A Hybrid CNN-BILSTM Model for Continuous Sign Language Recognition Using Iterative Training

S. Spandana[1], Bangali Madhura[2], A. Sandhya[2], A. Manish[2], K. Prem Kumar[2]

[1]Assistant Professor, [2]UG Scholar, [1,2]Department of Computer Science Engineering
[1,2]Malla Reddy Engineering College and Management Sciences, Medchal, Hyderabad, Telangana

ABSTRACT

This work develops a continuous sign language (SL) recognition framework with deep neural networks, which directly transcribes videos of SL sentences to sequences of ordered gloss labels. Previous methods dealing with continuous SL recognition usually employ hidden Markov models with limited capacity to capture the temporal information. In contrast, our proposed architecture adopts deep convolutional neural networks with stacked temporal fusion layers as the feature extraction module, and bi-directional recurrent neural networks as the sequence learning module. We propose an iterative optimization process for our architecture to fully exploit the representation capability of deep neural networks with limited data. Our proposed neural model consists of two modules for spatiotemporal feature extraction and sequence learning, respectively. Due to the limited scale of the datasets, we find an end-to-end training cannot fully exploit the deep neural network of high complexity. To address this problem, this work investigates an iterative optimization process to train our convolutional neural network based bidirectional long-short-term-memory (CNN-BILSTM) architecture effectively. We use gloss-level gestural supervision given by forced alignment from end-to-end system to directly guide the training process of the feature extractor. Afterwards, this work fine-tunes the BILSTM system with the improved feature extractor, and the system can provide further refined alignment for the feature extraction module. Through this iterative training strategy, the proposed CNN-BILSTM can keep learning and benefiting from the refined gestural alignments. To implement this project, 'SignumDataset' dataset is used, which contains 24 different signs or signatures.

Keywords: Sign recognition, Deep learning, LSTM networks, Bi-LSTM.

1. INTRODUCTION

Sign language (SL) is commonly known as the primary language of deaf people, and usually collected or broadcast in the form of video. SL is often considered as the most grammatically structured gestural communications [1]. This nature makes SL recognition an ideal research field for developing methods to address problems such as human motion analysis, human-computer interaction (HCI) and user interface design, and makes it receive great attention in multimedia and computer vision [2], [3], [4]. Typical SL learning problems involve isolated gesture classification [3], [5], [6], [7], sign spotting [8], [9], [10], and continuous SL recognition [11], [12], [13]. Generally speaking, gesture classification is to classify isolated gestures to correct categories, while sign spotting is to detect predefined signs from continuous video streams, with precise temporal boundaries of gestures provided for training detectors. Different from these problems, continuous SL recognition is to transcribe videos of SL sentences to ordered sequences of glosses (here we use "gloss" to represent a gesture with its closest meaning in natural languages [1]), and the continuous video streams are provided without prior segmentation. Continuous SL recognition concerns more about learning unsegmented gestures of long-term video streams and is more suitable for processing continuous gestural videos in real-world systems. Its training also does not require an expensive annotation on temporal boundary for each gesture. Recognizing SL indicates simultaneous analysis and integration of gestural movements and appearance features, as well as disparate body parts [1], and therefore

probably using a multimodal approach. In this paper, we focus on the problem of continuous SL recognition on videos, where learning the spatiotemporal representations as well as their temporal matching for the labels is crucial. Many studies [11], [14], [15], [16] have made their efforts on representing SL with hand-crafted features. For example, hand and joint locations are used in [11], [17], local binary patterns (LBP) is used in [16], histogram of oriented gradients (HOG) is utilized in [15], and its extension HOG-3D is applied in [11]. Recently, deep convolutional neural networks have achieved a tremendous impact on related tasks on videos, e.g., human action recognition [18], [19], [20], gesture recognition [6] and sign spotting [9], [10], and recurrent neural networks (RNNs) have shown significant performance on learning the temporal dependencies in sign spotting [4], [21]. Several recent approaches taking advantage of neural networks have also been proposed for continuous SL recognition [12], [13], [22]. In these works, neural networks are restricted to learning frame-wise representations, and hidden Markov models (HMMs) are utilized for sequence learning. However, the frame-wise labelling adopted in [12], [13], [22] is noisy for training the deep neural networks, and HMMs might be hard to learn the complex dynamic variations, considering their limited representation capability.

This work therefore develops a recurrent convolutional neural network for continuous SL recognition. Our proposed neural model consists of two modules for spatiotemporal feature extraction and sequence learning respectively. Due to the limited scale of the datasets, we find an end-to-end training cannot fully exploit the deep neural network of high complexity. To address this problem, we investigate an iterative optimization process to train our recurrent deep neural architecture effectively. We use gloss-level gestural supervision given by forced alignment from end-to-end system to directly guide the training process of the feature extractor. Afterwards, we fine-tune the recurrent neural system with the improved feature extractor, and the system can provide further refined alignment for the feature extraction module. Through this iterative training strategy, our deep neural network can keep learning and benefiting from the refined gestural alignments.

2. LITERATURE SURVEY

SL recognition systems on videos usually consist of a feature extraction module, which extracts sequential representations to characterize gesture sequences, and a temporal model mapping sequential representations to labels. Many hand-crafted features have been introduced for gesture and SL recognition. These features characterize handshape, appearance and motion cues, by using image pixel intensity [16], gradients [11], [15], [23] and motion trajectories or velocities [8], [11], [17]. In recent years, there has been a growing trend to learn feature representations by deep neural networks. Wu et al. [24] employ a deep belief network to extract high-level skeletal joint features for gesture recognition. Convolutional neural networks (CNNs) [25], [26] and 3D convolutional neural networks (3D-CNNs) [9], [10], [4] have also been employed to capture visual cues for hand regions. For instance, Molchanov et al. [4] apply 3D-CNNs for spatiotemporal feature extraction from video streams on color, depth and optical flow data.

Neverova et al. [9] present a multi-scale deep architecture on color, depth data and handcrafted pose descriptors. Temporal model is to learn the correspondences between sequential representations and gloss labels. HMMs are the most widely used temporal models in SL recognition [10], [11], [13]. Besides, dynamic time warping (DTW) [16] and SVMs [27] are also used for measuring similarity between gestures. Recently, RNNs have been successfully applied to sequential problems such as speech recognition [28] and machine translation [29], [30], and some progress has also been made for exploring the application of RNNs in SL recognition. Pigou et al. [21] propose an end-to-end neural model with temporal convolutions and bidirectional recurrence for sign spotting, which is taken as frame-wise classification in their framework. However, with only weak supervision in sentence level,

recurrent neural networks are hard to learn to match the over-length input sequence frame by frame with the ordered labels. Different from their model, we use temporal pooling layers to integrate the temporal dynamics before the bidirectional recurrence. Molchanov et al. [4] employ a recurrent 3D-CNN with connectionist temporal classification (CTC) [31] as the cost function for gesture recognition, while in our experiments, we find that our architecture shows a much superior performance compared to 3D-CNN model on the SL recognition benchmarks.

Due to lack of temporal boundaries for the sign glosses in the image sequences, continuous SL recognition is also a typical weakly supervised learning problem. There have been some attempts focusing on the problem of mining gestures of interest from large amount of SL videos, where signs and annotations are usually coarsely aligned with considerable noise. Different from our problem, they usually take more focus on local temporal dynamics but not long-term dependencies. Buehler et al. [15] propose a scoring function based on multiple instance learning (MIL) and search for signs of interest by maximizing the score. Pfister et al. [27] use subtitle text, lip and hand motion cues to select candidate temporal windows, and these candidates are further refined using MISVM [32]. Chung and Zisserman [33] use a ConvNet learned on image encoding representing human keypoint motion for recognition, and they locate temporal positions of signs via saliency map by back-propagation.

There have been a few works exploring the problem of continuous SL recognition. Gweth et al. [34] employ a one hidden-layer perceptron to estimate posterior from appearance based features, and use the probabilities as inputs to train an HMM-based recognition system. Koller et al. [12], [13], [25] adopt CNNs for feature extraction from cropped hand regions and also use HMMs to model the temporal relationships. As the amount of training data is not sufficient enough, training of deep neural networks is inclined to end in overfitting. To alleviate this problem, Koller et al. [12] embed a CNN within a weakly supervised learning framework. Weakly labelled sequence of hand shape annotations are brought in as an initialization, to iteratively train CNN and re-estimate hand shape labels within Expectation Maximization (EM) [35] framework. Similarly, annotations of finger and palm orientations are also imported as weakly supervised information to train CNN [25]. In their later works [13], [22], they use the frame-state alignment, provided by a baseline HMM recognition system, as frame labelling to train the embedded neural networks. In contrast with these works [12], [13], [25], [22], our sequence learning module of recurrent neural networks with end-toend training shows much more learning capacity and better performance for the dynamic dependencies. Besides, instead of using noisy frame-wise labelling as training targets of neural networks, we adopt the gloss-level alignment proposal to train our feature extraction module, which takes more local temporal dynamics into consideration. Moreover, no extra supervisory information such as hand shape annotations is imported in our approach. Notice that the development of such lexicon requires laborious annotation with expert knowledge, while our method is free from this limitation.

Different from the previous work [36], this project proposes a distinctive segment-gloss alignment method to learn from the outputs of our sequence learning module, and we provide an explicit illustration for our iterative training scheme, by proving the training of feature extraction module to be maximizing the lower bound of the objective function, instead of using an intuitive approach. We also contribute by investigating more on the multimodal integration of appearance and motion cues in this work.

3. PROPOSED SYSTEM

Our proposed neural model consists of two modules for spatiotemporal feature extraction and sequence learning, respectively. Due to the limited scale of the datasets, we find an end-to-end training cannot fully exploit the deep neural network of high complexity. To address this problem, we investigate an iterative optimization process to train our recurrent deep neural architecture effectively.

We use gloss-level gestural supervision given by forced alignment from end-to-end system to directly guide the training process of the feature extractor. Afterwards, we fine-tune the recurrent neural system with the improved feature extractor, and the system can provide further refined alignment for the feature extraction module. Through this iterative training strategy, our deep neural network can keep learning and benefiting from the refined gestural alignments. The main contributions of our work can be summarized as follows:

1) We develop our architecture with recurrent convolutional neural networks of more learning capacity to achieve state-of-the-art performance on continuous SL recognition, without importing extra supervisory information.

2) We design an iterative optimization process for training our deep neural network architecture, and our approach, with the neural networks better exploited, is proved to take notable effect on the limited training set in contrast to the end-to-end trained system.

3) We propose a multimodal version of our framework with RGB frames and optical flow frames and experiments present that our multimodal fusion scheme provides better representations for the gestures and further improves the performance of the system.

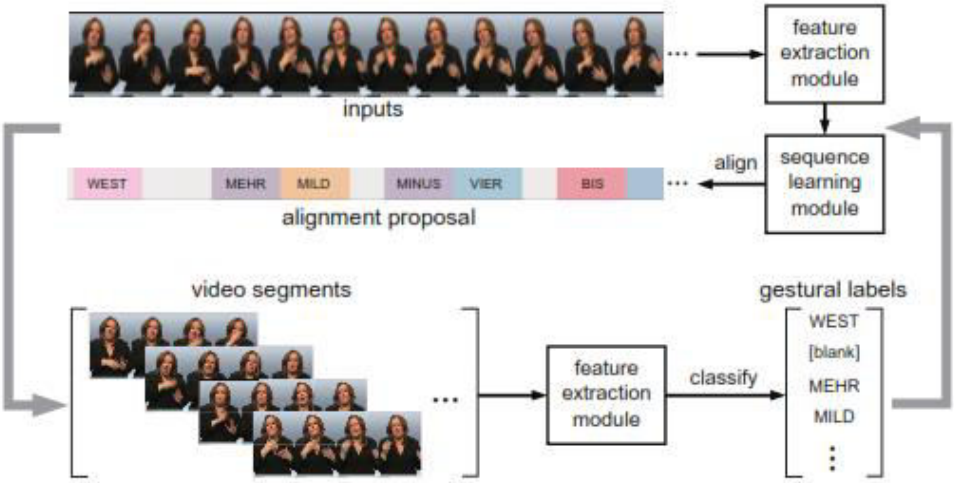

Fig.1. Iterative training process.

3.1 Model design

The proposed deep neural architecture consists of a deep CNN followed by temporal operations for representation learning, and Bi-LSTMs for sequence learning. For experiments with modalities from dominant hands as the inputs, we build the deep convolutional network based on the VGG-S model (from layer conv1 to fc6), which is memory-efficient and shows competitive classification performance on ILSVRC-2012 dataset. The input frames, which are the region of dominant hands signped from original frames, are resized to 101 × 101 in dimension, and they are then transformed to 1024-dimensional feature vectors through the fully connected layer fc6. The stacked temporal convolution and pooling layers are utilized to generate spatiotemporal representation for each segment.

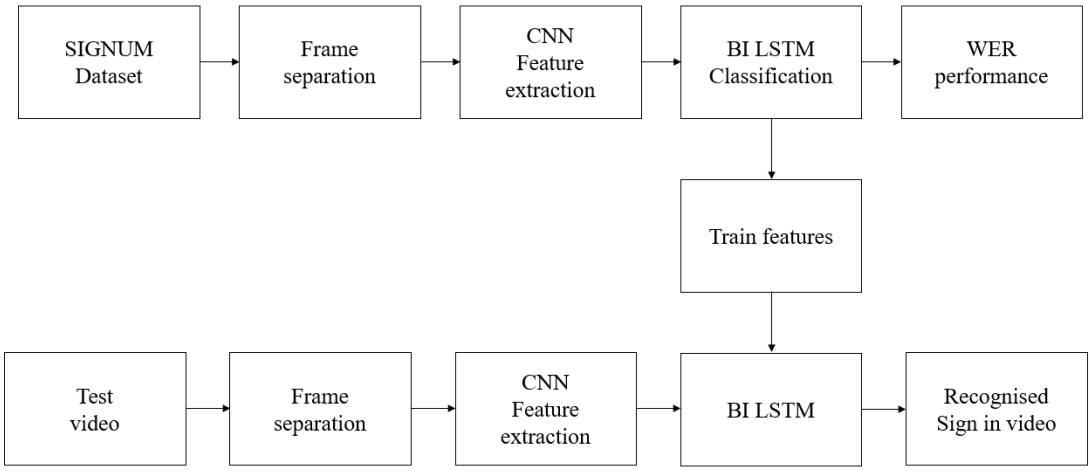

Fig.2. Proposed block diagram

Figure 2 shows the block diagram of proposed method, which is used identify the different signs from the test video using CNN feature extraction and BI-LSTM training. We select the temporal stride δ to ensure sufficient overlapping between neighboring segments, as well as pool the representation sequence to a moderate length. In the feature extraction module, rectifier and max pooling are adopted for all the nonlinearity and pooling operations. We use Bi-LSTMs with 2 × 512 dimensional hidden states and peephole connections to learn the temporal dependencies. The hidden states are then fed into the SoftMax classifier, with the dimension equal to the vocabulary size. We also investigate the performance of our training framework with full video frames as the inputs. We use CNN as the deep convolutional network in our feature extractor, and we adopt two stacked Bi-LSTMs to build the sequence learning module. Due to the limitations on GPU memory to fit in the whole system, we fix the parameters of CNN at the end-to-end stage and only tune the sequence learning module. The video frames are resized to 224×224 as the inputs of CNN, transformed to feature vectors after the average pooling layer, and then fed into the temporal fusion layers.

3.2 SIGNUM Dataset

The SIGNUM Database was created within the framework of a research project at the Institute of Man–Machine Interaction, located at the RWTH Aachen University in Germany. The SIGNUM (Signer-Independent Continuous Sign Language Recognition for Large Vocabulary Using Subunit Models) project was funded by the Deutsche Forschung gemeinschaft (German Research Foundation) and aimed to develop a video-based automatic sign language recognition system. In order to ensure user-friendliness, the system utilizes a single-color video camera for data acquisition. Since sign languages make use of manual and facial means of expression both channels are analysed by means of frame processing. The whole system, particularly the feature extraction and the subsequent classification stage, is designed for signer-independent operation and allows adaptation to an unknown signer. The reader interested in a more detailed description of this recognition system or an in-depth introduction to gesture and sign language recognition is directed to the publication list.

3.3 Multimodal Fusion

To incorporate the appearance and motion information, we also take colour frame and optical flow for dominant hand regions as the inputs of our deep neural architecture. We adopt sum fusion approach at the conv5 layer for fusing the two stream networks. It computes element-wise sum of the two feature maps at the same spatial location and channel for the fusion. Our intention here is to put appearance and motion cues at the same spatial position in correspondence, without introducing extra filters in

order to join the feature maps together. The sum fusion approach also shows a decent performance on the task of action recognition in video compared to other spatial fusion methods.

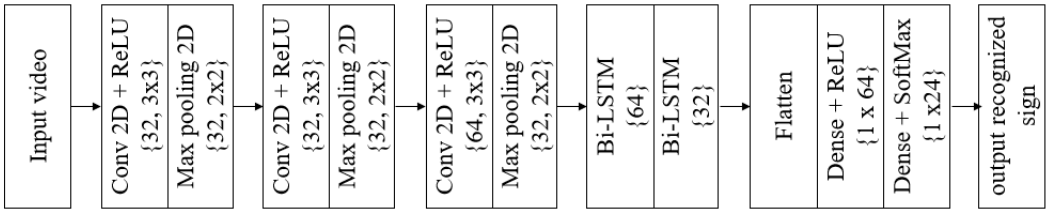

Fig. 3. Deep neural architecture for RGB and optical flow modalities of dominant hands.

Our end-to-end architecture for SL recognition from dominant hands is depicted in Fig. 2. Note that parameters for different modalities are not shared before the sum fusion. Figure 3 discloses the architecture of CNN-BILSTM that is utilized in proposed methodology for system for enhanced feature representation of word frame over conventional retrieval systems.

3.4 CNN-BILSTM

According to the facts, training and testing of CNN-BILSTM involves in allowing every source frame via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from [0,1].

Convolution layer as depicted in Figure 4 is the primary layer to extract the features from a source frame and maintains the relationship between pixels by learning the features of frame by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source frame $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of a frame (here $d = 3$, since the source frame is RGB) and a filter or kernel with similar size of input frame and can be denoted as $F(k_x, k_y, d)$.

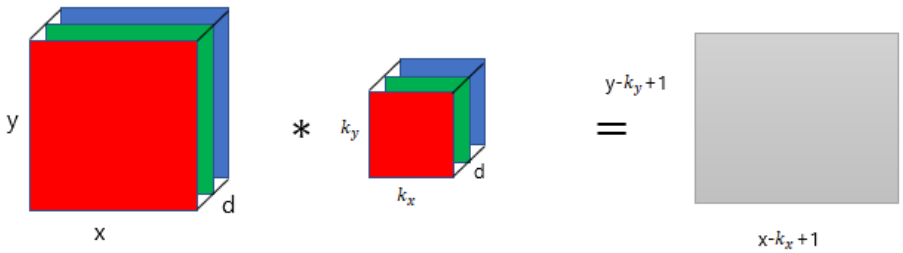

Fig. 4. Representation of convolution layer process.

The output obtained from convolution process of input frame and filter has a size of $C\left((x - k_x + 1), (y - k_y + 1), 1\right)$, which is referred as feature map. An example of convolution procedure is demonstrated in Figure 5. Let us assume an input frame with a size of 5×5 and the filter having the size of 3×3. The feature map of input frame is obtained by multiplying the input frame values with the filter values as given in Figure 5 (b).

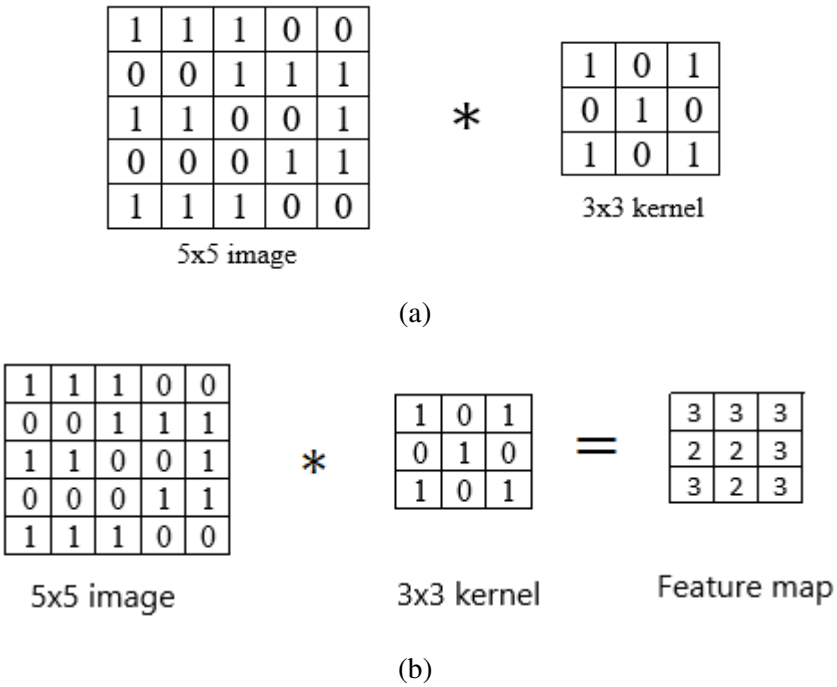

Fig. 5. Example of convolution layer process (a) a frame with size $\mathbf{5 \times 5}$ is convolving with $\mathbf{3 \times 3}$ kernel (b) Convolved feature map

3.4.1 ReLU layer

Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

3.4.2 Max pooing layer

This layer mitigates the number of parameters when there are larger size frames. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

3.5 SoftMax classifier

Generally, as seen in the above picture SoftMax function is added at the end of the output since it is the place where the nodes are meet finally and thus, they can be classified. Here, X is the input of all the models and the layers between X and Y are the hidden layers and the data is passed from X to all the layers and Received by Y. Suppose, we have 10 classes, and we predict for which class the given input belongs to. So, for this what we do is allot each class with a particular predicted output. Which means that we have 10 outputs corresponding to 10 different class and predict the class by the highest probability it has.

In Figure 6, and we must predict what is the object that is present in the picture. In the normal case, we predict whether the sign is A. But in this case, we must predict what is the object that is present in the picture. This is the place where SoftMax comes in handy. As the model is already trained on some data. So, as soon as the picture is given, the model processes the pictures, send it to the hidden layers and then finally send to SoftMax for classifying the picture. The SoftMax uses a One-Hot encoding

Technique to calculate the cross-entropy loss and get the max. One-Hot Encoding is the technique that is used to categorize the data. In the previous example, if SoftMax predicts that the object is class A then the One-Hot Encoding for:

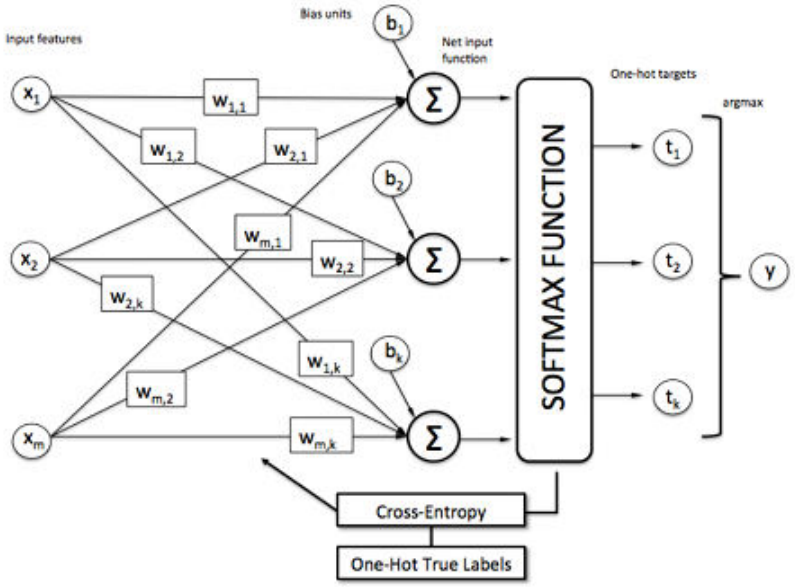

Fig.6. Sign class prediction using SoftMax classifier.

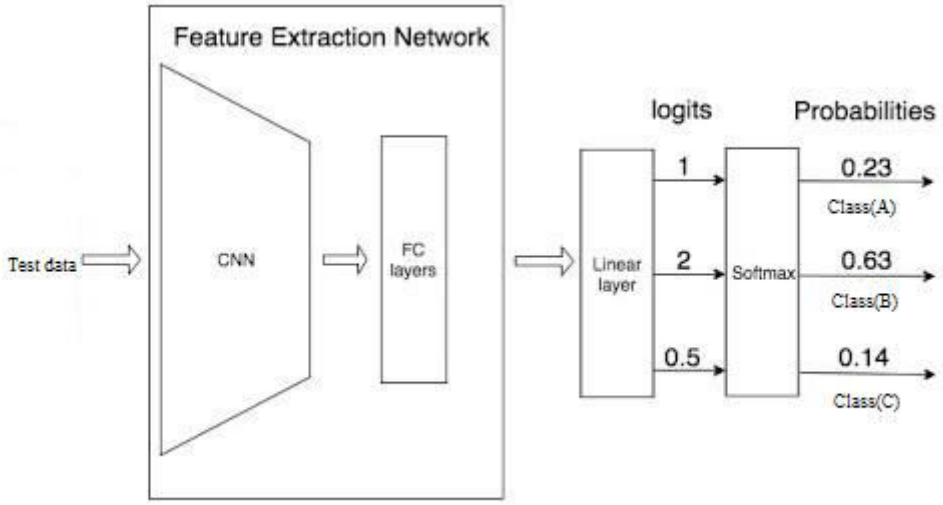

Fig.7. Example of SoftMax classifier.

Class A will be [1 0 0]

Class B will be [0 1 0]

Class C will be [0 0 1]

From the Figure 7, we see that the predictions are occurred. But generally, we don't know the predictions. But the machine must choose the correct predicted object. So, for machine to identify an object correctly, it uses a function called cross-entropy function. So, we choose more similar value by using the below cross-entropy formula.

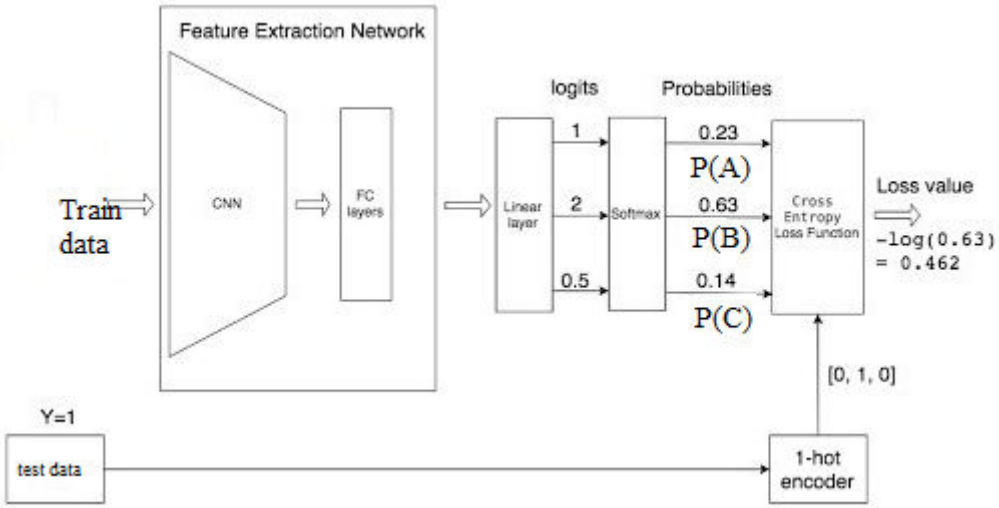

Fig.8. Example of SoftMax classifier with test data.

In the above example from figure 8 we see that 0.462 is the loss of the function for class specific classifier. In the same way, we find loss for remaining classifiers. The lowest the loss function, the better the prediction is. The mathematical representation for loss function can be represented as: -

$$LOSS = np.sum(-Y * np.log(Y\_pred))$$

4. RESULTS

This section gives the detailed analysis of simulation results implemented using "python environment". Further, the performance of proposed method is compared with existing methods using same dataset.

(a) (b) (c)

(d)

Fig.9. predicted outcomes (a) sign recognized as B, (b) sign recognized as S, (c) sign recognized as S, (d) sign recognized as U.

Figure 9 shows the predicted signs from the test video and Figure 10 shows the WER graph for multiple number of epochs (iterations).

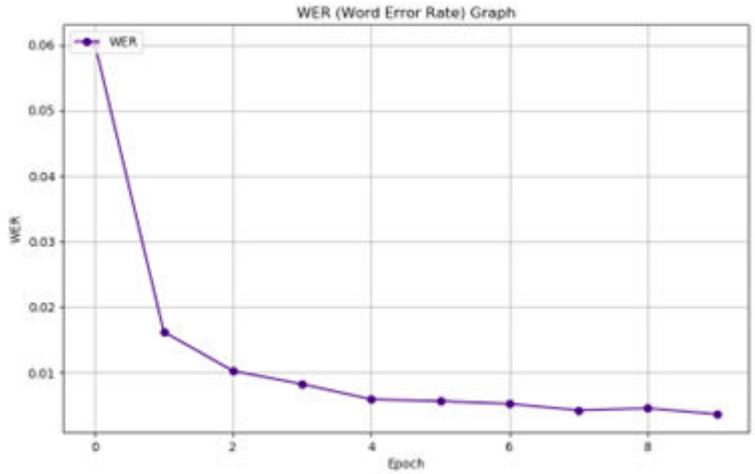

Fig.10. word error rate (WER) graph.

5. CONCLUSION

In this proposal, we have developed a continuous SL recognition system with recurrent convolutional neural networks on multimodal data of RGB frames and optical flow images. In contrast to previous state-of-the-art methods, our framework employs recurrent neural networks as the sequence learning module, which shows a superior capability of learning temporal dependencies compared to HMMs. The scale of training data is the bottleneck in fully training a deep neural network of high complexity on this task. To alleviate this problem, a novel CNN-BILSTM training scheme is proposed to make our feature extraction module fully exploited to learn the relevant gestural labels on video segments and keep on benefitting from the iteratively refined alignment proposals. In addition, a multimodal fusion approach also developed to integrate appearance and motion cues from SL videos, which presents better spatiotemporal representations for gestures. Further, our model is evaluated on two publicly available SL recognition benchmarks, and experimental results present the effectiveness of our method, where both the iterative training strategy and the multimodal fusion contribute to a better representation and the performance improvement

REFERENCES

[1] S. C. Ong and S. Ranganath, “Automatic sign language analysis: A survey and the future beyond lexical meaning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 873–891, 2005.

[2] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, “Robust part-based hand gesture recognition using Kinect sensor,” IEEE Trans. Multimedia, vol. 15, no. 5, pp. 1110–1120, 2013.

[3] C. Wang, Z. Liu, and S.-C. Chan, “Superpixel-based hand gesture recognition with Kinect depth camera,” IEEE Trans. Multimedia, vol. 17, no. 1, pp. 29–39, 2015.

[4] 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 network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2016, pp. 4207–4215.

[5] H. Cooper, E. J. Ong, N. Pugeault, and R. Bowden, “Sign language recognition using sub-units,” J. Mach. Learning Research, vol. 13, pp. 2205–2231, 2012.

[6] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, “Hand gesture recognition with 3D convolutional neural networks,” in IEEE Conf. Comput. Vis. Pattern Recog. Workshops, 2015, pp. 1–7.

[7] G. Zen, L. Porzi, E. Sangineto, E. Ricci, and N. Sebe, “Learning personalized models for facial expression analysis and gesture recognition,” IEEE Trans. Multimedia, vol. 18, no. 4, pp. 775–788, 2016.

[8] G. D. Evangelidis, G. Singh, and R. Horaud, “Continuous gesture recognition from articulated poses,” in Eur. Conf. Comput. Vis. Workshops, 2014, pp. 595–607.

[9] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, “Multi-scale deep learning for gesture detection and localization,” in Eur. Conf. Comput. Vis. Workshops, 2014, pp. 474–490.

[10] D. Wu, L. Pigou, P.-J. Kindermans, N. Le, L. Shao, J. Dambre, and J.-M. Odobez, “Deep dynamic neural networks for multimodal gesture segmentation and recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 8, pp. 1583–1597, 2016.

[11] O. Koller, J. Forster, and H. Ney, “Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers,” Comput. Vis. Image Understand., vol. 141, pp. 108–125, 2015.

[12] O. Koller, H. Ney, and R. Bowden, “Deep hand: how to train a CNN on 1 million hand images when your data is continuous and weakly labelled,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2016, pp. 3793–3802.

[13] O. Koller, S. Zargaran, H. Ney, and R. Bowden, “Deep sign: hybrid CNN-HMM for continuous sign language recognition,” in Proc. Brit. Mach. Vis. Conf., 2016.

[14] U. Von Agris, M. Knorr, and K.-F. Kraiss, “The significance of facial features for automatic sign language recognition,” in 8th IEEE Int. Conf. Autom. Face Gesture Recog., 2008, pp. 1–6.

[15] P. Buehler, A. Zisserman, and M. Everingham, “Learning sign language by watching TV (using weakly aligned subtitles),” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2009, pp. 2961–2968.

[16] L.-C. Wang, R. Wang, D.-H. Kong, and B.-C. Yin, “Similarity assessment model for Chinese sign language videos,” IEEE Trans. Multimedia, vol. 16, no. 3, pp. 751–761, 2014.

[17] C. Monnier, S. German, and A. Ost, “A multi-scale boosted detector for efficient and robust gesture recognition,” in Eur. Conf. Comput. Vis. Workshops, 2014, pp. 491–502.

[18] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 2625–2634.

[19] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, “Towards good practices for very deep two-stream convnets,” arXiv preprint arXiv:1507.02159, 2015.

[20] Y. Shi, Y. Tian, Y. Wang, and T. Huang, “Sequential deep trajectory descriptor for action recognition with three-stream CNN,” IEEE Trans. Multimedia, vol. 19, no. 7, pp. 1510–1520, 2017.

[21] L. Pigou, A. v. d. Oord, S. Dieleman, M. M. Van Herreweghe, and J. Dambre, “Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video,” Int. J. Comput. Vis., pp. 1–10, 2015.

[22] O. Koller, S. Zargaran, and H. Ney, “Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2017.

[23] T. Pfister, J. Charles, and A. Zisserman, “Domain-adaptive discriminative one-shot learning of gestures,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 814–829.

[24] D. Wu and L. Shao, “Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2014, pp. 724–731.

[25] O. Koller, H. Ney, and R. Bowden, “Automatic alignment of HamNoSys subunits for continuous sign language recognition,” in Int. Conf. Language Resources and Evaluation Workshops, 2016, pp. 121–128.

[26] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign language recognition using convolutional neural networks,” in Eur. Conf. Comput. Vis. Workshops, 2014, pp. 572–578.

[27] T. Pfister, J. Charles, and A. Zisserman, “Large-scale learning of sign language by watching TV (using co-occurrences),” in Proc. Brit. Mach. Vis. Conf., 2013.

[28] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in Proc. Int. Conf. Mach. Learning, 2014, pp. 1764–1772.

[29] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. Int. Conf. Learning Representations, 2014.

[30] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 3104–3112.

[31] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connection- ` ist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proc. Int. Conf. Mach. Learning, 2006, pp. 369–376.

[32] S. Andrews, I. Tsochantaridis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in Proc. Adv. Neural Inf. Process. Syst., 2002, pp. 577–584.

[33] J. S. Chung and A. Zisserman, “Signs in time: Encoding human motion as a temporal image,” in Eur. Conf. Comput. Vis. Workshop on Brave New Ideas for Motion Representations, 2016.

[34] Y. L. Gweth, C. Plahl, and H. Ney, “Enhanced continuous sign language recognition using pca and neural network features,” in IEEE Conf. Comput. Vis. Pattern Recog. Workshops, 2012, pp. 55–60.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” Journal of the royal statistical society. Series B (methodological), pp. 1–38, 1977.

[36] R. Cui, H. Liu, and C. Zhang, “Recurrent convolutional neural networks for continuous sign language recognition by staged optimization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2017.

[37] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” Neural Networks, vol. 18, no. 5, pp. 602–610, 2005.

[38] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in Proc. Brit. Mach. Vis. Conf., 2014.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vision, vol. 115, no. 3, pp. 211–252, 2015.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2015, pp. 1–9.

[42] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2016, pp. 1933–1941.

[43] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “Deepflow: Large displacement optical flow with deep matching,” in Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 1385–1392.

[44] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[46] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learning Representations, 2015.

[47] Theano Development Team, “Theano: A python framework for fast computation of mathematical expressions,” arXiv preprint arXiv:1605.02688, 2016.

[48] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney, “Extensions of the sign language recognition and translation corpus RWTHPHOENIX-Weather,” in Int. Conf. Language Resources and Evaluation, 2014, pp. 1911–1916.