Received April 5, 2020, accepted April 23, 2020, date of publication April 27, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990699

DeepArSLR: A Novel Signer-Independent Deep Learning Framework for Isolated Arabic Sign Language Gestures Recognition

SALEH ALY, (Associate Member, IEEE), AND WALAA ALY

Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Majmaah 11952, Saudi Arabia
Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt

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

This work was supported by the Deanship of Scientific Research at Majmaah University under Grant 1439-51.

ABSTRACT Hand gesture recognition has attracted the attention of many researchers due to its wide applications in robotics, games, virtual reality, sign language and human-computer interaction. Sign language is a structured form of hand gestures and the most effective communication way among hear-impaired people. Developing an efficient sign language recognition system to recognize dynamic isolated gestures encounters three major challenges, namely, hand segmentation, hand shape feature representation and gesture sequence recognition. Traditional sign language recognition methods utilize color-based hand segmentation algorithms to segment hands, hand-crafted feature extraction for hand shape representation and Hidden Markov Model (HMM) for sequence recognition. In this paper, a novel framework is proposed for signer-independent sign language recognition using multiple deep learning architectures comprising hand semantic segmentation, hand shape feature representation and deep recurrent neural network. The recently developed semantic segmentation method called DeepLabv3+ is trained using a set of pixel-labeled hand images to extract hand regions from each frame of the input video. Then, the extracted hand regions are cropped and scaled to a fixed size to alleviate hand scale variations. Extracting hand shape features is achieved using a single layer Convolutional Self-Organizing Map (CSOM) instead of relying on transfer learning of pre-trained deep convolutional neural networks. The sequence of extracted feature vectors are then recognized using deep Bi-directional Long Short-Term Memory (BiLSTM) recurrent neural network. BiLSTM network contains three BiLSTM layers, one fully connected and softmax layers. The performance of the proposed method is evaluated using a challenging Arabic sign language database containing 23 isolated words captured from three different users. Experimental results show that the performance of proposed framework outperforms with large margin the state-of-the-art methods for signer-independent testing strategy.

INDEX TERMS Arabic sign language recognition, deep learning, hand semantic segmentation, convolutional self-organizing map, signer-independent, deep BiLSTM network.

I. INTRODUCTION

Hand gestures are commonly used among people to convey their thoughts and feelings [1]. Hearing-impaired persons always relied on sign language to communicate among each others. However, most of the normal people are not aware of such language and face difficulties to communicate with deaf. Therefore, developing an automatic sign language recognition system helps to facilitate this communication and decrease the gap. The structured style of hand gestures in sign language helps to facilitate non-verbal communication among deaf and hearing-impaired people. Sign languages involve many vocabularies/words and has complex structure similar to oral languages.

Gestures of sign language are usually expressed by the combination of hand shapes, position, orientation, movements and facial expressions [2]. Sign language recognition problem can be divided into two categories, namely, static gesture recognition which focus on fingerspelling, and dynamic recognition which related to isolated words and

The associate editor coordinating the review of this manuscript and approving it for publication was Huazhu Fu.

continuous sentence recognition. Many continuous sign language recognition systems utilize an extended version of isolated words framework to recognize the whole sentence [3].

Practically, signer-independent dynamic sign language recognition systems encounter three challenges: (1) hand segmentation/detection, (2) hand shape feature representation, and (3) sequence classification. Most existing methods assume that hands are segmented or even ignore the hand segmentation step. The first challenge comes from the difficulties of detecting hand regions since hands are very articulated object and their shape and appearance changes dramatically from person to person and with hand motions. In addition, hand segmentation is an essential step to find the gesture region-of-interest and to build an efficient signer-independent sign language recognition system [4]. Developing an efficient hand segmentation algorithm not only improve the performance of the system but also make it run naturally without any need for special gloves. Existing hand segmentation algorithms are based on the modeling and classification of skin color. Skin color-based hand segmentation is the dominant technique before the widespread of deep convolutional neural networks. Hand detection problem can be precisely solved using dense prediction of every pixel that belongs to a hand (i.e., binary segmentation). Semantic hand segmentation is different from hand detection as it assigns one label to each pixel of the hand region [5]. This paper tackles hand segmentation using a new developed semantic segmentation deep convolutional neural network called DeepLab [6]. DeepLabv3+ [7] which is a recent variant of DeepLab network showed an extraordinary success in many object segmentation problems. The main characteristics of this network which make it appropriate for hand segmentation is its use of atrous spatial pyramid pooling module to encode multi-scale contextual information and its encoder-decoder structure which capture sharp hand boundaries by gradual recovery of spatial information. To the best of our knowledge this is the first work which exploits DeepLabv3+ [7] as a hand segmenter in sign language recognition problem.

It is obvious that, each sign composes of a set of frames expressing the hand shape primitives of the sign. However, similar frames may be appeared in different signs which cause the second challenge in building automatic sign recognition system. This challenge resulted from the ambiguity caused by giving same label for all frames in the same sign which will cause confusion in the supervised learning algorithm. Therefore, using a separate unsupervised feature learning module overcomes this problem by representing hand appearance features in each frame before performing sequence classification. The third challenge related to the existing temporal variations among signer caused by various performing styles of signs which results in different number of frames. Besides, the variations in the length of each gesture leads to misalignment problem. The recent development in recurrent neural network especially Long Short-Term Memory (LSTM) model can be used for hand gesture sequence modeling. LSTM not only able to absorb the temporal variations in the gestures but also can learn the dependency between sign primitives which further improve sign classification.

Recently, various deep convolutional neural networks are developed to tackle sign language recognition problem [8]–[12]. Although end-to-end supervised deep learning architectures exhibit great success in dynamic sign language recognition, it requires large amount of labeled training data to jointly learn features and classifier. It is known that separating feature extraction step from classification can mitigate this problem. In this paper, we propose an Arabic sign language recognition framework using a combination of three different deep learning architectures. Since hand is considered as the main region-of-interest which is essentially used to perform all signs, the goal of the first module is to segment hands in all frames of the input video. Hand semantic segmentation helps to focus on every pixel of the hands and eliminate background pixels which decrease the inter-class variations among signers. Each pixel of the hand is segmented using DeepLabv3+ semantic segmentation model [7]. A set of images contained labeled hand pixels is used to train the model based on Resnet-50 convolutional neural network. The trained model is utilized as a hand segmenter for all images in the video sequences. Then, hand regions are cropped and scaled to a fixed size to alleviate scale variations. A single layer convolutional SOM is trained for hand shape feature extraction [13]. Finally, the sequence of feature vectors extracted from the input video is classified using deep bi-directional long short-term network. The recurrent classification network consists of three Bi-directional LSTM layers followed by single fully connected and soft-max layers. The network is trained using adaptive moment gradient descent algorithm to recognize all gestures. The contributions in this paper are as follows:

1) A new framework is developed for signer-independent isolated Arabic sign language gesture recognition based on the combination of semantic segmentation network, convolutional SOM and deep Bi-directional LSTM network.
2) A hand segmentation module is proposed using DeepLabv3+ semantic segmentation network.
3) An unsupervised single layer convolutional SOM model is presented for efficient hand shape representation.
4) A new deep classification network containing three Bi-directional LSTM layers followed by single fully connected and softmax layers is utilized for gesture sequence classification.
5) The proposed framework is evaluated using real Arabic sign language database for signer independent scenario.

The rest of the paper is organized as follows: Section II reviews the works related to sign language recognition, Section III explains the details of proposed framework processing modules. Section IV reports the experimental results. Finally, Section V draws the conclusion.

II. RELATED WORKS

Sign language is an effective and natural way to communicate between hearing-impaired/mute people [1]. With the rapid increase in the multimedia communication systems, sign language became an interesting topic for the researchers to enhance the social communication between deaf people. Sign language recognition systems almost relied on either sensor or vision-based approaches. In vision-based, either depth or RGB cameras are used to capture depth or color images/videos, respectively [14]. Various machine learning algorithms are developed to process and classify the video data. While in the sensor based approach, different types of sensors are embedded in an especially designed electromechanical gloves to capture the data [15].

Plenty of research works have developed to solve various sign language recognition tasks for different languages [14], [16]–[18]. The Chinese sign language recognition systems was reviewed in [17]. Xiao *et al.* [19] utilized dual Long Short-Term Memory (LSTM) and a Couple Hidden Markov Model (CHMM) to fuse hand and skeleton sequence information. Hand segmentation is achieved using power rate transforms and fusion of RGB-D image. In [14], American sign language alphabet recognition was developed by extracting features from depth images using PCANet. Joy *et al.* [20] developed a quiz-based tool to learn finger spelled signs for Indian sign language. Arabic language recognition became a hot topic nowadays [21]. Several isolated Arabic sign gesture recognition has developed using various feature extraction and classification methods such as: accumulated difference images with DCT [22], Local Binary Patterns (LBP) with Hidden Markov Models (HMM) [23] and spatiotemporal LBP with Support Vector Machine (SVM) [24]. In [25], a hybrid Gaussian skin model and a region-growing technique were developed to segment the face and hands then HMM was proposed to recognize isolated Arabic signs. Al-Rousan *et al.* [26] proposed a recognition system of 30 isolated Arabic words using Discrete Cosine Transform (DCT) and Hidden Markov Models (HMMs). Their experiments was conducted in different modes including online, offline, signer-dependent, and signer-independent.

Isolated sign language gesture recognition systems usually involves three main steps, hand segmentation, feature extraction and sequence classification. These systems relied on different methods for hand segmentation to extract hand regions from each frame of the input video. Dahmani and Larabi [27] developed a neural network that exploit skin color and texture attributes to segment hand from complex backgrounds. In [22], hand segmentation was achieved with the help of color gloves to localize both hands of the signer. Then, zonal coding selection of DCT coefficients was used to extract the feature and finally polynomial classifier was used to classify the isolated Arabic signs.

The superiority of various deep learning architectures in solving many complex computer vision problems make them predominant. Recently, many researchers have employed different deep neural networks for static and dynamic sign language recognition problems [4], [8]–[12], [28]–[30]. Molchanov *et al.* [31] utilized a recurrent 3D convolutional neural network for simultaneous detection and classification of dynamic hand gestures from multimodal data captured by depth, color and stero-IR sensors. They trained 3D-CNN to extract local spatial and temporal features from short clips, then features are fed into a recurrent network which aggregates transitions across clips. The hidden state of the current clip is input into a softmax layer to estimate class-conditional probabilities using connectionist temporal classification as a cost function. Liao *et al.* [32] developed a deep 3-dimensional residual ConvNet and bi-directional LSTM networks for dynamic sign language recognition. Hand object was localized in the video frames using faster R-CNN, then a 3D ResNet jointly extracts spatial and temporal features from the input image sequences which classified using bi-directional LSTM. Jie *et al.* [33] proposed a Hierarchical Attention Network with Latent Space (LS-HAN) framework for continuous sign recognition. Their method aimed to eliminate temporal segmentation of words. LS-HAN contains three components, namely, two-stream Convolutional Neural Network (CNN) for video feature representation, a Latent Space (LS) to bridge semantic gap, and a Hierarchical Attention Network (HAN) for recognition. Huang *et al.* [34] presented an attention-based 3D-convolutional neural networks (3D-CNNs). This model can learn spatial and temporal features from raw video and the attention mechanism helps to focus on the areas of interest. After feature extraction, temporal attention was utilized to select the significant motions for classification. Cui *et al.* [11] adopted two deep neural networks modules, stacked temporal fusion module are utilized for feature extraction and bi-directional recurrent neural networks module was employed for sequence modeling. An iterative optimization technique was adopted to train the end-to-end model for sequence alignment proposal. The alignment proposal was directly utilized as a supervisory information to tune the feature extraction module to further improve network performance.

Recently, a great success have achieved in hand segmentation and activity recognition using Convolutional Neural Network (CNN). Semantic segmentation methods based on deep convolutional neural networks such as: fully convolutional neural network [35], SegNet [36], RefineNet [37], U-Net [38] and DeepLab [6] showed an extraordinary success in many object segmentation problems. Bambach *et al.* [39] applied CNN and GrabCut to detect and segment hand with the help of a set of candidate hand bounding boxes which were resulted from skin-based model. A depth-adaptive deep neural network using in-layer multiscale neurons was proposed in [40] for hand segmentation of RGBD images. A RefineNet deep network was fine-tuned in [41] for hand segmentation. Hand maps produced by the RefineNet were then used for hand activity recognition. Recurrent U-Net architecture was developed by Wang *et al.* [42] to run in a resource-constrained environment with limited computational power. Recently, DeepLab model was developed by

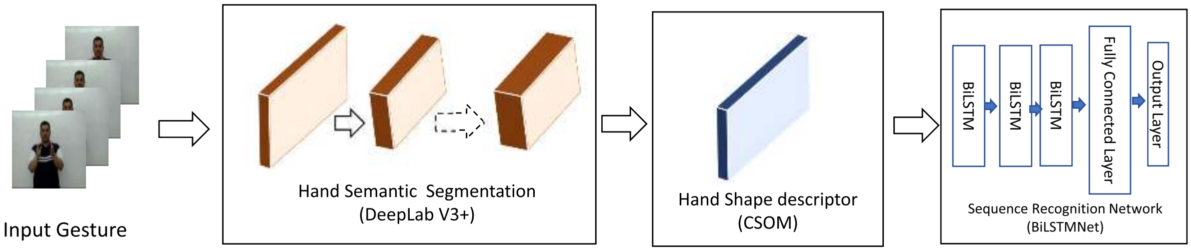
Input Gesture
Hand Semantic Segmentation (DeepLab V3+)
Hand Shape descriptor (CSOM)
BiLSTM
BiLSTM
BiLSTM
Fully Connected Layer
Output Layer
Sequence Recognition Network (BiLSTMNet)

FIGURE 1. Proposed framework of dynamic sign language recognition using DeepLabv3+ semantic Segmentation, convolutional SOM feature extraction and bi-directional LSTM network.

Chen *et al.* [6] to solve image segmentation problems. This model has three key features, namely, dilated convolution, Atrous Spatial Pyramid Pooling (ASPP) and improved localization object boundaries using probabilistic graphical models. Subsequently, Chen *et al.* proposed DeepLabv3 [43] and DeepLabv3+ [7] which combine cascade and parallel modules of dilated convolutions. DeepLab model has expressed excellent object segmentation results for PASCAL VOC challenge while its capability for hand segmentation has not yet been validated.

Sequence modeling using Hidden Markov Models (HMM) are commonly used in speech recognition, text classification and action recognition. HMM can model the transition between sequence states but the model has a limited factor with the size of context window that make the model computationally impractical for processing long range dependencies. Recently, Recurrent Neural Networks (RNN) considered to be the preferred learning model for sequence modeling [44]. RNN has better performance over HMM as it has larger memory and computational capacity [45]. Recurrent neural network has the ability to process sequential data with varying feature length which made it successful in different sequential fields such as speech recognition [46] and video processing [47]. Traditional RNNs have vanishing gradient problem, which leads to problems when processing long-term dependencies in data. These models can not remember the old values existed early in the sequence [48]. A Long Short-Term Memory (LSTM) was developed to overcome this problem by modifying the architecture of RNN. The basic structure of LSTM was proposed in [49]. There are various versions of LSTM that differs in the structures of LSTM [47]. The LSTM with peepholes and forget gates was existed in [50] while the Gated Recurrent Unit (GRU) was used in [51]. The depth gated RNNs was presented in [52]. Bidirectional LSTM was developed in [53] to learn time dependencies of the sequence from both forward and backward information. A combination of bi-directional LSTM, CNN, and CRF was developed for sequence labeling in [54]. Bidirectional LSTM has shown a successful performance in action recognition [55] and sign language recognition [19], [56], [57].

III. PROPOSED FRAMEWORK OF SIGNER-INDEPENDENT SIGN LANGUAGE RECOGNITION

The proposed framework for signer-independent sign language recognition comprises three different modules, namely, hand semantic segmentation and reprocessing, hand shape feature extraction and sequence classification. The block diagram of proposed framework is shown in Fig. 1. Instead of using complex hand detection techniques to locate hand regions in each frame of the input video, this framework exploits recent deep semantic segmentation network to locate every pixels in the hand region. After segmenting hand from every frame of the input video sequence, hand regions are cropped and scaled into a fixed resolution of 64 × 64 pixels. Hand shape representation is learned using a simple Convolutional Self-Organizing Map (CSOM) network. The proposed single layer CSOM utilizes unsupervised learning algorithm to learn shape features of hand without any need to label the data. Each frame of the input video will be converted into a feature vector, then the feature vectors of all frames in the input video are aggregated to create the feature representation of the sign. The temporal dependency in the video sequences are modeled using Bi-directional Long Short-Term Memory network (BiLSTM). The final layer of the BiLSTM network is trained to classify each gesture. The following subsections explain the details of proposed framework.

A. HAND SEMANTIC SEGMENTATION USING DeepLabv3+

Recently, DeepLab semantic segmentation model achieved promising results for various visual object segmentation problems [58]. Numerous improvements have been done to enrich the model, including DeepLabv2 [6], DeepLabv3 [43] and the most recent DeepLabv3+ [7]. Generally, DeepLabv3+ network is composed of two phases, namely, encoding and decoding. The encoding phase aims to extract discriminative information from the input image using a pretrained convolutional neural network as a backbone network. Convolutional layers of backbone CNN look for different features in an image and pass this information to subsequent layers. The information extracted in the encoding phase is used in the decoding phase to reconstruct pixel labeled output image with same dimension of input image.

Since the variations in hand size and shape are highly affect the quality of segmentation results, using multi-scale context information can alleviate this problem. The spatial pyramid pooling module in DeepLabv3+ can effectively encode multi-scale contextual information of hand. This is achieved using pooling operations of the incoming features at various field-of-view using the so called atrous convolution. DeepLab introduced the concept of atrous convolutions which is a generalized form of the convolution operation. A parameter called rate (r) is use in atrous convolutions to explicitly change the effective field-of-view of the convolutional filters. atrous convolutions operation can be generalized as follows:

$$y[i] = \sum_{k} x[i + r.k]w[k] \tag{1}$$

where the atrous rate r determines the stride with which input image is sampled. It can be inferred that standard convolution is a special case of atrous using rate r = 1. The field-of-view for convolutional filters can be deceptively modified by changing the rate value as shown in Fig. 2. Spatial pyramid pooling with atrous convolutions is firstly proposed and applied in DeepLab to create a new block called Atrous Spatial Pyramid Pooling (ASPP). Four parallel operations are utilized in ASPP including 1 × 1 convolution and three 3 × 3 atrous convolution with 6, 12 and 18 rates.

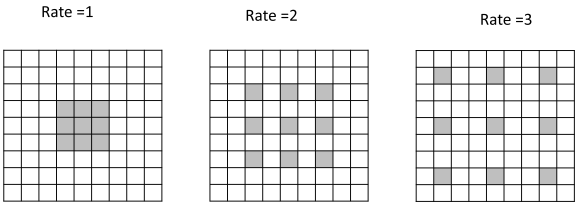

FIGURE 2. Atrous convolution with kernel size 3 × 3 and rates 1, 2, and 3. Employing large value of atrous rate enlarges the model field-of-view and enables hand encoding at multiple scales.

A simplified diagram of DeepLabv3+ architecture is shown in Fig. 3. DeepLabv3+ hand segmentation network utilizes a pretrained model Resnet-50 [59] trained on ImageNet dataset [60] as its backbone network. DeepLabv3+ remove the striding in the last convolutional block of Resnet-50 backbone network and apply atrous convolution with stride 2 to preserve feature map resolution. The resolution of the final block of Resnet-50 backbone network is 16 times smaller than the input image size. Then, Atrous Spatial Pyramid Pooling (ASPP) module containing four parallel atrous convolutional layers is added after the final feature maps of the Resnet-50 to extract multi-scale features. The multiscale features are stacked together to produce the final feature maps of the encoder subnetwork. The encoder features resulted from ASPP module are upsampled by a factor of 4 using transposed convolution operation and then concatenated with the corresponding feature maps from the Resnet-50 backbone network which have similar spatial resolution. After concatenation, multiple 3 × 3 convolutions are employed to refine the features followed by another dilated convolution which upsample the feature maps by factor of 4 to reach the same resolution of the input image.

B. UNSUPERVISED HAND SHAPE REPRESENTATION USING SINGLE LAYER CONVOLUTIONAL SELF-ORGANIZING MAP

Convolutional Self-Organizing Map (CSOM) module is utilized for hand shape feature extraction [13]. It includes three successive processing layers, namely, contrast normalization, convolutional SOM and local histogram output layer which shown in Fig. 4. Every local patch of the input image is normalized using contrast normalization layer which subtract each patch from its corresponding mean and divide by its corresponding standard deviation. The trained 2-dimensional SOM feature map is utilized to map every normalized patch into the index of best matching neuron. Similarity between every local image patch and all neuron centroids is calculated using Euclidean distance measure. SOM mapping operation will convert the input image into a feature index image in which each local patch is represent by the index of best matching neuron. Similar patches will be mapped into adjacent neurons thanks to the topographic order of neurons in SOM feature maps. The quantization process of SOM will also help to absorb the distortions of local patches and create invariant feature representation. The key advantage of using SOM instead of the commonly used K-means clustering algorithm lies in the topological order of neurons.

Assume that we have an input image I, The image is divided into a set of overlapped local image patches of size $k \times k$. The brightness and contrast of the collected local image patches are normalized using Z-score normalization. Which mean that, for each local patch we subtract the patch mean and then divide it by the standard deviation. Suppose the un-normalized local patches denoted as $\hat{x}_i$, the patches are normalize by applying the following equation:

$$x_i = \frac{\hat{x}_i - mean(\hat{x}_i)}{\sqrt{var(\hat{x}_i)} + \epsilon} \tag{2}$$

where normalized patches are denoted as x_i, *mean* and *var* are the mean and variance of the un-normalized local patches, respectively, and ϵ is a small number used to avoid dividing by zero.

1) SOM TRAINING ALGORITHM

Kohonen learning algorithm used to train the Self-Organized Map (SOM) is considered as one of the most successful competitive learning algorithm which approximate multidimensional input space into a set of fixed neurons [61]. The trained SOM can be viewed as a non-linear approximation of principal component analysis algorithm since each map dimension can be considered as a nonlinear principal component. SOM training algorithm utilizes a specified neighborhood function to organize neuron positions in the map which make it distinctive compared with other non-topological clustering algorithms. This process will strongly help to learn the distribution of data while avoiding outlier samples.

Sequential and batch learning algorithms are commonly used to train SOM. However, batch learning resulted in faster and more efficient map. Assume that training samples $X^{N \times d} = \{x_i | x_i \in R^d, i = 1, \ldots, N\}$, where N and d denotes the total number of samples and the input dimensions, respectively. Also, consider the total number of neurons in the 2D map is L, learning rate is α and σ represents the radius of neighborhood function. Every neuron c_j in the SOM is

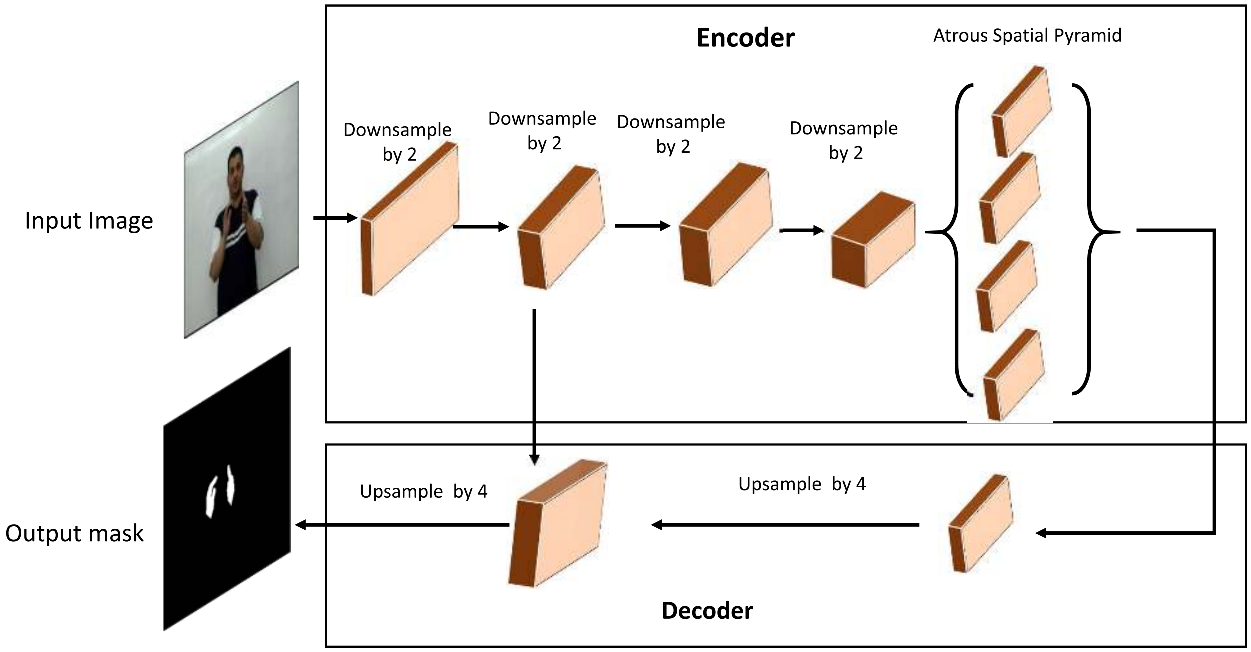

FIGURE 3. Architecture of DeepLabv3+ based on encoder-decoder structure with atrous spatial pyramid pooling.

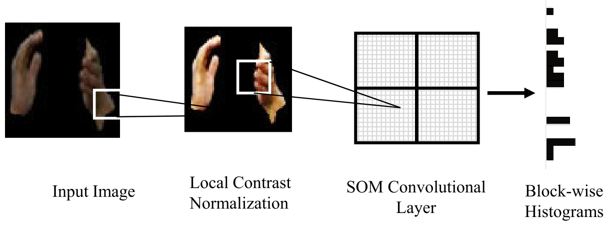

FIGURE 4. Block diagram of single layer convolutional self-organizing map.

associated with a mean vector m_j where $j = 1, .., L$. Random values are commonly assigned as initial values for the L neurons. However, linear initialization using 2-dimensional plane spanned by the largest two principal components of training samples is recommended to speed up the ordering and convergence of SOM training. SOM training algorithm repeats two main steps, namely, competition and update for a fixed number of epochs.

Competition Step: The first step in SOM learning algorithm is to find the best matching unit for each input sample. Every neuron in the SOM layer has a specific location in the map and associated with a mean vector. Using Euclidean distance measure, we can locate the winner neuron c by minimizing the distance between each image patch $x(t)$ and all neurons in the SOM as follows.

$$c = arg \min_i \|(x(t) - m_i(t))\|_2^2 \tag{3}$$

SOM Update: In each epoch of batch learning, SOM map is updated once for all samples of training dataset X. Eq.(3) is applied to compute the winning neuron c for each sample. The centroid vectors of all neurons ($i = 1, \ldots, L$) are updated as follows:

$$m_i(t+1) = \frac{\sum_t h_{ci} x(t)}{\sum_t h_{ci}} \tag{4}$$

where $h_{ci}(t)$ represents the neighborhood function of i^{th} neuron and winning neuron c. The winning neuron c and all its neighbor neurons are modified as shown in Eq. (4). The amount of changes for each neuron depends on the specified neighborhood function h_{ci} which plays an important role in self-organization. Radial basis function with radius σ is commonly used as a neighborhood function.

$$h_{ci} = \alpha(t) e^{(-\|c-i\|^2)/2\sigma^2(t)} \tag{5}$$

where $\alpha(t)$ is a monotonically linear decreasing function of t which change the learning rate, i and c represent the coordinate of i and c-winner neurons in the map. The neighborhood value is gradually decreased according to the neighborhood radius function $\sigma(t)$, its value is calculated for each iteration t using the following equation.

$$\sigma(t) = \sigma_i + \frac{t}{T}(\sigma_f - \sigma_i) \tag{6}$$

where T denotes the number of epochs, the initial and final radius parameters are denoted as σ_i, σ_f, respectively. The neighborhood radius function ($\sigma(t)$) is linearly decreased with t. The initial value of radius σ_i is chosen to be large at the beginning of training in order to develop the topological order of the neurons and gradually decreased to small value σ_f at the end of training.

2) LOCAL PATCH MAPPING USING CSOM

After training, the previous competition step is utilized again to map every local patch of the input image into the index of best matching neuron. This step will produce a feature index image (F) in which all patches are mapped. Given an image I of size $n \times n$, the resulted feature index image of size $(n - k + 1) \times (n - k + 1)$ is calculated. Shape representation of the patch at coordinates (i, j) in the input image will be encoded by the index position of winner neuron in SOM grid. The hard quantization process of the competition step will strongly help to tolerate for invariant feature representation of hand shape.

3) OUTPUT LAYER

The generated feature index image (F) is divided into a set of overlapped blocks with size $b \times b$. Feature representation of the hand shape can be determined by calculating histograms of all overlapped blocks in the feature index image (F). All local spatial histograms are concatenated to create the final hand shape feature vector. Exploiting local spatial histograms for representation help to increase the degree of invariance as same as the effect of pooling layer in the deep learning convolutional neural networks. In addition, using overlapped blocks helps to increase the robustness of the features to translation variations. The computed final feature vectors of all training video sequences are fed into a deep bi-directional LSTM network for video classification.

C. ISOLATED GESTURE RECOGNITION USING DEEP BI-DIRECTIONAL LONG SHORT-TERM MEMORY NETWORK

Since videos have sequence of frames, LSTM layer is employed to learn the sequential features from the frames. The main unit of LSTM hidden layer is the memory block, which contains both memory cell and gate units. Each memory cell has self-connected linear unit called cell state, which keeps the state over time. Gate units are used to control the flow of information inside the memory cell. Fig. 5 illustrates a single memory cell of LSTM. It is clear that LSTM contains three separate gates; forget gate f, input gate i, output gate o. To remove information from the cell state, f_t gate is used to decide what information has to be thrown from the cell state. While to add new information into the cell state, two outputs are combined to decide what information has to be stored in the cell state; i_t gate that decides the updated values and tanh function $\tilde{c}_t$ that develop a vector of new election values. Finally, to output the information from the cell, two values are multiplied; the output gate o_t that decides which part of the cell will be out and tanh function which modifies the values of c_t to be in the range $[-1, 1]$.

The equations from Eq. 7 to Eq. 12 explain functions performed by LSTM cell, where h indicate the hidden state, W denotes the weight matrix and b is the bias vector.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{7}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{8}$$
$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{9}$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{10}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{11}$$
$$h_t = o_t * \tanh(c_t) \tag{12}$$

The basic LSTM has some limitation in its behavior because it takes only information from past sequence. However, accessing both past and future context leads to better accuracy in sign recognition task. Bi-directional LSTM (BiLSTM) provides a good solution to this problem as it captures past and future information. In this paper, we employed bi-directional LSTM to learn bi-directional long-term dependencies between time step of sequence sign data [53], [62]. In bi-directional LSTM, two LSTM layers are trained on the input sequences to provide an additional information from the input and give faster results. The first LSTM trained on the input sequence using forward temporal information, while the second one trained on the reversed copy of the input sequence. The output from the forward and backward will be concatenated to model the bi-directional dependency of the sequence. More layers can be added to create a deep structure in which each layer can either receive the sequence output from previous layer or the last one. Fig. 6 illustrates the structure of one bi-directional LSTM layer. The performance of BiLSTM is increased by increasing depth of the network. In our model, three BiLSTM layers with different number of hidden states are stacked to create BiLSTM network namely BiLSTMNet. In addition BiLSTM layers can be replaced by LSTM layers to create another network called LSTMNet. These networks are followed by one fully connected and softmax layers for sequence recognition. The deep structure of both BiLSTMNet and LSTMNet helps to capture abstract features of sequence data.

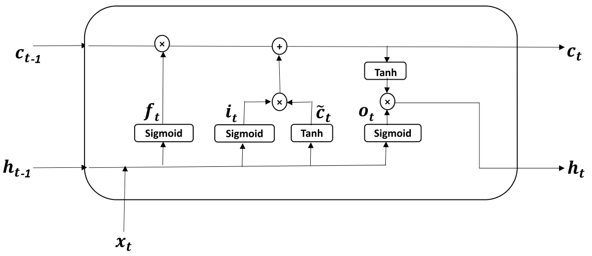

FIGURE 5. Structure of one LSTM cell adapted from [63].

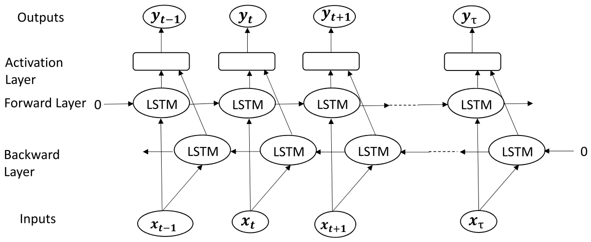

FIGURE 6. Structure of one bi-directional LSTM layer used to construct BiLSTM network.

IV. EXPERIMENTAL RESULTS

The performance of proposed framework is evaluated using real Arabic sign language database for signer-independent testing strategy. Hand regions of the input video sequences are segmented using DeepLabv3+ semantic segmentation network. Then, hand regions of the segmented images are cropped and resized into 64 × 64 pixels. In addition, to avoid skin color variations among signer, we convert all cropped color images into grayscale. Hand shape feature representation is computed using 2-dimensional convolutional SOM. Finally, deep BiLSTM network is utilized to classify each input video. The following experiments are conducted to find the optimal parameters of Deeplabv3+, CSOM and BiLSTM networks which achieve the highest signer-independent recognition accuracy.

A. ISOLATE WORDS ARABIC SIGN LANGUAGE DATABASE

All experiments in this paper are conducted using the Arabic Sign Language (ArSL) database reported in [64]. The ArSL dataset contains 23 isolated Arabic word signs performed by three different users. Some examples of the Arabic gestures performed by each user are shown in Fig. 7. There is no restriction in clothing or image background when recording gestures from each signer. It is noticed from the images shown in Fig. 8 that the database exhibits several variations in illumination, pose, scale, shape, position, clothes and temporal for same sign performed by the three different signers. The list of all recorded Arabic word signs and their English meaning is shown in Table. 1. Videos are recorded at 25 fps with 320 × 240 resolution. Every signer repeated each gesture 50 times in three different sessions which gives a total of 150 sequence for each of the 23 gestures. The total number of video clips reaches 3450. All gesture were temporally partitioned into short sequences. Fig. 9 shows the temporal variations among signs and users. It is clear that first user has long sequences compared to those of second and third users. Moreover, the average number of frames per sign for user second and third users are similar. In the experiments below, the performance of the proposed framework is evaluated using signer-independent testing strategy in which all signs from two users are used in training and other signs from the third user are used for testing. This process is repeated to calculate the accuracy for each user individually. The final accuracy is calculated as the average of the three users' accuracies.

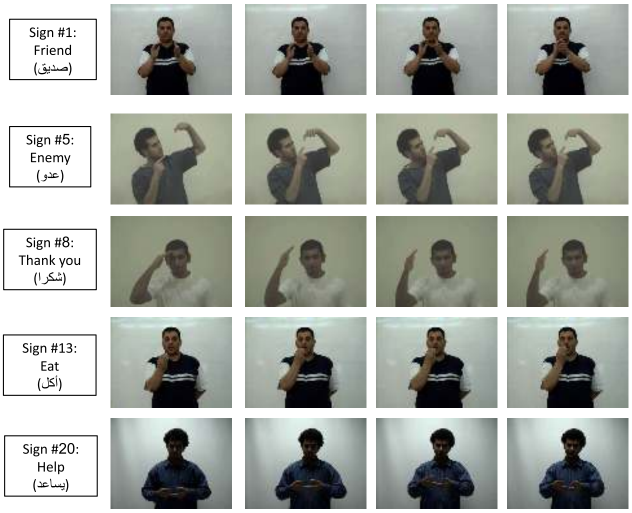

FIGURE 7. Example of gestures from the ArSL database.

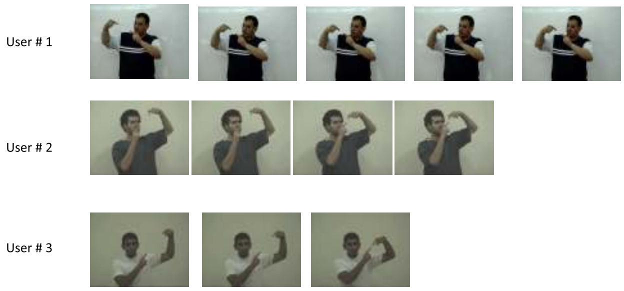

FIGURE 8. Example of various illumination, pose, scale, shape, position, clothes, and temporal variations for the same sign performed by the three signers.

TABLE 1. List of the recorded Arabic sign words with their English meaning.

#	Arabic word	English meaning	#	Arabic word	English meaning
1	صديق	Friend	13	اكل	Eat
2	جار	Neighbor	14	نام	Sleep
3	ضيف	Guest	15	يشرب	Drink
4	هديه	Gift	16	يستيقظ	Wake up
5	عدو	Enemy	17	يسمع	Listen
6	السلام عليكم	Peace upon	18	يسكت	Stop talking
7	اهلا و سهلا	Welcome	19	يشم	Smell
8	شكرا	Thank you	20	يساعد	Help
9	تفضل	Come in	21	امس	Yesterday
10	عيب	Shame	22	ذهب	Go
11	بيت	Home	23	اتى	Come
12	انا	I/me			

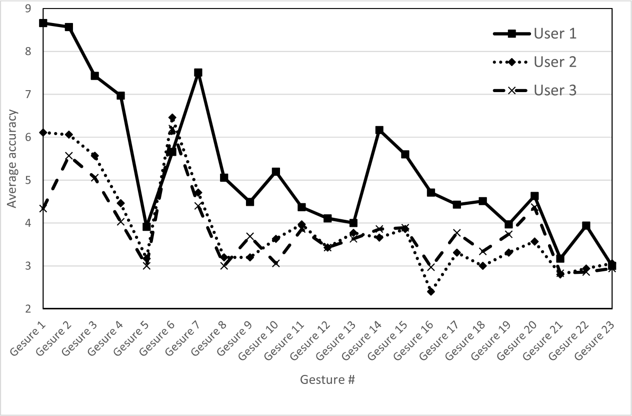

FIGURE 9. Average number of frames per gesture for each user.

B. EXPERIMENTS ON HAND SEGMENTATION

In this experiment, we examine the performance of DeepLabv3+ to segment hands. Resnet-50 is utilized as a backbone network to learn hand shape features in DeepLabv3+ model. The backbone network was initially trained on Imagenet dataset while a set of hand labeled images are employed for transfer learning. We manually labeled hand regions for one gesture sequence from each user for all the 23 signs to create our labeled ArSL data for DeepLabv3+ model learning. In addition, The created labeled dataset is augmented with another dataset called HandOverFace [5] to create a heterogeneous dataset for training consists of 610 images. All images are resized into size of 256 × 256 pixels. Stochastic gradient descent with momentum (SGDM) optimizer is employed to train the network with the following parameter setting: initial learning rate set to 0.001 and decrease linearly every 100 epochs with rate 0.1, momentum equals 0.95, the mini batch size used to update learnable parameters of the network is 20 with number of epochs equal 200.

In this experiment, the labeled ArSL dataset is divided into two disjoint training and testing data. All labeled signs from the first two users are used for training where images from the third user are used for testing. The training images are augmented with the labeled HandOverFace dataset. To evaluate the performance of semantic segmentation module, Mean Intersection-over-Union (MIoU) is calculated for all test images. The obtained MIoU reaches 94.2% which confirm the robustness of the model. Fig. 10 shows examples of hand segmentation results for one gesture from each user. Results prove the robustness of DeepLabv3+ model to accurately capture the boundaries of hand regions even when hands are connected or touching each other.

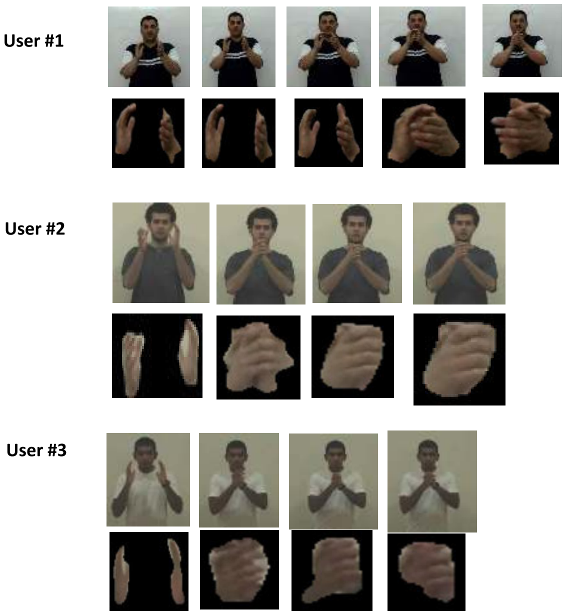

FIGURE 10. Examples of qualitative results for hand segmentation using DeepLabv3+ for each user.

C. EXPERIMENTS ON HAND SHAPE FEATURE EXTRACTION USING CSOM

A single Convolutional Self-Organizing Map (CSOM) layer is utilized to learn hand shape features. Two-dimensional cylindrical shape SOM with a rectangular lattice is trained to capture the distribution of local image patches. The parameters of CSOM are optimized empirically to improve the recognition accuracy in signer-independent testing scenario. The neighborhood radius parameter is chosen to take initial value of 0.2 and decrease linearly into a small final value of 0.01 to preserve the topological order of neurons and to keep them discriminative. Each image is divided into a set of overlapped patches with size of 5 × 5 pixels. Every local patch is approximated with the index of best matching neuron in the SOM grid. Example of the convolutional filters learned from SOM training is shown in Fig. 11. The obtained convolutional filters exhibits a topographic order in the map which helps to capture the distribution of the features in the local patches of the training hand images. The following experiments examine the effect of changing the number of neurons and block sizes of CSOM on the recognition accuracy.

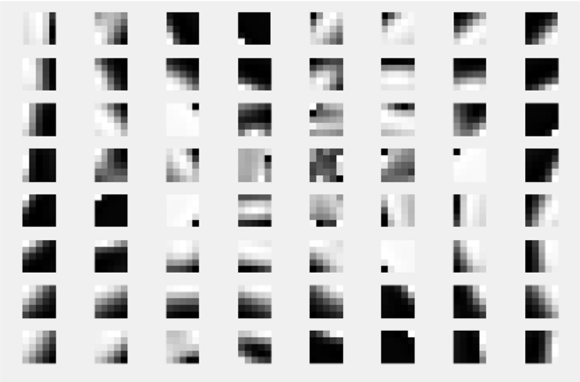

FIGURE 11. Example of topographic convolutional filters learned from convolutional SOM layer.

1) EFFECT OF CHANGING NUMBER OF NEURONS IN THE 2D SOM

The number of neurons used to approximate the feature space of the local image patches is studied in this experiment. The dimension of SOM map is fixed into two dimensions with equal number of neurons along each side. The number of neurons is changed from 2 × 2 till 20 × 20 with increase of one. Fig. 12 shows the change in accuracy when number of neurons vary. Using small number of neurons will underfit the distribution of data while large number of neurons causes overfitting. The optimum dimension of SOM map is selected to be 16 × 16 which better fit the distribution of local patches.

2) EFFECT OF CHANGING BLOCK SIZES

This experiment investigate how the average signer-independent recognition accuracy will be affected when changing the block size used for histogram computation in CSOM. The parameters of the CSOM are set as: $k = 5$ and SOM map size = 16 × 16. The block sizes are varied from 7 × 7 to 28 × 28. Fig. 13 shows that the accuracy is improved when increasing the block size till 14 × 14 block size. Although using large block size beyond 14 × 14 increases the tolerance for image variations, the accuracy decreased due

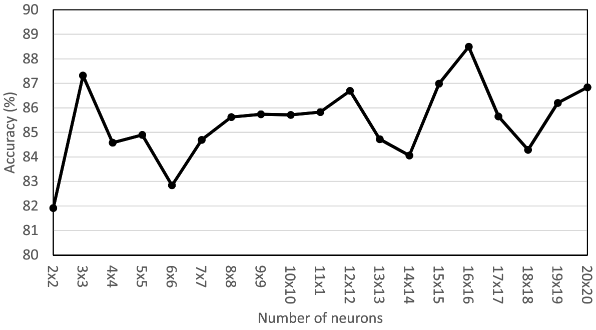

FIGURE 12. Average signer-independent recognition accuracy of the proposed framework by varying the number of neurons in CSOM.

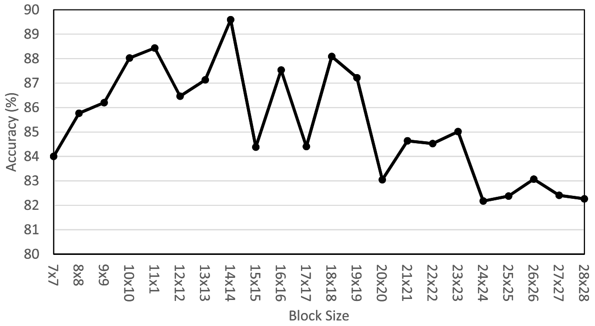

FIGURE 13. Average signer-independent recognition accuracy by varying the histogram block size in CSOM.

the loss of discrimination information among signs. Optimum block size should balance between feature discrimination and invariance.

D. EXPERIMENTS ON SEQUENCE RECOGNITION USING DEEP LSTM NETWORK

The last module in the proposed framework is sequence recognition. A new deep BiLSTM network architecture containing a stack of three BiLSTM layers followed by one fully connected layer and softmax layer is used. Stochastic gradient decent learning algorithm is utilized to optimize weights of deep BiLSTM network. Since our target is to create a signer-independent sign language recognition framework, the performance of all experiments is measured using signer independent testing strategy. Firstly, we compare the performance of the model when using LSTM instead of BiLSTM layers. We create a simple LSTMNet/BiLSTMNet comprising only one LSTM/BiLSTM layer with 512 hidden units to evaluate the effectiveness of the backward layer in BiLSTM. Signer-independent testing scenario is utilized to measure the average accuracies of each network module. The network based on LSTM and BiLSTM layers gives average accuracies of 83.5% and 86.2%, respectively. Accuracy obtained from BiLSTMNet is better than that of LSTMNet given the same number of hidden units. Since BiLSTM exploits both forward and backward sequence information to predict sign, it gives better results than LSTM which depends only on forward sequence information. In the subsequent experiment, we exploit BiLSTMNet module for gesture sequence recognition.

Secondly, the effect of changing the number of BiLSTM layers is examined. Number of BiLSTM layers is changed from one to four in order to study the feasibility of using deep architecture. Table 2 shows the average accuracy on signer-independent testing scenario with specified number of hidden units in each layer. The reported accuracies are calculated as the average from testing each of the three signers. The number of units gradually decrease from layer to layer as we go deeper. The first BiLSTM layer takes gesture sequence as input while the last output of the sequence is fed as input to the next layer and so on. Number of neurons in the fully connected layer equal to the number of gestures to be classified (23 signs). Results demonstrates that using three BiLSTM layers are sufficient to model the temporal dependencies between frames.

TABLE 2. Performance comparison of the proposed framework using different number of BiLSTM layers.

# BiLSTM layers	Layer (s) Size	Average accuracy
1	(512)	86.2%
2	(512, 256)	88.61%
3	(512,256,128)	89.5%
4	(512,256,128,64)	85.56%

E. PERFORMANCE COMPARISON OF MASK IMAGES AND GRAYSCALE IMAGES

The performance of the framework using hand masks and grayscale images is compared. Mask images resulted from the semantic segmentation network can be directly used to train CSOM and BiLSTM network. Table 3 shows the accuracies of each user by exploiting hand masks and grayscale images in the feature extraction and classification modules. The last row in the table shows the average accuracy from all users. It is clear that the accuracy is significantly improved when using grayscale images instead of mask images. This results proves that texture information of the hand is very important for gesture discrimination than relying only on shape information.

TABLE 3. Performance of the proposed framework using grayscale and mask images.

Test User	grayscale images	hand masks
1	83.65%	48.7%
2	92.09%	65.3%
3	93.04%	32.3%
Average	89.59%	48.8%

F. CROSS-SIGNER RECOGNITION ACCURACY

This experiment examines the performance of proposed framework using cross signer testing scenario. All images from one user are used for training while images from other users are employed for testing. The optimal values of the parameters obtained from previous experiments are

TABLE 4. Performance of the proposed framework using cross signer testing scenario.

User	Test user 1	Test user 2	Test user 3	Average
Train user 1	–%	73.56 %	71.04	72.30
Train user 2	80.34%	–%	85.39	82.87
Train user 3	64.78%	72.96 %	–	68.87

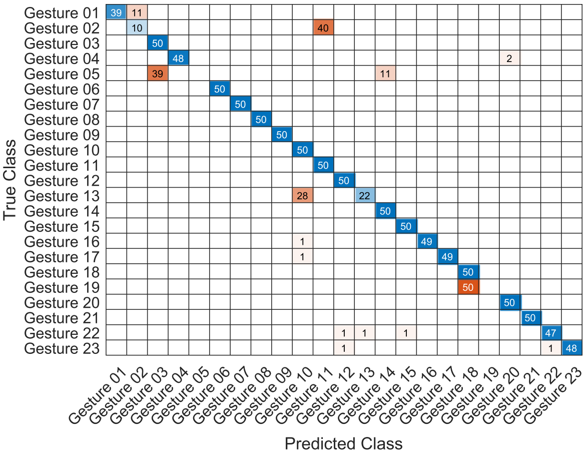

FIGURE 14. First user confusion matrix using all signs.

exploited. Table 4 shows the accuracy of testing each user using other users in training. Results reveal that using training images from second user give the highest average accuracy while using those from third user give the lowest. It can be concluded that second user performing most signs similarly to that performed by the first and third users. However, the performance of third user differs from that of other two users. Increasing the number of users used in training helps much to increase the similarities among same signs and hence improve the performance.

G. STATE-OF-THE-ART COMPARISONS

The performance of proposed framework is compared with state-of-the-art methods which reported their results using same database. Many research works have proposed to solve this problem with same database using signer-dependent scenario and achieved almost 100% accuracy. Signer-dependent sign language recognition is well studied and almost a solved problem, however signer-independent strategy has many difficulties and challenges. In this experiment, only results from the methods exploiting signer-independent testing strategy are reported. All signs from two users are selected to train the proposed model while signs from the remaining user are applied for testing. The best parameters of CSOM and BiLSTM network obtained from the previous experiments are utilized. The best BiLSTM network architecture contains three BiLSTM layers with number of BiLSTM cells equal 512, 256 and 128.

Results of comparison are shown in Table 5. The compared methods used in this experiment are based on accumulated differences and DCT [22], [64], bag of features and bag of postures [65], and 3DCNN architecture [66]. Results in the table show that hand segmentation is an essential step to improve the accuracy for all methods. The performance of proposed framework surpasses all other methods with large margin. It can be inferred from the results that using DeepLabv3+ semantic segmentation module significantly increases the performance by 70%. The average accuracy of the 23 signs computed from all users is 89.5%. Con-

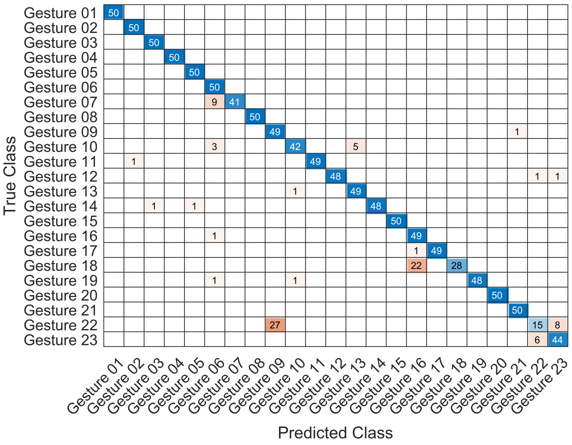

FIGURE 15. Second user confusion matrix using all signs.

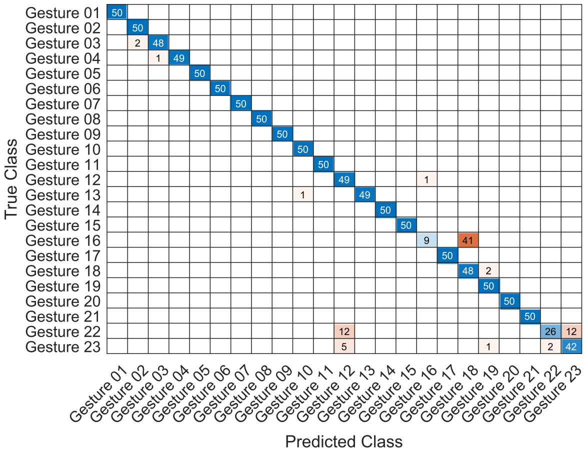

FIGURE 16. Third user confusion matrix using all signs.

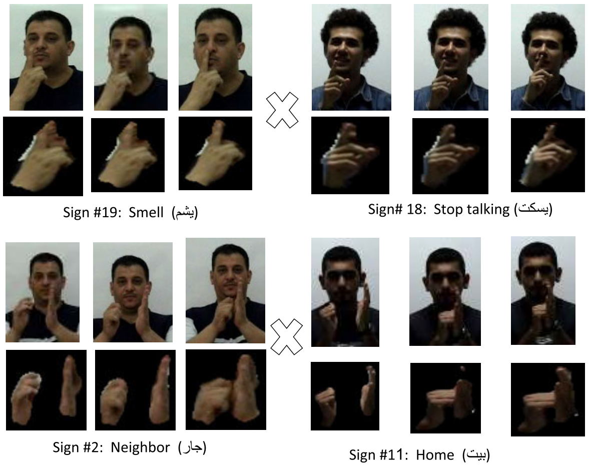

FIGURE 17. Example of confused signs among users.

TABLE 5. Comparison of proposed framework with state-of-the-art signer-independent methods.

Reference	Method	Hand segmentation	# users	Accuracy
Shanebla et al. [64]	accumulated differences + DCT +KNN	Without segmentation	3	17.67%
Shanebla and Assaleh [22]	accumulated differences + DCT +KNN	colored gloves	50	87%
Mahmoud and Sidig [65]	BoF+ BoP	Without segmentation	3	45.17%
Mahmoud and Sidig [65]	BoF+ BoP	skin segmentation	3	66.96%
Al-Hammadi et al. [66]	3D-CNN	Without segmentation	3	34.9%
Proposed framework	CSOM + BiLSTMNet	Without hand segmentation	3	20.50%
Proposed framework	(DeepLabv3+) + CSOM+BiLSTMNet	DeepLabv3+ mask images	3	48.8%
Proposed framework	(DeepLabv3+) + CSOM+BiLSTMNet	DeepLabv3+ Grayscale images	3	89.59%

fusion matrices for each of the three signer are shown in Figs. 14, 15 and 16. Example of the two confused sign pairs are depicted in Fig. 17. From confusion matrix of user 1: sign #2 (neighbor) confused with sign #11 (home), sign #5 (enemy) is confused with sign #3 (guest) and sign #18 (stop talking) confused with sign #19 (smell). It is noticed that some signs of user 1 are performed differently than that of other two users. Large differences in performing signs among users make it difficult to correctly recognize these signs in signer-independent evaluation strategy. Additionally, signs #18 and #19 have very similar hand shape and position which make them confused. Since the used database has only 3 users, the accuracy obtained from experiments are satisfactory. However, increasing the number of training users will highly improve the performance.

V. CONCLUSION

This paper proposes a new framework for signer-independent isolated Arabic sign language recognition based on the combination of DeepLabv3+ semantic segmentation, single layer convolutional SOM and Bi-directional long short-term memory network. Hand segmentation problem is efficiently solved through applying the state-of-the-art semantic segmentation DeepLabv3+ model based on Resnet-50 as a backbone encoder network and atrous spatial pyramid pooling. The obtained mask image from DeepLabv3+ are exploited to crop hand regions from each corresponding frame of the input video sequence and then normalized to fixed size for scale invariance. Hand shape features are learned and represented through a simple single layer convolutional SOM architecture. Features extracted from CSOM efficiently capture hand shape information, in addition it can handle various image condition variations thanks to the utilization of best matching neuron indexes. Deep bi-directional LSTM recurrent neural network consists of three BiLSTM, single fully-connected and softmax classification layers successfully model the dynamics of hand gestures. Experimental results show that the new framework is very efficient in solving signer-independent isolated words Arabic sign language recognition problem. The evaluation of the proposed framework on the Arabic benchmark dataset achieves an average accuracy of 89.5% using DeepLabv3+ hand semantic segmentation while it significant drooped by 69.0% without exploiting the hand segmentation module. The performance of the proposed method outperforms all state-of-the-art methods and can be extended to solve continuous sign language recognition problem for Arabic and other languages.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Majmaah University for funding this work.

REFERENCES

[1] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 131–153, Jan. 2019.

[2] P. Kumar, H. Gauba, P. Pratim Roy, and D. Prosad Dogra, “A multimodal framework for sensor based sign language recognition,” *Neurocomputing*, vol. 259, pp. 21–38, Oct. 2017.

[3] E. K. Kumar, P. V. V. Kishore, M. T. K. Kumar, and D. A. Kumar, “3D sign language recognition with joint distance and angular coded color topographical descriptor on a 2—Stream CNN,” *Neurocomputing*, vol. 372, pp. 40–54, Jan. 2020.

[4] W. Zhao, Y. Bao, and H. Qu, “Hand gesture understanding by weakly-supervised fusing shallow/deep image attributes,” *Signal Process., Image Commun.*, vol. 82, Mar. 2020, Art. no. 115760.

[5] A. U. Khan and A. Borji, “Analysis of hand segmentation in the wild,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4710–4719.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[8] T. Ozcan and A. Basturk, “Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition,” *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8955–8970, Dec. 2019.

[9] A. A. Q. Mohammed, J. Lv, and M. S. Islam, “A deep learning-based End-to-End composite system for hand detection and gesture recognition,” *Sensors*, vol. 19, no. 23, p. 5282, 2019.

[10] K. Suri and R. Gupta, “Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory,” *Comput. Electr. Eng.*, vol. 78, pp. 493–503, Sep. 2019.

[11] R. Cui, H. Liu, and C. Zhang, “A deep neural framework for continuous sign language recognition by iterative training,” *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1880–1891, Jul. 2019.

[12] J. Imran and B. Raman, “Deep motion templates and extreme learning machine for sign language recognition,” *Vis. Comput.*, vol. 36, no. 6, pp. 1233–1246, Jun. 2020.

[13] S. Aly, “Learning invariant local image descriptor using convolutional mahalanobis self-organising map,” *Neurocomputing*, vol. 142, pp. 239–247, Oct. 2014.

[14] W. Aly, S. Aly, and S. Almotairi, “User-independent American sign language alphabet recognition based on depth image and PCANet features,” *IEEE Access*, vol. 7, pp. 123138–123150, 2019.

[15] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. B. Lakulu, “A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017,” *Sensors*, vol. 18, no. 7, p. 2208, 2018.

[16] A. H. Vo, V.-H. Pham, and B. T. Nguyen, “Deep learning for vietnamese sign language recognition in video sequence,” *Int. J. Mach. Learn. Comput.*, vol. 9, no. 4, pp. 440–445, 2019.

[17] S. M. Kamal, Y. Chen, S. Li, X. Shi, and J. Zheng, “Technical approaches to Chinese sign language processing: A review,” *IEEE Access*, vol. 7, pp. 96926–96935, 2019.

[18] S. Aly, B. Osman, W. Aly, and M. Saber, "Arabic sign language fingerspelling recognition from depth and intensity images," in *Proc. 12th Int. Comput. Eng. Conf. (ICENCO) Boundless Smart Societies*, Cairo, Egypt, Dec. 2016, pp. 99–104.

[19] Q. Xiao, M. Qin, P. Guo, and Y. Zhao, "Multimodal fusion based on LSTM and a couple conditional hidden Markov model for Chinese sign language recognition," *IEEE Access*, vol. 7, pp. 112258–112268, 2019.

[20] J. Joy, K. Balakrishnan, and M. Sreeraj, "SignQuiz: A quiz based tool for learning fingerspelled signs in Indian sign language using ASLR," *IEEE Access*, vol. 7, pp. 28363–28371, 2019.

[21] M. Mohandes, M. Deriche, and J. Liu, "Image-based and sensor-based approaches to arabic sign language recognition," *IEEE Trans. Human-Machine Syst.*, vol. 44, no. 4, pp. 551–557, Aug. 2014.

[22] T. Shanableh and K. Assaleh, "User-independent recognition of arabic sign language for facilitating communication with the deaf community," *Digit. Signal Process.*, vol. 21, no. 4, pp. 535–542, Jul. 2011.

[23] A. A. Ahmed and S. Aly, "Appearance-based arabic sign language recognition using hidden Markov models," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Apr. 2014, pp. 1–6.

[24] S. Aly and S. Mohammed, "Arabic sign language recognition using spatio-temporal local binary patterns and support vector machine," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* Cham, Switzerland: Springer, 2014, pp. 36–45.

[25] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, "A signer-independent arabic sign language recognition system using face detection, geometric features, and a hidden Markov model," *Comput. Electr. Eng.*, vol. 38, no. 2, pp. 422–433, Mar. 2012.

[26] M. AL-Rousan, K. Assaleh, and A. Tala'a, "Video-based signer-independent arabic sign language recognition using hidden Markov models," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 990–999, Jun. 2009.

[27] D. Dahmani and S. Larabi, "User-independent system for sign language finger spelling recognition," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 1240–1250, Jul. 2014.

[28] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Comput. Appl.*, Jan. 2020.

[29] X. Jiang, M. Lu, and S.-H. Wang, "An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language," *Multimedia Tools Appl.*, early access, doi: 10.1007/s11042-019-08345-y.

[30] O. K. Oyedotun and A. Khashman, "Deep learning in vision-based static hand gesture recognition," *Neural Comput. Appl.*, vol. 28, no. 12, pp. 3941–3951, Dec. 2017.

[31] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4207–4215.

[32] Y. Liao, P. Xiong, W. Min, W. Min, and J. Lu, "Dynamic sign language recognition based on video sequence with BLSTM-3D residual networks," *IEEE Access*, vol. 7, pp. 38044–38054, 2019.

[33] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2257–2264.

[34] J. Huang, W. Zhou, H. Li, and W. Li, "Attention-based 3D-CNNs for large-vocabulary sign language recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2822–2832, Sep. 2019.

[35] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[36] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[37] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[39] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1949–1957.

[40] B. Kang, K.-H. Tan, N. Jiang, H.-S. Tai, D. Treffer, and T. Nguyen, "Hand segmentation for hand-object interaction from depth map," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2017, pp. 259–263.

[41] A. U. Khan and A. Borji, "Analysis of hand segmentation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4710–4719.

[42] W. Wang, K. Yu, J. Hugonot, P. Fua, and M. Salzmann, "Recurrent U-Net for resource-constrained segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2142–2151.

[43] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[44] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*. [Online]. Available: http://arxiv.org/abs/1506.00019

[45] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014, *arXiv:1410.5401*. [Online]. Available: http://arxiv.org/abs/1410.5401

[46] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[47] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[48] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *Proc. IEEE Int. Conf. Neural Netw.*, 1993, pp. 1183–1188.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[51] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[52] K. Yao, T. Cohn, K. Vylomova, K. Duh, and C. Dyer, "Depth-gated LSTM," 2015, *arXiv:1508.03790*. [Online]. Available: http://arxiv.org/abs/1508.03790

[53] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[54] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," 2016, *arXiv:1603.01354*. [Online]. Available: http://arxiv.org/abs/1603.01354

[55] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[56] O. Koller, C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, doi: 10.1109/TPAMI.2019.2911077.

[57] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified LSTM model for continuous sign language recognition using leap motion," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7056–7063, Aug. 2019.

[58] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: http://arxiv.org/abs/1412.7062

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[61] T. Kohonen, "Essentials of the self-organizing map," *Neural Netw.*, vol. 37, pp. 52–65, Jan. 2013.

[62] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.

[63] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[64] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in arabic sign language," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 641–650, Jun. 2007.

[65] S. A. Mahmoud and A. A. Sidig, "Automated sign language recognition," U.S. Patent 10 037 458, Jul. 31, 2018.

[66] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, and M. S. Hossain, "Hand gesture recognition using 3D-CNN model," *IEEE Consum. Electron. Mag.*, vol. 9, no. 1, pp. 95–101, Jan. 2020.

SALEH ALY (Associate Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Assiut University, Assiut, Egypt, in 1997 and 2004, and the Ph.D. degree from the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, in 2010. He is currently an Associate Professor at the Department of Information Technology, College of Computer and Information Sciences, Majmaah University, and on a leave from the Electrical Engineering Department, Faculty of Engineering, Aswan University, Egypt. In 2012 and 2014, he was a Visiting Researcher at the Department of Media Science, Nagoya University, and the Department of Computer Science, Tsukuba University, Japan, respectively. His research interests include deep learning, neural networks, pattern recognition, image processing, machine learning, and computer vision.

WALAA ALY received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Aswan University, Aswan, Egypt, in 2000 and 2006, respectively, and the Ph.D. degree from the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan, in 2010. She is currently an Assistant Professor at the Department of Information Technology, College of Computer and Information Sciences, Majmaah University, and on a leave from the Electrical Engineering Department, Faculty of Engineering, Aswan University, Egypt. In 2014, she was a Visiting Researcher at the Department of Computer Science, Tsukuba University, Japan. Her research interests include pattern recognition, machine learning, deep learning, image processing, computer vision, and neural networks.