Deep Learning Methods for Indian Sign Language Recognition

Pratik Likhar
Department of Electrical Engineering
Indian Institute of Science Bangalore
Bengaluru, India
pratiklikhar@iisc.ac.in

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

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

2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin) | 978-1-7281-5885-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCE-Berlin50680.2020.9352194

***Abstract*—Indian Sign Language is the language used by specially abled people in India. Unfortunately the general population has no understanding of the sign language which hampers the communication between the specially abled and the general population. We are proposing a methodology to bridge this gap. We have used two approaches to solve this problem. First using the depth+RGB data captured using a Microsoft Kinect and predicting the gestures in real time. For segmenting the hand region from the data obtained by the RGB-D camera we used 3D reconstruction and affine transformation to map the depth and RGB information. Convolutional neural networks were used and segmented hand images/videos were used as an input to them. 36 static hand gestures from Indian Sign Language were trained and a classification accuracy of 98.81% was achieved on the test data. This model also showed a good performance when we transfer learned the American Sign Language giving a classification accuracy of 97.71%. LSTM with a convolutional kernel was used for training 10 dynamic gestures. This model achieved a classification accuracy of 99.08%. But as soon as we implemented this system, we figured out there is an inherent problem with this methodology. It is practically unreasonable to carry the bulky Microsoft Kinect around along with a system capable of performing the computation to communicate with people. We attempted to solve this problem using semantic segmentation of the hands. We used U-Net with ResNet 101 as the backbone for the same. Semantic segmentation utilises the input from a normal RGB camera which completely removes the necessity of using a RGB-D Kinect camera. We performed multi-class semantic segmentation which gave an IOU score of 0.9920 and an F1 score of 0.9957 on the training data. The above models performed extremely well in real time.**

***Index Terms*—Indian Sign Language, Gestures, U-Net, ResNet, Semantic Segmentation, CNN, LSTM, Microsoft Kinect**

I. Introduction

Indian Sign Language system is the predominant sign language in India that contains standard hand-based gestures which are used by speech impaired people for communication purposes. The general population have no knowledge of the sign language gestures as these gestures are very extensive and complex which in turn hampers the knowledge sharing between speech impaired people and the general population. In spite of recent surge in the research work done in the fields of deep learning and computer vision [2], the research carried out in the recognition of ISL gestures has been very limited. Our paper focuses on establishing a benchmark for recognizing ISL gestures in real time. As we know that ISL utilises both hands to portray a gesture, it is quite a challenging task to recognize gestures in ISL. This in turn increases the complexity while applying feature extractors like Hough Transform [3] and Scale Invariant Feature Transform [4]. We know that background complexity also plays a huge role in altering the accuracy of predictions. We have also seen that although techniques like segmentation using colour spaces and Otsus Technique [5] can be employed, they have their limitations with respect to the background conditions. In this paper we have used two approaches to tackle this problem, the first is using depth based segmentation(Kinect RGB-D camera) and the second is using semantic segmentation(normal RGB camera). The problem of lack of one to one mapping between the RGB pixels and the depth information prohibits the segmentation of the hand region. This problem is tackled in section IV-A of this paper. Furthermore, semantic segmentation can be used to tackle this very problem without using the depth information. This enables us to get rid of the bulky RGB-D Kinect camera and utilize a normal RGB camera. But there is a tradeoff for using the same because of its computational complexity. For the method utilizing Kinect camera, basic CNNs were chosen to be the deep learning architecture as we were already getting the segmented hand regions using depth information. 90,000 depth and RGB images were generated using the Kinect camera and we used them to train the CNN models along with augmentation. For dynamic gestures we used LSTMs along with a convolutional kernel. A total of 10 dynamic gestures were trained and 1080 videos were generated for the purpose of training. For semantic segmentation we have used U-Net with ResNet 101 as the backbone. This model comprises 101 convolutional layers with skip connections. We used the same dataset for training the U-Net as we had annotated images apriori. This paper is an extension of work originally presented in Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2019 [1]. **This paper differs from the previous work as we have introduced a new methodology for Indian Sign Language Recognition which does not require a depth based camera for inference**. The rest of the paper is presented as follows: Section II is dedicated to the previous related work. Section III explains about the datasets used and the methodology used to achieve generalization. Section IV explains the process for

segmentation using depth information from the Kinect camera and different architectures used to train the models which include CNNs and the U-Net. Section V presents the results we have obtained and section VI concludes the paper.

II. Related Work

until now, we have seen that a lot of work has been done to recognise hand signals but very little work has been done in context of Indian Sign Language. A database of 240 images was made for 24 signs by Singha et al. [6]. They adopted a methodology in which they extracted the eigenvalues front the images and they performed clustering based on the Euclidean distance. Their model achieved a classification accuracy of 97% but the model was invariant to the background conditions and hand gestures were of static kind. Also a Human Computer Interaction framework was developed by Pei Xu et al. [7] which achieved an accuracy of 99.8% on 16 gestures. This involved preprocessing of the input using filtering, gaussian blurring and morphological transformations. After preprocessing, the images were subjected to a CNN to differentiate between different motions and a Kalman estimator was utilized to move the cursor in case it detected the signals. Furthermore Intel Real Sense RGB-D sensor was utilized by Liao et al. [8] where depth information was utilized for segmentation of the hand regions. It was then subjected to a dual channel convolutional neural network which achieved a classification accuracy of 99.4% on 24 static gestures from the American Sign Language. Again the paper discussed the outcomes of training and no real time implementation was done. Otiniano et al. [9] segmented the hand regions from the background using Gradient Kernel Descriptors for the depth images and Scale Invariant Feature Transform Descriptors for the RGB images. The combined information was then provided to a Support Vector Machine which achieved a classification accuracy of 90.2% on the American Sign Language database. For dynamic based gestures, Molchanov et al. [10] utilized a R3DCNN - Recurrent 3D Convolutional Neural Network along with temporal classification which progressively recognized the dynamic gestures. Depth, RGB and stereo-IR sensors were used to collect the data. They had transfer learned the model as that R3DCNN was pre trained with Sport-1M [11] dataset. This model managed to achieve a classification accuracy of 83.8%. A hierarchical CNN based design was proposed by Okan et al. [12]. This model utilized two convolutional neural networks, a relatively light weight CNN for detecting hand gestures and a bulky 3DCNN to predict the gestures. The model managed to achieve a classification accuracy of 94.04% on EgoGesture benchmark and 83.82% on the NVIDIA benchmark. Again for dynamic based gestures Xiaokai et al. [13] proposed a method utilizing a Dense Image Network (DIN) which encodes the video containing the dynamic gesture to a conservative structure which distils its spatio-transient advancement. After that the output of the DNN is fed to a CNN where feature extraction is done in a more effective way. Another method utilizing the optical flow information was proposed by Kopuklu et al. [14] in which the optical flow data along with the RGB image was fed to a Deep Neural Network which achieved an accuracy of 84.7% on the NVIDIA benchmark. This model also accomplished a classification accuracy of 96.28% on Jester benchmark and 57.4% on ChaLearn benchmark. Mukesh [15] trained a CNN using a handcrafted ISL dataset and achieved a classification accuracy of 85.51%. Again this model was not tested in real time.

According as far as anyone is concerned this is the first attempt for creating a model that offers real time ISL gesture recognition without utilising the depth based camera. The semantic segmentation approach totally eliminates the necessity of using a depth based sensor. Our models perform well on all background conditions and hands of any size.

III. Dataset

A. Details of the Dataset

We created our own dataset for training and testing using Microsoft Kinect camera. The dataset was collected from 5 subjects having mean age of 25 and included both the genders. We obtained data for both the static and dynamic based gestures. 45,000 RGB images and 45,000 depth maps were obtained for static based gestures. Again the segmentation method explained in section IV-A was used for mapping the RGB pixels to the depth map. Fig. 1 shows instances of the segmented RGB-D pairs for letter sets E, G, M and T captured during data procurement.

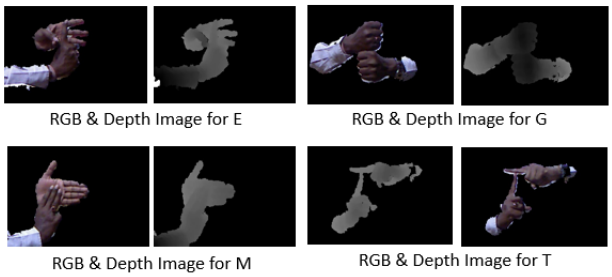

Fig. 1. ISL gestures as RGB images and depth maps.

For dynamic based gestures we captured 1080 recordings belonging to 10 distinct dynamic gestures frequently used in Indian Sign Language. The words chosen for the same were Airplane, Lock, Local, Licence, Low, Machine, Mall, Maths, Wifi and Win. Fig. 2 grandstands the gestures used for digits 0-9 and alphabets A-Z used in the Indian Sign Language System.

B. Techniques to attain Generalization

Our dataset had images in which the gestures were concentrated in an area, also we know that the CNNs are not scale, rotation and translation invariant and this can lead to a decline in performance of the model in real time. Therefore to achieve generalization, we used data augmentation. Further the dataset for static based gestures was captured in two lighting conditions so as to improve the real time performance. For creating the dataset for dynamic gestures, the videos were

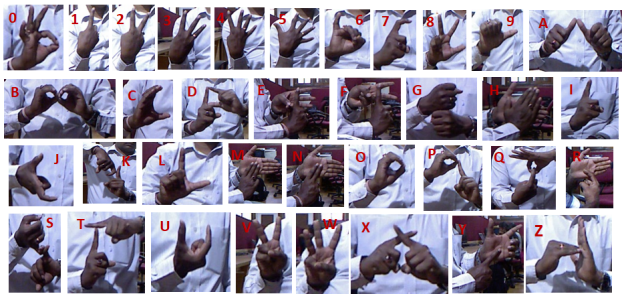

Fig. 2. ISL Signs for numerals and alphabets.

captured at different frame rates, i.e. 15, 18, 21, 24, 27, 30, 33, 36 and 39 fps to introduce variability in the training data allowing them to learn temporal features.

IV. Methodology

The purpose of this section is to discuss the methodology adopted for mapping RGB pixels to the depth map, and the architectures adopted for training the CNN models and the U-Net model.

A. Mapping depth and RGB pixels

We used Computer Vision techniques to map depth and RGB information. As shown in Fig. 3, we converted 2D depth map to 3D camera coordinate system by using extrinsic parameters such as distortion coefficient, skew coefficient, principal point and focal length. Using extrinsic parameters such as rotation and translation vectors of the depth camera, 3D camera coordinate system is converted to the 3D world coordinate system. Depth map is further converted from 3D world coordinate system to 3D RGB frame coordinate system by using extrinsic parameters of the RGB camera and then to 2D RGB frame coordinate system using the intrinsic parameters of the RGB camera.

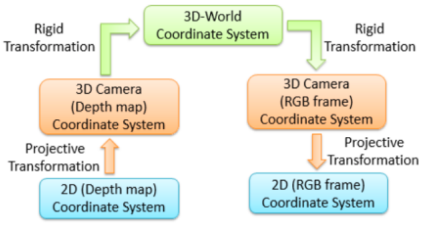

Fig. 3. Depth frame conversion from depth frame coordinate system to RGB frame coordinate system

B. Training architectures for static based gestures using Kinect(RGB-D) Camera

For static based gestures, we trained three separate convolutional neural networks. We at first trained a model using just the RGB images and a model utilizing the depth images. The architecture was the same for both the aforementioned models. The architecture for the same is shown in Fig. 4.

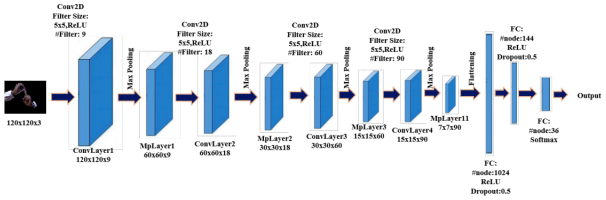

Fig. 4. Architecture for training depth or RGB only CNN model

Here ConvLayer refers to the convolutional layers, MpLayer refers to the max-pooling layer, FC refers to the fully connected dense layer and ReLU refers to the relu activation used for the convolutional and dense layers. The last layer uses the softmax activation for classification. Categorical crossentropy was used as the loss function. Fig. 5 depicts the whole workflow of this architecture.

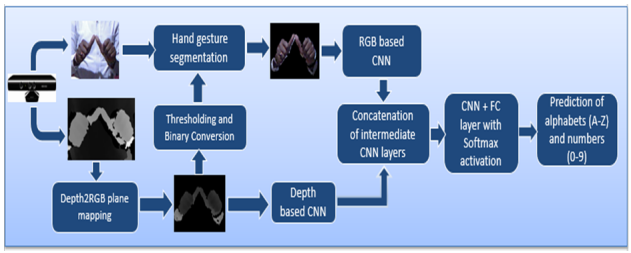

Fig. 5. ISL recognition flow diagram

The same architecture was used while transfer learning the American Sign Language. The weights from the convolutional layer were kept the same and the weights from the dense layers were reinstated to arbitrary small numbers. Another model was trained using both the depth and RGB images. Fig. 6 shows the architecture for the same.

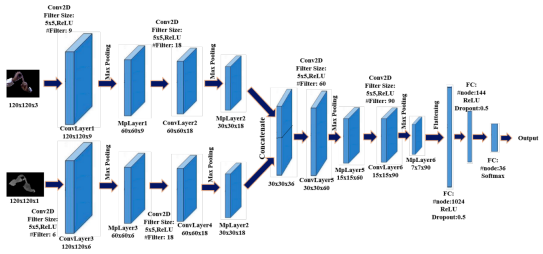

Fig. 6. CNN architecture for training RGB-D images simultaneously

The depth and RGB images are passed through two different channels until you get two sets of 30 feature maps having dimensions 18 by 18. After that these channels were fused into one after which they were followed with more convolutional layers, max-pooling layers and fully connected layers. Liao et al. [8] used almost similar strategy as ours.

C. Training architectures for dynamic based gestures using Kinect(RGB-D) Camera

For training dynamic based gestures we adopted the same strategy. We first trained separate LSTMs with convolutional

kernels for depth and RGB videos. The architecture for the same is shown in Fig. 7.

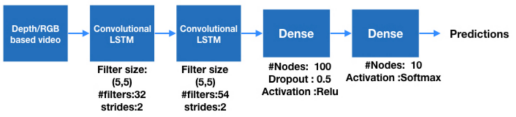

Fig. 7. Single channel architecture for video training

We also downsampled the videos by considering every 6th frame and discarded the others. This was done so as to remove the repetitive frames because as stated by Amin Ullah et al. [16] downsampling by 5 or 6 times does not decimate the temporal information.

The architecture for the dual channel LSTM is shown in Fig. 8. Here the depth and RGB recordings are sent through two unique channels of convolutional LSTMs at the same time before adding the layers together.

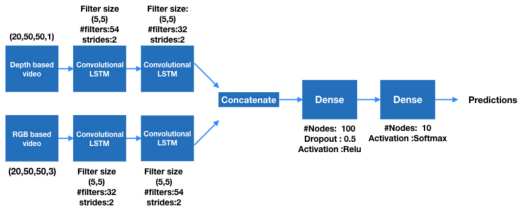

Fig. 8. Dual channel architecture for video training

D. Training architecture for static based gestures using normal (RGB) Camera

Olaf Ronneberger et al [17] developed U-Net for biomedical image segmentation. The architecture for the same consists of two paths, the first path is known as the contraction path or the encoder. The encoder consists of convolutional and max-pooling layers. It is used to capture the spatial features of the image. The second path is the expanding path also known as the decoder. The decoder is utilized for precise localization which uses transposed convolutions. The highlighted feature of a U-Net is that it only contains convolutional layers and no dense layers which enables the network to accept inputs of any size. Fig. 9 represents a typical U-Net architecture.

U-Nets perform extremely well in case of semantic segmentation as they combine the spatial information from the downsampling path with the contextual information in the upsampling path to output a highly precise segmentation mask. The loss function used in this particular problem was focal loss [18].

Here γ is a tunable parameter. We can analyze the implications of gamma value and understand the focal loss in the following manner. Let say the model misclassifies a hard sample with let say low confidence on the true class, that is γ_{nc} has a low value, in that case the compensation coefficient becomes close to 1, which in turn preserves the sample's contribution to the total loss. Now let's say an easy sample

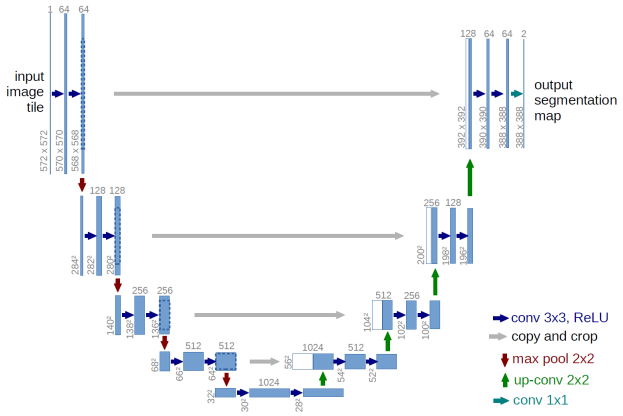

Fig. 9. Example of a U-Net architecture for 32x32 pixels in the lowest resolution. Here Each blue box represents a multichannel feature map. Here the white boxes depict copied feature maps and different arrows denote different operations.

$$L_{focal} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}(1-\hat{y_{nc}})^{\gamma}y_{nc}log\hat{y_{nc}}$$

is misclassified with high confidence value γ_{nc} . Then the compensation coefficient becomes close to 0 which in turn does not contribute to the total loss. This way focal loss focuses more on the hard samples that are misclassified rather than the easy samples.

ResNet 101 was used as the backbone for U-Net. Fig. 10 represents the ResNet 101 architecture in which we have 101 convolutional layers.

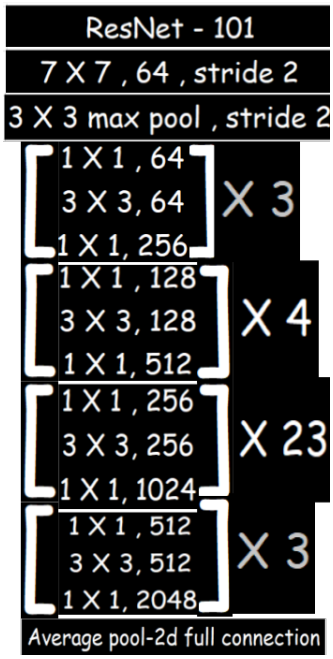

Fig. 10. ResNet - 101 Architecture

V. Training and Results

The first two models were trained on NVIDIA Tesla K80 GPU. The U-Net model was trained on NVIDIA GeForce GTX 1080 Ti. This section discusses the results obtained after training the models described in section IV-B, IV-C and IV-D. For training the CNN models, the dataset was divided as 70:30 between training data and testing data. For the U-Net model, the dataset was divided in the ratio 80:20 between training+validation data and testing data. The training+validation data was further divided in the ratio 80:20 between training data and validation data.

The hyperparameters used for training CNN models 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/Zeros

The hyperparameters used for training U-Net model were:

- Epochs: 40
- Batch Size: 2
- Optimizer: Adam
- Learning Rate: 0.01
- Weight Initializers: Xavier/Zeros

A. Predictions for CNN based models

We observed that the training and testing accuracies for the depth based CNN were 97.43% and 99.4% respectively. Also the training and testing accuracies for the RGB images were 97.21% and 99.75% respectively. The depth+RGB based CNN model was giving an accuracy of 98.81% for the training set and 99.6% on the testing set. The depth+RGB model clearly outperformed the depth only and RGB only based models. For real time testing, about 50 people having the age group ranging from 5 years to 50 years were invited. The above three methods predicted the signs almost accurately for all of them. We tried training the unsegmented images and the training accuracy came out to be 95.87% but after deploying the model in real time it showed a decline in the performance. We also tried to implement transfer learning using the American Sign Language dataset and the classification accuracy we got in that case was 97.71% which is almost equal to the accuracy obtained by the authors in [8]. For dynamic gestures the depth only LSTM with convolutional kernel achieved a classification accuracy of 97.52% on the training data. The RGB only LSTM model achieved a classification accuracy of 98.11% on the training data. The depth+RGB LSTM model achieved a classification accuracy of 99.08% on the training data. The performance of depth+RGB LSTM model showed a decline on the test data giving a classification accuracy of 78.3%. For depth only and RGB only LSTM models the classification accuracy was around the same for test data.

B. Predictions for U-Net model

The U-Net model with RESNET 101 as backbone achieved the final IOU score of 0.9920 and an F1 score of 0.9957 on the training data. The final IOU score and F1 score for the validation data was found to be 0.9844 and 0.9866. The model performed well on the test data giving the IOU score and F1 score of 0.97787 and 0.98087 respectively. The real time performance was measured on NVIDIA GeForce GTX 1080 Ti. Fig. 11 shows the prediction of the model on the image having label G, the ground truth mask for label G and the predicted mask for label G. Fig. 12 shows the prediction of the model on the image having label G, the ground truth mask for label L and the predicted mask for label L.

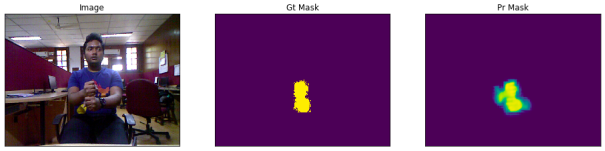

Fig. 11. Image for class G, Ground Truth Mask for class G and Predicted Mask for class G

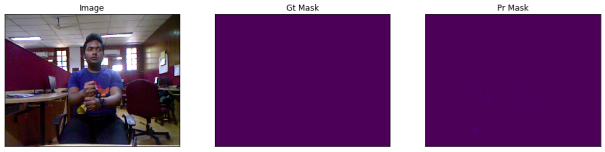

Fig. 12. Image for class G, Ground Truth Mask for class L and Predicted Mask for class L

Fig. 13 gives us the graph of IOU, loss and F1 scores versus epochs respectively for both training and validation data. Note that we expect the mask for label L to be empty for the image having label G. This way we are achieving both segmentation and classification at the same time. Fig. 14 depicts the real time performance of the U-Net model using a normal RGB camera. For real time inference we used argmax of the output(dimension of the output is 37x480x640, where we are getting 37 masks for each class) for each pixel which denoted the class to which that belongs and then assigned the class to the image which was assigned the most over all pixels.

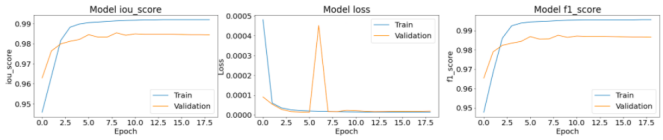

Fig. 13. IOU score, Model Loss and F1 score versus epochs

Again the performance was measured on a system having a GPU that is NVIDIA GeForce GTX 1080 Ti. The model was able to predict the signs accurately at 20 fps. For a normal system with no GPU the model took around 4 seconds to process a frame. Therefore significant improvement is needed in terms of computation time using the normal RGB camera on a system with relatively lower hardware specifications.

Fig. 14. Real time predictions on signs 5 and W

VI. Conclusion and Future Work

In this paper we have presented a method to implement real time Indian Sign Language gesture recognition using two methods. First using a depth+RGB based Microsoft Kinect camera and then using a normal RGB camera. For depth+RGB based techniques, the hand segmentation was done using depth perception techniques. For a normal RGB camera semantic segmentation approach was adopted. The usage of semantic segmentation completely removes the necessity of using a depth based camera and segmentation. The U-Net model achieved an IOU score of 0.9920 and an F1 score of 0.9957 using just the RGB camera. For the depth+RGB trained models, techniques like having different lighting conditions and data augmentation helped achieve the generalization in case of static gestures. For dynamic gestures, the procurement of data was done at various fps values so that the model can learn the temporal features. In case of models based on dynamic gestures, we observed high variance for the model performance and further research can be done to improve the same. Some research needs to be done on improving the real time performance of the U-Net model on systems with lower hardware specifications. Also we can take the opportunity to develop a real time sentence prediction model based on this paper.

Acknowledgment

We are very grateful to the staff at Digital Signal Processing Lab, Department of Electrical Engineering, Indian Institute of Science Bangalore for their continuous support. We would like to thank all the interns in DSP lab for their valuable contribution at various stages of the research.

References

[1] N. K. Bhagat, Y. Vishnusai and G. N. Rathna, "Indian Sign Language Gesture Recognition using Image Processing and Deep Learning," 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2019, pp. 1-8, doi: 10.1109/DICTA47822.2019.8945850.

[2] 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.

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

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

[5] 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.

[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] 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.

[17] Ronneberger, Olaf & Fischer, Philipp & Brox, Thomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS. 9351. 234-241. 10.1007/978-3-319-24574-4_28.

[18] Nguyen, Ty & Ozaslan, Tolga & Miller, Ian & Keller, James & Loianno, Giuseppe & Taylor, Camillo & Lee, Daniel & Kumar, Vijay & Harwood, Joseph & Wozencraft, Jennifer. (2018). U-Net for MAV-based Penstock Inspection: an Investigation of Focal Loss in Multi-class Segmentation for Corrosion Identification.