Deep Convolutional Neural Networks for Sign Language Recognition

G.Anantha Rao[1], K.Syamala[2] , P.V.V.Kishore[1], A.S.C.S.Sastry[1]
[1] Biomechanics and Vision Computing Research Center, Department of ECE,
K.L. University, Green Fields, Vaddeswaram, Guntur (DT), Andhra Pradesh, INDIA.
[2] Department of ECE, Avanthi Institute of Engineering and Technology, INDIA.
ananth.gondu @gmail.com, syamala.kanchimani@gmail.com, pvvkishore@kluniversity.in, ascssastry@kluniversity.in,

***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 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.**

Keywords*— *Selfie sign language, Convolutional Neural Networks (CNN), Stochastic pooling, Sign recognition, 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. 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 [2].

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 [3] 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.

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 dependant 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.

Andrew Ng , Hinton, LeCun , Bengio et al. have performed fundamental research on CNNs to achieve improved performance of CNN algorithms and structural optimization [4]. Yann LeCun et al. in [5], highlighted that deep CNN is a breakthrough in image, video, audio and speech processing. 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.

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.

II. SYSTEM ARCHITECTURE

We designed our multi stage CNN model by acquiring knowledge from [6]. 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 1 shows the proposed system architecture.

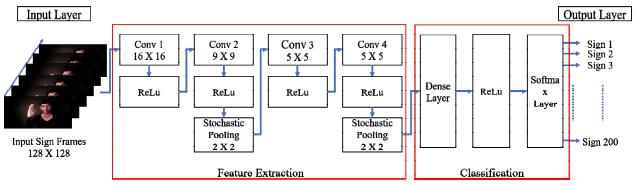

Fig.1. Proposed Deep CNN architecture

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 $16\times16, 9\times9, 5\times5$ 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.

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

$$S_{OUT} = \left(I - K + 2P\right)/S + 1 \tag{1}$$

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

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

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

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

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

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

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

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

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

$$J(\phi) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{l} l\left\{y^i = j\right\}\log_p\left(y^i = \mathrm{Z}\,|\,x^i;\phi\right)\right] \tag{6}$$

The classification probability in SoftMax regression layer for classifying an input x as a category Z is given as

$$p\left(y^i = \mathrm{Z}\,|\,x^i;\phi\right) = \frac{e^{\phi_j^T x^i}}{\sum_{l=1}^{k} e^{\phi_j^T x^i}} \tag{7}$$

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.

III. Results and Discussion

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.

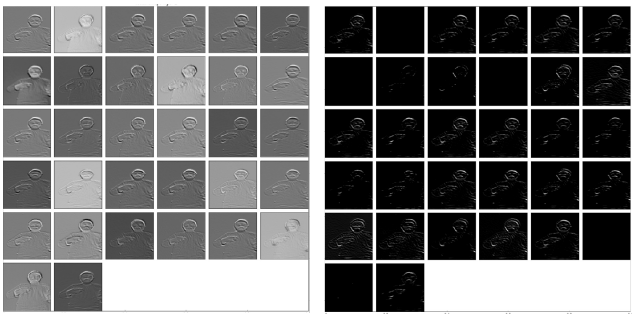

Fig.2. Feature maps (a) Outputs of convolutional layer1 (b) Outputs of convolutional layer2 with 32 filters each.

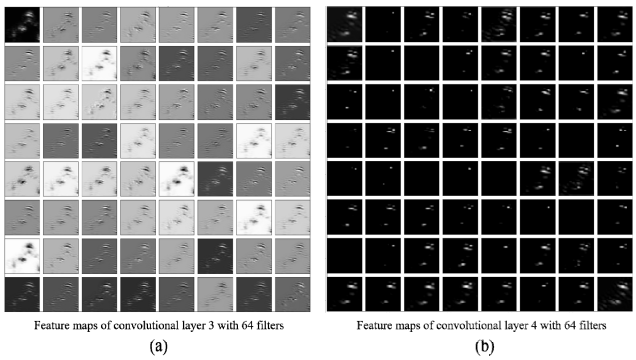

Fig.3. Feature maps (a) Outputs of convolutional layer3 (b) Outputs of convolutional layer4 with 64 filters each.

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

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 3.

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.

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 dependant 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. Case-I of testing uses same data set which is used in training. For Case-II of testing the third data set is used. 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 4 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.

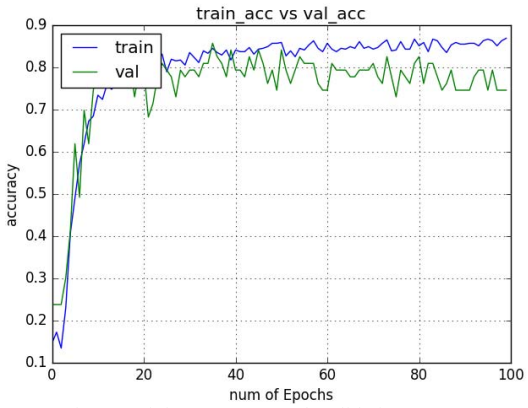

Fig.4. Training accuracy and Validation accuracy

An average confusion matrix is generated based on the recognition rates and number of matches for three training batches is shown in figure 5. For better visualization, it is shown for only 46 continuous ISL signs. All convolutional layers are implemented with different filter windows of sizes

$32\times32, 16\times16, 9\times9$ 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 $16\times16, 9\times9, 5\times5$ and 5×5 for conv1, conv2, conv3 and conv4 layers respectively.

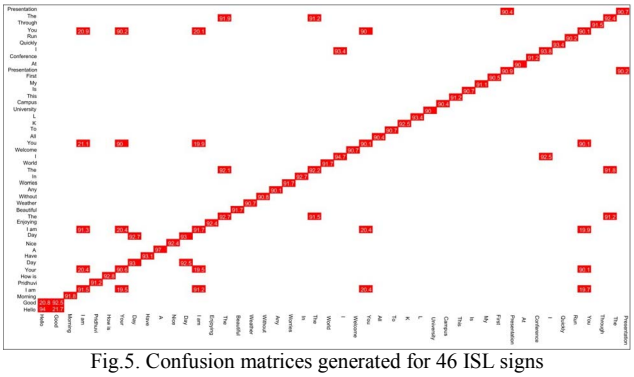

Fig.5. Confusion matrices generated for 46 ISL signs

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. 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 [7] and ended with a very low classification rates. Further, we replaced MDC with Adaboost classifier and found moderate recognition rates. In [8] we used a traditional artificial neural network (ANN) for selfie SLR recognition and found better recognition rates.

Table.1. Recognition rates 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 [7]	57.71	52.46	58.22	54.81	59.95	55.49
Adaboost [9]	65.68	60.36	66.47	61.19	67.81	62.91
ANN [8]	75.77	66.68	76.54	68.8	78.45	73.63
Deep ANN [10]	83.98	74.89	84.75	77.01	85.74	81.84
Our Proposed CNN Architecture	91.12	82.03	91.89	84.15	92.88	88.98

The recognition accuracy is further improved by replacing ANN with deep ANN in [10] 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 1. 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.

IV. 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 dependant 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 higer compared with the other state of the art classifiers.

References

[1] Becky Sue Parton. “Sign language recognition and translation: A multidisciplined approach from the field of artificial intelligence”. Journal of deaf studies and deaf education, 11(1):94–101, 2005.

[2] Zhengzhe Liu, Fuyang Huang, Gladys Wai Lan Tang, Felix Yim Binh Sze, Jing Qin, Xiaogang Wang, and Qiang Xu. “Real-time sign language recognition with guided deep convolutional neural networks”. In Proceedings of the 2016 Symposium on Spatial User Interaction, pages 187–187. ACM, 2016.

[3] Mukul Singh Kushwah, Manish Sharma, Kunal Jain, and Anish Chopra. “Sign language interpretation using pseudo glove”. In Proceeding of International Conference on Intelligent Communication, Control and Devices, pages 9–18. Springer, 2017.

[4] Yoshua Bengio et al. “Learning deep architectures for ai. Foundations and trends® in Machine Learning”, 2(1):1–127, 2009.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning. Nature”, 521(7553):436–444, 2015.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In Advances in neural information processing systems, pages 1097–1105, 2012.

[7] G Ananth Rao and PVV Kishore. “Sign language recognition system simulated for video captured with smart phone front camera”. International Journal of Electrical and Computer Engineering, 6(5):2176, 2016.

[8] G Anantha Rao, PVV Kishore, D Anil Kumar, and ASCS Sastry. “Neural network classifier for continuous sign language recognition with selfie video”. Far East Journal of Electronics and Communications, 17(1):49, 2017.

[9] KVV Kumar, PVV Kishore, and D Anil Kumar. “Indian classical dance classification with adaboost multiclass classifier on multifeature fusion”. Mathematical Problems in Engineering, 2017, 2017.

[10] G Ananth Rao and PVV Kishore. “Selfie video based continuous indian sign language recognition system”. Ain Shams Engineering Journal, 2017.