I.J. Intelligent Systems and Applications, **2018, 10, 63-71**
Published Online October 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2018.10.07

Selfie Sign Language Recognition with Convolutional Neural Networks

P.V.V. Kishore, G. Anantha Rao, E. Kiran Kumar, M. Teja Kiran Kumar, D. Anil Kumar
Department of Electronics and Communication Engineering, K L University, Green Fields, Vaddeswaram, Guntur, India
E-mail: pvvkishore@kluniversity.in, ananth.gondu@gmail.com, kiraneepuri@kluniversity.in, mtejakiran@kluniversity.in, danilmurali@kluniversity.in

Received: 22 November 2017; Accepted: 09 February 2018; Published: 08 October 2018

Abstract**—**Extraction of complex head and hand movements along with their constantly changing shapes for recognition of sign language is considered a difficult problem in computer vision. This paper proposes the recognition of Indian sign language gestures using a powerful artificial intelligence tool, convolutional neural networks (CNN). Selfie mode continuous sign language video is the capture method used in this work, where a hearing-impaired person can operate the Sign language recognition (SLR) mobile application independently. Due to non-availability of datasets on mobile selfie sign language, we initiated to create the dataset with five different subjects performing 200 signs in 5 different viewing angles under various background environments. Each sign occupied for 60 frames or images in a video. CNN training is performed with 3 different sample sizes, each consisting of multiple sets of subjects and viewing angles. The remaining 2 samples are used for testing the trained CNN. Different CNN architectures were designed and tested with our selfie sign language data to obtain better accuracy in recognition. We achieved 92.88 % recognition rate compared to other classifier models reported on the same dataset.

Index Terms**—**Selfie sign language, Convolutional Neural Networks (CNN), Stochastic pooling, Sign language recognition (SLR), Deep learning.

I. Introduction

Sign language recognition (SLR) is an evolving research area in computer vision. The challenges in SLR are video trimming, sign extraction, sign video background modelling, sign feature representation and sign classification. All the problems [1] are attempted in the past have met considerable amount of success and are instrumental in development of the state of the algorithms for SLR. Gesture recognition uses powerful imaging and artificial intelligence based algorithms for classification [2]. Current trends show an urge to bring gesture recognition into mobile environments [3].

Sign language is visual mode of communication between two hearing impaired or hard hearing people. The communication foundations are based on finger shapes, hand shapes, hand movements in space with respect to body, hand orientations and facial expressions. The humans are trained exclusively to hand such huge amounts of information for years. For machine translation, the problem transforms into a 2D natural language processing problem. Many 1D/2D/3D models are proposed in literature with little success to bring the model close to real time implementation [4-7].

In this work, the focus will be to recognize signs of Indian sign language using 2D selfie video captured using a mobile front camera. Even though the development of a mobile app is far from reality, the objective is to simulate algorithms that can optimally execute on a mobile platform. The primary module is to extract information frames to reduce input video data per frame. A visual attention based frame work proposed in [8] is chosen for accuracy and computation time. The model works well for constant video backgrounds and we will limit our work to this typical video sets.

Fig.1. Sample data base of selfie sign language

Unavailability of benchmark datasets for Selfie mode Indian sign language (ISL) prompted us to create our own dataset. The dataset is having 200 ISL commonly used words performed by 5 native ISL users (i.e. 5 sets) in 5 different viewing angles (user dependent angles) at a rate

of 30fps. Training is initiated with three different batch sizes. In Batch-I of training only one set, i.e. 200 signs performed by 1 user in 5 different viewing angles for 2 seconds at 30fps, total of $200\times1\times5\times2\times30=60000$ sign images. Batch-II of training is done using 2 sets i.e. a total of $200\times2\times5\times2\times30=120000$ sign images. In Batch-III of training 3 sets of sign images were used. The trained CNN's are tested with two discrete video sets having different signers and viewing angles with varying backgrounds. The robustness testing is performed in two cases. In case-I of testing same dataset i.e. already trained dataset is used and in case-II of testing different dataset is used. Figure 1 shows the sample data base created for this work. The performance of the CNN algorithms is measured based on their accuracy in recall and recognition rates.

The rest of the paper is as follows: Section 2 discuss the related works. In section 3, the proposed architecture of CNN is described. Section 4 discuss the results obtained in different training and testing cases. Finally, section 5 concludes the outcomes of this paper.

II. Related Works

Sign language recognition (SLR) has transformed with technology upgradation from 1D, 2D to 3D models in the last 2 decades. In 1D, SLR is based on 1D signals acquired from a hand gloves [8] and classified using signal processing methods [9].

Bhuyan et al. [10] used hand shapes and hand trajectories to recognize static and dynamic hand signs from ISL. Zhou and Chen [11] proposed a signer adaptation method, in which maximum a posteriori estimation was combined with iterative vector field smoothing to reduce the amount of data to be translated. In the past decade, with the advances in efficient computing and bigger parallel corpora, more efficient algorithms have been developed for training [12] and generation [13].

The attributes for a sign language recognizer chosen are global shape features using Haar wavelet [14] for hand and body shapes, small hand variations are captured using 2D point cloud generated from harr wavelet and local binary pattern (LBP) [15]. Other state of the art features such as Histogram of Oriented features (HOG), scale invariant feature transform (SIFT) and speed up robust features (SURF) are used by various researchers. Various classifiers are used in recognizing and classifying the Indian sign language gestures such as ANN [16], SVM, Adaboost [17]. Many hand crafted features were extracted by researchers in classifying the objects using traditional methods.

In recent research, application of deep learning in object recognition is most suitable. CNN is powerful in solving most computer vision based tasks [18-22] such as object recognition [23], classification [24]. Classifying at faster rate on a huge dataset is a complicated problem Without the knowledge of expert using deep hidden layers CNN extracts image information and avoids the process of complex feature extraction.

Andrew Ng , Hinton, LeCun , Bengio et al. have performed fundamental research on CNNs to achieve improved performance of CNN algorithms and structural optimization [25-28]. Yann LeCun et al. in [29], highlighted that deep CNN is a breakthrough in image, video, audio and speech processing. Karpathy, Andrej et al. in [30] implemented a multiresolution CNN architecture to classify the 1 million large scale videos of 487 classes from YouTube and achieved an improved classification rates. In [31], Simonyan, Karen et al. incorporated the spatial and temporal networks and proposed a two-stream ConvNet architecture for action recognition/classification in videos. So far, no extensive research has done which explores deep CNN for selfie sign language recognition. The aim of this paper is to bring out the CNN performance in recognizing the selfie mode sign language gestures.

Deep learning methods can also be used in evaluating the sign language recognition system. However, still there is great space for further improvements. Existing datasets for selfie sign language recognition is limited, but CNN needs large amount of data for training. Deep CNN is suitable for giving solutions to complex problems with huge quantity of data [32]. For example, the classification accuracy is improved in ImageNet dataset [33] which has 1.2 million images almost covering 1000 categories. In such cases we need to consider how to take advantage of CNN.

With convolutional neural networks, we need to consider how to design and train a network that adapts to various objects. The major problem to be solved is with the quality and sizes of the images. The unbalanced amounts of low and high quality images in the dataset leads to the unbalanced classification.

The motivation for implementing the deep CNN model for selfie sign language recognition is to remove the barrier between the hearing impaired and the normal hearing person with a simple mobile based application, such that the communication gap between them will be erased. Since the feature learning in CNNs is a highly automated from the input images, avoids the complexity in extracting the various features for traditional classifiers for sign recognition.

Through the deep architecture, the learned features are deemed as the higher level abstract representation of low level sign images. Hence, we develop the deep CNN model for selfie sign language recognition in this paper.

In this paper, a novel CNN based selfie sign language recognition is proposed to achieve higher recognition rates. Different CNN architectures are implemented, tested on our selfie data to bring out the best architecture for recognition. Three different pooling techniques namely mean pooling, max pooling and stochastic pooling are implemented and found stochastic pooling is the best for our case. To prove the capability of CNN in recognition, the results are compared with the other traditional state of the art techniques Mahalanobis distance classifier (MDC), Adaboost, ANN and Deep ANN. The performance of our proposed CNN architecture for selfie sign language recognition is further

compared with [30] and [31].

III. System Architecture

We designed our multi stage CNN model by acquiring knowledge from [29,34]. The model is constructed with input layer, four convolutional layers, five rectified linear units (ReLU), two stochastic pooling layers, one dense and one SoftMax output layer. Figure 2 shows the proposed system architecture.

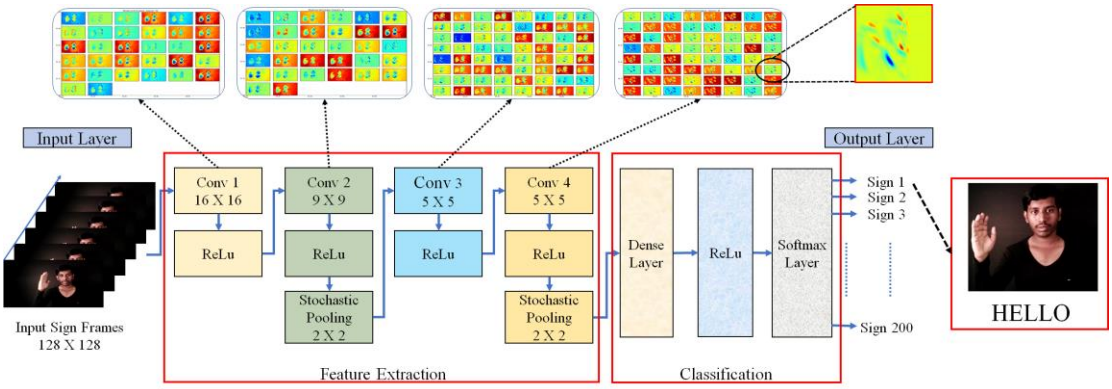


Fig.2. Proposed Deep CNN architecture

The proposed CNN architecture uses four convolutional layers with different window sizes followed by an activation function, and a rectified linear unit for non-linearities. The convolutional windows are of size 16×16, 9×9, 5×5 and 5×5 from layer 1 to 4 respectively. Three kinds of pooling strategies were tested via mean pooling, max pooling, stochastic pooling and found that stochastic pooling is suitable for our application. The feature representation is done by considering two layers of stochastic pooling. Only two layers of pooling is initiated to avoid a substantial information loss in feature representation. Classification stage is implemented with dense/fully connected layers followed by an activation functions. Softmax regression is adopted in classification.

Table 1. Layer Information and Parameters of CNN Architecture

Layer (type)	Function	Output Shape
input		$3\times640\times480$
conv_1	Convolution	$32\times128\times128$
activation_1	Activation	$32\times128\times128$
conv_2	Convolution	$32\times120\times120$
activation_2	Activation	$32\times120\times120$
stoch_pooling_1	Stochastic Pooling	$32\times60\times60$
dropout_1	Dropout	$32\times60\times60$
conv_3	Convolution	$64\times56\times56$
activation_3	Activation	$64\times56\times56$
conv_4	Convolution	$64\times52\times52$
activation_4	Activation	$64\times52\times52$
stoch_pooling_2	Stochastic Pooling	$64\times26\times26$
dropout_2	Dropout	$64\times26\times26$
flatten_1	Flatten	$43264\times1\times1$
dense_1	Fully connected	$64\times1\times1$
activation_5	Activation	$64\times1\times1$
dropout_3	Drop out	$64\times1\times1$
dense_2	Fully connected	$21\times1\times1$
activation_6	Activation	$21\times1\times1$
Output	SoftMax Regression	$21\times1\times1$

Selfie sign video frames of size 640×480 are taken as input to the system. As a first step the frames are pre-processed by resizing them to $128\times128\times3$. Resizing of an input video frames will increase the computational capability of the high-performance computing (HPC) on which the program is being implemented. The HPC used for training the CNN is a 6-node combined CPU-GPU processing machine.

Let us assume an input video frame of size $I \in R^{w\times h}$. The convolutional kernel with size K is considered for convolution with a stride of S and P padding for filling the input video frame boundary. The size of the output of convolution layer is given by

$$S_{OUT} = (I - K + 2P)/S + 1. \quad (1)$$

The architecture of our CNN model consists four convolutional layers. While the first two layers extract the low level features (like lines, corners and edges) and the last two layers learn high level features. The detailed layer information and their output sizes with parameters are tabulated in table 1.

The output of a convolutional layer is generally denoted with the following standard equation as:

$$y_j^n = f\left(\sum_{i\in c_j} y_i^{n-1} * k_{ij}^n + \zeta_j^n\right). \quad (2)$$

where n represents the n^{th} layer, k_{ij} is the convolutional kernel, ζ_j represents bias and the input maps are represented by c_j. The CNN uses a *tanh* activation function with an additive bias formulated as

$$\hbar_{ni}^{xy} = \tanh\left(\zeta_{ni} + \sum_{w=0}^{w_i-1}\sum_{h=0}^{h_j-1} W_{ij}^{wh}\hbar_{i-1}^{(x+w)(y+h)}\right). \quad (3)$$

ζ_{ni} represents feature map bias which are un supervisory trained, w_i , h_j are the kernel width and height respectively. W_{ij}^{wh} is the weight of the kernel at position (w,h) . Over a region the max value of a feature is obtained using pooling technique, which reduces the data variance. We implemented our architecture with stochastic pooling technique by calculating the probability values for each region. For every feature map c , the probability is given by

$$\chi_{w,h}^{n,k} = Stochastic_{(w,h,i,j)\in p}\left(\chi_{w,h}^{n-1,k}\ u(i,j)\right) . \quad (4)$$

where $\chi_{w,h}^{n,k}$ is the neuron activation function at a point (w,h) in spatial coordinates, and $u(i,j)$ is the weighing function of window. When compared to other pooling techniques, stochastic pooling makes CNN to converge at faster rate and improves the ability of generalization in processing invariant features.

This selfie sign language recognition is a multi-class classification problem. Hence, a SoftMax regression layer given by a hypothesis function $h_\phi(x)$ is being used as

$$h_\phi(x) = \frac{1}{1+e^{\left(-\phi^T x\right)}} . \quad (5)$$

ϕ must be trained in a way that the cost function $J(\phi)$ is to be minimized.

$$J(\phi) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{l} l\left\{y^i = j\right\}\log_p\left(y^i = \mathbb{Z} \mid x^i;\phi\right)\right] . \quad (6)$$

The classification probability in SoftMax regression layer for classifying an input x as a category $\mathbb{Z}$ is given as

$$p\left(y^i = \mathbb{Z} \mid x^i;\phi\right) = \frac{e^{\phi_j^T x^i}}{\sum_{l=1}^{k} e^{\phi_j^T x^i}} . \quad (7)$$

The network is trained to learn the features of each sign by means of a supervised learning. The internal feature representation reflects the likeness among training samples. We outline 200 signs from ISL performed by 5 native ISL users in 5 different viewing angles. The size of the total dataset is 5000 signs with each sign recording is normalized to 2 secs or 60 frames per second. All together to know the feature representation learned by the CNN system, the maximized activation neuron is extracted to recognize the sign accurately. Finally, the feature maps were visualized by averaging the image patches with stochastic response in higher layers.

IV. Results and Discussion

The main goal of this work is to correctly identify the sign from a selfie ISL dataset in order to build a bridge between the hearing impaired and normal hearing people. In [35-37] we attempted selfie sign recognition using a Mahalanobis distance classifier(MDC), traditional ANN and Deep ANN. Among these three classifiers Deep ANN out performed with massive recognition rates at lower speed. But, mobile application based implementation of selfie sign recognition demands a high-speed classifier. To improve the speed of the recognition Adaboost classifier is introduced. Even though the recognition is fast, the classification results were found to be somewhat unreliable at times. Hence this paper introduces the powerful CNN tool to classify signs at faster speed with best recognition rates.

The proposed model of CNN is applied to the selfie sign language database for classification. As the database is not available publicly, we have created the data for 200 Indian sign language words with 5 different signers considered as 5 sets in 5 various orientations. The orientations are due to variations in capture modes by different signs. The holding of the selfie stick in one hand and performing the sign with the other hand creates different orientations. Each sign image in the data set is pre-processed by reducing its dimensions to 128×128 which will improve the computational speed of CNN.

A. Batch-I: CNN training with only one set of data

Training of our proposed CNN model is done in three batches. In Batch-I of training only one set of data i.e. 200 signs performed by one ISL user in 5 user interested orientations for 2 seconds at 30 fps forming a data set with a total of 60000 images are used. The images are pre-processed and training is initiated using our proposed CNN architecture. The CNN algorithm is implemented on Python 3.6 platform using a high-performance computing(HPC) machine with 6 CPU-GPU combination.

The CNN is trained using a gradient-descent algorithm at two stages. Stage one handles the multi class classification problem with feedforward pass having S training samples from c classes. Stage two is the back-propagation pass. The error function is computed as

$$\varepsilon_e^{\ S} = \frac{1}{2}\sum_{m=1}^{S}\sum_{k=1}^{c}\left(l_k^m - v_k^m\right)^2 . \quad (8)$$

where l_k^m is the label of m^{th} pattern of k^{th} dimension and v_k^m is the corresponding value of the layer unit. The output of the convolutional layer is the *tanh* activation function of this value. The back-propagation pass is from higher to lower layers and the error in n^{th} layer is β_e^n calculated as

$$\beta_e^n = \left(w^{n+1}\right)^T \beta_e^{n+1} \odot f'\left(w^n y^{n-1} + \zeta^n\right) . \quad (9)$$

The weight in n^{th} layer is updated according to $\Delta w^n = -\lambda \frac{\partial \varepsilon}{\partial w^n}$ and $\odot$ is the convolutional operator.

During the training different feature maps were observed at different layers. Figure 3 visualizes the feature maps of one sign frame obtained in convolutional layer 1 and convolutional layer 2 with 32 filters.

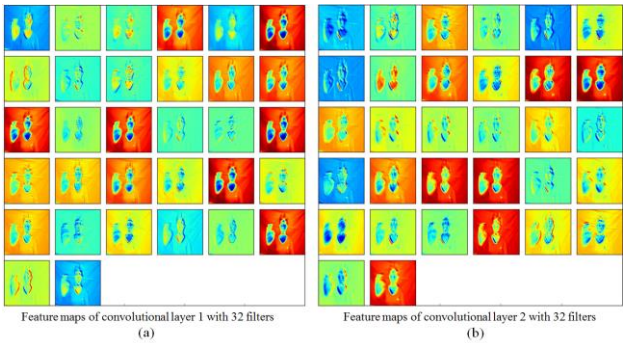


Fig.3. Feature maps (a) Outputs of convolutional layer1 (b) Outputs of convolutional layer2 with 32 filters each

Low level features like lines, edges and corners are learned from Convolutional layer 1 and 2. High level features learned from Convolutional layer 3 and 4 are visualized in figure 4.

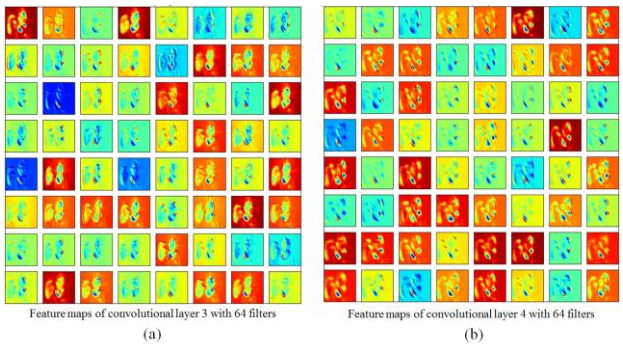


Fig.4. Feature maps (a) Outputs of convolutional layer3 (b) Outputs of convolutional layer4 with 64 filters each

A stochastic pooling which combines the advantages of both mean and max pooling techniques is implemented. It also overcomes the problem of over fitting. Increasing the number of pooling layers will increase substantial information loss. Hence, the stochastic pooling is implemented in only two layer which is achieved by calculating the probability values of each region.

In Batch-I we have used one data set for both training and testing. Testing was carried out in two cases. In case-I of testing same data set is used (i.e. training and testing was done on same data set), for case-II of testing different data set is used. In both the cases good recognition rates were obtained and are tabulated in table 2.

B. Batch-II: CNN training with two sets of data

In this case two sets of data created from two signers is used for training. For this batch data set is created with 200 Indian sign language signs of three native ISL signers in five user dependent viewing angles for 2 seconds each at 30fps. Training is performed for two sets of data on HPC machine in 100 epochs. Testing is initiated in two cases as mentioned previous section.

Table 2. Recognition Rates in Different CNN Training Cases with Two Testing Conditions

Training Batch	No. of training data sets	Training datasets	Testing data sets	Recognition rates (%)
Batch-I	1	Dataset-1	Dataset-1	91.12
			Dataset-2	82.03
Batch -II	2	Dataset-1 + Dataset-2	Dataset-1	91.89
			Dataset-3	84.15
Batch -III	3	Dataset-1 + Dataset-2 + Dataset-3	Dataset-1	92.88
			Dataset-4	87.74
			Dataset-5	88.98

Case-I of testing uses same data set which is used in training. For Case-II of testing the third data set is used. The acquired recognition rates were tabulated in table 2. Here, by increasing the number of data sets for training it is observed that a good amount of recognition is achieved compared to Batch-I training. It is also observed that the accuracy in recalling the sign is substantially increased as the number of training data sets increased. However, the training time increased by 50% than the Batch-I training process.

C. Batch-III: CNN training with three sets of data

Further improvement in recognition rates is achieved by increasing the training to CNN. A total of five data sets were created, out of which three sets were used in training and two sets for testing. An increase in recognition rates was obtained using this batch for training. Figure 5 shows the training accuracy versus validation accuracy plot for Batch-III training set. It shows that the validation accuracy is good and with less amount of over fitting.

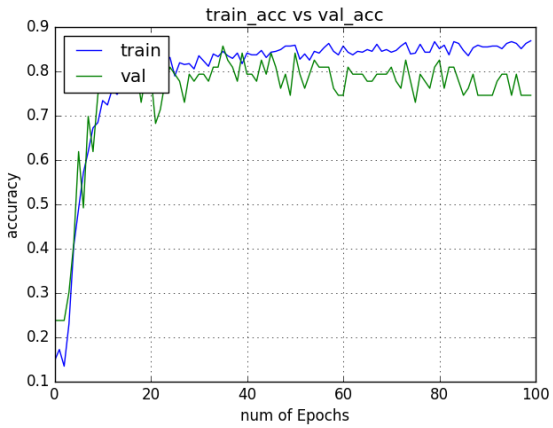


Fig.5. Training accuracy and Validation accuracy

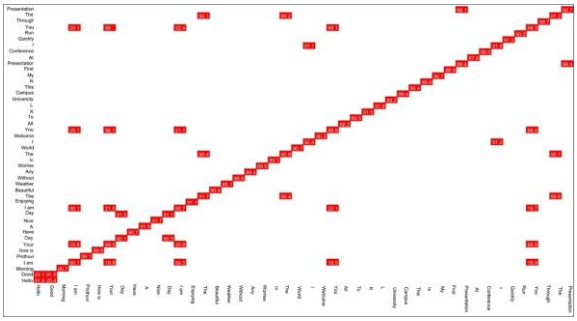
The figure 6, plots losses during training of Batch-III and there is small difference in training and validation losses with an overall less than normal loss coefficient.

An average confusion matrix is generated based on the recognition rates and number of matches for three training batches is shown in figure 7. For better visualization, it is shown for only 46 continuous ISL signs. Batch – III with 3 multiple dataset training is showing better recognition of signs compared to other two batches. However, we sacrifice training computation time for recognition. Time of real time recognition is 0.4 sec per frame and it quite fast compared to algorithms like SVM and Fuzzy classifiers.

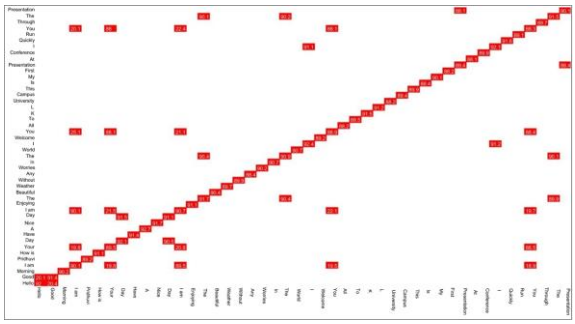
All convolutional layers are implemented with different filter windows of sizes 32×32, 16×16, 9×9 and 5×5. Reducing the filter size improves the recognition rates but increases the computational time due to the increase in number of filters. So, we used convolutional windows of sizes 16×16, 9×9, 5×5 and 5×5 for conv1, conv2, conv3 and conv4 layers respectively. Table 3 compares the performance of choosing different filtering window sizes.

A stochastic pooling adoption attained an average recognition rate of 92.88%. Implementing max pooling and mean pooling produces a recognition rate of 91.33% and 89.84% respectively.

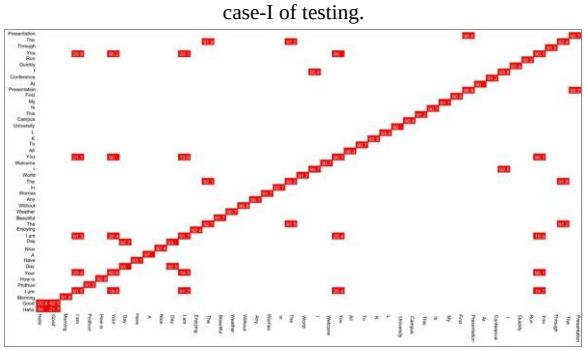
Fig.6. Training loss and Validation loss



(a) Average confusion matrix generated in Batch-I of training with case-I of testing.



(b) Average confusion matrix generated in Batch-II of training with case-I of testing.



(c) Average confusion matrix generated in Batch-III of training with case-I of testing.

Fig.7. Confusion matrices generated for 46 ISL signs

Table 3. Performance Comparision of CNN with Different Convolutional Filter Sizes

	Layers				*Recognition Rate (%)*	*Training Times (in Hrs.)*
	Conv_1	*Conv_2*	*Conv_3*	*Conv_4*		
Convolutional filter window	16×16	9×9	5×5	5×5	92.88	207
	9×9	5×5	5×5	5×5	95.54	296
	9×9	9×9	9×9	9×9	93.73	205
	16×16	16×16	16×16	16×16	90.15	168
	32×32	32×32	32×32	32×32	89.86	142

To further know the robustness and efficiency of implementation of selfie sign language recognition with CNN, it is compared with other classifiers used in our previous works. For faster recognition, we used Mahalanobis distance classifier (MDC) in [35] and ended with a very low classification rates. Further, we replaced MDC with Adaboost classifier [38] and found moderate recognition rates. In [36] we used a traditional artificial neural network (ANN) for selfie SLR recognition and found better recognition rates. We also compared the results with the methods in [30] and [31] and achieved closer classification rates.

The recognition accuracy is further improved by replacing ANN with deep ANN in [37] and reported an increase in recognition rate by 5%. A much better improvement of 4% in the recognition accuracy and an upward 15% in testing speed were observed in this work with convolutional neural networks. Even though CNN takes more time for training, the testing takes a comparatively far lesser computation times. Recognition

rates obtained with different classifiers is compared in table 4. Hence, CNN's are a suitable tool for simulating sign language recognition on mobile platforms. Testing is done on a 64 bit CPU with a 4GB ram memory in python 3.6 with OpenCV and Keras Deep learning libraries.

Table 4. Recognition Rates obtained with Different Classifiers

Classifier	*Recognition Rates (%)*					
	Batch-I Training		*Batch-II Training*		*Batch-III Training*	
	Testing with same dataset	*Testing with different dataset*	*Testing with same dataset*	*Testing with different dataset*	*Testing with same dataset*	*Testing with different dataset*
MDC [35]	57.71	52.46	58.22	54.81	59.95	55.49
Adaboost [38]	65.68	60.36	66.47	61.19	67.81	62.91
ANN [36]	75.77	66.68	76.54	68.8	78.45	73.63
Deep ANN [37]	83.98	74.89	84.75	77.01	85.74	81.84
Multiresolution CNN [30]	90.98	82.01	90.99	83.09	92.01	87.83
Two-Stream CNN[31]	91.05	81.19	91.75	82.06	91.74	86.47
Our Proposed CNN Architecture	91.12	82.03	91.89	84.15	92.88	88.98

V. Conclusion

CNN is a powerful artificial intelligence tool in pattern classification. In this paper, we proposed a CNN architecture for classifying selfie sign language gestures. The CNN architecture is designed with four convolutional layers. Each convolutional layer with different filtering window sizes is considered which improves the speed and accuracy in recognition. A stochastic pooling technique is implemented which combines the advantages of both max and mean pooling techniques. We created the selfie sign language data base of 200 ISL sign with 5 signers in 5 user dependent viewing angles for 2 secs each at 30fps generating a total of 300000 sign video frames. Training is performed in different batches to know the robustness of enormous training modes required for CNN's. In Batch-III of training, the training is performed with three sets of data (i.e. 180000 video frames) and maximizing the recognition of the SLR. Training accuracy and validation accuracies for this CNN architecture are better than the previously proposed SLR models. A less amount of training and validation loss is observed with the proposed CNN architecture. The average recognition rate of proposed CNN model is 92.88 % and is higher compared with the other state of the art classifiers.

References

[1] Parton, Becky Sue. "Sign language recognition and translation: A multidisciplined approach from the field of artificial intelligence." Journal of deaf studies and deaf education, winter:11, no.1, 2006, pp:94-101. doi:10.1093/deafed/enj003.

[2] Mitra, Sushmita, and Tinku Acharya. "Gesture recognition: A survey." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37, no.3, 2007, pp: 311-324. doi: 10.1109/TSMCC.2007.893280.

[3] Raffa, Giuseppe, Lama Nachman, and Jinwon Lee. "Efficient gesture processing." U.S. Patent 9,535,506, issued January 3, 2017.

[4] Liu, Zhengzhe, Fuyang Huang, Gladys Wai Lan Tang, Felix Yim Binh Sze, Jing Qin, et al. "Real-time Sign Language Recognition with Guided Deep Convolutional Neural Networks." In Proceedings of the 2016 Symposium on Spatial User Interaction, pp. 187-187. ACM, 2016. doi:10.1145/2983310.2989187.

[5] Chen, Feng-Sheng, Chih-Ming Fu, and Chung-Lin Huang. "Hand gesture recognition using a real-time tracking method and hidden Markov models." Image and vision computing 21, no.8, 2003,pp: 745-758. doi: 10.1016/S0262-8856(03)00070-2.

[6] Cavender, Anna, Rahul Vanam, Dane K. Barney, Richard E. Ladner, and Eve A. Riskin. "MobileASL: Intelligibility of sign language video over mobile phones." Disability and Rehabilitation: Assistive Technology 3, no. 1-2 , 2008 pp: 93-105. doi: 10.1080/17483100701343475.

[7] Starner, Thad, Joshua Weaver, and Alex Pentland. "Real-time american sign language recognition using desk and wearable computer based video." IEEE Transactions on Pattern Analysis and Machine Intelligence 20, no. 12, 1998, pp:1371-1375. doi: 10.1109/34.735811.

[8] Kushwah, Mukul Singh, Manish Sharma, Kunal Jain, and Anish Chopra. "Sign Language Interpretation Using Pseudo Glove." In Proceeding of International Conference on Intelligent Communication, Control and Devices, pp. 9-18. Springer Singapore, 2017.

[9] Kumar, Pradeep, Himaanshu Gauba, Partha Pratim Roy, and Debi Prosad Dogra. "Coupled HMM-based Multi-Sensor Data Fusion for Sign Language Recognition." Pattern Recognition Letters, Vol. 86, pp.1-8, 2017. doi: 10.1016/j.patrec.2016.12.004

[10] Bhuyan, M. K., D. Ghoah, and P. K. Bora. "A framework for hand gesture recognition with applications to sign language." In India Conference, 2006 Annual IEEE, pp. 1-6. IEEE, 2006. doi: 10.1109/INDCON.2006.302823.

[11] Yu Zhou and Xilin Chen, "Adaptive sign language recognition with Exemplar extraction and MAP/IVFS", IEEE signal processing letters, Vol 17, No-3, March 2010, pp297-300. doi: 10.1109/LSP.2009.2038251.

[12] Och, J., Ney, H., "A systematic comparison of various alignment models". Computational Linguistics 29 (1), pp.19–51, 2003. doi: 10.1162/089120103321337421

[13] Koehn, Philipp. "Pharaoh: a beam search decoder for phrase-based statistical machine translation models." In Conference of the Association for Machine Translation in the Americas, pp. 115-124. Springer, Berlin, Heidelberg, 2004.

[14] Kishore PVV, Rajesh Kumar P. "A video based Indian Sign Language Recognition System (INSLR) using wavelet transform and fuzzy logic". International Journal of Engineering and Technology. 4(5), pp.537-42, 2012. doi: 10.7763/IJET.2012.V4.427.

[15] Inthiyaz Syed, B.T.P.Madhav, and P.V.V.Kishore. "Flower segmentation with level sets evolution controlled by colour, texture and shape features." Cogent Engineering 4, no.1(2017):1323572.doi:10.1080/23311916.2017.1323572.

[16] Shimada, Mitsuaki, Satoshi Iwasaki, and Toshiyuki Asakura. "Finger spelling recognition using neural network with pattern recognition model." In SICE 2003 Annual Conference, vol. 3, pp. 2458-2463. IEEE, 2003.

[17] Rätsch, Gunnar, Takashi Onoda, and K-R. Müller. "Soft margins for AdaBoost." Machine learning, vol.42, no.3, pp.287-320, 2001. doi: 10.1023/A:1007618119488.

[18] Z. Dong, X. Tian, "Multi-level photo quality assessment with multi-view features", Neurocomputing. Vol.168, pp.308-319, 2015. doi: 10.1016/j.neucom.2015.05.095.

[19] Z. Dong, X. Shen, H. Li, X. Tian, "Photo quality assessment with DCNN that understands image well", In proceedings of the International Conference on MultiMedia Modeling (MMM), 2015, pp.524-535.

[20] X. Lu, Z. Lin, H. Jin, J. Yang, J. Wang, "Rating pictorial aesthetics using deep learning", In proceedings of the ACM Conference on Multimedia, 2014, 457-466.

[21] A. Krizhevsky, I.Sutskever, G.E. Hinton, "ImageNet classification with deep convolution neural networks", In proceedings of the Annual Conference on Neural Information Processing System (NIPS), 2012, pp.1097-1105.

[22] Y. Sun, X. Wang, X. Tang, "Deep learning face representation from predicting 10,000 classes", In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1891-1898.

[23] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, "What is the best multi-stage architecture for object recognition", In proceedings of the IEEE International Conference on Computer Vision (ICCV), 2009, pp. 2146-2153. doi: 10.1109/ICCV.2009.5459469.

[24] H. Lee, R. Grosse, R. Ranganath, A.Y.Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", In proceedings of the International Conference on Machine Learning (ICML), 2009, pp. 609-616. doi: 10.1145/1553374.1553453.

[25] Y. Bengio, "Learning deep architectures for AI, Foundations and trends in Machine Learning", Vol. 2, No. 1, pp. 1-127, 2009. doi: 10.1561/2200000006.

[26] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", In proceedings of the IEEE , Vol. 86, No. 11, pp. 2278-2324, 1998. doi: 10.1109/5.726791.

[27] H. Lee, A. Battle, R. Raina and A. Y. Ng, "Efficient sparse coding algorithms", In Advances in neural information processing systems, pp. 801-808, 2006.

[28] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann Machines", In proceedings of the International Conference on Artificial Intelligence and Statistics, Clearwater Beach, Florida USA, pp. 448-455, 2009.

[29] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, vol. 521, No. 7553, pp. 436-444, 2015. doi: 10.1038/nature14539.

[30] Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725-1732. 2014. doi: 10.1109/CVPR.2014.223.

[31] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." In Advances in neural information processing systems, pp. 568-576. 2014.

[32] H. Lee, R. Grosse, R. Ranganath, A.Y.Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations", In proceedings of the International Conference on Machine Learning (ICML), 2009, pp. 609-616. doi: 10.1145/1553374.1553453.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "ImageNet: a large-scale hierarchical image dataset", In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR) , 2009, pp. 248-255. doi: 10.1109/CVPR.2009.5206848.

[34] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", In Advances in Neural Information Processing Systems(NIPS), Lake Tahoe, Nevada, USA pp. 1097-1105, 2012.

[35] Rao, G. Anantha, and P. V. V. Kishore. "Sign language recognition system simulated for video captured with smart phone front camera." International Journal of Electrical and Computer Engineering 6.5 (2016): 2176. doi: 10.11591/ijece.v6i5.11384

[36] Rao, G. Anantha, P. V. V. Kishore, D. Anil Kumar, and A. S. C. S. Sastry. "Neural network classifier for continuous sign language recognition with selfie video." Far East Journal of Electronics and Communications 17.1: 49,2017.

[37] Rao, G. Anantha, and P. V. V. Kishore. "Selfie video based continuous Indian sign language recognition system." Ain Shams Engineering Journal (2017). doi: 10.1016/j.asej.2016.10.013

[38] K. V. V. Kumar, P. V. V. Kishore, and D. Anil Kumar, "Indian Classical Dance Classification with Adaboost Multiclass Classifier on Multifeature Fusion," Mathematical Problems in Engineering, vol. 2017, Article ID 6204742, 18 pages, 2017. doi: 10.1155/2017/6204742

Authors' Profiles

P.V.V.Kishore is having Ph.D degree in electronics and communications Engineering from Andhra University College of engineering in 2013. He received M.Tech from Cochin University of science and technology in the year 2003. He received B.Tech degree in electronics and communications engineering from JNTU, Hyd. in 2000. He is currently full professor and Image, signal and speech processing Head at K.L.University. His research interests are digital signal and image processing, Artificial Intelligence and human object interactions.

G. Anantha Rao received B.Tech Degree from, GMRIT, JNTU, Hyderabad, In 2007. M.Tech. Degree from STIET, JNTUK, Kakinada, India In 2011, Pursuing Ph.D. In The Department of Electronics and Communication Engineering, KL University, Vijayawada, India. His research interest includes on Signal Processing, Image and Video Processing.

E.Kiran Kumar received the B.Tech degree in Electronics and Communication Engineering from the JNT University, Kakinada, India, in 2009, M.Tech degree in Systems and Signal Processing branch from JNT University, Kakinada, India, in 2013, specializing in evolving optimized object segmentation and recognition, and pursuing the Ph.D. degree from KL University, India. He is currently a Junior Research Fellow at the KL University, India. His research interests include the analysis of musculoskeletal movements of hand and movement strategies of the wrist and fingers in Indian sign language recognition.

M. Teja Kiran Kumar received the B.Tech degree in Electronics and Communication Engineering from the Vignan's Institute of Information Technology affiliated to JNT University, Kakinada, India, in 2013, M.Tech degree in Communication Engineering and Signal Processing from Nagarjuna University, Guntur, India, in 2015 and pursuing the Ph.D. degree from KL University, India. He is currently a Research Scholar at the KL University, India. His research interests include the Deep learning, Motion Recognition and Bio-mechanical analysis.

Anil Kumar Dande received M.Tech degree from KL University, Vijayawada In 2016. B.Tech degree from GEC, JNTUK, Kakinada, India in 2014. Currently he is Pursuing Ph.D in the Department of Electronics and Communication Engineering, KL University, vijayawada, India. His research interests are Signal processing, Image and 3D-Video Processing. He is currently a member of SIEEE. He has published 15 research papers in Various National and International journals and conferences.

How to cite this paper: P.V.V. Kishore, G. Anantha Rao, E. Kiran Kumar, M. Teja Kiran Kumar, D. Anil Kumar, "Selfie Sign Language Recognition with Convolutional Neural Networks", International Journal of Intelligent Systems and Applications(IJISA), Vol.10, No.10, pp.63-71, 2018. DOI: 10.5815/ijisa.2018.10.07

