Indian Sign Language Gesture Recognition using Image Processing and Deep Learning

Neel Kamal Bhagat
Department of Electrical Engineering
Indian Institute of Science
Bengaluru, Karnataka
neelbhagat@iisc.ac.in

Vishnusai Y
Department of E&C
R V College of Engineering
Bengaluru, Karnataka
vishnusai.ec15@rvce.edu.in

Rathna G N
Department of Electrical Engineering
Indian Institute of Science
Bengaluru, Karnataka
rathna@iisc.ac.in

***Abstract*—Speech impaired people use hand based gestures to communicate. Unfortunately, the vast majority of the people are not aware of the semantics of these gestures. In a attempt to bridge the same, we propose a real time hand gesture recognition system based on the data captured by the Microsoft Kinect RGB-D camera. Given that there is no one to one mapping between the pixels of the depth and the RGB camera, we used computer vision techniques like 3D contruction and affine transformation. After achieving one to one mapping, segmentation of the hand gestures was done from the background noise. Convolutional Neural Networks (CNNs) were utilised for training 36 static gestures relating to Indian Sign Language (ISL) alphabets and numbers. The model achieved an accuracy of 98.81% on training using 45,000 RGB images and 45,000 depth images. Further Convolutional LSTMs were used for training 10 ISL dynamic word gestures and an accuracy of 99.08% was obtained by training 1080 videos. The model showed accurate real time performance on prediction of ISL static gestures, leaving a scope for further research on sentence formation through gestures. The model also showed competitive adaptability to American Sign Language (ASL) gestures when the ISL models weights were transfer learned to ASL and it resulted in giving 97.71% accuracy.**

***Index Terms*—Gesture Recognition, Indian Sign Language, Convolutional Neural Networks, LSTMs, Microsoft Kinect.**

I. Introduction

Indian Sign Language system contains standard hand-based gestures which are used by speech impaired people for communication purposes in India. Given the complexity and the extensivity of these signs, the knowledge of these gestures is not known to many people, thus hampering communication between the speech impaired and the non-speech impaired people. With the recent surge in deep learning a lot of active applied research in the field of computer vision [1] is being carried out. But despite all this, research carried out in recognition of ISL gestures has been limited and insufficient. With a motivation to improve the same, our paper focuses on setting a benchmark for recognition of ISL gestures as well to develop a model to aid the communication of the speech impaired.

Recognition of gestures in ISL is a challenging task. ISL uses both the hands for portraying a gesture as opposed to ASL (alphabets), which uses only one hand. This increases complexity while applying feature extractors like Hough Transform [2] and Scale Invariant Feature Transform (SIFT) [3]. Also while trying to predict gestures in real time, the problem of background complexity occurs which might inhibit accurate prediction of the gestures. So, it becomes essential to segment the hand gesture region from the background. Though there are techniques like segmentation using colour spaces and Otsus technique [4], they all have their limitations with respect to the background conditions. So, we used depth based segmentation by taking data from the Microsoft Kinect RGB-D camera. Using the information from the depth channel of the Kinect, the hand gesture region can easily be segmented from the background. But here, there is a problem of lack of one-to-one mapping between the pixels of the depth channels and the RGB channels. This prohibits segmentation of hand region in the RGB frame by direct overlap, which is crucial for feature extraction process. The problem is solved and explained in the later section of this paper. This technique of using the Kinect enables segmentation of the hand region from any background conditions.

As discussed earlier, applying handcrafted feature techniques to extract features from a gesture will not generalise well on all gestures. For example, Fig. 1 demonstrates Gradient Hough Transform (GHT) [5] applied to different orientations of a hand. In Fig. 1(a), the circles computed along the edges of the hand are based on parameters which suit this particular pose. Fig. 1(b) shows the circles computed by using the same parameters on a different pose of the hand. Clearly, all the computed circles are outside the region of interest and would prove to be of little use while extracting features.

Similarly, the application of any other handcrafted feature technique would have its limitations with respect to generalising to all different postures and flexural angles of the hand gestures. Application of learning based techniques like CNNs to image processing techniques have proven to provide excellent results and are setting new benchmarks. Thus, CNNs were chosen as the primary deep learning architecture for the task of recognition. A multi-layered CNN based architecture was used to train the images of static gestures from ISL. Around 90,000 depth and RGB images were generated using the Kinect. For dynamic based hand gestures, LSTMs with a convolutional kernel were chosen for training. Totally around 1,080 videos were generated for the purpose of training. The results obtained for the two methods are showcased later in

978-1-7281-3857-2/19/$31.00 ©2019 IEEE

Fig. 1. (a) Successful application of GHT to a posture of a hand by estimating parameters and (b) Unsuccessful application of GHT to a different view of the hand applying the same parameters.

the paper.

The rest of the paper is organised as follows. Section II deals with the previous related work. Section III explains the technique to map between depth and RGB pixels. Section IV explains about the datasets used and techniques applied to achieve generalisation. Section V explains the different architectures for training the two datasets. Section VI shows the results obtained and Section VII concludes the paper.

II. Related Work

Up until now, a lot of research work has been dedicated to recognition of hand gestures, though there has been no good research work with respect to ISL. Singha et al. [6] created a database of 240 images for 24 signs of ISL. They extracted eigen values from the images and classified depending upon the Euclidean distance. Their model achieved a classification accuracy of 97% , but the images were limited to a set of constant background conditions and the gestures were of static type. Pei Xu et al. [7] designed a real time Human-Computer interaction system based on hand gestures, where each hand image was preprocessed by hand color filtering, Gaussian blurring, morphological transformations, etc. After that, a CNN was trained used to identify the gestures and an Kalman estimator was used to move the mouse pointer in response to the gestures identified. This system achieved an average accuracy of 99.8% on 16 gestures. Liao et al. [8] used an Intel Real Sense RGB-Depth sensor, where depth perception techniques were used to segment the hand region from the background. They also use Generalized Hough transform to detect and segment the hand in RGB images. The segmented depth and RGB images were trained using a dual channel convolutional neural network. The proposed technique achieves a classification accuracy of 99.4% while classifying 24 gestures from ASL. Also, the results achieved were with respect to only the images taken for training. The paper did not specify the performance of the model in real time scenario. Otiniano et al. [9] used gesture recognition in sign language and finger spelling. First, the hand region was segmented from the background using depth map and accurate hand shapes were obtained using depth data and color data. Gradient kernel descriptor was used as a feature extractor in depth images and SIFT was used as a feature extractor in RGB images. The combined data were given as input to an SVM and a classification accuracy of 90.2% was obtained on the ASL database. Molchanov et al. [10] used a Recurrent 3D Convolutional Neural Network (R3DCNN) with temporal classification to identify dynamic gestures in real time. The data was collected using depth, color and stereo-IR sensors. The 3D CNN was pre-trained with the large-scale Sport- 1M [11] human action recognition dataset. Then the entire model was trained on the R3DCNN which also employed a technique called as Connectionist temporal classification. The model achieved a classification rate of 83.8%. Okan et al. [12] used a hierarchical CNN based architecture for gesture recognition. The model uses a detector, which is a light weight based CNN for detection of hand gestures and a heavy weight 3DCNN for gesture prediction. The model achieves a classification rate of 94.04% and 83.82% accuracy on the EgoGesture and NVIDIA benchmarks. Xiaokai et al. [13] proposes a novel technique based on neural network called as Dense Image network (DIN) which encodes the video containing the hand gestures to a compact form which distils its spatio-temporal evolution. The DIN is fed to a CNN, where features of the video are learnt in a more efficient manner consuming less time and space. The method achieved benchmark results on action and gesture recognition. Kopuklu et al. [14] uses a technique which uses optical flow data and RGB modalities as input to a deep neural network. Each RGB image has its corresponding optical flow images, through which the movement of hand can be traced. A neural network trains on these images and a classification accuracy of 84.7% was obtained on the NVIDIA benchmark, while achieving 96.28% and 57.4% on the Jester and ChaLearn benchmarks. Mukesh [15], used ASL database (alphabets and numbers) and trained on CNNs and achieved a training accuracy of 85.51%. Then a custom made ISL database was generated and using the same network of CNN achieved 85.51% training accuracy. The models performance again was not tested in real time.

As per our knowledge this is the first time a model has been developed to recognize gestures of ISL in real time. Our model performs well on all background conditions and hands of any size.

III. Mapping Depth and RGB Pixels

As discussed earlier, there is no one to one mapping between the pixels of RGB and Depth. We used computer vision techniques to achieve the same. Firstly, the 3D model of the entire scene in the field of view of the depth camera is computed. Since the kinect gave out raw values of depth between 0 and 2047, triangulation was applied to calculate the value of depth in mm. The equation for the same is given by equation 1.

$$z_{world}^{-1} = \left(\frac{m}{f \cdot b}\right) \cdot d' + \left(Z_o^{-1} + \frac{n}{f \cdot b}\right) \tag{1}$$

where z_{world} is the distance between the Kinect and the real world point in mm, d is the normalized disparity value

by normalizing the raw disparity value d between 0 and 2047, thus $d = m \cdot d + n$, m and n are the de-normalization parameters. b and f are the base length and the focal length of the depth camera, z_o is the distance between the Kinect and the predefined reference pattern.

After obtaining the real world location of each pixel from the depth camera, the entire 3D world model can be estimated from the image locations (x, y) by equations 2 and 3.

$$x_{world} = -Z_{world} \cdot \big(x - x_o + \delta x\big) \tag{2}$$

$$y_{world} = -Z_{world} \cdot \big(y - y_o + \delta y\big) \tag{3}$$

where x_{world} and y_{world} includes the coordinates of the point in 3D space, (x_0, y_0) is the principal location of the depth image and δx and δy are the corrections due to lens distortions.

From the obtained 3D model, Affine transformation is applied to convert the model to RGB camera point of view. The equation for the same is given by equation 4.

$$\begin{bmatrix} x'_{world} \\ y'_{world} \\ z_{world} \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix} \tag{4}$$

where R and T are the rotation and translation parameters, and the LHS of equation 4 indicates the 3D coordinates with respect to the RGB camera point of view.

From the deduced 3D co-ordinates, the true location of the points in the RGB plane can be obtained by equation 5.

$$\begin{bmatrix} x_{RGB} \\ y_{RGB} \\ 1 \end{bmatrix} = \frac{f_{RGB}}{z'_{world}} \cdot \begin{bmatrix} x'_{world} \\ y'_{world} \\ z'_{world} \end{bmatrix} \tag{5}$$

where f_{RGB} is the focal length of the RGB camera.This technique effectively registers the pixels the depth and RGB camera. Fig. 2(a) shows an example of gesture segmentation in RGB plane without registration and Fig. 2(b) shows the result of applying the registration techniques mentioned above.

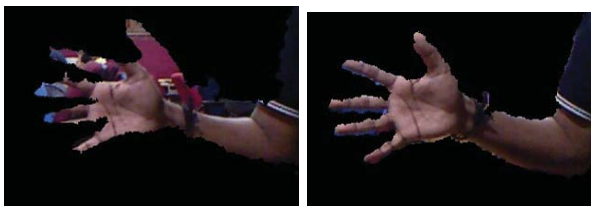

Fig. 2. (a) Segmentation of gesture in RGB without registration (b) Segmentation of gesture in RGB after segmentation

The above example proves to be effective for gesture segmentation in both the RGB and depth planes.

IV. Dataset

A. Details of the dataset

Since there is no standard dataset available on ISL, we created our own dataset for training, as well as for testing purposes. The dataset was divided into two parts, one for static based gestures and the other for dynamic based gestures. As stated in the previous section, after registration of depth and RGB pixels, hand based gestures can be segmented in the RGB plane. This in turn reduces the complexity while trying to maximise the performance of the model irrespective of the background condition. The dataset were captured from five subject matters, with an age average of 25, inclusive of both male and female genders. For the static based gestures i.e. for dataset containing gestures of alphabets and numbers, our dataset contained 45,000 images based out of the depth camera and 45,000 images based out of the RGB camera. Apart from this, we also captured and stored another 45,000 images without background segmentation. The dataset were captured in our lab with two different lighting conditions, in order to generalize the parameters of the training algorithm while predicting in real-time.

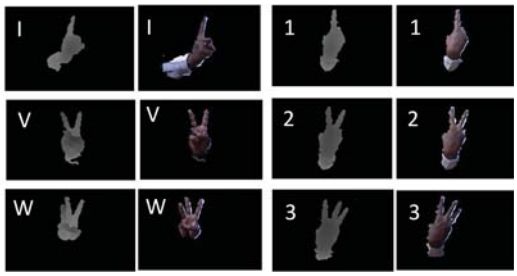

Fig. 3. Examples of depth and RGB pairs for alphabets I, V, W and numbers 1, 2, 3

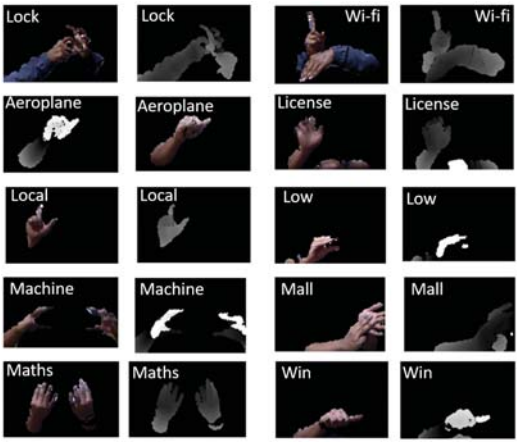

Fig. 4. Database for Dynamic words

For the dynamic gestures, we captured videos pertaining to 10 commonly used words of ISL. The words chosen were Wifi, Lock, Maths, Mall, Low, Win, Machine, Local, License and Aeroplane. A total of 1,080 videos were captured at 9 different frame rates. The standard frame rate of the Kinect is 30 fps. But we captured the videos at frame rates of 27, 24, 21, 18, 15, 33, 36 and 39 fps. This was introduced in order to have temporal variability between the same gestures, allowing the training algorithm to extract additional temporal features.

Fig. 3 show examples of the segmented RGB-D pairs for

Fig. 5. Database for Static gestures

alphabets I V, W and numbers 1,2 and 3 captured during data acquisition. It is to be noted that the pairs 1 and I, 2 and V, 3 and W look quite similar, except for the fact that the postures are inverted. Section VI shows the results where the model accurately distinguishes between the pairs of signs. Fig. 4 shows the RGB-Depth single frame database image for dynamic ISL words. Fig. 5 showcases the entire database for static gestures in RGB plane.

B. Techniques to attain Generalization

CNN's are not scale, rotational and translation invariant. The dataset captured for the static gestures contains gestures which are concentrated to a limited area in the frame of the image. Though this might not affect the training performance, it might affect real-time performance because the CNN requires the gesture to be portrayed to the camera at the same limited area concentrated during data acquisition. This is not desirable in real-time, because the CNN needs to accurately predict regardless of the location of the gesture in the frame. So to achieve generalization and to boost real-time performance, we used artificial data synthesis to generate more data.

Fig. 6 shows the result of application of label preserving transformations to the original image containing a dog to represent it in a slightly different orientation. Application of these images to the CNN will enable it to learn more features, making it robust and invariant to scaling, rotation and translation.

V. Methodology

This section discusses about the methodology and the architectures adopted for training.

Fig. 6. **Left:** Example of original image of a dog. **Right:** Various images obtained after applying artificial data synthesis

A. Training architectures for static based gestures

Liao et al. [8] used double channel convolutional neural network based architecture to simultaneously train segmented depth and RGB images. We also adopted a similar method to train our RGB-D pairs simultaneously. We also trained depth and RGB images separately on a identical CNN based architecture shown in Fig. 7 and observed its performance both offline and real time.

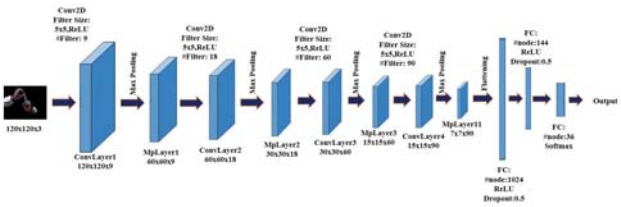

Fig. 7. CNN architecture for training RGB and depth images separately

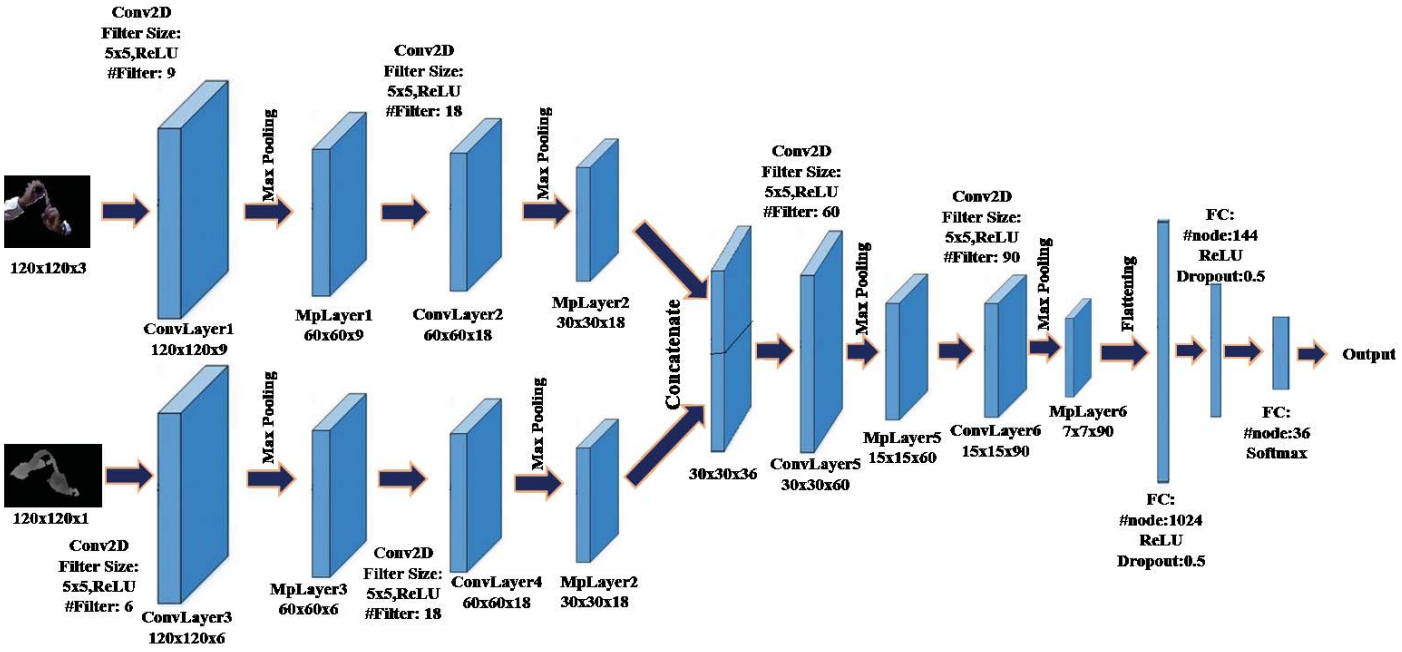

Fig. 8. Architecture for training RGB-D images simultaneously

The images are re-sized into a size of 120x120 by maintaining the aspect ratio. They are passed through a series of convolutional layers, ReLu layers and max-pooling layers where they are reduced to 90 feature maps of size 7x7. Then they are passed onto two fully-connected layers with 1024 and 144 neurons respectively. Finally, the last layer uses softmax activation to classify the gestures.

We also wanted to test our model for its performance with respect to ASL. Using datasets from [16], we used transfer learning from the the model trained with only RGB images. The weights from the convolutional layers were retained and the weights from the dense layer were re-initialized to random numbers. The results for the same is explained in the next section.

Fig. 8 shows the CNN based architecture used to train the RGB and depth images simultaneously. Here the images after re-sizing to sizes 120x120 maintaining the aspect ratios are sent into two channels separately, where they are passed through a series of convolutional layers, Relu and max-pooling layers to produce 30 feature maps of size 18x18. Then the two channels are fused into 2 channels. Then they are again passed through 5 convolutional layers where they are reduced to 90 feature maps of size 7x7. Then they are passed through two fully-connected layers containing 1024 and 144 neurons respectively. The last layer uses softmax activation for classification.

B. Training architectures for dynamic based gestures

As discussed in the previous section, the dataset for dynamic gestures were captured at 9 different frame rates. Amin Ullah et al. [17] states that downsampling a video by 5 or 6 times will lead to removal of redundant frames without destroying the temporal information. So all the videos were down-sampled by considering every 6th frame and removing the rest. Also, the maximum number of frames was set at 20 and any video sequence having number of frames less than 20 were zero padded in the beginning.

LSTM's with a convolutional kernel was chosen for training the videos over a 3D-CNN architecture, because of the computational heaviness of the 3D CNN's. We used separate architectures for training the depth and RGB based videos separately and also one dual channel architecture was developed to train them simultaneously.

Fig. 9 shows the Convolutional LSTM based architecture to train the depth and RGB videos separately. There are two convolutional LSTM layers with kernel sizes of (5,5) with number of filters being 32 and 54 respectively. The strides was set as 2. The output from these two layers is then passed into a fully-connected layer with 100 nodes and RelU activation. The final softmax layer classifies the videos into one of the 10 classes.

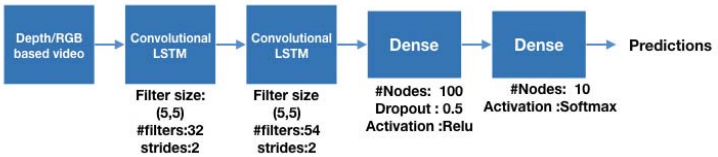

Fig. 9. Single channel architecture for video training

The architecture for the dual channel LSTM is similar to Fig. 9, except the depth and the RGB videos are sent through

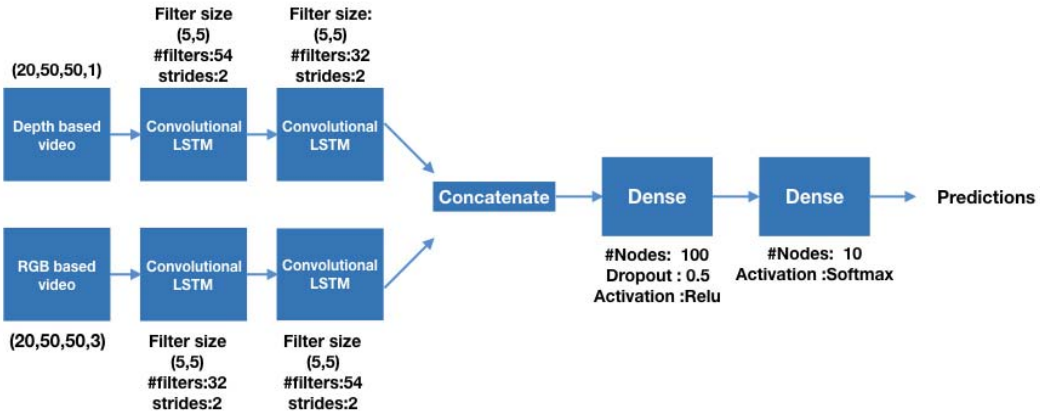

Fig. 10. Dual channel architecture for video training

two different channels of Convolutional LSTM's simultaneously before fusing the layers. Fig. 10 shows the architecture adopted for the same.

VI. Training and Results

This section discusses the results obtained by training the models discussed in the previous section. All the models were trained on NVIDIA Tesla K80 GPU's. The dataset was divided in the ratio 70:30 between the training and the testing set, after shuffling the data captured from all five subject matters.

The hyper-parameters for the training were:

- Epochs : 40 for static based gestures and 30 for dynamic based gestures
- Batch Size : 32
- Optimizer : Adam
- Learning rate : 0.01
- Weight Initializers : Xavier/Zero's

Artificial data synthesis was incorporated in real time for training of static based gestures. Table. I provides the results obtained after training all the models employing the specified hyper-parameters.

The depth only based CNN architecture and the RGB based CNN architecture achieve training accuracies of 97.43%and 97.21% respectively. The testing accuracies of both the depth and the RGB based models came up to 99.4 % and 99.75% respectively. The testing accuracy is higher than the training accuracy because artificial data synthesis was not adopted for the testing set, thus making the testing set much easier to predict than the training set. The combined RGB + Depth based CNN model achieved a training accuracy of 98.81 % and the testing set accuracy came up to 99.6 %, outperforming the performance of the other two models. Image to the left of Fig. 11 shows the plot of epochs vs Loss/Accuracy for the same. To evaluate the performance of the models in real-time, people of age groups ranging from (5-50), about 50 people in number (non-inclusive of the subject matters chosen for training) were asked to perform the gestures in front of the camera in real time. It was observed that the performance of all the three models was similar i.e. all the gestures shown by all the people were predicted accurately.

Training unsegmented RGB images gave 95.87 % training accuracy. When the model was deployed in real-time, it showed a decline in performance when compared to the model trained with segmented RGB images. Thus, background subtraction proved to be crucial and effective while deploying our model in real-time. Figs. 12,13,14,15,16,17 shows the examples of real-time prediction for alphabets I, V, W and numbers 1, 2, 3. As it can be seen, the model achieves accurate prediction of pairs V and 2 , I and 1, W and 3.

Transfer learning on using ASL dataset resulted in an accuracy of 97.71 %, almost equal to the accuracy obtained in [8]. Moreover, the model in [8] was trained for 48 epochs, eight more than our epoch hyper-parameter value. Thus, the model showed adaptability by extracting effective features from ASL dataset.

For the dynamic dataset, artificial data synthesis was not adopted due to limitations in the computational resources, but as stated earlier the data was recorded at 9 different frame rates to achieve temporal variability. The depth only dynamic model obtained an accuracy of 97.52% and the RGB only obtained an accuracy of 98.11% on the training set. The combined RGB + Depth model obtained an accuracy of 99.08 % on the training set. Image to the right of Fig. 11 shows the plot of Epochs vs Loss/Accuracy for the same. However, the performance on the testing set was 78.3 % and around the same level for depth only and RGB only dynamic model, even though regularisation techniques such as dropout were used, insisting the need to generate dataset having more variability.

VII. Conclusion and Future Work

This paper proposes a real-time model for ISL gesture recognition, based on the incoming image data from the Kinect. Effective real time backgorund subtraction was done using depth perception techniques. Computer vision techniques were used to achieve one-to-one mapping between the depth and the RGB pixels. Custom dataset were generated

TABLE I
RESULTS OF TRAINING

Sl No.	Method	Classes	Training accuracy (%)	Testing accuracy (%)
1.	Depth only CNN	36	97.43	99.4
2.	Segmented RGB only CNN	36	97.21	99.75
3.	Unsegmented RGB only CNN	36	95.87	99.68
4.	Depth + Segmented RGB CNN	36	98.81	99.6
5.	ASL with transfer learning	24	97.71	99.0
6.	RGB based ASL from [8]	24	97.8	100
6.	Dynamic Depth only	10	97.52	76.4
7.	Dynamcic RGB only	10	98.11	77.6
8.	Dynamic Depth + RGB	10	99.08	78.3

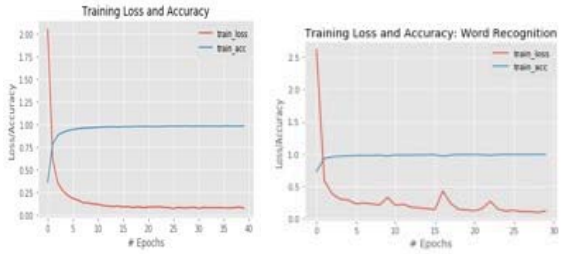

Fig. 11. **Left:** Plot of training loss/accuracy for dual channel static gesture prediction model.**Right:** Plot for dual channel dynamic gesture prediction model

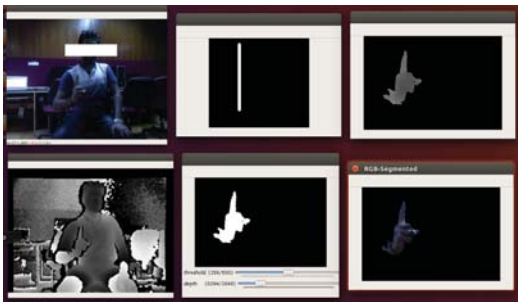

Fig. 12. Example of real time prediction for alphabet I

and different models were used for training. The depth + segmented static model achieves accuracy of 98.81 % and the dynamic model achieves 99.08% on the training set. Incorporating artificial data synthesis, helped achieve real time implementation of all 36 static based gestures. Effective adaptability to ASL was also attained through transfer learning. The model trained on the dynamic dataset showed high variance leaving scope of further research in this area for achieving real-time performance. Also further research can be focused on real-time prediction of more words related to ISL and also on sentence formation.

ACKNOWLEDGMENT

We would like to extend our gratitude to Project staff at the DSP Lab, Department of Electrical Engineering, IISc Mr. Navin Kumar H A, Ms. Srujana Subramanya and all other interns at the lab for their support and help without which this work could not have been accomplished.

Fig. 13. Example of real time prediction for number 1

Fig. 14. Example of real time prediction for alphabet V

Fig. 15. Example of real time prediction for number 2

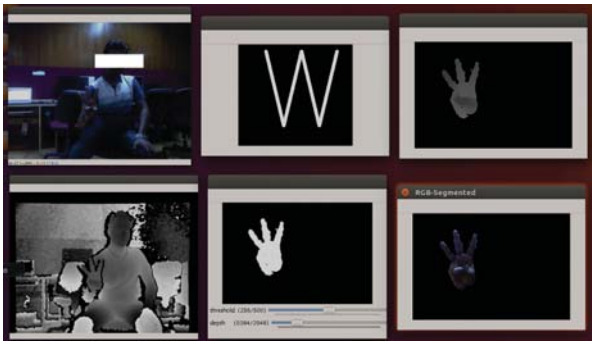

Fig. 16. Example of real time prediction for alphabet W

Fig. 17. Example of real time prediction for number 3

REFERENCES

[1] Q. Wu, Y. Liu, Q. Li, S. Jin and F. Li, "The application of deep learning in computer vision," 2017 Chinese Automation Congress (CAC),Jinan, 2017, pp. 6522-6527.

[2] D. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition, vol. 13, no. 2, pp. 111122, 1981.

[3] D. G. Lowe,Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 13, no. 2, pp. 111122, 1981.

[4] V. Bhavana, G. M. Surya Mouli and G. V. Lakshmi Lokesh, "Hand Gesture Recognition Using Otsu's Method," 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, 2017, pp. 1-4.

[5] Y. Liu, J. Zhang, and J. Tian, An image localization system based on gradient Hough transform, MIPPR 2015: Remote Sensing Image Processing, Geographic Information Systems, and Other Applications , 2015

[6] J. Singha and K. Das, "Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique", International Journal of Advanced Computer Science and Applications, vol. 4, no. 2, 2013..

[7] Pei Xu, A real time hand gesture recognition and human-computer interaction system, In: Proceeding of the Computer Vision and Pattern Recognition, 2017.

[8] B. Liao, J. Li, Z. Ju and G. Ouyang, "Hand Gesture Recognition with Generalized Hough Transform and DC-CNN Using Realsense," 2018 Eighth International Conference on Information Science and Technology (ICIST), Cordoba, 2018, pp. 84-90.

[9] K. O. Rodriguez and G. C. Chavez, Finger Spelling Recognition from RGB-D Information Using Kernel Descriptor, 2013 XXVI Conference on Graphics, Patterns and Images, 2013.

[10] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[11] A.Karpathy,G.Toderici,S.Shetty,T.Leung,R.Sukthankar, and L. Fei-Fei "Large-scale video classification with convolutional neural networks" In CVPR, 2014.

[12] Okan Kpkl, Ahmet Gunduz, Neslihan Kose, Gerhard Rigoll, Real-time Hand Gesture Detection and Classification using Convolutional Neural Networks, paper accepted to IEEE International Conference on Automatic Face and Gesture Recognition (FG 2019).

[13] Xiaokai Chen, Ke Gao DenseImage Network: Video Spatial-Temporal Evolution Encoding and Understanding, paper submitted to ArXiv on 19 May 2018.

[14] O. Kopuklu, N. Kose, and G. Rigoll, Motion Fused Frames: Data Level Fusion Strategy for Hand Gesture Recognition, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018.

[15] Mukesh Kumar Makwana, Sign language Recognition, Mtech thesis submitted to Indian Institute of Science, Bengaluru, June 2017.

[16] ASL database source: Kaggle - ASL alphabet dataset

[17] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad and S. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features", IEEE Access, vol. 6, pp. 1155-1166, 2018.