IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 11, 2020, accepted August 16, 2020, date of publication September 1, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020866

Utilizing Convolutional Neural Networks for Image Classification and Securing Mobility of People With Physical and Mental Disabilities in Cloud Systems

ISMAIL HABABEH, (Member, IEEE), IBRAHIM MAHAMEED, ABDELHADI A. ABDELHADI, AND AHMAD BARGHASH
School of Electrical Engineering and Information Technology, German Jordanian University, Amman 11180, Jordan

Corresponding author: Ismail Hababeh (ismail.hababeh@gju.edu.jo)

This research was supported by The Scientific Research of German Jordanian University (No. 56/2008-191/2018/2019).

ABSTRACT Image recognition is widely used for detecting human obstructions and identifying people with disabilities. The accuracy of identifying images of handicapped people is powered by image classification techniques that are based on deep learning methodologies. Specifically, convolutional neural networks are employed to improve image classification of people with mental and physical disabilities. In this research, images of people with different disabilities are used to extract hidden features that symbolize each disability. Three different deep learning image classifiers are built to classify images of people in wheelchairs, blind people, and people with Down syndrome. A security technique is developed that is based on multiprotocol label switching headers to secure the image mobility over cloud nodes. The proposed approach is validated by measuring the impact of the deep learning image classifiers on image classification and securing image mobility on cloud system performance. The experimental results show the effectiveness of the proposed approach in improving image prediction of disabled people and enhancing the performance of securing image mobility in cloud systems.

INDEX TERMS Deep learning, image classification, wide convolutional neural networks, image feature maps, securing image mobility, MPLS header.

I. INTRODUCTION

Aiding handicapped people has attracted the consideration of scientists regarding the challenges of identifying people with disabilities. The new technologies in image processing help in identifying features of human incapacities and can precisely recognize the images of people with disabilities such as blind people, people with Down syndrome, or people in wheelchairs. The image recognition is realized by using the deep neural networks (DNNs) [1] and convolutional neural networks (CNNs) [2] to identify image features and help solve various computer vision challenges.

Several methods are presented for image recognition that are based on filtering techniques that classify images according to a certain key feature or activity [3]–[5]. A methodology for face recognition has been developed based on feature extraction and recognition [6]. A feature-based approach has implemented SIFT on the COIL dataset that has enhanced the performance of object recognition by using special descriptors such as GLOH, PCA-SIFT, and COLOUR-SURF [7]. Face recognition structure has been applied to obtain the important information of the face features [8]. However, face recognition is a challenge because a face is a composite of small parts such as the mouth, eyes, and nose.

The ability of recognizing facial expressions of people with Down syndrome is discussed in [9]. However, people with Down syndrome cannot recognize unfamiliar or strange faces with different facial expressions and emotions. Down syndrome can be detected by some physical feature variations including up-slanting palpebral fissures, flat facial contours, and epicanthic folds [10]. People with different syndromes can recognize human faces as a whole or as parts, such as the mouth, nose, and eyes, and matching these parts with the original face [11]. In autism, face recognition abilities

The associate editor coordinating the review of this manuscript and approving it for publication was Wu-Shiung Feng.

have shown a variance reliance on feature information [12]. A classification tool is presented for optimizing and extracting Down syndrome features that help in classifying Down syndrome images [13].

The technology of wearable computers is applied to facilitate the interaction between blind people and computers that depends on obtaining the appearance representation of the image components to extract the important features [14]. The image detection is used for helping blind people to understand the images [15] by obtaining the description of the image features and converting it into a voice.

The image recognition technique for wheelchairs is applied to extract and analyze the image of people in wheelchairs efficiently [16]. Analysis of wheelchair images helps medical staff to identify significant features, apply efficient treatment procedures, and achieve high diagnostic performance. A high level of image recognition efficiency could be reached by extracting the most related image features. Since medical images are varied in their content, deep learning techniques are suitable for extracting complex image features. CNNs have been applied in medical image investigations, such as disease discovery, segmentation, classification, and abnormality recognition, using images from different clinical resources. However, the main problem of using CNNs in image recognition is the overfitting of the learning classifier. Applying the dropout procedure with an appropriate dropout level will help in resolving the overfitting problem and enhance the image learning classification.

In addition, a regularization technique [17] is applied to limit the number of parameters from proliferation when the learning classifier requires a large number of parameters to be modified. An image smoothing technique can help in filtering the image by withdrawing the noise from the image [18]. A mix of convolutional neural networks based on the weight of CNN prototypes is applied to identify image content [19]. The weight of the CNN prototype is transformed as a linear regression equation that could be learned by using the ordinary least squares method [20].

Sharing images of people with physical or mental disabilities over cloud nodes is useful for making decisions on medical consultation, diagnosis, and surgeries [21]. Securing disabled people's image mobility between clouds is needed to protect personal information from unauthorized use and to increase the privacy of people with disabilities. The security of image exchange between cloud systems is a challenge, and different security techniques are implemented for image migration between cloud nodes [22], [23]. However, the current techniques need encryption and decryption procedures that might increase the migration time complexity. Labeling MPLS headers [24] is an effective technique that can be used for securing image mobility over cloud nodes.

In this paper, we significantly classify the images of people with physical or mental disabilities, specifically people who are blind, have Down syndrome, or are in wheelchairs, and improve the image classification performance in the cloud system environment. In addition, we introduce a securing mobility technique to promote disabled people's privacy. Figure 1 presents the proposed approach architecture.

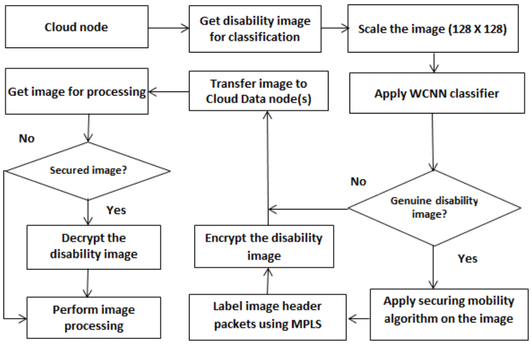

Cloud node
Get disability image for classification
Scale the image (128 X 128)
Get image for processing
Transfer image to Cloud Data node(s)
Apply WCNN classifier
No
Secured image?
Yes
Decrypt the disability image
Perform image processing
No
Genuine disability image?
Encrypt the disability image
Yes
Label image header packets using MPLS
Apply securing mobility algorithm on the image

FIGURE 1. Image Classification and Securing Mobility Architecture.

The contributions of this work are summarized as follows:

- Developing three binary image classifiers based on utilizing convolutional neural networks of wide hidden layers, WCNN, that have more neurons rather than adding more deep hidden layers with fewer neurons. Three different human disabilities are precisely classified as a proof of superiority of the proposed classifiers.
- Improving the image classification performance of people with physical or mental disabilities that in turn resolves the challenges associated with identifying the image features and speeds up the process of determining the particular image files that need to be secured and transmitted.
- Developing an effective technique for securing image mobility in cloud systems based on labeling MPLS headers to keep the bio-information of the people with physical or mental disabilities, support their privacy, and reduce the security costs of the transferred images between cloud nodes.
- Integrating an image classification method of the people with physical or mental disabilities with a securing mobility technique in one approach to achieve high throughput in terms of security classification processing time and IP spoofing detection.

The remainder of the paper is organized as follows: Section II presents the related works on image recognition of human disabilities. The data sets of classifiers are described in Section III. The image classifier architectures of the people with physical or mental disabilities are presented in Section IV. The security of the image mobility of disabled people is illustrated in Section V. The performance evaluation and analysis are presented in Section VI. Finally, the conclusion and future work are presented in Section VII.

II. RELATED WORK

Object recognition is intensively used in many useful systems where a camera records the object mobility in a certain area and a computer control system determines the object identity. Object recognition is worthwhile in monitoring systems, especially for people with disabilities to facilitate their mobility and advise them when critical situations arise. However, the mechanisms of recognizing images of disabled people are not significantly addressed. In this section, we present a literature review on identifying the images of people with disabilities and securing the mobility of images over the cloud.

Face recognition has been identified by important features that are extracted by a descriptor and a graph structure technique WSLGS [25] that is based on the Weber law that depends on selecting an image pixel and its surroundings to generate the important feature information, where the local graph structure converts the pixel from binary to decimal. The principal component analyses technique has been applied to reduce the image dimension, and then the image features are fed into an extreme learning machine that is a feed-forward neural network. Nevertheless, WSLGS is based on the local graph structure technique that causes loss of information.

In a face recognition system [26], an image of a human face has been detected and determined by the Viola-Jones algorithm. The face features are identified by using the principal component analysis that converts the image into face vectors and then calculates the average face vectors such that if the average face vector is greater than a threshold, then the face is recognized; otherwise, the face is considered as unknown. However, the Viola-Jones algorithm is not significant in distinguishing abnormal faces.

The self-organizing map (SOM) technique [27] has been used for decreasing the input image dimensions to support a CNN in extracting image features and simulating face recognition where number of training images is low. Different face features are tested such as the nose width and height, the mouth position, and the chin shape. Nonetheless, it is hard to determine the face feature weights in an SOM, and it assumes the similarity behavior of the face image of nearby pixels, which is not case for the people with disabilities.

A study on face recognition has been discussed in which smart environments and wearable computers are used to establish a human-computer interaction of face and voice [28]. Dimensionality reduction technologies have been applied, such as principal component analysis (PCA), feature histograms, and independent component analysis, that are based on obtaining an appearance representation of many image components to learn about each class and help to achieve the important features. On the other hand, the support vector machine (SVM) learns about the differences between features.

A deep learning method for identifying the visual infirmity of persons is discussed in [29] and is focused on classifying the images of normal people with no sticks and blind people with a white cane. The public architecture of Cipher-10 is used to build a special dataset that contains images from sideways and diagonal perspectives to test only the classification accuracy. However, this method did not consider blind people wearing sunglasses, special black masks that cover only the eye area, the eye shape of blind people, and guiding helpers.

Physical feature variations can indicate Down syndrome at birth or at an older age. The local binary pattern approach has been applied to detect Down syndrome that recognizes facial remarks automatically using a constrained local classifier [30]. Geometric features and anatomical landmarks are extracted and fed to a classifier, such as the SVM, k-NN and random forest [31], which recognizes the features and confirms the possibility of Down syndrome. The important parts of the image can be detected, such as the nose, ear, and mouth, and determined as anatomical landmarks and then plugged into the classifiers to generate the prediction results. However, dealing only with frontal images might not detect Down syndrome perfectly. In addition, the small number of trained images might not accomplish high predictions.

An algorithm of Down syndrome recognition has been proposed based on the local binary pattern LBP [32] that classifies images by calculating the Euclidean and the Manhattan distances [33], extracts important features, and then recognizes the Down syndrome images. However, the training dataset consists only of the Down syndrome faces with no important features such as the ears or neck. In addition, classifying Down syndrome with datasets of a mixture of body positions are not considered.

The facial expressions of people with Down syndrome can be recognized [34]. People with Down syndrome can identify unacquainted or abnormal faces with different facial expressions and feelings. Nonetheless, the prediction results of this technique were not high even though they deal with familiar faces, and the people with Down syndrome could not recognize neither the person nor the emotion.

A study of Down syndrome disability recognition [35] has built a detection model based on a face analogy where distortion near the areas of the mouth and eye are detected and extracted. This model depends on extracting a set of facial key points. However, there are no common facial features that make the model inaccurately identify Down syndrome disability.

In a recent study [36], a method has been proposed based on a geometric descriptor that is used to identify facial features of the person with Down syndrome. However, the geometric descriptors need multiconfirmation techniques to determine the bioactive confirmation molecule that requires geometry optimization and hence increase the time complexity of the Down syndrome detection.

An approach of recognizing disabled people in wheelchairs has been described in [37] where the image features are based on the generic distance invariant. Extracting image features of this type of disability depends on the distance range scanned in order to detect starting points of the object in the image. Nevertheless, the results are nonspecific regarding the people in wheelchairs, the classifier needs more information about the people who need walkers, and the technique is limited by a constraint of a constant distance between the object and the laser camera that scans the image.

A distance-robust wheelchair detector [38] has been introduced that is based on three 2D image processing phases:

preparation, classification, and prediction. This detector considers a tested area nearby each image pixel and calculates the recognized positions in the coordinate plane. The CNN classifies the proposed tested area and predicts the related recognized positions. The voting technique is used for detecting that the image represents real people in wheelchairs or normal walkers. However, a fixed size of the testing area is estimated for determining recognized positions that affects the prediction probability.

Identifying wheelchair users has been presented by Martín-Nieto *et al.* [39]; two different detection models are integrated based on 3 different algorithms, specifically, the DPM part-based model for scanning and the Faster-RCNN and YOLOv3 for the object detection. Each detector model implements a category of different people: the people in wheelchairs and people standing normally. The dataset was recorded by video processing. However, the dataset ground truth was not available and was created manually. In addition, each detector works on disjoint group of people, and normalization for each detector is needed.

Hirunwattanakun *et al.* [40] proposed an image technique for classifying a wheelchair image. The Gaussian mixture model is used to extract a dynamic object from the image background. In addition, the histogram of oriented gradients is used to generate the underlying feature vectors of the dynamic image objects. Moreover, support vector machines are applied to classify a wheelchair image. However, the SVM algorithm needs accurate settings for several key parameters to realize correct image classification. The SVM training time complexity is longer than other classification methods in the literature, which makes it uncompetitive on large datasets.

The need to secure sensitive data migration over cloud systems is discussed in [41]. The application components have to be protected from possible attacks during migration processes between cloud nodes. Ignoring sensitive data security may allow precarious threats on the cloud system and cause loss of critical information that increases risk possibilities. A security risk might occur when different cloud clients share the same component resources.

The security watermarking method has been used for the technique of transmitting medical images, as proposed by Priya and Santhi [42]. The patient data record is embedded within the transmitted image to produce the watermarked medical image for securing the patient information. This medical image is encrypted with a fingerprint scanner that does not consider the physical changes of the patient that may lead to rejection of responses.

A hybrid technique has been proposed by Okediran [43] for medical image security transmission and is based on RC4 and RSA security methods. The RC4 is used to encrypt the image, while the RSA is used for image key encryption. The image encrypted key is hidden in the encrypted image by using a direct sequence and frequency hopping spread spectrum methods. However, RC4 is susceptible to bit-flipping and stream cipher attacks. On the other hand, RSA requires a third party to verify the consistency of the encrypted key that may allow active hackers to intrude the image.

The medical image exchange provides image information remotely that is stored in a digital format [44]. Digital imaging and communication is used in medicine for exchanging patients' image file packets through cloud nodes, where the header part consists of image identification and the data part forms the genuine image. The image file might be secured in the transfer phase between the provider and the receiver clouds. Nevertheless, trust in a third party to protect personal images may affect the privacy of image mobility.

Therefore, it is essential that the personal bio-information, especially the images that represent human disabilities, is recognized precisely and secured against nonauthorized access by people sharing the same cloud resources. Thus, using MPLS header file labels will increase the security of the image mobility between cloud nodes without degrading the mobility performance.

III. WCNN IMAGE CLASSIFIERS DATASETS

The WCNN classifiers require thousands of images per class, heuristically, to generate a significant classification decision. To the best of our knowledge, there is no specific image source for people with physical or mental disabilities such as blind people, people with Down syndrome, and people in wheelchairs; accordingly, we collect our images from Google [45]. However, the collected images were not sufficient for the WCNN classifiers to significantly predict the images that represent the actual disabilities; therefore, different image preprocessing techniques are applied to produce a large number of images from the original ones to enhance the proposed classifiers performance. A zero-phase component analysis [46] is applied in order to transform the processed images into black and white. The image is also sheared with shear intensity [47], where the shear angle is in degrees and the direction is counterclockwise. Moreover, a zooming effect is applied on the image so that it is flipped horizontally, vertically, and rotated randomly within a given range by applying transformations and random rotation functions with Ketkar [48]. The image scaling [49] is applied that enables generating images with different scales, where the image is multiplied by a scaling constant that could generate a new image size. Our original images consist of RGB coefficients in the range 0-255, but such values would take a long time to process due to limited hardware configurations; instead, we target values between 0 and 1 by scaling with a factor of 1/255 to generate image colors that become simpler for analyses.

Each image is downloaded separately, and later the input images are categorized into three different datasets. The first dataset includes the genuine images of people in wheelchairs and the images of people who are sitting on different types of chairs. The second dataset includes the images of blind people directed by walking sticks and the images of normal people who hold several types of sticks, and the third dataset includes the images of people with Down syndrome and the images of normal people in different situations.

A total number of 7214 images were collected and generated for the three datasets that represent image sources of people with physical or mental disabilities of the WCNN classifiers. The dataset images are divided into two categories: 5128 images for training, which are approximately 71% of the total images, and 2086 images for testing that are equivalent to 29%. We consider 50% of the images in each part for people with physical or mental disabilities and another 50% for nondisabled people. The assumption of splitting dataset images into the mentioned ratios leads to the highest significant results of classifying images of people with physical or mental disabilities by wheelchairNet, blindNet, and DownsyndromeNet classifiers.

The image training dataset of each classifier was divided into batches [50] to enable faster training and minimize the memory requirement for classification. Decreasing the batch size will reduce the classifier prediction accuracy, whereas increasing the batch size will increase the needed memory for classification and also cause model overfitting that might not provide sufficient help for identifying image features [51]. Therefore, a compromise on batch size, small to adequately large (10-70), is considered in our classifier architecture to achieve the highest model accuracy in the minimum classification time, and hence enhances the cloud system performance. A randomized set of images was collected at each epoch [52], so some images might be considered in multiple batches because each new batch is analogous to new pictures for the classifier.

Algorithm 1 Convolutional Image Processes

```
Input: Input_Image_Matrix [x,y] from the data set, Filter_Matrix[i,j]
Output: Convolutional_Image_Matrix[i,j]
Begin
Step 1:
for row_count1 = 0 to i do
    for column_count1 = 0 to j do
        Set the Multiplication_Sum = 0
        for row_count2 = 0 to i do
            for column_count2 = 0 to j do
                Multiplication_Sum = Multiplication_Sum + Input_Image_Matrix [row_count1+ row_count2, column_count1+ column_count2] * Filter_Matrix[row_count2, column_count2]
                column_count2 = column_count2 + 1
            end for
            row_count2 = row_count2 + 1
        end for
        column_count1 = column_count1+ 1
    end for
    row_count1 = row_count1 + 1
end for

Step 2:
Convolutional_Image_Matrix [i,j] = Multiplication_Sum
Step 3:
Repeat steps (1-2) for each Convolutional Layer
End
```

IV. DISABLED PEOPLE'S IMAGE CLASSIFICATION

Three WCNN binary classifiers are developed, WheelchairNet, DownsyndromeNet, and BlindNet, to identify the genuine images of blind people, people with Down syndrome and people in wheelchairs from other images of normal people in different situations. The proposed classifiers were trained by the WCNN using Python 3.5.3 [53]. The WCNN classifier is based on convolutional layers [54], max pooling layers [55], flatten layers [56], dense/fully connected layers [57], the dropout technique [58], and the output layer [59].

A. WCNN IMAGE CLASSIFIERS ARCHITECTURE

The WCNN classifiers common architecture is described as follows:

- The input image is formatted in a two-dimensional matrix of 128*128 with RGB channels.
- The formatted image is passed through a convolutional layer that applies 32 feature detectors (filters), each of size 3*3. The output convolutional image matrix is extracted by the convolutional processes with no padding, and these processes are defined in Algorithm 1.
- A max pooling layer is applied on all feature detectors produced from the convolutional layer and designed to reduce spatial image dimensions with a 2*2 pooling filter. The max pooling layer works on the brightest (maximum) pixel value that represents the important image feature, such as the cane, the face of a person with Down syndrome and the wheelchair. The result of the max pooling layer is a two-dimensional image matrix with 32 pooled features that keep the important image features and discard the less important features; the image is then reduced to a half size of the matrix that results from the previous convolutional layer.
- The flatten layer is applied after the last max pooling layer that transforms the two-dimensional matrix of image features into a one-dimensional vector which helps in reducing the image matrix complexity, hence simplifying computations of activation functions.
- The fully connected/dense layer is applied on the generated image feature vector where the activation function ReLU [60] is used due to its high speed of classifying convergence in order to activate a lower number of neurons [61] and simplify the classifier computations. The trained neuron weight is defined based on the following metrics: the input variables of the current layer, the weight of the input variables that affects the output results of the current layer, and the bias

offset value that allows shifting the activation function to the positive or negative direction in order to avoid data set overfitting and best control the neuron activation decision. Let *NWS* represent the trained neuron weight sum; then, *NWS* is calculated as in Equation (1):

$$NWS_{x,l} = \sum_{i=1}^{n} v_{i,l} * w_{i,l} + b_l \tag{1}$$

where $NWS_{x,l}$ represents the weight sum of neuron x in the current layer l, n represents the number of neuron input variables in the current layer l, $v_{i,l}$ represents the neuron input variable value in the current layer l, $w_{i,l}$ represents the weight of the input variable in the current layer l, and b_l represents the bias offset value in the current layer l. Based on the neuron weight sum, the activation function determines the firing/not-firing value according to Equation (2):

$$f(NWS) = \begin{cases} 0 & NWS \leq 0 \\ NWS & NWS > 0 \end{cases} \tag{2}$$

If the activation function firing value is positive, then the neuron is fired; otherwise, the neuron is not fired.

- The dropout technique might be applied in a fully connected layer in order to switch off some neurons randomly at some nodes and help in reducing the classifier over-fitting by decreasing dependent learning between the neurons. This regularization method is based on the dropout probability DP ratio (0-1) that helps in reaching stability during the training phase and maximizing the classifier filtering performance. When a large number of neurons are switched off, a diminished understanding of image features might occur, whereas a lower number of switched off neurons may increase understanding image features. However, decreasing this DP ratio might not prevent the over-fitting; whereas increasing this ratio might increase the chances of under-fitting. Therefore, a compromise on the DP value is considered to avoid both cases of over-fitting and under-fitting. Our experiments show that the DP value between 0.2 and 0.4 could yield full advantage of the filtering capacity. On the other hand, the existence of the active neurons probability ANP that represents the probability of the important image features is computed, as in Equation (3):

$$ANP = 1 - DP \tag{3}$$

- The output layer determines the image class according to the output activation function of the Sigmoid [62] that predicts a binary value. There are two neurons in the output layer representing two classes: an image of people with physical or mental disabilities and an of non-disabled people. Figure 2 summarizes the WCNN classifiers' architecture.

The detailed architecture of each classifier is described in the following subsections

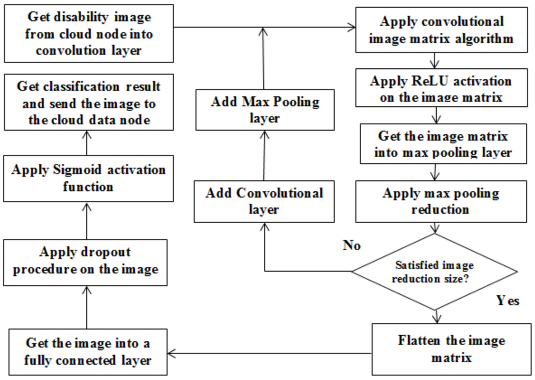

Get disability image from cloud node into convolution layer
Apply convolutional image matrix algorithm
Get classification result and send the image to the cloud data node
Add Max Pooling layer
Apply ReLU activation on the image matrix
Get the image matrix into max pooling layer
Apply Sigmoid activation function
Add Convolutional layer
Apply max pooling reduction
No
Apply dropout procedure on the image
Satisfied image reduction size?
Yes
Get the image into a fully connected layer
Flatten the image matrix

FIGURE 2. WCNN Classifier Architecture.

1) WHEELCHAIRNET ARCHITECTURE

The WheelchairNet classifier architecture produces the maximum significant fittings based on two convolution layers with 32 filters, each of size 3*3, to recognize the features of the people in wheelchairs. Two max pooling layers, each with a 2*2 pooling filter, are applied after convolution layers to simplify the complexity computations of identifying the actual image features of the people in wheelchairs. Then, the flatten layer is applied to simplify the image matrix into a vector to fit in the dense layer. The image vector is passed to a dense layer with 4 neurons that implement the ReLU activation function to decide the active neurons. The dropout technique with a dropout level value of 0.2 is applied to switch off non essential neurons and reduce the overfitting. The output layer is finally applied that uses the sigmoid activation function which determines the prediction decision on whether the input image represents a genuine image of people in wheelchairs or not. Table 1 describes the WheelchairNet classifier architecture that produces the most significant results.

2) BLINDNET ARCHITECTURE

The highest significant fittings of the BlindNet architecture are built on 3 convolution layers with 32 filters, each of size 3*3, to identify the features of the actual blind people. Applying a max pooling layer with a filter of size of 2*2 after each convolution layer will abridge the computations of extracting image features of blind people. The flatten layer is used to convert the image matrix for the dense layer. The extracted features are passed to two dense layers that apply the ReLU activation function which decides the active neurons from the set of input neurons. No dropout is needed for reducing the overfitting in this experiment of the most significant results. One final output layer applies the sigmoid activation function that determines whether the person in the input image is blind or not. Table 2 describes the blindNet classifier architecture that generates the highest prediction results.

3) DOWNSYNDROMENET ARCHITECTURE

The input image from the dataset into the DownsyndromeNet is passed through four convolution layers with 32 filters,

TABLE 1. WheelchairNet Architecture.

Layer Name (order)	Input Size	Output Size	(Kernel) WheelchairNet
Convolution (1)	128 x 128 x 3	126 x 126 x 32	(3 x 3) convolution
Max Pooling (1)	126 x 126 x 32	63 x 63 x 32	(2 x 2) max-pooling
Convolution (2)	63 x 63 x 32	61 x 61 x 32	(3 x 3) convolution
Max Pooling (2)	61 x 61 x 32	30 x 30 x 32	(2 x 2) max-pooling
Flatten	30 x 30 x 32	28800	
Dense	28800	4	4 neurons
Dropout	4	4	0.2 level
Output Dense	4	1	Sigmoid

TABLE 2. BlindNet Architecture.

Layer Name (order)	Input Size	Output Size	(Kernel) BlindNet
Convolution (1)	128 x 128 x 3	126 x 126 x 32	(3 x 3) convolution
Max Pooling (1)	126 x 126 x 32	63 x 63 x 32	(2 x 2) max-pooling
Convolution (2)	63 x 63 x 32	61 x 61 x 32	(3 x 3) convolution
Max Pooling (2)	61 x 61 x 32	30 x 30 x 32	(2 x 2) max-pooling
Convolution (3)	30 x 30 x 32	28 x 28 x 32	(3 x 3) convolution
Max Pooling (3)	28 x 28 x 32	14 x 14 x 32	(2 x 2) max-pooling
Flatten	14 x 14 x 32	6272	
Dense (1)	6272	64	64 neurons
Dense (2)	64	64	64 neurons
Output Dense	64	1	Sigmoid

TABLE 3. DownsyndromeNet Architecture.

Layer Name (order)	Input Size	Output Size	(Kernel) DownsyndromeNet
Convolution (1)	128 x 128 x 3	126 x 126 x 32	(3 x 3) convolution
Max Pooling (1)	126 x 126 x 32	63 x 63 x 32	(2 x 2) max-pooling
Convolution (2)	63 x 63 x 32	61 x 61 x 32	(3 x 3) convolution
Max Pooling (2)	61 x 61 x 32	30 x 30 x 32	(2 x 2) max-pooling
Convolution (3)	30 x 30 x 32	28 x 28 x 32	(3 x 3) convolution
Max Pooling (3)	28 x 28 x 32	14 x 14 x 32	(2 x 2) max-pooling
Convolution (4)	14 x 14 x 32	12 x 12 x 32	(3 x 3) convolution
Max Pooling (4)	12 x 12 x 32	6 x 6 x 32	(2 x 2) max-pooling
Flatten	6 x 6 x 32	1152	
Dense	1152	64	64 neurons
Dropout	64	64	0.4 level
Output Dense	64	1	Sigmoid

each of size 3*3. A max-pooling layer with a filter of size of 2*2 is applied after every convolution layer to reduce computation needed for finding the Down syndrome features. The flatten layer is used to simplify the image matrix to a vector. Two dense layers are applied to determine the resulting neurons by using the ReLU activation function. The dropout method is used with a dropout level value of 0.4 to switch off less important neurons and retain the important features. One output layer is applied that uses the sigmoid activation function to decide whether the image represents an actual Down syndrome person or not. Table 3 describes the DownsyndromeNet classifier architecture that generates superlative results.

B. WCNN IMAGE CLASSIFIER COMPUTATIONS

Numerous experiments were performed on training and testing datasets with different classification hyperparameters in order to identify the proper threshold values and to record the change effects on the prediction results. Several experiments were conducted by the WCNN classifiers based on the following hyperparameters [63]: convolutional layer, max pooling layer, dense layer, neurons in dense layer, feature maps, input, dense, and output activation functions, dropout level value [64], epochs, optimizer [65], and batch size. Such parameters were selected and optimized to enhance the classification predictions. Table 4 summarizes the hyperparameters and their domain values.

TABLE 4. Performance Hyperparameters.

Hyper-parameter	Experimental Domain Values
Convolutional Layer	1- 4
Pooling Layer	Max 2,2
Dense Layers	1-2
Feature maps	32
Neurons in Dense/Hidden Layer	2-256
Input Activation function	ReLU
Output activation function	Sigmoid
Dense activation function	ReLU
Dropout Level	0.0 – 0.5
Epochs	5 - 100
Batch Size	10 - 70
Optimizer	Adam / Adadelta

TABLE 5. WheelchairNet output layers and number of parameters in each layer.

Layer Name	Output Shape	Number of Parameters
Convolution2d$_4$6	(None, 126, 126, 32)	896
Max$_{p}ooling2d_{4}6$	(None, 63, 63, 32)	0
Convolution2d$_4$7	(None, 61, 61, 32)	9248
Max$_{p}ooling2d_{4}7$	(None, 30, 30, 32)	0
Flatten$_1$7	(None, 28800)	0
Dense$_3$8	(None, 4)	115204
Dropout$_1$8	(None, 4)	0
Dense$_3$9	(None, 1)	5
Total Classifier Model Parameters		125353
Trainable Parameters		125353
None-Trainable Parameters		0

TABLE 6. BlindNet output layers and number of parameters in each layer.

Layer Name	Output Shape	Number of Parameters
Convolution2d$_1$34	(None, 126, 126, 32)	896
Max$_{p}ooling2d_{1}34$	(None, 63, 63, 32)	0
Convolution2d$_1$35	(None, 61, 61, 32)	9248
Max$_{p}ooling2d_{1}35$	(None, 30, 30, 32)	0
Convolution2d$_1$36	(None, 28, 28, 32)	9248
Max$_{p}ooling2d_{1}36$	(None, 14, 14, 32)	0
Flatten$_4$8	(None, 6272)	0
Dense$_1$19	(None, 64)	401472
dense$_1$20	(None, 64)	4160
Dense$_1$21	(None, 1)	65
Total Classifier Model Parameters		425089
Trainable Parameters		425089
None-Trainable Parameters		0

The experiments are executed with a small number of batches, increasing the number gradually and recording the prediction. The output image matrix of each layer in each classifier is determined by the number of trainable parameters that are generated from different layers in the WCNN classifier model. In each classifier model, the input and pooling layers have no parameters to learn and no parameter computation is needed as the image is just read in the input layer, and the pooling layer is only reducing the size the image dimensions. In addition, no parameters are learned in the dropout layer as it only used to drop down the neurons that are underweighted. The total number of convolution layer parameters is based on the feature map weights and the bias. The bias is a constant value added to the feature map weights and used to adjust the neuron output to best fit the classifier learning model. In our proposed approach, the bias value 1 is considered in building each classifier model to regulate the activation function decision on firing the neuron to the next layer. On the other hand, the feature map weights are defined by the filter size, input, and output feature maps. Hence, the total number of parameters of the convolution layers in the classifier model is defined as follows: Let c represent the number of convolution layers in the classifier model, x and y represent the filter size in the current convolution layer, n represent the number of input feature maps in the current convolution layer, m represent the number of output feature maps in the current convolution layer, the constant value 1 represent the bias that adjusts the activation function decision in the current convolution layer, and CP represent the total number of parameters of the convolution layers in the classifier model; then, the CP is computed as in Equation (4):

$$CP = \sum_{i=1}^{c} (x_i * y_i * n_i + 1) * m_i \tag{4}$$

In addition, the total number of parameters of the fully connected layers FP is based on the fact that the input neurons of this layer have a single weight to each next fit neuron and computed as in Equation (5)

$$FP = \sum_{i=1}^{f} (n_i + 1) * m_i \tag{5}$$

where f is the number of fully connected layers in the classifier model, n is number of input neurons of the current layer, m is number of output neurons of the current layer, and 1 is the bias value. Finally, the output layer is a typical fully connected layer. The total number of parameters of the output layer OP in the classifier model is defined by the current layer input neurons n, output neurons m, and the bias value. Then, OP is computed as in Equation (6)

$$OP = (n + 1) * m \tag{6}$$

Based on the total number of parameters computed in Equations (4), (5), and (6), the total number of trainable parameters TP in each classifier model is computed as in Equation (7)

$$TP = CP + FP + OP \tag{7}$$

Tables 5-7 describe the output layers shape and number of parameters in each layer of each classifier

TABLE 7. DownsyndromeNet output layers and number of parameters in each layer.

Layer Name	Output Shape	Number of Parameters
Convolution2d$_8$	(None, 126, 126, 32)	896
Max$_{p}ooling2d_{8}$	(None, 63, 63, 32)	0
Convolution2d$_9$	(None, 61, 61, 32)	9248
Max$_{p}ooling2d_{9}$	(None, 30, 30, 32)	0
Convolution2d$_1$0	(None, 28, 28, 32)	9248
Max$_{p}ooling2d_{1}0$	(None, 14, 14, 32)	0
Convolution2d$_1$1	(None, 12, 12, 32)	9248
Max$_{p}ooling2d_{1}0$	(None, 6, 6, 32)	0
Flatten$_4$	(None, 1152)	0
Dense$_7$	(None, 64)	73792
Dropout$_1$	(None, 64)	0
Dense$_8$	(None, 1)	65
Total Classifier Model Parameters		102497
Trainable Parameters		102497
None-Trainable Parameters		0

The speed of our WCNN classifier models is measured by the computations of floating point operations FLOPs it do. In our WCNN classifiers, many layers' computations are dot products of the classifier layer's weights and inputs as described in equation 1. Therefore, a dot product between weights and inputs vectors of size n performs 2n - 1 FLOPs and n - 1 additions.

In the convolutional layers, the FLOPs are computed in terms of feature maps of size h × w × c where h is the feature map height, w is the width, and c is the number of channels, in addition to the square kernel size k. Thus, the number of FLOPs is computed as a dot product of h-out × w-out × k × k × c-in × c-out.

In contrast, in a fully-connected layer that is defined as a vector x consists of all inputs n connected to all outputs in a weight matrix w (n × m) and a vector b consists of the bias values, the FLOPs computations are accomplished by the dot product between the input vector x and one column in the output matrix w. Both matrices have n elements, and this requires n FLOPs. In addition, m dot products should be computed for the weight matrix. However, the bias value does not affect the number of FLOPs because the dot product has one less addition than multiplication, hence adding this bias value will be engrossed in the final dot product operation. Therefore, the dot product of an input vector of length n with a weight matrix w of size n × m to get a vector of length m, executes (2n – 1) × m FLOPs.

In each of the proposed WCNN classifiers, a non-linear activation function, such as a ReLU or a Sigmoid, is followed a classifier layer which consumes time for computing operations. The ReLU is a single operation that is only applied to the layer output. For a fully-connected layer with m output neurons, the ReLU takes m FLOPs, whereas the Sigmoid costs four times the ReLU, since there are four distinct operations for this activation function. Consequently, Sigmoid takes 4 × m FLOPs for the layer output.

The other types of layers in our WCNN classifiers, such as max-pooling are consume time but they don't use dot products. Accordingly, the number of FLOPs of max-pooling layer is not significant and can be ignored compared to the FLOPs numbers of convolution and fully-connected layers when computing the complexity of the WCNN classifiers.

The complexity of the WCNN model is also measured by the amount of memory that is used for reading layer inputs and the number of memory accesses executed for writing layer outputs. The layers learned parameters or weights are stored in main memory. The speed of the WCNN classifier depends on its weights saved in the memory; the less the model has the faster it runs. For example, a fully-connected layer of n input neurons, m outputs and a bias has a total weight of (n + 1) × m retains in the memory. Whereas a convolutional layer has a total weight of k × k × c-in × c-out in addition to the bias values of c-out. The convolutional layers have fewer weights and thus faster than fully-connected layers.

V. SECURING DISABLED PEOPLE'S IMAGE MOBILITY

In this section, we propose a security technique for image mobility that helps in protecting the classified images of people with physical or mental disabilities while transmission occurs between cloud systems nodes, preventing possible threats that might occur during image transmission and avoiding the loss of sensitive images, thus increasing the disabled people's privacy. The proposed WCNN classifiers can significantly determine the genuine images of people with physical or mental disabilities; accordingly, the file header packet information is used to secure the image file by one of three protection levels categorized from simplest to strongest: authentication, confidential, or restricted, depending on the image sensitivity. The image security decision of the people with physical or mental disabilities is described in Algorithm 2.

The protocol field in the IP networking traffic header is used to determine the security type required on the image file. Accordingly, cipher encryption algorithms are applied, such as AES-128 [66] or SHA-256 [67], to protect the image from unauthorized access. We assume that the sender and receiver cloud network infrastructures support Multiprotocol Label Switching MPLS [68] where the image file packet header is used for securing the images' traffic mobility. The MPLS

Algorithm 2 Image Security Decision

```
Input: disabled people's image file packets
Output: Image securing mobility decision
Begin
Step 1: import numpy as np // bind the python import to the local variable np
Step 2: For each epoch in each trail
Step 3: Load test_image =
    image.load_img('image_name',size = (128,128))
Step 4: Transform test_image =
    image.img_to_array(test_image)
Step 5: Expand test_image =
    np.expand_dims(test_image, axis = 0)
Step 6: Customize test_image = test_image/255.0
Step 7: Get image securing mobility prediction result = model.predict_classes(test_image)
Step 8: Apply for training dataset #training_set.class_indices
Step 9: Classify the image for securing mobility need
    if prediction result[0][0] = 0 then
        predictions.append("Securing image mobility is needed")
        send classified image to the cloud Name node
    else
        predictions.append("Securing image mobility is not needed")
    end if
Repeat steps (2-9) for each image in the dataset
```

labels are created from network packet header information, assuming the availability of an underlying network core that supports image packet header labeling. Our security technique utilizes the network gateway for distributing the labeled image to the appropriate destination cloud node(s) for further security processes. The image securing mobility architecture of people with physical or mental disabilities is described in Algorithm 3.

The main idea of the security on image mobility depends on the use of network labels that could filter and categorize the image data packets. The primary network labels are used to help the network receiving node to decide, in a later phase, the security type that is needed on the classified disabled people's image. The use of the MPLS header and labeling distribution protocols help in reducing the burden of security processes on the destination node(s), and hence improves the performance of the image mobility in the cloud systems. The Gateways are responsible for completing and handling the mapping between the destination nodes that control the image data packets arriving from the core network.

The MPLS labels support the traffic separation that is a crucial security feature that prevents mixing the image traffic between several parties who use the same internet provider services. The separation process can be achieved by using the MPLS-VPN capability. This will avoid the overhead effect of further processing and delay, thus enhancing the network performance. In addition, the image traffic is forwarded or switched inside the intranet by using the labels only without using the IP header information of the sharing routers. Such labels prevent attacks such as IP spoofing and denial of service.

Algorithm 3 Securing Image Mobility

```
Input: Classified image file resulting from Algorithm 1
Output: Secured image mobility
Begin
Step 1: Get classified image files traffic from the cloud node(s)
Step 2: For each classified disabled people's image Do:
Step 3: Label image header packets using Multiprotocol Label Switching
Step 4: Label the Gateway number that is used to send the labeled images traffic
Step 5: Label image type (wheelchair, blind, or Down syndrome)
Step 6: Set security service type (confidential, authentication, or restricted)
Step 7: Apply security encryption protocol on the image file according to the security service type
Step 8: Transfer the labeled image packets to the destination cloud node(s)
Step 9: Repeat steps (3-8) for all classified disabled peoples images
End
```

VI. PERFORMANCE EVALUATION AND ANALYSIS

In this work, we aim to provide image classification guidelines to researchers whose future applications might target people with physical or mental disabilities. Our proposed approach is based on building and validating three WCNN classifiers that detect the features of disabilities in a large number of images and determine if each image represents an actual disability of people in wheelchairs, blind people, or people with Down syndrome.

The proposed image classification technique is evaluated by conducting 20 different experiments on the training and testing data sets for each classifier. The classifier experiments were executed by using the OS-X, 3.1 GHz Intel core i5 64 bit CPU and different classification hyperparameters in order to identify the proper threshold values and to record the change effects on the results. We made our code publicly available for regenerating our results https://github.com/ibbu10/Utilizing-Convolutional-Neural-Networks-for-Image-Classification-and-Securing-Mobility-of-People.

To externally validate our approach, we implemented each of the proposed image classifiers and compared the resulting prediction performances with state-of-the-art techniques in the literature. The comparisons are carried out in terms of the most significant prediction performance measures:

recall and precision [69] and [70]. The results of averages of 20 experiments for 100 epochs were considered for validating the image classifier behaviors under different hyperparameter conditions.

We implemented two state-of-the-art Down syndrome disability techniques proposed in [35] and [36] and compared their results against the proposed DownsyndromeNet classifier according to the performance hyperparameters described in Table 4. Figures 3 and 4 show the comparisons of approaches regarding the average prediction results of recall and precision.

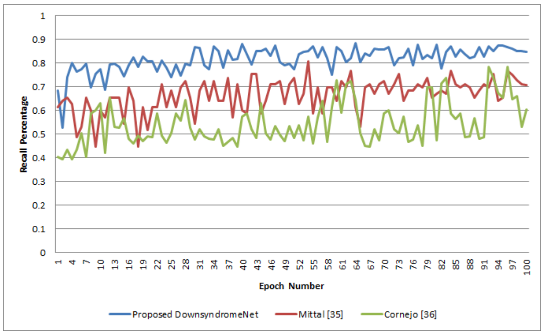


FIGURE 3. **Average recall performance of Down syndrome disability classifiers.**

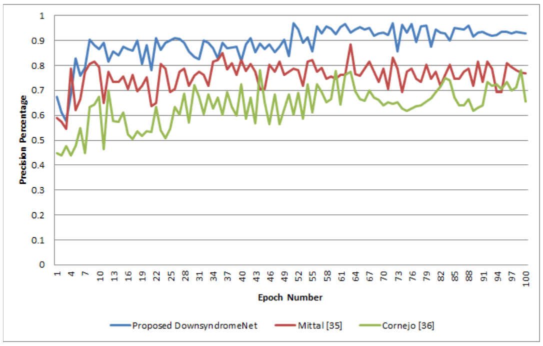


FIGURE 4. **Average precision performance of Down syndrome disability classifiers.**

The results depicted in Figure 3 show that the DownsyndromeNet classifier started at a high average recall percentage of 80% in the first 30 epochs (excluding the startup epochs) and increased up to 86% for the rest of the epochs with small fluctuations that showed the actual image was not identified by the classifier.

On the other hand, the approach proposed in [35] starts at low prediction of 50% but increases slowly up to 70% with fluctuations of more than 15% due to imbalanced tested classes, while the approach proposed in [36] generates less predictions with fluctuations of more than 20% in a large number of epochs due to imbalanced tested classes and fixed batch sizes. Therefore, the results emphasize that the proposed DownsyndromeNet classifier significantly outperforms the comparative approaches.

In Figure 4, the average precision of the proposed DownsyndromeNet classifier started at high rate in the first 50 epochs (excluding the startup epochs) and increased to approximately 93% in the rest of the epochs where the classifier had higher probabilities of classifying actual images when it predicted images as positive. Minor fluctuations occurred when the classifier identified the image incorrectly. For example, one of the symptoms of Down syndrome is a short neck, so when someone looks down, the neck becomes slightly bloated and causes the classifier to consider the image of normal person as having Down syndrome since he or she has a short neck. This will affect the classifier prediction and results in a false classification, consequently reducing the classifier precision performance.

In addition, it can be inferred from Figure 4 that the rate of false positives has been very low throughout all epochs, except in the epochs 1-7 where the precision rate was still increasing. The false negatives show increasing and decreasing rates, but at some epochs, we could also realize that the false negative rate might be lower than the false positive rate, for example, at epoch 40, where the recall is higher than the precision. The overall precision was dominant, indicating that a lower false positive rate rather than a lower false negative rate occurred. Despite the small fluctuation occurrences, the overall precision was shown to increase slightly, but with stability.

In contrast, the approach proposed in [35] starts at low predictions of 50% and then increases slowly up to 75% with fluctuation ranges exceeding 15% due to imbalanced tested classes; however, the approach proposed in [36] generates less predictions due to imbalanced tested classes and fixed batch sizes. Hence, the experimental results confirm that the proposed DownsyndromeNet classifier outperforms the counterpart approaches.

We also implemented two state-of-the-art wheelchair disability classifiers that are proposed in [39] and [40] and compared their results against the proposed WheelchairNet according to the performance hyperparameters described in Table 4. Figures 5 and 6 show the average prediction results in terms of recall and precision, respectively.

Figure 5 shows that the average recall in the WheelchairNet classifier started at a low percentage of 60% due to misclassifying large image groups of people in wheelchairs as nonwheelchairs which indicates a relatively high false negative rate. Then, the WheelchairNet predictions are correctly increased to approximately 80%. Despite small fluctuations in the WheelchairNet experimental results, the recall predictions increased, which means that our classifier predicted less false negatives throughout the experiments; hence, our approach accurately recognizes the disabled people in wheelchairs.

On the other hand, the approach proposed in [39] starts at slightly higher predictions, but the performance increases slowly throughout all epochs due to different image subsets that are treated as different people classes, while the approach proposed in [40] generates less predictions with

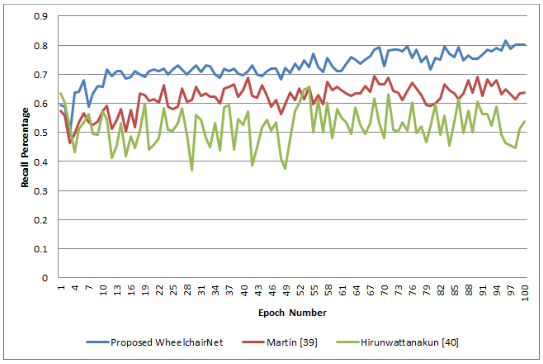


FIGURE 5. Average recall performance of wheelchair disability classifiers.

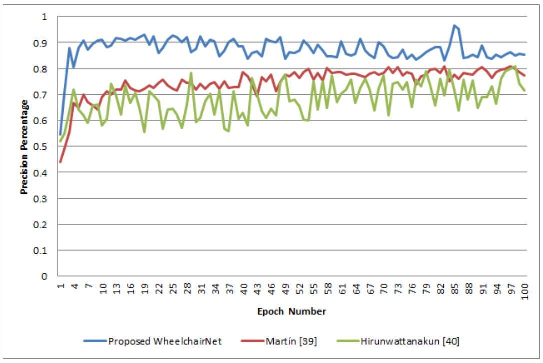


FIGURE 6. Average precision performance of wheelchair disability classifiers.

more fluctuations due to overlapping of the image target classes. Therefore, the results highlight the superiority of the proposed WheelchairNet classifier.

Figure 6 shows that our WheelchairNet classifier achieves much higher precision over all epochs and starts at 80% in the early epochs and increases to approximately 90% at most epochs, which specifies that WheelchairNet initially had a low false positive rate. Despite the lower precision rate at some epochs, it remains high throughout all epochs. Accordingly, the proposed WheelchairNet classifier behaves correctly and can precisely distinguish between the genuine wheelchair images from images of normal people sitting on different types of chairs.

In contrast, the approach described in [39] starts at lower predictions and then increases slowly by less than 10% from epoch 10 onward because of the variety of image subsets, while the other approach in [40] still generates less predictions with more fluctuations because of the target classes overlapping. Therefore, the proposed WheelchairNet classifier outdoes the classifiers in comparison.

In addition, we implemented the state-of-the-art Blind disability classifier proposed in [29]. To the best of our knowledge, there are no other studies on image classification of blind people in the literature; however, there are studies on helping blind people to identify objects or determine some eye diseases and these are not the main objectives of our comparisons. The prediction performance of the proposed BlindNet classifier compared with the approach in [29] is based on the performance hyperparameters described in Table 4. Figures 7 and 8 show the compared average prediction results in terms of recall and precision, respectively.

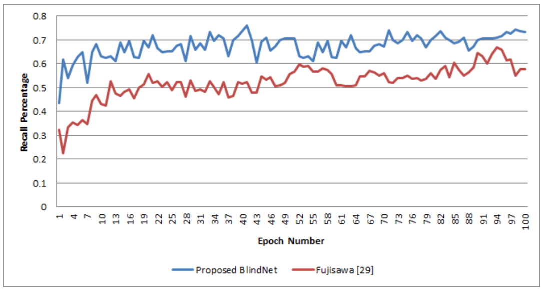


FIGURE 7. Average recall performance of Blind disability classifiers.

Based on the blind disability classifier experiments, it can be inferred from Figure 7 that the recall started at low predictions and increased eventually to approximately 80%. Despite some prediction fluctuations of 3%-10%, the BlindNet predictions generally increased, indicating that the proposed classifier is stable and correctly classifies the actual blind person image from a normal person who might be holding a guidance stick or any walking tool.

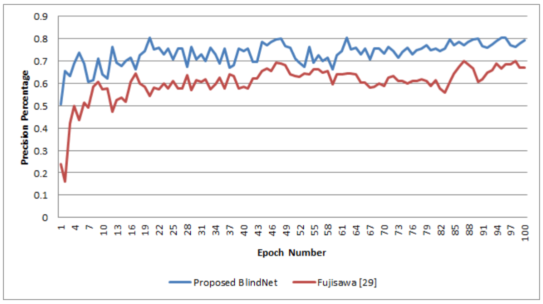


FIGURE 8. Average precision performance of Blind disability classifiers.

On the other hand, the approach proposed in [29] shows unstable behavior of the classifier that generates less predictions and more fluctuations due to the consideration of the image nature regarding perspectives from diagonal and sideways positions. For example, images of healthy people holding white cans or guiding tools lead the classifier to predict false detections, hence decreasing the classifier predictions.

In addition, Figure 8 shows that the proposed BlindNet precisions start at low predictions and then depict an increasing trend that reached a high prediction level of 80% at some epochs. However, small fluctuations that occurred between epochs 50-60 were due to the classifier overfitting. The recall and precision fluctuations indicate there exist increases and decreases in the rate of false negatives and false positives; however, there is a gap or threshold that

TABLE 8. Prediction performance evaluations of image classifiers of people with disabilities.

Image Classifier Approach	People Disability Image Category	Average recall Performance %	Average Precision Performance %
Proposed DownsyndromeNet	Down syndrome	81.67%	88.97%
Mittal [35]	Down syndrome	70.66%	77.56%
Cornejo [36]	Down syndrome	59.43%	68.35%
Proposed WheelchairNet	Wheelchair	76.12%	87.32%
Martín [39]	Wheelchair	65.83%	77.84%
Hirunwattanakun [40]	Wheelchair	58.21%	69.17%
Proposed BlindNet	Blind	71.51%	78.345%
Fujisawa [29]	Blind	56.20%	63.09%

remains at which both false negative and false positive rates could not go above or below due to the rates of recall and precision. In general, the classifier recognizes a large number of images correctly regardless the fluctuations; hence, the BlindNet significantly classifies the blind disability images.

Through experimental tests with the proposed WCNN classifiers, we concluded that the learning trend of the people with disabilities usually increases and reaches the steady state by using a large number of epochs, with small variations, at a high prediction percentage. Table 8 summarizes the comparative prediction performance evaluations of image classifiers of people with disabilities and shows the superiority of the proposed DownsyndromeNet, WheelchairNet, and BlindNet techniques.

Various experiments are conducted to validate the effect of MPLS labels on the image security classification processing time and detecting IP spoofing attacks. The experimental analyses are based on the simulation using the NS-2.35 [71]. The experiments consider several network topologies: LAN, WAN, and MAN with the open shortest path first routing agent protocol and different scenarios of the traffic rate, packet size, and distances between nodes in order to evaluate the image security classification processing time, detecting data classification, and bandwidth overhead. Table 9 depicts the security parameters that are used in the experiments and their domain values.

TABLE 9. Security Simulation Parameters.

Security Parameter	Experimental Domain Values
File Types	Image files
Distance between cloud nodes (Km)	0.1, 1, 10, 100, 1000, 2000
Routing Agents	OSPF
Transmitting Frequency (Pkts/s)	100, 200, 500, 1000
Image Packet size (Kbytes)	0.1, 0.3, 0.6, 1
Network Infrastructure type	LAN,WAN,MAN
Number of cloud nodes	5, 10, 20
Traffic Type	TCP, UDP
Traffic throughput	CBR
Simulation time (seconds)	550

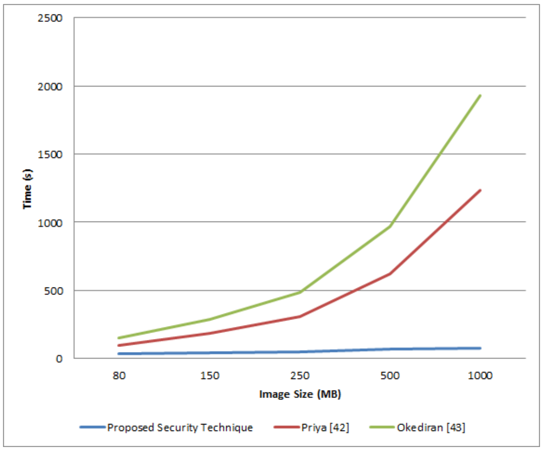


FIGURE 9. Image Security Classification Processing Time.

We implemented the image security techniques proposed by Priya and Santhi [42] and Okediran [43] and compare their performance results with the proposed securing technique in terms of image securing time and IP spoofing performance. Figure 9 shows the security classification processing times of the proposed image security technique against the other techniques in comparison.

The results presented in Figure 9 indicate that more time is required by the two compared approaches due to image encryption, decryption and authentication for verifying the fingerprint in [42], while a weak security method, RC4, is used in [43] to encrypt the image that can be easily broken through. Therefore, the proposed security technique significantly outperforms the counterpart approaches.

In addition, the data threats may occur in the security techniques [42] and [43] during sending and receiving images. The attacker might capture the patient metadata transmitted from the source node to the target node, which makes the data unsecured from intruders. Figure 10 shows

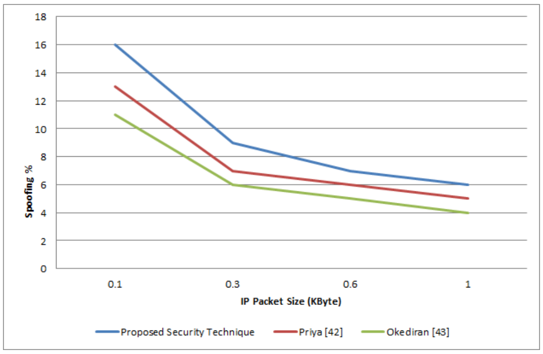


FIGURE 10. IP Spoofing Detection.

the IP spoofing results of the proposed technique against the results of the other techniques in comparison.

The obtained results in Figure 10 specify that more IP spoofing attacks were detected by the proposed image security technique and less by the compared approaches due to weak image encryption and decryption algorithms or unsecured image transmission protocols. The experimental results show the efficiency and superiority of the proposed image securing technique compared with the state-of-the-art techniques [42] and [43].

VII. CONCLUSION

This research presents an integrated image classification and security approach for classifying images of people with physical or mental disabilities, specifically, people in wheelchairs, blind people, and people with Down syndrome, by using convolutional neural networks. This approach is built on wide convolutional neural network classifiers that extract the most significant image features that are used to determine the disabled people's image ingenuity. The proposed image classifiers had a mixture between close ups and full and partial images of their bodies that achieved high accurate classification results of more than 83% in many testing experiments.

The experimental results show that images of people with physical or mental disabilities can be recognized and accurately identified with the proposed classifiers. However, more disabled people's images are needed for the classifier data sets to reduce the prediction fluctuations. In addition, utilizing larger training data sets can potentially improve our classification prediction results. Furthermore, the obtained image securing mobility results show significant processing time improvements in simplifying image security classification by providing labeling to the processed packet traffic.

In future work, we aim to include more images of disabled people for developing one multi-class classification model that identifies if a person is blind from an image where only the person's eyes are detected and distinguishes a person in a wheelchair who cannot walk from an image or a scene of him or her sitting on a normal chair. Additionally, more image security techniques will be investigated to reduce the image security processing time and IP spoofing detection.

REFERENCES

[1] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.

[2] S. Hijazi, R. Kumar, and C. Rowen, "Using convolutional neural networks for image recognition," Cadence Des. Syst., San Jose, CA, USA, Tech. Rep., 2015, pp. 1–12.

[3] P. G. Schyns, "Object recognition: Complexity of recognition strategies," *Current Biol.*, vol. 28, no. 7, pp. R313–R315, Apr. 2018.

[4] S. Soleimanizadeh, D. Mohamad, T. Saba, and J. S. Al-ghamdi, "A taxonomy of 3D occluded objects recognition techniques," *3D Res.*, vol. 7, no. 1, p. 4, Mar. 2016.

[5] D. Kim, O. D. Hilliges, S. Izadi, P. L. Olivier, J. D. J. Shotton, P. Kohli, D. G. Molyneaux, S. E. Hodges, and A. W. Fitzgibbon, "Gesture recognition techniques," U.S. Patent 9 372 544, Jun. 21, 2016.

[6] M. Agarwal, N. Jain, M. M. Kumar, and H. Agrawal, "Face recognition using eigen faces and artificial neural network," *Int. J. Comput. Theory Eng.*, vol. 2, no. 4, p. 624, 2010.

[7] R. Rani, R. Kumar, and A. P. Singh, "A comparative study of object recognition techniques," in *Proc. 7th Int. Conf. Intell. Syst., Modeling Simulation (ISMS)*, Jan. 2016, pp. 151–156.

[8] A. Rikhtegar, M. Pooyan, and M. T. Manzuri-Shalmani, "Genetic algorithm-optimised structure of convolutional neural network for face recognition applications," *IET Comput. Vis.*, vol. 10, no. 6, pp. 559–566, Sep. 2016.

[9] J. G. Wishart and T. Pitcairn, "Recognition of identity and expression in faces by children with down syndrome," *Amer. J. Mental Retardation*, vol. 105, no. 6, pp. 466–479, 2000.

[10] K. Diamandopoulos and J. Green, "Down syndrome: An integrative review," *J. Neonatal Nursing*, vol. 24, no. 5, pp. 235–241, Oct. 2018.

[11] P. Lanz, L. Guevara, M. Poo, E. Smeke, and M. Gallego, "Association between coping behaviors, domestic violence and depression in parents of children with Down's syndrome," *Brit. J. Appl. Sci. Technol.*, vol. 6, no. 1, pp. 8–14, Jan. 2015.

[12] A. Arumugam, K. Raja, M. Venugopalan, B. Chandrasekaran, K. K. Sampath, H. Muthusamy, and N. Shanmugam, "Down syndrome—A narrative review with a focus on anatomical features," *Clin. Anatomy*, vol. 29, no. 5, pp. 568–577, 2016.

[13] A. Catic, L. Gurbeta, A. Kurtovic-Kozaric, S. Mehmedbasic, and A. Badnjevic, "Application of neural networks for classification of Patau, Edwards, Down, Turner and Klinefelter Syndrome based on first trimester maternal serum screening data, ultrasonographic findings and patient demographics," *BMC Med. Genomics*, vol. 11, no. 1, p. 19, Dec. 2018.

[14] A. Bhowmick and S. M. Hazarika, "An insight into assistive technology for the visually impaired and blind people: State-of-the-art and future trends," *J. Multimodal User Interface*, vol. 11, no. 2, pp. 149–172, Jun. 2017.

[15] Y. Wong, J. Lai, S. Ranjit, A. Syafeeza, and N. Hamid, "Convolutional neural network for object detection system for blind people," *J. Telecommun., Electron. Comput. Eng.*, vol. 11, no. 2, pp. 1–6, 2019.

[16] F. de Chaumont, B. Marhic, L. Delahoche, and C. Cauchois, "Generic method for recognition of a wheelchair, even with a low resolution-effective sensor," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, vol. 1, Dec. 2004, pp. 56–60.

[17] E. Phaisangittisagul, "An analysis of the regularization between L2 and dropout in single hidden layer neural network," in *Proc. 7th Int. Conf. Intell. Syst., Modeling Simulation (ISMS)*, Jan. 2016, pp. 174–179.

[18] W. Jifara, F. Jiang, S. Rho, M. Cheng, and S. Liu, "Medical image denoising using convolutional neural network: A residual learning approach," *J. Supercomput.*, vol. 75, no. 2, pp. 704–718, Feb. 2019.

[19] T. Connie, M. Al-Shabi, and M. Goh, "Smart content recognition from images using a mixture of convolutional neural networks," in *IT Convergence and Security*. Singapore: Springer, 2018, pp. 11–18.

[20] S. V. C. de Souza and R. G. Junqueira, "A procedure to assess linearity by ordinary least squares method," *Anal. Chim. Acta*, vol. 552, nos. 1–2, pp. 25–35, Nov. 2005.

[21] A. Umamageswari, M. Ferni Ukrit, and D. G. R. Suresh, "A survey on security in medical image communication," *Int. J. Comput. Appl.*, vol. 30, no. 3, pp. 41–45, Sep. 2011.

[22] B. Madhu, G. Holi, and S. Murthy, "An overview of image security techiques," *Int. J. Comput. Appl.*, vol. 154, no. 6, pp. 37–46, Nov. 2016.

[23] X. Zhang, V. Beeram, I. Bryskin, D. Ceccarelli, and O. Gonzalez de Dios, Eds., *Network-Assigned Upstream Label*, document RFC 8359, Mar. 2018, doi: 10.17487/RFC8359.

[24] G. Murugesan and T. L. Cignetti, "Method and apparatus securing traffic over MPLS networks," U.S. Patent Appl. 14/254 144, Dec. 4, 2014.

[25] J. Yang, L. Zhang, Y. Wang, T. Zhao, W. Sun, and D. S. Park, "Face recognition based on Weber symmetrical local graph structure," *TIIS*, vol. 12, no. 4, pp. 1748–1759, 2018.

[26] M. Sahani, S. Subudhi, and M. N. Mohanty, "Design of face recognition based embedded home security system," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 4, pp. 1751–1767, Apr. 2016, doi: 10.3837/tiis.2016.04.016.

[27] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[28] A. Pentland and T. Choudhury, "Face recognition for smart environments," *Computer*, vol. 33, no. 2, pp. 50–55, 2000.

[29] S. Fujisawa, R. Mandai, R. Kurozumi, S.-I. Ito, and K. Sato, "Identification of visually impaired person with deep learning," in *Proc. Int. Conf. Intell. Hum. Syst. Integr.* Cham, Switzerland: Springer, 2018, pp. 601–607.

[30] Q. Zhao, K. Rosenbaum, K. Okada, D. J. Zand, R. Sze, M. Summar, and M. G. Linguraru, ''Automated down syndrome detection using facial photographs,'' in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 3670–3673.

[31] P. Thanh Noi and M. Kappas, ''Comparison of random forest, K-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery,'' *Sensors*, vol. 18, no. 2, p. 18, Dec. 2017.

[32] K. Burçin and N. V. Vasif, ''Down syndrome recognition using local binary patterns and statistical evaluation of the system,'' *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8690–8695, Jul. 2011.

[33] R.-T. Mora-Garcia, P. Marti-Ciriquian, R. Perez-Sanchez, and M. F. Cespedes-Lopez, ''A comparative analysis of manhattan, Euclidean and network distances. why are network distances more useful to urban professionals?'' in *Proc. 18th Int. Multidisciplinary Sci. GeoConf. SGEM, Informat., Geoinform. Remote Sens.*, Jun. 2018, pp. 3–10.

[34] F. Carvajal, C. Fernández-Alcaraz, M. Rueda, and L. Sarrión, ''Processing of facial expressions of emotions by adults with down syndrome and moderate intellectual disability,'' *Res. Develop. Disabilities*, vol. 33, no. 3, pp. 783–790, May 2012.

[35] A. Mittal, H. Gaur, and M. Mishra, ''Detection of down syndrome using deep facial recognition,'' in *Proc. 3rd Int. Conf. Comput. Vis. Image Process.* Singapore: Springer, 2020, pp. 119–130.

[36] J. Y. R. Cornejo, H. Pedrini, A. Machado-Lima, and F. D. L. dos Santos Nunes, ''Down syndrome detection based on facial features using a geometric descriptor,'' *J. Med. Imag.*, vol. 4, no. 4, 2017, Art. no. 044008.

[37] C. Weinrich, T. Wengefeld, M. Volkhardt, A. Scheidig, and H.-M. Gross, ''Generic distance-invariant features for detecting people with walking aid in 2d laser range data,'' in *Intelligent Autonomous Systems*. Cham, Switzerland: Springer, 2016, pp. 735–747.

[38] L. Beyer, A. Hermans, and B. Leibe, ''DROW: Real-time deep learning-based wheelchair detection in 2-D range data,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 585–592, Apr. 2017.

[39] R. Martín-Nieto, A. García-Martín, and J. M. Martínez, ''Incorporating wheelchair users in people detection,'' *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 14109–14127, Jun. 2019.

[40] S. Hirunwattanakun, C. Chianrabutra, T. Charoenpong, and T. Chanwimalueng, ''A Robus method for wheelchair detection: A combination of the Gaussian mixture models and histogram of oriented gradients,'' in *Proc. 1st Int. Symp. Instrum., Control, Artif. Intell., Robot. (ICA-SYMP)*, Jan. 2019, pp. 57–60.

[41] N. Sighom, J. Raphael, P. Zhang, and L. You, ''Security enhancement for data migration in the cloud,'' *Future Internet*, vol. 9, no. 3, p. 23, 2017.

[42] S. Priya and B. Santhi, ''A novel visual medical image encryption for secure transmission of authenticated watermarked medical images,'' *Mobile Netw. Appl.*, pp. 1–8, Feb. 2019, doi: 10.1007/s11036-019-01213-x.

[43] O. O. Okediran, ''A hybrid cryptosystem and watermarking for secure medical image transmission,'' *Asian J. Res. Comput. Sci.*, vol. 5, no. 1, pp. 1–14, Jan. 2020.

[44] S. G. Shini, T. Thomas, and K. Chithraranjan, ''Cloud based medical image exchange-security challenges,'' *Procedia Eng.*, vol. 38, pp. 3454–3461, Jan. 2012.

[45] Google. *Images of People On Wheelchair, Blind People, and People With Down Syndrome*. Accessed: Feb. 2020. [Online]. Available: https://www.google.jo/search?q=images

[46] A. Kessy, A. Lewin, and K. Strimmer, ''Optimal whitening and decorrelation,'' *Amer. Statistician*, vol. 72, no. 4, pp. 309–314, Oct. 2018.

[47] Keras. *Image Data Preprocessing*. Accessed: Feb. 2020. [Online]. Available: https://keras.io/preprocessing/image/

[48] N. Ketkar, ''Introduction to PyTorch,'' in *Deep Learning With PyThon*. Berkeley, CA, USA: Apress, 2017, pp. 195–208.

[49] N. Jiang and L. Wang, ''Quantum image scaling using nearest neighbor interpolation,'' *Quantum Inf. Process.*, vol. 14, no. 5, pp. 1559–1571, May 2015.

[50] A. Devarakonda, M. Naumov, and M. Garland, ''AdaBatch: Adaptive batch sizes for training deep neural networks,'' 2017, *arXiv:1712.02029*. [Online]. Available: http://arxiv.org/abs/1712.02029

[51] Y.-N. Chen, C.-C. Han, C.-T. Wang, B.-S. Jeng, and K.-C. Fan, ''A CNN-based face detector with a simple feature map and a coarse-to-fine classifier–withdrawn,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Mar. 12, 2009.

[52] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, ''On deep learning-based channel decoding,'' in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[53] LearnPython. *LearnPython Advanced Tutorials*. Accessed: Feb. 2020. [Online]. Available: https://www.learnpython.org/

[54] Q. V. Le, ''A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks,'' in *Proc. Google Brain*, 2015, pp. 1–20.

[55] N. Murray and F. Perronnin, ''Generalized max pooling,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2473–2480.

[56] J. Jin, A. Dundar, and E. Culurciello, ''Flattened convolutional neural networks for feedforward acceleration,'' 2014, *arXiv:1412.5474*. [Online]. Available: http://arxiv.org/abs/1412.5474

[57] S. H. Bae, I. Choi, and N. S. Kim, ''Acoustic scene classification using parallel combination of LSTM and CNN,'' in *Proc. Detection Classification Acoust. Scenes Events Workshop (DCASE)*, 2016, pp. 11–15.

[58] H. Wu and X. Gu, ''Towards dropout training for convolutional neural networks,'' *Neural Netw.*, vol. 71, pp. 1–10, Nov. 2015.

[59] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, ''Ordinal regression with multiple output CNN for age estimation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4920–4928.

[60] A. Fred Agarap, ''Deep learning using rectified linear units (ReLU),'' 2018, *arXiv:1803.08375*. [Online]. Available: http://arxiv.org/abs/1803.08375

[61] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, ''Learning activation functions to improve deep neural networks,'' 2014, *arXiv:1412.6830*. [Online]. Available: http://arxiv.org/abs/1412.6830

[62] T. M. Jamel and B. M. Khammas, ''Implementation of a sigmoid activation function for neural network using FPGA,'' in *Proc. 13th Sci. Conf. Al-Ma'moon Univ. College*, 2012, pp. 1589–1591.

[63] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, ''Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition,'' *IET Intell. Transp. Syst.*, vol. 12, no. 8, pp. 939–946, Oct. 2018.

[64] S. Cai, Y. Shu, G. Chen, B. Chin Ooi, W. Wang, and M. Zhang, ''Effective and efficient dropout for deep convolutional neural networks,'' 2019, *arXiv:1904.03392*. [Online]. Available: http://arxiv.org/abs/1904.03392

[65] F. Xiong, Y. Xiao, Z. Cao, K. Gong, Z. Fang, and J. T. Zhou, ''Towards good practices on building effective CNN baseline model for person re-identification,'' 2018, *arXiv:1807.11042*. [Online]. Available: http://arxiv.org/abs/1807.11042

[66] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, ''A new chaos-based fast image encryption algorithm,'' *Appl. Soft Comput.*, vol. 11, no. 1, pp. 514–522, Jan. 2011.

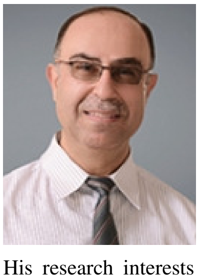
[67] A. W. Appel, ''Verification of a cryptographic primitive: Sha-256,'' *ACM Trans. Program. Lang. Syst. (TOPLAS)*, vol. 37, no. 2, pp. 1–31, 2015.

[68] M. A. Panhwar, K. A. Memon, A. Abro, D. Zhongliang, S. A. Khuhro, and Z. Ali, ''Efficient approach for optimization in traffic engineering for multiprotocol label switching,'' in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2019, pp. 1–7.

[69] S. Visa, B. Ramsay, A. Ralescu, and E. Van Der Knaap, ''Confusion matrix-based feature selection,'' in *Proc. MAICS*, vol. 710, 2011, pp. 120–127.

[70] K. Oksuz, B. Can Cam, E. Akbas, and S. Kalkan, ''Localization recall precision (LRP): A new performance metric for object detection,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 504–519.

[71] M. L. Rajaram, E. Kougianos, S. P. Mohanty, and U. Choppali, ''Wireless sensor network simulation frameworks: A tutorial review: MATLAB/simulink bests the rest,'' *IEEE Consum. Electron. Mag.*, vol. 5, no. 2, pp. 63–69, Apr. 2016.



ISMAIL HABABEH (Member, IEEE) received the B.Sc. degree in computer science from The University of Jordan, Jordan, the M.S. degree in computer science from Western Michigan University, USA, and the Ph.D. degree in computer science from Leeds Beckett University, U.K. He is currently an Associate Professor with the Computer Science Department, German Jordanian University. He has delivers invited talks and keynotes in many local and international forums. His research interests include big data security, neural networks, image processing, and biomedical systems performance. He is a member of ACM. He has served as an organizer, a technical program committee member, and a Reviewer for many international conferences and journals.

IBRAHIM MAHAMEED received the B.S. degree in computer science from German Jordanian University, in 2019. He is currently a Research Assistant with the Computer Science Department, German Jordanian University. His research interests include deep learning technologies, handicapped people identification, vehicle trajectory prediction, human-trajectory prediction, vehicle lane change identification, big data analysis, and visualization tools development.

ABDELHADI A. ABDELHADI received the B.S. degree in computer science from German Jordanian University, in January 2020. He is currently a Research Assistant with the Computer Science Department, German Jordanian University. His research interests include machine learning, medical image processing, natural language processing, and computer vision.

AHMAD BARGHASH received the bachelor's degree in computer engineering, in 2005, the master's degree in bioinformatics (statistics of biological data), and the Ph.D. (Dr.Ing.) degree in informatics. For a period of three years, he was an IT Specialist with German Jordanian University, where he has established a thorough research program analyzing high-throughput datasets in statistical environment of R-Cran, in 2015. He has several ongoing projects in collaboration with companies in Jordan in augmented and virtual reality aiming to improve the learning process in schools. In 2019, he was an Associate Professor. On the administrative side, he has been the Director of the Admission and Registration with German Jordanian University, since 2015.