Human Activity Recognition using LRCN, and InceptionV3 with RNN variants

Gaurav Bajpai
MEng. Electrical and Computer Engineering
Toronto Metropolitan University
Toronto, Canada
gaurav.bajpai@ryerson.ca

***Abstract*—Human Activity Recognition is the process of recognizing, classifying, and monitoring human actions based on videos and wearable sensors with the help of machine learning and deep learning algorithms. The field of computer vision mainly uses video-based Human Activity Recognition (HAR) models. In this report, Long-term Recurrent Convolutional Network(LRCN), pre-trained Inception v3 model with Simple Recurrent neural network(RNN), Gated recurrent units(GRU), and Long short-term memory(LSTM) are applied to videos belonging to 4 action categories of UCF50 dataset. When these models are applied to a test set, and unknown YouTube videos, the train and validation accuracy, loss, and prediction probabilities of activity recognition have been calculated. [1].**

***Index Terms*—HAR, LRCN, RNN, GRU, LSTM**

I. Introduction

Artificial Neural Network (ANN), used to give predictive actions on order-based or time-series data, is a Recurrent Neural Network. Deep learning layers are used for order-based or temporal tasks. Devices such as Google Assistance, Alexa, and Siri are built with these networks for their users. RNN is made with memory to take information from previous inputs to affect the present input and output and generates the work based on the last intake and its setting. The RNN parameters are the same for each layer of the network. The feed-forward networks have different weights for each neuron, whereas RNNs share the same weight inside individual layers of the network. RNNs are suitable for sequence data processing but have short-term memory problems. LSTMs and GRUs were created to deal with short-term memory using gates. Gates are neural networks that control the flow of information passing through the chain of layers. [2].

The three layers typically make a Convolutional Network are the Convolution layer, Max pooling layer, and fully connected layers. The convolutional layer extracts the features from the input images. The convolution operation is done between an image and a filter of a particular size NxM. The dot product is taken between the filter and the input image and summed. The new resultant image output is called the 'feature map,' which learns patterns about the image, such as the corners and edges. The feature map is sent to other layers to learn additional patterns. A pooling layer follows convolution, reducing the size of the convolved feature map. Fully connected layers consist of weights, biases, and neurons. The feature map from the previous layers before a fully connected layer is flattened (converted to a 1-dimensional vector) and fed to the fully connected layer for classification at the end. [3].

A video is a sequence of images captured and displayed at a fixed interval/frequency. A long-term Recurrent Convolutional Network (LRCN) combines CNN and an LSTM. A CNN takes one image, and convolution is applied to the image to find features on the image that helps to predict the object in that image. An essential property of LRCN is the time-distributed layer. Each of the time-distributed frames is fed to the same CNN, and the output of these parallel CNN processes is fed to the LSTM; the result of the LSTM is provided to the dense layer containing neurons equal to the number of classes, and finally, using a SoftMax activation function the probabilities for the four categories are displayed. The predicted action in the video is the class with the highest probability [4] [5]. This LRCN model representation/flow can be seen in figure 1.

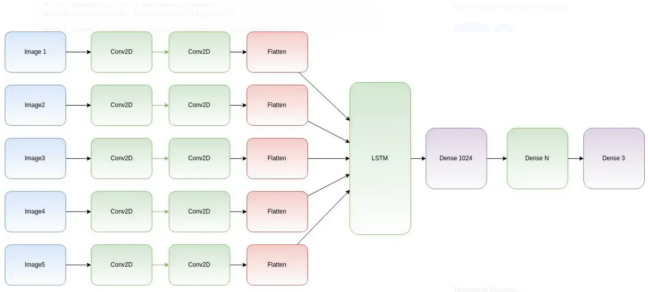

Fig. 1. LRCN model representation [4]

Inception V3 is a deep Convolutional Neural Network model designed for analyzing images and object detecting objects. The architecture of the Inception v3 network is 48 layers deep. It is one of the best models to achieve an accuracy of 78.1% on the ImageNet dataset consisting of over 14 million images into 1000 categories. Usually, deep learning models require a large amount of data for training, and if the data set is small, under-fitting occurs. To deal with this, pre-trained networks such as InceptionV3 can be used. The weight parameters of the pre-trained Inception V3 model are the weights obtained by this model from its training on

classifying the ImageNet dataset into 1000 categories. When the Inception V3 model is imported from Keras, this is a pre-trained Inception V3 model which can be used for the dataset we want to work with [6]. The output from the Inception V3 model is fed to a Simple RNN, the output of the Simple RNN is fed to the dense layer containing neurons equal to the number of classes, and finally, using a SoftMax activation function, the probabilities for the four classes are displayed. The class with the highest probability is the predicted class for the activity in the video. The Inception V3 model with GRU and the Inception V3 model with LSTM follow the same workflow described above.

The architecture of the pre-trained Inception V3 model is shown in figure2

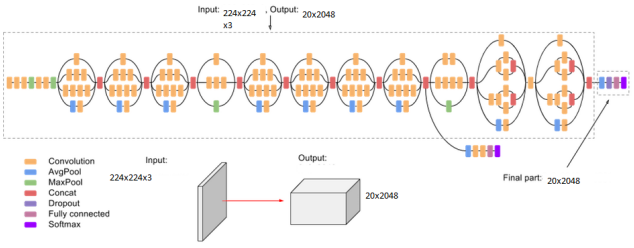

Fig. 2. InceptionV3 model [5]

II. Data set

UCF50 data set has 50 action categories of different videos taken from YouTube used for action recognition. The 50 categories have short video clips, some of which may share features such as the person and the background or location. The dataset is available at the following link: https://www.crcv.ucf.edu/data/UCF50.php UCF50 data set contains 50 action categories, but only a subset of the UCF50 dataset with only 4 action categories – 'HorseRace', 'SalsaSpin', 'WalkingwithDog' and 'TaiChi' has been used because of the computational resource constraints for training the model. These 4 categories contain 533 short video clips [7].

III. Literature Review

Ahmad et al. designed a multi-head CNN-LSTM architecture for human activity recognition. Their dataset is from the UCI database and divided training and test sets in a 7:3 ratio. Six activities sitting, standing, walking, upstairs, downstairs, and lying, are used as data. They compared their results with traditional machine learning methods like SVM. Experimental results on the UCI dataset showed that their proposed multi-head CNN-LSTM approach is promising in terms of performance accuracy compared with other methods, such as a single CNN model and SVM and KNN classifiers for the human activity recognition task [8].

Nafea et al. approach used deep-learning architecture based on CNN's spatial and BiLSTM, allowing deep-learning to obtain temporal signals. Then the activity recognition is accomplished using features obtained from both. Using BiLSTM, the characteristic temporal connection of the spatial deep-learning map can be understood. A new method was developed using the spatial stream with many convolution layers with varying kernel dimensions to achieve feature capture at various resolutions. Data is collected using accelerometers, sensors, and gyroscopes; wireless sensor data mining (WISDM) and UCI datasets are used. Their proposed method obtained an accuracy of 98.53% in the WISDM dataset compared to 97.05% in the UCI dataset. [9].

Deotale et al. applied CNNs to find and classify the sub-activity present in the sequence of frames with the help of considering features, background subtraction, and pose estimation. Then used, the LSTM is to identify the correct sports activity based on the correct sequence of movement present in the frame number and separate link sub-activity to predict the right one. They can classify the sub-activities with 80% [10].

Singh et al. model use two deep-learned architectures. First is the pre-trained CNN and Bi-LSTM to extract the discriminative features from the given RGB frames, and the second one is pre-trained CNN tuned with fully connected layers on the input, which is a dynamic moving image for the recognition of single, multi-person, and human–object interaction (HOI) activities in the video sequence. The CNN-Bi-LSTM combination classifies the activity classes with better recognition accuracy. On the other hand, the dynamic images are used to boost the prediction with the CNN-LSTM (SoftMax layer) on the activities with high motion. The prediction accuracy is computed on four publicly available video benchmarks such as SBU (98.70%), MIVIA (99.41%), MSR Action Pairs (98.30%), and MSR Daily Activity (94.37%) [11].

Based et al. used a deep convolutional neural network-based residual network model to extract discriminating visual-spatial features and an LSTM neural network to deal with the long-term temporal features of the performed action. Their approach was validated on two benchmark datasets and got good results with an accuracy of 91.18% for the CAD-60 dataset, and an accuracy of 91.56% for the MSRDailyActivity3D dataset [12].

Domingo et al. approach used 3D convolution networks. They created a model that correctly combines several recurrent networks with processed data from image feature extraction, object detection, and the skeletal layout of people—then integrated these three techniques to improve the recognition of specific actions by taking the benefit of the best result obtainable by each of the methods. On the STAIR dataset, where they selected 78 classes and 64,282 videos, the accuracy for CNN was 73.7%, and 3DCNN was 76.5%. Additionally, using SVM with the model, they obtained an accuracy of 87.3% [13].

Hassan et al. proposed a Deep Belief Network (DBN) for activity training and recognition. This was compared with the traditional multiclass SVM algorithm. The dataset has twelve different physical activities.7767 and 3162 events were used for training and testing activities, respectively, and every event consisted of 561 basic features. And using the model mean recognition rate of 89.61% and an overall accuracy of 95.85%

was obtained. In addition, it has shown its capacity to differentiate between transitional and non-transitional activities [14].

Ankita et al. model, a deep and fully connected neural network with CNN-LSTM for HAR, primarily focused on weighing parameters. The model parameters were 10 epochs, 32 samples batch size, and exposure of 32 windows of data and obtained an accuracy of 97.89% [15].

Baik et al. proposed an action recognition deep DB (bi-directional) LSTM architecture that utilizes the CNN's frame-level deep features for processing it. In-depth features are taken from every sixth frame of the videos, which reduces the repeatability and difficulty. The DB-LSTM network learns the sequential information among frame features. To increase its depth, the multiple layers in the backward pass and forward pass of DB-LSTM are collectively stacked. Using their approach on the UCF-101, YouTube 11 Actions, and HMDB51 datasets, the accuracy obtained was 91.21%, 92.84%, and 87.64% [16].

IV. Problem Statement

Video-based Human Activity recognition is used in the field of computer vision in applications of telemedicine and e-health, such as remote home monitoring of people with disabilities for fall detection, remote mental health care to recognize facial expressions and background, remote tracking of medicine intake to avoid inappropriate use of medicine, etc [17].

A video is an ordered sequence of frames. Each frame consists of spatial information, and the series of those frames contain temporal information. To model both this information and build action recognizers using video classification, CNN-RNN models are proposed. The convolutions are for spatial processing, and the recurrent layers are for temporal processing. The model comprises a Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) variants: simple RNN, GRU, and LSTM [18].

We use a subset of the UCF50 dataset to build models for action recognition in video data. The UCF50 dataset has 50 categories: short video clips of various daily activities such as walking with dog, Salsa, TaiChi, and Horse Race. These are also the four categories used in the dataset.

V. Methods

UCF50 data set contains 50 action categories, but a subset of the UCF50 dataset with only 4 action categories – 'HorseRace', 'SalsaSpin', 'WalkingwithDog', and 'TaiChi' have been used because of computational resource constraints. These 4 categories contained 533 short video clips. For the LRCN model, out of the 426 video clips in the train set, 80:20 is used for Training: Validation when building the model. or Training: Validation. The model is fit on the 107 video clips in the test set.

For the three models InceptionV3 with SimpleRNN, InceptionV3 with LSTM, and InceptionV3 with GRU, out of the 493 action clips in the train set, 80:20 is used for Training: Validation for model building. 40 action clips in the test set have been duplicated to make the test set have 120 video clips. Each model is evaluated on these 120 clips to calculate the test accuracy and test loss. The models are used on a random video from the test set to recognize/predict the action in the video clip. Four probabilities for the four action categories are calculated by the model(s) for the videos; the highest probability is the predicted action category of a particular video [19].

InceptionV3_LSTM compared to other models has the highest number of parameters as it has three gates. The hyper parameters for building the four models are listed in Table I.

TABLE I
Model Hyper Parameters

Hyper Parameters	**LRCN**	**SNN**	**GRU**	**LSTM**
Epochs	70	200	200	200
Batch Size	4	32	32	32
Learning rate	0.001	0.001	0.001	0.001
Trainable weights	73,060	33,348	99,900	133,068

A. Model architectures

The Long-term Recurrent Convolutional Network (LRCN) model is shown in figure3. In our model we are using 20 sequences or frames for a short video clip. We want to feed 20 sequences of images to the same Convolution Network and the outputs from these parallel CNN process to the LSTM. Using a Time Distributed layer, we can apply the same CNN layers to series of inputs and it produce series of outputs to be fed to the LSTM network. The input to the Long-term Recurrent Convolutional Network (LRCN) is 20x64x64x3. For the first layer we have got 16 convolutions on 20 images that are shaped 64x64 with 3 channels (RGB). These 20 images should be processed in an order of the frames. The LRCN model has a "Time Distributed" layer and its purpose is to apply CNN layers (Convolution, ReLU, Pooling, Flattening, Full Connection) to series of input data. This 20-convolution flow will be trained in parallel to detect the activity in the video. We process the images with time notion and process the frames in an order they originally are in. For this we use LSTM. The outputs from these 20 convolution flows will be flattened and fed to the LSTM with 32 neurons. The output layer has 4 neurons and SoftMax activation is used to predict the probabilities for 4 action categories – 'HorseRace', 'SalsaSpin', 'WalkingwithDog' and 'TaiChi'. The class with the highest probability is the predicted activity in the video clip [4] [20].

The second approach is to build a network model based on Inception v3 model and RNN variants. We used a pretrained Inception V3 model from keras. The output of Inception V3 model is fed to the RNN variants - simple RNN,GRU, and LSTM . The three models are Inception V3 with SimpleRNN, Inception V3 with LSTM, and Inception V3 with GRU.

A simple RNN is a fully-connected RNN where the output is to be fed back to input. The architecture of the RNN unit

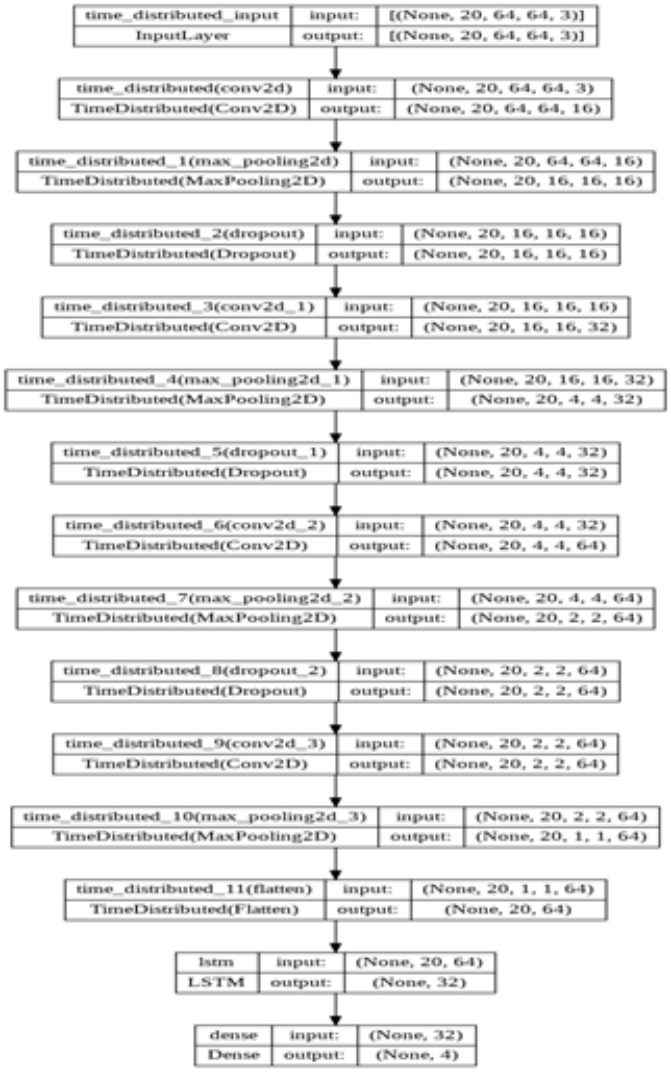

Fig. 3. LRCN model [4]

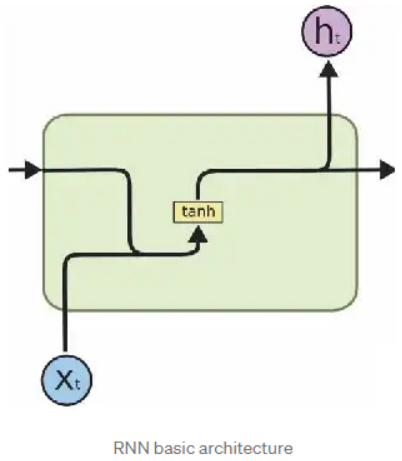

Fig. 4. Simple RNN basic unit [5]

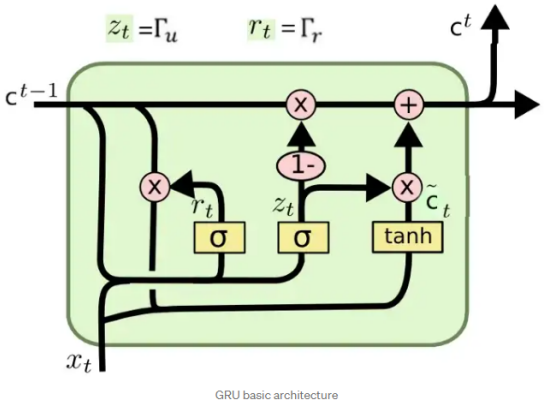

Fig. 5. GRU basic unit [5]

takes input from the previous step and current input and tanh is the activation function(any other activation function can also be used) [8] is shown in figure4.

Formula for SimpleRNN:

$$h_t = RNN_{enc}(x_t, h_{t-1})$$
$$h_t = \tanh\left(W_{hh} * h_{t-1} + W_{xh} * x_t\right)$$

GRU is the same as Simple RNN, but the processes inside the GRU unit differ. GRU unit is shown in figure5. It has two gates 1) reset gate and 2) update gate. Gates are neurons with their weights and biases. Whether the cell state should be updated with the candidate state (current activation value) is decided by the Update gate. Whether the previous cell state is essential or not is decided by the Reset gate. The hidden state(activation) of RNN is the Candidate cell. The Update gate determines the final cell, which may or may not be updated with the candidate state. some information is removed from the previous cell state and passed to the new cell Activation is applied to the final cell state and passed to the next cell. If the reset value is close to 0, previous hidden state information that is irrelevant in the future is dropped. If the update value is close to 1, then a part of the information is carried forward [21].

Input gate, Forget gate, and Output gate are the three gates belonging to an LSTM cell. Basic LSTM unit is shown in figure6. If the information from the previous cell should be carried or if the cell should forget, it is decided by the Forget Gate. If the value of the function is 1, it will carry the information; if the value is 0, the network will forget everything. The Input gate chooses if the new information is important and needs to be transferred next. The new information is erased from the cell state if its value is negative and is added to the cell state if it is positive. A combined function of the current output and the long-term memory (Ct) form the Hidden state. The SoftMax activation is applied to the hidden state to obtain the current output [21].

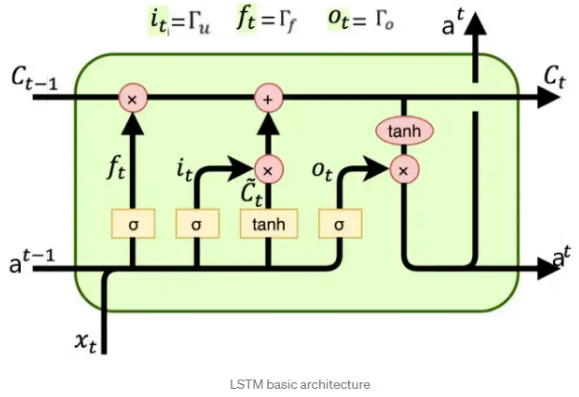

Fig. 6. LSTM basic unit [21]

Formula for GRU:

Candidate cell:

$$\tilde{C}^t = \tanh\left(W_c[\Gamma_r * C^{t-1} + x^t] + b_c\right)$$

Update gate:

$$\Gamma_u = \sigma(W_u[C^{t-1} + x^t] + b_u)$$

Reset gate:

$$\Gamma_r = \sigma(W_r[C^{t-1} + x^t] + b_r)$$

Cell state:

$$C^t = \Gamma_u * \tilde{C}^t + \Gamma_f * C^{t-1}$$

Activation:

$$a^t = C^t$$

Formula for LSTM:

Candidate cell:

$$\tilde{C}^t = \tanh\left(W_c[a^{t-1} + x^t] + b_c\right)$$

Input gate:

$$\Gamma_u = \sigma(W_u[a^{t-1} + x^t] + b_u)$$

Forget gate:

$$\Gamma_f = \sigma(W_f[a^{t-1} + x^t] + b_f)$$

Output gate:

$$\Gamma_o = \sigma(W_f[a^{t-1} + x^t] + b_o)$$

Cell state:

$$C^t = \Gamma_u * \tilde{C}^t + \Gamma_f * C^{t-1}$$

Activation:

$$a^t = \Gamma_o * \tanh C^t$$

The output of Inception V3 463*20*2048 is fed to the Simple RNN. There are two inputs to the Simple RNN Shape of feature input is 463*20*2048 and mask input 463*20 and the output is 463*4. Similarly, for the models with Inception V3 with GRU and Inception V3 with LSTM will have the same dimensions for the inputs and the output is 463*4. That is each video will have 4 probabilities denoting the 4 classes: the highest probability representing the prediction of the activity in the video clip [6] [21].

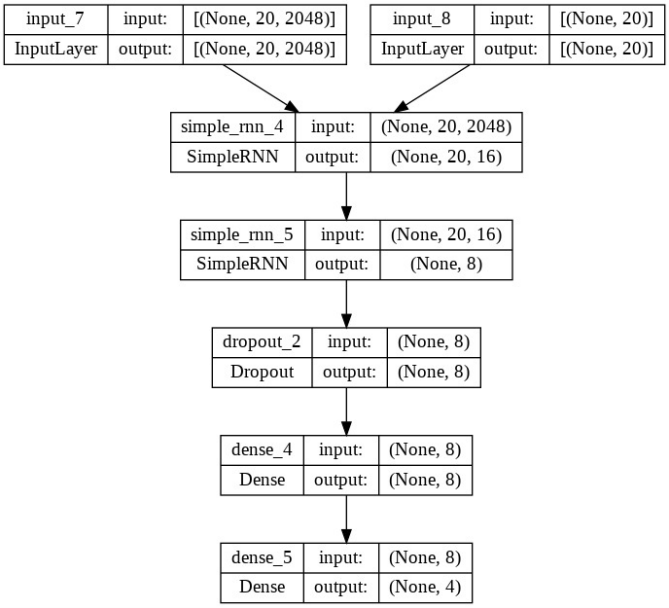

Fig. 7. Simple RNN model; GRU and LSTM have the same layer structure but the number of parameters differ [21]

VI. Results

The model accuracy for Test Data of all the four models is in TableII

TABLE II
ACCURACY FOR TEST DATA

model	Accuracy
LRCN	89.25%
InceptionV3_SRNN	85%
InceptionV3_GRU	92.5%
InceptionV3_LSTM	87.5%

The LRCN model was applied to three YouTube videos given below, the predicted probabilities from the model for each of those videos is also given below

1) Link: 'https://youtu.be/fc3w827kwyA'
The video title is: Comparison of Four Styles of Tai Chi
Action Predicted: TaiChi
TaiChi: 70.88%
HorseRace: 14.44%
WalkingWithDog: 12.66%
SalsaSpin: 2.01%

2) Link: 'https://www.youtube.com/watch?v=tBEc9Kni6I0&ab_channel=PlesniCentarMimbao'
The video title is: AMAZING SALSA Dance With Most Beautiful Sunset View!

Action Predicted: SalsaSpin
SalsaSpin: 84.18%
HorseRace: 14.43%
WalkingWithDog: 0.87%
TaiChi: 0.52%

3) Link : 'https://www.youtube.com/watch?v=wIYD42DV3Ro&ab_channel=NBCSports'
The video title is: Kentucky Derby 2022 (FULL RACE) — NBC Sports
Action Predicted: HorseRace
HorseRace : 99.28%
SalsaSpin: 0.56%
TaiChi: 0.11%
WalkingWithDog: 0.05%

All the three models InceptionV3 plus the RNN variants were tested on a Random video from test set and the predicted probabilities of the activity recognition by each model is given in the below TableIII Random Test video's path is '/content/drive/MyDrive/UCF50sub/test/TaiChi/Copy of v_TaiChi_g24_c02.avi' shown in figure 16

Fig. 8. v_TaiChi_g24_c02.avi video from Test set

TABLE III
PREDICTED PROBABILITIES OF V_TAICHI_G24_C02.AVI VIDEO FROM TEST SET

Model	TaiChi	SalsaSpin	HorseRace	WalkingWithD
InceptioV3_SRNN	95.22%	2.42%	1.79%	0.58%
InceptioV3_GRU	84.29%	0.21%	0.42%	15.08%
InceptioV3_LSTM	89.50%	7.80%	1.65%	1.06%

The train loss and validation loss of LRCN, InceptionV3_SimpleRNN, Inceptionv3_GRU, and InceptionV3_LSTM models are plotted in figure9, 10, 11, 12 respectively

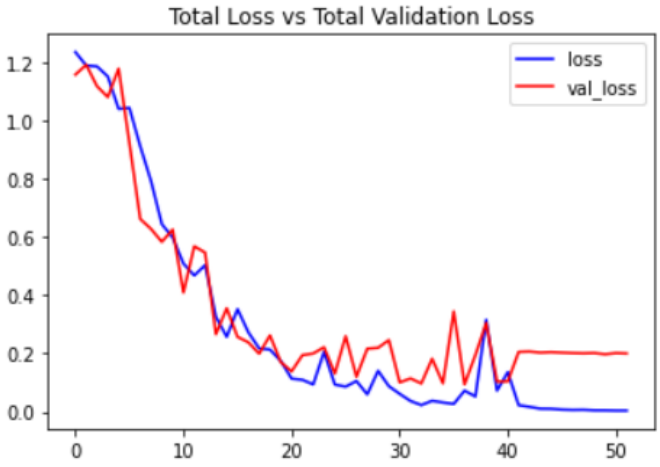

Fig. 9. LRCN Train Loss vs Validation Loss

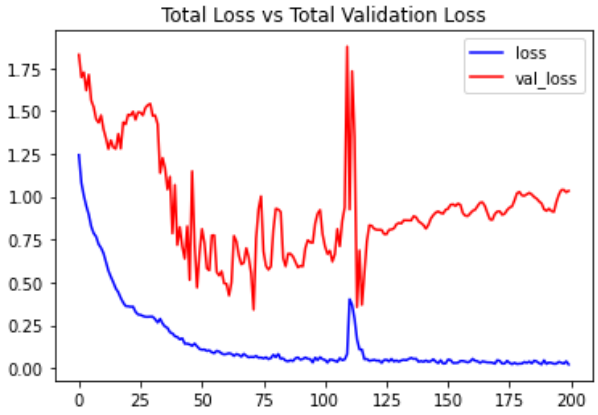

Fig. 10. InceptionV3SimpleRNN Train Loss vs Validation Loss

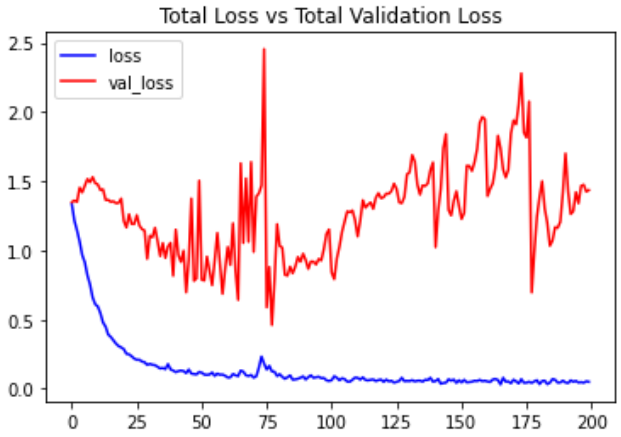

Fig. 11. InceptionV3GRU Train Loss vs Validation Loss

The train accuracy and validation accuracy of LRCN, InceptionV3_SimpleRNN, Inceptionv3_GRU, and InceptionV3_LSTM models are plotted in figures13,14,15,16

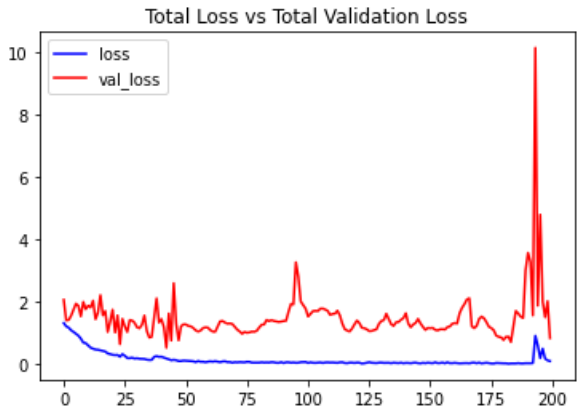

Fig. 12. InceptionV3LSTM Train Loss vs Validation Loss

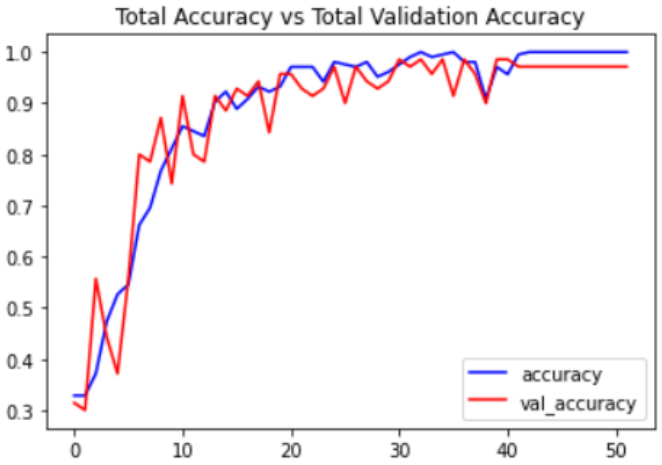

Fig. 13. LRCN Train Accuracy vs Validation Accuracy

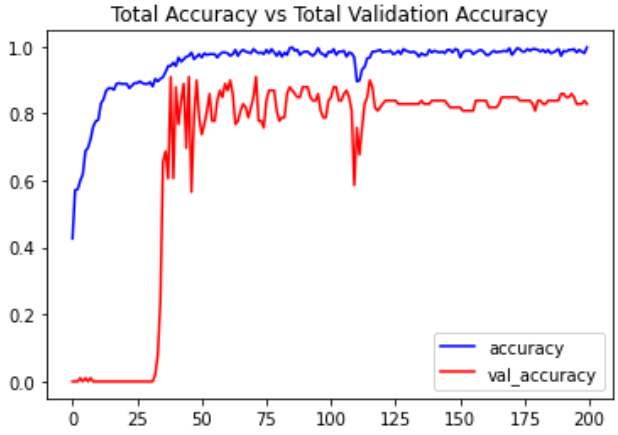

Fig. 14. InceptionV3SimpleRNN Train Accuracy vs Validation Accuracy

VII. Conclusions and Future Work

The training and validation loss, training and validation accuracy of the four models are plotted. The LRCN model has been used to predict/recognise activity on three videos from Youtube and the model gave 70% plus accuracy on those videos. The InceptionV3 with RNN variant models gave an accuracy over 84% when tested on the same video. InceptionV3_GRU model has the highest test accuracy. Only four categories from the dataset were used to build models because of computation. Further experiments can be done using other action categories, other pre-trained CNN variants combined with RNN variants to build different models. The four obtained models can be modified for for real-time activity recognition that is to recognize activity at every frame instead of recognizing one activity for the whole video.

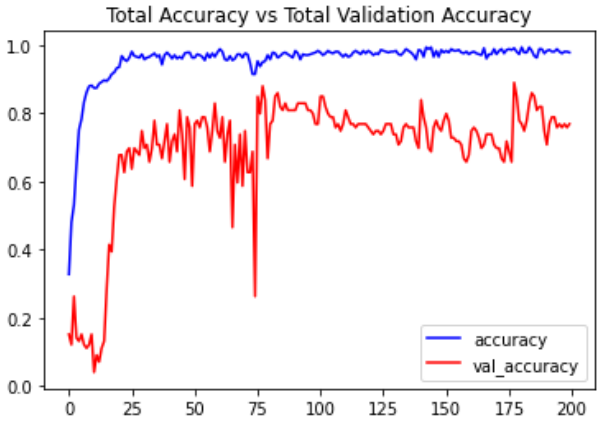

Fig. 15. InceptionV3GRU Train Accuracy vs Validation Accuracy

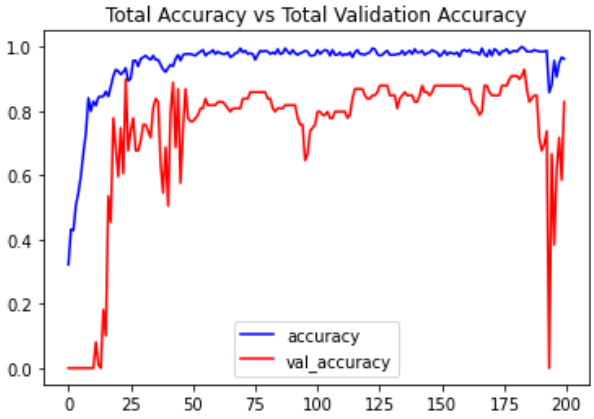

Fig. 16. InceptionV3LSTM Train Accuracy vs Validation Accuracy

References

[1] J. Di and H. Liu, "Research of moving target tracking technology based on lrcn," *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, pp. 789–792, 2017.

[2] V. Sharma, M. Gupta, A. K. Pandey, D. Mishra, and A. Kumar, "A review of deep learning-based human activity recognition on benchmark video datasets," *Appl. Artif. Intell.*, vol. 36, 2022.

[3] H. Shuo and H. Kang, "Deep cnn for classification of image contents," *2021 3rd International Conference on Image Processing and Machine Vision (IPMV)*, 2021.

[4] R. Darelli and P. K. Kollu, "A deep learning framework for human action recognition on youtube videos," 2022.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015.

[6] N. Dong, L. Zhao, C. H. Wu, and J. Chang, "Inception v3 based cervical cell classification combined with artificially extracted features," *Appl. Soft Comput.*, vol. 93, p. 106311, 2020.

[7] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, pp. 971–981, 2012.

[8] W. Ahmad, M. Kazmi, and H. Ali, "Human activity recognition using multi-head cnn followed by lstm," *2019 15th International Conference on Emerging Technologies (ICET)*, pp. 1–6, 2019.

[9] O. Nafea, W. Abdul, M. Ghulam, and M. Alsulaiman, "Sensor-based human activity recognition with spatio-temporal deep learning," *Sensors (Basel, Switzerland)*, vol. 21, 2021.

[10] D. Deotale, M. Verma, and S. P., "Human activity recognition in untrimmed video using deep learning for sports domain," *Social Science Research Network*, 2021.

[11] T. Singh and D. K. Vishwakarma, "A deeply coupled convnet for human activity recognition using dynamic and rgb images," *Neural Computing and Applications*, vol. 33, pp. 469–485, 2020.

[12] H. Basly, W. Ouarda, F. Sayadi, B. Ouni, and A. M. Alimi, "Dtr-har: deep temporal residual representation for human activity recognition," *Vis. Comput.*, vol. 38, pp. 993–1013, 2022.

[13] J. D. Domingo, J. Gómez-García-Bermejo, and E. Zalama, "Improving human activity recognition integrating lstm with different data sources: Features, object detection and skeleton tracking," *IEEE Access*, vol. PP, pp. 1–1, 2022.

[14] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. S. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Gener. Comput. Syst.*, vol. 81, pp. 307–313, 2018.

[15] Ankita, S. Rani, H. Babbar, S. Coleman, A. Singh, and H. M. A. Aljahdali, "An efficient and lightweight deep learning model for human activity recognition using smartphones," *Sensors (Basel, Switzerland)*, vol. 21, 2021.

[16] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[17] V. Japtap, P. Gawande, Y. K. Rathore, A. Rao, S. Oke, and P. Didwania, "Video-based human activity detection," *2022 2nd International Conference on Intelligent Technologies (CONIT)*, pp. 1–5, 2022.

[18] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, pp. 2259–2322, 2020.

[19] C. Yeole and H. Singh, "Deep neural network approaches for video based human activity recognition," 2021.

[20] M. R. Raza, W. Hussain, and J. M. Merigó, "Cloud sentiment accuracy comparison using rnn, lstm and gru," *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–5, 2021.

[21] A. N. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, pp. 235 – 245, 2019.