Connectionist Temporal Classification Model for Dynamic Hand Gesture Recognition using RGB and Optical flow Data

Sunil Patel[1] and Ramji Makwana[2]
[1]Computer Engineering Department, Gujarat Technological University, India
[2]Managing Director, AIIVINE PXL Pvt. Ltd, India

Abstract: *Automatic classification of dynamic hand gesture is challenging due to the large diversity in a different class of gesture, Low resolution, and it is performed by finger. Due to a number of challenges many researchers focus on this area. Recently deep neural network can be used for implicit feature extraction and Soft Max layer is used for classification. In this paper, we propose a method based on a two-dimensional convolutional neural network that performs detection and classification of hand gesture simultaneously from multimodal Red, Green, Blue, Depth (RGBD) and Optical flow Data and passes this feature to Long-Short Term Memory (LSTM) recurrent network for frame-to-frame probability generation with Connectionist Temporal Classification (CTC) network for loss calculation. We have calculated an optical flow from Red, Green, Blue (RGB) data for getting proper motion information present in the video. CTC model is used to efficiently evaluate all possible alignment of hand gesture via dynamic programming and check consistency via frame-to-frame for the visual similarity of hand gesture in the unsegmented input stream. CTC network finds the most probable sequence of a frame for a class of gesture. The frame with the highest probability value is selected from the CTC network by max decoding. This entire CTC network is trained end-to-end with calculating CTC loss for recognition of the gesture. We have used challenging Vision for Intelligent Vehicles and Applications (VIVA) dataset for dynamic hand gesture recognition captured with RGB and Depth data. On this VIVA dataset, our proposed hand gesture recognition technique outperforms competing state-of-the-art algorithms and gets an accuracy of 86%.*

Keywords: *Connectionist temporal classification, Long-short term memory, Hand gesture, Convolutional neural network, VIVA.*

Received February 2, 2019; accepted July 28, 2019
https://doi.org/10.34028/iajit/17/4/8

1. Introduction

Gesture recognition is an essential and significant research topic in Computer Vision and Pattern Recognition. It encompasses a wide-ranging number of applications in sign language recognition, face recognition [7, 20], online character recognition [3], human-computer interaction and so on. Recently, People can use deep convolutional neural for identification and recognition of gestures because of the implicit way of strong feature extraction. Due to numerous difficulties in hand gesture recognition task, the very first thing is to find a fitting and an exact route of motion in the video. This input Red, Green, Blue (RGB) video and Depth video can be used to get the appearance and pixel distance from the camera. But we can get appropriate motion by using optical flow. Therefore, we have found an optical flow from all the RGB frames for getting suitable hand movements in the video. This model's efficiency may be improved by adding optical flow because it can find dense motion in the video [24]. The second Connectionist Temporal Classification (CTC) is a costing parametric method that is used to training a Recurrent Neural Networks (RNN) to label unsegmented input sequence data in supervised learning and it can be successfully applied in speech recognition [5, 25], hand character recognition [11, 14] and action recognition [12, 13] in the video. After successful application of CTC network, we have applied this network to a dynamic hand gesture recognition. By using the CTC network, we can find proper label alignment for each gesture class in a supervised manner. This CTC network model is used for hand gesture recognition with direct frame level classification which completely removes both the need for pre-segmentation of gesture as well as post-processing of frame after classification. The whole architecture of the CTC network is trained beginning-to-end with a calculation of CTC loss. Our main attention is successfully recognized each hand gesture with greater precision by using this CTC network.

In this proposed paper, we have implemented a dynamic hand gesture recognition technique built on the input of RGB Video. In the beginning, to acquire appropriate information about the gestures all the input videos are converted into a fixed number of 35-

frames. After that, we have measured an optical flow to find an appropriate route of hand motion present in all the sequences of RGB frames. Next, we pass each frame one-by-one to the CNN model. This two-stream CNN model extracts and learns deep features implicitly. We aggregate this deep feature for improving the performance of this model. The Long Short-Term Memory network (LSTM) has applied for temporal recognition of this merged feature. This output of the recurrent network is connected to CTC Network for proper gesture label alignment. The last stage is CTC Decoding which generates a final label sequence for each class. The code of my proposed work is put on my blog https://sunilpatel.home.blog/research/. The step-by-step process for our suggested approach is characterized in Figure 1. The key elements of our proposed approach are summarized as follows:

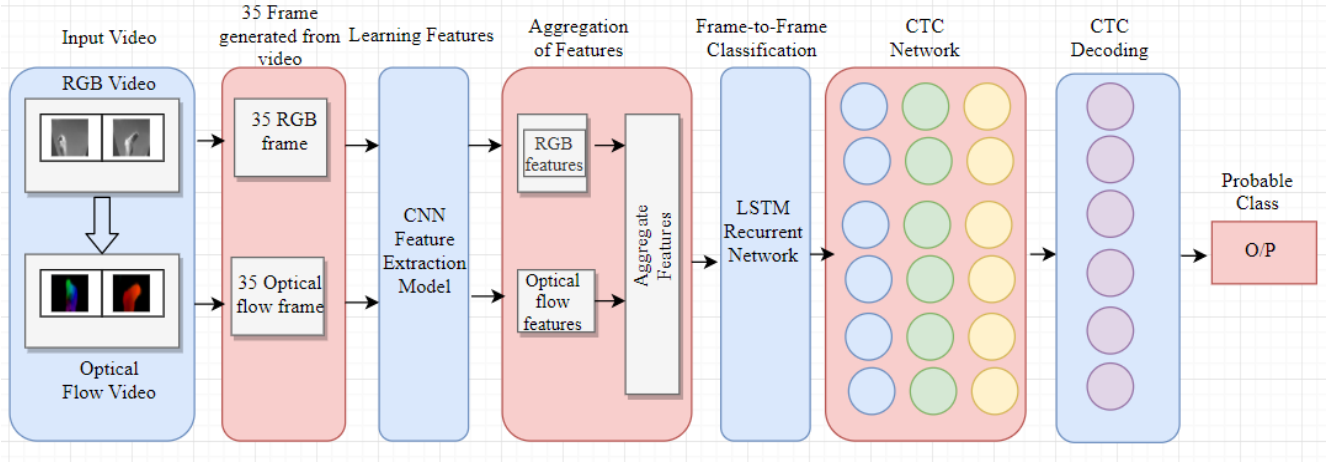

Figure 1. This is the broad architecture of this proposed model. It contains 35-frame unification process, motion calculation block, feature extraction block, LSTM recurrent network block, CTC Network block, CTC decoding process and class probability of each gesture.

- The standard way to assign input from the video is 16 frames. We have evaluated our model in 35-frames after the unification of frames. Due to the unification process, it can extract robust and more meaningful features. Therefore, this proposed model easily recognizes similar gestures with an ignorable difference. We select 35-frame as unified value and convert all the videos to this same number of frames.
- We introduce a method for joint feature extraction, classification and calculate CTC loss for complex hand gesture recognition using RGB and Optical flow data stream.
- We have introduced a CTC model for this task, a way to effectively evaluate all the frame for proper gesture label alignment using dynamic programming. This whole CTC network is trained beginning-to-end with the calculation of loss for hand gesture recognition.
- This CTC model had applied only on speech recognition and handwritten character recognition. we have successfully applied on hand gesture recognition. We experimentally evaluate that CNN, LSTM, and CTC Network can learn better feature for all explored architecture.
- We experiment with our CTC model on lengthy and complex hand gesture recognition videos from the standard Vision for Intelligent Vehicles and Applications (VIVA) dataset and show that our method achieves 86% accuracy.

The following is the form of this research paper. Secti on 2 describes the related work. After that in section 3, we have discussed our proposed hand recognition work based on the CTC network and CTC Decoding. In the 4th section, we discuss results analysis and experimental environment. In the 5th section, we have given concluding remarks of our proposed model and gives a different way for further work.

2. Related Work

In this work, we will briefly represent recent previous work on hand gesture recognition. All of the preceding approaches use handcrafted features extraction way, including probabilistic neural network [26], decision support system rules [2] and contour-shape based hand gesture features [19] with a hidden markov model used for classification. Ohh-Bar and Trivedi [18] applied the Support Vector Machine (SVM) as a classification method and Histogram Of Gradients (HOG) as a feature extractor for RGBD data for hand gesture recognition. Rather, handcrafted features, people can work on recent techniques on the Convolutional Neural Network (CNN) with the inclusion of RN) for activity recognition and sign language recognition with automatic learning complex features. Molchanov *et al.* [15, 16] successfully applied a CNN with three-dimension (3D-CNN) in video sequence for classification of hand gesture along with space and time. In [22] 3D convolution neural network is used for extraction

space-temporal feature and use SVM classifier for classification. In the dynamic hand gesture recognition task given in [4] use RGB and depth as input for getting background information, use the Recurrent Neural Network (RNN) for temporal recognition and finally classified using the SoftMax classifier. The method presented by Molchanov *et al.* [17] use Recurrent Convolutional neural networks with three-dimension using image gradient for recognition of hand gesture, that supports online and offline classification of gesture with zero or no lagging, all the features are fused using various fusion schemes and trained the whole architecture with weak and strongly segmented videos. Tsironi *et al*. [23] published an approach use the deconvolutional neural network for visualizing the features learned from a homemade prepared dataset, and use a sequential convolutional layer with Convolutional Neural Network- Long Short Term network (CNN-LSTM) for extraction of features from the original input video sequences. The author has applied deep learning using deep network architecture for this standard VIVA hand gesture dataset [1].

CTC model has been successfully applied to speech recognition and handwritten character recognition. In speech recognition, during decoding a large number of frames are blank and CTC Lattice network can be used to remove bank frame by completely remove traditional method Frame Synchronous Decoding (FSD) into Phone Synchronous Decoding (PSD) [5]. A novel method used with the combination of CNN-RNN-CTC classification model for multi-accent mandarin for automatic recognition of speech to improve the performance [25]. The author published a method with the combination of CTC model with Lattice-Free Maximum Mutual Information (LFMMI) for offline handwritten character recognition, all the features are learned by using a combination of CNN with dimension reordering using Principal Component Analysis (PCA) and feature extractors, whole architecture is trained beginning-to-end through very deep unique long short term bidirectional memory network [11]. Extended Connectionist Temporal Classification (ECTC) can be used for learning a temporal model of action in a weakly supervised setting to explicitly capture visual similar dependency between consecutive frame using a dynamic programming-based algorithm [13]. CTC and statistical Language model are used for recognizing actions sequence where boundaries of the actions are not given but several actions are concatenated [12].

Among the discussed above method, but our proposed effort has three extensions. We have found 35-frame as a unified value after the unification of frames. We have given more attention to motion information using optical flow to completely eliminate irrelevant information from the context and find a fitting motion path. From this perception, our proposed work is similar to the effort given by Danafar and Gheissari [6] but the object movement data is calculated based on the optical flow histogram. We have measured an optical flow from each RGB fame to obtain hand movement information present in the RGB video. At the same time, we have supplied RGBD frames and optical flow frames as the CTC model's input. This proposed three-stream RGBD and Optical flow network aggregate features using various fusion schemes. The third one is we used CTC Network till it is applied only speech recognition and handwritten character recognition. We have successfully applied this model to hand gesture recognition. Our work is an extension of Tsironi *et al*. [23] as they mention in their future work. CTC network is used for finding a frame for proper gesture label alignment. Finally, CTC max decoding can be used for finding proper gesture sequences for classification without the need for post-processing of data.

3. Proposed Method

Compared with all the preceding works, our gesture recognition work, pointing to solve video-based dynamic recognition of hand gesture, faces many complexities in the extraction of important features. This model uses video as an input to the system instead of a single image therefore it required more effort for temporal learning of features. This proposed deep learning-based model automatically extracts and learn more about the features in the temporal domain and spatial domain simultaneously. The CTC network initially applied only to handwritten character recognition and speech recognition for proper label alignment. We have successfully applied this model to hand gesture recognition and it can identify frames that are part of the relevant gesture class. In our framework, we have trained CNN-LSTM with the CTC model for spatiotemporal feature extraction and find the proper label aligned frames. In the beginning, at once, after examining all the RGB video, we have converted unique 35-frames for better-quality feature extraction. Then we have calculated an optical flow for getting proper motion direction. This process automatically removed irrelevant content from the background and focus only on hand motion present in the video. Subsequently, this RGB and Optical flow frames are passing one-by-one to CNN model to obtain features of the video. After that, the learned RGB and Optical features are aggregated for further advance refinement. Then, these aggregated learned features are passing LSTM one-by-one to find the relationship between two frames. Finally, the CTC network can find a proper gesture relevant frame and CTC decoding generate gesture sequence.The details of our CNN model implementation and calculation of

optical flow, the 35-frame unification strategy, CTC network with decoding will be presented in later subsections.

3.1. Unification of Frame and Calculation of Optical Flow

The prerequisite of the CNN model, input to the system is the same number of frames, that means the same width and same height of each frame, all should be in the same form. After observing all the 1460 videos in the VIVA dataset, we have found that approximately 600 videos have frames less than or equal to 35 and the remaining videos have frames greater than 35.We have calculated an average of these frames is near to 37, but we select 35 as a benchmark. To extract the same characteristics of features from gesture as much as possible, after the conversion of unification of frames, we calculate optical flow from consecutive RGB Frames using Farneback [8]. Optical flow can be used to find the proper route of hand motion shown in the video. In this recommended CTC based model, we have used three different sets: training samples, testing samples, and validating samples. The more details for this broken-down VIVA dataset categories are described in Table1.

Table 1. Three different sets of categories for this standard VIVA dataset.

Three sets	Class Category	Total Video
Training Sample	19	1168
Testing Sample	19	146
Validating Sample	19	146

3.2. Feature Extraction using CNN and LSTM

Gestures are normally presented in the video, solving gesture recognition depends only on feature extraction. This two-stream CNN network has a 5-convolutional layer with 14 different convolutional operations. It also has 4 different Maxpooling layer for reducing the dimension of image and forwarded by 2-Fully connected classification layers is illustrated in Figure 2. On the first layer it generates 32 feature maps with 32 kernels. The (3,3)-pixel of kernel size is used for feature extraction in the image and the padding is set to 2. After that, the next hidden convolutional layer has a total of 64 kernels and each kernel size is (3, 3)-image pixel. The total number of kernels used for third, fourth and fifth hidden convolutional layers has 128,256,512 respectively.The final output of fully connected layer fc6 and fc7 is 4096-dim.We have a total of 1460 videos and the size of each frame is 115X250-pixels. We have used (2,2)-pixel Max-Pooling for reducing the size of each frame by a factor of 2. We have two stream networks with RGB and Optical flow feature extractors. The final score after merging two streams is 4096-dim which is input to the LSTM network at the different timestamp. LSTM is a special recurrent network it can store data for a long period of time [10].

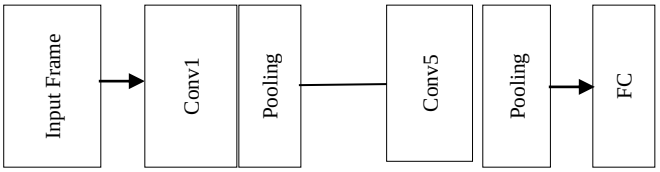

Figure 2. This is a pipeline of the CNN model. This model contains a 13-convolutional layer,4-Max-Pooling operations with (2X2) filters, and 2-Fully connected layers with 4096-dim array vector.

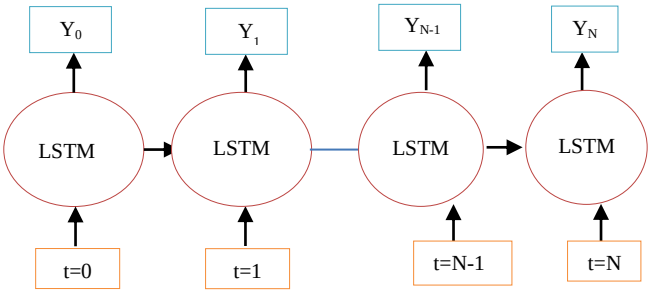

Figure 3. This is an architecture of LSTM Model. It has 512 cells of the memory. Total LSTM node is 35. The final output is (19,35).

Therefore, it can learn deep features and merge present and past information in the memory cell. In our proposed architecture we have one hidden layer with N number of LSTM node, where N=35. The input to LSTM node is 4096-dim feature vector produced by after CNN. Each timestamp (t=0 to N) it can generate a probability score for 19 different class. Final score after LSTM is (19, 35) probability value which is input to CTC Network. Final LSTM architecture is represented in Figure 3.

3.3. CTC for Label Alignment and Calculation of Loss

CTC is a cost function that is used to train Recurrent Networks to label unsegmented input sequence data in supervised learning [9]. CTC network can transform the output into a conditional probability distribution over gesture label sequence for a single class. The network then identifies the most probable frame to ground truth sequence. As shown in Figure 4, for example, ground truth sequence is [1, 2, 17,18, 19] for class M. then CTC network generates the highest probability score on this possible alignment. There is an exponential number of possible paths along with this ground truth sequence because we have 35 different timestamps with 19 possible probability value. The network collects all the possible alignment and calculates CTC loss. The entire network trained back propagation after calculating CTC loss. Our video clip can contain only gesture information not irrelevant information, so we do not include "no gesture" class. In this work, we have considered CTC forward and backward algorithm for calculating loss and gradient.

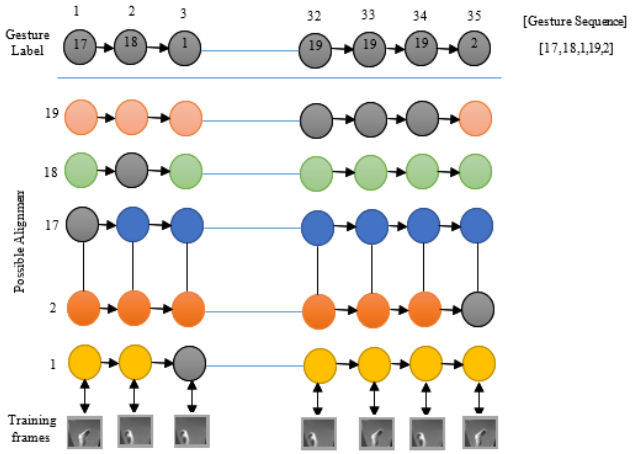

Figure 4. CTC Network with training frame and gesture label. Ground truth and probable selection of gesture label are represented in purple colour.

The output of the LSTM recurrent network is connected to CTC network. In our case output of LSTM recurrent layer is (*y*, *X*) where y=19(gesture label) and X=35(total input frame).The video output of LSTM for each activation frame with different timestamp is X=(x_1, x_2, x_3…. x_T) and probable gesture sequence is y=(l1, l2, l3…l_N)., here T>N means number of activations is larger than gesture label sequence.CTC states the probability of input frame X to gesture label sequence L is:

$$P(\frac{y}{X}) = \sum_{\pi \in \beta^{-1}(y)} P(\frac{\pi}{X}) \quad (1)$$

Where, $\pi = \{\pi^1, \pi^2, \pi^3 \ldots \pi^T\}$ is a path denoting gesture sequence label along with input X, and β is an operator to eliminate consecutively repeatedly gesture labels sequence in a possible path π. That means both the gesture path β [1, 1, 2, 3] and β [1, 2, 2, 3] are mapped to a many to one gesture path β [1, 2, 3]. In this way every x^t in *X* gives a contribution to the generation of different path by aligning each label in y through β. Then, CTC assumes that each π^t in π is conditionally independent given input *X*: the probability of each gesture path is given by:

$$P(\frac{\pi}{X}) = \prod_{t=1}^{T} P(\frac{\pi^t}{X}) \quad (2)$$

We can calculate the value of $P(\frac{\pi}{X})^t$ by using the output of the LSTM function at a timestamp t. It can map x^t to the probability of gesture classes via linear and a Soft Max output Layer. The complexity of $P(\frac{y}{X})$ grows exponentially with respect to the input frame length *X*. we could try the straightforward approach and compute the score for each gesture alignment and summing all the gesture sequence. if we calculate the CTC loss by using the Equation (1) is very expensive to compute because of a massive number of alignments between a gesture sequence. Therefore, we can efficiently calculate the loss by using dynamic programming. Let π_1^t represent a partial gesture label sequence path of π from 1 to t, and y_1^k denotes a gesture label sequence composed by first k label in order from y. From that we can calculate the value of $\alpha_t(k)$ variable, which is a summation of the probability of π_1^t that can be aligned to y_1^k:

$$\alpha_t(k) = \sum_{\pi \in \{\pi:\beta(\pi_1^t)=y_1^k\}} P(\frac{\pi_1}{X})^t \quad (3)$$

We can calculate, $\alpha_t(k)$ recursively as below:

$$\alpha_t(k) = [\alpha_{t-1}(k) + \alpha_{t-1(k-1)}]\, s^t(y^k) \quad (4)$$

Where the emitting probability of $s^t(y^k)$ is produced by the gesture label y^k at timestamp t. That means, $s^t(y^k) = P(\frac{\pi}{X})^t$ is all the possible path π and it has gesture label y^k at timestamp t. As per defines in Equation (4), the mapping of different input frame *X* to ground truth gesture label y up to timestamp t and k^{th} label of gesture, permitting the transition from $(k-1)^{th}$ and k^{th} label of gesture at timestamp t-1. We can calculate $P(\frac{y}{X})$ by deriving $\alpha_t(k)$ in linear time by using dynamic programming with respect to the input length of the video. CTC defines the loss function as negative loglikelihood for the LSTM recurrent neural network as:

$$L = -\log P(\frac{y}{X}) \quad (5)$$

CTC introduces another variable $\beta_t(k)$ that expresses mapping of different input frame *X* to ground truthgesture label y up to timestamp t and k^{th} label of gesture. The gradient of loss is calculated by using this backward variable with respect to the parameter of LSTM recurrent neural network.

$$\beta_t(k) = [\beta_t(k) + \beta_{t+1(k+1)}]\, s^t(y^k) \quad (6)$$

Where all the variable describes in Equation (6) is the same as in Equation (5). The value of $\beta_t(k)$ defines the mapping of different input frame X to gesture label y up to timestamp t and k^{th} gesture label but from the end of input frame *X*. The gesture label y does not start from the beginning as we calculate the value of $\alpha_t(k)$. By calculating $\alpha_t(k)$ and $\beta_t(k)$, the total loss is calculated by equation 5 as described in preceding part and with respect to the output of $s^t(m)$, it is the SoftMax output of the recurrent neural network for the m^{th} label class at a time t. By calculating $\alpha_t(k)$, $\beta_t(k)$, we can find total loss, Here the total loss is defined as below:

$$L = -\log P\left(\frac{y}{X}\right) = \sum_{s=1}^{S} \frac{\alpha_t(k) * \beta_t(k)}{s^t(m)} \quad (7)$$

So, we can train an entire network using stochastic gradient descent with back propagation through time t and with the support of CTC loss [21].

- **CTC Decoding**

The output of the CTC network is different gesture sequence with respect to timestamp t. It is decoded by max encoding method as follows:

$$y = \underset{\pi}{\text{argument max}}\ P\left(\frac{y}{X}\right) \quad (8)$$

$$y = \beta\left(\underset{\pi}{\text{argument max}}\ P\left(\frac{\pi}{X}\right)\right) \quad (9)$$

We can calculate $P(\frac{y}{X})$ by using $P(\frac{\pi}{X})$ where, the most probable path is π among the gesture label y. In the max encoding method path with the highest probability value at each timestamp is considered and concatenate them with the best path. We can apply β (.) operator to remove and concatenate them which has the same gesture label.

3.4. Classification and Training

The last part of our network is the classification and training. We get the probability value of the label sequence for each class by the SoftMax classifier. We have used Back Propagation Through Time (BPTT) algorithm for training this proposed entire CNN-LSTM and CTC network. The entire CTC loss is calculated by giving Equation (7) in section 3. We have used a stochastic gradient descent method for optimizing network weight. This proposed model is not fine-tuned and therefore it is not pretrained with network parameter. For training of this proposed model, all the initial weights are set to a random value with standard mean to zero. For train a whole network, the momentum and weight decay are adjusted to 0.7 and 0.005. The initial rate of learning is adjusted between 0 to 1. After improved accuracy, the process would stop automatically.

4. Result and Discussion

In this section, we authenticate the efficiency of our proposed method by a number of experimentations. In this first sub-section 4.1., the VIVA dataset is briefly introduced in the experiment. This dataset mainly contains gestures for smart interfaces in vehicles. Next, in section 4.2., we have explained the experimental environment used for this running model. After that, the evaluation criteria in the form of accuracy are illustrated with an Equation. In the last 4.4-4.7 section, we have performed a series of experiments on this set of data and check the result, means the impact of CTC model and without CTC model, conduct experiment in 35-frames and 16-frames unification strategy, the CTC model is tested on RGB and optical flow, this proposed method is compared with all the conventional method applied to this dataset.

4.1. Dataset

The VIVA dataset comprises data about usual human action to provide a touchless interface in cars with 1460 RGB and Depth video [18]. All the gestures are captured under low resolution, the similarity in action and varying illumination. This dataset is designed through a Microsoft Kinect camera invented in 2010. The resolution of each video is 115X250 pixels and varying lengths. There are 8 different anchors performed this dynamic hand gesture and it has 19 different classes. To create complexity in recognition some gestures are captured by the driver or passenger left or right hand. All the gestures with some screenshot are represented in Figure 5.

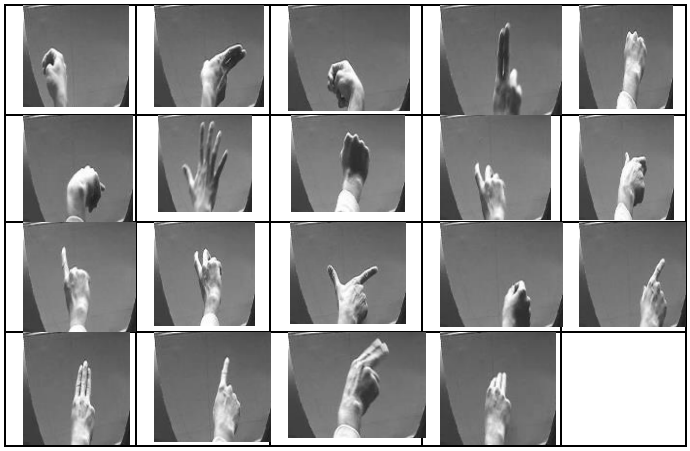

Figure 5. VIVA dataset has 19 different class. A single snapshot of each class of gesture is represented with resolution of 115 X 250.

4.2. Experimental Environment

We have done an experiment on a PC with Intel Core i7-8086K processor and 6 cores, 16 GB RAM and Nvidia GeForce 940MX GPU. This hi-tech PC contains NVIDIA GeForce 940MX GPU for improving accuracy and reducing training time. We have used Keras with OpenCV, an open-source library written in python and TensorFlow as a backend. The specification of this platform is Windows 10, 64-bit operating system and cuDNN 8.0 framework. We have trained our CNN-LSTM and CTC model for the extraction of features and classification by the above PC specification and Keras library

4.3. Evaluation Criteria

The recognition rate for this hand gesture task is defined below:

$$\text{Accuracy} = \frac{\text{True Positive sample}}{\text{Total Video in the dataset}} \quad (10)$$

Where True Positive means correctly classified by the proposed model. The predicted value is the same as the ground truth value.

4.4. Comparison with CTC Model and without CTC Model

In this subsection, we validate the efficiency of our approach with CTC and without CTC in detail. The classification accuracy with this approach is described in Figure 6. As per observing the result, the involvement of the CTC model gives a better recognition performance**.** CTC Network can find proper frame alignment between input and output. There is no alignment issue for proper recognition of hand gesture. On the other hand, without CTC model used all the random frame for final probability calculation in the video. Finally, we can say that the approach represented with CTC is proper and it can find proper label alignment for a different gesture.

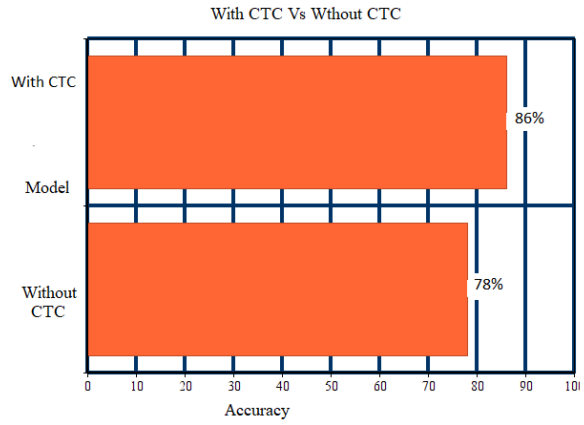

Figure 6. The CTC model and without CTC model result comparison. The accuracy is increased with the support of the CTC model and optical flow.

4.5. Comparison with 16-frame CTC and 35-Frame CTC

In this subdivision, we validate the efficiency of our 16- frame and 35-frame approach with CTC in detail. The classification result of 16-frames and 35-frames input to the proposed CTC network is described in Figure 7.The results indicate that the 35-frames CTC model gives a better recognition performance compared to 16-frames CTC. The performance of the network increases due to support of a greater number of frames because it finds very close gesture alignment in the video. Due to involvement of a greater number of frames 35-Frame CTC network can easily differentiate those gesture which are very similar in nature. Finally, we can conclude that our 35-frame strategy with the CTC network is sufficient, andwith a larger number of frames, it can find more likely frames for each gesture class.

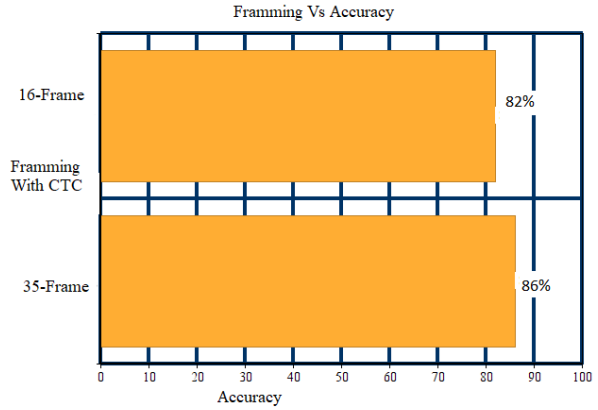

Figure 7. This is a 16-frames and 35-frames CTC model result comparison. The accuracy is increased with the support of 35-frames CTC model and optical flow due to proper gesture alignment.

4.6. Comparison of CTC with RGB and CTC with RGB and Optical flow

In this subsection, we have checked the efficiency of our method on RGB data with CTC and RGB+OF data with CTC in detail. The classification accuracy with this approach is described in Figure 8.

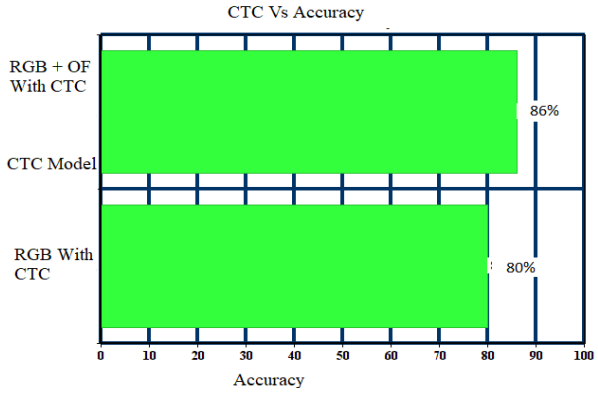

Figure 8. Result comparison with 35-frame CTC model with RGB data and 35-frame CTC model with the inclusion of (RGB, Optical flow) data as the machine input.

We have found motion information from RGB video data. By using proper motion path, we can get proper movement and direction of gestures. After getting rgb and motion information we can find proper motion alignment using CTC network. According to the remark, we can conclude that the classification result of this proposed CTC network is enhanced with the inclusion of measured optical flow. At last, we believe that the methodology represented with CTC is appropriate and it can find a proper route of hand motion available in these frames of video.

4.7. Comparison with proposed CTC Method and Existing Method

The precise classification rate of comparison with the various existing technique is specified in Figure 9. We compare our proposed CTC Network classifier with Ohh-Bar and Trivedi [18], HOG+HOG features and Molchanov *et al.* [15], use Deep Neural network by Alghamdi *et al*. [1], Use Low Resolution Network (LRN), High Resolution Network (HRN) and LRN+HRN convolutional neural network. Our approach is superior to 86% with recognition performance.For that kind of, dynamically recognize hand gesture task in the car, the outstanding recognition efficiency of the conceptual CTC model is performing greater than the existing literature's approach. There is some limitation is there that needs to deserve more study and analysis. There are some class are easily confused like clockwise motion using single finger and two fingers. Therefore, motion information is not adequate to discriminate some similar gesture, and alternate method should be employed to eliminate this confusion. Our proposed method is better in terms of accuracy, but it contains more hyper parameter to learn because of CNN, LSTM and CTC and therefore training time increases.

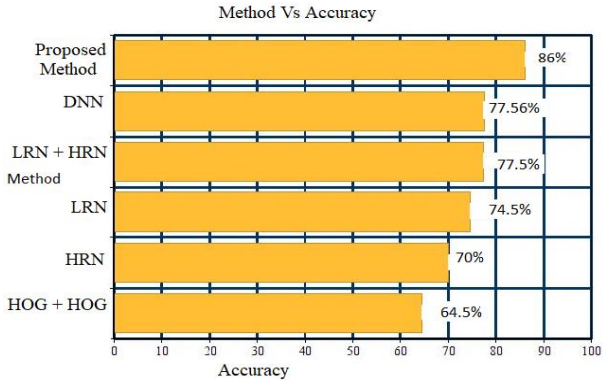

Figure 9. The evaluated results compared with the suggested CTC based model and various existing literature's methods.

5. Conclusions

Our focus on work is improving the accuracy of the dynamic hand gesture recognition method. We have developed an effective two-stream CNN and LSTM recurrent neural networks with CTC based-model to robustly recognize this hand gesture. By the unification process, the input to this proposed model is unique 35-frames. It will maintain a proper route of hand gesture motion and alignment of each label. We have successfully measured an optical flow from each RGB video for eliminating irrelevant object movement information from the context. After that, the deep features vector of RGB and Optical flow stream are extracted and learned using CNN. Next, these features are aggregated and passing through one-by-one to LSTM for temporal recognition. The final output of the LSTM network is connected to CTC Network for proper label alignment using dynamic programming. The entire CTC network is trained through back propagation with the calculation of CTC loss. Among discussed aforementioned some state-of-the-art approaches on the standard VIVA dataset, our proposed CTC based model is appropriate in terms of recognition performance. With the number of empirical evaluations, we observed that the CTC network can align proper frame with appropriate movement information and completely remove post-processing of gesture data. Our proposed method achieves additional progress and goes 86% of recognition accuracy in this paper.

Nevertheless, still, the hand gesture recognition performance is affected by many factors. The motion information is still not adequate for those gestures which have a subtle difference. Therefore, this information is not enough for proper gesture label alignment using the CTC network. Thus, an additional advanced feature should be analysed for this hand gesture challenge. Meanwhile, there is new deep learning architecture, like a deep belief network, viewing the fantastic result of object detection and recognition. Finally, in the future work, to add a very large set of a gesture it can contain more gesture per class and add some new gesture per class for the user interface in vehicles.

References

[1] Alghamdi M., Alwajeeh T., Aljabeer F., Assegaff S., and Budiarto R., "Experimenting Hand-Gesture Image Recognition using Simple Deep Neural Network," *International Journal of Engineering and Technology*, vol. 7, no. 3, pp.103-105, 2018.

[2] Altoff F., Lindl R., and Walchshausl L., "Robust Multimodal Hand And Head Gesture Recognition for Controlling Automotive Infotainment Systems," *VDI-Tagung: DerFahrer im 21*, no. 4, pp. 1-10, 2005.

[3] Belbachir K., and Tlemsani R., "Temporal Neural System Applied to Arabic Online Characters Recognition," *The International Arab Journal of Information Technology*, vol.16, no. 3A, pp.1-19, 2019.

[4] Chai X., Liu Z., Yin F., Liu Z., and Chen X., "Two streams Recurrent Neural Networks for Large-Scale Continuous Gesture Recognition," *in Proceedings of 23rd International Conference on Pattern Recognition (ICPR)*, Cancum, pp. 31-36, 2016.

[5] Chen Z., Zhuang Y., Qian Y., and Yu K., "Phone Synchronous Speech Recognition with CTC Lattices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 90-101, 2017.

[6] Danafar S., and Gheissari N., "Action Recognition for Surveillance Applications Using Optic Flow and SVM," *in Proceedings of Asian Conference on Computer Vision, Springer*, Japan, pp. 457-466, 2007.

[7] El-Alfy E., Baigh Z., and Abdel-Aal R., "A Novel Approach for Face Recognition Using Fused GMDH-Based Networks," *The International Arab Journal of Information Technology*, vol. 15, no. 3, pp. 369-377, 2018.

[8] Farneback G., "Two-Frame Motion Estimation Based on Polynominal Expansion," *in Proceedings of Scandinavian Conference on Image analysis*, Berlin, pp. 363-370, 2003.

[9] Graves A., "Supervised sequence labeling, in Supervised Sequence Labelling with Recurrent Neural Networks," *Springer Berlin Heidelberg*, Berlin, pp. 5-13, 2012.

[10] Hochreiter S., and Schmidhuber J., "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[11] Hu W., Cai M., Chen K., Ding H., Sun L., Liang S., Mo X., and Huo Q., "Sequence Discriminative Training for Offline Handwriting Recognition by an Interpolated CTC and Lattice-Free MMI Objective Function," *in Proceedings of IAPR International Conference on Document Analysis and Recognition*, Kyoto, pp. 61-66, 2017.

[12] Huang D., Fei-Fei L., and Niebles J., "Connectionist Temporal Modeling For Weakly Supervised Action Labeling," *in Proceedings of in European Conference on Computer Vision-Springer*, Amsterdam, pp. 137-153, 2016.

[13] Lin M., Inoue N., and Shinoda K., "CTC Network with Statistical Language Modeling for Action Sequence Recognition in Videos," *in Proceedings of Workshop on ACM Multimedia*, California, pp. 393-401, 2017.

[14] Liu Q., Wang L., and Huo Q., "A Study on Effects of Implicit and Explicit Language Model Information for DBLSTM-CTC Based Handwriting Recognition," *in Proceedings of International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, pp. 461-465, 2015.

[15] Molchanov P., Gupta S., Kim K., and Kautz J., "Hand Gesture Recognition with 3D Convolutional Neural Networks," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Boston, pp. 1-7, 2015.

[16] Molchanov P., Gupta S., Kim K., and Pulli K., "Multi-Sensor System for Driver's Hand-Gesture Recognition," *in Proceedings of 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, pp. 1-8, 2015.

[17] Molchanov P., Yang X., Gupta S., Kim K., Tyree S., and Kautz J., "Online detection and Classification of Dynamic Hand Gestures with Recurrent 3d Convolutional Neural Network," *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, pp. 4207-4215, 2016.

[18] Ohh-Bar E., and Trivedi M., "Hand Gesture Recognition In Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations," *IEEE Trans. on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2368-2377, 2014.

[19] Parada-Loira F., Gonzalez-Agulla E., and Alba-Castro J., "Hand Gestures to Control Infotainment Equipment in Cars," *in Proceedings of IEEE Intelligent Vehicles Symposium*, Dearborn, pp. 1-6, 2014.

[20] Reddy N., Rao M., and Satyanarayana C., "A Novel Face Recognition System by the Combination of Multiple Feature Descriptors," *The International Arab Journal of Information Technology*, vol.16, no. 4, pp. 669-676, 2019.

[21] Rumelhart D., Hinton G., and Williams R., "Learning representations by Backpropagating Errors," *Neurocomputing*, vol. 5, pp. 696-699, 1988.

[22] Tran D., Bourdev L., Fergus R., Torresani L., and Paluri M., "Learning Spatiotemporal Features With 3d Convolutional Networks," *in Proceedings of IEEE International Conference on Computer Vision*, Santiago, pp. 4489-4497, 2015.

[23] Tsironi E., Barros P., Weber C., and Wermeter S., "An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for Gesture Recognition," *Neurocomputing*, vol. 268, pp. 76-86, 2017.

[24] Tu Z., Xei W., Zhang D., Poppe R., Veltkamp R., Li b., and Yuan J., "A Survey of Variational and CNN-based Optical Flow Techniques," *Signal Processing: Image Communication*, vol. 72, pp. 9-24, 2019.

[25] Yi J., Ni H., Wen Z., Liu B., and Tao J., "CTC Regularized Model Adaptation for Improving LSTM RNN Based Multi-Accent Mandarin Speech Recognition," *Journal of Signal Processing Systems*, vol. 90, no. 7, pp. 985-997, 2018.

[26] Zobl M., Nieschulz R., Gieger M., Lang M., and Rigoll G., "Gesture Components for Natural Interaction with in-Car Devices," *in Proceedings of Gesture-Based Communication in Human-Computer Interaction*, Genova, pp. 448-459, 2004.

Sunil Patel is a Research Scholar at the Gujarat Technological University, Ahmedabad. He is currently working as an Assistant Professor in Government Engineering College, Patan, Gujarat, India. He received master's degree from S. P. University, Vallabh Vidyanagar in 2008. He is a Computer Vision researcher and his research interests includes visual representation learning, object recognition, action recognition, video analysis, and deep learning.

Ramji Makwana is a Managing Director of AIIVINE PXL PVT. LTD. He received Ph.D. degree from S. P. University, Vallabh Vidyanagar in 2011. He has authored several papers in major computer vision and multimedia conferences and journals. His research interests include Data mining, Soft computing and deep learning with applications on computer vision tasks, like object recognition, action recognition and Object tracking.