Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Deep Learning Strategy for Braille Character Recognition

Tasleem Kausar[1], Muhammad Sajjad[1], Adeeba Kausar[3], Yun Lu[2*], Muhammad Wasif[4] and M. Adnan Ashraf[1]

[1]Mirpur Institute of Technology, Mirpur University of Science and Technology, Mirpur, 10250, AJK, Pakistan
[2]School of Computer Science and Engineering, Huizhou University, Huizhou, Guangdong 516007, China
[3]Department of Computer Science and Information Technology, University of Narowal, Punjab, Pakistan
[4]Department of Electrical Engineering, University of Gujrat, Punjab, Pakistan

Corresponding author: Yun Lu (luyun_hit@163.com)

This research is supported in part by the joint fund of basic and applied basic research fund of Guangdong province under grant number 2020A1515110498.

ABSTRACT. People with vision impairment use Braille language for reading, writing, and communication. The basic structure of the Braille language consists of six dots arranged in three rows and two column cells, which are identified by visually impaired people using finger touch. However, it is difficult to memorize the pattern of dots that form the Braille characters. This research presents a novel approach for automatic Braille characters recognition. The designed approach works in two main stages. In first stage, image alignment & enhancement are performed using several image preprocessing techniques. In second stage, character recognition is performed with a proposed lightweight convolution neural network (CNN). As CNN shows promise for accurate recognition of optical characters. Therefore, we adopted several recently proposed state-of-the-art CNN networks for Braille characters' recognition. To make the networks light and improve their recognition performance, we proposed a strategy by replacing few modules in the original CNNs with an inverted residual block (IRB) module with less computational cost. The novelty of this work lies in CNN model design and output performance. We executed the effectiveness of the designed setup through experiments on two different publicly available benchmark Braille datasets obtained from visually impaired people. On the English Braille and Chinese double-sided Braille image (DSBI) datasets, the proposed model shows a prediction accuracy of 95.2% and 98.3%, respectively. The reported test time of model is about 0.01s for English and 0.03s DSBI Braille images. In comparison to state-of-the-art, designed method is robust, effective, and capable to identify the Braille characters efficiently. In future, the functional performance of the proposed Braille recognition scheme will be tested through accessible user interfaces.

INDEX TERMS Braille Images, Image Alignment, Principal Component Analysis, Wiener Filtering, Convolution Neural Networks, Inverted Residual Block.

I. INTRODUCTION

According to World Health Organization (WHO), about 2.2 billion people in world are blind or visually impaired [1]. It is hard for these visually impaired people to read and write text, so they use Braille, a system of raised dots that can be read by the sense of finger touch [2]. The basic structure of Braille system is matrix of six dots aligned in 3x2 order as shown in Figure 1a. Each character in a Braille cell is formed by arrangement of these six dots in a special manner. Therefore, a dot may be raised at any combination for the six positions hence, in total 64 combinations are available ($2^{\wedge 6} = 64$). In Braille, every character is identified by pattern formed by the dots that are raised in cell, The codes of Braille characters, alphabets, and symbols formed through different combinations of Braille dots are shown in Figure 1b. Due to its effectiveness, Braille system is used worldwide by visually impaired for written communication. However, a lot of people, especially ordinary people are not able to identify the Braille characters.

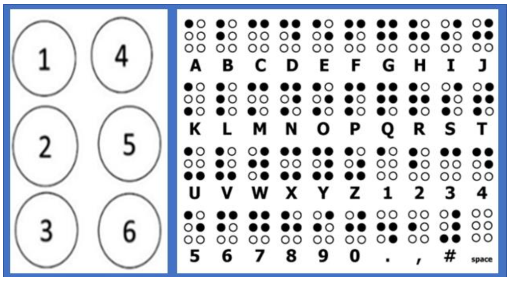
Recently, deep learning has successfully made major advances in various domains [3][4][5][6]. The deep learning methods particularly convolutional neural networks (CNNs) have dramatically improved the state-of-the-art in natural image classification [7][8][9], object detection [10][11] [12], image segmentation [13][14][15], speech recognition [16][17], and medical image analysis [18][19][20][21]. Although deep learning techniques have made tremendous advances in image recognition and achieved high performance results, however, deep learning is sparsely used for Braille character recognition. In this modern era, there is also growing interest in methods exploration for Braille character recognition [22][23][24]. However, there are a few papers on the use of deep models for Braille recognition. To bridge differences between blind people and ordinary people and facilitate the blind people to quickly read Braille language, work is carried out for automatic Braille recognition [25][26][27][28]. Furthermore, in [29], the authors have used convolutional neural network (CNN) and radio character segmentation algorithm (RCSA) to recognize Braille characters and convert them into English language. They claimed a high accuracy of their proposed method. A Chinese character recognition model is proposed in [30] that obtained an accuracy of 94.42%. The proposed method in [30] is extended to read the characters in natural speech. A combination of conventional sequence mapping method and deep learning method has been used in [31] to convert Braille characters into Hindi language, where output was generated as a speech. In [32], the authors used YOLO for real time Braille character recognition. In [33], a method so-called novel technique has been proposed which translates mandarin Braille words to Chinese characters using the N-Best algorithm. The authors claimed that the proposed technique obtained an overall 94.38% translation precision.

In [34], authors used a deep learning strategy for Braille music recognition from Braille music images. They applied few image preprocessing operations to increase the model performance. Another deep learning technique is used in [35] to help blind people to read the Braille characters. They compared the results with classical machine learning techniques such as Naïve Bayes (NB), Decision Trees (DT), SVM, and KNN.

There is almost no open source dataset to carry out a fair comparison between the different algorithms. The lack of publicly available datasets also inhibits the training of deep neural networks to evaluate the accuracy of recognition algorithms. As the performance values provided in above mentioned works are measured on proprietary datasets, Therefore, in Braille image recognition, it is impossible to have a fair comparison of proposed work with previous works. Li et al. [36] proposed a deep semantic segmentation framework named BraUNet. The BraUNet is based on standard UNet model [37] in which they used auxiliary learning strategy and post processing strategy to improve the recognition performance. Li et al. [38] also proposed another Braille recognition technique named TS-OBR that used a traditional machine learning cascaded classifier. In [28], authors used an object detection network named as RetinaNet [39] for optical Braille character recognition. All the models of U-Net, BraUNet, TS-OBR, and RetinaNet are trained only on the DSBI datasets [40]. DSBI dataset [40] is the only publicly available dataset with labeled Braille text images, and we would like to express our deep appreciation to its creators. We evaluated model performance on the two publicly available English Braille datasets [41] and Chinese double-sided Braille image dataset [40].

Early deep CNN models e.g., AlexNet [42] and VGGNet [43] mainly consist of the convolution layers and fully connected layers with strict rules about the resolution of input images. This is because, these networks are mainly composed of two types of layers i.e., convolution layer, and fully connected layer. The convolution layer can accept an arbitrary size input image to generate output feature map of any size. The output feature maps of convolution layer are used as an input to fully connected layer. Essentially fully connected layer accepts fixed size input, therefore, the fixed-size constraint is imposed by fully connected layer. Current CNNs (e.g., Inception [44], ResNet [45], and DenseNet [46]) contain the fully convolution layer, however they also adopt global average pooling (GAP) layer [47]. This layer solves the problem of a large number of parameters, so network computations are considerably reduced. Meanwhile, the GAP layer squeezed the output feature maps by taking the average across height and width dimensions to realize image input of any size. Recently, another model called InceptionV3 [44] is designed using small convolution kernels. In this model, large convolution kernel is factorized into small kernels where depth and width are increased by removing unnecessary layers. These modifications considerably reduce the parameters and computational costs of InceptionV3. Meanwhile, an increase in network depth and width primarily improves the traditional convolution layer in the network. The DenseNet [46] used dense connection in which input of each layer uses output from previous layers. These patterns of dense connections alleviate the gradient vanishing problem. As a result, the output features are strengthened which leads to better performance with fewer parameters. Based on the advantages of the DenseNet network, we used it as the backbone network of our final proposed deep CNN model.

Moreover, automatic recognition of characters from Braille images has some limitations. Images captured with portable camera devices and scanners are usually of low contrast, visually disturbed, misaligned, have illumination variation, and are contaminated with nonregular noises. These input image distortions can affect the detection performance of deep CNN models and make the automatic recognition of Braille characters challenging. It is, therefore, necessary to process and align them in standard orientation before feeding them to CNN models. Image preprocessing is important such that deep models can recognize them correctly. It is also a promising solution to obtain a better regularization in deep CNNs.



(a) (b)

FIGURE 1. (a) Braille dot matrix (b) Alphabets using combinations of Braille dots.

In literature, several image preprocessing techniques such as image filtering, histogram equalization [48], and chromatic adaptation [49] have been applied to increase image clarity. The authors in [50] discussed various image preprocessing methods that are used to develop the security systems using face, fingerprint, and iris biometric features of humans. Similarly, a lot of work has been done on image alignment. In [51] the authors have proposed a principal component analysis (PCA) based image alignment algorithm. In [52], two stage image alignment scheme has been proposed for image stitching using line-guided local warping. In [53] a method for joint face alignment has been proposed and compared with existing techniques. They considered rigid variations of faces and nonrigid distortions in process of joint image alignment. In [54], Likassa et al. proposed a new image alignment algorithm to handle the impact of outliers and heavy sparse noises on a set of linearly correlated data using affine transformations and Frobenius & norms. Recently, Huang et al. [55] have proposed a typical low-rank pairwise alignment bilinear classification network (LRPABN). In the designed model, they used a feature alignment layer to match the image features with the query ones. In [56], an optimization framework is proposed that uses fiducial markers placed in the scene, reducing visual artifacts caused by misalignments.

There are two main challenges in the use of deep CNNs in image recognition: (i) How the accuracy can be further increased? and (ii) How the network computational cost can be reduced?. In this paper, we have addressed the Braille character recognition problem as an image classification problem and established a more effective and suitable CNN based algorithm for Braille character recognition. The high computational complexity of previously proposed deep CNNs was undesirable for real time applications. Unlike previous approaches, we focused on network computational cost issue and proposed a new approach for Braille character recognition that uses the most appropriate image preprocessing techniques and lightweight deep convolution neural network. To solve network computational cost issue, a dynamic type CNN modification technique using the inverted residual block (IRB) module [57] is employed in original CNN model. We tested the performance of proposed method using different widely used state-of-the-art networks [43][45][46][44] as a backbone of modified model. As far as our knowledge goes, our work is the first to successfully perform Braille character recognition using a lightweight deep CNN model. The experimental analysis shows that the suggested technique not only improves recognition accuracy but also reduces the model parameters and computational cost. The main contributions of our work are discussed as follows:

- A novel approach for automatic Braille character recognition is proposed. The proposed approach is based on combination of image preprocessing techniques and an advanced lightweight convolution neural network model.
- The fast lightweight CNN network is designed using an IRB module having less computational cost. Different effective strategies are also applied to improve network efficiency.
- The proposed network works effectively on two types of Braille datasets and outperforms the state-of-the-art in terms of prediction accuracy and recognition speed. On the English Braille and DSBI datasets, proposed approach shows a prediction accuracy of 95.2% and 98.3%, respectively. The reported test time of proposed model on English and Chinese datasets is about 0.01s and 0.03s, respectively.

The paper is further organized as follows; in Section I, the image preprocessing steps are discussed. In Section II, the technical description of proposed method is given. Experimental setup and results are described in Section III. Section IV concludes the paper.

II. METHODOLOGY

A. PREPROCESSING

This Braille character recognition scheme is composed of two main stages. In the first stage, image alignment & enhancement is performed using a series of image preprocessing techniques and then in next stage, characters' recognition is performed with proposed convolution neural network. The flow diagram of Braille character recognition is shown in Figure 2. The preprocessing techniques used for image alignment and enhancement are discussed in the following section.

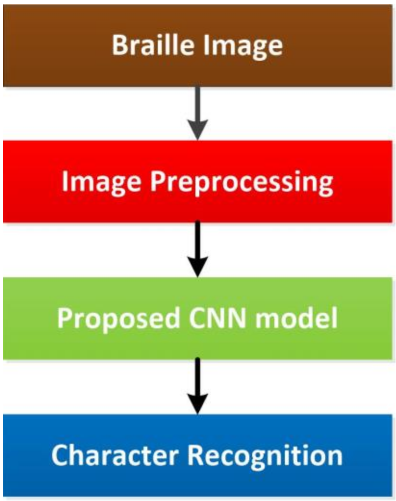


FIGURE 2. Flow diagram of Braille character recognition process.

1) IMAGE ALIGNMENT WITH PRINCIPAL COMPONENT ANALYSIS

The Braille images obtained with scanners and camera devices usually are geometrically disturbed. The Braille images consist of six dots, aligned in 3x2 matrix that are identified by visually impaired people using finger touch. If the image is not in an accurate position, the visually impaired people encounter a problem in prediction of characters. Besides, performance of the automatic Braille character recognition systems trained on geometrically disturbed images is also remarkably affected by image rotation problems. To recognize such images correctly, it is important to resolve the geometric misalignment and align images in a proper position. In this study, we achieved the image alignment using principal component analysis algorithm [51]. Image alignment with PCA [51] is described as following steps:

Steps 1: Given the input image, the coordinates (columns and rows) of 1-valued pixels are extracted in single matrix Z as:

$$Z = \begin{bmatrix} c_1 & r_1 \\ c_2 & r_2 \\ . & . \\ . & . \\ c_N & r_N \end{bmatrix} \quad (1)$$

where r, c, and N show row, column, and total number of pixels in the given input image, respectively. In PCA algorithm, computing the mean and covariance matrix is essential to find the first component which represents the direction of maximum variance of data spread. Mean can be computed as:

$$m_z = \frac{1}{N}\sum_{k=1}^{N} Z_k \quad (2)$$

The covariance is general 2×2 matrix can be computed as:

$$c_z = \frac{1}{N-1}\sum_{k=1}^{N}(Z_k - m_z)(Z_k - m_z)^T \quad (3)$$

Eigenvalues and eigenvectors indicate the magnitude and direction of variances, respectively. Eigenvalue problem can be solved as:

$$(c_z - \lambda I)e = 0 \quad (4)$$

where e, λ and I represent eigenvalue, eigenvector, and identity matrix, respectively. To find solution for e in Eq. (4), $(c_z - \lambda I)$ must be a nonsingular matrix as:

$$\det(c_z - \lambda I) = 0 \quad (5)$$

The solution of determinant gives a second-degree equation as:

$$\lambda^2 - \lambda(x+y) + (xy - b^2) = 0 \quad (6)$$

Eigenvalues can be computed by solving the quadratic formula of second-degree as:

$$\lambda_1, \lambda_2 = \frac{tr(c_z) \pm \sqrt{\{tr(c_z)\}^2 - 4|c_z|}}{2} \quad (7)$$

$$tr(c_z) = x + y \ \ and \ \ |c_x| = xy - b^2 \quad (8)$$

Eigenvectors (2×1 vector each) of covariance matrix c_z corresponding to eigenvalues which can be computed as:

$$u = \frac{1}{\sqrt{b^2 + (\lambda_1 - x)^2}}\begin{bmatrix} b \\ \lambda_1 - x \end{bmatrix} \quad (9)$$

$$v = \frac{1}{\sqrt{b^2 + (\lambda_2 - x)^2}}\begin{bmatrix} b \\ \lambda_2 - x \end{bmatrix} \quad (10)$$

Let X is a matrix whose columns $u = [x_{11}, x_{21}]^T$ and $v = [x_{21}, x_{22}]^T$ formed first and second eigenvectors, respectively. The first vector corresponds to largest eigenvalue and second correspond to smallest eigenvalue as:

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \quad (11)$$

Step2: Using the matrix X, rotation angle of any given misaligned image can be computed as:

$$\theta = \cos^{-1}\left(\frac{tr(X)}{2}\right) \quad (12)$$

where $tr(X) = x_{11} + y_{22}$. To find the standard orientation of the original image, the rotation angle is computed as:

$$\text{Rotation} = \left\{\begin{matrix} \text{if } \theta \neq 0 \text{ rotation yes} \\ \text{otherwise No} \end{matrix}\right\} \quad (13)$$

where θ denotes the true side of the rotated object.

Step 3: The θ' is desired angle to convert the rotated image into its standard orientation, can be computed as:

$$\theta' = -(x_{11} \times x_{21})\theta \quad (14)$$

PCA aligned the images spatially in the direction of their principal spread. The misaligned images are converted back to true standard orientation by rotating them with the angle calculated in Eq. (14) using bilinear interpolation. Results of image alignment are shown in Figure 3.

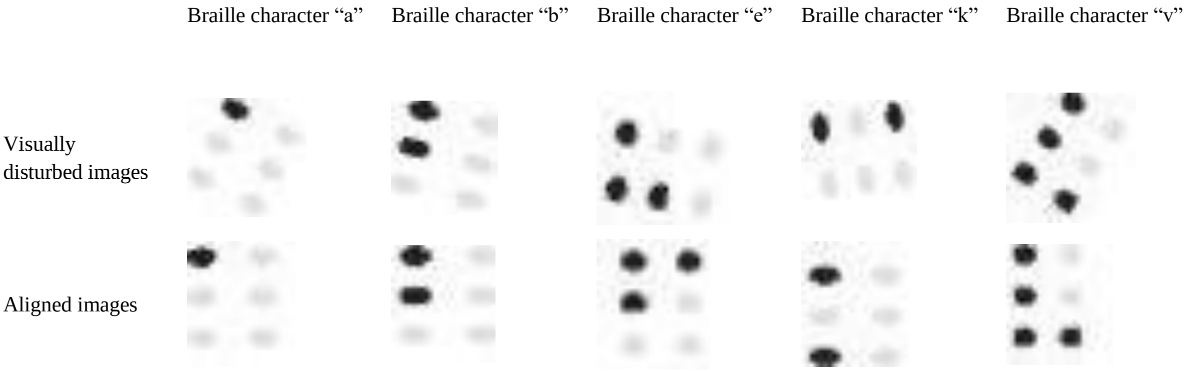


FIGURE 3. Standard orientation results on English Braille Character Dataset [41]**. Visually disturbed Braille images and corresponding aligned images; images are rotated around the vertical axis.**

2) IMAGE ENHANCEMENT AND NOISE REMOVAL

Most Braille images are exaggerated by noise and deep models trained on noisy images would result in poor recognition performance. Therefore, there is a high need to enhance image quality before feeding them to any deep CNN model. The image enhancement steps are given as follows:

Step 1: At first step of object ROI extraction (dots patterns in case of Braille images), input noisy image I is filtered with Wiener filter [58]. From the input noisy images, it smooths the noise and inverts the blur effect simultaneously. This filtering process enhances the doted regions and preserves smooth boundaries.

Step 2: In the next step, hysteresis threshold [59] is applied. The binary image obtained after thresholding can be expressed as:

$$I_w = \begin{cases} 0, \text{ if } I(x,y) < T_1 \\ 1, \text{ if } I(x,y) > T_2 \\ 0, \text{ if } T_1 \le I(x,y) \le T_2 \text{ and } I(x,y) \text{ is a neighbour of pixels with label } 0 \\ 1, \text{ if } T_1 \le I(x,y) \le T_2 \text{ and } I(x,y) \text{ is a neighbour of pixels with label } 1 \end{cases} \quad (15)$$

where, T_1 and T_2 denote the upper and lower threshold. In experiments, we set the upper threshold T_1 to 0.9 and lower threshold T_2 to 0.45.

Step 3: Binary images are then processed with morphological opening by reconstruction using a disc-shaped structuring element [60].

$$\phi(I) = I - p_s(I_w \mid I) \quad (16)$$

where $p_s(I_w \mid I)$ represent the re-constructor operator of structuring element s and mask image I_w. Whereas, the mask image can be obtained after the opening process as follows:

$$I_w = (I \Theta S) \oplus S \quad (17)$$

Step 4: The mask image I_w is then processed to output image I_o using morphological erosion and dilation operations [60]. The grayscale erosion is described as:

$$(I_w \Theta S)(x,y) = \min\left\{ I_w(x+x', y+y') \mid (x', y') \in V_s \right\} \quad (18)$$

where Θ denotes binary erosion between r and structuring elements. The V_s is the domain of structuring elements.

Step 5: Similarly, grayscale dilation is written as:

$$(I_w \oplus S)(x,y) = \max\left\{ I_w(x-x', y-y') \mid (x', y') \in V_s \right\} \quad (19)$$

The results of Wiener filtering, thresholding, and morphological operations are shown in Figure 4. From Figure 4, it can be seen that the Wiener filter highlights the core area and improves the image quality by eliminating additive noises even when the image is blur having low intensity. The morphological operations help in removing unnecessary objects and preserving smooth boundaries. Furthermore, the erosion and dilation help in removing unnecessary objects in final ROI images. The ultimate goal of image alignment and enhancement is to improve image quality. From experimental analysis, we found that image preprocessing significantly improves the performance of the deep CNN model in terms of recognition efficiency. Detail of image alignment and enhancement steps is given in Algorithm 1.

B. PROPOSED IMAGE CLASSIFICATION METHOD

In this section, the proposed deep CNN model is systematically described and various used techniques are discussed. The convolutional neural network is one form of artificial neural network (ANN) that is mostly applied to pattern recognition tasks. Unlike other ANN networks, CNNs use spatial type of kernels and strides to maintain temporal and spatial details, while the ANNs suppress the information into one dimensional metrics. Many researchers have been using CNNs in the field of image processing and obtained robust performance. Deep CNN mainly consists of alternative, convolutions, max-pooling, and fully connected layers that learn high level details from low level representations [42]. These layers are trained by tuning several hyper parameters such as stride, kernel size, and padding.

Algorithm 1 for preprocessing of Braille image

```
Input: noisy image $I$
Output: processed image $I_o$
Procedure:
Image alignment
##Given the input image
    $X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$    /eigenvalues and corresponding eigenvectors are computed from
    $\theta = \cos^{-1}\left(\frac{tr(X)}{2}\right)$    /rotation angle of given misaligned image is computed
    $\theta^{'} = -(x_{11} \times x_{21})\theta$    /misaligned images are rotated back to standard orientation using $\theta^{'}$
Image enhancement
    $I_w = Wiener(I, noise)$    /wiener filtering
    $I_w = \begin{cases} 0, \text{ if } I(x,y) < T_1 \\ 1, \text{ if } I(x,y) > T_2 \\ 0, \text{ if } T_1 \le I(x,y) \le T_2 \text{ and } I(x,y) \text{ is a neighbour of pixels with label } 0 \\ 1, \text{ if } T_1 \le I(x,y) \le T_2 \text{ and } I(x,y) \text{ is a neighbour of pixels with label } 1 \end{cases}$    /hysteresis thresholding
    $\phi(I_w) = I_w - p_s(I_m \mid I_w)$    / morphological opening by reconstruction
    $(I_w \Theta S)(x,y) = \min\left\{I_w(x+x^{'}, y+y^{'}) \mid (x^{'}, y^{'}) \in V_s\right\}$    /morphological erosion
    $(I_w \oplus S)(x,y) = \max\left\{I_w(x-x^{'}, y-y^{'}) \mid (x^{'}, y^{'}) \in V_s\right\}$    /morphological dilation
```

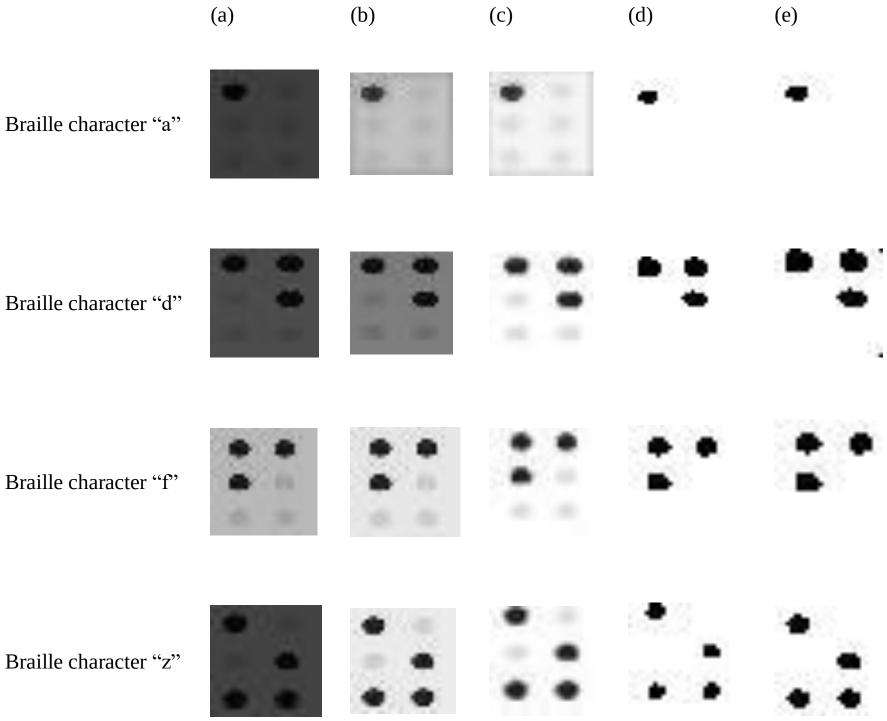


FIGURE 4. (a) Original image, (b) filtered with Wiener filter, (c) after hysteresis thresholding, (d) after morphological reconstruction, (e) morphological flood-fill i.e., erosion followed by dilation. The images are taken from English Braille dataset

In this paper, we have designed a deep CNN model consisting of two main layers, i.e., feature extraction and classification layer. For feature extraction, we selected previously proposed deep CNN networks: VGG16 [43], ResNet50 [45], Inceptionv3 [44] and DensNet201 [46]. In each original network, last few modules are replaced with IRB module [57].

We selected four networks as the backbone of our model i.e. VGG16 [43], ResNet50 [45], InceptionV3 [44], and DenseNet201 [46]. In case of VGG16 as backbone, last three convolution layers are replaced with IRB module. In case of ResNet50, last six convolution layers are replaced with IRB module. In case of InceptionV3 as backbone, the last Inception module is replaced with IRB module. Essentially, main structure of DenseNet201 is composed of four different dense blocks and transition layers. The transition layers (1×1 convolution followed by 2×2 average pooling) are used between two contiguous dense blocks to connect them. Each dense block consists of 6, 12, 48, 32 Bottleneck layers, respectively. In each Bottleneck layer, BN-ReLU-1×1 Conv is used before BN-ReLU-3×3 Conv, where BN represents the Batch Normalization. In case of DenseNet201, we removed 10 Bottleneck layers in the last dense block with IRB module. The network architecture of designed deep CNN model (with DenseNet201 backbone) is shown in Figure 5. In the IRB module, a 1×1 pointwise convolution is added, before and after the depthwise convolution layer. Because in depthwise convolution, the number of channels in output feature map (C_{out}) are same as input feature map (C_{in}). The pointwise convolutional layer increases the dimension of output feature maps. As the feature maps are expanded by a factor n, number of output channels are increased to $n \times C_{in}$. In this way, network extracts more discriminative features in higher dimensional space and increases the network performance. Using the second pointwise convolution layer, output dimension reduces such that the output channels are same as input channels and features information are concentrated in a channel. Architecture of inverted residual block is shown in Figure 6.

Moreover, modified network consists of convolution, batch normalization, ReLU activation function followed by global average pooling layer. In IRB module, we used 512 filters and for the comparative analysis, the hyperparameter expansion factor n is set as 1, 2, and 3. The suggested modifications in proposed model reduce the computational cost and lead to fewer network parameters.

We performed classification process mainly with Softmax function, which is used to calculate the probability value of each given class. The classification with Softmax is formulated as:

$$\sigma = \frac{e^{y_l}}{\sum_{j=1}^{J} e^{yj}} \qquad (20)$$

where y_i is input to softmax, j is total number of categories, and σ is normalized Softmax output probability. Image is assigned with a class label for which the predicted Softmax class probability value is highest. The Softmax class probability is computed as:

$$o = \arg\max(\sigma) \qquad (21)$$

In recognition model, the categorical cross entropy loss function of Softmax is used to find the proximity between actual and desired output. The loss function is computed as:

$$J = \frac{1}{N}\sum_{t=1}^{N} -\log\left(\frac{e^{y_i}}{\sum_{j=1}^{J} e^{y_j}}\right) \qquad (22)$$

where N is total number of training samples. The network training is considered as optimization process, where group of optimal solutions in parameter space is found to make the loss minimum. The gradient of loss function is calculated and weight parameters are updated until training loss is minimized to lowest value.

1) TEST TIME MINIMIZATION THROUGH INVERTED RESIDUAL BLOCK

Let the convolution layer with kernel size $k \times k$ takes the input C_{in} from previous layer and produce output feature map of size C_{out}, time computations involve by standard convolution layer [61] can be computed as:

$$O_{std}\left(C_{in} \cdot k^2 \cdot M_{out} \cdot N_{out} \cdot C_{out}\right) \qquad (23)$$

where, M_{out}, N_{out} and C_{out} are spatial dimensions of output feature maps. For instance, time computation for the first convolution layer is written as:

$$O_{std}(C_0.k_1^2.M_1.N_1.C_1) \qquad (24)$$

Equation 24 shows that time complexity in CNN architecture mainly relies on spatial dimensions of the input image, number of kernels, and kernel size used. In our proposed architecture, use of IRB and small size image are the major contributions to reduce computational complexity. For instance, when the size of input image is reduced from 28×28×3 to 14×14×3, time complexity of first convolution layer is reduced to 75% and is computed (using Eq. 24) as:

$$O_{std}(64 \times 3^2 \times 28 \times 28 \times 3) \rightarrow O_{std}(64 \times 3^2 \times 14 \times 14 \times 3) \quad (25)$$

Furthermore, use of IRB [57] effectively reduces the computational cost of proposed model. The IRB is depthwise separable convolution that separates the traditional convolution into depthwise convolution and 1×1 pointwise convolution to offer less computational cost. The computational cost of depthwise separable convolution is defined as sum of the calculation amount of deep convolution and 1×1 convolution. The computational cost of depthwise separable convolution is defined as:

$$O_{dpt}\left(C_{in} \cdot f^2 \cdot M_{out} \cdot N_{out} + C_{in} \cdot C_{out} \cdot M_{out} \cdot N_{out}\right) \qquad (26)$$

The computational cost ratio of depthwise separable convolution to standard convolution can be computed as:

$$c_r = {O_{dpt}\left(C_{in} \cdot f^2 \cdot M_{out} \cdot N_{out} + C_{in} \cdot C_{out} \cdot M_{out} \cdot N_{out}\right)} \Big/ {O_{std}\left(C_{in} \cdot f^2 \cdot M_{out} \cdot N_{out} \cdot C_{out}\right)}$$

(27)

$$c_r = {}^{1}\!/_{C_{out}} + {}^{1}\!/_{f^{2}} \tag{28}$$

where Eqs. (26) and (28) represent the advantage of depth separable convolution that changed the multiplication operation into multiplication and addition operations. For instance, using the downsampled input Braille image of 14x14 pixels at first layer, in standard convolution, 64 3x3x3 kernels move 12x12 times (i.e., 64x3x3x3x12x12=248,832 multiplications). In contrast with separable convolution, in depthwise convolution, 3 5x5x1 kernels move 12x12 times (i.e., 3x3x3x12x12 = 3,888 multiplications) and in pointwise convolution, 64 1x1x3 kernels move 12x12 times (i.e., 64x1x1x3x12x12=27,648 multiplications). Summing the multiplications of depthwise convolution and point wise convolutions gives 31,536 multiplications. These calculations show that using depth separable convolution, the time complexity of first convolution layer is decreased by 87.3%.

III. EXPERIMENTAL SETTINGS AND RESULTS ANALYSIS

This section discusses architecture of modified model, settings of experiments, dataset used, and output results. Image preprocessing algorithms are implemented with MATLAB R2018a program. Whereas, the CNN algorithm is implemented with TensorFlow [62] and Keras [63] libraries on 2.4 GHz Intel(R) Xeon(R) E5-2630 CPU with one NVIDIA Tesla M40 GPU of 12GB memory. Firstly, we investigate the impact on performance of different preprocessing techniques used. The primary aim is to check the effects of different previously proposed backbone networks on output recognition performance.

1

A. STATISTICAL MEASURES

Following measures are taken to evaluate the recognition performance of the proposed model:

$$Sensitivity = \frac{TP}{TP + FN} \tag{29}$$

$$Percesion = \frac{TP}{TP + FP} \tag{30}$$

$$F\ score = 2 \times \frac{percesion \times sensitivity}{percesion + sensitivity} \tag{31}$$

$$Predictive\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{32}$$

where TP, FP, TN & FN represent true positives, false positives, true negatives, false negatives, respectively.

Entropy [64] of perdition as:

$$H = -\sum_{y \in Y} p\left(y \mid i, D\right) \log p\left(y \mid i, D\right) \tag{33}$$

Peak signal to noise ratio (PSNR) [65] as:

$$PSNR = 20 \log MAX_I \big/ \sqrt{MSE} \tag{34}$$

where MAX_I represents the maximum intensity value in the image and MSE is the mean squared error between the original image and noise contaminated image.

B. DATASET:

In this paper, the Braille character detection performance of proposed method is checked with two publicly available Braille datasets, namely English Braille character [41] and double-sided Braille image dataset [40].

Detailed description of datasets is given in subsequent paragraphs.

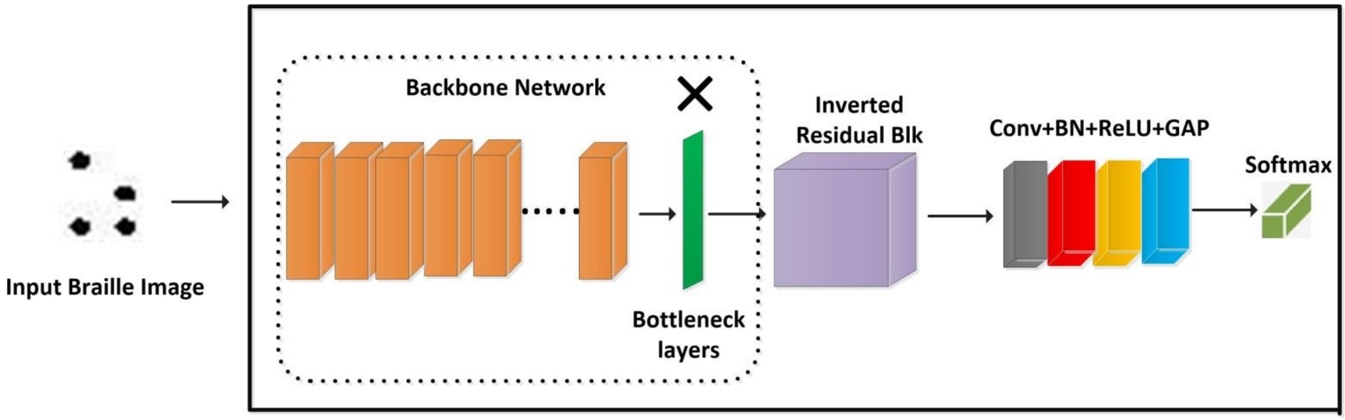


FIGURE 5. The network architecture of proposed deep CNN model (with DenseNet201 backbone). Ten Bottleneck layers in last dense block of DenseNet in backbone network are replaced with inverted residual block.

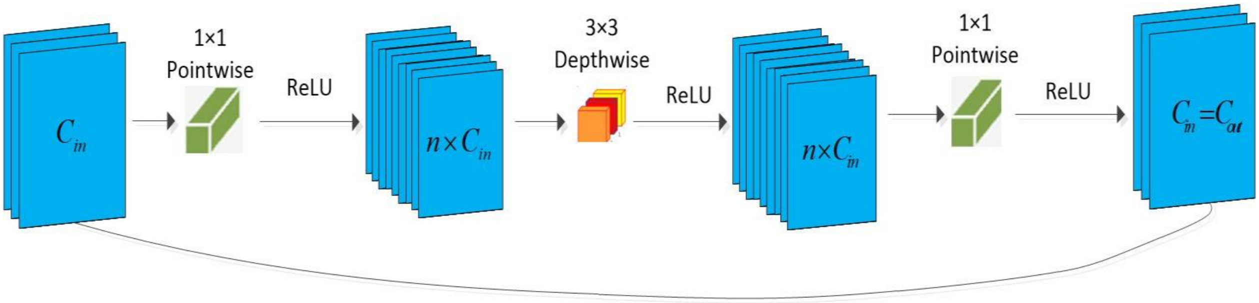


FIGURE 6. Inverted residual block.

English Braille character dataset [41].: This Braille image dataset is publicly available at open source Kaggle repository [41]. It is composed of 1560 images including 26 characters. These Braille characters' images are grayscale, 3 channel images having black and white color scales with resolution of 28x28×3 pixels. We split the dataset into training, validation, and test sets. About 20% and 10% of the Braille English character dataset are selected in the test set and validation set, respectively. While the remaining images are used for model training. The size of training dataset is relatively small so, augmented versions of original images are generated with rotation, width height shift, and brightness correction. These data augmentations increased the size of data by 60 times i.e., 3 augmentations × 20 different augmentation values.

Chinese double-sided Braille image (DSBI) dataset [40]: This Braille image dataset is also publicly available at open source Github repository [40]. It contains 114 double-sided Braille images obtained from several Braille books. Augmented versions of this dataset are also generated with rotation, width height shift, and brightness correction of the original images. In this dataset, each Braille image is of different resolution. We divided this dataset into training, validation, and test sets. Each set is proportionally sampled to construct training, validation, and test parts with 70, 14, and 30 images, respectively. We evaluated the results on both recto and verso Braille characters. Although this dataset is not large enough to provide full training of the deep CNN model, it allows us to compare different methods to the problem.

C. *EVALUATION ON ENGLISH BRAILLE DATASET*

In this section, we have shown the results obtained on English Braille character dataset [41].

Ablation Experiments: In order to check the impact in performance of preprocessing, data augmentations, and image downsampling techniques, we performed several experiments using our improved model (Backbone: DenseNet201), results are given in Table 1. In first experiment, the model is trained with original images (without preprocessing). In second experiment, the model is trained with images that are aligned using PCA algorithm ('A' denotes image alignment). In the third experiment, the model is trained with images that are preprocessed with image enhancement techniques ('E' denotes image enhancement). Similarly, in fourth experiment, the training process is done by incorporating data augmentations (rot: rotation, whs: width height shift, and dim: brightness correction) techniques. Furthermore, in the fifth experiment, the effect of image downsampling on model performance is analyzed. The resolution of input images is downsampled from 28×28×3 pixels to 14×14×3 pixels using bilinear interpolation. In sixth experiment, training process is repeated by involving all the image preprocessing, data augmentation, and downsampling techniques.

TABLE 1
ANALYSIS OF IMPACT IN PERFORMANCE OF IMAGE PREPROCESSING (A: ALIGNMENT, E: ENHANCEMENT), DATA AUGMENTATIONS (dim: BRIGHTNESS CORRECTION, rot: ROTATION, whs: WIDTH HIGHT SHIFT), AND DOWNSAMPLING ON PROPOSED METHOD WITH DENSENET201 AS BACKBONE. RESULTS ARE EVALUATED ON ENGLISH BRAILLE DATASET.

Experiments	Techniques	Precision (%)	Sensitivity (%)	F score (%)	Entropy (%)
1	Proposed method with original images	88.7	87.9	88.2	32.3
2	Proposed method+A	89.7	91.5	90.5	20.5
3	Proposed method+E	92.4	91.9	92.1	20.1
4	Proposed method+dim+ rot+ whs	93.9	93.7	92.2	19.7
5	Proposed method+downsampling	91.9	91..3	92.2	19.8
6	**Proposed method+(rot+whs+dim)+E+A+downsampling**	**95.8**	**94.7**	**95.2**	**19.2**

It can be observed from Table 1, image preprocessing improves the image quality and significantly increases the performance of the deep CNN model in terms of recognition efficiency. Particularly, with image alignment and enhancement steps, it is possible to obtain high accuracy in the Braille character recognition. From experimental analysis, we found that the incorporation of data augmentations indicates consistent improvements in classification accuracy. The impact of image downsampling on recognition performance can also be observed. It can be seen that small size of input image did not significantly affect the recognition accuracy; however, the parameters with downsampling are lower than without downsampling. Overall results demonstrate that the model achieved superior performance when all image preprocessing, augmentation, and downsampling techniques are considered. Consequently, the model obtained overall F score of 95.2% for Braille character recognition.

Furthermore, three experiments are performed with different settings of hyper-parameters. The hyper-parameters help in proper training of deep CNN model. The settings of hyper parameters and training time of each experiment are given in Table 2. We repeatedly revised model and adjusted the hyper-parameters of model to obtain better performance. Best performance is obtained in third experiment when the classifier is trained for 150 epochs at learning rate of 0.0002 and batch size of 8 using root mean square propagation (rmsprop) optimizer. These hyper-parameter values are heuristically chosen after rounds of experimentation on training data.

Experiments with different backbones: To further evaluate the ability of the designed model, results are evaluated using different backbone networks with changing the expansion factor n. The F score and parameters of our proposed models with different backbone networks are reported in Table 3.

Results show that our method with different backbones obtains the improvement to different degrees. Generally, our method with backbone as DenseNet201 achieves the highest F score in most cases. It is also important to note that use of different expansion factors of 1, 2, or 3 has little impact on the F score, it's almost similar for all values of n. Nevertheless, computational parameters are much higher with n=2 or n=3 than that with n=1. In the deep convolution layer, if the value of n increases the parameters rise sharply. Therefore, we selected n=1, in our experiments which not only makes parameters fewer but also provides a significant classification accuracy.

TABLE 2
HYPER-PARAMETER SETTINGS OF THE EXPERIMENTS USING ENGLISH BRAILLE DATASET.

Experiments	Learning rate	Batch size	Epochs	Trining Time
1	0.0001	256	60	1950s
2	0.0003	64	90	2890s
3	**0.0002**	**8**	**150**	**4800s**

The precision-recall curve of proposed model to compare the performance evaluated with different backbone networks [43][45][44][46] on test dataset is shown in Figure 7. After comparing performance with different backbones (using n=1), it is found that the modified model achieved best classification results with the DenseNet201. The evaluated results show that our method improves input Braille images quality and effectively decreases the model parameters. It also provides benefits to achieve robust recognition accuracy of Braille characters.

Classification accuracy (%) comparison with the State-of-the-art techniques: In this section, we elaborately relate our results to the state-of-the-art used in image classification. The performance of our method with backbone DenseNet201 [46] (using expansion factor n=1) is compared with original InceptionV3 [44], ResNet50 [45], VGG16 [43] networks, and even with most advanced DenseNet201 network [46]. It can be noted that our modified model has fewer parameters than the original networks [43][45][44][46] (reported in Table 4).

TABLE 3
COMPARISON OF F SCORE AND PARAMETERS OF OUR MODIFIED NETWORK (USING DIFFERENT BACKBONES) WITH CORRESPONDING ORIGINAL NETWORKS ON ENGLISH BRAILLE DATASET. F SCORE AND PARAMETERS ARE ALSO COMPARED USING DIFFERENT VALUES OF EXPANSION FACTOR n.

		n=1		n=2		n=3	
Methods	Entropy (%)	F score (%)	Parameters	F score (%)	Parameters	F score (%)	Parameters
Proposed method (Backbone:DenseNet201)	**19.2**	**95.2**	**18,430,747**	**95.4**	**26,706,764**	**95.9**	**32,080,252**
Proposed method (Backbone:InceptionV3)	20.6	94.3	21,611,487	94.8	28,830,234	95.1	33,902,231
Proposed method (Backbone:ResNet50)	21.4	89.8	29,430,123	90.1	34,50,231	91.6	39,630,456
Proposed method (Backbone:VGG16)	21.3	87.3	141,231,342	87.7	149,361,165	88.3	156,410,123

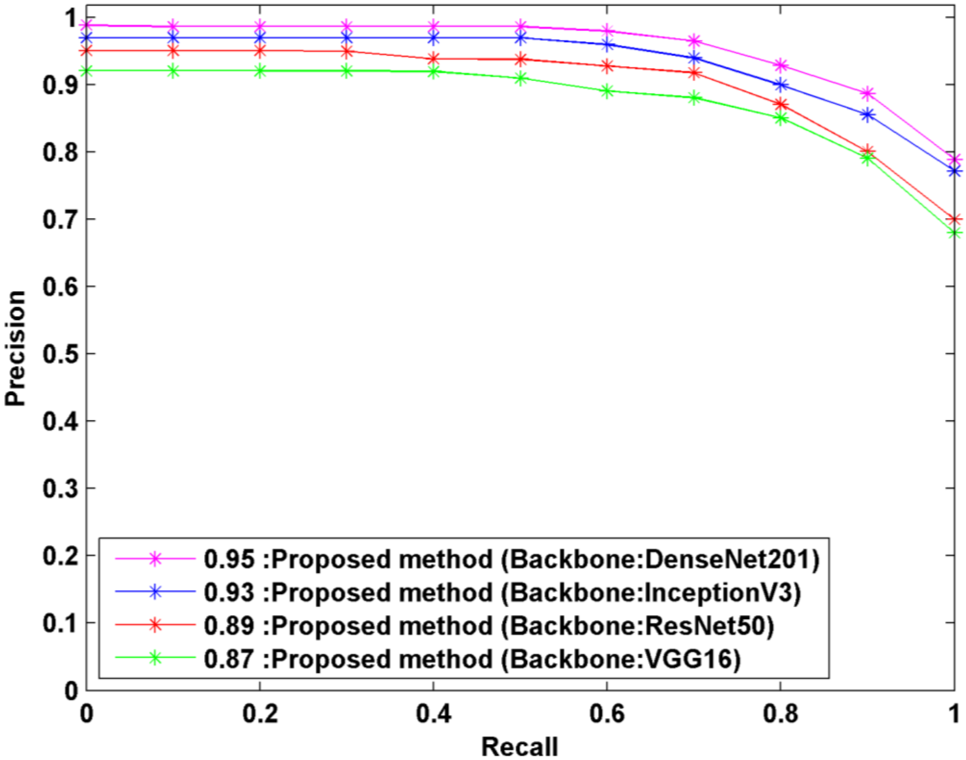


FIGURE 7. **The precision-recall curve of proposed method with different backbone networks on English Braille test data.**

This is because we have replaced few modules in original networks with the IRB which reduces the parameters. From the performance analysis, it is also clear that F score of our modified method is relatively better compare to the corresponding original InceptionV3, DenseNet201, ResNet50, and VGG16 models (reported in Table 4). Overall, experiments performance indicates that proposed approach not only improves the character recognition accuracy but also reduces the parametric load effectively.

Entropy measure is used to compute the uncertainty of predictions. It captures the fact that misclassification mainly occurred for highly uncertain predictions. Therefore, to measure the Aleatoric uncertainty [66] of model, we computed the entropy of predictions. In comparison to the state-of-the-art models (see Table 4), our proposed method (Backbone: DenseNet201) shows low entropy of prediction i.e. 19.2%.

D. *TIME ANALYSIS*

Training a deep learning model is most challenging, takes a long time. However, in real time scenarios, it is foremost priority to make the recognition speed of automatic classification models fast [67][68]. The results shows that our proposed method (Backbone:DenseNet201) has fewer parameters as compared to original DenseNet201 model. This is because, in original DenseNet201 model, few modules are replaced with an IRB module (using n=1) which has fewer parameters compared to original bottleneck layers and therefore makes the computational cost lower. In our improved versions of DenseNet, InceptionV3, ResNet50, and VGG16 network, the use of IRB effectively reduced the memory requirement and inference time. The appropriate use of expansion factor n values in IRB provides the mechanism to control the parametric load. Furthermore, the concept of image downsampling also helps to control computational costs. We repeated the training process several times until optimized set of parameters are obtained. The traing time of proposed model for different experiments is given in Table 5. Moreover, the overall train and test time of our improved versions of DenseNet, InceptionV3, ResNet50, and VGG16 network is given in Table 5. The train time is model training time while test time is inference time of model computed for a single image.

TABLE 4
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART MODELS ON ENGLISH BRAILLE DATASET.

Methods	F score (%)	Parameters	Entropy (%)
Proposed method (Backbone:DenseNet201)	**95.2**	**18,430,747**	**19.2**
DenseNet201	92.9	18,610,731	23.5
InceptionV3	92.7	21,852,176	26.4
ResNet50	87.5	29,623,164	33.1
VGG16	85.9	198,231,342	32.7

TABLE 5
TRAIN AND TEST TIME ON ENGLISH AND DSBI BRAILLE DATASETS.

Methods	English Dataset		DSBI Dataset	
	Train Time (s)	Test Time /Image (s)	Train Time (s)	Test Time /Image (s)
Proposed method (Backbone:DenseNet201)	**4800**	**0.01**	**5300**	**0.03**
Proposed method (Backbone:InceptionV3)	5700	0.08	6300	0.06
Proposed method (Backbone:ResNet50)	7300	0.15	7900	0.10
Proposed method (Backbone:VGG16)	6200	0.30	6800	0.07

E. ROBUSTNESS AGAINST INPUT PERTURBATIONS

In this analysis, the capability of proposed model (backbone DenseNet20) is checked against input image noise perturbations. To evaluate robustness of the model to input image perturbations, we added different levels of multiplicative Gaussian noise to test images and input the contaminated noisy images into the classifiers. The image perturbation is described as:

$$\bar{i} = i + m.randn(c_1, c_2, 3) \qquad (35)$$

where i represents the input image of size $c_1 \times c_2 \times 3$, $\bar{i}$ denotes contaminated data, $randn()$ represents the Gaussian noise function. We selected the different noise levels $m = 1, 3, 5, 7, 9, 11$. In image perturbation the pixel values may exceed [0, 255], we clipped the exceeded values to limit them in range [0, 255]. We performed separate experiments by using (i) original input images, (ii) preprocessed input images (iii) preprocessed images contaminated with different noise levels. From robustness test results, as given in Table 6, it can be concluded that our modified network has strong robustness compared to original DenseNet201 network for all types of input images. This analysis also demonstrates strong generalization capability of modified model to accept wider range of Braille data with inevitable noise perturbations. Moreover, we studied the effect of increase in noise level on the PSNR [65] values. The PSNR computed (using Eq. 34) for preprocessed images and noisy images (contaminated with different noise levels) are given in Table 7. In Table 7, it can be seen that PSNR values reduce for the images which are degraded by high noise. It is confirmed corrupting noise affects the fidelity of image representation.

F. EVALUATION ON CHINESE DOUBLE-SIDED BRAILLE (DSBI) DATASET

To further check the ability of the designed model, we evaluated its performance on double-sided Braille image dataset [40].

Similar to the English Brail dataset [41], we performed image alignment and enhancement with image preprocessing techniques. We adopted same training protocols as were set in case of English Braille images. On this dataset, the performance is also calculated in terms of precision, sensitivity, and F score. The application of deep learning model on the DSBI dataset has been reported in few previously published works [37] [28] [40] [36]. We compared effectiveness of our algorithm with these approaches. The compared methods used U-Net [40] and BraUNet [36] models in their approaches. U-Net [40] and BraUNet [36] are deep CNN based models. BraUNet is segmentation model based on modified U-Net architecture with auxiliary foreground segmentation task to determine the area occupied by characters. Furthermore, the results are compared with the recent TS-OBR method [38] that used a traditional machine learning cascaded classifier. The results are also compared to a recently proposed Braille recognition method [28]. The authors of [40][36] compared the accuracy of algorithms at dot detection, while in [27][38] accuracies are compared both on character and dots levels. To make a fair performance comparison, methods are tested on same hardware. We retrain and test the U-Net [40], BraUNet [36], TS-OBR [38], and RetinaNet [28] methods on our newly divided sets of DSBI Braille images (i.e. training, validation, and test parts with 70, 14, and 30 images). The performance comparison of tested methods is given in Table 8. The results show that our model achieved an accuracy of 0.983% in character level detection on DSBI dataset. Our method took about time 0.03s/image on 2.4 GHz Intel(R) Xeon(R) E5-2630 CPU with one NVIDIA Tesla M40 GPU of 12GB memory. Our method obtained a remarkable performance in terms of F score, outperforms other methods in terms of recognition speed. This Braille character recognition speed can be acceptable in real time applications.

TABLE 6
ANALYSIS OF ROBUSTNESS IN TERMS OF PREDICTIVE ACCURACY (%) ON ENGLISH BRAILLE DATASET. Where m DENOTES DIFFERENT NOISE LEVELS.

Techniques	Original images	Preprocessed images	Preprocessed images contaminated with different noise levels				
			m=1	m=3	m=5	m=7	m=9
DenseNet201	82.1	92.9	92.6	79.1	53.5	39.4	23.6
Our proposed model (DenseNet201)	**88.2**	**95.2**	**94.3**	**87.6**	**72.3**	**69.7**	**57.2**

TABLE 7
PSNR OF IMAGES CONTAMINATED WITH DIFFERENT NOISE LEVELS.

Datasets	Preprocessed images	Preprocessed images contaminated with different noise levels				
		m=1	m=3	m=5	m=7	m=9
English Braille	34.1	32.0	29.2	26.1	23..4	20.2
DSBI	33.2	31.3	28.2	26.3	23.1	20.2

All compared methods [40][38][28][36] applied several post processing strategies which increase their computational burden. The use of IRB module in our proposed scheme provides the mechanism to control the parametric load and effectively reduce the memory requirement and inference time of CNN model.

G. MCNEMAR'S STATISTICAL TEST

To evaluate the statistical significance of the difference between the results obtained with our method and other previously proposed methods, we have performed a McNemar's statistical test. This statistical test is widely used to check whether one machine learning classifier outperforms other on a given problem. The null hypothesis of McNemar's test defines that two methods disagree with same value. However, if two methods disagree with different values, the null hypothesis is rejected. In this test, we set the significance level at alpha=5%. The acceptance of null hypothesis assumed that the p-value is greater than the alpha value. The p-values obtained during experimental analysis are given in Table 9. The obtained p-values are greater than alpha value of 5%. The null hypothesis for given test proves that there is no difference in the disagreement. The tested methods have similar proportion of errors. The null hypothesis shows that classification methods have similar performance metric values, the difference in any is by chance only and rejected in each case. From this analysis, it can be concluded that proposed method takes about 0.03s/image for DSBI dataset, outperforms the existing Braille recognition techniques, and provides state-of-the-art performance for real time applications.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, an automatic Braille image character recognition algorithm using a lightweight convolution neural network is proposed. In preprocessing stage, designed approach applied principal component analysis strategy to resolve the misalignment problem by assessing the orientation of the true side of the object i.e., Braille dots pattern in Braille image. Furthermore, the mathematical, adaptive, and geometric operators are used to improve the Braille image features. Subsequently, proposed lightweight convolution neural with IRB is used to perform character recognition. Comparison with state-of-the-art methods concluded that the designed method performed fast character recognition with better accuracy (i.e. 95.2% and 98.3% for English and Chinese Braille datasets, respectively), and robust to input image noise perturbations. The time complexity of proposed model is also compared with state-of-the-art methods. Our model involves fewer time computations and performs the Braille character recognition fastly (i.e. 0.01s and 0.03s for English and Chinese Braille images, respectively). The state-of-the-art character recognition accuracy and speed prove that the proposed can be suited for real-time applications. In the future, this method can be extended to perform recognitions on larger Braille datasets. Future work may also investigate the impacts of Braille image qualities on Braille character recognition systems due to the difference in image sources

TABLE 8
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART MODELS ON DSBI DATASET.

Techniques	Dot level detection			Character level detection			
	Precision	Sensitivity	F score	Precision	Sensitivity	F score	Test time/image (s)
U-Net [40]	0.9172	0.9811	0.948				
Haar [40]	0.9765	0.9638	0.970				
HOG,SVM [38]	0.9314	0.9869	0.958				15.02
TS-OBR [38]	0.9965	0.9997	0.996	0.9928	0.9996	0.9962	1.45
BraUNet [36]				0.9943	0.9988	0.9966	0.25
RetinaNet [28]	0.9995	0.9986	0.9991	0.9985	0.9978	0.9981	0.18
Our method				**0.9841**	**0.9812**	**0.9831**	**0.03**

TABLE 9
STATISTICAL TEST ON DSBI DATASET.

Techniques	p-value
U-Net [40]	0.0001
Haar [40]	0.0032
HOG,SVM [38]	0.0067
TS-OBR [38]	0.012
BraUNet [36]	0.018
RetinaNet [28]	0.035

REFERENCES

[1] S. Shokat, R. Riaz, S. S. Rizvi, K. Khan, F. Riaz, and S. J. Kwon, “Analysis and Evaluation of Braille to Text Conversion Methods,” *Mob. Inf. Syst.*, 2020, doi: 10.1155/2020/3461651.

[2] A. Mousa, H. Hiary, R. Alomari, and L. Alnemer, “Smart braille system recognizer,” *IJCSI Int. J. Comput. Sci. Issues*, 2013.

[3] G. LeCun, Y., Bengio, Y., Hinton, “Deep learning. nature 521 (7553): 436,” *Nature*, 2015.

[4] T. Patil, S. Pandey, and K. Visrani, “A Review on Basic Deep Learning Technologies and Applications,” in *Lecture Notes on Data Engineering and Communications Technologies*, 2021.

[5] R. Aggarwal *et al.*, “Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis,” *npj Digital Medicine*. 2021, doi: 10.1038/s41746-021-00438-z.

[6] T. Kausar, M. A. Ashraf, A. Kausar, and I. Riaz, “Convolution Neural Network based Approach for Breast Cancer Type Classification,” 2021, doi: 10.1109/IBCAST51254.2021.9393249.

[7] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, “Revisiting multiple instance neural networks,” *Pattern Recognit.*, 2018, doi: 10.1016/j.patcog.2017.08.026.

[8] P. Wang, E. Fan, and P. Wang, “Comparative analysis of image classification algorithms based on traditional machine learning and deep learning,” *Pattern Recognit. Lett.*, 2021, doi: 10.1016/j.patrec.2020.07.042.

[9] Y. Yuan, C. Wang, and Z. Jiang, “Proxy-Based Deep Learning Framework for Spectral-Spatial Hyperspectral Image Classification: Efficient and Robust,” *IEEE Trans. Geosci. Remote Sens.*, 2021, doi: 10.1109/TGRS.2021.3054008.

[10] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, “A survey and performance evaluation of deep learning methods for small object detection,” *Expert Syst. Appl.*, 2021, doi: 10.1016/j.eswa.2021.114602.

[11] A. W. Kabani and M. R. El-Sakka, “Object detection and localization using deep convolutional networks with softmax activation and multi-class log loss,” 2016, doi: 10.1007/978-3-319-41501-7_41.

[12] B. Wu *et al.*, “FF-CNN: An efficient deep neural network for mitosis detection in breast cancer histological images,” 2017, doi: 10.1007/978-3-319-60964-5_22.

[13] J. Ji, X. Lu, M. Luo, M. Yin, Q. Miao, and X. Liu, “Parallel Fully Convolutional Network for Semantic Segmentation,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2020.3042254.

[14] A. Ouahabi and A. Taleb-Ahmed, “Deep learning for real-time semantic segmentation: Application in ultrasound imaging,” *Pattern Recognit. Lett.*, 2021, doi: 10.1016/j.patrec.2021.01.010.

[15] N. Alalwan, A. Abozeid, A. A. A. ElHabshy, and A. Alzahrani, “Efficient 3D Deep Learning Model for Medical Image Semantic Segmentation,” *Alexandria Eng. J.*, 2021, doi: 10.1016/j.aej.2020.10.046.

[16] Z. Song, “English speech recognition based on deep learning with multiple features,” *Computing*, 2020, doi: 10.1007/s00607-019-00753-0.

[17] Y. Jin *et al.*, “Deep-Learning-Enabled MXene-Based Artificial Throat: Toward Sound Detection and Speech Recognition,” *Adv. Mater. Technol.*, 2020, doi: 10.1002/admt.202000262.

[18] T. Kausar, M. Wang, M. A. Ashraf, and A. Kausar, “SmallMitosis: Small Size Mitotic Cells Detection in Breast Histopathology Images,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2020.3044625.

[19] T. Kausar, M. J. Wang, M. Idrees, and Y. Lu, “HWDCNN: Multi-class recognition in breast histopathology with Haar wavelet decomposed image based convolution neural network,” *Biocybern. Biomed. Eng.*, vol. 39, no. 4, pp. 967–982, Oct. 2019, doi: 10.1016/j.bbe.2019.09.003.

[20] Y. Lu *et al.*, “Entropy-Based Pattern Learning Based on Singular Spectrum Analysis Components for Assessment of Physiological Signals,” *Complexity*, 2020, doi: 10.1155/2020/4625218.

[21] M. Bakator and D. Radosav, “Deep learning and medical diagnosis: A review of literature,” *Multimodal Technologies and Interaction*. 2018, doi: 10.3390/mti2030047.

[22] V. V. Murthy, M. Hanumanthappa, and S. Vijayanand, “Braille Cell Segmentation and Removal of Unwanted Dots Using Canny Edge Detector,” 2021, doi: 10.1007/978-981-15-3514-7_7.

[23] I. G. Ovodov, “Semantic-based Annotation Enhancement Algorithm for Semi-supervised Machine Learning Efficiency Improvement Applied to Optical Braille Recognition,” 2021, doi: 10.1109/ElConRus51938.2021.9396534.

[24] S. Alufaisan, W. Albur, S. Alsedrah, and G. Latif, “Arabic Braille Numeral Recognition Using Convolutional Neural Networks,” 2021, doi: 10.1007/978-981-33-4909-4_7.

[25] R. F. Turkson, F. Yan, M. K. A. Ali, and J. Hu, “Artificial neural network applications in the calibration of spark-ignition engines: An overview,” *Engineering Science and Technology, an International Journal*. 2016, doi: 10.1016/j.jestch.2016.03.003.

[26] S. T and V. Udayashankara, “A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters,” *Int. J. Comput. Appl.*, 2013, doi: 10.5120/13993-2015.

[27] H. Burton, A. Z. Snyder, T. E. Conturo, E. Akbudak, J. M. Ollinger, and M. E. Raichle, “Adaptive changes in early and late blind: A fMRI study of Braille reading,” *J. Neurophysiol.*, 2002, doi: 10.1152/jn.00285.2001.

[28] I. G. Ovodov, “Optical Braille recognition using object detection CNN,” *arXiv*, 2020.

[29] B. M. Hsu, “Braille recognition for reducing asymmetric communication between the blind and non-blind,” *Symmetry (Basel).*, 2020, doi: 10.3390/SYM12071069.

[30] J. Mao, J. Zhu, X. Wang, H. Liu, and Y. Qian, “Speech Synthesis of Chinese Braille with Limited Training Data,” 2021, doi: 10.1109/icme51207.2021.9428160.

[31] P. Kaur, S. Ramu, S. Panchakshari, and N. Krupa, “Conversion of Hindi Braille to Speech using Image and Speech Processing,” 2020, doi: 10.1109/UPCON50219.2020.9376566.

[32] A. A. Choudhury, R. Saha, S. Z. Hasan Shoumo, S. Rafsun Tulon, J. Uddin, and M. K. Rahman, “An efficient way to represent braille using YOLO algorithm,” 2019, doi: 10.1109/ICIEV.2018.8641038.

[33] M. Jiang *et al.*, “Braille to print translations for Chinese,” *Inf. Softw. Technol.*, 2002, doi: 10.1016/S0950-5849(01)00220-8.

[34] W. Richards, “Music Braille Pedagogy: The Intersection of Blindness, Braille, Music Learning Theory, and Laban,” vol. 1994, 2020, [Online]. Available: https://researchspace.auckland.ac.nz/handle/2292/51655.

[35] S. Shokat, R. Riaz, S. S. Rizvi, A. M. Abbasi, A. A. Abbasi, and S. J. Kwon, “Deep learning scheme for character prediction with position-free touch screen-based Braille input method,” *Human-centric Comput. Inf. Sci.*, 2020, doi: 10.1186/s13673-020-00246-6.

[36] R. Li, H. Liu, X. Wang, J. Xu, and Y. Qian, “Optical braille recognition based on semantic segmentation network with auxiliary learning strategy,” 2020, doi: 10.1109/CVPRW50498.2020.00285.

[37] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015, doi:

10.1007/978-3-319-24574-4_28.

[38] R. Li, H. Liu, X. Wang, and Y. Qian, "Effective Optical Braille Recognition Based on Two-Stage Learning for Double-Sided Braille Image," 2019, doi: 10.1007/978-3-030-29894-4_12.

[39] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020, doi: 10.1109/TPAMI.2018.2858826.

[40] R. Li, H. Liu, X. Wang, and Y. Qian, "DSBI: Double-Sided Braille image dataset and algorithm evaluation for Braille dots detection," 2018, doi: 10.1145/3301506.3301532.

[41] "No Title." https://www.kaggle.com/shanks0465/braille-character-dataset/ Accessed 20 August 2020.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, 2017, doi: 10.1145/3065386.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016, doi: 10.1109/CVPR.2016.308.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, doi: 10.1109/CVPR.2016.90.

[46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2017, doi: 10.1109/CVPR.2017.243.

[47] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014.

[48] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," 2014, doi: 10.1109/ICACCI.2014.6968381.

[49] R. Kreslin, P. M. Calvo, L. G. Corzo, and P. Peer, "Linear chromatic adaptation transform based on Delaunay triangulation," *Math. Probl. Eng.*, 2014, doi: 10.1155/2014/760123.

[50] A. K. Singh and B. K. Singh, "Applications of Human Biometrics in Digital Image Processing," *Int. J. Innov. Sci. Res. Technol.*, 2020, doi: 10.38124/ijisrt20jul748.

[51] H. Z. Ur Rehman and S. Lee, "Automatic Image Alignment Using Principal Component Analysis," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2882070.

[52] T. Z. Xiang, G. S. Xia, X. Bai, and L. Zhang, "Image stitching by line-guided local warping with global similarity constraint," *Pattern Recognit.*, 2018, doi: 10.1016/j.patcog.2018.06.013.

[53] G. Zhang, L. Pan, J. Chen, Y. Gong, and F. Liu, "Joint Alignment of Image Faces," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3003332.

[54] H. T. Likassa, "New Robust Principal Component Analysis for Joint Image Alignment and Recovery via Affine Transformations, Frobenius and L2,1 Norms," *Int. J. Math. Math. Sci.*, 2020, doi: 10.1155/2020/8136384.

[55] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, "Low-Rank Pairwise Alignment Bilinear Network for Few-Shot Fine-Grained Image Classification," *IEEE Trans. Multimed.*, 2021, doi: 10.1109/TMM.2020.3001510.

[56] T. Madeira, M. Oliveira, and P. Dias, "Enhancement of RGB-D image alignment using fiducial markers," *Sensors (Switzerland)*, 2020, doi: 10.3390/s20051497.

[57] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018, doi: 10.1109/CVPR.2018.00474.

[58] D. S. khudayer Jadwa, "Wiener Filter based Medical Image De-noising," *Int. J. Sci. Eng. Appl.*, 2018, doi: 10.7753/ijsea0709.1014.

[59] Nobuyuki Otsu, "A Threshold Selection Method from Gray-Level Histograms," 1979.

[60] "Morphological Image Analysis: Principles and Applications," *Sens. Rev.*, 2000, doi: 10.1108/sr.2000.08720cae.001.

[61] K. He and J. Sun, "Convolutional neural networks at constrained time cost," 2015, doi: 10.1109/CVPR.2015.7299173.

[62] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," 2016.

[63] F. Chollet, "Keras," *J. Chem. Inf. Model.*, 2013.

[64] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," 2018.

[65] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," 2010, doi: 10.1109/ICPR.2010.579.

[66] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?," *Struct. Saf.*, 2009, doi: 10.1016/j.strusafe.2008.06.020.

[67] T. Makkar, Y. Kumar, A. K. Dubey, Á. Rocha, and A. Goyal, "Analogizing time complexity of KNN and CNN in recognizing handwritten digits," 2018, doi: 10.1109/ICIIP.2017.8313707.

[68] C. Malon *et al.*, "Mitotic figure recognition: Agreement among pathologists and computerized detector," *Anal. Cell. Pathol.*, 2012, doi: 10.3233/ACP-2011-0029.

TASLEEM KAUSAR was born in Pakistan, in 1988. She received B.Sc. degree in electrical engineering from University of Azad Jammu and Kashmir Pakistan in 2010. She received M.Sc. degree in electronics engineering from University of Engineering and Technology Taxila Pakistan in 2013. In 2020, she received a Ph.D. degree in Electronics from Harbin Institute of China. She is working as a lecturer at Mirpur University of Science and Technology Pakistan since 2013. Her research interests include medical image analysis with deep learning techniques.

SAJJAD MANZOOR is currently assistant professor in Mirpur Institute of technology, Mirpur University of science and technology (MUST). He completed his from Hanyang University South Korea in 2016 in field of electronics systems engineering. His research interest includes robotics, vision and target tracking.

ADEEBA KAUSAR received B.Sc. degree in computer system engineering from University of Azad Jammu and Kashmir Pakistan in 2017. She received M.Sc. degree in computer system engineering from University of Engineering and Technolgy Taxila Pakistan in 2020. She is also a lecturer with the department of computer science at university of Narowal Pakistan. Her research interests focused on computer vision, image processing, and deep learning techniques. Her research interests include next-generation wireless communication systems and wireless sensor networks.

YUN LU was born in Hengyang City, Hunan Province, China in 1985. He received the B.S. degree in microelectronics from Xiangtan University, Hunan Province, China in 2009, and the M.S. degree in optical engineering from Sun Yat-sen University, China in 2011. From 2011 to 2014, he was with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, working on thedesign of mixed signal front-end circuits for biomedical applications. From 2014 to 2017, he was also a senior engineer with Launch Tech Co., Ltd. In 2020, he received the Ph.D. degree in microelectronics and solid state electronics from Harbin Institute of Technology at Shenzhen, Shenzhen, China. Since August 2020, he is working as an associate professor at Huizhou University, Guangdong Province, China. His current research interests include cognitive computing from neuroscience to engineering, innovative methods for brain-machine interface, and machine learning.

WASIF MUHAMMAD received his BSc degree in Electrical Engineering in 2006 from the Riphah University, Pakistan. He received an MSc degree in Mechatronics in 2008, and a PhD in Robotics in 2016 from the King’s College London, UK. He is currently working as an Assistant Professor in the Department of Electrical Engineering, University of Gujrat, Pakistan. He is currently the group leader of Intelligent Systems Laboratory, Department of Electrical Engineering, University of Gujrat. His research interests are mainly in the areas of robotics, intelligent control, active vision, machine vision and machine learning.

M.ADNAN ASHRAF received B.Sc. and M.Sc. degrees in electrical engineering from Mirpur University of Science and Technology Pakistan in 2009 and 2020 respectively. He started job as a trainee engineer in high voltage and short circuit lab wapda Pakistan from 2011 to 2012. He was a design Engineer with Transfopower Pvt Ltd Pakistan from 2012 to 2014. He joined the Mirpur University of Science and Technology Pakistan as a Lecturer in 2016, where he is pursuing his Ph.D. degree in control.