Chapter 5 HUMAN ACTIVITY RECOGNITION MODELS USING DEEP RESIDUAL NETWORKS

5.1. Introduction

In this chapter, two approaches based on deep residual networks has been proposed. First model investigates and uses deep residual networks with fusion based dual stream pre-trained models for activity recognition from video streams. The architecture is further trained and evaluated using standard video actions benchmarks of UCF-101, HMDB-51 and NTU RGB. Performance of depth-based variants of residual networks is also analyzed. The proposed approach not only provides competitive results but also better at exploiting pre-trained model and annotated image data. The second model is encoder-decoder based model using CNN and RNN. Introduction of residual connections in traditional CNN model to design very deep architectures known as Residual Networks are very promising for computer vision tasks. To exploit capabilities of both CNN and RNN the proposed model is based on CRNN which is trained from scratch as well as using ResNet 152 which is pre trained on ImageNet dataset. The architecture is trained and validated on popular UCF-101 dataset on the basis of accuracy and average loss. From results, it can be observed that proposed approach provides better results than state of art methods.

5.2. Dual Stream HAR Model Exploiting Residual-CNN

A video consists of sequence of images or frames along the temporal dimension. Identification of activity can be simply accomplished by using 2D convolutions on images/frames separately to learn activity representation. This approach however does not take account of motion encoded in sequence of frames. Identification of some activities is possible from static appearance only; however, for other activities this may

not hold true. Hence, different approaches are adopted to take account of temporal information. Using some additional input modality like optical flow, motion history images, and binary motion images etc. is one way to do this. Contribution of additional input modality to learn activity class labels cannot be ignored, but at same time it needs additional pre-processing of video data to get desired input modality. Thus here, work idea is to get comparable results using only RGB frames and minimal training by using very deep residual models in contrast to shallow networks.

5.2.1. Proposed Method

The proposed solution comprises of two network streams: “Spatial stream” and “Spatio-temporal” stream as shown in Fig.5.1. Each network stream refers to CNN with RGB frames extracted from video dataset as input modality. “Spatial stream” learns activity representation from RGB frames using 2D convolutions, hence it ignores temporal dimension for activity class labeling. Whereas “Spatio-temporal stream” learns features along spatial as well as temporal dimension using 3D convolutions. Training network using RGB frames provide opportunity to exploit large image datasets such as ImageNet [321]. Class scores of both networks are then combined, and class with maximum scores is predicted as final activity class label for the video. Architecture of network streams is illustrated and described in Section-5.2.2

5.2.1.1. Network Architecture

In this section proposed network architecture is discussed. In contrast to shallow networks used in [84], [85], proposed model uses relatively deep residual networks [333].

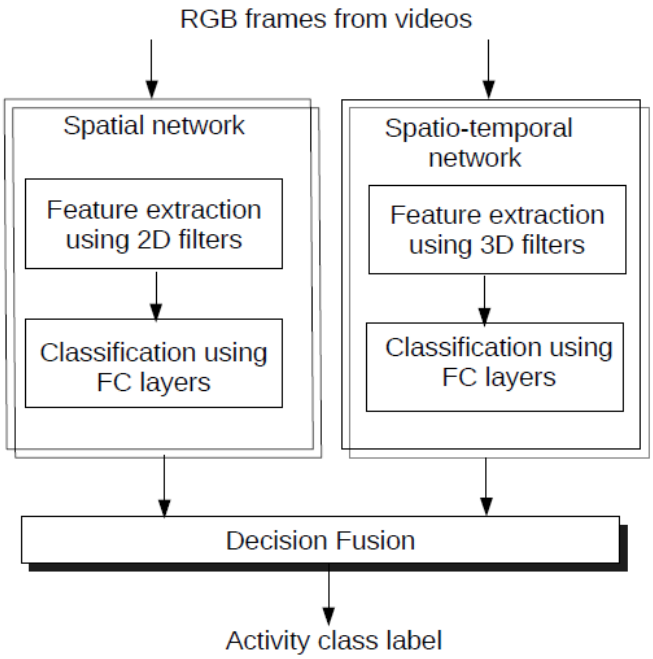


Figure 5.1 Proposed model using 2D and 3D CNN for activity recognition

Residual network architecture is powerful network architecture which bagged all the ImageNet challenges, including classification, detection, and localization. The residual learning framework includes shortcut connections as shown in Fig. 5.2, that allow signal to bypass one layer and jump to the next layer in the sequence.

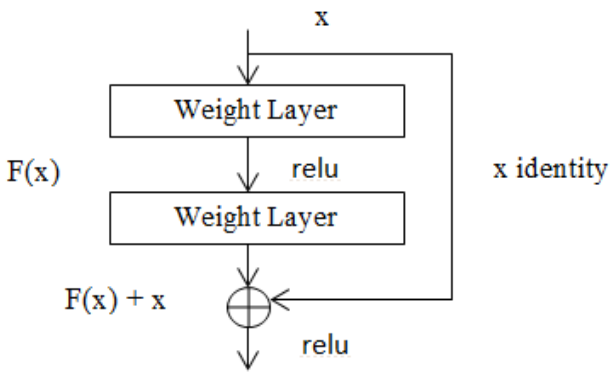


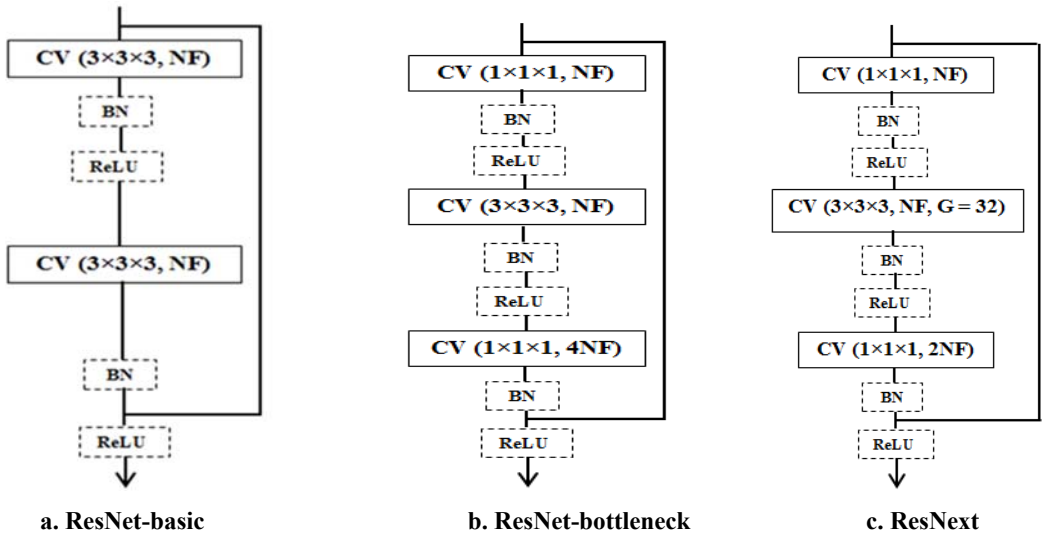
Figure 5.2 Shortcut connection in residual learning [333]

Such shortcut connections allow design of very deep networks with minimal parameters and improved performance.

5.2.1.2. Variants of Residual Network

There are different residual networks depending on shortcut connections and depth that refers to the number of layers in the network. Firstly, Resnet-18, 50, 101 and ResNext101 with both 2D and 3D convolutions are experimented separately. For spatial stream, above mentioned 2D residual networks are evaluated and for Spatio-temporal stream, similar 3D variants are evaluated. Residual networks comprise of blocks, shown in Fig.5.3. Here CV marks convolution layers, X×X×X indicates size of convolution filters and NF represents number of feature maps. Convolution layers are followed by Batch Normalization (BN) and ReLU (Rectified Linear Unit) activation.

ResNet-18 uses ResNet basic blocks (a), ResNet 50 and ResNet 101 uses bottleneck blocks (b) and ResNext 101 uses ResNext block(c). Basic block comprises of two convolution layers indicated by CV(convolution layer) with filter size 3×3×3 for 3D ResNets (For 2D ResNets this is 3×3.) as shown in Fig. 5.3(a).



a. ResNet-basic **b. ResNet-bottleneck** **c. ResNext**

Figure 5.3 Block structure for different residual networks (CV= Convolution Layers, NF= Number of Feature Maps, BN= Batch Normalization, G=Group)

Each CV layer is followed by batch normalization (BN) and Rectified Linear Unit (ReLU). There is shortcut pass from beginning to layer just before last ReLU. A ResNet

bottleneck block comprises of three convolution layers indicated by CV in Fig. 5.3 b. First and third convolution layers have filters of size 1×1×1 whereas second layer has filter of size 3×3×3 for 3D ResNets (for 2D ResNets these sizes are 1×1 and 3×3 respectively. Each CV layer is followed by batch normalization (BN) and ReLU. There is shortcut pass from beginning to layer just before last ReLU. One complete network is then formed by organizing these blocks. Specifications of ResNet-18, 50, 101 and ResNext101 are mentioned in Table.5.1. Type of block used in each network is specified by row-3. In column-1, named layers represent layer structure of the complete network in order from top to bottom. First layer represented by CV1 with complete specification given by cell corresponding to each variant, ex. [7×7×7, NF=64, temporal stride=1, Spatial stride=2] for ResNet18. Here NF marks number of feature maps**.**

5.2.1.3. Spatial Network Stream: 2D Variants of Residual CNN

Spatial network is responsible for activity recognition using spatial information from video frames. Previous works by different authors have shown that deeper structures improve performance of computer vision tasks [334]. This is the reason to choose relatively deep 2D residual nets for proposed methodology in contrast to popular two-stream model [85]. Network is initialized with ImageNet [321] weights, as it is rare to have dataset of enough size to train the network with random initialization of weights.

Table 5.1 Layer specification for different 3D residual networks (Similar for 2D ones except 2D convolution kernels)

3D Residual networks				
Architecture →	**ResNet-18**	**ResNet-50**	**ResNet-101**	**ResNext-101**
Block → Layers	Basic	Bottleneck	Bottleneck	ResNext
CV1	7×7×7, NF=64, temporal stride=1, Spatial stride=2	7×7×7, NF=64, Temporal stride=1, Spatial stride=2	7×7×7, NF=64, Temporal stride=1, Spatial stride=2	7×7×7, NF=64, Temporal stride=1, Spatial stride=2
CV2_x	BN=2, NF=64	BN=3, NF=64	BN=3, NF=64	BN=3, NF=128
CV3_x	BN=2, NF=128	BN=4, NF=128	BN=4, NF=128	BN=4, NF=256
CV4_x	BN=2, NF=256	BN=6, NF=256	BN=23, NF=256	BN=23, NF=512
CV5_x	BN=2, NF=512	BN=3, NF=512	BN=3, NF=512	BN=3, NF=1024
Fully connected	FC(Fully Connected) layer	FC(Fully Connected) layer	FC(Fully Connected) layer	FC layer

5.2.1.4. Spatio-Temporal Network Stream: 3D Variants of Residual CNN

Convolution networks with 2D kernels do not preserve temporal information. Therefore, either input modalities like optical flow [84], [85], motion history images etc. or CNN with 3D convolutions which are able to learn spatio-temporal features [141] are applied. To gain from the temporal information, 3D residual networks are thus used in spatio-temporal stream. 3D Residual networks use 3D convolutional kernels that provide temporal information without additional processing to get input modality like optical flow. Rather than training model from scratch, pre-trained model is fine-tuned for the problem at hand to reduce training needs of proposed network model. Spatio-Temporal network stream, pre-trained on Kinetics [110] dataset has been used for implementing transfer learning. Kinetics dataset contains 400 action classes, with around 400 video clips for each action, thus it has enough data for training CNN. Fine-tuning of last

convolutional layer (CV5_x) and the fully connected layer of the network is done for datasets used in this work.

5.2.1.5. Decision Fusion

For final activity prediction, class scores of both streams are combined using different fusion techniques such as Sum fusion, Product fusion, Max fusion and Weighted Average fusion which generate final activity class scores, whereas class with maximum score is final activity class label for input video stream.

5.2.2. Experimental Setup

Hardware and software setup used in experiments is given in section 5.2.2.1 Experiments are performed on widely used and challenging benchmarks for action recognition: UCF-101 and HMDB-51 and NTU RGB described in section 5.2.2.2. Section 5.2.2.3, 5.2.2.4 and 5.2.2.5 specifies training and testing setup for spatial and spatio-temporal streams. Section 5.2.3 presents experimental results and comparison of proposed work with state-of-art methods.

5.2.2.1. Hardware and Software Setup

Fig.5.4 shows the hardware and software setup used for experiments. The network's size is limited by resource constraints mainly the amount of GPU memory and training time that one is willing to tolerate. Hardware is composed of CPU and Nvidia Quad GPU having 8 gigabytes of memory. Linux OS Ubuntu 16.04 is used in conjunction with Python and CUDA toolkit. Python provides various deep learning libraries and CUDA toolkit supports GPU-accelerated computing. PyTorch deep learning development platform is used, that provide very simple API for implementation of neural nets.

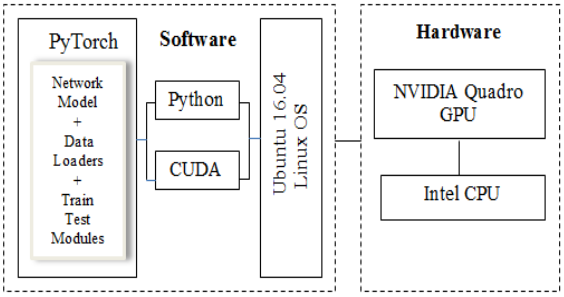


Figure 5.4 Hardware and software setup

5.2.2.2. Dataset and Performance Measure

UCF101 [109] is a common video dataset that consists of total 13320 video. Different action videos are grouped into 25 groups, where each group contains 4-7 videos of an action. The action categories include five types: 1) Body-Motion Only 2) Human- Object Interaction 3) Human-Human Interaction 4) Playing Musical Instruments 5) Sports. Fig.5.5 (a) represents frames of some four different activity classes. The reason behind choosing this dataset is that it offers high diversity in camera motion, object appearance, pose and scale, illumination conditions etc. This allows testing and verifying the robustness and effectiveness of recognition model in the harsh real-world scenarios. Second dataset we considered for our experiments is HMDB-51 [111], the largest action video database that has 51 action categories. It consists of around 7,000 manually annotated clips. The actions can be grouped into five types: 1) General facial actions 2) Facial actions with object manipulation 3) General body movements 4) Body movements with object interaction 5) Body movements for human interaction. Sample activity frames are represented by Fig.5.5 (b). NTU RGB [289] is public benchmarking action recognition. It includes 56,880 action samples each for RGB, depth, skeleton and infra-red videos. There are 40 human subjects performing 60 different actions including 50 actions by single person and 10 two-person interactions.

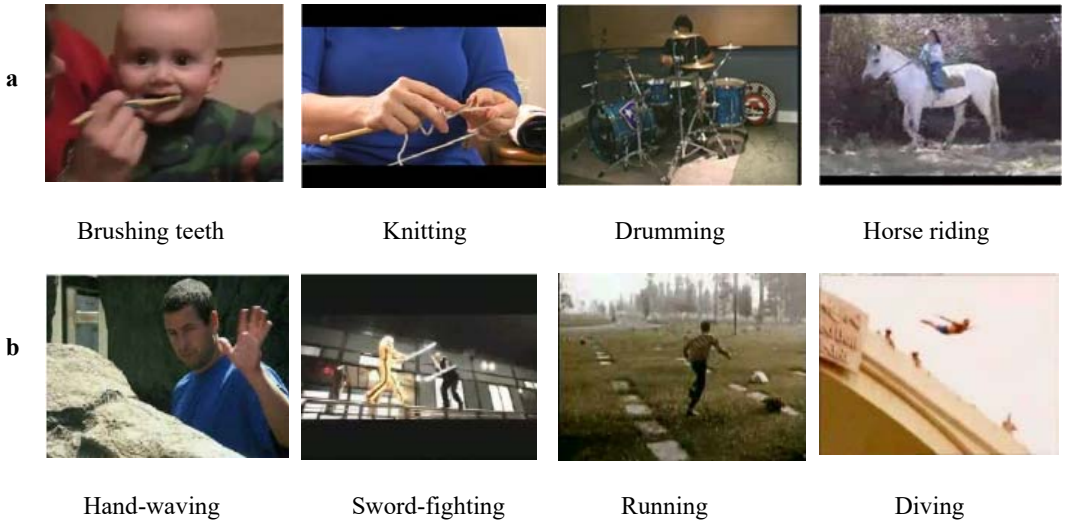


Figure 5.5 RGB frames extracted from different activity classes of UCF101 [210].

5.2.2.3. Network Training and Testing

The first step for network training consists of extraction of RGB frames for each dataset using ffmpeg. Mini batch training is used, that is combination of batch and stochastic training, as it uses specified number of items (batch size) to compute gradients. Batch size is adjusted to fit data in available GPU memory. Transformation functions are applied to video frames for data augmentation and generalization of trained model.

5.2.2.4. Train/Test Settings for Spatial Stream

This section summarizes train/test setting used for 2D variants of residual networks. As spatial stream is concerned with activity recognition from still video frames using 2D kernels, three frames are selected from each video in mini-batch of size 10. From each selected frame, 224 ×224 sub-image is randomly cropped. Random flipping is also applied for generalization of trained model as shown in Fig 5.6. Other train/ test settings are mentioned in the Table.5.2. Learning rate is initially set to 5e-4. learning rate scheduler (ReduceLROnPlateau with patience=2) available with PyTorch is used, that

reduce learning rate when metric stops improving. For calculating loss, video level prediction is obtained by consensus among 16 frames.

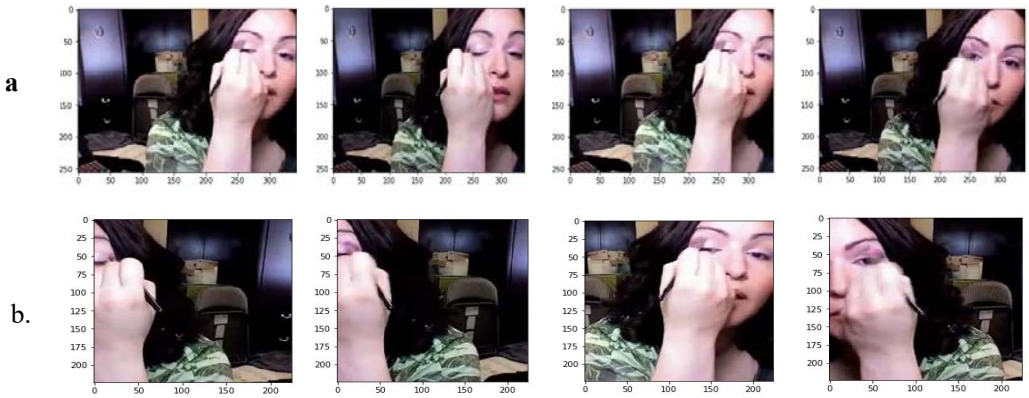


Figure 5.6 Illustrates image transformations: a) Original frames of Apply Eye Make-Up activity b) Transformed frames: Randomly cropped to 224×224 and flipped with probability 0.5

5.2.2.5. Train/Test Settings for Spatio-Temporal Stream

This section summarizes train/test setting used for evaluating 3D variants of residual network in spatio-temporal stream. Test settings are mentioned in the Table.5.2, 16 frames are selected randomly from each video in mini-batch of size 25. To acquire more temporal information more number of frames are also experimented, with reduction in the spatial extent of each frame to fit data into available memory. The size of each sample is thus 3 channels×16 frames×112 pixels×112 pixels.

Table 5.2 Training / Testing details for 2D residual network

Parameter	Value
Input of spatial stream:	Size of single frame = 3*224*224 , Random crop method
No. of frames (Training) :	3 frames, selected randomly from 1/3 of total number of frames of each video
Mini-batch training:	Batch size=10
No. of epochs	50
Iterations	9537/10=953.7 =954 iterations
Initial learning rate	5e-4
No. of frames(Testing)	16 frames selected with equal temporal spacing
Accuracy measure	Video level accuracy: combined class scores each of 19 frame clips

Table 5.3 Training / Testing details for 3D residual networks

Parameter	Value
Input of spatio-temp stream:	Size of single frame = 3*112*112, Random crop
No. of frames (Training) :	16 frames selected randomly, Temporal random crop, Sample size= 3*16*112*112
Mini-batch training:	Batch size=25
No. of epochs	50
Iterations	9537/25=381.48 =382 iterations
Initial learning rate	0.001
No. of frames (Testing)	Multiple 16 frame clips from each video
Accuracy measure	1.Clip accuracy 2.Video level accuracy: combined class scores of all clips

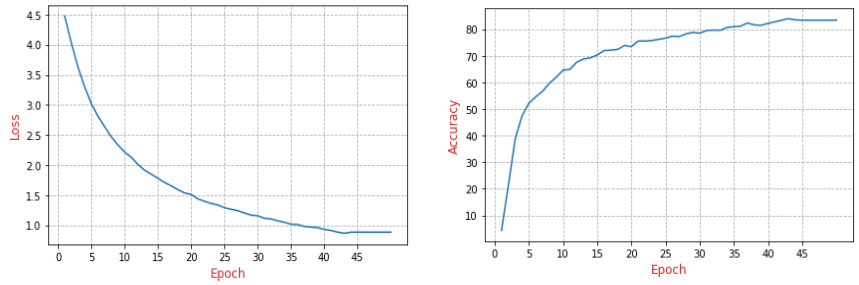
With all variants of network Stochastic Gradient Descent (SGD) with momentum is used to train the network. Cross-entropy losses and back propagation of gradients is used. Weight decay and momentum are 0.001 and 0.9 respectively.

5.2.3. Results and Discussion

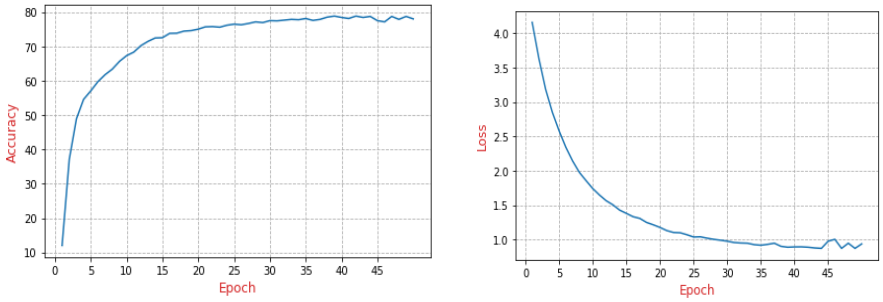
Section 5.2.3.1 presents results for 3D Residual networks and 5.2.3.2 represent results of 2D residual networks on split-1 of UCF101 video dataset. Based on accuracy attained, best model from 2D CNNs and 3D CNNs is selected to work as Spatial-stream and Spatio-temporal stream respectively. Hence for HMDB-51 and NTU RGB experiments were performed with only those depth variants of 2D and 3D CNN that provided best results on UCF-101. Results on HMDB-51 and NTU RGB are presented by Table 5.5.

5.2.3.1. Results: 3D Residual CNN

Fig.5.7 to 5.10 shows accuracy and loss curves over 50 epochs during training and testing for UCF-101.

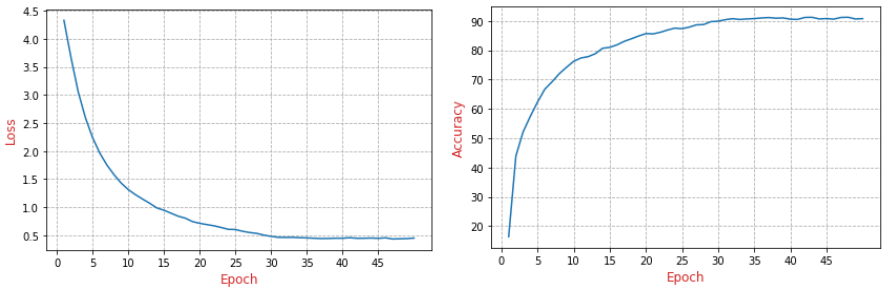


a. Loss and accuracy curve for training set

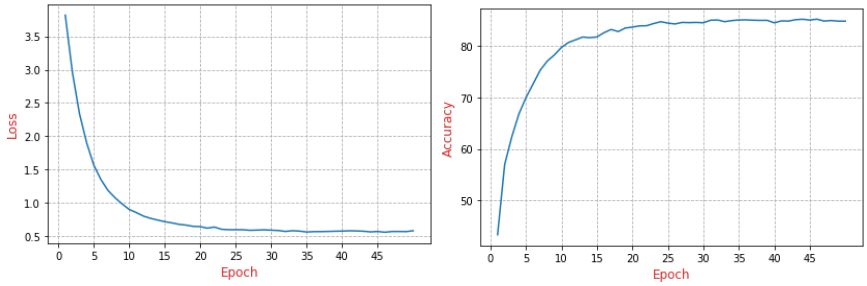


b. Loss and accuracy curve for testing set

Figure 5.7 Performance of 3D Resnet-18; Best prediction: [Epoch=42, Accuracy=78.18]

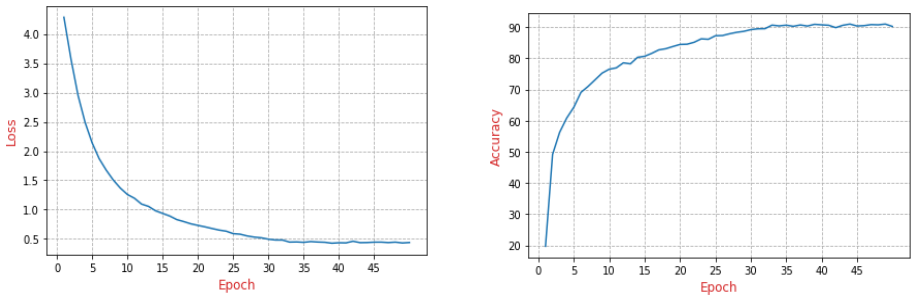


a. Loss and accuracy curve for training set

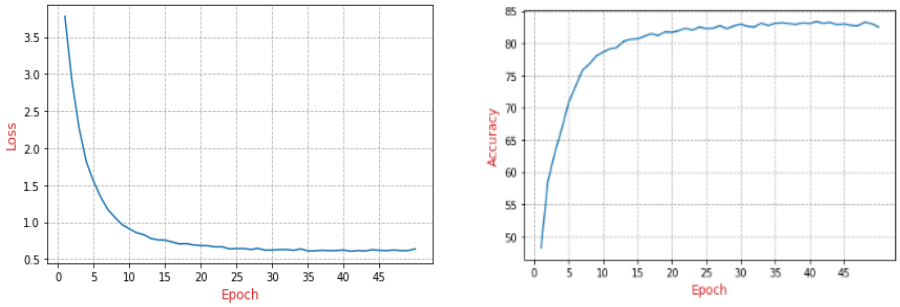


b. Loss and accuracy curve for testing set

Figure 5.8 Performance of 3D Resnet-50; Best prediction: [Epoch=43, Accuracy=85.18%]



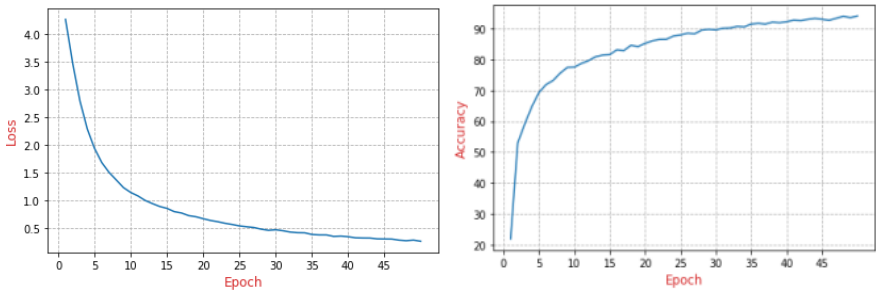
a. Loss and accuracy curve for training set



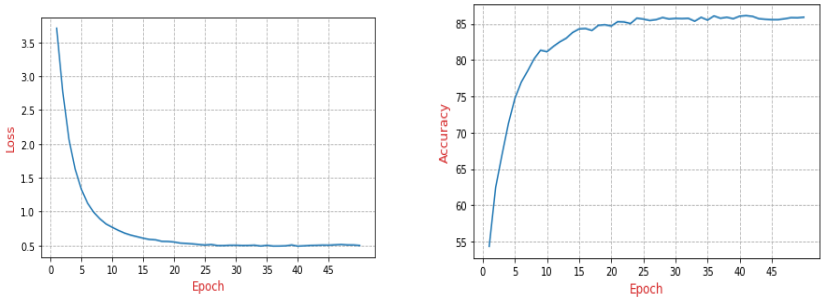
b. Loss and accuracy curve for testing set

Figure 5.9 Performance of 3D Resnet-101; Best prediction: [Epoch=41, Accuracy=83.34%]

Validation accuracies of 3D ResNet-18, 50, 101, ResNext101 are 78.8, 85.18, 83.34, 86.1 percent respectively. 3D ResNext-101 gives maximum accuracy of 86.1% (Clip accuracy) on UCF101. It should be noted that ResNet-50 outperforms ResNet-101 by around 2%, hence it cannot be directly assumed that deeper net always gives higher accuracy than less deep one. Thus, depth of network that fits best depends on problem at hand.



a. Loss and accuracy curve for training set

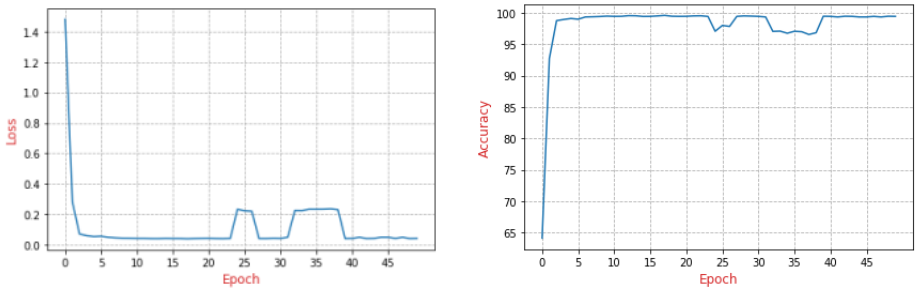


b. Loss and accuracy curve for testing set

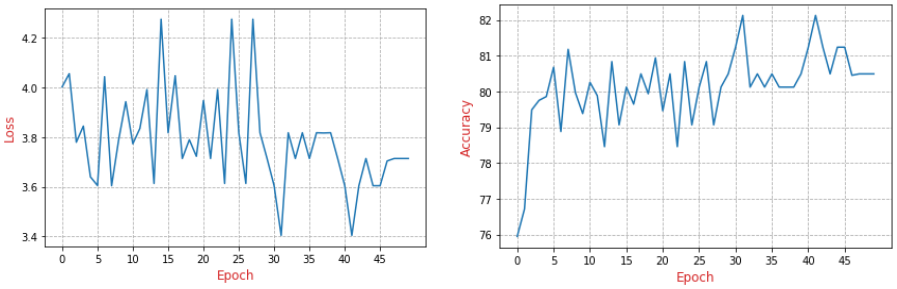
Figure 5.10 Performance of 3D Resnext-101; Best prediction: [Epoch=41, Accuracy=86.1%]

5.2.3.2. Results: 2D Residual CNN

Similarly, performance of 2D networks is evaluated. ResNet -101 provides best predictions with accuracy of 82.1 percent. Loss and accuracy curve for both training and testing are shown in Fig.5.11.



a. Loss and accuracy curve for training set



b. Loss and accuracy curve for testing set

Figure 5.11 Performance of 2D Resnet-101; Best prediction: [Epoch=31, Accuracy=82.23%]

Results shows that among 3D nets ResNext provides maximum accuracy of 86.1 and ResNet101 provide maximum accuracy of 82.1% among 2D nets. Hence these two networks are selected for final inclusion into proposed dual stream model. During testing of 3D ResNext clip accuracy is used as performance measure. Class scores of different clips corresponding to each video are combined. Combined score is used to predict final

activity class label. Video level accuracy of 3D ResNext-101 is 88.23%. Finally, class scores generated by 3D ResNext-101and 2D ResNet-101 are combined using different fusion techniques. Using weighted product fusion accuracy of 92.89% was obtained on UCF-101 dataset as shown in Table 5.4. Dual stream model is then trained and tested using 3D ResNext101 and 2D ResNet101 architectures on HMDB-51 and NTU RGB benchmarks, results for the same are represented by Table 5.5.

Table 5.4 Results of Dual-stream model on UCF split-01

Stream	**Model**	**UCF101 split-01(Recognition accuracy)**
CNN spatial (2D-CNN)	ResNet-101	82.23%
CNN spatio-temp (3D-CNN)	ResNext-101	88.23%
CNN fusion	Sum fusion	88.67%
CNN fusion	Max fusion	91.56%
CNN fusion	Weighted Average fusion	92.04%
CNN fusion	Weighted Product fusion	92.89%

Table 5.5 Results of Dual-stream model on HMDB-51 and NTU RGB Datasets

HMDB-51			
Stream	**Model**	**(Recognition accuracy)**	
CNN spatial (2D-CNN)	ResNet-101	57.20%	
CNN spatio-temp (3D-CNN)	ResNext-101	61.89%	
CNN fusion	Sum fusion	62.32%	
CNN fusion	Max fusion	62.79%	
CNN fusion	Weighted Average fusion	63.87%	
CNN fusion	Weighted Product fusion	64.13%	
NTU RGB dataset			
		Cross Subject	**Cross View**
CNN spatial	ResNet-101	58.52%	61.22%
CNN spatio-temp	ResNext-101	60.66%	62.36%
CNN fusion	Sum fusion	61.78%	63.63%
CNN fusion	Max fusion	62.12%	63.53%
CNN fusion	Weighted Average fusion	62.01%	64.22%
CNN fusion	Weighted Product fusion	62.69%	64.89%

5.2.3.3. Comparison with State-of-Arts

Proposed approach uses only RGB frames for activity recognition in multi-stream model; hence some of RGB based and multi-streams methods on UCF101 are considered for comparison. Comparison is summarized in Table.5.6 and presented using bar chart in figure 5.12. $CNN_{spatial}$ (proposed) outperforms spatial stream (VGGM-2048) in [84], [85] by good margin. CNN spatio-temp outperforms by ~ 7% (81.2 versus 88.23) when compared with temporal stream of [85] and by ~ 2% (86.25 versus 88.23) for [84]. Combined result of both streams CNN fusion also outperforms combined results of [85] by ~5%.It also improves from 90.62 obtained by late fusion (VGG-16) in [84] by ~2%. It also outperforms three stream architecture with dynamic flows and IDT features implemented in [335] by good margin

Table 5.6 Comparison of proposed Deep-dual stream model with state-of-art methods on on UCF-101 dataset [210]. (*mean accuracy over three UCF splits)

Method	**Input modality**	**Recognition Accuracy (%)**
[50]Slow fusion	RGB	65.4*
[85] Spatial stream	RGB	72.7
[85] Temporal stream	Optical flow	81.2
[85] Fusion by average	RGB + Optical flow	86.2
[85] Fusion by SVM	RGB + Optical flow	87.0
[84] Spatial stream	RGB	74.2 (VGGM-2048) 82.61 (VGG-16)
[84] Temporal stream	Optical flow	82.34(VGGM-2048) 86.25(VGG-16)
[84] Late fusion	RGB + Optical flow	85.94 VGGM-2048), 90.62(VGG-16)
[335] Decision fusion	Dynamic Flow + RGB	84.93% (AlexNet) 87.63% (VGG 16)
[335] Fusion by SVM	Dynamic Flow + RGB + Optical Flow	88.63% (AlexNet) 90.30% (VGG 16)
[335] Fusion by SVM	Dynamic Flow + RGB + Optical Flow +IDT-FV	89.20% (AlexNet)
[335] Fusion by SVM	Dynamic Flow + RGB +IDT-FV	91.10% (AlexNet)
$CNN_{spatial}$ (ours)	RGB	**82.23**
$CNN_{spatio-temp}$ (ours)	RGB	**88.23**
$CNN_{sum\ fusion}$ (ours)	RGB	**88.67%**

CNN $_{\text{max fusion}}$ (ours)	RGB	**91.56%**
CNN $_{\text{weighted average fusion}}$ (ours)	RGB	**92.04%**
CNN $_{\text{weighted product fusion}}$ (ours)	RGB	**92.89%**

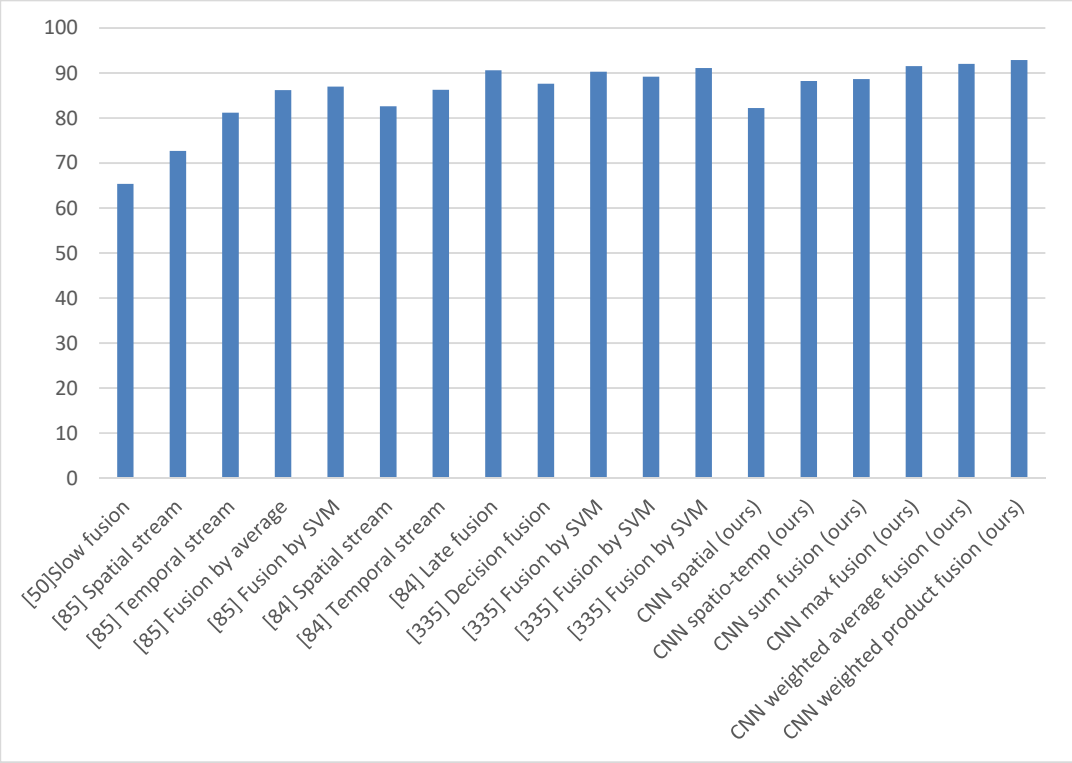


Figure 5.12 Comparision chart of the porposed model with state-of-the-art methods on UCF-101 dataset

5.3. Human Activity Recognition using Convolutional Recurrent Neural Network (CRNN)

A video consists of sequences of images or frames along the temporal dimension. Identification of activity can be simply accomplished by using 2D convolutions on images/ frames separately to learn activity representation. This approach doesn't take motion encoded in frames into account. Identification of some activities is possible by using static frames but doesn't hold true for other activities. Hence different approaches are taken into account for temporal information. Using some additional input modality like optical flow, dynamic images and binary motion images etc. is one way to do this.

Contribution of additional input modality to learn activity class labels cannot be ignored, but at the same time it needs additional pre- processing of video data to get desired input modality. Thus here, work idea is to get comparable results using only RGB frames and minimal training by using very deep residual models in contrast to shallow networks. In the proposed CRNN model we have combined features of CNN as well of RNN. Two variants of the proposed model is presented, one with the training from scratch and other using pre-trained ResNet 152 architecture.

5.3.1. The Proposed Method

A video consists of sequences of images or frames along the temporal dimension. Identification of activity can be simply accomplished by using 2D convolutions on images/ frames separately to learn activity representation. This approach doesn't take motion encoded in frames into account. Identification of some activities is possible by using static frames but doesn't hold true for other activities. Hence different approaches are taken into account for temporal information. Using some additional input modality like optical flow, dynamic images and binary motion images etc. is one way to do this. Contribution of additional input modality to learn activity class labels cannot be ignored, but at the same time it needs additional pre- processing of video data to get desired input modality. Thus here, work idea is to get comparable results using only RGB frames and minimal training by using very deep residual models in contrast to shallow networks. In the proposed CRNN model we have combined features of CNN as well of RNN. Two variants of the proposed model is presented, one with the training from scratch and other using pre-trained ResNet 152 architecture.

5.3.1.1. Overview of CRNN Model

In CRNN combination of 2D CNN with RNN or LSTM's is used as shown in figure 5.12. First step involves to get frames from input. We have used UCF 101 dataset for training purposes. Generated frames over the time are feed to CNN. CNN encodes every 2D image X(t) into a 1D vector Z(t) by using equation 5.1.

$$f_{\mathrm{CNN}}(\mathbf{x}^{(t)}) = \mathbf{z}^{(t)} \tag{5.1}$$

CNN acts as encoder which encodes all the information generated from input images into a single vector. CNN is trained from scratch having four convolutional layers with Relu as the activation function. Moreover, two pooling and two Fully Connected layers are also used. Then this generated vector is feed into RNN which acts as decoder. Which in return predicts the class of the action.

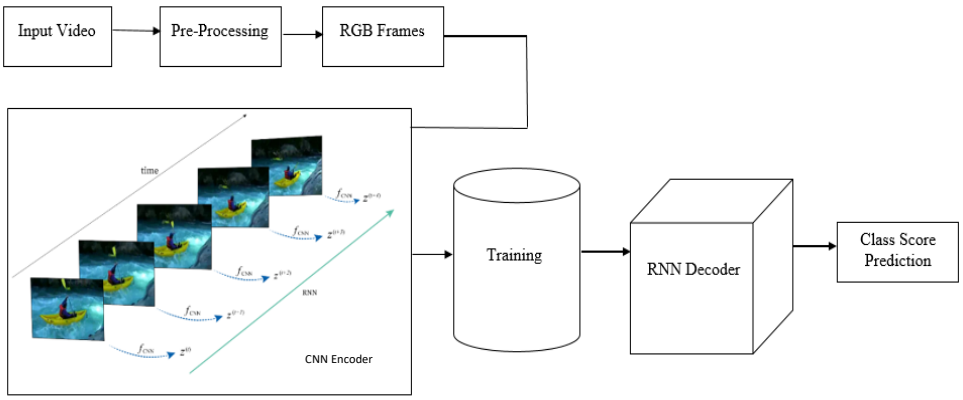


Figure 5.13 Proposed CRNN model

5.3.1.2. Overview of the ResNet CRNN Model

In ResNet CRNN model there is combination of 2D CNN with RNN as shown in figure 5.13. First step involves to get frames from input. We have used UCF 101 dataset for training purposes. Generated frames over the time are feed to CNN. CNN encodes every 2D image X(t) into a 1D vector Z(t) by using equation 5.2.

$$f_{\mathrm{CNN}}(\mathbf{x}^{(t)}) = \mathbf{z}^{(t)} \tag{5.2}$$

CNN acts as encoder which encodes all the information generated from input images into a single vector. The training of the model takes place by using Resnet 152. ResNet is pre trained on large image datasets such as ImageNet [321]. The output from the trained model is used by RNN for decoding purpose. Which in return predicts the class of the action. The proposed model is shown below in figure 5.13.

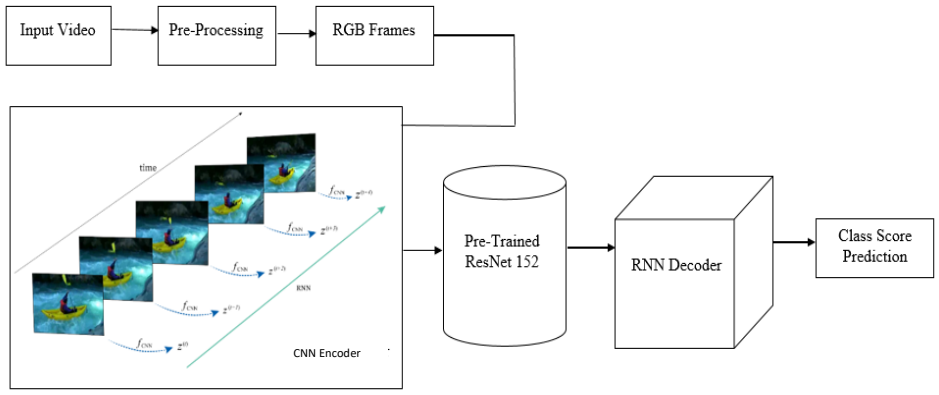


Figure 5.14 ResNet CRNN model

ResNet 152 is having 152 layers in total and description of its layer is specified in Table 5.7. Type of block used is specified by Column 2 of table 5.7. Column-1 named layers represent layer structure in order from top to bottom, first layer represented by CV1 with complete specification given by cell corresponding to each variant, ex. [7×7, NF=64, temporal stride=1, Spatial stride=2] and for other layers is specified in the other columns.

Table 5.7 Layer specification for 2D ResNet 152.

Residual Network ResNet 152	
Layer Name	Configuration
CV1	7x7, NF=64 Stride=2
CV2_x	NB=3, NF=64
CV3_x	NB=8, NF=128
CV4_x	NB=36, NF=256
Cv5_x	NB=3, NF=512
Fully Connected	FC Layer

5.3.2. Implementation and Experimental Results

To demonstrate the proposed model we have conducted experimentation and training using UCF 101 [109] dataset. Section 5.3.2.1 consists of training and testing setting required for the proposed model.

5.3.2.1. Network Training

For training purposes, we have used all the three splits of UCF 101 dataset. For CRNN model we have selected 29 frames from each video in mini-batch of size 30. To acquire more temporal information, we used a greater number of frames; hence we reduced the spatial extent of each frame fit data into available memory. The size of each sample is thus 3 channels×256 frames×342. Total of 120 iterations are done with initial learning rate 1e-4. For validation purposes, we have selected 19 frames. For ResNet CRNN model we have selected 29 frames from each video in mini-batch of size 40. To acquire more temporal information, we used a greater number of frames, hence we reduced the spatial extent of each frame fit data into available memory. The size of each sample is thus 3 channels×256 frames×342. Total of 120 iterations are done with initial learning rate 1e-4.

5.3.2.2. Results: CRNN

Fig. 5.14 to Fig. 5.15 shows accuracy and loss curves over 120 epochs during training and testing. Validation accuracy of 2D CRNN is 68.32 percent with average validation loss as 3.26%.

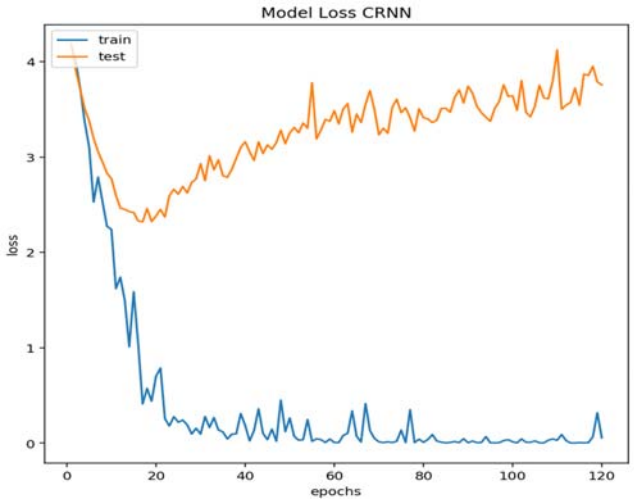


Figure 5.15 Overall loss of 2D CRNN during Training and Testing

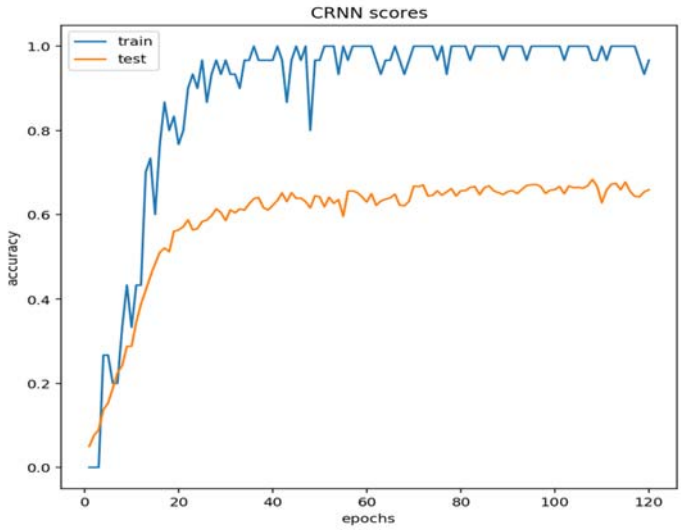


Figure 5.16 Accuracy of 2D CRNN during Training and Testing. Best Epoch 107.

5.3.2.3. Results: ResNet CRNN

Fig. 5.16 to 5.17 shows accuracy and loss curves over 120 epochs during training and testing. Validation accuracy of ResNet CRNN is 90.32 percent with average validation loss as 0.94%. Results indicate that there is increase of accuracy almost 23 percent from CRNN model. As we have used ResNet 152 pre trained model which increase the accuracy as shown in Fig 5.17 below.

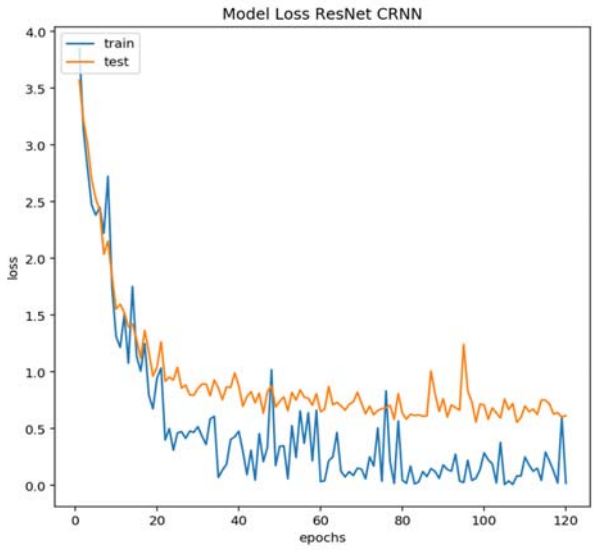


Figure 5.17 Overall loss of ResNet CRNN during Training and Testing

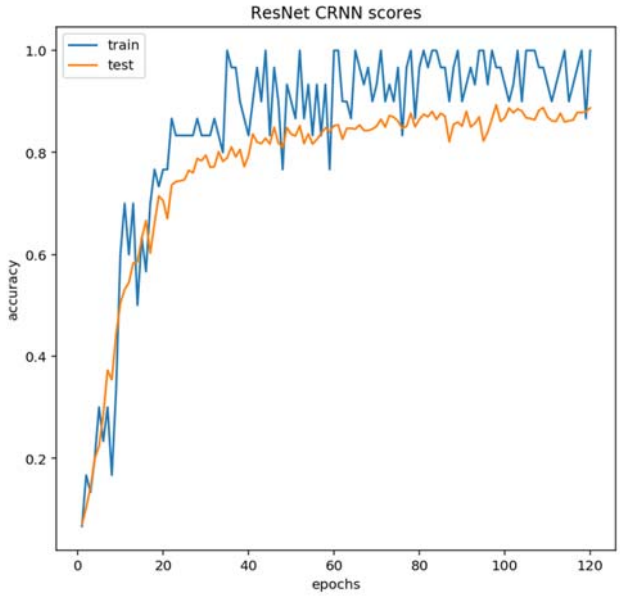


Figure 5.18 Accuracy of ResNet CRNN during Training and Testing. Best Epoch 97.

Summary of results for the proposed two models named CRNN and ResNet CRNN is presented in Table 5.8 below.

Table 5.8 Results of the proposed model on UCF 101 Dataset

Model	**Accuracy (in %)**	**Loss (in %)**	**Best Epoch**
CRNN	68.32	3.26	107
ResNet CRNN	90.32	0.94	97

5.3.2.4. Comparison with State-of-the-Art Methods

Here we compare performance of the proposed model with various state-of-the-art methods based on UCF101 dataset. As we have used only single modality i.e. RGB Frames so we considered some of RGB based and other modality-based methods on UCF101 for comparison. Comparison is summarized in Table.5.9 and Figure 5.19. Proposed ResNet CRNN outperforms spatial stream (VGGM-2048) in [90], [336], [84] by good margin. CRNN outperforms by ~ 3% (68.32 versus 65.4) when compared with slow fusion of [90] and by ~ 2% (86.25 versus 88.23) for [336].

Table 5.9 Comparison of the proposed CRNN model with state-of-the-art methods on UCF101 dataset.

Method	**Input modality**	**Video Accuracy (%)**
[90]Slow fusion	RGB	65.4
[336]Spatial stream	RGB	72.7
[336]Temporal stream	Optical flow	81.2
[336]Fusion by average	RGB + Optical flow	86.2
[336]Fusion by SVM	RGB + Optical flow	87.0
[84]Spatial stream	RGB	74.2 (VGGM-2048) 82.61 (VGG-16)
[335]Decision fusion	Dynamic Flow + RGB	84.93% (AlexNet) 87.63% (VGG 16)
CRNN (proposed)	RGB	**68.32**
ResNet CRNN (Proposed)	RGB	**90.32**

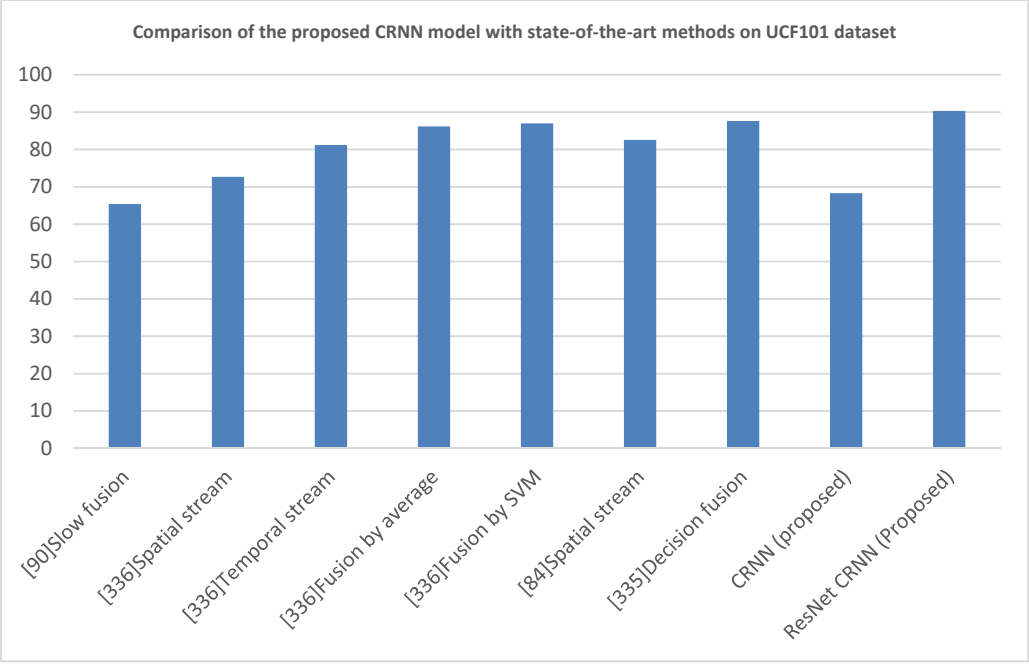


Figure 5.19 Comparision chart of the porposed CRNN model with state-of-the-art methods on UCF-101 dataset

5.4. Conclusion

In this chapter two different models for human activity recognition using deep residual networks has been presented. First model is a two stream model for activity recognition using very deep CNN with residual connection. Firstly, performance of 2D and 3D residual networks is evaluated at different depths. For 3D network stream fine-tuning of Kinetics pre-trained model for UCF101 is examined, that provided very good results with minimal training and only RGB frames as input modality. Results of residual CNN with 3D convolutions clearly emphasize that deeper nets with residual connections have the potential to contribute to significant progress in fields related to various video analysis tasks. Result obtained using decision fusion is also comparable to state-of-the-art models that have even used additional input modality. And second model uses ResNet CRNN for activity recognition using very deep CNN with residual connections. We first evaluated performance of CRNN trained from scratch. The proposed methodology provides very good results with minimal training and only RGB frames as input modality, hence it addresses stated problem very well. Results of ResNet CRNN clearly emphasize that deeper nets with residual connections have the potential to contribute to significant progress in fields related to various video analysis tasks. Both the model perform better with pre-trained very deep residual network. After evaluation of both models we can see that the deep learning framework with residual networks gives better results in comparison to various state of the arts model with minimum training, by using only RGB modality as input.