CNN and RNN based Deep Learning Models for Hand Gesture Recognition

PhD Synopsis

Submitted To

Gujarat Technological University

For the Degree

of

Doctor of Philosophy

in

Computer / IT Engineering

By

Sunilkumar Arvindbhai Patel

Enrollment No: 159997107022

Supervisor:

Dr. Ramji M. Makwana

Managing Director,

AIIVINE PXL Pvt. Ltd. Rajkot, Gujarat, India

Table of Contents

1. Title of the Thesis and Abstract

1.1 Title of the Thesis

CNN and RNN based Deep Learning Models for Hand Gesture Recognition

1.2 Abstract

Hand gestures are the most natural, friendly, useful and intuitive non-verbal communication medium while using a computer or machine, and related research efforts have recently boosted interest. It encompasses a wide range of applications in sign language recognition, virtual reality, human-computer interaction, robotics and so on. In the earlier hand recognition task, features were extracted using the hand-crafted technique for improving accuracy. But the challenge remains the same because of the diversity and flexibility of gesture; the small size of a hand, the speed of action is changing, lower recognition time, and gesture similarity. Besides, data generated by current commercial RGB-Depth cameras can be exploited in different gesture-based recognition systems and led to the robustness of gesture. Recently, Researchers are using a deep convolutional neural network-based technique with the combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for the identification of gestures because of robust implicit feature extraction. In the earlier hand gesture recognition task, the input to the system is directly given RGB and Depth information for feature extraction and Classification. As per the literature survey, the exact and smooth motion plays a significant role in the identification and Classification of gestures. In this work, an additional optical flow is calculated for motion estimation from each RGB video. The optical flow stream works for reliable and robust motion calculations. In this work, three different models, 3DCNN, 2DCNN and Connectionist Temporal Classification (CTC) have been proposed with an RGB, Depth and a further optical flow stream use for appearance, Depth and motion-based feature extraction. The 3DCNN-OF with Bidirectional recurrent network model extract space-temporal features from the video after several convolution and pooling. Now, in the second model, This 2DCNN-OF with recurrent network extract space and temporal features sequentially from an image. Third, the CTC network with an optical flow can find proper frame-to-frame alignment and calculate CTC loss. In all the models, three different parallel streams extract RGB, Depth and optical flow features. But the accuracy is improved by applying more weight to an optical flow stream by a weighted fusion scheme. The experimental results and comparisons on VIVA, ChaLearn databases have shown the effectiveness and superiority of the proposed methods for hand gesture recognition and Classification.

2. A brief description of the state of the art of the research topic

Human-generated movements or gestures are commonly applied in real-time applications such as automatic television control [1], robotics, smart surveillance, virtual reality [2], sign language recognition [3], entertainment, smart interface [4][5] etc. Gestures are a continuous movement that can be recognized by the person comfortably. The human gesture involves physical movements of head, hand, arms, fingers etc. In Human-Machine Interaction (HMI), hands are the most intuitive and useful device compared to other human body parts. It has divided into two types of gestures, such as static and dynamic. A static gesture is observed overs fixed period. And, A dynamic gesture is the combination of posture and video sequence concerning time. All the static gesture uses a single frame with limited information, and it has a little computational cost. On the other hand, dynamic gesture contains more meaningful information with greater complexity, and it is in the form of video and more suitable for real-time application. This gesture also required high configured hardware equipment for training and testing.

In the traditional approach, once spatial-temporal hand gesture descriptors have been extracted explicitly, machine learning algorithms are used to perform the recognition process. A large number of features reported in the literature have been manually designed, or "hand-crafted," with an eye for overcoming specific issues like occlusions and variations in scale and illumination. In the hand-crafted technique, the way of feature extraction is problem-oriented. The design of hand-crafted features often involves finding the right trade-off between accuracy and computational efficiency. However, recent advances in terms of data and computational resources with powerful hardware lead to a change of paradigm in computer vision, with the uprising of deep learning. Recently, these deep learning-based models extract useful information implicitly from videos. Three are two way of feature extraction from image/video. 1. Explicit, and 2. Implicit. All the conventional method used a hand-crafted feature extraction or explicit feature extraction like SIFT [6], SURF [7], LBP [8] etc. and finally classified with SVM [9] or HMM [10]. In the hand-crafted feature extraction, Low-level features like appearance, motion, and shape are extracted from an image/video. But now all the deep learning-based model uses an implicit feature extraction. The idea behind the deep learning-based approach is to discover multiple levels of representation so that higher-level features can represent the semantics of the data, which in turn can provide greater robustness to intra-class variability. The major difference is that the features extracted by CNN are learned using the data in contrast to hand-crafted features that are designed beforehand by human experts to extract a given set of chosen characteristics.

2.1 The traditional approach for feature extraction

Recognizing hand gestures with low resolution, speed of action changing by an object, varying length, the involvement of the fingers, similarity of gesture, uncontrolled light environment,

flexibility and diversity in gesture is still a big research issue. A various researcher has presented several papers using hand-crafted feature extraction approach for dynamic hand gesture recognition over the last decades are as follows.

The method describes by M. Zobl. et al. in [4] used segmentation, background subtraction and thresholding operation used for preprocessing an image. Then hu moments as a feature vector used for hand-crafted explicit feature extraction. Finally, the Hidden Markov Model is applied for Classification of a hand gesture. It focuses only on RGB data for feature extraction. In [11] author used robust detection and segmentation by a combination of Edge-based Foreground-Background model (EBM), Mixture of gaussian background model (BG_MOG) and Maximally Stable Extremal Regions segmented (MSER). This silhouette of an object is overlapped and tracked each hand part using the Kalman filter in the successive frames and classified using Hidden Markov Model. Klaser et al. [12] compute 3D gradient orient histogram for spatiotemporal domain feature extraction in the video or image and classified using nonlinear support vector machines for action recognition. It extracts only appearance and depth information from the video. Trivedi et al. [5] described a vision-based system that combines RGB and Depth descriptor to classify hand gesture. In the feature extraction phase histogram of gradients (HOG) is employed. These gestures are classified using a support vector machine. The limitation of work is difficult to recognize similar gestures using only RGB and Depth data. In work represented in [13] utilizes a Bayesian model of visual attention to segment hand region from the background and used SVM classifier to classify hand gestures using the shape and texture features of the hand region against cluttered backgrounds. The main limitation of all the traditional approaches is the way of feature extraction is application-oriented.

2.2 Deep learning-based approach feature extraction

Recently, Deep learning is used for gesture recognition and activity recognition by using automatic implicitly learning sophisticated features using a convolutional neural network (CNN) and recurrent neural network (RNN). For proper handling of the data in the form of video, it is required to extract feature in the spatiotemporal domain.

2.2.1 2D CNN feature extraction

Tsironi et al. [14] have applied CNNLSTM network to analyze and successfully learn complex feature in the varying duration. This network learns temporal feature of each gesture and classifies correctly in Kendon's stroke phase. For the visualization of each feature-map applies a deconvolution process for activation of original image pixels. The work described in [15] is a pure region-based convolutional neural network for segmentation, localization and Classification. The R-CNN technique has applied for proper segmentation of each hand region in the complex background. Then, Each ROI is supplied to the CNN network for feature extraction and Classification. Dadashzadeh et al. [16] use a deep learning-based HGR system with a two-stage CNN architecture. Hand segmentation is the first stage for preprocessing an image. In the second stage two-stream, CNN

network is applied for feature extraction. All the Deep Learning models are using only RGB and Depth information for recognition of hand gesture. And therefore, it does not create a major impact on the results. There is a need to provide some additional stream that can extract robust features and leads to better results.

2.2.2 3D CNN feature extraction

The feature extraction using 3D CNN is also suitable for action recognition. This model simultaneously extracts spatial and temporal feature by performing 3D convolutional and 3D pooling in the multiple adjacent frames [17] [18]. In [19], very deep neural architecture is applied for standard VIVA dataset. Such designed algorithms fail to classify the gesture which appears to be similar in action with minor variation, and recognition accuracy is very minimal as per present real-time requirements. The model in [20] is a more compact model with size only 1 MB for accurate hand gesture recognition using Deep learning-based 3DCNN- LSTM model. Therefore, this model is highly suited for real-time in the mobile device. The combination of LRN (Low-resolution network) and HRN (High-resolution network) is applied for feature extraction and classifying using SoftMax classifier using depth and image gradient data using 3D convolutional neural network [21]. This model has used depth and gradient data for feature extraction and use more hyper parameter for training.

2.2.3 CTC based approach for hand gesture recognition

CTC model has been successfully applied to speech recognition and handwritten character recognition. The combination of R3DCNN-RNN-CTC network extracts spatial-temporal features for dynamic hand gesture recognition. These features are input to a recurrent network, which aggregates transitions across several clips. The output of the recurrent network is connected to the SoftMax layer to estimate class-conditional probabilities and trains with CTC costing function [22]. This network cannot find proper frame alignment because of an insufficient number of frames. CTC model is successfully applied for speech recognition and hand character recognition using CTC loss calculation [23] [24]. CTC and statistical Language models are used for recognizing actions sequence where several actions are concatenated. The Extended Connectionist Temporal Classification (ECTC) framework to efficiently evaluate all possible alignments via dynamic programming and explicitly enforce their consistency with frame-to-frame visual similarities [25]. The future work mentioned in [14] indicates the CTC model can learn proper label alignment for hand gesture recognition and speech recognition. To apply this model for hand gesture recognition that can find frame-to-frame CTC loss and boost accuracy up to some extent.

3. Objective, Scope of the work and problem statement

3.1 The objective of the work: The objective of this work is to develop a hand gesture recognition system to recognize hand gestures from videos with high recognition accuracy. The PhD research work proposes to achieve the following objectives:

- To study and investigate various Deep learning-based models for hand gesture recognition with hand gesture dataset in the form of video.
- To design useful models that combine CNN and RNN techniques and optimize the recognition performance of various hand gesture dataset.
- To develop robust Deep learning-based 2DCNN, 3DCNN and CTC models that recognize hand gestures with improved accuracy. All the models implicitly extract features from a video.
- To evaluate the performance of the proposed model and validate the results with existing state-of-the-art methods.

3.2 The Scope of the work: The Scope of the study involves the following probable improvements in the proposed models for hand gesture recognition.

- The research work presented here is tested for two different kinds of databases which contain videos of complex hand gesture, having varying lighting conditions and low resolution.
- The research work is carried out for video information with RGB and Depth data with minor gesture variation.
- The work is tested for calculated motion estimation as optical flow from RGB data and weighted fusion scheme.
- The CTC network is applied to find proper frame alignment for VIVA and ChaLearn dataset with 19 and 8 different categories of class.

3.3 Problem Definition: Dynamic hand gesture recognition task is very challenging due to various challenges like low resolution, varying illumination, smaller object size, and frequently changing object position. Video is a series of frames, and gesture is a part of this video. The combination of CNN and RNN network is an implicit deep learning-based technique for feature engineering. And then, the problem definition is:

"**To design and develop a deep learning-based model for hand gesture recognition in the video sequences.**"

4. Research Contribution

A hand gesture recognition system must be able to analyze and recognize the universal hand gesture robustly and efficiently in video sequences. The main objective of this research work is to develop an algorithm which recognizes hand gesture under varying illumination and low resolution. The gestures were made by the driver and passenger left or right hand. The gestures are also controlled by hand or finger motion. Also, the system should be such that recognition should be done accurately with improved accuracy. The main contributions of the research divide into four parts. The following subsections brief outline the significant contributions of the Research.

4.1 Unification of frames:

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. It is essential to extract as much as possible those relevant features from every frame. Eventually, after analyzing the entire dataset, convert all of the RGB and Depth videos to the same number of frames for processing.

4.2 Calculation of an optical flow

All the deep learning-based model use RGB and Depth information for input to the system. RGB information gives appearance information, and Depth information gives the distance from the camera. There is a need to add one more stream for motion-based feature extraction. Optical flow gives proper motion path present in the video. As well as remove some irrelevant information present in the background. It also supports the uniqueness of motion-based feature transformation. For getting proper and fitting motion path, optical flow is calculated from each RGB video. Therefore, more attention paid on motion estimation rather than on RGB and Depth information.

4.3 Feature extraction

This step is consisting of extracting the features of hand gesture from preprocessed frames. Gestures are usually presented in the video, and solving gesture recognition depends only on feature extraction. The proposed approach used deep learning for hand gesture recognition by using implicitly automatic learning complex feature using a convolutional neural network (CNN) and recurrent neural network (RNN). Based on the way of feature extraction, three different models proposed.

1. Optical flow-based 2DCNN model
2. Optical flow-based 3DCNN model
3. CTC model with optical flow

4.4 Features fusion

As per the literature, All the conventional method used RGB and Depth information for feature extraction. In these proposed models, use an additional optical flow stream for proper motion estimation. All the features are fused using a weighted fusion scheme. These models give better accuracy when higher weight is given to an optical flow stream.

5. The methodology of Research, Results / Comparisons

This gesture recognition work, pointing to solve video-based dynamic recognition of hand gesture, faces many complexities in the extraction of essential features. If inputs are video instead of images, then this task required more efforts because it requires temporal feature for learning. Appreciation due to the rapid growth of deep learning, it can automatically learn features from the spatial domain and temporal domain at the same time. The entire task for hand gesture recognition as follows:

step-1: Convert all the video to a unique number of the frame for input to the model.

step-2: Calculate Optical flow from each RGB video.

step-3: Feature extraction using 2DCNN, 3DCNN and CTC model

step-4 Merge feature using a weighted fusion scheme

step-5 Train the entire model through Backpropagation with SGD or AdaDelta optimizer.

5.1 Unification of frame

The input to the CNN model requires the same number of frames. It indicates that all video should be in the same dimension, the same height and width of each frame. In this context, all the videos must transform into an equivalent number of frames. The unified value is calculated by using the below Equation.

$$\text{Total sum of frames in the dataset } (T_s) = \sum_{i=0}^{i=N} V_i \quad (1)$$

Where, N represents the total number of video and V_i indicate, i=0 to M frames in each video.

$$\text{Unified value} = \frac{T_s}{N} \quad (2)$$

The unified value is the total sum of frames divided by the total number of videos.

(1) VIVA Dataset

$$\text{Unified value} = \frac{54524}{1460}$$

$$\text{Unified value} = 37.34 \quad (3)$$

(2) ChaLearn Dataset

$$\text{Unified value} = \frac{10942}{322}$$

$$\text{Unified value} = 33.98 \quad (4)$$

The average value for all the frames for VIVA dataset is 37, and ChaLearn dataset is 34. In these proposed methods, the unified value 35 is selected as a standard by near to average using the above formula. The more details of this dataset describe below.

Table 1. Three different sets of categories for this standard VIVA dataset[5].

Three sets	Class category	Total video
Training sample	19	1168
Testing sample	19	146
Validating sample	19	146

Table 2. Three different sets of categories for this standard ChaLearn dataset[26].

Three sets	Class category	Total video
Training sample	8	258
Testing sample	8	32
Validating sample	8	32

5.2 Motion Detection using Optical flow from RGB Video

The proper motion path supports higher recognition accuracy and removes the irrelevant information from the background. Therefore, more attention to be paid on moving parts rather than the fixed

object. Therefore, the optical flow notion is applied to find fitting and precise motion information.[27]. Optical flow is the pattern of apparent motion of image objects, edge, the surface between two consecutive frames. It is a 2D vector field where each Vector is a displacement vector showing the movement of points from the first frame to second.

Assume, a pixel I(x, y, t) in a frame, where t is time. It moves by distance (dx, dy) in the exact next frame after dt time. So, this pixel has the same intensity in the next frame is represented as below.

$$I(x, y, t) = I(x+dx, y+dy, t+dt) \tag{5}$$

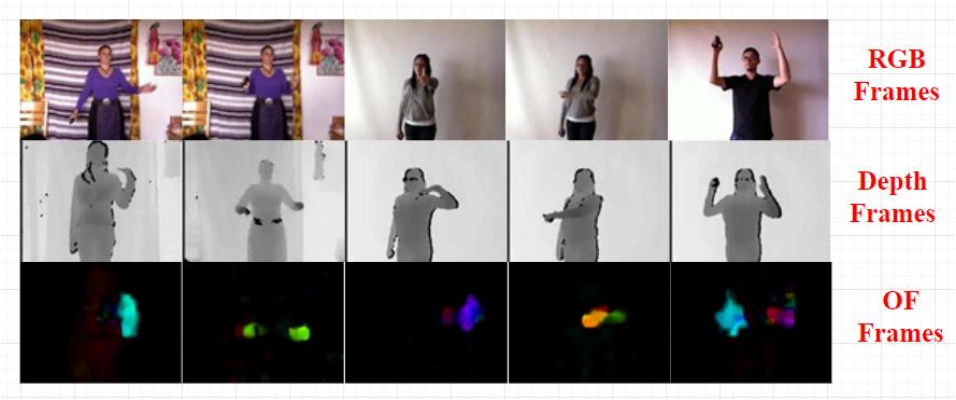

Figure 1. RGB, Depth and optical flow frames for ChaLearn dataset.

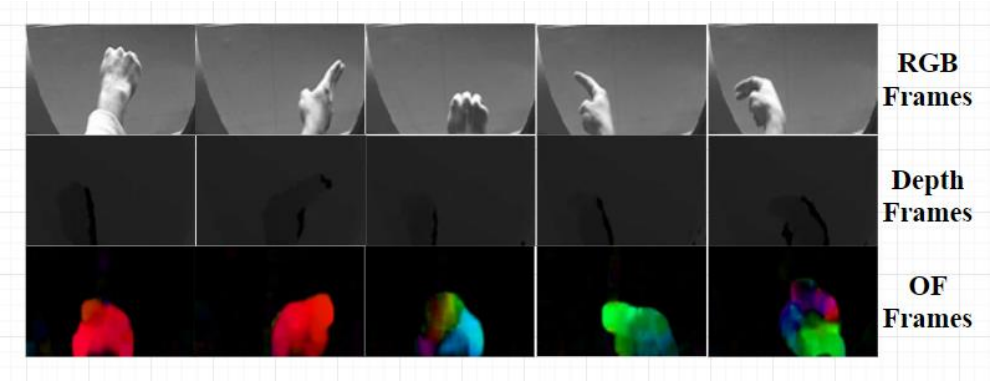

Figure 2. RGB, Depth and optical flow frames for VIVA dataset.

By taking Taylor series approximation on right side, after removing common term and divide by dt final equation is:

$$f_x u + f_y v + f_t = 0 \tag{6}$$

$$\text{where, } f_x = \frac{df}{dx} \text{ and } f_y = \frac{df}{dy} \tag{7}$$

$$u = \frac{dx}{dt} \quad \text{and } v = \frac{dy}{dt} \tag{8}$$

As represented in above for the calculation of (u, v) is an optical flow equation. Where f_x and f_y is gradients of an image and f_t is a gradient along time. The assumption is that all the neighbor pixel have similar motion and we consider 3 X 3 pixel around one patch. A simple solution after using least square fit Method for 9 points and two unknown variables after solving the Equation is as below:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i {f_{x_i}}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i {f_{y_i}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix} \quad (9)$$

Where, the value of (u, v) is optical flow for consequent two frames. The optical flow is calculated for all the frames in this existent RGB dataset and given input to the proposed model, as shown in Figure 1. and 2. It also supports the uniqueness of feature irrespective of objects present in the video.

5.3 Optical flow-based 3DCNN model

This gesture recognition task required many efforts compared to all other work, faces many issues in equalization of frames, extraction of essential features, temporal recognition and Classification.

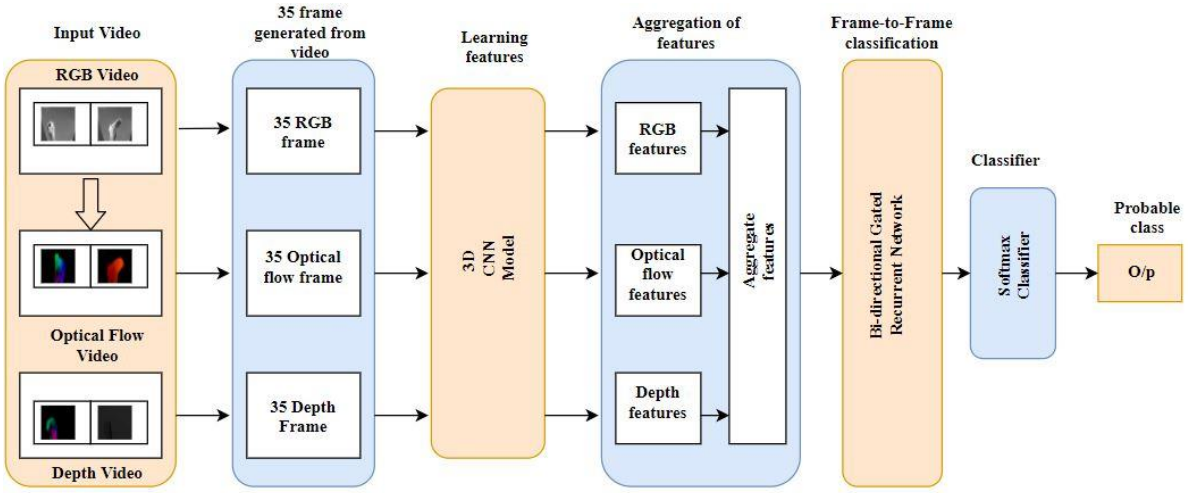

Figure 3. It is an entire pipeline of 3DCNN architecture. It has convolutional layers, fusion schemes, bi-directional recurrent network and SoftMax classifier.

This proposed 3DCNN-Bidirectional GRU with an optical flow-based model can extract spatiotemporal features simultaneously. First of all, all the videos are converted into equal 35-frame for equalization using section 5.1. Then an optical flow is calculated from a subsequent frame for getting proper motion information using section 5.2. The detail working of this model is describing in Figure 3.

5.3.1 3DCNN for spatiotemporal feature extraction

A convolutional neural network is used for the extraction of features from each frame. In the 3DCNN approach size of the kernel is (row, col, depth). In this proposed architecture, as given in Figure 4 and Figure 5. where the size of the kernel is 3 X 3 X 3; therefore, at a time, it can process three frames simultaneously and extract space-temporal features from all the frame. The number of filters used for the hidden layers is 16,32,64,128 and 256, respectively. But for ChaLearn dataset input frame size is 240 X 320 so it has one additional convolution layer with 512 different kernels. There is a batch normalization and ReLu after each convolutional layers. Max pooling 2 X 2 X 2 is performed in each

layer to shrink the size of the feature vector divided by a factor of 2. The Batch normalization is applied for faster convergence of training [28].

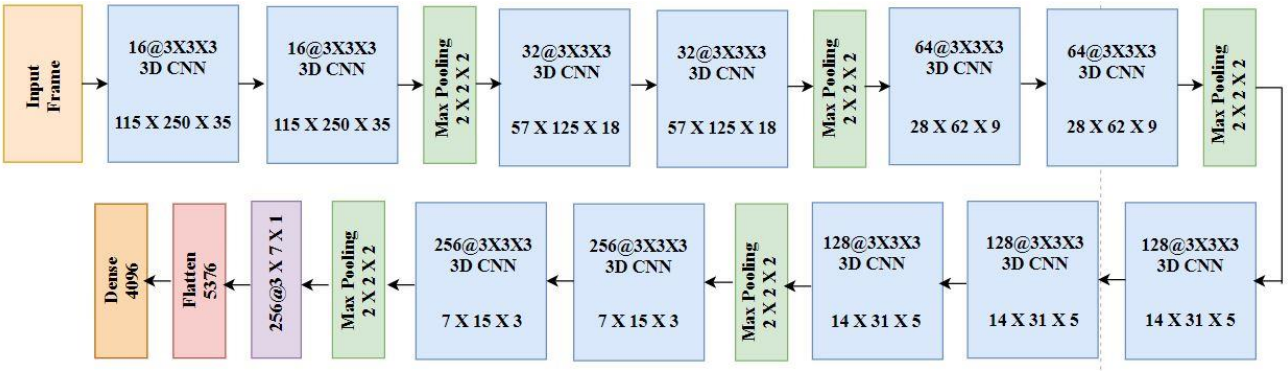

Figure 4. Convolutional Parameter for VIVA dataset.

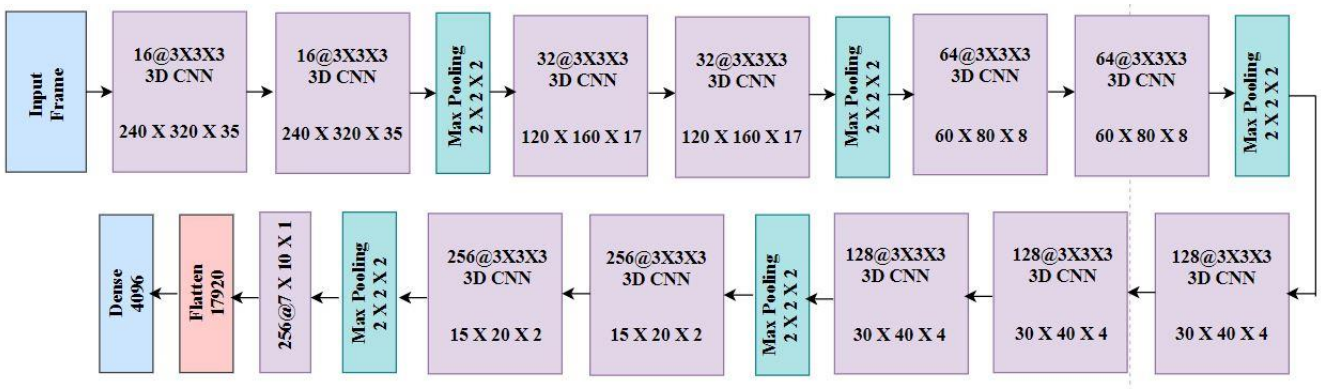

Figure 5. Convolutional parameter for ChaLearn dataset.

5.3.2 Multimodal Fusion scheme

Multimodal fusion is used for features at the different stream at the same timestamp are blended for further work [29].

Table 3. Different fusion scheme for feature integration using Max, Sum, Avg, Concatenation, and Weighted.

Sr NO	Fusion Scheme	Mathematical Representation	Remarks
1	Max fusion	$Y^{max} = f^{max}\{X^a, X^b, X^c\}$ means $Y_{i,j,d}{}^{max} = \max\{x_{i,j,d}{}^a, x_{i,j,d}{}^b, x_{i,j,d}{}^c\}$	X^a = RGB Feature Vector X^b = Depth Feature Vector X^c = Optical Flow Feature Vector Size of Each feature Vector is 4096-dim
2	Sum fusion	$Y^{sum} = f^{sum}\{X^a, X^b, X^c\}$ means $Y_{i,j,d}{}^{sum} = x_{i,j,d}{}^a + x_{i,j,d}{}^b + x_{i,j,d}{}^c$	
3	Avg fusion	$Y^{avg} = f^{avg}\{X^a, X^b, X^c\}$ means $Y_{i,j,d}{}^{avg} = \frac{X^a_{i,j,d} + X^b_{i,j,d} + X^c_{i,j,d}}{3}$	
4	concatenation fusion	$Y^{concat} = f^{concat}\{X^a, X^b, X^c\}$ means $Y_{i,j,2d}{}^{concat} = x_{i,j,d}{}^a$, $Y_{i,j,2d-1}{}^{concat} = x_{i,j,d}{}^b$, $Y_{i,j,2d-2}{}^{concat} = x_{i,j,d}{}^c$	
5	Weighted fusion	$Y^{weighted} = f^{weighted}\{X^a, X^b, X^c\}$ means $Y_{i,j,d}{}^{weighted} = \frac{X^a_{i,j,d} * w1 + X^b_{i,j,d} * w2 + X^c_{i,j,d} * w3}{w1+w2+w3}$ $\sum_{i=1}^{3} w_i = 1$	

This proposed method is evaluated on different fusion scheme and proved that fusion could play a significant role in improving accuracy. Finally, this method itself proved that weighted based fusion improved accuracy to some extent. In late fusion, fusion scheme first extracts the unimodal feature from the different stream like RGB, Depth and Optical flow with a different score and finally this score is integrated for learning. There is various fusion scheme as described in Table 3.

5.3.3 Advance feature extraction using Bidirectional GRU

A Gated recurrent network is used for sequence-to-sequence prediction of the frame. In this proposed method, the bi-directional gated recurrent unit (GRU) is used for higher-level advance feature extraction. The architecture of bidirectional GRU represents in Figure 6. This bidirectional unit can extract feature in both direction left-to-right and right-to-left. In this proposed method, it has been applied two GRU layer for both forward and backward pass. The size of the GRU memory is 64 and 128. This memory cell unit identifies the exact interclass relationship in a single video to a time t. it also recognizes the gesture moving pattern from a sequence of frames.

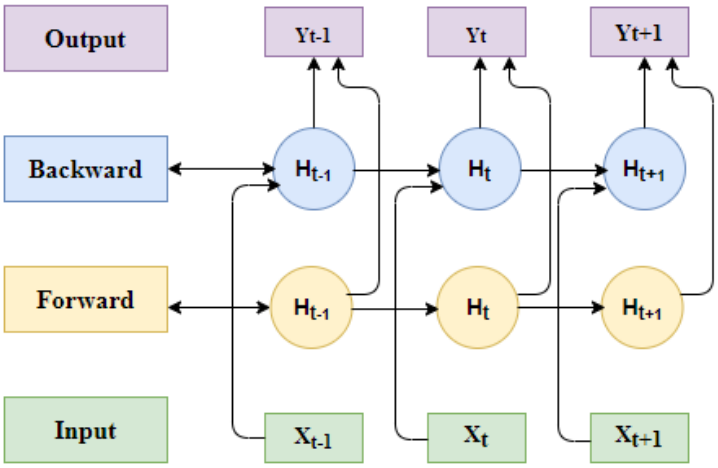

Figure 6. Structure of Bidirectional GRU with forwarding and backward connection for each cell in the network.

5.3.3 Training and Classification

The last part of this network is the Classification and training. The SoftMax activation function, as defined in Equation (10), outputs the probability of each class. The predicted output is the class with the maximum probability.

$$\text{SoftMax}\ (x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \qquad (10)$$

where x_i is the corresponding class, and k is the number of classes. The entire 3DCNN bi-directional recurrent network with an optical flow train through Back Propagation Through Time (BPTT) algorithm. The stochastic gradient descent(SGD) weight optimizer with step decay techniques is applied for VIVA dataset. Step decay schedule drops the learning rate by a factor every ten epochs.

The ChaLearn dataset is trained through AdaDelta optimizer. For training of this proposed model, all the initial weights set to a normal distribution with a standard mean to zero and standard deviation to 0.5. It has been applied 50% dropout after each fully connected layer to avoid overfitting [30].

5.4 Optical flow-based 2DCNN model

This main difference from the previous model is that it uses 2DCNN approach for feature extraction. In the 2DCNN approach size of the kernel is (row, col). Therefore, the implemented 2DCNN-GRU model can extract spatiotemporal features sequentially. The detail working of this model is illustrating in Figure 7.

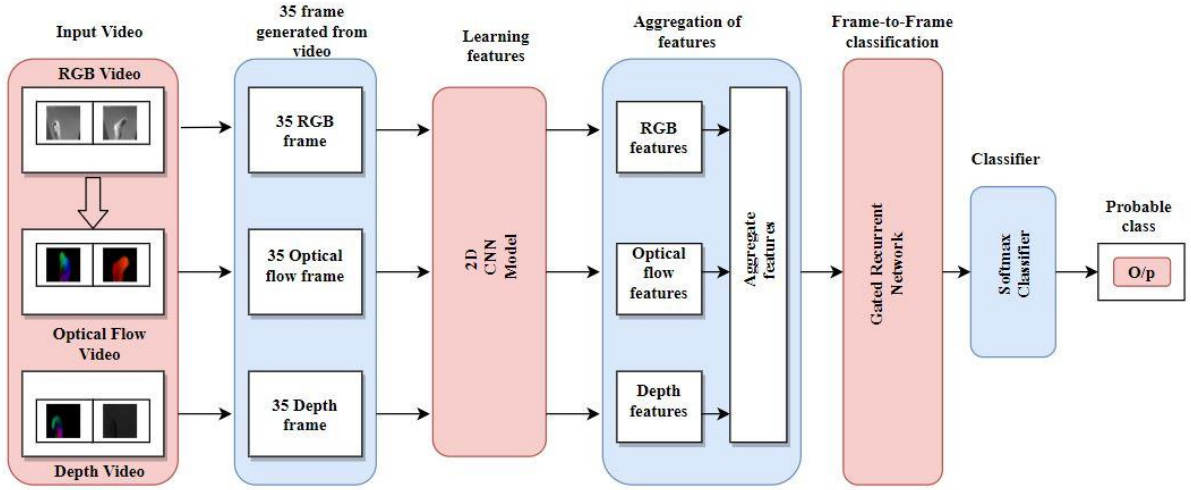

Figure 7. It is an entire pipeline of 2DCNN architecture. It has convolutional layers, fusion schemes, bi-directional recurrent network and SoftMax classifier.

5.4.1 2D CNN approach for feature extraction

Gestures are always part of a video, and solving gesture recognition depends only on feature extraction. In the 2DCNN approach, the model takes a single frame at time t for learning features. In the conventional CNN model, it concerns only appearance information present in a video using RGB, which is not suitable for little interclass variation in a gesture. Hence, to add flow information as an extra feature to support the robustness of the model.

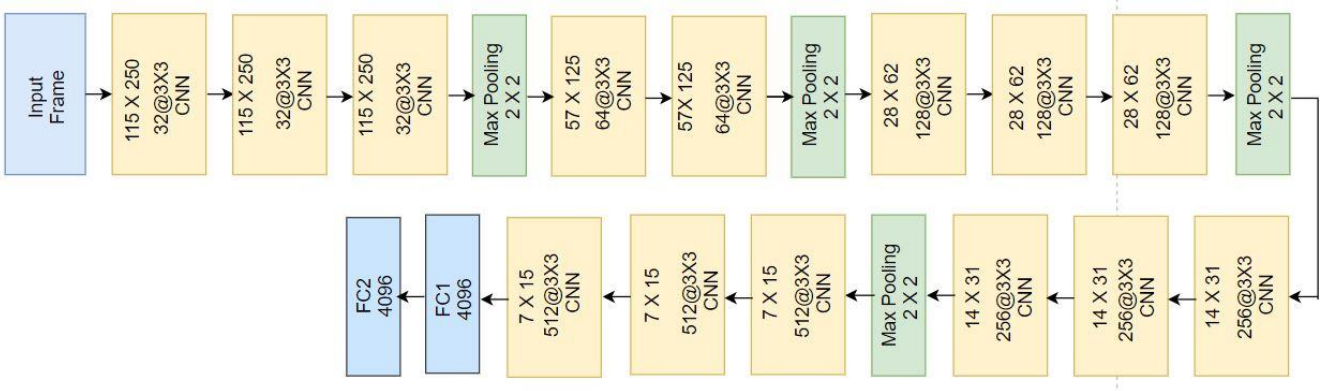

Figure 8. The architecture of 2D CNN model. It involves 14 convolution operations, 4 Maxpooling layers and two fully connected layers, the final size of the feature vector is 4096-dim for VIVA dataset.

The overall proposed CNN architecture for VIVA dataset has five convolution layers, four pooling layer, two fully connected layers, in each stream is illustrates in Figure. 8. And for ChaLearn dataset has six convolution layers, five pooling layer, two fully connected layers, in each stream is illustrates in Figure. 9. Each convolutional layer has a different number of kernels. The first convolutional layer has 16 kernels with a size of 3 X 3-pixel. The second convolutional layer contains 32 kernels with the same number of pixel. The third, fourth and fifth layer has 64,128,256,512 kernels with a size of 3 x 3. The final output of fully connected layer fc6 and fc7 is 4096-dim. The size of Max-pooling is 2 X 2, and it is used to reduce the size of a factor of 2. In this model, the Rectifier Linear Unit (ReLu) and batch normalization are used for faster training[28].

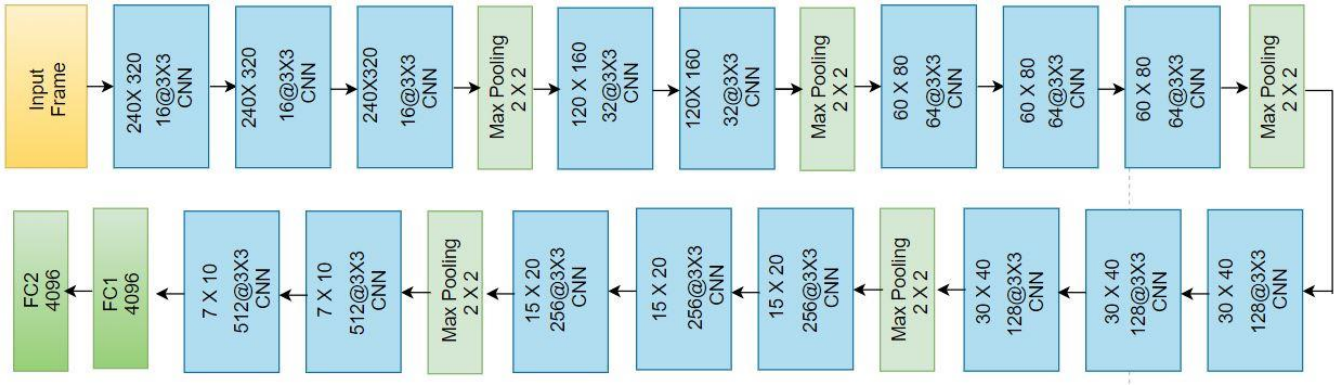

Figure 9. The architecture of 2D CNN model. It involves 16 convolution operations, 5 Maxpooling layers and two fully connected layers, the final size of the feature vector is 4096-dim for ChaLearn dataset.

5.4.2 Aggregation of features

The intention here is to fuse these three streams such that pixel response at the same spatial location is fused. From these three streams, to get appearance information of hand from RGB, sharp edge feature from the Depth and find motion from left-to-right, up-down using optical flow.

$$Y_{i,\,j,\,d}{}^{\text{weighted}} = \frac{X^{a}_{i,j,d} * w1 + X^{b}_{i,j,d} * w2 + + X^{c}_{i,j,d} * w3}{w1+w2+w3} \qquad \sum_{i=1}^{3} w_i = 1 \tag{11}$$

This weighted fusion function aggregates all the obtained features after CNN, as illustrates in Equation (11). The size of each X^a, X^b, X^c vector is a 4096-dim Vector. Where X^a represents RGB vector, X^b represents Depth vector, and X^c represents an optical flow vector. After weighted fusion, the final value is 4096-dim Vector. And it is followed by the gated recurrent network. The adding of a new weighted average scheme for the fusion of feature because it gives more chance to an optical flow stream which is appropriately related to the Classification of gesture those has little interclass variation.

5.4.3 Temporal recognition using Gated Recurrent unit

The input of this layer is fusion features with size 4096-dim after the convolution and pooling. This gated recurrent network extracts temporal information from a series of frames. It is a many-to-one recurrent network as describes in Figure 10. At time t=0, it fetches fist frame, and t=1 it fetches the second frame and merges with the previous frame. And subsequently, it extracts remaining frames up

to t=35 and aggregates all the information. The SoftMax layer is on last GRU cell, and it generates probability value for each class. The memory block inside the GRU network has a different gate that can store information. In this architecture, two different recurrent layers are applied with N=35 different memory cell. The size of each cell is 32 and 64 for two different layers. On the first layer, it has a many-to-many connection, and on the second layer, it has a many-to-one connection.

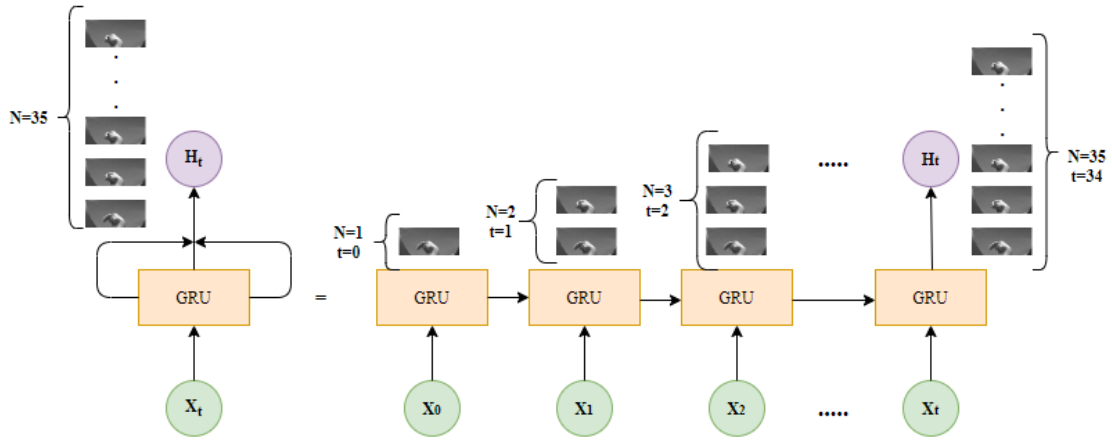

Figure 10. Unrolling of Gated Recurrent Network. Each aggregated feature is passing one-by-one to this unit for temporal recognition to gather past and present information.

5.4.4 Classification and training

The last part of this network is training and Classification, which converts video level representation of temporal frame to many-to-one network and generate probability score as illustrates in Equation (12).

$$\text{SoftMax}\ (x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \tag{12}$$

where x_i is the corresponding class, and k is the number of classes. The entire 2DCNN and gated recurrent network with an optical flow train through Back Propagation Through Time (BPTT) algorithm. The stochastic gradient descent (SGD) weight optimizer with step decay techniques is applied for VIVA dataset. The ChaLearn dataset is trained through AdaDelta optimizer. For training of this proposed model, all the initial weights set to a normal distribution with a standard mean to zero and standard deviation to 0.5. To avoid overfitting and enhance the model generalization on test data, it has been applied 50% dropout after each fully connected layer [30].

5.5 CTC model with an optical flow

The proposed hand gesture recognition-based CTC model with an optical flow uses deep learning for the extraction of essential features. The CTC network initially applied only to handwritten character recognition and speech recognition for proper label alignment. In this proposed approach, the CTC model is successfully applied to hand gesture recognition, and it can identify frames that are part of the relevant gesture class. In this framework, the proposed CTC model with an optical

flow is trained for spatiotemporal feature extraction and find the proper label aligned frames. The CTC network can find a proper gesture relevant frame and CTC decoding generate gesture sequence. The detail working of this process describes in Figure 11.

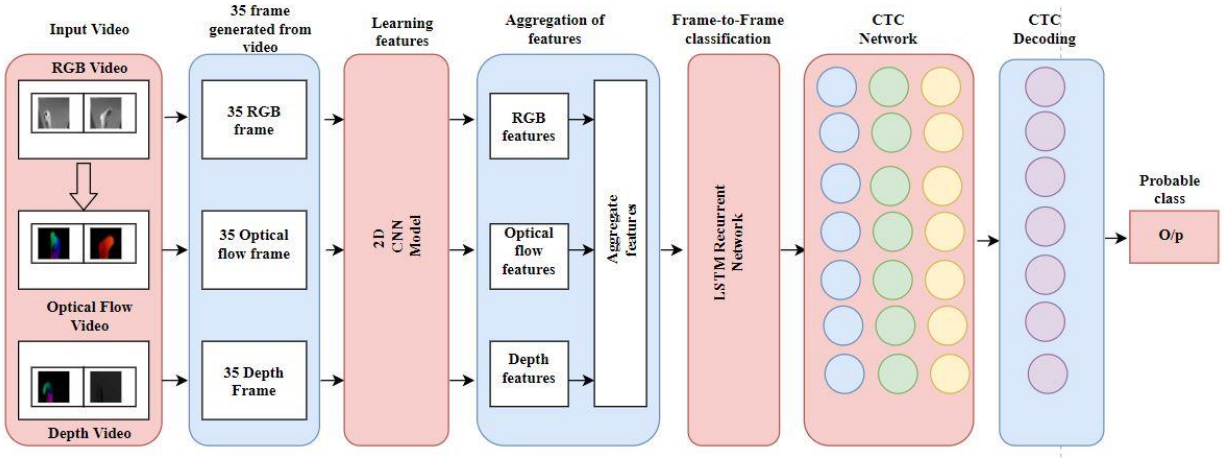

Figure 11. This is the broad architecture of this proposed model. It has preprocessing step, convolutional layer, fusion layer, recurrent layer, CTC Network layer and CTC decoding process.

5.5.1 Feature extraction using CNN and LSTM

The CNN and LSTM models are used for implicitly constant and robust features learning. The pipeline of this proposed architecture represents in Figure 12. It has a similar parameter to describe in 2DCNN model in section **5.4.1**. In this proposed architecture, it has one hidden layer with N number of LSTM nodes, where N=35. The input to the LSTM node is a 4096-dim feature vector produced after CNN. Each timestamp (t=0 to N) can generate a probability score for 19 different classes. The final score after LSTM is (19,35) probability value which is input to the CTC network. The size of each memory cell is 32 and 64 in each layer. It has a many-to-many connection in both layers, as described in Figure 13.

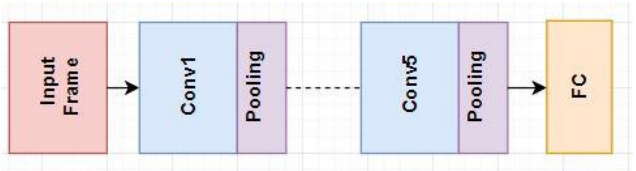

Figure 12. This is a pipeline of the CNN model with the convolutional and pooling operation

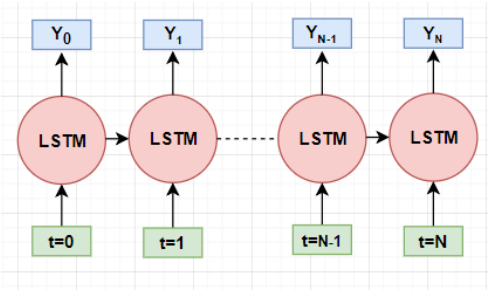

Figure 13. It is an LSTM architecture with 35 node, and the final output is (19,35).

5.5.2 CTC for label alignment and calculation of loss

It is a supervised learning approach and used to train Recurrent Networks with Connectionist Temporal Classification (CTC) costing parametric function to assign a unique tag to not properly

segmented input sequence data [31]. CTC network can transform the output into a conditional probability distribution over the gesture label sequence for a single class. The network then identifies the most probable frame to ground truth sequence. As shown in Figure 14, for example, the ground truth sequence is [17,18,1,19,2] 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 backpropagation after calculating CTC loss. This video clip can contain only gesture information, not irrelevant information, so it does not include "no gesture" class. In this work, the CTC forward and backward algorithm have been considered for calculating loss and gradient.

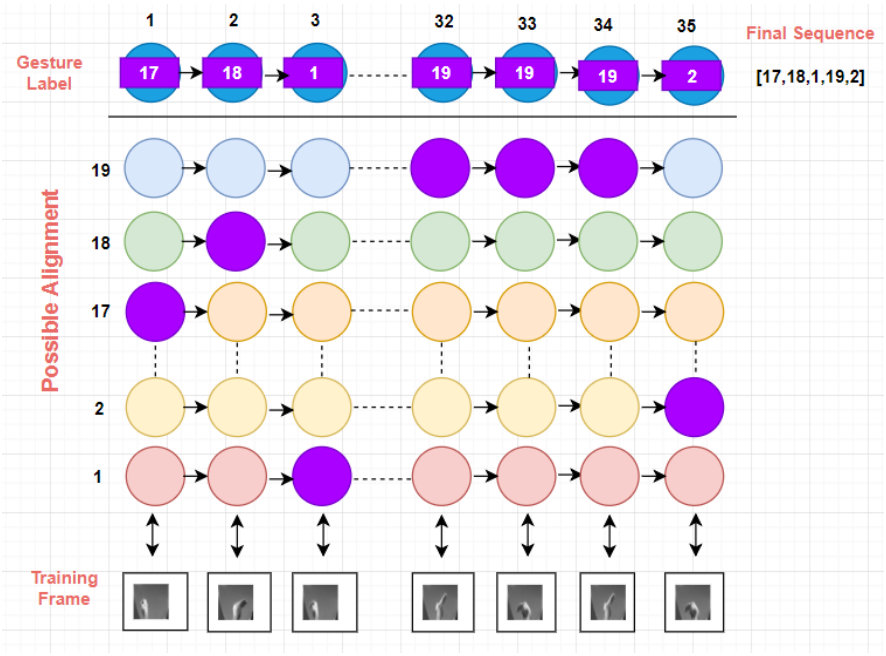

Figure 14**.** CTC Network with training frame and gesture labels. Ground truth and probable selection of gesture label describe in purple colour.

The output of the LSTM recurrent network is connected to the CTC network. In this case output of the LSTM recurrent layer is (z, Y) where z= 19(gesture label) and Y = 35(total input frame). The video output of LSTM for each activation frame with different timestamp is Y = (Y_1, Y_2, Y_3…. Y_T) and probable gesture sequence is z= (l_1, l_2, $l_{3...}l_N$)., here T > N means number of activations is larger than the gesture label sequence. CTC states the probability of input frame Y to gesture label sequence L is:

$$P\left(\frac{z}{Y}\right)=\sum_{\Pi\in\beta^{-1}(z)} P\left(\frac{\pi}{Y}\right) \quad (13)$$

Where, $\pi = \{\pi^1, \pi^2, \pi^3 \dots \pi^{T}\}$ is a path denoting gesture sequence label along with input Y, and β is an operator to eliminate consecutively repeatedly the 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 y^t in Y gives a contribution to the generation of the different path by aligning each label in z through β. Then, CTC assumes that each π^t where π is conditionally independent given input Y. the probability of each gesture path is given by:

$$P\left(\frac{\pi}{Y}\right) = \prod_{t=1}^{T} P\left(\frac{\pi^t}{Y}\right) \quad (14)$$

The value of $P\left(\frac{\pi^t}{Y}\right)$ Calculates using the output of the LSTM function at a timestamp t. It can map y^t to the probability of gesture classes via linear and a SoftMax output Layer.

The complexity of P($\frac{z}{Y}$) increases exponentially concerning the given each input frame Y. It is vital to try a straightforward approach and compute the score for each gesture alignment and summing all the gesture sequence. The calculation the CTC loss by using the Equation (13) is costly to compute because of a massive number of alignments between a gesture sequence. Therefore, It is efficiently calculated the loss by using dynamic programming. Let π_1^t represent a partial gesture label sequence different π path from 1 to t, from that z_1^k refers to a series of gesture labels consisting of the first k label sequence from z. The value of $\alpha_t(k)$, which is a summation of the probability of π_1^t that aligns to z_1^k is.

$$\alpha_t(k) = \sum_{\pi \in \{\pi:\, \beta(\pi_1^t) = z_1^k\}} P\left(\frac{\pi_1^t}{Y}\right) \quad (15)$$

The value of $\alpha_t(k)$ recursively as below:

$$\alpha_t(k) = [\alpha_{t-1}(k) + \alpha_{t-1(k-1)}]\, s^t(z^k) \quad (16)$$

Where the emitting probability of $s^t(z^k)$ is produced by the gesture label z^k at timestamp t. That means, $s^t(z^k) = P\left(\frac{\pi^t}{Y}\right)$ have all the possible path π, and it has gesture label z^k at timestamp t. As per defined in Equation (16), the mapping of different input frame Y to ground truth gesture label z 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. By deriving $\alpha_t(k)$ in linear time, the value of P ($\frac{z}{Y}$) measures by using dynamic programming for the input of the video. CTC defines the loss function as a negative log-likelihood for the LSTM recurrent neural network as:

$$L = -\log P\left(\frac{z}{Y}\right) \quad (17)$$

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

$$\beta_t(k) = [\beta_t(k) + \beta_t{+}1(k{+}1)]\, s^t(z^k) \quad (18)$$

Where all the variable describes in Equation (18) is the same as in Equation (16). The value of $\beta_t(k)$ determines the mapping of the specific input frame Y to gesture label z from the end of the input frame Y to the timestamp t and k^{th} gesture label. The gesture label z does not start from the beginning

as we measure by using this $\alpha_t(k)$. By calculating $\alpha_t(k)$ and $\beta_t(k)$, the total loss is calculated by Equation (17) as described in the preceding part, and concerning the output of $s^t(m)$, this recurrent neural network has SoftMax output for the m^{th} label class at a time t. By calculating $\alpha_t(k)$, $\beta_t(k)$, The total CTC loss is as below:

$$L = -\log P\left(\frac{z}{Y}\right) = \sum_{s=1}^{S} \frac{\alpha_t(k) * \beta_t(k)}{s^t(m)} \quad (19)$$

So, this entire CTC network is trained using stochastic gradient descent with backpropagation through time t and with the support of CTC loss.

CTC Decoding

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

$$z = \text{argument} \max_{z} P\left(\frac{z}{Y}\right) \quad (20)$$

$$z = \beta\left(\text{argument} \max_{\pi} P\left(\frac{\pi}{Y}\right)\right) \quad (21)$$

The value of $P\left(\frac{z}{Y}\right)$ calculates by using $P\left(\frac{\pi}{Y}\right)$ Where the most probable path is π among the gesture label z, in the max encoding method, the path with the highest probability value at each timestamp is considered and concatenate them with the best path. It can apply β(.) operator to remove and concatenates them, which has the same gesture label.

5.5.3 Classification and training

The last part of this network is training and Classification. The entire CTC network train through backpropagation through time (BPTT) algorithm with the calculation of CTC loss. The entire CTC loss is calculated by giving Equation (19) in section 5.5.2. The stochastic gradient descent(SGD) weight optimizer with step decay techniques is applied for VIVA dataset. The ChaLearn dataset is trained through AdaDelta optimizer. For training of this proposed model, all the initial weights set to a normal distribution with a standard mean to zero and standard deviation to 0.5.

6. Results and Discussion

In this section, these proposed models have been validated by a series of experiments. In this first sub-section confusion matrix is created concerning to all models for VIVA and ChaLearn dataset. Next, the impact of optical is checked by the performance of the models. Finally, the results are compared by all the conventional method is applied to this dataset.

6.1 Results comparison of all proposed models

Confusion Matrix is a performance measurement for machine learning classification. The confusion matrix is generated from a 10% sample taken from the VIVA and ChaLearn dataset. The diagonal matrix elements show the number of times the class is correctly identified and represent the accuracy of the models. The precision measured: What proportion of positive identifications was actually correct? And the recall measured: What proportion of actual positives was identified correctly? Here

Table 4. Shows the performance measurement for all the proposed model with different metric calculated from the confusion matrix. The highest accuracy of 86% is achieved for the VIVA dataset on the CTC model, for ChaLearn Dataset 2DCNN model returned 88% accuracy. The misclassification rate 21%, which is highest of ChaLearn dataset on 3DCNN model. CTC model which has the lowest misclassification rate of 14 % for VIVA dataset. According to the results, 2DCNN model returns better accuracy than the CTC model for ChaLearn dataset because it can find a proper motion path compared to proper alignment of frames in the CTC model.

Table 4. It contains accuracy, precision, recall and F1-score for all the models.

Sr.No	Model	Dataset	Accuracy	Precision	Recall	F1-Score	Misclassification Rate
1	3DCNN+OF	VIVA	80.5%	80.26%	80.70%	80.20%	19.5%
2	3DCNN+OF	ChaLearn	79%	78.47%	78.96%	78.99%	21%
3	2DCNN+OF	VIVA	85%	84.87%	84.32%	84.86%	15%
4	2DCNN+OF	ChaLearn	88%	87.50%	89.38%	87.35%	12%
5	CTC+OF	VIVA	86%	86.18%	86.62%	86.29%	14%
6	CTC+OF	ChaLearn	84%	84.38%	84.38%	84.38%	16%

6.2 Comparison of proposed models with RGB + Depth + OF feature extraction streams

The optical flow supports strong-motion present in the video. It is calculated from RGB video. The impact of this optical flow is tested on three proposed models as below.

6.2.1 3DCNN model with optical flow and without optical flow

Here the graph shows the performance of RGB + Depth and RGB + Depth + OF for 3DCNN model in Figure 15. (VIVA dataset) and Figure 16. (ChaLearn dataset). The results show that performance is 80.5% and 79 % for VIVA and ChaLearn dataset with optical flow. It is evident that optical flow-based features result in better accuracy.

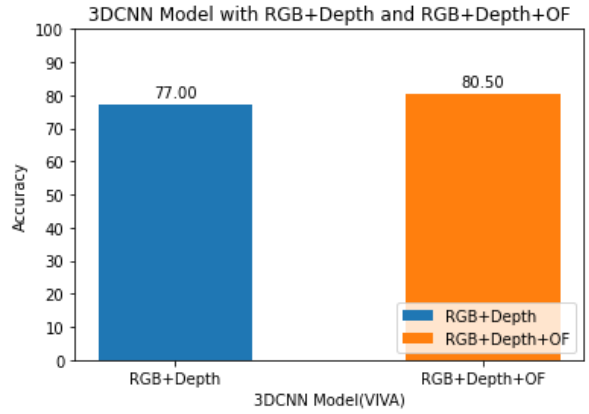

Figure15. Performance comparison with 3DCNN Model with VIVA dataset.

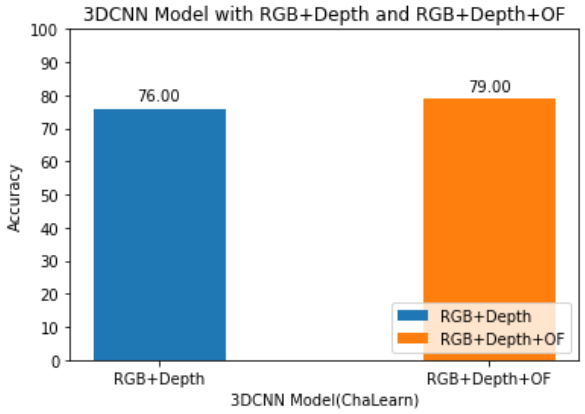

Figure 16. Performance comparison with 3DCNN Model with ChaLearn dataset.

6.2.2 2DCNN model with optical flow and without optical flow

Here the graph shows the performance of RGB + Depth and RGB + Depth + OF for 2DCNN model in Figure 17. (VIVA dataset) and Figure 18. (ChaLearn dataset). The Figure shows that, with optical flow, performance is 85% and 88 % for VIVA and ChaLearn dataset respectively. The result proves that, the optical flow has a significant role in boosting the accuracy of models.

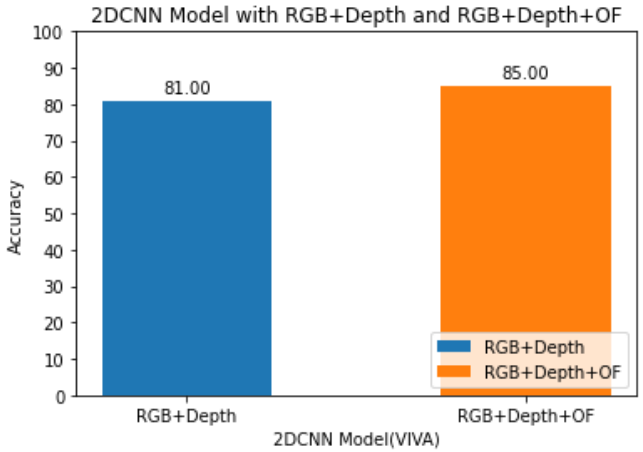

Figure 17. Performance comparison with 2DCNN-GRU Model with VIVA dataset.

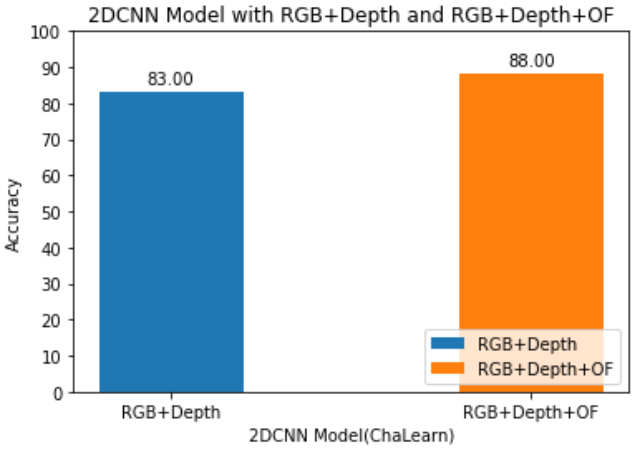

Figure 18. Performance comparison with 2DCNN-GRU Model with ChaLearn dataset.

6.2.3 CTC model with optical flow and without optical flow

Here the graph shows the performance of RGB + Depth and RGB + Depth + OF for CTC model in Figure 19. (VIVA dataset) and Figure 20. (ChaLearn dataset). It is observed that the accuracy is 86% and 84% for VIVA and ChaLearn dataset respectively, with optical flow. This model has higher accuracy compared to 2DCNN and 3DCNN models for VIVA dataset. But It shows lower accuracy to 2DCNN model and higher accuracy to 3DCNN model because it can not find proper alignment of frames. This proposed CTC model can find better frame alignment and learn better spatial and temporal features with optical flow.

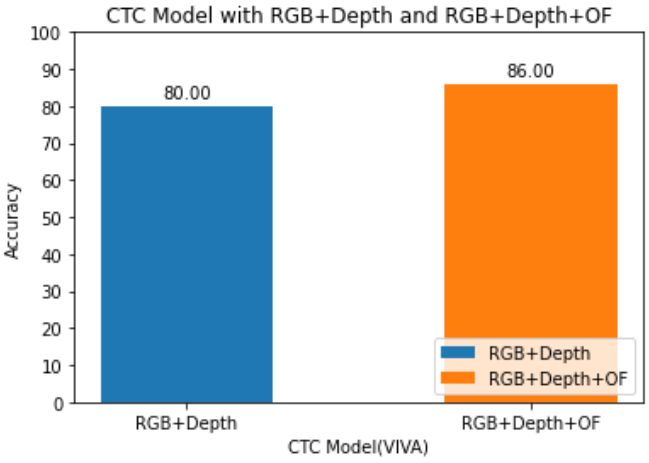

Figure 19. Performance comparison with CTC Model with VIVA dataset.

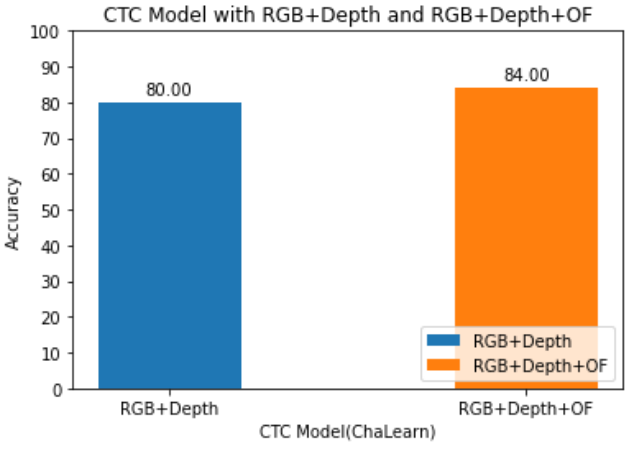

Figure 20. Performance comparison with CTC Model with ChaLearn dataset.

6.3 Comparison with the Proposed Methods and existing Methods

In this part, the classification rate for proposed methods and the existing methods are described here. The performance is also compared with the various existing implicit and explicit feature extraction approaches.

6.3.1 State-of-the-art comparison with proposed methods for VIVA dataset

This section compared proposed 3DCNN + OF, 2DCNN + OF and CTC + OF approach with existing HOG+HOG explicit feature extraction approach by Trivedi et al. [5], the LRN, HRN and, the combination of LRN + HRN model with implicit feature extraction approach by P. Molchanov et al. [21], deep neural network-based model and implicit feature extraction are used by Mostafa Alghamdi et al. [19]. It is observed that the accuracy for 3DCNN + OF, 2DCNN + OF and CTC + OF models are 80.5%, 85%, 86% respectively. It is evident that the proposed models achieved superior accuracy. These experimental results prove that for dynamically hand gesture task, in the car, the outstanding recognition efficiency achieved for all the models. These models are performing significantly better than the existing approaches mentioned in the literature.

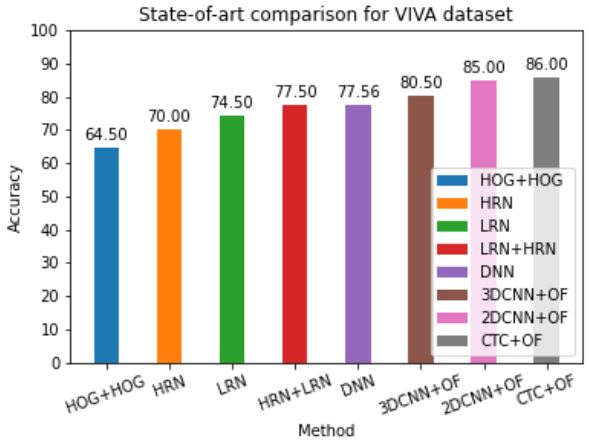

Figure 21. The evaluated results compared with the suggested models and various existing literature's methods.

6.3.2 Comparison with proposed methods for ChaLearn dataset

The comparison of these proposed methods describes in Figure 22. It gives better accuracy for 2DCNN and CTC model but gives lower accuracy for 3DCNN model. It is observed that the accuracy for 3DCNN + OF, 2DCNN + OF and CTC + OF models are 79%, 88%, 84% respectively. From results, it is apparent that the performance is increased due to optical flow. The results indicate that the 2DCNN model performs better than the CTC model. This CTC model cannot find proper frame alignment for the gesture, but 2DCNN model can find proper motion fitting path with weighted fusion scheme. The main thing is that the full object is visible in this dataset. But it returns flow irrespective to the object present in the video.

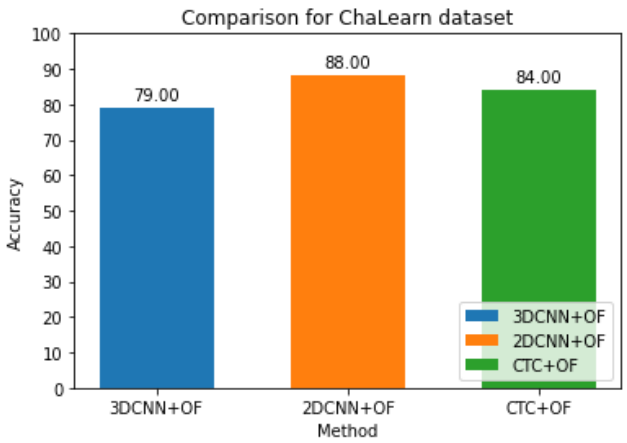

Figure 22. Comparison of ChaLearn dataset.

7. Achievements concerning objectives

The objective of this research work was to devise a deep learning-based method for hand gesture recognition which should work on differently illuminated videos, varying illumination, captured in a controlled environment as well as in real-time and without any human intervention.

Three different methods have been proposed to achieve the stated objectives. All proposed models automatically learn implicit features using 2DCNN, and 3DCNN approach. It can find enough information from the video using the unification strategy. This motion-based approach worked effectively and achieved a better result for all the methods. The proposed CTC method found proper label alignment and improved accuracy. Challenging gestures are strictly recognized with the use of optical flow.

8. Conclusion

Various models have been investigated for dynamic hand gesture recognition using RGB and Depth information. In this research work, the main focus on work is improving the accuracy of the dynamic hand gesture recognition task using various deep learning-based approach on low resolution and varying illumination video captured under a controlled environment. Existing models are improved by using motion-based feature extraction, and deep learning-based algorithms have been presented. Three different 2DCNN, 3DCNN and CTC based models have been presented using optical flow. For capturing enough information from video, unification strategy is applied. Therefore, all the videos are converted into 35 unique figures for standard feature extraction. This unification strategy extracts collective and important information present in the video from enough number of frames. The adding of an optical flow can find proper and fitting motion path from the video irrespective to the object present in the video. It has been calculated from each RGB video. The proposed 2DCNN, 3DCNN and CTC model with an optical flow learned appropriate feature from three different parallel streams. The weighted based fusion is appropriate for motion-based feature extraction irrespective to the object present in the video. The work is tested on two datasets, VIVA and ChaLearn.

In the proposed 3DCNN with an optical flow-based model extracts Spatial-temporal feature

simultaneously with 3D convolutional and pooling with three different parallel RGB, Depth and Optical flow streams. The late fusion technique integrated unique features from each stream and merged on the last layer. It will be helpful to improve the unique feature extraction process as well as the performance of hand gesture recognition using a weighted fusion scheme. In the weighted scheme, more weight is assigned to an optical flow stream for motion-based feature extraction. The recognition accuracy for VIVA and ChaLearn dataset is 80.5% and 79%, respectively. Obtained experimental results show improvement in recognition accuracy compared with the existing state-of-the-art methods with this approach.

In the proposed 2DCNN with an optical flow-based approach, spatial and temporal feature extracts sequentially using Convolutional and recurrent neural network through three different RGB, Depth and optical flow stream. The proposed model extracts spatial features from a single frame using 2D convolutional and pooling and weighted fusion scheme. After that, the recurrent neural network finds proper and exact motion path with merging present and past information instead of only appearance and depth information. This fitting motion path accurately recognized those gestures which as similar action with minor variation and provide robust and superior performance. The recognition accuracy for VIVA and ChaLearn dataset is 85% and 88%, respectively. Obtained experimental results show improvement in recognition accuracy compared with the existing state-of-the-art methods with this approach.

In the proposed CTC model can find proper frame alignment on RGB, Depth and Optical flow data on three parallel streams after feature extractions and weighted fusion. Initially, it is applied to handwritten character recognition and speech recognition. But in this work successfully applied for hand gesture recognition. The CTC costing function is applied for training the entire network with 35-unique frames. This sufficient frames accurately major CTC loss regardless of the CNN used. The recognition accuracy for VIVA and ChaLearn dataset is 86% and 84%, respectively. Overall result analysis shows that CTC model outperforms well on VIVA dataset compared to other models. but bit confuse on ChaLearn dataset for finding proper frame alignment. The results also indicate that 2DCNN model perform better on ChaLearn dataset compared to other models. Because it finds proper fitting motion path on ChaLearn dataset irrespective to the object present in the video. This proposed 3DCNN, 2DCNN, and CTC models with an optical flow-based approach on VIVA and ChaLearn dataset are proper and accurate for hand gesture recognition task using motion-based feature extraction and weighted based fusion.

9. Research Publications

1. Sunil A. Patel and Ramji M. Makwana, “Connectionist Temporal Classification Model for Dynamic Hand Gesture Recognition using RGB and Optical flow Data”, *The International Arab*

Journal of Information Technology, ISSN:1683-3198, Vol-17, No-4, 2020, pp.497-506(SCIE Indexed).

2. Sunil A. Patel and Ramji M. Makwana," Dynamic Hand Gesture Recognition through three stream 3DCNN and Bidirectional gated recurrent unit using multimodal fusion", *International journal of Imaging and Robotics*, ISSN:0974-0627, Vol-19, No-4,2019, pp.20-32(UGC Approved).

3. Sunil A. Patel and Ramji M. Makwana, "2D CNN and Gated Recurrent Network for Dynamic Hand Gesture Recognition with A Fusion of RGB-D and Optical Flow Data"," *International Journal of Innovative Technology and Exploring Engineering*", ISSN:2278-3075, Vol-8, No-10, 2019, pp.1784-1792(Scopus Indexed).

References

[1] S. Lian, W. Hu, and K. Wang, "Automatic user state recognition for hand gesture based low-cost television control system," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 107–115, 2014.

[2] C. Wang, Z. Liu, and S. C. Chan, "Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera," *IEEE Trans. Multimed.*, vol. 17, no. 1, pp. 29–39, 2015.

[3] G. A. Rao and P. V. V. Kishore, "Selfie video based continuous Indian sign language recognition system," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 1929–1939, 2018.

[4] M. Zobl, R. Nieschulz, M. Geiger, M. Lang, and G. Rigoll, "Gesture components for natural interaction with in-car devices," *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.*, vol. 2915, pp. 448–459, 2004.

[5] M. M. Trivedi and E. Ohn-Bar, "Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, 2014.

[6] L. D. G, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, 2004.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.

[8] M. PietikÃ¤inen, "Local Binary Patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

[9] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001, vol. 2049 LNAI, pp. 249–257.

[10] B. Schuster-Böckler and A. Bateman, "An Introduction to Hidden Markov Models," in *Current Protocols in Bioinformatics*, 2007.

[11] F. Parada-Loira, E. Gonzalez-Agulla, and J. L. Alba-Castro, "Hand gestures to control infotainment equipment in cars," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2014, pp. 1–6.

[12] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *BMVC 2008 - Proceedings of the British Machine Vision Conference 2008*, 2008, pp. 1–8.

[13] P. K. Pisharady, P. Vadakkepat, and A. P. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 403–419, 2013.

[14] E. Tsironi, P. Barros, C. Weber, and S. Wermter, "An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76–86, 2017.

[15] J. O. Pinzon Arenas, R. J. Moreno, and P. C. U. Murillo, "Hand gesture recognition by means of region-based convolutional neural networks," *Contemp. Eng. Sci.*, vol. 10, no. 27, pp. 1329–1342, 2017.

[16] A. Dadashzadeh, A. T. Targhi, M. Tahmasbi, and M. Mirmehdi, "HGR-Net: A fusion network for hand gesture segmentation and recognition," *IET Comput. Vis.*, vol. 13, no. 8, pp. 700–707, 2019.

[17] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2013.

[18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 4489–4497.

[19] M. Alghamdi, T. Alwajeeh, F. Aljabeer, S. Assegaff, and R. Budiarto, "Experimenting hand-gesture image recognition using simple deep neural network," *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 103–105, 2018.

[20] K. Mullick and A. M. Namboodiri, "Learning deep and compact models for gesture recognition," in *Proceedings - International Conference on Image Processing, ICIP*, 2018, vol. 2017-Septe, pp. 3998–4002.

[21] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2015, vol. 2015-Octob, pp. 1–7.

[22] 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,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 4207–4215.

[23] J. Yi, Z. Wen, J. Tao, H. Ni, and B. Liu, “CTC Regularized Model Adaptation for Improving LSTM RNN Based Multi-Accent Mandarin Speech Recognition,” *J. Signal Process. Syst.*, vol. 90, no. 7, pp. 985–997, 2018.

[24] W. Hu *et al.*, “Sequence Discriminative Training for Offline Handwriting Recognition by an Interpolated CTC and Lattice-Free MMI Objective Function,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2017, vol. 1, pp. 61–66.

[25] D. A. Huang, L. Fei-Fei, and J. C. Niebles, “Connectionist temporal modeling for weakly supervised action labeling,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9908 LNCS, pp. 137–153.

[26] J. Wan, S. Z. Li, Y. Zhao, S. Zhou, I. Guyon, and S. Escalera, “ChaLearn Looking at People RGB-D Isolated and Continuous Datasets for Gesture Recognition,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 761–769.

[27] B. K. P. Horn and B. G. Schunck, “Determining optical flow.,” in *Computer vision*, 1981, pp. 185–203.

[28] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 1, pp. 448–456.

[29] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional Two-Stream Network Fusion for Video Action Recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 1933–1941.

[30] I. Jindal, M. Nokleby, and X. Chen, “Learning deep networks from noisy labels with dropout regularization,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017, pp. 967–972.

[31] A. Graves, “Supervised Sequence Labelling with Recurrent Neural Networks (Studies in Computational Intelligence),” in *Springer*, 2012, p. 160.