IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

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

Digital Object Identifier 10.1109/ACCESS.2017.DOI

User-Independent American Sign Language Alphabet Recognition Based on Depth Image and PCANet Features.

WALAA ALY[1,2], SALEH ALY[1,2,*], SULTAN ALMOTAIRI[3]

[1]Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Majmaah 11952, Saudi Arabia.
[2]Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt.
[3]Department of Natural and Applied Science, Community College, Majmaah University, Majmaah 11952, Saudi Arabia.

*Corresponding author: Saleh Aly (s.haridy@mu.edu.sa).

ABSTRACT Sign language is the most natural and effective way for communications among deaf and normal people. American Sign Language (ASL) alphabet recognition (i.e. fingerspelling) using marker-less vision sensor is a challenging task due to the difficulties in hand segmentation and appearance variations among signers. Existing color-based sign language recognition systems suffer from many challenges such as complex background, hand segmentation, large inter-class and intra-class variations. In this paper, we propose a new user independent recognition system for American sign language alphabet using depth images captured from the low-cost Microsoft Kinect depth sensor. Exploiting depth information instead of color images overcomes many problems due to their robustness against illumination and background variations. Hand region can be segmented by applying a simple preprocessing algorithm over depth image. Feature learning using convolutional neural network architectures is applied instead of the classical hand-crafted feature extraction methods. Local features extracted from the segmented hand are effectively learned using a simple unsupervised Principal Component Analysis Network (PCANet) deep learning architecture. Two strategies of learning the PCANet model are proposed, namely to train a single PCANet model from samples of all users and to train a separate PCANet model for each user, respectively. The extracted features are then recognized using linear Support Vector Machine (SVM) classifier. The performance of the proposed method is evaluated using public dataset of real depth images captured from various users. Experimental results show that the performance of the proposed method outperforms state-of-the-art recognition accuracy using leave-one-out evaluation strategy.

INDEX TERMS American Sign Language Alphabet Recognition, Hand Segmentation, Wrist Line Detection, Deep Learning, PCANet, Signer Independent

I. INTRODUCTION

Sign and gestures are considered as the most natural way to convey messages among people through body movements [1]–[4]. Though signs and gestures are classified as a non-verbal communication, they can effectively deliver the communicating messages among deaf and hearing-impaired people [5]. The most widely used method of conveying words/vocabularies using body gestures is sign language.

A plenty of research works in automatic Sign Language Recognition (SLR) have been started two decades ago especially for American [6]–[8], Australian [9], Indian [10], Korean [11], Chinese [12], Polish [13] and Arabic [14]–[16]. Many techniques based on different sensor types have been developed. These approaches employed variety of methods based on the combination of multiple sensors, machine learning, pattern recognition and image analysis techniques.

The approaches used to solve sign language recognition problems can be classified into sensor-based and vision-based methods [17]. In the sensor-based approaches, signer almost wear a special glove or sensor in order to present information of hand orientation, position, rotation and movements. However, the goal of vision-based approaches is to use images captured from camera without any need for extra sensors or gloves [5]. Existing vision-based approaches apply various image processing and machine learning techniques to analyze and represent signs using color images.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

Recently, the existence of the low-cost depth cameras allows researchers to extensively apply them in many computer vision applications [18], [19]. Depth cameras such as Microsoft Kinetic capture both color and depth images. Microsoft Kinect device contains both an infrared emitter/sensor and video camera to simultaneously capture depth and color images of the scene. It permits long range interaction between hands and camera in the range from 0.5 to 4 meters which make it appropriate for indoor hand gesture recognition applications. Depth information not only preserves the 3D shape of the object but also exhibits invariance to illumination and background variations. The robustness of depth images against lighting conditions makes them relevant to many real world applications. Since then, new systems have been introduced to facilitate human-machine interaction using Microsoft Kinect^{TM} [20], SoftKinetic [21] and Leap Motion Controller (LMC) [22] sensors.

Automatic sign language recognition systems can be categorized into three classes, namely sentence, words, and fingerspelling recognition. Alphabetic sign language recognition systems (i.e. fingerspelling) are considered as an essential part to learn sign language for new users. It helps signers to perform signs for names of people, cities and other words without known signs. These systems always rely on color images to capture texture and shape information of the hand gestures. Color-based signer independent fingerspelling methods suffer from many problems such as hand segmentation, complex backgrounds, inter-class and intra-class variations. However, depth-based methods overcome these problems as they exploit the distance information of hand from the camera while discarding other non-relevant texture information.

In this paper, a new signer independent fingerspelling recognition method is proposed based on leaning features from depth image using Convolutional Neural Network (CNN). Extracted high-level features from CNN can efficiently represent the shape of hand gestures more robustly than that of hand-crafted features. The overwhelming success of convolutional neural network models and deep leaning algorithms motivates many researchers to apply them in sign language recognition problems. However, the complex structure of recent CNN architectures and the high computational cost of training prevent their utilization in real-time applications. Motivated by the successful achievements of PCANet deep learning architecture [23] in many object recognition problems, our proposed method employs this model to automatically learn depth features from the segmented hand regions.

The first step of the proposed method is to segment the hand region from depth image. Hand segmentation based on depth image can be achieved by thresholding depth values to find the nearest object to the camera. A new simple algorithm is also proposed to detect the wrist line and remove the hand forearm. Another important step after hand segmentation is to normalize the depth values to make it more relevant to the next feature extraction stage. Two strategies are proposed to train PCANet models, namely single PCANet and user-specific PCANet feature model. The first one is trained using samples collected from all users, while the second strategy trains multiple PCANet models in which each model learns specific features from a single user. The extracted features are then recognized using linear Support Vector Machine (SVM) classifier [24]. Extensive experiments using public ASL benchmark dataset are conducted to evaluate proposed method using leave-one-out strategy. The comparative study with other state-of-the-art CNN-based fingerspelling recognition methods reveals the robustness of the proposed method.

The main contributions of the proposed method can be summarized as follows:

1) A new efficient hand segmentation and wrist line detection algorithm based on depth image is proposed.
2) A simple unsupervised convolutional neural network using PCANet is employed to describe hand gestures.
3) Evaluation of the proposed method using real database of depth ASL alphabetic for signer independent scenario.

The paper is organized as follows: Section II reviews the related works of sign language recognition, Section III explains the proposed method in details. The experimental results are discussed in Section IV and finally, conclusions are presented at the end of the paper.

II. RELATED WORKS

Recent development of various sensor types especially those depend on depth information leads to the development of many real time applications such as gesture and sign language recognition [18], [20]. As a result of their low cost, sensors such as Microsoft Kinect and leap motion controller [25] are widely spread and used by many researchers [26]–[28]. Sign language recognition problem can be divided into three sub-problems: sentence, isolated words, and alphabet recognition. This paper focuses only on recognizing American Sign Language (ASL) alphabets. Most developed ASL alphabet recognition systems consist of three stages: hand segmentation, feature extraction and classification. Although there are many research works toward solving ASL fingerspelling problem, few works have been presented to tackle user independent scenario since the amount of variations among signers are very large.

Pugeault and Bowden [29] proposed an American sign language hand gesture recognition system using Microsoft Kinect sensor. In their work, The RGB and depth of 24 English alphabetic images are collected from five different signers. Gabor filters was used to extract texture features while multiclass random forest classifier is trained to predict the label of each fingerspelling letter. Experiments showed that features extracted from Gabor filters could not efficiently discriminate different signs. Their dataset was utilized in many research works and considered as a benchmark dataset in this paper. Later works have been done by Keskin et al. [30] utilizing a Randomized Decision Forest (RDF) for

hand pose estimation and hand shape classification. They introduced a multilayered RDFs to assign each input depth pixels to hand shape classes then hand pose is estimated.

Li et al. [31] used sparse auto-encoder (SAE) and principle component analysis to learn features of hand gestures from RGB-D images. Using two separate sparse auto-encoder with convolutional neural networks, features are learned respectively from color and depth channels. The combined features from both channels was obtained using multiple PCA layers. However, they showed experimental results on American sign language (ASL) dataset for signer dependent scenario only without any discussion about the feasibility of their method to solve signer-independent problem.

In Dong et al. [8] works, hand region is segmented into parts using depth contrast feature and per-pixel classification method. They developed a method to localize hand joint positions using a hierarchical mode-seeking under kinematic constraints. Random Forest (RF) classifier was then built to recognize ASL signs from the obtained joint angles. Using publicly available dataset, their method achieved above 70% and 90% accuracy in recognizing all static ASL alphabet signs using "leave-one-out" and "half-half" experimental tests, respectively.

Zhang et al. [32] encoded the 3D shape information from depth maps using Histogram of 3D Facets (H3DF). The 3D local support surface was characterized by the 3D Facet associated with the 3D cloud point. The 3D shapes and structures of various signs are represented by the H3DF descriptor. The recognition results using SVM and sparse representation (SR) classifiers for ASL alphabet reached 73.3% and 77.2%, respectively.

Wang et al. [33] proposed a superpixel earth mover distance metric for hand gesture recognition using Kinect depth camera. The extracted depth, skeleton information, and textures are represented in the form of superpixels. The robust Superpixel Earth Mover's Distance (SP-EMD) metric was applied to measure the dissimilarity between the hand gestures. The accuracy using their distance metric and features was 75.8% on the tested dataset.

Arif-Ul-Islam and Akhter [34] utilized PCA based feature extraction with Gabor filter and orientation base hash code to represent American sign language alphabets features. Artificial Neural Networks (ANN) was then used for classification. Their method was evaluated using a database containing 576 ASL alphabet sign images of the 24 alphabets. Results using both RGB and depth images proved to work comparatively better in both time and accuracy. However, authors did not measure the performance of their method against signer independent scenario and report results using only their own collected database which is not publicly available.

Kang et al. [35] used convolutional neural networks (CNNs) to built a recognition system from depth images. They trained different CNNs for the classification of more than 30 alphabets and numbers using five different subjects. They tried different learning hyper-parameters and achieved 83.58% accuracy for leave-one-out test strategy using their benchmark dataset. On the other hand, Ameen and Vadera [36] employed both color and depth images in the ASL fingerspelling recognition using CNN model. The developed CNN model contains two convolutional layers to extract features from each input. Extracted features from both layers are concatenated and fed into a fully connected layer for classification. The reported accuracy reached 80.34% using leave-one-out test strategy on the same benchmark dataset.

Tao et al. [6] use CNN with multiview augmentation and inference fusion. More perspective views are generated from the original depth image and used in the training to improve the performance of the CNN model. In the classification, the scores of the different generated views were combined to calculate the final decision. Although their method achieved state-of-the art accuracy, it requires a high computational cost to generate and test different views from the original depth image.

Another approach for American sign language recognition exploiting Recurrent Neural Networks (RNN) and Leap Motion Controller (LMC) was presented by Avola et al. [22]. LMC device was employed to detect and track the hand and fingers and to provide position and motion information. LMC was utilized to capture features of the angles between finger bones. In addition to data acquisition using LMC, RNN was used to model the long term contextual information of temporal sequences in the dynamic gestures. Their system was evaluated using an American sign alphabets containing both static and dynamic gestures.

Deep learning algorithms are currently the predominant strategy to solve many computer vision and gesture recognition problems [4], [31]. Recent researches have employed different deep learning algorithms to solve hand gesture recognition [37], [38]. Among other deep learning algorithms, PCANet [23] considered as a new effective simple unsupervised deep learning method which successfully used to solve many object recognition problems. In this work, we employ the unsupervised PCANet model instead of the commonly used supervised CNN architecture to learn features from depth images of American fingerspelling. The classification is performed using linear support vector machine classifier to label the extracted PCANet features.

III. PROPOSED METHOD

The proposed signer independent fingerspelling recognition method comprises three different stages, hand segmentation and preprocessing, feature extraction, and classification. Fig. 1 shows the block diagram of the proposed method. Hand segmentation is an important step of the method and can be efficiently achieved by thresholding depth image to find the pixels which represent hand region. After segmenting hand from depth image, precise hand region is cropped by finding the wrist line and remove hand forearm region. Pixel values of the cropped hand is then normalized to limit their values within small range.

Extracting discriminative and invariant features considered as the most critical step in the system and can lead to a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

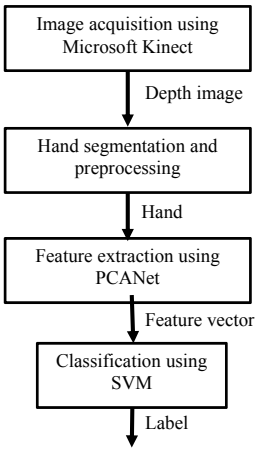

FIGURE 1: Proposed method for American fingerspelling recognition

successful fingerspelling recognition system. However, selecting appropriate invariant feature extraction method for input depth images is a difficult task. PCANet is employed to automatically learn invariant features from the segmented depth images to efficiently represent various alphabetic sign classes. The two strategies illustrated in Fig. 2 are implemented to learn different depth features from the available training images. First, single PCANet and SVM models are trained using a set of collected samples from all users while the second strategy trains multiple PCANet and SVM models in which each model learns specific features from a single user. The combination of CNN/PCANet with linear SVM classifier was previously utilized in [39], [40] to solve handwritten digit recognition problems. It is well-known that SVM training is based on solving a margin maximization optimization problem which deliver a unique solution since the optimization problem is convex. The decision boundary learned by SVM classifier greatly improve the generalization capability in comparison with other linear classifiers. In addition, SVM classifier is better than other non-linear classifiers which are based on gradient descent optimization algorithm and have local minima problem. The following subsections explain the proposed method in more details.

A. HAND SEGMENTATION AND WRIST LINE LOCALIZATION

A fundamental step in most gesture recognition applications is to efficiently segment hand regions from input image. Segmenting hand from depth images is much easier than that of color images [18]. Hand segmentation problem can be easily tackled by assuming that the hand is the nearest object to the depth camera [28]. That is, the depth values of the hand are usually smaller than other objects in the image [6]. Assume that the raw depth image D is captured by Microsoft Kinect camera, and the minimum value of the depth image is D_m. All pixels lies in the range from D_m to $D_m + T$ are selected to represent hand region, the threshold value T is selected empirically. The pixel values is then normalized to increase the contrast of the hand region using the formula $D' = D_m + T - D \;\; (if D \neq 0)$. This step makes hand region near to the camera more brighter, while far regions become more darker. Then, this region is filtered using median filter to remove noise.

Wrist line detection helps to remove hand forearm and focuses on the hand region of interest [41]. The procedure is based on the observation that the wrist line is directly located under the palm region and above the forearm. To find the wrist line, the following steps illustrated in Fig. 3 are explained in details as follows:

1) The binary hand mask M is computed by thresholding hand region image ($D' > 0$).
2) The circle enclosed the palm region is estimated using distance transform of the depth image denoted as $Dist(M)$. The center (X_c, Y_c) and the radius R of the circle are calculated as the position and the maximum value of $Dist(M)$, respectively.
3) Find the orientation of the hand using second moment of the binary hand mask M, denoted as θ. The orientation is obtained using the second order central moment [41].
4) Find the line pass through the center of the circle along the orthogonal direction of hand orientation $\theta_n = 90 + \theta$. The coordinates of the two points $P_1(X_1, Y_1)$ and $P_2(X_2, Y_2)$ can be calculated as follows:

$$\begin{aligned} X_1 &= X_c + R\cos(\theta_n), \\ Y_1 &= Y_c - R\sin(\theta_n) \\ X_2 &= X_c - R\cos(\theta_n), \\ Y_2 &= Y_c + R\sin(\theta_n) \end{aligned} \tag{1}$$

5) The line pass through the palm center is shifted toward the direction of the forearm to touch the enclosed palm circle. The two shifted points $P_1'(X_1', Y_1')$ and $P_2'(X_2', Y_2')$ are calculated as follows:

$$\begin{aligned} X_1' &= X_1 + R|\sin(\theta_n)|, \\ Y_1' &= Y_1 + R|\cos(\theta_n)| \\ X_2' &= X_2 + R|\sin(\theta_n)|, \\ Y_2' &= Y_2 + R|\cos(\theta_n)| \end{aligned} \tag{2}$$

Hand region is cropped by removing the lower part of the hand forearm based on the above wrist line detection algorithm. Finally, all cropped hands are centered and scaled to a fixed size to exhibit shift and scale invariance.

B. SINGLE PCANET MODEL STRATEGY

In this strategy, a single PCANet model is trained using all depth images of the available users to learn hand shape features. Gesture recognition uses various types of geometric features such as corners, edges, blobs, or ridges to represent the shape of gesture. The complexity of extracting good

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

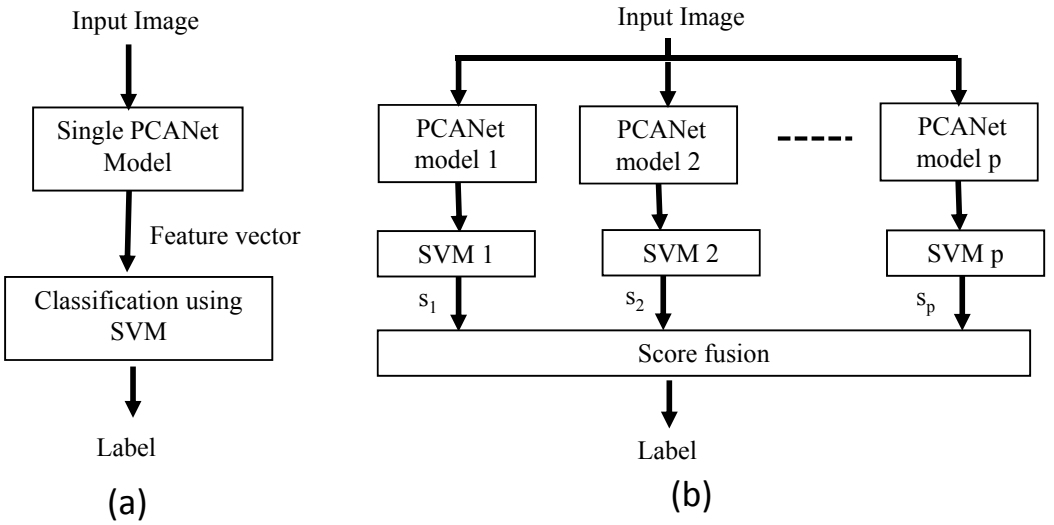

FIGURE 2: (a) Single PCANet model strategy, (b) User-specific PCANet model strategy.

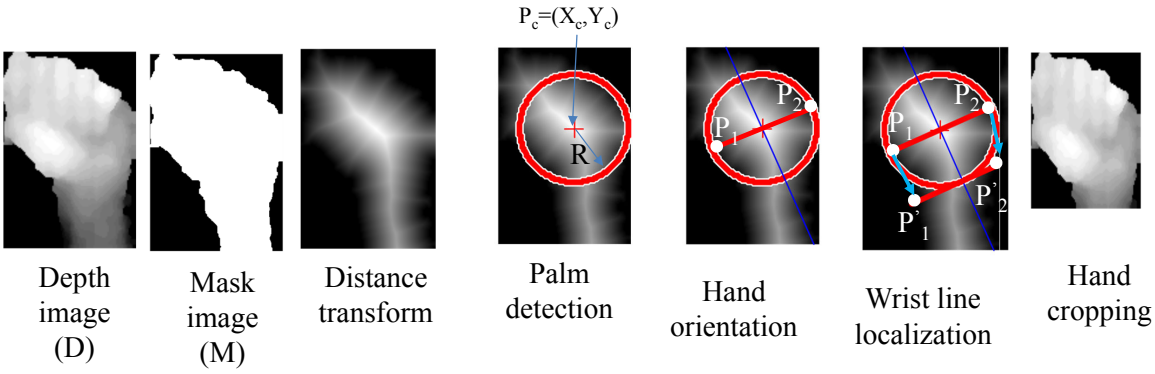

FIGURE 3: Steps of hand and wrist line localization using depth image.

geometric features makes appearance-based approaches predominant among other features. Appearance-based feature extraction methods have been applied successfully for color and gray images. However, there is no recommended feature extraction method for depth images. Therefore, feature learning algorithms using convolutional neural networks can be directly applied to learn features from depth images without relying on the hand-crafted features.

The computational complexity of current convolutional neural networks architectures makes them difficult to apply in many real time applications. However, the recently developed architecture named PCANet [23] utilizes a simple unsupervised learning algorithm compared to other methods based on the expensive back-propagation algorithm. PCANet model is trained to learn and represent both low and high-level fingerspelling features using depth input images. The feature extracted from PCANet is classified using linear support vector machine classifier [24]. Fig. 4 shows the structure of PCANet model containing two convolutional layers used to learn different feature types from input depth images.

1) PCANet model

PCANet contains two stages of convolutional layers in which all input patches are previously normalized using zero-mean normalization method. The first stage of PCANet contains a filter bank of L_1 filters used to convolve the input depth image and produce L_1 feature map images. Each of the generated L_1 feature map image is then convolved with another high level L_2 filters which gives another $L_1 \times L_2$ feature map images. The first convolutional layer of PCANet learns low-level features while second convolutional layer captures high level features. The output layer of PCANet has L_1 image banks each containing L_2 feature map images. Then, the output feature map images from the second convolutional layer are binarized using simple thresholding method to convert them into binary images. The binary images in each bank is combined into one integer image. Each of the L_1 output images is partitioned into B blocks. The histogram of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

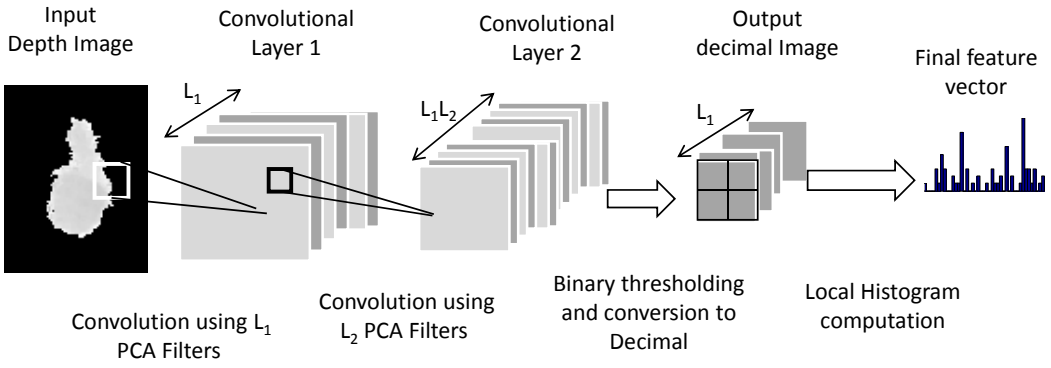

FIGURE 4: The architecture of PCANet model

the values in each block is then calculated. All the histograms of blocks are concatenated to generate the final feature vector of the input image.

The convolution layers of PCANet utilizes Principal Component Analysis (PCA) algorithm to learn a set of orthogonal filters. PCA is an orthogonal linear transformation method used to reduce the dimensionality of the input data. The covariance matrix of the original dataset is factorized to find eigenvalues and eigenvectors. The most important information can be retained by keeping the eigenvectors corresponding to the largest eigenvalues. Eigenvectors correspond to the smallest eigenvalues are always discarded. The selected eigenvectors are used as the basis/filters in the convolutional layers while all local patches are convolved with them to highlight the most important features of the input depth image.

Let assume that we have N input training depth images of size $m \times n$ $\{D_i\}_{i=1}^{N}$, and a patch of size $k_1 \times k_2$ is extracted and collected from each pixel of all input images. The collected patches are represented by $\{d_{i,1}, d_{i,2}, d_{i,mn}\} \in \Re^{k_1 k_2}$ where $d_{i,j}$ denotes the $j-th$ patch in D_i. The patch mean is subtracted from each patch, then the set of patches are vectorized and arranged into a matrix $\hat{D}_i = [\hat{d}_{i,1}, \hat{d}_{i,2}, ..., \hat{d}_{i,Nmn}]$. For all training depth images, the same matrix is constructed to give:

$$\hat{D} = [\hat{D}_1, \hat{D}_2, ..., \hat{D}_N] \in \Re^{k1k2 \times Nmn} \tag{3}$$

PCA is utilized to remove the redundancy in $\hat{D}$ and to minimize the reconstruction error by finding an orthonormal filters along the directions of maximum variance. Let the number of selected filters in the first convolutional layer equals L_1. The problem can be solved using Singular Value Decomposition (SVD) method to find the principal eigenvectors of $\hat{D}\hat{D}^T$ using:

$$\min_{v \in \Re^{k_1 k_2 \times L_1}} \|\hat{D} - VV^T\hat{D}\|^2, s.t. V^T V = I_{L_1} \tag{4}$$

where I_{L_1} represents the identity matrix with size $L_1 \times L_1$. The selected L_1 eigenvectors are rearranged to matrix to generate the convolutional filter kernel W_l^1 as:

$$W_l^1 = mat_{k_1,k_2}(q_l(XX^T)) \in \Re^{k_1 \times k_2}, l = 1, 2, ..., L_1, \tag{5}$$

where $mat_{k_1,k_2}(v)$ is a function that rearrange the vector $v \in \Re^{k1k2}$ into a matrix $W \in \Re^{k1 \times k2}$ and $q_l(XX^T)$ represents the $l-th$ principal eigenvectors. The $l-th$ filter output of the $i-th$ input depth image at the first stage is:

$$D_i^l = \hat{D}_i * W_l^1, i = 1, 2, ..., N, \tag{6}$$

where $*$ denotes two-dimensional convolution operator. The input image $\hat{D}_i$ is zero-padded before convolution to produce an output image of the same size as the input image. Output of the first convolutional layer will be used to learn PCA filters in the consecutive second layer.

The procedure have been done in the first stage is repeated to learn high-level features at the second stage. Similar to the first stage, all image patches collected from the output of L_1 filters $\hat{D}_i^l$ images are mean-removed, vectorized, and concatenated to generate the matrix:

$$\hat{Y} = [\hat{Y}^1, \hat{Y}^2 ..., \hat{Y}^{L_1}] \in \Re^{k_1 k_2 \times L_1 Nmn} \tag{7}$$

The eigenvectors of $\hat{Y}\hat{Y}^T$ are calculated and L_2 second stage PCA filters are selected and rearranged into convolutional kernel W_l^2 as:

$$W_l^2 = mat_{k_1,k_2}(q_l(YY^T)) \in \Re^{k_1 \times k_2}, l = 1, 2, ..., L_2, \tag{8}$$

For every input image D_i^l in the second stage, it will be convolved with L_2 filters to produce L_2 output images,

$$O_i^l = \{D_i * W_l^2\}_{l=1}^{L_2}. \tag{9}$$

The number of output images in the second stage will be $L_1 L_2$ since each input image will be convolved with $L1$ and $L2$ filters in the first and second stage respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

This process can be repeated to build deeper architecture of PCANet and learn other high-level features.

The output stage in PCANet is a simple thresholding and histogram calculation. The real-valued L_1L_2 output images obtained from convolutions in the second stage are converted into binary using a simple Heaviside step function as:

$$H(D_i^l * W_l^2)_{l=1}^{L_2}. \tag{10}$$

where $H(.)$ denotes the Heaviside step function which produce zero for negative inputs and one for positives. The L_1L_2 output binary images are combined and converted into O_i^l decimal image. Each pixel in the output image will take an integer value in the range $[0, 2^{L_2} - 1]$.

$$T_i^l = \sum_{l=1}^{L_2} 2^{l-1} H(D_i^l * W_l^2). \tag{11}$$

Local histograms are calculated from the decimal images to efficiently represent local information captured by filters in the second convolutional layer. Each L_1 decimal image $T_i^l, l = 1, ..., L1$ is divided into B blocks. Local histogram with 2^{L_2} bins is calculated for each block and all histograms are concatenated into one vector denoted as Bhist(T_i^l). The final feature vector of the input depth image D_i can be defined by the set of block-wise histograms as:

$$f_i = [Bhist(T_i^1), ..., Bhist(T_i^{L_1})]^T \in \Re^{2^{L_2} L_1 B}. \tag{12}$$

In case of fingerspelling application, the divided local blocks can be either overlapped or non-overlapped. Using local histogram and overlapped blocks helps to increase the robustness of the features as it offers some degree of translation invariance.

C. USER-SPECIFIC PCANET MODEL STRATEGY

One drawback of the single PCANet model based learning method is the requirement of re-training the PCANet model when presenting new users. This problem can be handled through training a single PCANet model for each user as depicted in Fig. 2(b). In order to add new user to the system, the images captured from the new user is used to train a new PCANet and SVM models without any need to use dataset from previous users. Therefore, the simplification of this learning procedure helps to reduce the computational cost of training the system. Compared with the single PCANet model strategy, the user-specific PCANet model strategy train multiple PCANet and SVM models. Formally, let the new user depth images be presented to the system, hands are segmented and preprocessed. Then, a separate PCANet model will be trained to learn user-specific features using the previously-described PCANet learning algorithm. After that, linear SVM model is trained to classify signs of this user. The trained SVM user models generate p scores denoted as: $s_1, s_2, ..., s_p$ which are fused to find the final score of the input depth image. In this work, maximum fusion operation is employed to find the label of input unknown fingerspelling sign.

IV. EXPERIMENTAL RESULTS

In this section, we examine the performance of the proposed system for signer independent recognition scenario. All depth images in the database are first segmented to extract the hand region from all users. The segmented hand region is then preprocessed and resized into 32×32 pixels. A prototype has been built using hand segmentation, proposed features extraction and linear SVM classifier to test the effectiveness of the proposed method. In the following experiments, the optimal parameters of PCANet model is obtained empirically by changing filter sizes, number of filters, block sizes and block overlap ratio. In all experiments, the performance of the proposed method is evaluated using leave-one-subject-out test strategy.

A. AMERICAN FINGERSPELLING DATABASE

In order to make a fair comparison with other related works [8], [36], the publicly available ASL fingerspelling dataset [29] was used in the evaluation. The dataset contains both color and depth images of the ASL fingerspelling alphabet recorded by five different users. The dataset contains 24 ASL fingerspelling signs excluding letters J and Z as both of these contains motion. The dataset was generated using Microsoft Kinect camera and contained more than 60,000 images. Also, there are more than 500 images for each particular sign of each user. Moreover, the alphabetic signs are captured at different viewpoint by slightly rotating hand in front of the camera. Example of the 24 ASL fingerspelling images in both color and depth are shown in Fig. 5 and Fig. 6.

B. EFFECT OF CHANGING PATCH SIZE

In this experiment, the effect of changing patch/filter size on the recognition accuracy of the proposed method is examined. The number of filters in the first and second stage is fixed to 8. The size of block used for histogram computation is selected to be 8×8 pixels without block overlap. The recognition accuracy is calculated using testing depth images from the first user while tacking the training samples from the remaining users (2, 3, 4 and 5). We vary the size of filters in the first and second stage from 3×3 to 13×13. The results shown in Fig. 7 reveal that PCANet achieve best accuracy when the filter size become 7×7 pixels. These results show that using moderate filter size is preferable than both small and large sizes. Mostly, deep CNN models which have many convolutional layers utilize small filter size. However, the shallow structure of PCANet model (two convolutional layers only) permits it to use a slightly larger filter size than other CNN models. In the next experiments, the size of convolutional layer filters is chosen to be 7×7 pixels.

C. EFFECT OF CHANGING THE NUMBER OF FILTERS

The next experiment examines the impact of changing the number of filters in the first stage of PCANet. The filter size of the network is fixed into $k_1 = k_2 = 7$, and the non-overlapping block size is set as same value of the previous experiment. We vary the number of filters in the first stage L_1

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

FIGURE 5: Example of color American sign language alphabet images performed by five different users.

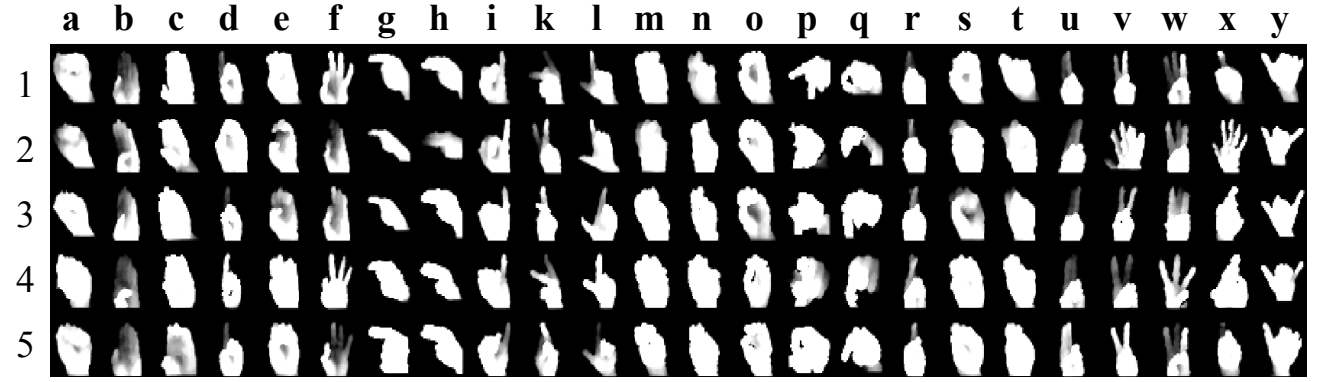

FIGURE 6: Example of depth American sign language alphabet images performed by five different users.

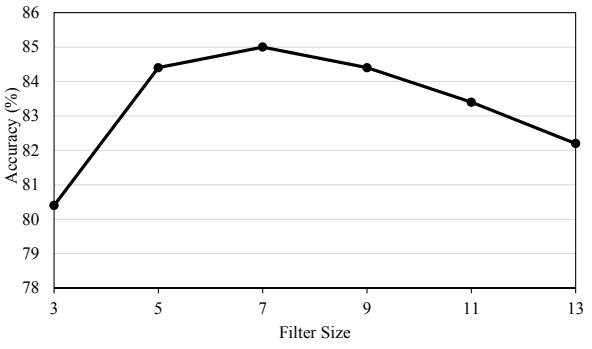

FIGURE 7: Cross-user recognition accuracy of the PCANet by varying patch/filter size

from 2 to 10 while the number of filters in the second stage L_2 is fixed into 8. It can be observed from the results shown in Fig. 8 that the accuracy is improved when the number of filters reach 7. However, increasing the number of filters beyond 7 leads to significant increase in the computational cost of the PCANet. Thus, choosing the appropriate number of filters should compromise between computational complexity and accuracy. One major drawback of PCANet model is its exponential increase of the number of feature maps when going more deeper. As a consequence, the computational complexity will increase exponentially for training and testing. Therefore, the commonly used structure of PCANet model consists of two convolutional layers with eight filters.

D. EFFECT OF CHANGING BLOCK SIZES AND BLOCK OVERLAP RATIO

In this work, we show the changes of the recognition accuracy of the PCANet by varying the block size used for histogram computation. The parameters of the PCANet are set to $k_1 = k_2 = 7$ and $L_1 = 7, L_2 = 8$. Various block sizes are considered for evaluation which varies from 4×4 to 24×24. The results shown in Fig. 9 explain the robustness of PCANet features as the block size increases until it reaches half of the image size. Since increasing the block size leads to increasing the tolerance for image variations, block size of 8×8 provides more robustness against fingerspelling variations across users. However, increasing the block size

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

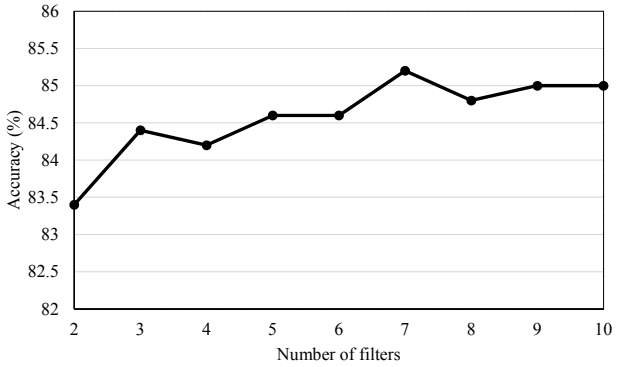

FIGURE 8: Cross-user recognition accuracy of the PCANet by varying the number of filters in the 1st stage

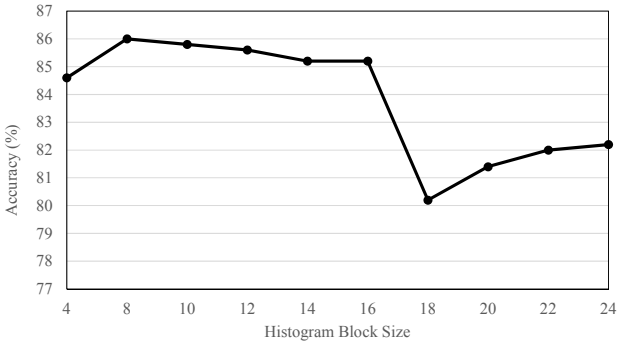

FIGURE 9: Cross-user recognition accuracy of the PCANet by varying the block size used for histogram computation

beyond half of the image size decreases the accuracy as the local spatial information of the hand shape will be lost.

In addition, we study the impact of changing block overlap ratio from 0.1 to 0.7. Fig. 10 reveals that increasing the overlap ratio improves the accuracy but at the expense of increasing the size of feature vector. The selected block overlap ratio for all subsequent experiments will be 0.4. Table 1 shows the optimum selected parameters of PCANet structure for both single and user-specific feature models.

TABLE 1: Optimal PCANet parameter values for both single PCANet and user-specific PCANet models employed in the experiments

PCANet parameter	Value
Input image size ($m \times n$)	32×32
Patch size ($k_1 \times k_2$)	7×7
Number of first layer filters (L_1)	7
Number of second layer filters (L_2)	8
Block size	8×8
Block overlap ratio	0.4

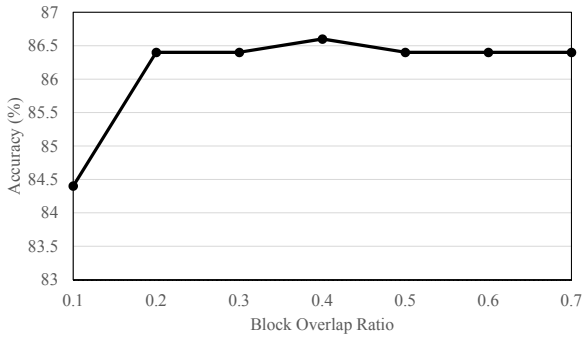

FIGURE 10: Cross-user recognition accuracy of the PCANet by varying the block overlap ratio

TABLE 2: Performance of the proposed system using leave-one-subject-out strategy

Test user	Single PCANet model	User-specific PCANet model
1	87.30%	84.10%
2	88.30%	83.70%
3	93.90%	89.90%
4	86.00%	82.50%
5	87.80%	82.40%
Average	88.70%	84.50%

E. COMPARISON BETWEEN SINGLE AND USER-SPECIFIC MODELS

This experiment compares the performance of single and user-specific models using leave-one-subject-out evaluation strategy. Table 2 shows the recognition accuracy for each user individually. The accuracy for each user depends on how much of the performed signs of this user are similar to other remaining users. User 3 gives the highest accuracy while user 4 gives the lowest one compared with others. The superiority results of user 3 is achieved because most of the signs are performed in similar way to other users. For all test users, the performance of single PCANet is always better than user-specific PCANet model. In addition, feature representation using a single PCANet trained from all users is more invariant than features extracted from user-specific models. One important limitation of utilizing user-specific model strategy is its extra computational cost (time and space complexity) compared to that of single model as we need to calculate features and scores from multiple PCANet and SVM models, respectively.

F. COMPARISON BETWEEN COLOR AND DEPTH INPUT IMAGES

The performance of using color instead of depth images is measured in this experiment. The recognition accuracy for each user is measured using either depth images (with and without preprocessing) or color images without preprocessing. The results shown in Fig. 11 reveal that depth images performs better compared to color images. Moreover, the proposed preprocessing algorithm significantly improves the results. Since preprocessing helps to focus on the region of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

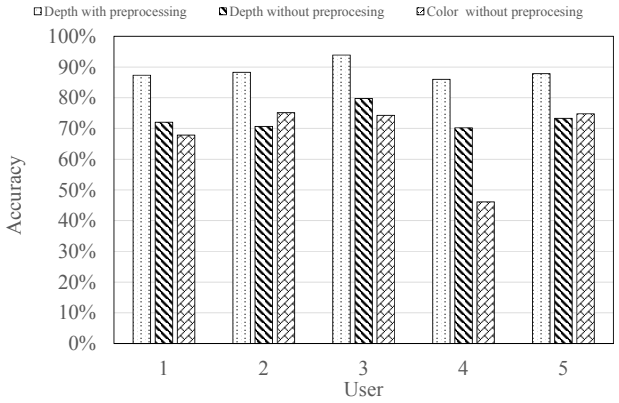

FIGURE 11: Recognition accuracy for each user using color and depth images

interest while other non-relevant regions are discarded. The proposed hand and wrist line detection algorithm works efficiently for depth image as we can easily discard the forearm and cluttered background. While the complex background and the skin-like regions can not be easily removed from color images. Thus, using depth image in sign language and hand gesture recognition is more efficient than color images.

G. COMPARISON WITH STATE-OF-THE-ART METHODS

In this experiment, the performance of the proposed method is compared with other state-of-the-art methods. Leave-one-out strategy is employed for testing. Training images from four users are used to train the proposed PCANet model while images of the remaining user are used for testing. The model parameters of PCANet are set according to the best parameters obtained from the previous experiments shown in Table 1.

Table 3 shows a comparison with state-of-the-art hand-crafted features and CNN architectures. Single PCANet model outperforms all models with acceptable margin. It is clear that using single PCANet model trained from depth images successfully represents the shape of hand gesture and improve the accuracy. Results reveal that splitting up the training of the feature extraction stage from the classification is better than the results obtained from end-to-end training of deep convolutional neural networks used in [6], [36]. Although the computation of the hand-crafted features like Gabor filters [29], LBP [42] and HOG [33] is faster, its performance can not be further improved as compared to the versatile CNN structures.

The average accuracies of all the 24 signs are computed from all users and illustrated in Fig. 12. The model gives good performance ($> 90\%$) on 15 signs (a, b, c, d, f, g, h, i, l, m, o, u, v, w and y). However, for the remaining 9 signs (e, k, n, p, q, r, s, t, and x), the average accuracies are lower than or equal 80%. The unsatisfactory performance of these 9 signs is due to the large variations among users in performing these signs and the large similarities among sign

TABLE 3: Comparison of proposed method with other state-of-the-art using leave-one-out strategy

Method	Accuracy
Pugault and Bowden [29]	49.00%
Keskin et al. [30]	84.30%
Kuznetsova et al. [45]	57.00%
Dong et al. [8]	70.00%
Maqueda et al. [42]	83.70%
Wang et al. [33]	75.80%
Zhang and Titan [32]	83.30%
Ameen and Vedera [36]	80.00%
Nai et al. [46]	81.10%
Tao et al. [6] (CNN without multiview augmentation)	84.70%
Proposed user-specific PCANet model	84.50%
Proposed single PCANet model	88.70%

themselves. The confusion matrices and the most confused sign pairs for each of the five users are shown in Figs. 13, 14, 15, 16, 17 and 18, respectively. For example, as shown in Fig. 18, different subjects perform signs in different ways than other users which make it difficult to recognize signs for the unseen user in the leave-one-out evaluation strategy. In addition, some sign pairs have very similar shape which can not be easily discriminated.

Obviously, PCANet has similar structure with many CNN architectures and can be considered as a special type of auto-encoder with the goal of minimizing the reconstruction error. However, as contrary to the CNN architectures used in fingerspelling recognition [6], [36], PCANet does not contain non-linearity and pooling layers between stages which highly reduce its computational cost. Thus, the time and space complexity of PCANet can be approximated as a linear order $\mathcal{O}(nmk_1k_2(L_1 + L_2))$ [23] which depends on the product of input image resolution $n \times m$, patch size $k_1 \times k_2$, and number of filters in each stage(L_1, L_2). The linear SVM classifier is also depends linearly on the length of final feature vector calculated from PCANet. The complexity of training the linear SVM classifier using LIBLINEAR library [43] is approximately linear in the number of features [44]. In addition, the preprocessing stage is linearly depends on the input image resolution.

There are many advantages of the proposed method such as: 1) Extracting hand region and wrist line using depth image is simple and efficient than using color images. 2) Using unsupervised PCANet model instead of the supervised CNN is computationally efficient in both training and testing. 3) PCANet training does not require any labeled data as it utilizes a simple unsupervised learning algorithm. 4) Training PCANet does not require extra GPU processing power as contrary to the recent CNN deep learning algorithms. 5) Separating the feature extraction stage from classification helps to reduce the computational cost of training and allows the pretrained PCANet feature extractor to be reused. However, one important limitation of this method is that increasing the number of convolutional layers of PCANet beyond two exponentially increases the number of features and hence increases the time and space complexity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

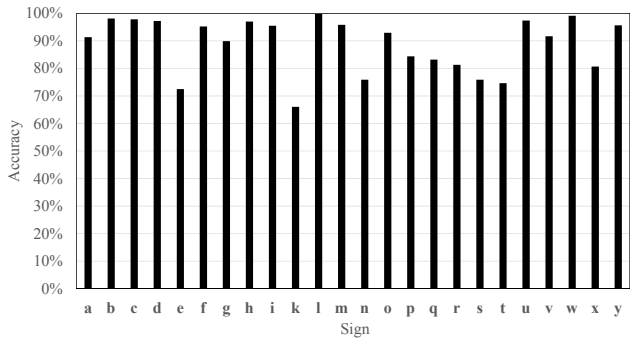

FIGURE 12: Average recognition accuracies of all alphabets using leave-one-out strategy on the ASL benchmark dataset

Predicted Signs

Actual Signs	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	459	0	0	0	0	0	0	1	0	0	0	2	0	0	2	2	0	10	48	0	0	0	0	0
b	0	533	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	0	0	727	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0
d	0	0	0	521	0	0	0	0	0	0	0	0	0	3	1	0	0	0	0	1	0	0	0	0
e	1	1	22	0	478	0	0	0	0	0	0	1	0	1	0	20	0	3	0	2	0	0	0	0
f	0	4	0	0	0	513	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
g	0	0	0	0	0	0	526	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	0	0	0	0	0	0	1	535	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0	0	0	0	0	513	0	0	15	1	1	0	0	0	0	0	0	0	0	0	0
k	0	0	0	52	0	0	11	0	0	310	29	2	5	11	21	0	59	9	1	0	3	0	25	1
l	0	0	0	0	0	1	0	0	0	0	583	0	0	0	0	0	0	0	0	0	0	0	1	1
m	0	0	0	0	1	0	0	0	0	0	0	527	7	0	0	0	0	2	1	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	550	0	0	0	0	7	6	0	0	0	1	0
o	0	0	0	0	1	0	0	0	0	0	0	0	0	546	0	0	0	1	0	0	0	0	0	0
p	0	0	0	0	0	0	0	0	0	0	0	0	0	1	531	2	0	0	0	0	0	0	0	0
q	0	0	0	0	0	0	0	0	0	0	0	0	0	2	19	519	0	0	0	0	0	0	1	0
r	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	526	0	0	0	0	0	0	0
s	7	0	0	0	69	0	1	0	0	0	0	1	0	11	0	0	0	403	14	0	0	0	1	0
t	9	0	0	0	0	0	0	2	0	0	0	1	54	0	5	0	0	0	458	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	2	0	520	2	0	0	0	0
v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	9	495	15	3	0	0
w	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	3	5	878	0	0	0
x	4	0	0	1	0	0	0	0	0	0	0	9	0	40	5	0	0	0	0	0	0	472	0	0
y	0	0	0	5	0	0	38	0	3	0	0	0	0	1	10	4	10	0	0	0	0	0	0	472

FIGURE 15: Confusion matrix of all alphabets using leave-one-out strategy for user #3

Predicted Signs

Actual Signs	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	528	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	506	0	0	1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	0	0	557	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	507	1	0	0	0	0	0	0	0	0	20	0	0	2	4	0	0	0	0	2	0
e	0	0	4	0	481	0	0	0	4	0	0	6	5	2	0	0	0	1	21	0	0	0	0	0
f	0	0	5	0	0	514	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g	0	0	0	0	0	0	528	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h	0	0	0	0	0	0	1	522	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0	0	0	0	0	504	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
k	0	0	0	0	0	0	43	129	0	318	0	0	0	7	0	19	0	0	0	0	0	0	2	0
l	1	0	0	0	0	0	0	0	0	0	515	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	506	21	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	528	0	0	0	1	1	0	0	0	0	0	0
o	0	0	0	0	0	0	0	0	0	0	0	0	0	502	0	0	0	10	0	0	0	0	0	0
p	0	0	0	0	0	0	0	0	0	0	0	0	0	0	477	95	0	0	0	0	0	0	0	0
q	1	0	6	0	0	0	0	2	0	0	0	0	0	2	9	488	0	0	1	0	0	0	6	1
r	0	0	0	10	0	0	2	0	0	191	3	0	0	1	0	100	0	0	195	0	0	0	1	25
s	2	0	8	0	244	0	0	0	0	0	0	4	3	0	0	0	207	0	0	0	0	0	0	3
t	3	0	0	0	0	0	0	13	2	0	0	0	103	7	1	3	0	13	332	0	0	0	40	7
u	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	492	0	0	0	0
v	0	0	0	0	0	0	6	0	0	30	0	0	0	0	0	0	0	0	2	468	1	0	0	0
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	525	0	0	0
x	3	0	0	0	0	0	26	0	0	138	2	0	0	0	0	14	0	0	0	0	0	343	0	0
y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	528	0

FIGURE 13: Confusion matrix of all alphabets using leave-one-out strategy for user #1

Predicted Signs

Actual Signs	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	511	0	7	0	0	0	0	0	0	0	0	0	27	0	0	2	0	0	0	0	0	0	0	0
b	0	557	0	0	0	0	0	0	0	0	0	2	0	0	0	3	7	0	0	3	0	0	0	0
c	0	0	495	0	1	0	0	0	0	0	0	0	0	0	2	26	0	0	0	7	0	0	0	0
d	0	0	0	508	0	0	0	0	0	1	0	0	0	0	0	1	8	0	0	4	1	0	3	0
e	0	0	1	0	426	0	0	0	4	0	0	4	6	27	2	21	4	8	12	0	10	0	3	0
f	0	0	0	0	0	429	0	0	0	0	0	0	0	0	0	1	50	0	0	5	0	32	7	1
g	2	0	0	0	0	0	418	71	0	8	0	0	0	1	8	4	0	0	19	0	0	0	11	0
h	0	0	0	0	0	1	555	0	0	0	0	0	0	0	0	3	3	0	0	0	0	0	0	0
i	0	0	0	0	1	0	0	0	511	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
k	0	0	0	0	0	0	0	0	3	533	3	0	0	0	0	0	0	0	0	0	0	0	1	0
l	0	0	0	0	0	0	0	0	0	0	565	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	0	0	531	0	0	2	7	0	0	0	0	0	0	0	0
n	3	0	1	0	3	0	0	0	3	2	0	375	82	8	15	20	0	7	5	0	0	0	6	0
o	3	0	3	0	4	0	0	0	0	0	0	1	0	461	4	40	0	18	0	3	0	0	0	2
p	22	3	25	0	2	1	2	0	1	5	2	0	5	1	307	84	4	0	0	2	6	0	59	1
q	0	1	0	1	0	0	1	0	0	0	0	0	0	0	16	529	1	0	0	0	0	0	1	0
r	0	0	0	1	0	0	0	0	17	0	0	0	0	0	0	0	707	0	0	39	0	0	10	0
s	0	0	0	0	49	0	0	0	0	0	0	42	24	9	1	18	1	371	15	0	1	0	4	0
t	5	0	0	0	10	0	0	1	0	0	0	56	45	0	2	3	0	0	408	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	511	1	0	0	0
v	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0	0	0	489	32	0	0
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	532	0	2
x	0	0	2	1	0	0	0	0	0	1	0	0	86	0	21	26	1	0	3	28	0	0	368	1
y	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	532	

FIGURE 16: Confusion matrix of all alphabets using leave-one-out strategy for user #4

Predicted Signs

Actual Signs	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	475	0	0	0	0	0	21	0	0	0	0	2	0	0	6	9	0	0	16	0	0	0	7	0
b	16	553	0	2	0	1	0	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	0	0	537	0	3	0	0	0	0	1	0	0	0	3	1	0	0	0	0	4	0	0	0	0
d	0	0	0	532	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
e	9	0	12	0	515	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0	0	0	0
f	0	8	0	0	0	513	0	0	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g	0	0	0	0	0	0	467	47	0	14	0	0	0	0	0	0	0	0	9	0	0	0	0	7
h	0	0	1	0	0	0	6	536	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
i	0	0	0	29	7	1	0	0	491	8	0	0	0	4	0	0	0	0	0	0	0	0	0	3
k	0	0	0	0	0	0	2	0	0	391	0	0	0	0	0	43	0	0	31	302	7	0	1	
l	0	1	0	0	0	0	0	0	0	0	576	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	0	0	0	0	0	1	0	539	33	0	0	1	0	5	3	0	0	0	0	0
n	12	0	0	0	0	0	0	0	0	2	0	9	485	0	0	3	0	4	23	0	0	0	6	0
o	9	2	0	10	1	0	11	16	0	3	12	13	0	439	3	5	0	1	2	0	0	1	0	1
p	3	0	0	0	0	0	4	6	0	8	0	0	0	7	570	23	0	0	5	0	0	0	15	1
q	20	0	0	1	0	0	0	14	0	0	0	0	0	12	493	0	0	0	0	0	0	0	0	0
r	0	1	0	24	0	5	0	0	0	13	0	0	0	0	7	0	499	0	0	0	1	0	0	0
s	4	0	0	2	38	0	0	0	0	2	0	30	15	32	0	0	0	610	16	0	0	0	0	0
t	65	1	0	0	30	0	0	0	0	0	0	108	2	5	0	0	0	44	257	0	0	0	0	2
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	536	1	0	4	0	
v	1	0	0	0	1	13	0	0	0	0	6	5	0	0	1	22	14	4	0	6	535	2	0	18
w	0	0	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0	0	0	5	637	0	0	
x	0	0	1	0	0	17	0	0	0	0	4	0	0	0	2	0	1	0	0	0	0	0	596	1
y	0	0	0	0	0	0	0	0	22	0	0	4	0	0	0	0	6	0	0	0	0	0	0	505

FIGURE 14: Confusion matrix of all alphabets using leave-one-out strategy for user #2

Predicted Signs

Actual Signs	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	475	4	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	16	20	0	0	0	1	0
b	0	533	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	0	0	541	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	543	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
e	0	0	0	0	47	0	0	0	0	0	0	5	0	50	0	0	0	0	438	0	0	0	0	0
f	0	0	0	0	526	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
g	7	0	1	0	0	0	474	21	0	8	0	0	0	1	4	1	0	0	10	0	0	0	8	3
h	0	0	0	0	0	0	59	472	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0
i	0	0	0	0	0	0	0	0	498	0	0	0	0	0	0	0	0	3	0	0	0	0	0	25
k	0	0	0	0	0	0	0	0	14	393	0	0	0	0	0	0	1	0	0	11	151	0	0	0
l	0	0	0	0	0	0	0	0	0	0	515	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	0	7	0	0	0	0	0	0	499	6	0	0	0	0	0	14	0	0	0	0	0
n	2	0	0	0	2	0	2	0	0	0	0	16	403	0	0	0	0	15	91	0	0	0	0	0
o	0	0	0	0	7	0	0	0	0	0	0	0	0	525	0	0	0	1	0	0	0	0	0	0
p	2	0	2	0	2	0	0	0	1	0	0	0	1	0	484	1	0	1	1	0	0	0	16	17
q	0	0	96	0	3	0	0	0	0	0	0	7	25	23	164	198	0	3	0	0	0	0	2	9
r	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	562	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	526	0	0	0	0	0	0
t	0	0	0	9	0	0	0	0	0	0	1	11	0	0	0	3	508	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	530	0	0	0	0	0
v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	528	0	0	0	0
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	514	0	0	0
x	0	0	0	72	0	0	0	0	0	0	0	0	0	0	0	0	0	14	9	0	0	0	428	0
y	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	518

FIGURE 17: Confusion matrix of all alphabets using leave-one-out strategy for user #5

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938829, IEEE Access

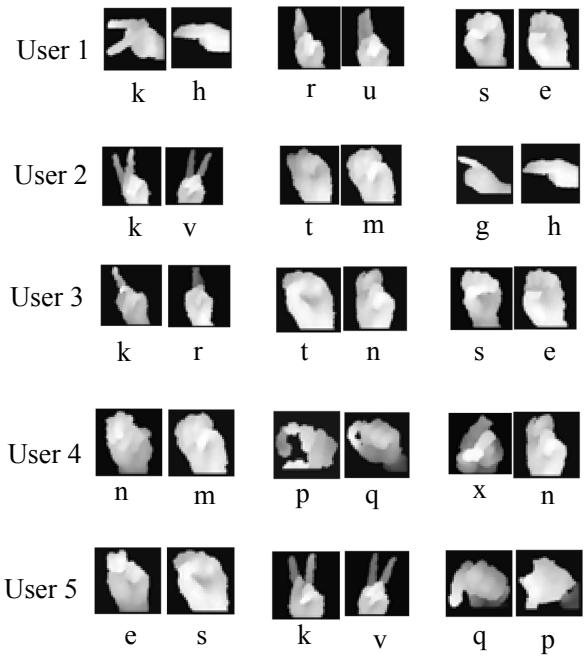

FIGURE 18: Example of confused-pair samples from each user.

V. CONCLUSION

This paper proposes a new efficient method for user independent American fingerspelling recognition based on depth images and PCANet features. Hand segmentation is performed using thresholding operation on the depth image. Depth values of the segmented hand region is normalized to improve the performance. Hand shape features are learned through the efficient PCANet deep learning architecture. Features extracted from depth images can handle cross user differences and image condition variations and thus give a promising results compered to color images. Experimental results show that using single PCANet model is better than using multiple user-specific PCANet models. The proposed system is tested using a public benchmark dataset collected from five different users and give average accuracy of 88.7% using leave-one-out evaluation strategy. The performance of the proposed method outperforms other state-of-the-art methods.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Majmaah University for funding this work under project number No. (RGP-2019-24).

REFERENCES

[1] H. S. Badi and S. Hussein, "Hand posture and gesture recognition technology," Neural Computing and Applications, vol. 25, no. 3-4, pp. 871–878, 2014.

[2] R. Rastgoo, K. Kiani, and S. Escalera, "Multi-modal deep hand sign language recognition in still images using restricted boltzmann machine," Entropy, vol. 20, no. 11, p. 809, 2018.

[3] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," International Journal of Machine Learning and Cybernetics, vol. 10, no. 1, pp. 131–153, 2019.

[4] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," Computer Vision and Image Understanding, vol. 141, pp. 152–165, 2015.

[5] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," Artificial Intelligence Review, vol. 43, no. 1, pp. 1–54, 2015.

[6] W. Tao, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion," Engineering Applications of Artificial Intelligence, vol. 76, pp. 202–213, 2018.

[7] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in Proceedings of the 13th international conference on multimodal interfaces. ACM, 2011, pp. 279–286.

[8] C. Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 44–52.

[9] E.-J. Holden, G. Lee, and R. Owens, "Australian sign language recognition," Machine Vision and Applications, vol. 16, no. 5, p. 312, 2005.

[10] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "A multimodal framework for sensor based sign language recognition," Neurocomputing, vol. 259, pp. 21–38, 2017.

[11] J.-S. Kim, W. Jang, and Z. Bien, "A dynamic gesture recognition system for the korean sign language (ksl)," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 2, pp. 354–359, 1996.

[12] J. Cheng, X. Chen, A. Liu, and H. Peng, "A novel phonology-and radical-coded chinese sign language recognition framework using accelerometer and surface electromyography sensors," Sensors, vol. 15, no. 9, pp. 23 303–23 324, 2015.

[13] D. Warchoł, T. Kapuściński, and M. Wysocki, "Recognition of fingerspelling sequences in polish sign language using point clouds obtained from depth images," Sensors, vol. 19, no. 5, p. 1078, 2019.

[14] S. Aly and S. Mohammed, "Arabic sign language recognition using spatio-temporal local binary patterns and support vector machine," in International Conference on Advanced Machine Learning Technologies and Applications. Springer, 2014, pp. 36–45.

[15] A. A. Ahmed and S. Aly, "Appearance-based arabic sign language recognition using hidden markov models," in Engineering and Technology (ICET), 2014 International Conference on. IEEE, 2014, pp. 1–6.

[16] M. Mohandes, M. Deriche, and J. Liu, "Image-based and sensor-based approaches to arabic sign language recognition," IEEE transactions on human-machine systems, vol. 44, no. 4, pp. 551–557, 2014.

[17] J. Singha, A. Roy, and R. H. Laskar, "Dynamic hand gesture recognition using vision-based approach for human–computer interaction," Neural Computing and Applications, pp. 1–13, 2016.

[18] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in Proc. IEEE RO-MAN: The 21st IEEE Int. Symp. Robot and Human Interactive Communication, Sep. 2012, pp. 411–417.

[19] S. Jadooki, D. Mohamad, T. Saba, A. S. Almazyad, and A. Rehman, "Fused features mining for depth-based hand gesture recognition to classify blind human communication," Neural Computing and Applications, vol. 28, no. 11, pp. 3285–3294, 2017.

[20] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," IEEE transactions on cybernetics, vol. 43, no. 5, pp. 1318–1334, 2013.

[21] S. Aly, B. Osman, W. Aly, and M. Saber, "Arabic sign language fingerspelling recognition from depth and intensity images," in Computer Engineering Conference (ICENCO), 2016 12th International. IEEE, 2016, pp. 99–104.

[22] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," IEEE Transactions on Multimedia, vol. 21, no. 1, pp. 234–245, 2019.

[23] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?" IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 5017–5032, 2015.

[24] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, p. 27, 2011.

[25] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," IEEE Signal Processing Letters, vol. 23, no. 9, pp. 1188–1192, 2016.
[26] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," Communications of the ACM, vol. 56, no. 1, pp. 116–124, 2013.
[27] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2834–2841.
[28] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in Consumer depth cameras for computer vision. Springer, 2013, pp. 119–137.
[29] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011, pp. 1114–1119.
[30] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in European Conference on Computer Vision. Springer, 2012, pp. 852–863.
[31] S.-Z. Li, B. Yu, W. Wu, S.-Z. Su, and R.-R. Ji, "Feature learning based on sae–pca network for human gesture recognition in rgbd images," Neurocomputing, vol. 151, pp. 565–573, 2015.
[32] C. Zhang and Y. Tian, "Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition," Computer Vision and Image Understanding, vol. 139, pp. 29–39, 2015.
[33] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," IEEE transactions on multimedia, vol. 17, no. 1, pp. 29–39, 2015.
[34] Arif-Ul-Islam and S. Akhter, "Orientation hashcode and articial neural network based combined approach to recognize sign language," in 2018 21st International Conference of Computer and Information Technology (ICCIT), Dec 2018, pp. 1–5.
[35] B. Kang, S. Tripathi, and T. Q. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map," in 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). IEEE, 2015, pp. 136–140.
[36] S. Ameen and S. Vadera, "A convolutional neural network to classify american sign language fingerspelling from depth and colour images," Expert Systems, vol. 34, no. 3, p. e12197, 2017.
[37] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," Neural Computing and Applications, vol. 28, no. 12, pp. 3941–3951, 2017.
[38] K. Otiniano-Rodríguez, E. Cayllahua-Cahuina, G. Cámara-Chávez et al., "Finger spelling recognition using kernel descriptors and depth images," in 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images. IEEE, 2015, pp. 72–79.
[39] X.-X. Niu and C. Y. Suen, "A novel hybrid cnn–svm classifier for recognizing handwritten digits," Pattern Recognition, vol. 45, no. 4, pp. 1318–1325, 2012.
[40] S. Aly and A. Mohamed, "Unknown-length handwritten numeral string recognition using cascade of pca-svmnet classifiers," IEEE Access, vol. 7, pp. 52 024–52 034, 2019.
[41] G. Modanwal and K. Sarawadekar, "A robust wrist point detection algorithm using geometric features," Pattern Recognition Letters, vol. 110, pp. 72–78, 2018.
[42] A. I. Maqueda, C. R. del Blanco, F. Jaureguizar, and N. García, "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns," Computer Vision and Image Understanding, vol. 141, pp. 126–137, 2015.
[43] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," Journal of machine learning research, vol. 9, no. Aug, pp. 1871–1874, 2008.
[44] T. Joachims, "Training linear svms in linear time," in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006, pp. 217–226.
[45] A. Kuznetsova, L. Leal-Taixé, and B. Rosenhahn, "Real-time sign language recognition using a consumer depth camera," in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 83–90.
[46] W. Nai, Y. Liu, D. Rempel, and Y. Wang, "Fast hand posture classification using depth features extracted from random line segments," Pattern Recognition, vol. 65, pp. 1–10, 2017.