IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 30, 2020, accepted October 8, 2020, date of publication October 19, 2020, date of current version November 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032140

Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation

MUNEER AL-HAMMADI[1,2], (Member, IEEE), GHULAM MUHAMMAD[1,2], (Senior Member, IEEE), WADOOD ABDUL[1,2], (Member, IEEE), MANSOUR ALSULAIMAN[1,2], MOHAMMED A. BENCHERIF[1,2], TAREQ S. ALRAYES[3], HASSAN MATHKOUR[2,4], AND MOHAMED AMINE MEKHTICHE[2]

[1]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[2]Center of Smart Robotics Research, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[3]Department of Special Education, College of Education, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Ghulam Muhammad (ghulam@ksu.edu.sa)

This research project was funded by the Targeted Research Grant Program - The National Transformation Program in King Abdulaziz City for Science and Technology - Kingdom of Saudi Arabia - Grant No. 5-18-03-001-0003.

ABSTRACT Hand gesture recognition is an attractive research field with a wide range of applications, including video games and telesurgery techniques. Another important application of hand gesture recognition is the translation of sign language, which is a complicated structured form of hand gestures. In sign language, the fingers' configuration, the hand's orientation, and the hand's relative position to the body are the primitives of structured expressions. The importance of hand gesture recognition has increased due to the prevalence of touchless applications and the rapid growth of the hearing-impaired population. However, developing an efficient recognition system needs to overcome the challenges of hand segmentation, local hand shape representation, global body configuration representation, and gesture sequence modeling. In this paper, a novel system is proposed for dynamic hand gesture recognition using multiple deep learning architectures for hand segmentation, local and global feature representations, and sequence feature globalization and recognition. The proposed system is evaluated on a very challenging dataset, which consists of 40 dynamic hand gestures performed by 40 subjects in an uncontrolled environment. The results show that the proposed system outperforms state-of-the-art approaches, demonstrating its effectiveness.

INDEX TERMS 3DCNN, hand gesture recognition, hand segmentation, deep learning, computer vision, sign language recognition.

I. INTRODUCTION

Hand gesture recognition is the first step for a computer to understand human body language. It plays a pivotal role in a wide range of human-computer interaction (HCI) applications such as smart TV control, video games, telesurgery, and virtual reality [1]. Sign language translation is one of the most important applications of hand gesture recognition. The hand gestures involved in sign language are structured in a very complex way as they convey important human communication information and feelings. The primitives of these manual expressions are the global configuration (the hand's orientation and its relative position to the body) and the local fingers' configuration. An efficient recognition system should consider all these complementary primitives in a sequence of frames. However, the time dependence of these frames makes it difficult to directly compare the primitives in Euclidean space. Most of the existing recognition systems only consider the local configuration of the hand. These systems either receive a segmented hand region as input or perform a hand segmentation preprocessing step using skin color models or colored gloves [2]–[10]. However, such systems perform well only for gestures involving simple alphabets and numbers, which slightly rely on the global configuration, but not for real sign language gestures.

Other existing systems ignore the local configuration of the fingers and consider only the global body configuration. These systems have been successful for some HCI applications with a small number of simple and well-defined gestures but have failed for real sign language gesture recognition [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

Traditionally, dynamic hand gesture recognition systems use different techniques to extract handcrafted features followed by a sequence modeling technique such as a hidden Markov model (HMM). However, the recent success of deep learning techniques in image classification, object recognition, speech recognition, and human activity recognition [12]–[14] has encouraged many researchers to exploit them for hand gesture recognition. For example, convolutional neural networks (CNN) have been widely used for learning visual features in computer vision.

On the other hand, a 3D convolutional neural network (3DCNN) has been used for video modeling, which is an extended version of standard CNNs that uses spatiotemporal filters. This architecture has been explored previously in several video analysis fields for spatiotemporal feature representation; e.g., [15]–[18]. The most important characteristic of 3DCNN is its ability to directly create hierarchical representations of spatiotemporal data. However, it requires more parameters than 2DCNN, which is one of its disadvantages. Moreover, 3DCNN has an additional kernel dimension, which makes it harder to train. Hence, instead of training a 3DCNN from scratch, using domain adaptation on pretrained instances is preferred.

In a previous hand gesture recognition work [19], we implemented a variation of the C3D architecture [17] and used knowledge transfer from human action recognition to hand gesture recognition. The C3D architecture comprises eight convolutional layers, five pooling layers, and two fully connected (FC) layers. However, even though we obtained encouraging results in that work, we noticed that the direct application of 3DCNN for hand gesture modeling has two main drawbacks. Firstly, 3DCNN modeling is not robust enough to capture the long-term temporal dependence of the hand gesture signal. Secondly, modeling the hand gesture signal in a video should be slightly different than other video-based analysis for human activity recognition or event recognition in general. For the latter case, the whole scene and maybe multiple interacting objects in the frame are involved discriminative descriptors for the overall recognition. In contrast, the discriminative features in hand gesture recognition are located mainly in the fingers' configuration, the hand's orientation, and the hand's relative position to the body. In other words, most of the frame area contains non-relevant features that increase the misclassification ratio. In another work, we addressed the first mentioned drawback of modeling the long-term temporal dependence [20] by using independent instances of 3DCNN to model the local spatiotemporal features of different temporal segments. We also explored different techniques to globalize the local representations. Our experimental results showed that using temporal modeling enhancement can improve the performance of the 3DCNN model. In this study, we address the second drawback by using both the local and global configurations of the hand gesture while giving more attention to the fingers' configuration and eliminating the most non-relevant features.

The contributions of the paper are as follows:

(1) Optimizing the level of C3D architecture knowledge transfer between human activity recognition and hand gesture recognition.
(2) Presenting a hand gesture recognition system based on an optimized C3D architecture. The proposed system uses local and global configurations efficiently with more attention to the hand region.
(3) Presenting a novel method for hand segmentation based on the openpose framework.
(4) Optimizing two architectures for local features aggregation.

The rest of this paper is organized as follows. Section II reviews related works on hand gesture recognition. Section III describes our dataset. Section IV presents the proposed system. Section V discusses the experimental results. Finally, Section VI concludes the paper.

II. RELATED WORK

During the last three decades, several works have been conducted to tackle hand gesture recognition. Most works have followed two approaches: a vision-based approach and a non-vision-based approach. In the non-vision-based approach, hand gesture data are collected via interfacing devices such as data gloves, motion sensors, and position trackers [21]–[25]. However, the hardware setup of this approach is costly and is inconvenient because it restricts the signer's movement. On the other hand, the vision-based approach overcomes these downsides by collecting the data via cameras and imaging sensors. However, research works using this approach have encountered many challenges that degrade the performance of existing systems such as lighting inconsistency, motion blur, background clutter, and hands occlusion. Moreover, studies using the vision-based approach can be classified into two categories: conventional techniques (e.g., [2]–[9] and [26]–[33]) and deep learning-based techniques (e.g., [10], [11], and [34]–[41]).

The paper by Murakami *et al.* is one of the earliest papers in the field [26]. In that paper, they used an artificial neural network (ANN) to recognize 42 alphabets of the Japanese sign language. The ANN was also used with data gloves to recognize isolated words of the American sign language (ASL) in two stages, i.e., phonemic and word recognition, but it was evaluated on a relatively limited lexicon [2]. Another robust method based on ANN classifier and skin color segmentation was recently presented for recognizing Thai alphabets [3]. The histogram of oriented gradient (HOG) was used in this approach to represent the segmented hand shape. In another work, skin color was used for hand region segmentation [4]. The segmented hand motion trajectory was then modeled by a time-delay neural network to recognize 40 ASL words.

HMMs, on the other hand, were extensively used for hand gesture recognition. For example, Starner *et al.* proposed HMMs to recognize sentence-level continuous ASL [5], where the skin color was used for hand segmentation. They used a lexicon of 40 words to construct the test sentences.

Other HMM-based methods used different combinations of principal component analysis, kurtosis position, and motion chain code descriptors [27]. The best accuracy was achieved on the RWTH-BOSTON-50 database by combining the three descriptors. Killy *et al.* used a single HMM for each hand with colored gloves for hand segmentation and tracking [6] and they evaluated their method using a small dataset of eight gestures. Pu *et al.* also used HMMs to model the segmented trajectory of hand gesture for 100 Chinese sign words [28]. The trajectory segments were represented as histograms of shape context. In another work proposed by Li *et al.*, an entropy-based K-means was used to evaluate the number of states in each HMM model [29]. A combination of the Baum-Welch algorithm and the artificial bee colony algorithm was used to determine and learn the structure of HMM. Recently, Yang *et al.* classified the hand gesture trajectory of ASL in a hierarchical way to generate a sequence of observations [30]. HMMs were then applied to model and classify the sequences.

An SVM classifier was also used for recognizing the Irish sign language [7] and ASL [31]. A skin color model was used in [7] for hand segmentation and a combination of weight eigenspace size function and Hu moments was used to represent the hand shape. On the other hand, the fingertips' coordinates collected by Leap Motion and Intel RealSense 3D cameras were used in [31]. In another work, Aly *et al.* used SVM to recognize 23 Arabic sign language words [32]. They proposed a local binary pattern in three orthogonal planes to represent the appearance and motion features of signs. The proposed method in [8] used particle filtering for hand tracking. Feature covariance matrix and the minimum Riemann distance metric were then used on the detected hand for representation and classification. Lim *et al.* used sparse observations from a video of RGB-D frames [9], where the skin color and depth maps were used for hand segmentation and the HOG was used for posture representation. The similarity between the postures of different samples was then measured. Abid *et al.* used bag-of-visual words with a local part model approach to recognize six simple dynamic gestures [33].

Recently, deep neural network architectures, such as CNN and long short-term memory (LSTM) network, have been used for hand gesture recognition. For example, Huang *et al.* used CNN and ANN for the representation and classification of 20 Italian gestures [34]. To perform well, this method requires a multimodality input, which includes the RGB frames, the depth maps, and the skeleton joints. Similarly, Lionel *et al.* investigated temporal convolutions with bidirectional recurrence for gesture recognition in the Montalbano dataset [35]. Another deep learning architecture was proposed for ASL hand posture recognition [36], where the depth data were used for segmenting the hand region and the deep belief neural network and CNN were used for feature learning and classification. Another recent approach proposed by Okan *et al.* involved the fusion of optical flow and RGB frames to adapt the pretrained inception model for hand gesture recognition [37]. Another CNN-based architecture was proposed in [10] for static hand gesture recognition. The input to this architecture was a small image with a size of 32×32 that contains only the hand region. A CNN and an LSTM were combined for temporal 3D pose gesture recognition [38], where the input frames contain the 3D joints of the human body. Furthermore, in [39], two streams of 3DCNN were presented for gesture recognition. The inputs for the two streams were interleaved volumes of depth maps and preprocessed Sobel gradient with different resolutions. The ResNet architecture was used by Chen *et al.* for encoding the features of frames' sequence in a single 2D matrix [40]. Then, another CNN was used to capture the evolution of the spatiotemporal features for classification. Recently, Hu *et al.* used the skeletal data of hand gestures to design a deep learning-based control system for unmanned aerial vehicles [11]. Both CNN and different multilayer perceptron (MLP) architectures were investigated for feature learning. Another recent work for Arabic sign language recognition used semantic segmentation for detecting the hand [41]. Unsupervised learning via convolutional self-organizing map was then applied for feature extraction and a bidirectional LSTM was used for sequence modeling.

FIGURE 1. Sample frames from the KSU-SSL dataset.

The proposed system in this study is based on a single modality input (RGB video) and does not require other modalities such as the depth maps or skeleton joints. It also combines both the local and global configurations of hand gestures.

III. KSU-SSL DATASET

Our experiments were conducted on the King Saud University Saudi Sign Language (KSU-SSL) dataset reported in [20]. The dataset contains isolated words and phrases from common expressions in the SSL dictionary. The dataset was recorded by 40 participants over five recording sessions. Some of the participants are deaf and some are well trained by sign language experts. The recorded gestures are listed in Appendix I. Sample frames from the dataset are illustrated in FIGURE 1. There was no restriction on the recording

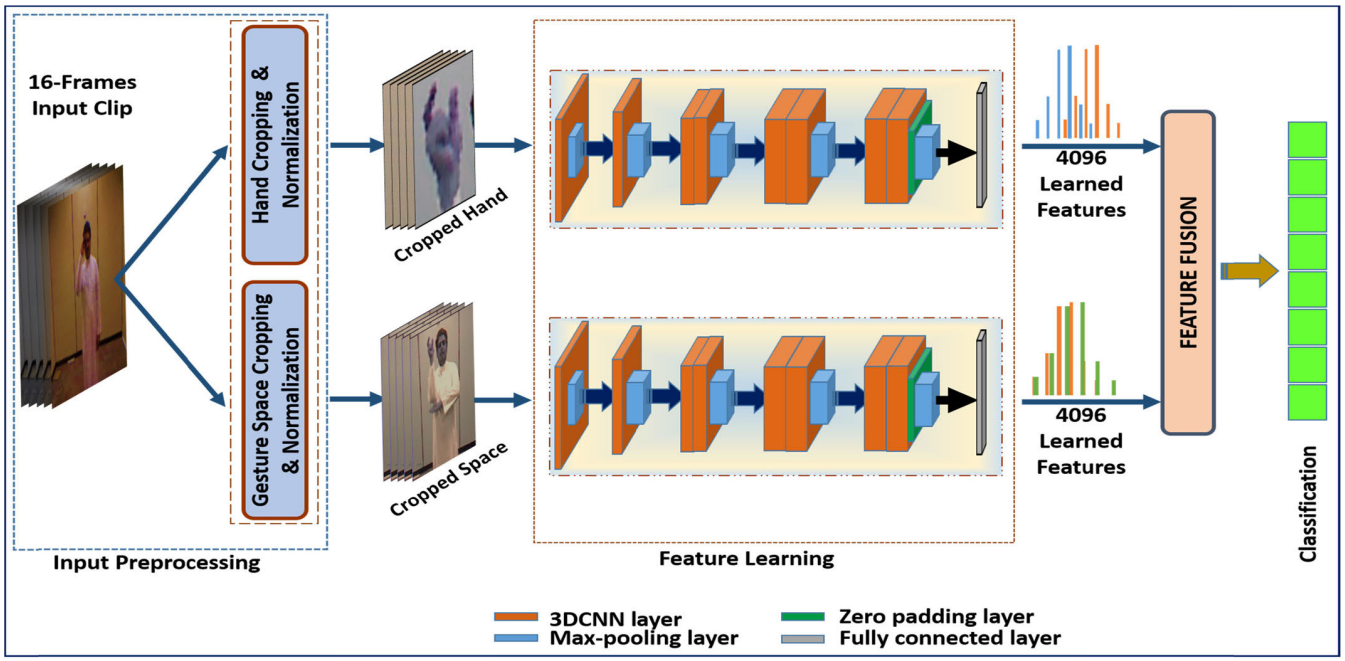

FIGURE 2. Proposed system for hand gesture recognition using local and global configuration features.

background, participants clothes or lighting conditions. The KSU-SSL dataset exhibits high variations in illumination and participants' clothes, position, scale, and gesturing speed.

IV. PROPOSED SYSTEM

Consider a set of M training video samples $\{x_i, \mathbf{y_i}\}_{i=1}^{M}$ of variable duration t_i such that x_i is the ith sample in the set and $\mathbf{y_i}$ is the corresponding label vector. This label vector is of length K, where K is the number of targeted gesture classes.

One-hot encoding in a multiclass setup sets each vector element to one if the corresponding class is present, otherwise, it is set to zero. FIGURE 2 illustrates the proposed system.

It consists of three main phases: input preprocessing, feature learning and feature fusion, and classification. In the next subsections, we discuss the details of the different phases.

A. INPUT PREPROCESSING

The input videos are converted into sequences of RGB frames of different lengths. Then, linear sampling is used for temporal dimension normalization, where only 16 frames are linearly selected from each video sequence.

This temporal normalization step for the input can be achieved by different techniques such as the bag-of-visual words. These techniques are very efficient when the sequence order is of low importance for discrimination such as in video event and human action recognition. For hand gesture recognition, the sequence order should be preserved because it encodes highly discriminative features; hence, linear sampling is the preferred technique to be used. Two cropping and normalization methods are then performed simultaneously on the selected frames. The first method locates the signer's face using the Viola and Jones algorithm [42]. Then, the gesture space is estimated and cropped in each frame based on the detected facial length and body parts ratios information [43]. Each frame is resized to a fixed size of 112×112 pixels while preserving the aspect ratio.

This method outputs a sequence $X_B \in \mathcal{R}^{112\times112\times3\times16}$ of 16 frames, where each frame includes the entire gesture space.

In addition to avoid the effects of the variations of the signer's height and distance from the camera, this spatial normalization and cropping reduce the effects of nonrelevant features in each frame. The second method, on the other hand, crops and normalizes the hand region to focus more on the fingers' configuration.

HAND CROPPING AND NORMALIZATION

This method uses an open-source real-time human pose estimation framework called openpose, which is a deep learning-based framework for detecting the 2D key points of each individual in an image. This framework improves the machine understanding of human activity in an image or video sequence [44]. It takes as an input an RGB image and returns as an output a list of (x, y) coordinates for all human body key points. FIGURE 3 illustrates the upper body openpose key points. From the whole list of returned key points, only the wrist and elbow joints are used for cropping the hand region.

For instance, the vector from the elbow joint (x_e, y_e) to the wrist joint (x_w, y_w) indicates the arm axis. Based on the arm axis direction, we propose an efficient method to estimate a small square region around the hand to be cropped. The length of this square region is equal to the absolute value of the distance between the wrist and the elbow joints as in **Error! Reference source not found.**):

$$length = \sqrt{(x_w - x_e)^2 + (y_w - y_e)^2} \quad (1)$$

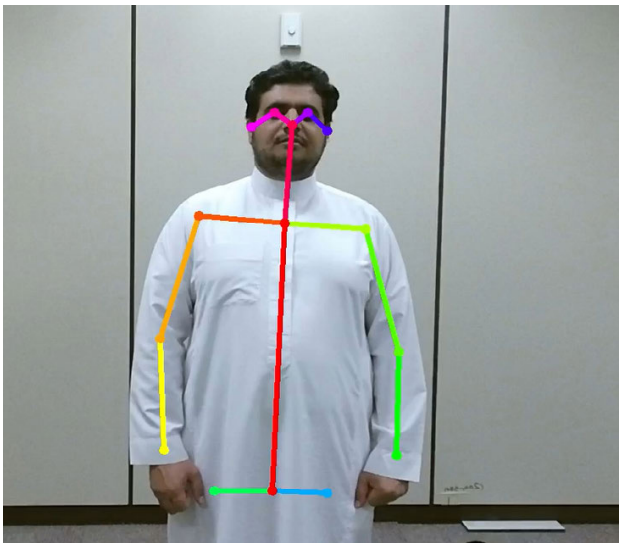

FIGURE 3. Upper body openpose key points.

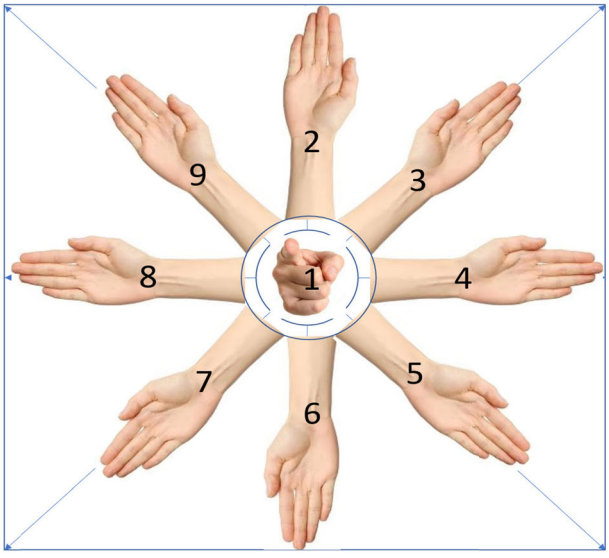

FIGURE 4. Estimated hand directions.

The proposed method estimates the hand orientation to one of the nine basic directions illustrated in FIGURE 4. These directions are:

1- The hand axis is perpendicular to the frame's plane pointing at the camera.
2- The hand axis is vertical pointing up.
3- The hand axis is diagonal pointing to the top right.
4- The hand axis is horizontal pointing to the right.
5- The hand axis is diagonal pointing to the bottom right.
6- The hand axis is vertical pointing down.
7- The hand axis is diagonal pointing to the bottom left.
8- The hand axis is horizontal pointing to the left.
9- The hand axis is diagonal pointing to the top left.

The calculation steps for estimating the top left (x_B, y_B) point and the bottom right (x_E, y_E) point of the square region to be cropped are summarized in Algorithm 1 in the Appendix II. The cropped hand region is then resized

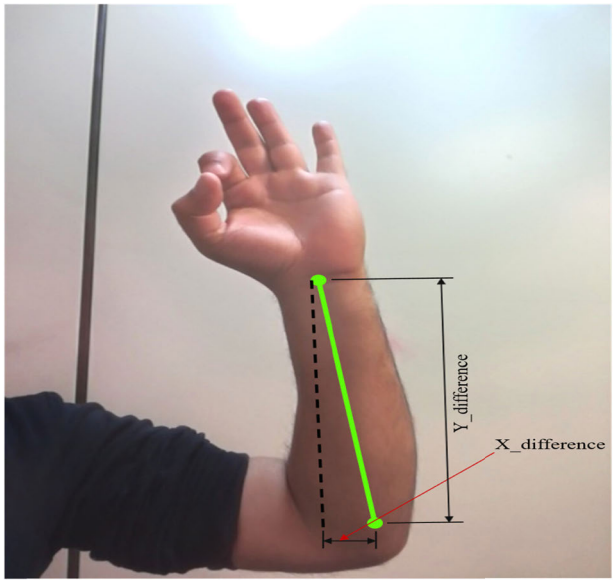

FIGURE 5. Hand direction estimation.

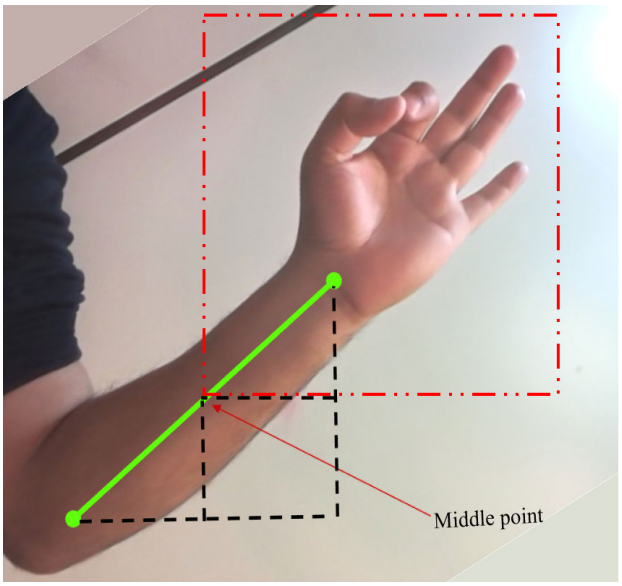

FIGURE 6. Estimating the middle point and cropped region.

to 112 × 112 pixels. The horizontal and vertical distances between the wrist and the elbow joints (X difference and Y difference) are illustrated in FIGURE 5. Based on these two values, the hand direction, and as a result, the square region to be cropped can be estimated as follows:

i. If both the horizontal and vertical distances are negligible (i.e., less than α), the two joints are nearly identical. In other words, the hand axis is perpendicular to the frame's plane (case 1 in FIGURE 4). Hence, the cropped region is centered on the wrist joint. We have found that an appropriate value for α is 40 pixels.
ii. If only the horizontal distance is negligible (i.e., less than α), the hand axis is nearly vertical. The vertical coordinates of the wrist and elbow joints are used to

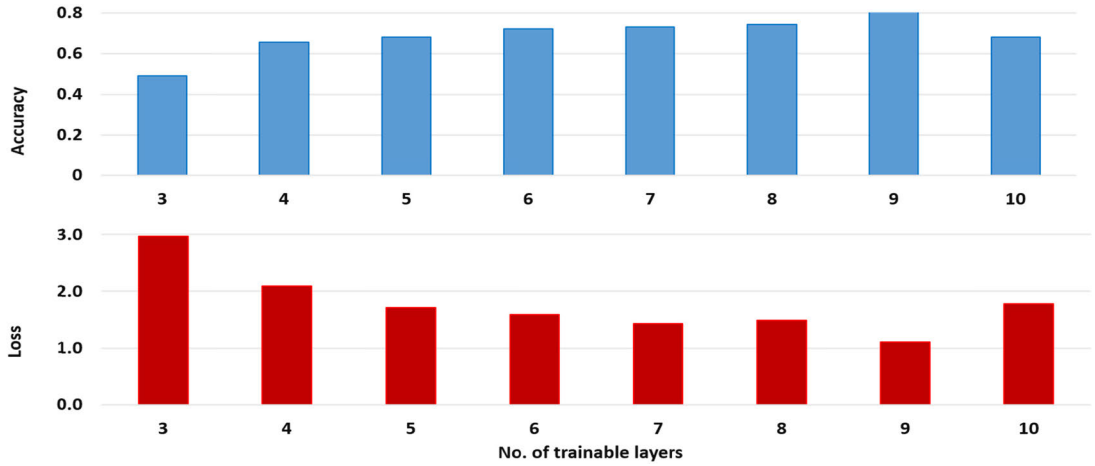

FIGURE 7. Effect of the number of trainable layers on the efficiency of knowledge transfer.

	256	256-256	256-256-256	512	512-256	512-512	512-512-256	512-512-512	1024	1024-256	1024-512	1024-1024	1024-1024-256	1024-1024-512	1024-1024-1024	2048	2048-256	2048-512	2048-1024	2048-2048	2048-2048-256	2048-2048-512	2048-2048-1024	2048-2048-2048	4096	4096-256	4096-512	4096-1024	4096-2048	4096-4096	4096-4096-256	4096-4096-512	4096-4096-1024	4096-4096-2048	4096-4096-4096	8192	8192-256	8192-512	8192-1024	8192-2048	8192-4096	8192-8192	8192-8192-256	8192-8192-512	8192-8192-1024	8192-8192-2048	8192-8192-4096	8192-8192-8192
5.E-03	3	3	3	3	3	3	3	5	3	3	3	3	3	7	3	3	3	3	3	3	3	3	3	9	3	3	3	3	3	3	3	3	3	3	17	3	3	3	3	3	2	2	3	3	3	3	9	26
1.E-03	35	13	8	82	43	66	58	43	82	83	81	83	81	79	82	82	83	82	82	83	83	82	83	84	84	82	83	83	83	81	83	82	82	82	83	83	82	83	83	82	82	84	83	84	83	83	83	81	81
5.E-04	83	62	40	83	82	82	75	71	83	84	84	83	82	82	84	84	84	83	83	83	83	82	84	83	84	83	83	84	84	84	82	83	82	83	83	83	83	83	83	83	83	83	83	83	84	83	83	83
1.E-04	85	86	78	85	86	85	84	85	85	85	85	85	84	84	85	85	85	85	85	85	84	85	85	85	84	84	86	85	85	85	84	85	84	86	84	85	85	86	85	85	85	85	84	85	85	85	85	85
5.E-05	85	85	79	85	85	86	85	84	86	87	86	86	85	86	86	86	88	87	86	86	85	86	86	86	86	86	87	86	86	86	85	85	86	86	86	86	85	87	86	87	86	86	86	86	86	86	86	86
1.E-05	85	83	63	85	83	85	82	82	85	84	85	84	83	83	84	85	84	85	85	85	85	84	84	85	85	85	85	85	85	85	86	84	85	85	84	86	86	85	85	85	86	86	86	84	85	84	85	85	86
5.E-06	84	79	42	83	80	82	73	76	83	82	83	83	80	81	83	83	83	83	83	83	82	82	82	82	83	83	83	84	85	85	82	82	83	83	83	83	83	84	84	84	84	84	83	84	83	85	84	84
1.E-06	75	47	19	80	68	67	40	42	81	72	72	74	63	62	62	81	77	79	81	80	68	71	75	75	82	79	79	81	82	82	67	76	76	79	79	84	78	80	83	83	82	82	71	80	80	82	83	83
5.E-07	65	33	16	73	46	50	28	24	77	60	65	68	45	49	49	79	66	72	76	76	50	57	61	64	82	63	74	76	78	80	44	67	71	74	73	82	57	73	78	81	80	80	52	69	74	76	80	79
1.E-07	25	11	4	42	17	17	8	5	48	26	29	27	11	14	17	59	24	34	43	36	16	21	21	22	67	17	37	48	51	51	15	19	24	32	31	70	16	44	55	49	55	58	19	24	30	37	41	43

FIGURE 8. Heat map shows the system accuracy in the searched space of the MLP hyperparameters, where the darkest color is the highest accuracy and the lightest color is the lowest accuracy.

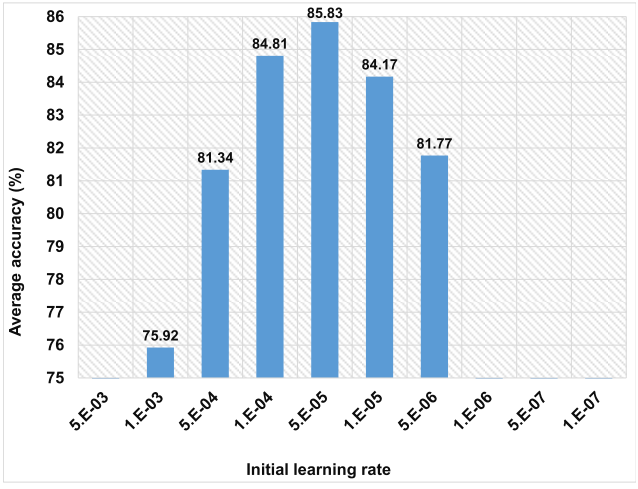

FIGURE 9. Average accuracy of all architectures with different initial learning rates.

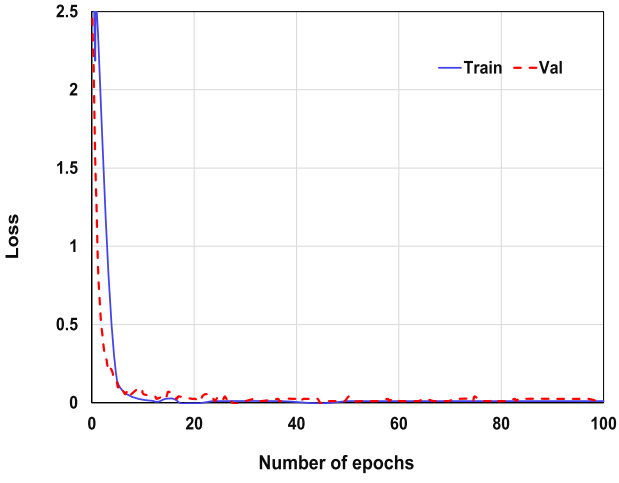

FIGURE 10. Behavior of the optimal MLP architecture in the signer independent mode.

indicate the direction of the axis. If the wrist joint is vertically less than the elbow joint, then the direction is up (case 2 in FIGURE 4). Hence, the lower border of the cropped region will pass through the wrist joint. To avoid inaccuracies, the cropped region is shifted down by a small value of ε. On the other hand, if the wrist joint is vertically greater than the elbow joint, then the direction is down (case 6 in FIGURE 4). Hence, the upper border of the cropped region will pass through the wrist joint. To avoid inaccuracies, the cropped region is shifted up by a small value of ε.

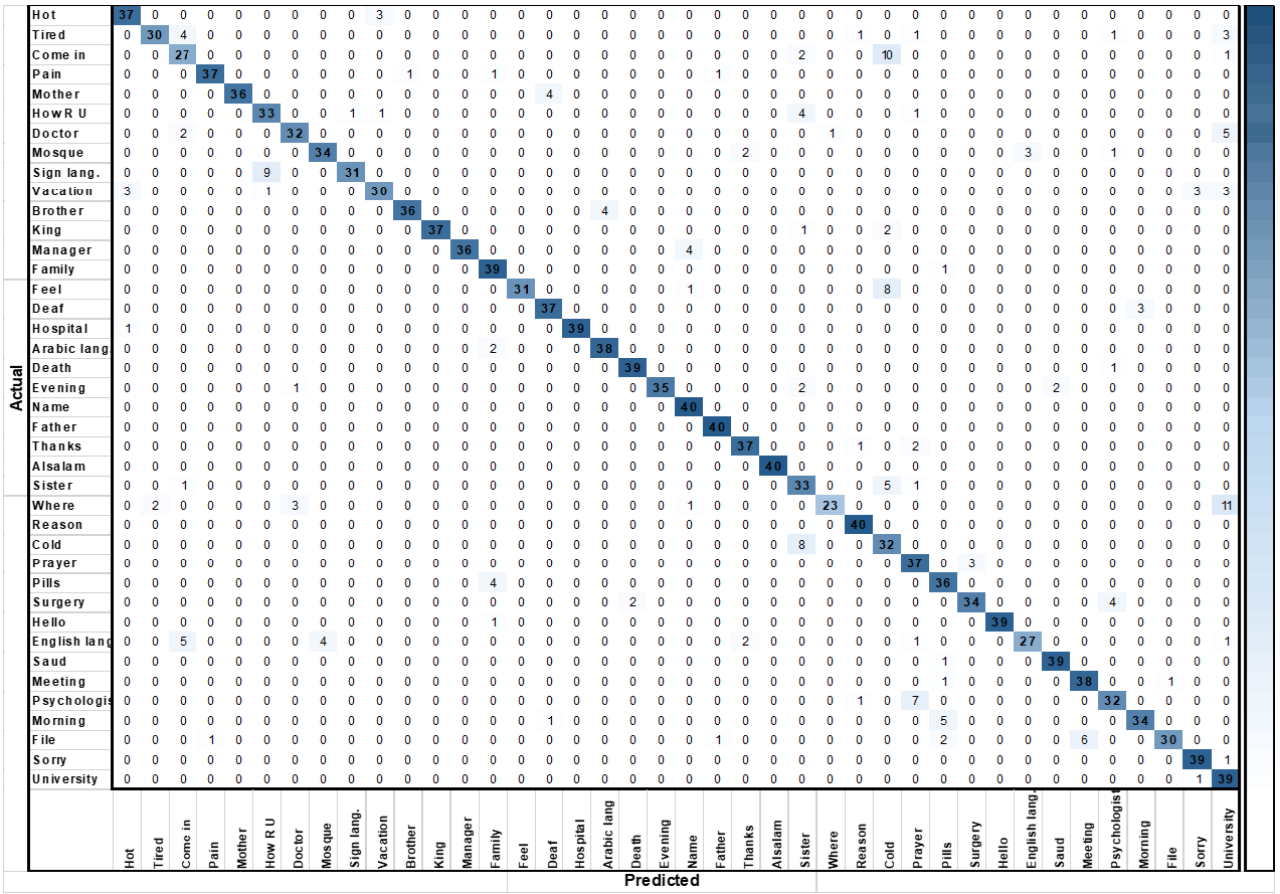

FIGURE 11. Confusion matrix of MLP two streams fusion in the signer-independent mode.

iii. If only the vertical distance is negligible (i.e., less than α), the hand axis is nearly horizontal. The horizontal coordinates of the wrist and elbow joints are used to indicate the direction of the axis.
If the wrist joint is horizontally less than the elbow joint, then the direction is to the left (case 8 in FIGURE 4). Hence, the right border of the cropped region will pass through the wrist joint. To avoid inaccuracies, the cropped region is shifted to the right by a small value of ε. On the other hand, if the wrist joint is horizontally greater than the elbow joint, then the direction is to the right (case 4 in FIGURE 4). Hence, the left border of the cropped region will pass through the wrist joint. To avoid inaccuracies, the cropped region is shifted to the left by a small value of ε.

iv. If both the horizontal and vertical distances are not negligible (i.e., each of them is greater than α), the hand axis is nearly diagonal. Hence, there are four possible directions for the hand axis, as shown in cases 3, 5, 7, and 9 in FIGURE 4. In all these cases, a middle point on the hand axis is estimated, as illustrated in FIGURE 6.

- If the horizontal and vertical coordinates of the wrist joint are greater than those of the elbow joint, the hand axis direction is down right. This is case 5 in FIGURE 4. Hence, the estimated middle point is considered as the top left corner for the cropped region.
- If the horizontal and vertical coordinates of the wrist joint are less than those of the elbow joint, the hand axis direction is top left. This is case 9 in FIGURE 4. Hence, the estimated middle point is considered as the bottom right corner for the cropped region.
- If the vertical coordinate of the wrist joint is greater than that of the elbow joint and the horizontal coordinate of the wrist joint is less than that of the elbow joint, the hand axis direction is down left. This is case 7 in FIGURE 4. Hence, the estimated middle point is considered as the top right corner for the cropped region.
- Finally, If the vertical coordinate of the wrist joint is less than that of the elbow joint and the horizontal coordinate of the wrist joint is greater than that of the elbow joint, the hand axis direction is top right. This is case 3 in FIGURE 4. Hence, the estimated middle point is considered as the bottom left corner for the cropped region, as depicted in FIGURE 6.

The preprocessing phase outputs two volumes per sample, each with a size of $112 \times 112 \times 3 \times 16$. These two volumes are delivered to the feature learning phase where one of them represents the entire gesture space and the other is dedicated to the hand region.

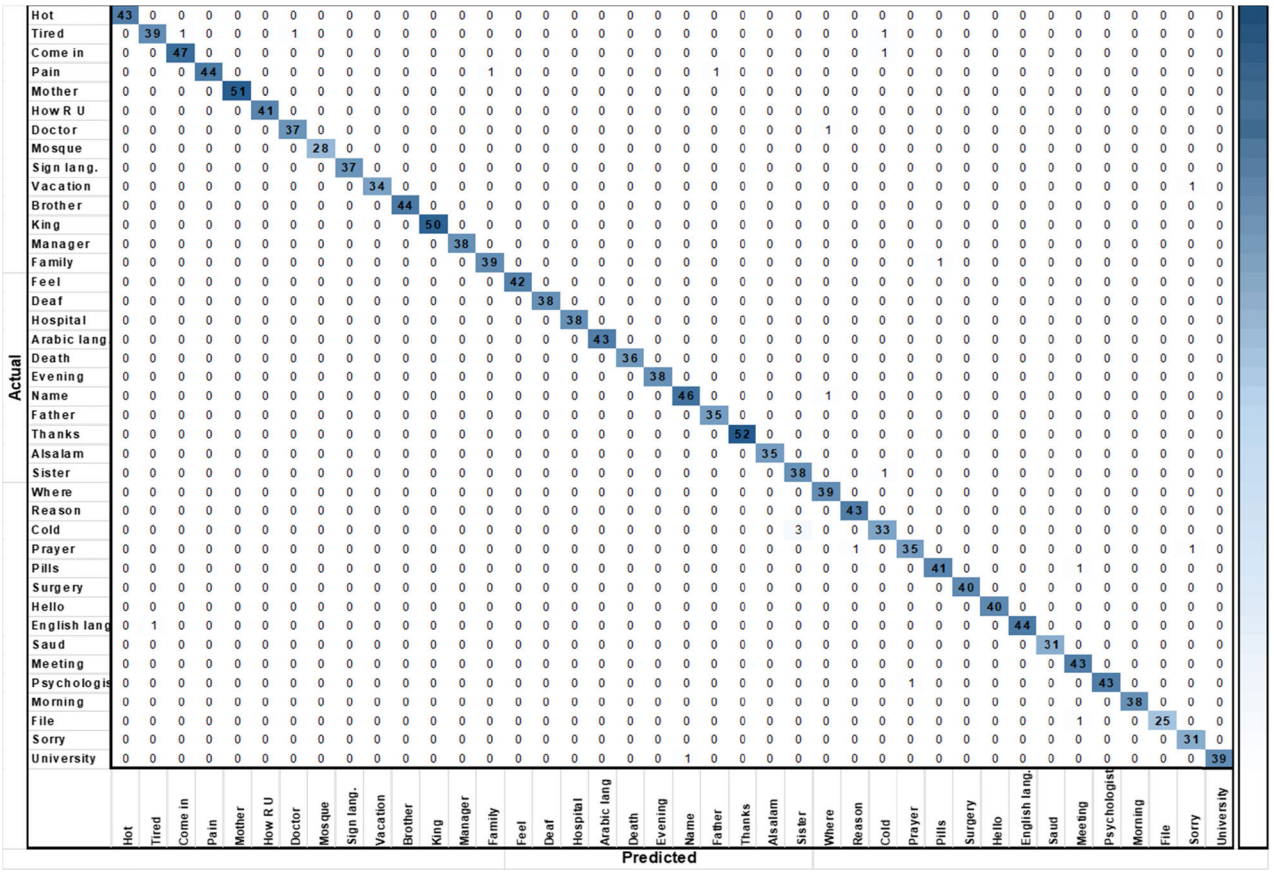

FIGURE 12. Confusion matrix of MLP two streams fusion in the signer-dependent mode.

	512-256-512	1024-256-1024	2048-256-2048	4096-256-4096	8192-256-8192	1024-512-1024	2048-512-2048	4096-512-4096	8192-512-8192	2048-1024-2048	4096-1024-4096	8192-1024-8192	4096-2048-4096	8192-2048-8192	8192-4096-8192	1024-512-256-512-1024	2048-512-256-512-2048	4096-512-256-512-4096	8192-512-256-512-8192	2048-1024-256-1024-2048	4096-1024-256-1024-4096	8192-1024-256-1024-8192	4096-2048-256-2048-4096	8192-2048-256-2048-8192	8192-4096-256-4096-8192	2048-1024-512-1024-2048	4096-1024-512-1024-4096	8192-1024-512-1024-8192	4096-2048-512-2048-4096	8192-2048-512-2048-8192	8192-4096-512-4096-8192	4096-2048-1024-2048-4096	8192-2048-1024-2048-8192	8192-4096-1024-4096-8192	8192-4096-2048-4096-8192
5.E-03	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	6	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
1.E-03	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	10	3.5	3.5	5	15	4	6	5	4	3.5	3.5	3.5	3.5	3.5	3.5	30	3.5	3.5	24	3.5
5.E-04	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	15	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	5	3.5	3.5	13	3.5
1.E-04	3.5	3.5	3.5	3.4	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.4	3.4	3.5	3.4	3.5	3.5	3.5	19	3.5	3.5	3.5	3.5	3.5	3.6	3.4	3.5	3.5	3.5	3.5	3.4	3.4	3.5	3.4	3.5
5.E-05	3.5	3.5	48	53	59	69	79	79	49	75	83	83	81	82	81	3.5	3.5	3.5	3.6	3.5	3.5	3.4	3.5	3.5	3.4	3.5	3.5	3.5	3.5	3.5	3.2	3.4	3.5	3.7	3.4
1.E-05	77	80	82	81	82	82	83	82	82	84	83	83	83	85	84	79	80	79	79	80	79	81	80	82	82	81	81	83	83	82	82	82	83	82	84
5.E-06	76	77	78	80	82	80	81	82	84	82	83	83	84	83	84	67	76	79	77	72	78	78	78	75	81	80	80	81	81	81	81	81	82	82	84
1.E-06	50	56	69	72	76	67	71	74	76	78	78	80	80	82	82	35	27	45	50	43	48	56	56	57	67	56	63	66	65	69	71	66	76	73	76
5.E-07	32	44	63	61	68	59	63	64	73	69	74	76	76	80	79	18	33	27	39	35	38	52	41	47	49	38	39	58	47	59	61	55	65	64	69
1.E-07	14	16	22	17	24	18	22	27	36	27	34	44	38	47	52	6.5	9.9	15	20	9.9	16	16	11	20	17	14	17	18	17	25	20	21	24	32	40

FIGURE 13. Heat map of the system accuracy in the searched space of the autoencoder hyperparameters, where the darkest color is the highest accuracy and the lightest color is the lowest accuracy.

B. FEATURE LEARNING

We start with the pretrained C3D architecture with eight convolutional layers, five pooling layers, and two FC layers [17]. This model is already trained on the large-scale Sport-1M human action recognition dataset [13]. In domain adaptation learning, the transferred knowledge has less impact as we move toward the layers at the top of the model, especially when the source and target domains are far away from each other.

Hence, we replace the last block, which has two FC layers with each having 4096 neurons, with a new FC layer of 4096 neurons to reduce the training cost of the two FC layers with an expansive number of parameters. Then, we optimize the level of knowledge transfer from the source domain to the target domain. This optimization step is detailed in the experimental results and discussion section. Two instances of the optimized C3D architecture are used to represent the spatiotemporal features in different levels of the frames'

sequence (i.e., the hand region and the entire gesture space region).

The first C3D instance learns the fine spatiotemporal features of the hand configuration. The hand is dominant in each input frame of this instance. On the other hand, the second C3D instance learns the coarse spatiotemporal features of the whole-body configuration. This phase produces as an output two feature vectors with each having a dimension of 4096.

C. FEATURE FUSION AND CLASSIFICATION

Two different techniques, i.e., MLP and autoencoder, are investigated to fuse the two feature vectors before feeding them to the classifier. In contrast to the system proposed in [20], we avoid the use of LSTM with this system because the two streams are not temporal segments of the gesture. Then, we perform end-to-end training for the fusion architecture with the classifier. The classification layer is activated by a SoftMax function.

V. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the proposed system, we conducted extensive experiments in two scenarios as follows:

- Signer-dependent mode: In this scenario, the samples were randomly shuffled and split into two subsets for training and evaluation. In other words, we divided the samples of each signer into training and evaluation with a random ratio.
- Signer-independent mode: In this scenario, the signers were divided into two sets. All the samples performed by the first set of signers were used for training, while all the samples performed by the other set of signers were used for testing.

A. FEATURE LEARNING

1) C3D KNOWLEDGE TRANSFER OPTIMIZATION

Typically, when using transfer learning, some of the architecture layers are iteratively fine-tuned on the target domain data to adapt their parameters for the target domain. On the other hand, the other layers are frozen to keep the original values of their parameters. In this experiment, we investigated how the performance of the C3D architecture is affected by changing the number of trainable layers to find the optimal case. This optimization step was performed in the signer-independent mode. All the samples that were recorded by the first 32 signers (80% of the samples), were used for training the architecture. The remaining 1600 samples, that were recorded by the other eight signers (20% of the samples), were used for evaluation.

We linearly sampled 16 frames from each sequence with each frame containing the entire gesture space. Then, end-to-end training was conducted for the C3D architecture after replacing the last two FC layers and the classification layer. The mini-batch gradient descent with a learning rate of 10^{-4}, a weight decay of 10^{-6}, and a momentum of 0.9 was used to fit the entire model over 100 iterations. The batch size was 16 samples. We repeated the experiment by changing the number of trainable and frozen layers each time to find the optimal level for knowledge transfer. We started by training only the last 3DCNN layer with the FC layer and the classification layer, while the remaining layers were frozen. Then, in each repetition, we incremented the number of trainable layers by activating the next nearest layer to the previously activated ones. FIGURE 7 illustrates the results of the experiment in terms of evaluation loss and recognition accuracy. It shows that the performance of the model is improved as we increase the number of trainable layers as long as the first layer is frozen. That is, the best performance (80.94%) was achieved by fine-tuning all the layers except the first one. This result supports the intuition that the first layer learns common preliminary motifs in both the source and target domains. As a result, the parameters of this layer were optimized well on the source data and there was no need to distort them by a small and maybe noisy data of the target domain.

This optimal case of knowledge transfer was used in our experiments for feature representation by taking the output of the FC layer as a feature vector for the next phase.

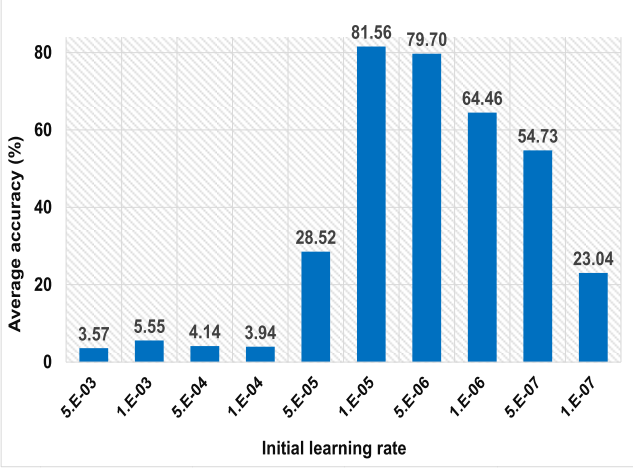

FIGURE 14. Average accuracy of all architectures with different initial.

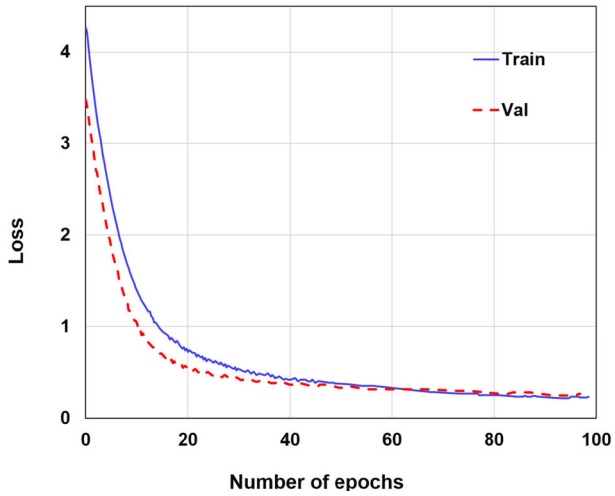

FIGURE 15. Behavior of the optimal autoencoder in the signer-independent mode.

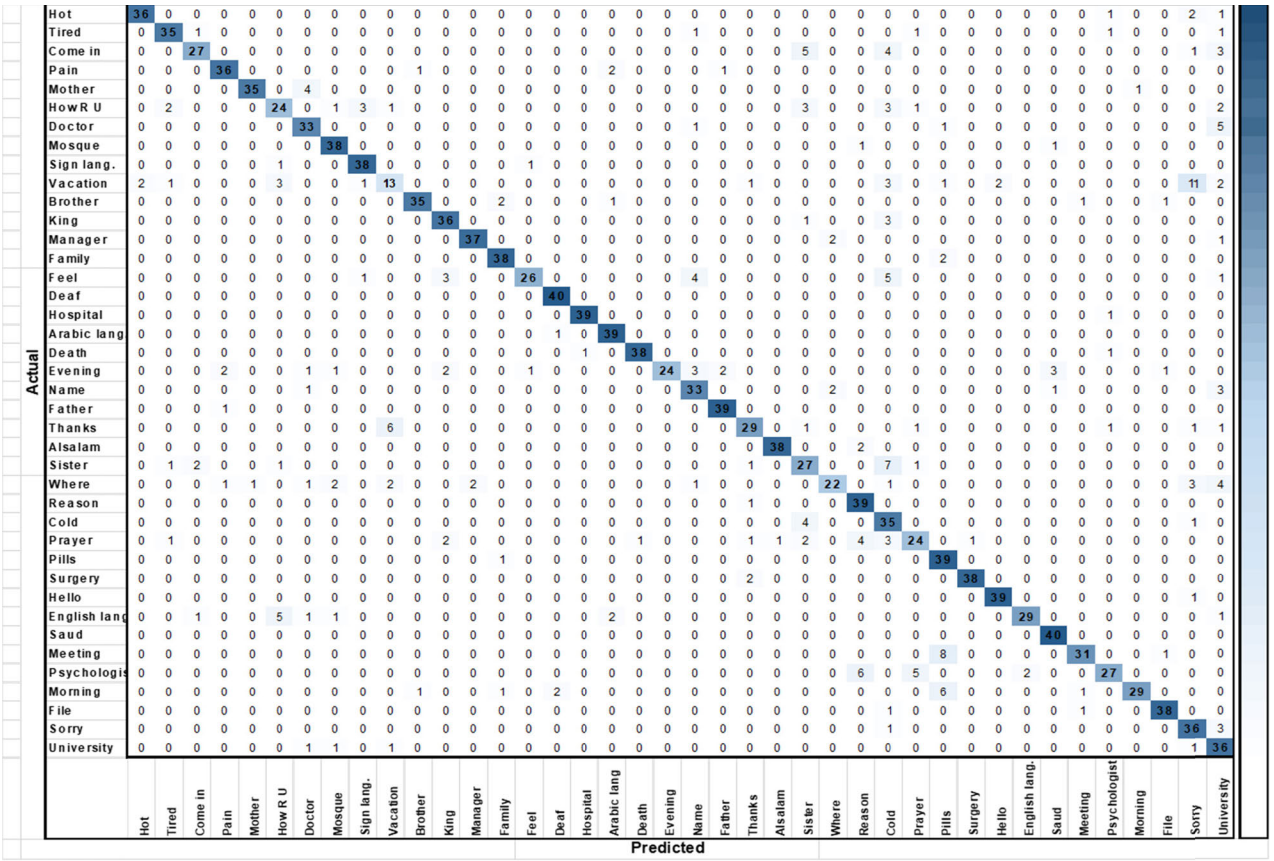

FIGURE 16. Confusion matrix of autoencoder two streams fusion in the signer-independent mode.

2) SIGNER-INDEPENDENT MODE

For data separation, we repeated the same criterion used in the previous experiment; i.e., we choose 80% of the KSU-SSL dataset for training and the remaining 20% for testing. As detailed in the input preprocessing section, the final output of the preprocessing phase was two clips. Each clip had a shape of $112 \times 112 \times 3 \times 16$, where 112×112 is the frame size, 3 is the RGB channels in each frame, and 16 is the number of frames in each clip. Each of the two clips in the training samples was used for refining the corresponding C3D instance.

In other words, we used two instances of the C3D architecture, which was optimized in the first part to separately learn the two types of features. Then, the trained instances were utilized to extract the features from the corresponding clips in the dataset samples.

To achieve this, we removed the classification layers from the two instances and replaced them with a single concatenation layer followed by a fusion and classification network. The two output vectors of the C3D modules were concatenated to form a single vector of length 8192?. End-to-end training was conducted for the whole architecture while freezing all the layers except the fusion and classification network.

3) SIGNER-DEPENDENT MODE

In contrast to the signer-independent case where the signers were divided into two mutually exclusive sets, in this scenario, we randomly selected 80% of the dataset samples for training and the remaining 20% for evaluation. Except for this data separation step, the same process was repeated as in the previous mode.

B. MLP FUSION

1) SIGNER-INDEPENDENT MODE

We investigated the MLP network for feature fusion. We studied the effect of the number of layers of the MLP (the depth) and the number of neurons per layer on the model's performance. The mini-batch gradient descent optimizer was used with an initial learning rate, a decay of 10^{-6}, and a momentum of 0.9.

We conducted an extensive grid search to optimize the architecture and the initial learning rate because they are the most important hyperparameters for the MLP fusion network.

The search space was defined as follows:

- The architecture depth in terms of number of layers $d \in \{1, 2, 3\}$.
- The number of neurons in any layer $N \in \{256, 512, 1024, 2048, 4096, 8192\}$ with a constraint that the number of neurons in any layer should be less than the number of neurons in the previous layer.
- The initial learning rate is $lr = x \times 10^n : x \in \{1, 5\}$ and $n \in \{-3, -4, -5, -6, -7\}$.

The result of this grid search step is illustrated as a heat map in FIGURE 8.

The average accuracy of all architectures with different learning rates is illustrated in **Error! Reference source not found.** From the heat map and the average accuracy figures,

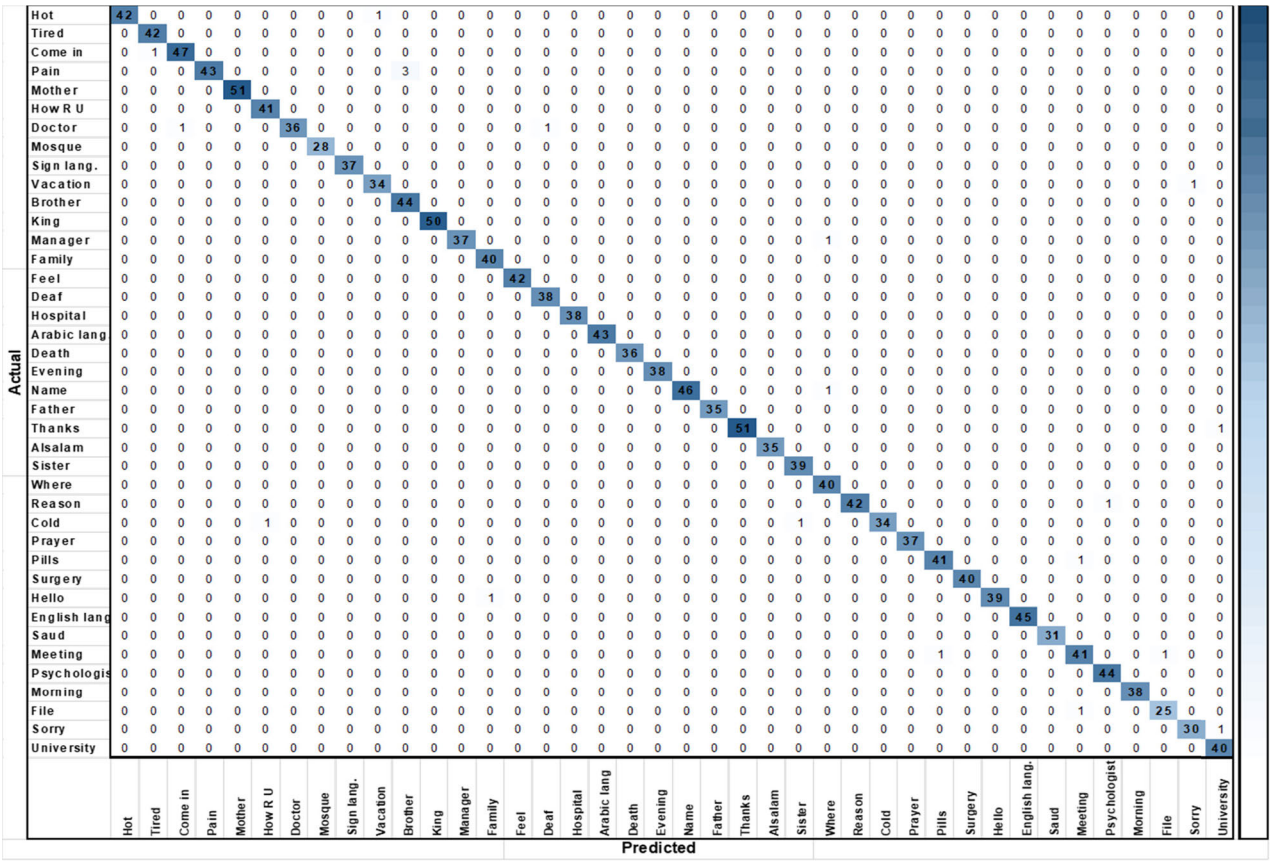

FIGURE 17. Confusion matrix of autoencoder two streams fusion in the signer-dependent mode.

we find that the learning rates between 10^{-4} and 10^{-5} achieved competing accuracies for all architectures.

The highest recognition accuracy of 87.69% was achieved by the two-layer architecture, where the first layer has 2048 neurons, the second layer has 256 neurons, and the initial learning rate is 5×10^{-5}. In addition, there is no clear trend for the performance with respect to architecture.

The behavior of the system performance during training iterations on the training and validation datasets is illustrated in FIGURE 10. The performance of the trained system on the evaluation dataset is detailed in the confusion matrix in FIGURE 11.

2) SIGNER-DEPENDENT MODE

The optimal hyperparameters obtained in the signer-independent scenario were utilized to evaluate the system performance in the signer-dependent scenario.

In this scenario, some of the signer's samples were used for training the model and the remaining of the samples performed by the same signer were used for evaluation. The ratios of the two sets of samples were random and varied from one signer to another. The evaluation results are illustrated in the confusion matrix in FIGURE 12. We find that this scenario achieved an accuracy of 98.62%.

C. AUTOENCODER FUSION

1) SIGNER-INDEPENDENT MODE

We investigated the autoencoder network for feature fusion.

TABLE 1. Recognition accuracy (%) achieved by MLP and autoencoder fusion in the different modes.

	Batch size	MLP	Autoencoder
Signer-dependent	16	98.62	**98.75**
	32	97.31	97.70
	64	97.00	97.12
Signer-independent	16	**87.69**	84.89
	32	84.17	81.87
	64	83.56	81.44

We also investigated the effect of the autoencoder depth (number of hidden layers) and width (number of neurons in each layer) on the performance of the system.

The mini-batch gradient descent optimizer was used in this part with the same parameter setup used in the MLP fusion. We conducted an intensive grid search to optimize the architecture of the autoencoder and the initial learning rate.

The search space was defined as follows:

- The encoder architecture depth in terms of the number of layers $d \in \{1, 2, 3\}$.
- The number of neurons in any layer $N \in \{256, 512, 1024, 2048, 4096, 8192\}$ with a constraint that the

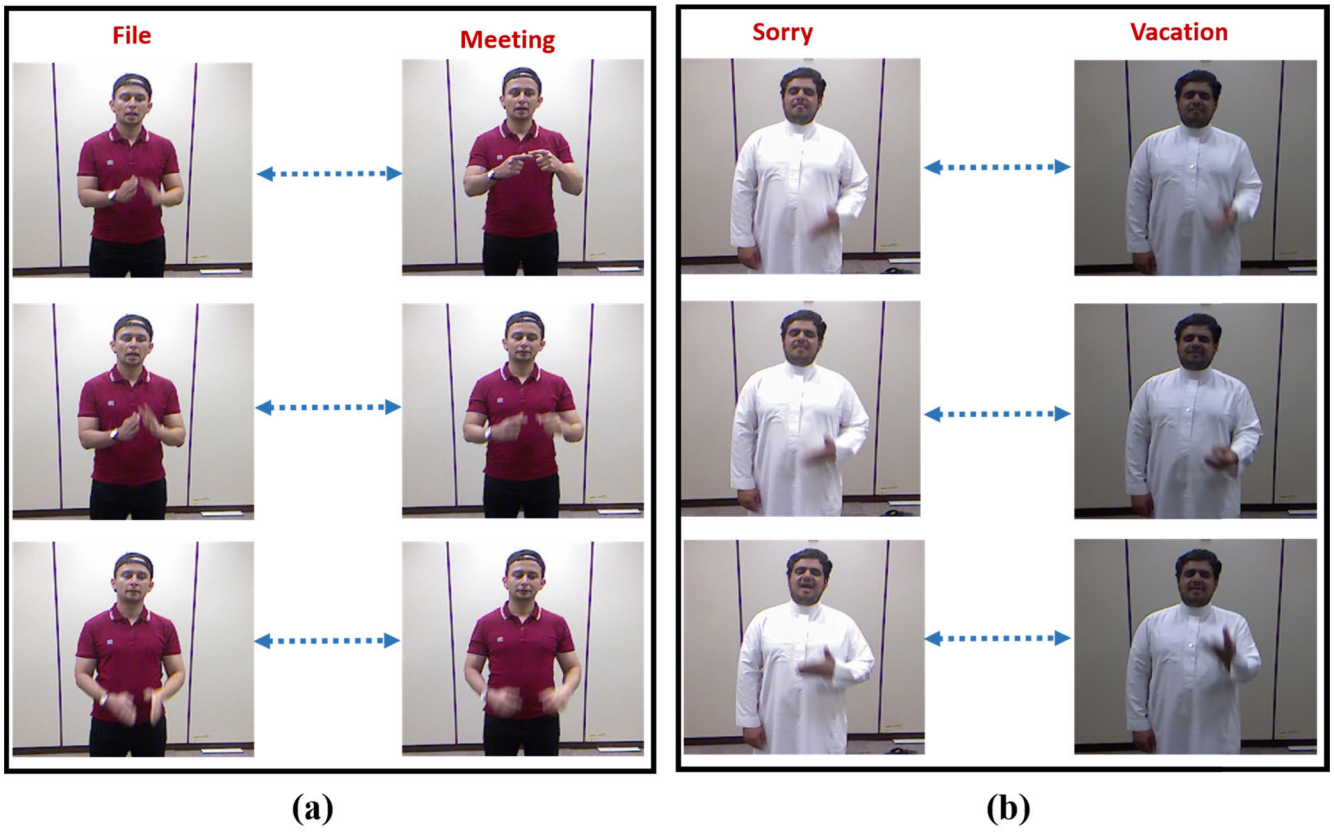

FIGURE 18. Four examples of confused gestures from the KSU-SSL dataset.

number of neurons in the layers is reduced as we move from the input layer toward the latent layer.

- The initial learning rate is $lr = x \times 10^n : x \in \{1, 5\}$ and $n \in \{-3, -4, -5, -6, -7\}$.

The result of this grid search step is illustrated as a heat map in FIGURE 13. The average accuracy of all architectures with different learning rates is illustrated in FIGURE 14. From the heat map and average accuracy figures, we find that the system with one pair of hidden layers performed better than the system with two pairs of hidden layers. We also find that the maximum accuracy and the best average accuracy were achieved using an initial learning rate of 10^{-5}.

The highest accuracy of 84.89% was achieved by the architecture with 2048 neurons in the latent layer and a single pair of hidden layers with 8192 neurons each.

The system performance during training iterations on the training and validation datasets is illustrated in **Error! Reference source not found.**. On the other hand, the recognition rate of the trained system on the evaluation dataset is detailed in the confusion matrix in FIGURE 16.

2) SIGNER-DEPENDENT MODE

The optimal hyperparameters obtained in the signer-independent scenario were utilized to evaluate the system performance in the signer-dependent scenario. The evaluation results are illustrated in the confusion matrix in FIGURE 17. A recognition accuracy of 98.75% was achieved in this scenario.

TABLE 2. Recognition accuracy (%) compared with state-of-the-art systems.

Methods ↓	Signer Dependent	Signer Independent
[37] RGB + Flow, inception (2018)	98.25	84.22
[40] DenseImage Net (2018)	78.12	59.35
[20] 3DCNN + MLP	98.12	84.38
Proposed system with MLP	98.62	**87.69**
Proposed system with autoencoder	98.75	84.89

DISCUSSION AND COMPARISON

Table I summarizes the MLP and the autoencoder accuracies using different batch sizes. We find that both architectures obtained comparable performance in the signer-dependent mode, while the performance of the MLP in the signer-independent mode was much better than that of the autoencoder. From the optimization heatmaps of both MLP and autoencoder systems, we can note that:

- In MLP, there was no change in accuracy when the depth of the architectures was changed.

TABLE 3. KSU-SSL dataset classes.

#	KSU-SSL Gesture	English Translation	#	KSU-SSL Gesture	English Translation
1	السلام عليكم	Peace be upon you	21	عملية جراحية	Surgery
2	كيف حالك؟	How are you?	22	أخصائي نفسي	Psychologist
3	تفضل	Come in	23	شكراً	Thanks
4	أصم	Deaf	24	متعب	Tired
5	أم	Mother	25	أخت	Sister
6	أسرة	Family	26	أين؟	Where?
7	دكتور	Doctor	27	السبب	Reason
8	مسجد	Mosque	28	بارد	Cold
9	مساء	Evening	29	صلاة	Prayer
10	إجازة	Vacation	30	ملف	File
11	أخ	Brother	31	جامعة	University
12	ملك	King	32	مرحبا	Hello
13	مدير	Manager	33	حار	Hot
14	بماذا تشعر	What are you feeling?	34	الملك سعود	King Saud
15	مستشفى	Hospital	35	اجتماع	Meeting
16	ألم	Pain	36	أب	Father
17	وفاة	Death	37	صباح	Morning
18	لغة الإشارة	Sign language	38	حبوب الدواء	pills
19	اسم	Name	39	آسف	Sorry
20	لغة الاشارة العربية	Arabic sign language	40	الإنجليزية لغة الاشارة	English sign language

- The performance of the autoencoder was slightly enhanced by increasing the number of neurons while fixing the depth of the architecture.
- The performance of the autoencoder was degraded when the depth of the architecture was increased.
- The smallest batch size achieved the highest accuracy for both architectures. This might be attributed to the fact that minimizing the batch size leads to updating the model weights more frequently. Even though, such updates using a few noisy samples involve a regularizing effect, which reduces generalization error.
- Moreover, in the confusion matrices, the system performance in the signer-independent mode was weaker than that in the signer-dependent mode.

As the gestures in the KSU-SSL dataset were recorded by a large number of participants, the samples of the dataset could exhibit significant variations. When the training and evaluation samples were recorded by two mutually exclusive sets of signers (i.e., the signer-independent scenario), the intra-class variation was very high, and the recognition accuracy was low. Furthermore, we investigated the highly confused classes for the two architectures by analyzing the confusion matrices. We focused more on the pairs of gestures that exhibited a high level of confusion in both the signer-dependent and signer-independent scenarios.

As illustrated in FIGURE 18, the sampled frames from some of the confused gestures showed that the signers had nearly common global body configuration and almost the same relative position and orientation for the hand. Any differences between the gestures are mainly on the fingers' configuration.

There are two pairs of confusing gestures in FIGURE 18, i.e., "Sorry" with "Vacation," "File" with "Meeting," and "Sorry" with "Vacation". It is clear in the figure that the frame sequences of each pair are highly correlated.

The proposed system gave more consideration to the hand region by dedicating a separate stream to learn the hand configuration features. This consideration led to excellent improvement in system performance. Compared to the results achieved by the base C3D architecture in the first experiment and those achieved by the temporally enhanced system in [20], this system achieved the best recognition rate with both MLP and autoencoders in all the scenarios.

Algorithm 1 Hand Region Estimation

Input: The elbow coordinates (x_e, y_e)
The wrist coordinates (x_w, y_w)
The square region length $(length)$
Output: The top left coordinate of the square region (x_B, y_B)
The bottom-right coordinate of the square region (x_E, y_E)

$Calculatetheregionlength = \sqrt{(x_w - x_e)^2 + (y_w - y_e)^2}$
If $abs\,(x_w - x_e) < \alpha and abs\,(y_w - y_e) < \alpha$
$x_B = x_w - length/2$
$y_B = y_w - length/2$
$x_E = x_w + length/2$
$y_E = y_w + length/2$
Else If $abs\,(x_w - x_e) < \alpha and abs\,(y_w - y_e) > \alpha$
$x_B = x_w - length/2$
$x_E = x_w + length/2$
If $y_w < y_e$
$y_B = y_w - (length - \varepsilon)$
$y_E = y_w + \varepsilon$
Else
$y_B = y_w - \varepsilon$
$y_E = y_w + (length - \varepsilon)$
End If

Else If $abs\,(y_w - y_e) < \alpha and abs\,(x_w - x_e) > \alpha$
$y_B = y_w - length/2$
$y_E = y_w + length/2$
If $x_w > x_e$
$x_B = x_w - \varepsilon$
$x_E = x_w + (length - \varepsilon)$
Else
$x_B = x_w - (length - \varepsilon)$
$x_E = x_w + \varepsilon$
End If
Else If $(y_e - y_w) > \alpha and\ (x_w - x_e) > \alpha$
$y_{mid} = round(y_w + (y_e - y_w)/2)$
$x_{mid} = round(x_w - (x_w - x_e)/2)$
$y_B = y_{mid} - length$
$y_E = y_{mid}$
$x_B = x_{mid}$
$x_E = x_{mid} + length$
Else If $(y_w - y_e) > \alpha and\ (x_e - x_w) > \alpha$
$y_{mid} = round(y_w - (y_w - y_e)/2)$
$x_{mid} = round(x_w + (x_e - x_w)/2)$
$y_B = y_{mid}$
$y_E = y_{mid} + length$
$x_B = x_{mid} - length$
$x_E = x_{mid}$
Else If $(y_e - y_w) > \alpha and\ (x_e - x_w) > \alpha$
$y_{mid} = round(y_w + (y_e - y_w)/2)$
$x_{mid} = round(x_w + (x_e - x_w)/2)$
$y_B = y_{mid} - length$
$y_E = y_{mid}$
$x_B = x_{mid} - length$
$x_E = x_{mid}$

Else If $(y_w - y_e) > \alpha and\ (x_w - x_e) > \alpha$

$y_{mid} = round(y_w - (y_w - y_e)/2$

$x_{mid} = round(x_w - (x_w - x_e)/2)$

$y_B = y_{mid}$

$y_E = y_{mid} + length$

$x_B = x_{mid}$

$x_E = x_{mid} + length$

Else:

Wrong input values

However, despite of this performance enhancement, the system failed in recognizing some of the confusing gestures.

The misclassifications were almost caused by hand blurring issues and bad lighting conditions, which also illustrated in FIGURE 18. The recording cameras had a frame rate of 30 fps, which was not sufficient to eliminate this motion blur. The hand configuration details targeted by this system were eliminated by the motion blur and bad lighting in many cases, which are some of the challenges of the KSU-SSL dataset.

In Table, we compare the performance of the proposed system with those of state-of-the-art systems. We noticed that there is a lack in the single-modality systems that are tested on comprehensive sign language datasets of RGB frames only. Most of the recent works utilized multimodality inputs, that compose multiple channels such as depth maps and human skeleton joints in addition to the RGB frames.

To make fair comparisons, we only considered those systems with an RGB video input rather than the systems with multimodality inputs. The selected systems for comparison used deep CNN architectures in different ways for hand gesture representation. The system in [37] generated the horizontal and vertical optical flow from the RGB sequence. These optical follow channels were stacked with the RGB frames to enhance the model performance.

On the other hand, the system in [40] started by compressing the entire input sequence in a two-dimensional matrix. This matrix was then fed as an input to the proposed architecture.

The proposed systems with MLP and autoencoder fusion outperformed the DenseImage Net by a large margin. In its worst case, the proposed system with autoencoder fusion slightly outperformed the other two state-of-the-art systems in both scenarios. The highest accuracy of 87.69% in the signer-independent scenario was achieved by the system with the MLP fusion. This outperforming result can be attributed to the enhancement of the spatial aspect as well as the good temporal modeling of the hand gesture in the proposed system.

The good performance achieved by the systems in [20] and [37] can be attributed to the efficient way of utilizing the temporal features of the hand gesture. In this regard, the system in [20] utilized 3DCNN to model three temporal segments, from the beginning, the middle, and the end of the input video and then aggregated the segments' features to achieve a robust temporal representation.

To enhance the temporal representation, the system in [37] combined the RGB frames with the auxiliary optical flow channels, which involve more temporal motifs.

On the other hand, the low accuracy of the DenseImage Net [40] might be attributed to losing the temporal aspect by compressing the entire sequence of the video in a 2D matrix and dealing with such matrix as a static image.

VI. CONCLUSION

This study proposed a novel system for dynamic hand gesture recognition via a combination of multiple deep learning techniques. The proposed system represented the hand gesture using local hand shape features as well as global body configuration features, which is very efficient for complicated structured hand gestures of the sign language. The openpose framework was used in this study for hand region detection and estimation. A robust face detection algorithm and the body parts ratios theory were utilized for gesture space estimation and normalization. Two 3DCNN instances were used separately for learning the fine-grained features of the hand shape and the coarse-grained features of the global body configuration. MLP and autoencoders were utilized to aggregate and globalize the extracted local features and the SoftMax function was used for the classification. Furthermore, to reduce the training cost of the 3DCNN module, we investigated domain adaptation and conducted extensive experiments to optimize the level of knowledge transfer. The proposed system was evaluated on a real and challenging sign language dataset. The experimental results showed that the proposed system outperformed state-of-the-art methods in terms of recognition rate, demonstrating its effectiveness.

For future work, we will utilize other strategies for temporal aspect modeling. We will perform extensive experiments to optimize the length of the input clip. We will also test the system for real-time hand gesture recognition.

APPENDIX I.

See Table 3.

APPENDIX II.

See Algorithm 1.

REFERENCES

[1] S. Kausar and M. Y. Javed, "A survey on sign language recognition," in *Proc. Frontiers Inf. Technol.*, Islamabad, Pakistan, Dec. 2011, pp. 95–98.

[2] M. B. Waldron and S. Kim, "Isolated ASL sign recognition system for deaf persons," *IEEE Trans. Rehabil. Eng.*, vol. 3, no. 3, pp. 261–271, Sep. 1995.

[3] C. Chansri and J. Srinonchat, "Hand gesture recognition for thai sign language in complex background using fusion of depth and color video," *Procedia Comput. Sci.*, vol. 86, pp. 257–260, Jan. 2016.

[4] M.-H. Yang, N. Ahuja, and M. Tabb, "Extraction of 2D motion trajectories and its application to hand gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1061–1074, Aug. 2002.

[5] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.

[6] D. Kelly, J. Mc Donald, and C. Markham, "Continuous recognition of motion based gestures in sign language," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Kyoto, Japan, Sep. 2009, pp. 1073–1080.

[7] D. Kelly, J. McDonald, and C. Markham, "A person independent system for recognition of hand postures used in sign language," *Pattern Recognit. Lett.*, vol. 31, no. 11, pp. 1359–1368, Aug. 2010.

[8] K. M. Lim, A. W. C. Tan, and S. C. Tan, "A feature covariance matrix with serial particle filter for isolated sign language recognition," *Expert Syst. Appl.*, vol. 54, pp. 208–218, Jul. 2016.

[9] H. Wang, X. Chai, and X. Chen, "Sparse observation (SO) alignment for sign language recognition," *Neurocomputing*, vol. 175, pp. 674–685, Jan. 2016.

[10] A. Mohanty, S. S. Rambhatla, and R. R. Sahay, "Deep gesture: Static hand gesture recognition using CNN," in *Proc. Comput. Vis. Image Process.*, Roorkee, India, Sep. 2017, pp. 449–461.

[11] B. Hu and J. Wang, "Deep learning based hand gesture recognition and UAV flight controls," *Int. J. Autom. Comput.*, vol. 17, no. 1, pp. 17–29, Feb. 2020.

[12] M. Alhussein and G. Muhammad, "Voice pathology detection using deep learning on mobile healthcare framework," *IEEE Access*, vol. 6, pp. 41034–41041, 2018.

[13] A. Ghoneim, G. Muhammad, S. U. Amin, and B. Gupta, "Medical image forgery detection for smart healthcare," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 33–37, Apr. 2018.

[14] R. Hou, C. Chen, and M. Shah, "An end-to-end 3D convolutional neural network for action detection and segmentation in videos," 2017, *arXiv:1712.01111*. [Online]. Available: http://arxiv.org/abs/1712.01111

[15] G. Muhammad, M. F. Alhamid, and X. Long, "Computing and processing on the edge: Smart pathology detection for connected healthcare," *IEEE Netw.*, vol. 33, no. 6, pp. 44–49, Nov./Dec. 2019.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4489–4497.

[18] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

[19] 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.

[20] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, and M. A. Mekhtiche, "Hand gesture recognition for sign language using 3DCNN," *IEEE Access*, vol. 8, pp. 79491–79509, 2020.

[21] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 6, pp. 1064–1076, Nov. 2011.

[22] V. E. Kosmidou and L. J. Hadjileontiadis, "Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 12, pp. 2879–2890, Dec. 2009.

[23] T. D. Bui and L. T. Nguyen, "Recognizing postures in vietnamese sign language with MEMS accelerometers," *IEEE Sensors J.*, vol. 7, no. 5, pp. 707–712, May 2007.

[24] G. Fang, W. Gao, and D. Zhao, "Large-vocabulary continuous sign language recognition based on transition-movement models," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 1, pp. 1–9, Jan. 2007.

[25] U. Cote-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.

[26] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. Reaching Through Technol. (CHI)*, New Orleans, LA, USA, Apr. 1991, pp. 237–242.

[27] M. M. Zaki and S. I. Shaheen, "Sign language recognition using a combination of new vision based features," *Pattern Recognit. Lett.*, vol. 32, no. 4, pp. 572–577, Mar. 2011.

[28] J. Pu, W. Zhou, J. Zhang, and H. Li, "Sign language recognition based on trajectory modeling with HMMs," in *Proc. Int. Conf. Multimedia Modeling*, Miami, FL, USA, Jan. 2016, pp. 686–697.

[29] T.-H.-S. Li, M.-C. Kao, and P.-H. Kuo, "Recognition system for home-service-related sign language using entropy-based K-means algorithm and ABC-based HMM," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 1, pp. 150–162, Jan. 2016.

[30] J. Yang, J. Yuan, and Y. Li, "Parsing 3D motion trajectory for gesture recognition," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 627–640, Jul. 2016.

[31] L. Quesada, G. López, and L. Guerrero, "Improving deaf people accessibility and communication through automatic sign language recognition using novel technologies," in *Advances in Design for Inclusion*. Orlando, FL, USA: Walt Disney World, Jul. 2016, pp. 497–507.

[32] 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.*, Cairo, Egypt, Nov. 2014, pp. 36–45.

[33] M. R. Abid, E. M. Petriu, and E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 3, pp. 596–605, Mar. 2015.

[34] J. Huang *et al.*, "Sign language recognition using convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 572–578.

[35] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 430–439, Apr. 2018.

[36] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, "A real-time hand posture recognition system using deep neural networks," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 2, pp. 1–23, May 2015.

[37] O. Kopuklu, N. Kose, and G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 2103–2111.

[38] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognit.*, vol. 76, pp. 80–94, Apr. 2018.

[39] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Boston, MA, USA, Jun. 2015, pp. 1–7.

[40] X. Chen and K. Gao, "DenseImage network: Video spatial-temporal evolution encoding and understanding," May 2018, *arXiv:1805.07550*. [Online]. Available: http://arxiv.org/abs/1805.07550

[41] S. Aly and W. Aly, "DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition," *IEEE Access*, vol. 8, pp. 83199–83212, 2020.

[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Kauai, HI, USA, Jun. 2001, p. 1.

[43] A. Özaslan, M. Y. İşcan, İ. Özaslan, H. Tuğcu, and S. Koç, "Estimation of stature from body parts," *Forensic Sci. Int.*, vol. 132, no. 1, pp. 40–45, Mar. 2003.

[44] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," Dec. 2018, *arXiv:1812.08008*. [Online]. Available: http://arxiv.org/abs/1812.08008