Analysis	98.0%

Fig. 6. Detection preview from sample rescue action images in various environments. We can see how the proposed system works in case of rescue in flood, drown in water, fire, mine, mountain and snow avalanche, earthquakes, collapsed buildings, and car accidents. We can see that different conditions of light and air transparency do not influence detection efficiency.

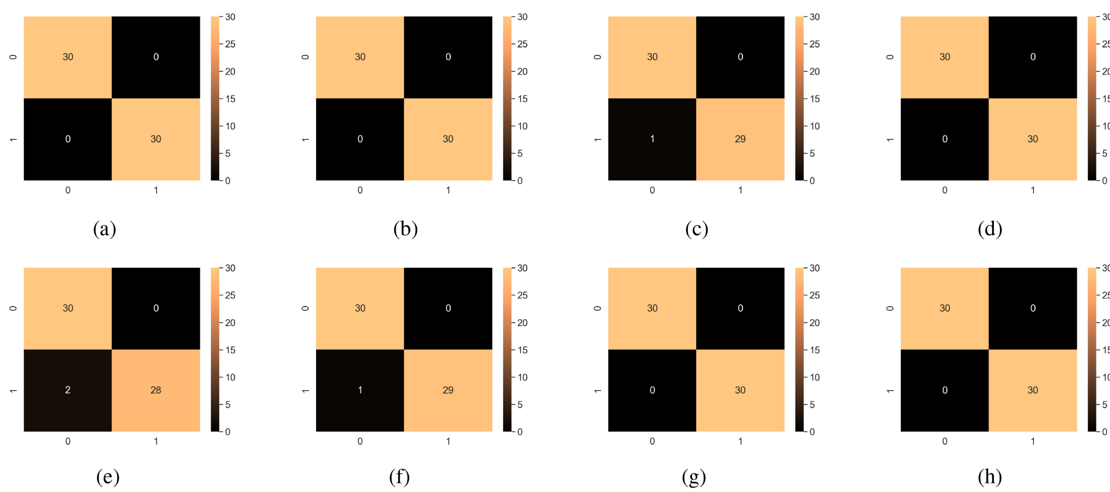


Fig. 7. Results of face detection by using our developed lightweight CNN model on validation set in eight different cases: Rescue in flood, drown in water, fire, mine, mountain and snow avalanche, earthquake, collapsed buildings, and car accidents. In each confusion matrix 0 means face detection and 1 is for no face detected result. (a) Flood. (b) Water drown. (c) Fire. (d) Mine. (e) Mountain and avalanche. (f) Earthquake. (g) House rubble. (h) Car accident.

require powerful electronics. The detector works in a form of a simple script and is efficient on most Android systems. Our proposed structure of CNN is very light, however, efficient in simple detection task. We are not making recognition of faces but just simple detection of faces in various conditions. From our experiments we have assumed that this model will work also for partly covered faces, i.e., when snow avalanche is partly covering survivors or when in a mine faces are covered with coal dust. We have not tested this model for detection of faces covered with face mask in times of pandemic. From initial results from similar cases as above we can conclude that proposed model may be also efficient, however, further experiments toward this direction are still necessary to test and possibly improve our solution. Except for positive aspects, our idea may also have some drawbacks. Our system allows the rescue crew to draw attention to the people in dangerous situations. Unfortunately, some users may rely too much on this solution, we define it as multimedia tool but human expertise is never to be replaced. Our solution is run on mobile which needs battery so it may not allow for a very long use, i.e., such as large forest fires, long see actions. We can also say with certainty that in the future, if the system would be used universally, it could be combined with GPS, so that the search commander could have greater control over the entire operation. In future it is possible to use this system not only on normal cameras but also on night vision and thermal imaging camera so it can be used in any environments.

Face detection in our system is based on sliding window model. Therefore it would be also important to boost that model. We are working to make it work faster. From our initial experiments we conclude that we can parallelize the sliding window algorithm and also vectorize the whole process which can lead to huge speedups in execution times. Our tests showed that multithreading performance rises almost linearly with the number of used cores, and vectorizing the sliding window was about 10 to 100 times faster than the initial method, depending on the image size.

IV. Conclusion

The research presented in this article discuss our idea for a system which may serve for faster detection of survivors in variety of risky situations. The proposed model is uses the developed lightweight CNN architecture to detect faces of people. The model is efficient in conditions of low air transparency when used in smoke, but also in lower light when used in darker places. The proposed architecture is possible to run on mobile devices therefore the solution can be used just on a simple mobile to scan the action horizon and detect survivors. We have applied the model of constant training on external server, as a result with each new confirmed face classification our system is retrained. Therefore the more the system is used in rescue action the more efficient it becomes.

In our future research we want to work on connecting the system between several mobile devices. Therefore in such way we can create an intelligent Internet of things infrastructure, in which detection results would be shared between devices in real time so that users could communicate and exchange detection results for more efficient coordination of rescue action. Another interesting possibility would be an introduction of constant real time computer vision for augmented reality, where users will not share images of detection but could share computer vision result in real time.

References

[1] X. Li, Z. Yang, and H. Wu, "Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks," *IEEE Access*, vol. 8, pp. 174922–174930, 2020.

[2] Z. Ni and Q. Li, "Fusion learning model for mobile face safe detection and facial gesture analysis," *IEEE Access*, vol. 8, pp. 61043–61050, 2020.

[3] X. Ding, Z. Raziei, E. C. Larson, E. V. Olinick, P. Krueger, and M. Hahsler, "Swapped face detection using deep learning and subjective assessment," *EURASIP J. Inf. Secur.*, vol. 2020, 2020, Art. no. 6.

[4] G. Guo, H. Wang, Y. Yan, J. Zheng, and B. Li, "A fast face detection method via convolutional neural network," *Neurocomputing*, vol. 395, pp. 128–137, 2020.

[5] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture," *Comput. Netw.*, vol. 171, 2020, Art. no. 107138.

[6] L. M. Dang, S. I. Hassan, S. Im, and H. Moon, "Face image manipulation detection based on a convolutional neural network," *Expert Syst.Appl.*, vol. 129, pp. 156–168, 2019.

[7] S. Liu, Y. Song, M. Zhang, J. Zhao, S. Yang, and K. Hou, "An identity authentication method combining liveness detection and face recognition," *Sensors*, vol. 19, no. 21, 2019, Art. no. 4733.

[8] Y. A. U. Rehman, L. M. Po, and M. Liu, "LiveNet: Improving features generalization for face liveness detection using convolution neural networks," *Expert Syst. Appl.*, vol. 108, pp. 159–169, 2018.

[9] Q. Huang, C. K. Jia, X. Zhang, and Y. Ye, "Learning discriminative subspace models for weakly supervised face detection," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2956–2964, Dec. 2017.

[10] Y. A. U. Rehman, L.-M. Po, and M. Liu, "SLNet: Stereo face liveness detection via dynamic disparity-maps and convolutional neural network," *Expert Syst. Appl.*, vol. 142, 2020, Art. no. 113002.

[11] J.-G. Wang, S.-C. Tai, and C.-J. Lin, "The application of an interactively recurrent self-evolving fuzzy CMAC classifier on face detection in color images," *Neural Comput. Appl.*, vol. 29, no. 6, pp. 201–213, 2018.

[12] R. Koshy and A. Mahmood, "Enhanced deep learning architectures for face liveness detection for static and video sequences," *Entropy*, vol. 22, no. 10, 2020, Art. no. 1186.

[13] M. Kopaczka, L. Breuer, J. Schock, and D. Merhof, "A modular system for detection, tracking and analysis of human faces in thermal infrared recordings," *Sensors*, vol. 19, no. 19, 2019, Art. no. 4135.

[14] C. Zhang, T. Li, S. Guo, N. Li, Y. Gao, and K. Wang, "Aggregation connection network for tiny face detection," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

[15] J. Shen, X. Zuo, J. Li, W. Yang, and H. Ling, "A novel pixel neighborhood differential statistic feature for pedestrian and face detection," *Pattern Recognit.*, vol. 63, pp. 127–138, 2017.

[16] Y. Zhang, P. Lv, X. Lu, and J. Li, "Face detection and alignment method for driver on highroad based on improved multi-task cascaded convolutional networks," *Multimedia Tools Appl.*, vol. 78, no. 18, pp. 26661–26679, 2019.

[17] J. Zheng and J. Peng, "A novel pedestrian detection algorithm based on data fusion of face images," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 5, 2019.

[18] D.-Y. Huang, C.-H. Chen, T.-Y. Chen, W.-C. Hu, Z.-B. Guo, and C.-K. Wen, "High-efficiency face detection and tracking method for numerous pedestrians through face candidate generation," *Multimedia Tools Appl.*, vol. 80, pp. 1247–1272, 2021.

[19] G. Storey, A. Bouridane, and R. Jiang, "Integrated deep model for face detection and landmark localization from "in the wild" images," *IEEE Access*, vol. 6, pp. 74442–74452, 2018.

[20] H. Zhang, X. Wang, J. Zhu, and C.-C. J. Kuo, "Fast face detection on mobile devices by leveraging global and local facial characteristics," *Signal Process.: Image Commun.*, vol. 78, pp. 1–8, 2019.

[21] N. Dagnes, E. Vezzetti, F. Marcolin, and S. Tornincasa, "Occlusion detection and restoration techniques for 3D face recognition: A literature review," *Mach. Vis. Appl.*, vol. 29, no. 5, pp. 789–813, 2018.

[22] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu, "Detecting faces using inside cascaded contextual CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3190–3198.

[23] H. F. S. Simbolon and P. Sirait, "The effect of bicubic interpolation on Viola–Jones and principal component analysis in detecting faces and helmet wearers," in *Proc. 3rd Int. Conf. Mech. Electron. Comput. Ind. Technol.*, 2020, pp. 158–164.

[24] Z. Li, W. Wang, X. Liu, and K. Lu, "An efficient multi-threshold AdaBoost approach to detecting faces in images," *Multimedia Tools Appl.*, vol. 74, no. 3, pp. 885–901, 2015.

[25] C. E. Bencheriet, "New face features to detect multiple faces in complex background," *Evolving Syst.*, vol. 10, no. 2, pp. 79–95, 2019.

[26] T. Alafif, Z. Hailat, M. Aslan, and X. Chen, "On detecting partially occluded faces with pose variations," in *Proc. 14th Int. Symp. Pervasive Syst. Algorithms Netw. 11th Int. Conf. Front. Comput. Sci. Technol. 3rd Int. Symp. Creative Comput.*, 2017, pp. 28–37.

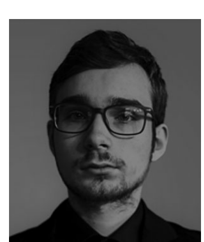
[27] M. K. Dabhi and B. K. Pancholi, "Face detection system based on Viola–Jones algorithm," *Int. J. Sci. Res.*, vol. 5, no. 4, pp. 62–64, 2016.

[28] E. Owusu, J.-D. Abdulai, and Y. Zhan, "Face detection based on multilayer feed-forward neural network and haar features," *Softw.: Pract. Experience*, vol. 49, no. 1, pp. 120–129, 2019.

[29] D. Meena and R. Sharan, "An approach to face detection and recognition," in *Proc. Int. Conf. Recent Adv. Innov. Eng.*, 2016, pp. 1–6.

[30] L. Song and C. Liu, "Face liveness detection based on joint analysis of RGB and near-infrared image of faces," *Electron. Imag.*, vol. 2018, no. 10, pp. 3731–3736, 2018.

[31] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proc. Int. Joint Conf. Biometrics*, 2011, pp. 1–7.



Michał Wieczorek is currently working toward the bachelor's diploma in computational intelligence with the Faculty of Applied Mathematics, Silesian University of Technology, Gliwice, Poland.

He is a participant of programs for highly talented students from the Rector of the Silesian University of Technology and other projects ongoing at the Faculty. His research interests include various aspects of computational intelligence, computer vision, and image processing.

Jakub Siłka is currently working toward the bachelor's diploma in computational intelligence with the Faculty of Applied Mathematics, Silesian University of Technology, Gliwice, Poland.

He is a participant of programs for highly talented students from the Rector of Silesian University of Technology and other projects ongoing at the Faculty. His research interests include various aspects of computational intelligence, optimization, and image processing.

Marcin Woźniak received the M.Sc. degree in applied mathematics from the Silesian University of Technology, Gliwice, Poland, in 2007, the Ph.D. degree in computational intelligence and the D.Sc. degree in computational intelligence from the Czestochowa University of Technology, Czestochowa, Poland, in 2012 and 2019, respectively.

He is currently an Associate Professor with the Faculty of Applied Mathematics, Silesian University of Technology. He is a Scientific Supervisor in editions of "The Diamond Grant" and "The Best of the Best" programs for highly talented students from the Polish Ministry of Science and Higher Education. He participated in various scientific projects as a Lead Investigator, Scientific Investigator, Manager, or Participant with Polish and Italian universities and IT industry. He was a Visiting Researcher with universities in Italy, Sweden, and Germany. He has authored/coauthored over 100 research papers in international conferences and journals. His research interests include neural networks with their applications together with various aspects of applied computational intelligence.

Dr. Woźniak was the recipient of the Polish Ministry of Science and Higher Education with a scholarship for an outstanding young scientist, and "TOP 2% Scientists in the World" by Stanford University for his career achievements, in 2020. He the Editorial Board Member or an Editor for *Sensors*, IEEE ACCESS, *Measurement*, *Frontiers in Human Neuroscience*, *PeerJ CS*, *International Journal of Distributed Sensor Networks*, *Computational Intelligence and Neuroscience*, *Journal of Universal Computer Science*, etc., and a Session Chair at various international conferences and symposiums, including IEEE Symposium Series on Computational Intelligence, IEEE Congress on Evolutionary Computation, etc.

Sahil Garg (Member, IEEE) received the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, Punjab, India, in 2018.

He is currently a Postdoctoral Research Fellow with École de technologie supérieure, Université du Québec, Montréal, QC, Canada. He has authored or coauthored over 60 publications in high ranked Journals and Conferences, including 40+ top-tier journal papers and 20+ reputed conference articles. His research interests include machine learning, Big Data analytics, security & privacy, Internet of Things, and cloud computing.

Dr. Garg is a Member of ACM. He wasthe recipient of the IEEE International Conference on Communications best paper award, in 2018, at Kansas City, Missouri. He is currently a Managing Editor for Springer's *Human-centric Computing and Information Sciences* journal. He is also an Associate Editor for *IEEE Network Magazine*, IEEE SYSTEM JOURNAL, *Elsevier's Applied Soft Computing*, *Elsevier's Future Generation Computer Systems*, and Wiley's *International Journal of Communication Systems*. He guest-edited a number of special issues in top-cited journals, including IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE IOT JOURNAL, IEEE NETWORK, and *Future Generation Computer Systems (Elsevier)*. In addition, he also serves as the Workshops and Symposia Officer for the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He serves/served as the workshop Chair/publicity co-Chair for several IEEE/ACM conferences, including IEEE Infocom, IEEE Globecom, IEEE International Conference on Communications, Annual International Conference On Mobile Computing And Networking, etc.

Mohammad Mehedi Hassan (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in 2011.

He is currently a Professor with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and coauthored over 210 publications, including refereed IEEE/ACM/Springer/Elsevier journals and conference papers, books, and book chapters. His research interests include edge/cloud computing, the Internet of things, cyber security, deep learning, artificial intelligence, body sensor networks, 5G networks, and social networks.