Early Action Prediction using 3DCNN with LSTM and Bidirectional LSTM

Mrs. Manju D[1] ,
Assistant Professor, Dept.of CSE. GNITS,
Hyderabad, India
s.r.manju@gnits.ac.in,

Dr. Seetha M[2],
Professor & HOD, Dept. of CSE, GNITS,
Hyderabad, India
maddala.seetha@gnits.ac.in,

Dr. Sammulal P[3]
Professor, Dept.of CSE, JNTUH CEJ
Hyderabad, India
sam@jntuh.ac.in

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Predicting and identifying suspicious activities before hand is highly beneficial because it results in increased protection in video surveillance cameras'. Detecting and predicting human's action before it is carried out has a variety of uses like autonomous robots, surveillance, and health care. The main focus of the paper is on automated recognition of human actions in surveillance videos. 3DCNN (3 Dimensional Convolutional Neural Network) is based on 3D convolutions, there by capturing the motion information encoded in multiple adjacent frames. The 3DCNN is combined with Long short team memory (LSTM) and Bidirectional LSTM for prediction of abnormal events from past observations of events in video stream. It is observed that 3DCNN with LSTM resulted in increased accuracy compared to 3DCNN with Bidirectional LSTM. The experiments were carried out on UCF crime Dataset.

Keywords: 3DCNN, Bi-directional LSTM, LSTM, Prediction, surveillance.

1. INTRODUCTION

Due to huge amount of videos in surveillance cameras for security reason requires monitoring systems that can process the videos automatically, detect and identify unusual events beforehand, to be careful when an incident is detected and initiate preventive measures whilst events are detected. Video processing falls under Computer Vision that provides useful tools for applications like video monitoring, smart health care, driver assistance systems, robotics, and autonomous vehicles. The ability to automatically identify suspicious events in long video streams is a key challenge in intelligent video surveillance. Early action-prediction in video frames is also significant because it is linked to topic of computer vision. One of the limitation in surveillance videos is motion prediction. Greater difficulty is encountered due to the fact the start of motion observations from distinct motion classes are generally ambiguous. To get over this limitation, 3DCNN is used.

This paper presents 3DCNN that uses 3D convolutional kernels, which extract spatial temporal features from raw videos. 3DCNN for videos gives outstanding progress in image recognition. Another reason for using 3DCNN is better capturing of the spatio-temporal information in videos, where it captures motion characteristics in videos and shows promising results on video action recognition. The proposed method uses 3DCNN along with LSTM and a Bidirectional LSTM for early action predictions from videos. LSTMs and their bidirectional variants are popular because they have tried to learn how and when to forget and when not to using gates in their architecture. LSTM processes information from inputs that have already passed through it using the hidden state. LSTM stores information of the past because it has seen only the input from the past. Bidirectional LSTM will manage inputs in two ways, one from past to future and one from future to past and it differs this approach from unidirectional LSTM which runs backward to preserve information from the future and using the two hidden states combined, will be able to access any point in time to preserve information from both past and future.

Bidirectional LSTMs show quite good results as they understand the context better. Comparison of accuracy of 3DCNN with LSTM and 3DCNN with Bidirectional LSTM is done.

In this paper, our aim is to focus on the most prominent model of 3DCNN for early action frame prediction from surveillance videos. Early action-detection has primary applications in predicting anomalous events where change in motion pattern and object appearances deviate from ideal patterns. During this process or model, video frames are considered as temporal patterns or time series in contrast to some models were goal of reconstruction is to learn model by reconstructing frames of a video, our approach goal is to model the 3DCNN along with LSTM and Bi-directional LSTM that predicts the action frames similar to that is shown in figure 1. The common problem faced by all these established models is the representation learning which includes feature extraction or transformation of input training data that helps in predicting or detecting an anomaly event.

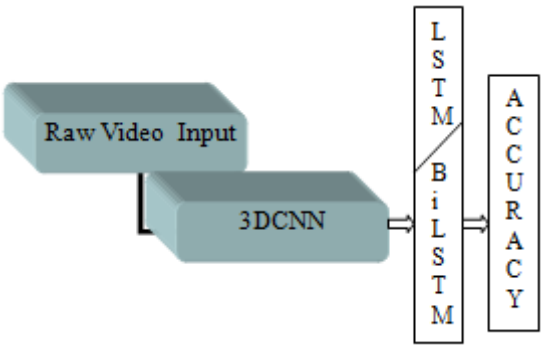


Figure 1. The architecture of the proposed comparison model

As indicated in Fig 1, the raw video input is given as an input to the 3DCNN model. The 3DCNN Model extracts the spatial temporal features from raw videos.

For connecting the data used LSTM and Bi-directional Lstm, for finding the relevant features of spatial data consider the second last layer of the model and extract the features. Those features are given as input to the LSTM as well as to the Bi-directional Lstm. LSTM outperforms compared to traditional RNN, feed-forward neural networks and Bi-directional Lstm takes care of spatial as well as temporal features both in backward and forward direction. As shown in the Fig 1, the input dataset is divided in to a training dataset and test dataset. The test dataset contains 30% of the videos of the whole dataset whereas for training used 70% of the dataset. The loss function indicates how good model predicts; if model prediction is good then the loss value is less otherwise loss value is more. In the test phase, based on the correct predictions the accuracy value changes. The loss function used is Mean Squared Error and Optimizer is Adam. Metrics used are accuracy, MAE, MSE, RMSE.

The following Section 2 provides an overview of existing methods, followed by proposed work, implementation results and conclusion in Section 3, 4 and 5 respectively.

2. Literature Survey

Many authors have carried out work on different models and used different datasets in early action prediction, where work can be focused on spatio-temporal features to improve accuracy by recognizing the actions accurately by which prediction can be done efficiently. This paper focuses on predictive temporal LSTM for spatial temporal predictions [16]. Multi-stage LSTM, takes in to consideration features of context aware and action-aware, introduced a novel loss function for early prediction [1]. Used Lstm model to fine tune the losses [2]. The paper focuses on the localization of human action for spatio-temporal movement during a look at fencing video from UCF-101-24 dataset. For classification used CNN model and tend to style an inspired, on-line model for action label construction from single shot multi-box detector which resulted in performance. Two important advances are presented in the paper. First, used CNNs in real-time SSD (Single Shot Multi Box Detector) to lapse and identify bounding boxes that potentially contain area of interest. Secondly, from the SSD frame level detections developed an algorithm which is used to create efficiently action tubes that works on [3]. The main focus is on the identification of early events that have not been completed. To avoid miscellaneous activities and to stop the suspicious activities, behavior prediction is necessary. For this, two dynamic Bagofwords (BOW) and Integral BOW methods were developed that recognize human activities in a sequential order [4]. The frame work used for detecting ongoing action recognition is MSRNN. In this training of the soft label is done which includes executions by partial intervention and subsequence is given during test process [5]. For addressing this issue of action prediction, the generative adversarial network was introduced by increasing the accuracy of observed videos by reducing the distance between partially observed videos to the maximum one [6]. The framework used for detecting ongoing action recognition is MSRNN. In this training of the soft label is done which includes

executions by partial intervention and subsequence is given during test process [7]. The early predictor system called GAN Predictor is used. This method, creates the images using generator of specific GAN model which are registered and physically tested to find if any abnormality is present [8]. A quick and precise deep-learning approach was proposed to perform localization and prediction of real-time action. In order to localize many acts, proposed method that uses convolutionary neural networks. The technique begins with the use of presence and detection of motion using networks known as "you only look once" (YOLO networks) to use a two-stream model to localize and distinguish behavior from RGB frames and optical flow frames [9]. An innovative method for identification and prediction of human behavior is different software for computer vision, including video monitoring, human, home entertainment that needs real time and online approximations. Suggest a method that is used for joint motion data sources for identifying and predicting behavior in online and in real time, linear latent spaces operating. The strategy is based on an approach that Supervised and has dimensionality reduction [10].

3. Proposed Method

The first and foremost thing to be considered in a video is spatial and temporal features that need be extracted from a video and detection of motion changes that occurs in a video. 3DCNN is an intelligent convolutional model with a capability to learn from frames provided during training. A fundamental challenge in video surveillance is human action recognition which means to recognize the activity of a person and the other major challenge is action prediction, which tries to predict the action of a human. In 3DCNN, input is a video and will perform a three-dimensional convolution, the intermediate convolution and pooling layer, they are in 3D shape, which moves forward and gives the classification. In a 2DCNN , the filters the convolutional kernel which is a 2D, scans throughout the image and produces the feature maps, then goes for pooling and then to the fully connected layer finally classification. In 3DCNN, the input is a volume of data which is a 3 dimensional data and kernel filter is also a 3D kernel, the feature maps is also a 3D volume. There are feature map, each feature map is a volume of data. The main problem with 2DCNN is the temporal information is lost. So, 3DCNN is a best approach for classifying and detecting event in videos. Small kernel sizes 3*3 is the most optimal ones. It reuses the weights at multiple places.

3.1. 3DCNN with LSTM

Compared to an image , an video is a stack of frames, for one second of video we get 30 frames per second so for a dataset preeprocessing on video data is done by cropping them to a fixed size length which is problematic because in sequence learning , need to deal with variable length sequences. Therefore to classify a 30 sec video with a 45 sec video.The 3DCNN, take advantage of CNN's convolutional neural networks across different time scales. In this process from each frame CNN helps to extrcat frame features in the video. The output of the action classification is given as an input to the LSTM model. There are 20 layers in this network. Convolutional layers are 12, followed by 5 pooling layers and one layer for FC, LSTM and output. The convolution block with paired with two or three 2D CNN and a pooling layer. It is followed by a dropout layer. The dropout layer has a dropout rate of 25%. Features are extracted using convolutional layer with 3*3 kernel. ReLU activation function is used in the convolutional layer. The input dimension of image is reduced using max pooling layer with 2 × 2 kernels. LSTM is at last to extract time information. The output shape after convolution is (none, 7, 7, 512). The input size of LSTM layer becomes (49,512) due to reshape method. After analyzing the time characteristics, the architecture sorts the video frames through a fully connected layer to predict whether they belong under any of the two categories (Abnormal/Normal). LSTM is an adapted version of recurrent neural networks to solve the problem of vanishing gradient. LSTM has a memory unit. This memory unit encodes the knowledge learnt. It learns when to forget and update hidden states when new information is provided as input. Memory unit functionality is controlled by three gates: input gate(i), forget gate(f) and output gate(o). The update and output functions are defined as below

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + b_f) \quad (1)$$

$$g_t = \sigma(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad h_t = o_t \odot c_t \quad (3)$$

$$\sigma(x) = (1 + e^{-x})^{-1} \quad (4)$$

$\sigma(x)$ is input mapping sigmoid nonlinearity function. W is the matrix representing the parameters of the gates. $\odot$ represents product operation with values of gate. LSTM control multiple gates to mitigate vanishing gradient problem and capture temporal dependencies. The 3DCNN model with LSTM is given in Figure 2.

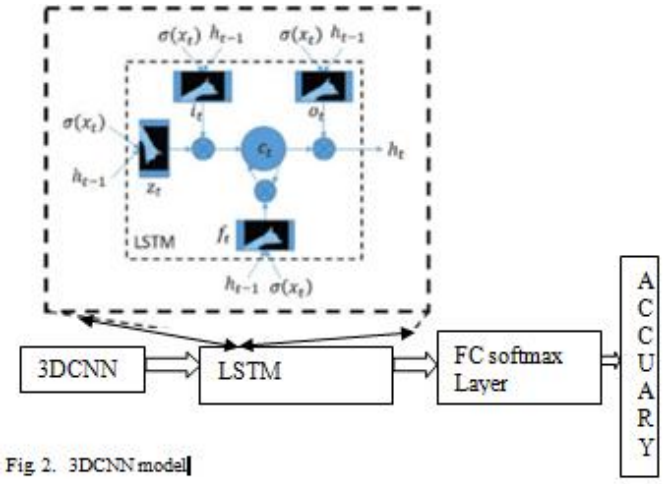


Fig 2. 3DCNN model

Figure2. 3DCNN Model with LSTM

Procedure for 3DCNN with LSTM

- **Load the 3DCNN model in keras**
- **Keras provides an application interface for loading and using pre-trained models.**

1.Get a sample video dataset(Input from user)

2.Load the 3DCNN model

3.Spatial temporal feature extraction

4.Make a prediction(LSTM Model)

5.Accuracy.

3.2. 3DCNN with Bidirectional LSTM

Bi-directional is an extension of a traditional LSTM that improves the model performance on a sequence classification problem. Bidirectional LSTM network connects to hidden layers of opposite direction. In forward LSTM it moves from left to right where as in backward LSTM it moves from left to right and now in a backward LSTM it will pass for each time stamp. Then it concatenates the result of both forward and backward LSTM at each time stamp. Bidirectional LSTMs display very accurate results as they can better grasp the data. Based on the error rate or loss obtained inthe previous iterations, it is the practise of fine-tuning the weights of a neural net. Proper weight tuning means that error rates are smaller, making the model accurate by increasing its generalisation. The procedure is as shown below in the fig 3.

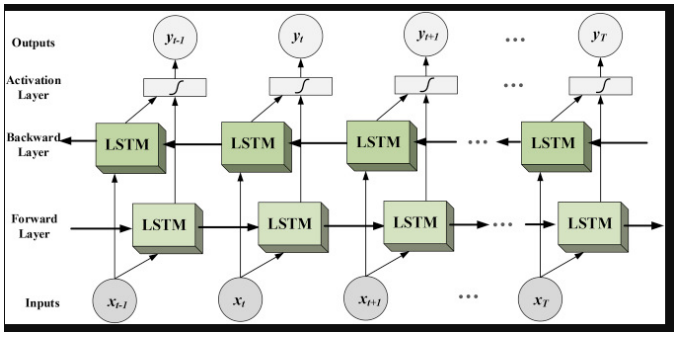


Figure 3. 3DCNN with Bidirectional LSTM

Procedure for 3DCNN with Bidirectional LSTM

- **Load the 3DCNN model in keras**
- **Keras provides an application interface for loading and using pre-trained models.**

1.Get a sample video dataset(Input from user)

2.Load the 3DCNN model

3.Spatial temporal feature extraction

4.Importing Bidirectional LSTM Model

5.Accuracy

Initially start with importing all the libraries and packages to implement 3DCNN with LSTM and Bidirectional Lstm models. From keras.preprocessing.image import img_to_array to transform the frames into arrays. Numpy is used to deal with arrays and has functions for linear algebra, fourier transformation, and matrix domain action. Model groups layers with training and inference features into an object. For creating 3DCNN architecture use the following commands. Load the 3DCNN model in to the variable model and use model.layers.pop() to remove the layers. Given model.inputs and model.layers[-1], output as parameters to model and load into model to get the output of the last layers. For extracting the features define a function that takes video as parameter from the dataset and converts the video into frames. Next frames are resized, converted into array and passed as parameter to pre_process() method for preprocessing. Then input is send to predict() function in to the model for predicting each instance in the array and flattened using ravel(). This is stored in the feature variable. This process is applied for all the frames and features are appended into features variable. Finally get an array of features from this function. Reading videos from the dataset specified by input data directory path and append them to all video file paths list. Then get the count of directories and list of videos from this function. Then import train_test_split method from sklearn to split inputs into random train and test subsets. Take train and test subsets returned from train_test_split method into x_train,y_train and x_test, y_test respectively.

Now, send each video in all_video_file_paths list as a parameter to extract_3DCNN_features method and store the return values (features and frames) into x and train_frame respectively. Append each x to x_train_samples list and each train_frame to train_frames list. By the end of the loop, have all the features of respective frames of all videos in x_train_samples and train_frames. Apply similar process as above for testing and obtain frames as test_frames and their features as xtest_samples, xtrain and xtest values are updated as array of x_train_samples and xtest_samples respectively. After pickling xtrain and xtest as xtrain_samples and xtest_samples consider first element in them as frame and append it to xtrain_frames_list. Using mean function on xtrain_frames_list get xtrain_expected_results. By applying similar process for xtest_samples get xtest_expected_results. For labelling each action uses test_labels dictionary. Each frame in Ytrain give label in accordance with the length of the test_labels. Load sequential model into model and can add LSTM or bidirectional LSTM with dropout 0.7 in one and dropout 0.8 in the other.. Finally, add softmax Activation function and compile with loss as categorical_crossentropy and optimizer as rmsprop. Used batch size as 512 and generate batches of this size using generate method with parameters x_samples and y_samples. Then load previously saved xtrain, xtest ,ytrain and ytest python files into xtrain, ytrain,xtest, ytest using load method. By passing xtrain and ytrain as parameters to generate_batches method get array of batches of xtrain and ytrain hold that in train_gen variable. Similarly for xtest and ytest hold in test_gen variable. Here considered number of echos as 150. After execution, its observed that loss in minimum and accuracy is quite good.

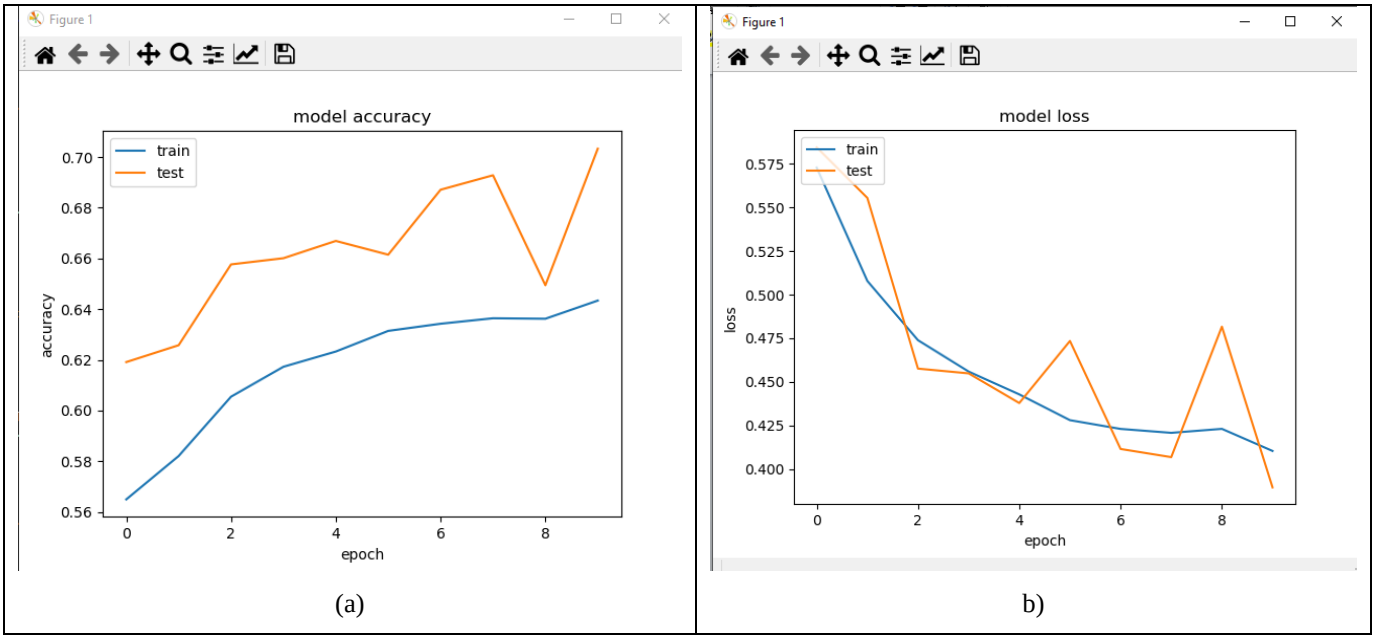
During this process of the working model used train_test_split function and to evaluate the model's output, the testing subset is used for using the model on unknown data to gauge the performance. As discussed the first layer in a Sequential, a linear stack of layers that tells each layer has exactly an input layer and one output layer. Activations may be used either through the activation layer, or through the activation statement that all forward layers endorse. A metric is a feature used to calculate the model performance.

Metric functions are similar to loss functions. Mean Absolute Error (MAE) computes the mean absolute error between the labels and predictions. Mean squared error (MSE) finds the average squared difference between y_true and y_pred and even Root Mean Squared Error computes root mean squared error metric between y_true and y_pred. In the training dataset epochs, represents the number of training iterations which is 150 epochs.

4. **Results and Discussions**

13 different category videos are present in UCF-Crime dataset which is a large-scale dataset consisting of 128 hours of videos. It consists of 1900 long and untrimmed real-world surveillance videos. To start the work used front end as keras and back end as Tensor flow. Keras is an open-source library which provides artificial neural networks with a Python interface. Keras serves as a Tensor Flow library GUI. Keras uses Tensor Flow 2.0's high-level API, an open, highly efficient framework to solve machine learning issues with a focus on modern deep learning. For learning the environment, used LSTM and Bidirectional-LSTM model is used, which in turn helps

for predicting the next frames given the past frames as input and for assuring the quality of frame prediction used Mean Square Error. For the experimentation, epochs are used to measure the MSE of Training and Test dataset. On changing the different parameters like epochs and batch size above 250, best results can be achieved. Batch size tells how many frames are loaded. The loss function used is MSE; it is the simplest of all the errors which is defined as error in the divergence of prediction from actual rating and the activation function used. It is also noted that our method, obtained best accuracy. The graph is for 3DCNN+LSTM

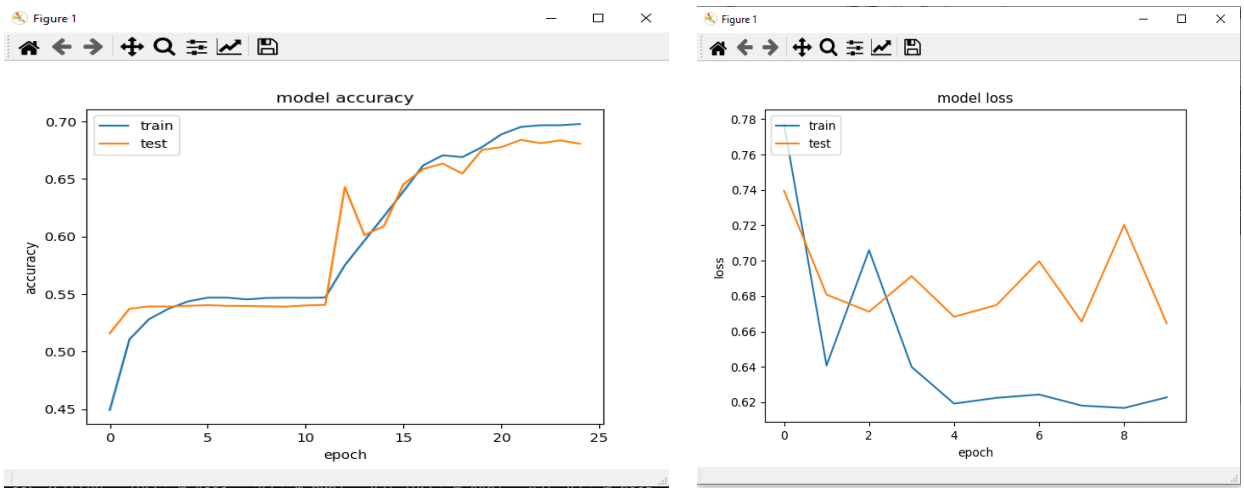


(a) b)

Figure 4. (a) Accuracy of 3DCNN with LSTM, (b) Loss of 3DCNN with LSTM

Table 1. Comparison of Model accuracy in terms of training and test data.

Model	Training		Test	
	Loss	Accuracy	Loss	Accuracy
3DCNN+LSTM	0.41	0.64	0.38	0.70
3DCNN+Bidirectional LSTM	0.60	0.72	0.67	0.68



(a) (b)

Figure 5. (a) Accuracy of 3DCNN with Bidirectional LSTM, (b) Loss of 3DCNN with Bidirectional LSTM

5. Conclusion

In this paper it is observed that 3DCNN learns spatial and temporal information which aims to provide early action-prediction. The performance of the early action-prediction is mainly depending on learning the structures. Therefore 3DCNN model acquires rich feature representation which is combined with LSTM and Bidirectional LSTM for early action-prediction. The 3DCNN model works extremely well in terms of certainty. From the experimental output values, it was shown that the performance of 3DCNN+ LSTM and 3DCNN+ Bidirectional LSTM is based on the quality of recognizing the actions, the probability of predicting the actions and on the epochs. It's observed that more reliable future motions are captured using Bidirectional dynamics in videos. Our experimental work carried out on the UCF-Crime Dataset, has shown that they performed well on the dataset. The loss value can be reduced by increasing the epochs. It is observed that 3DCNN+LSTM model performed well on the test dataset. This work can be extended further with variations of LSTM Models.

References

[1] Mohammad Sadegh Aliakbarian1,3 , Fatemeh Sadat Saleh1,3 , Mathieu Salzmann2 , Basura Fernando1 , Lars Petersson1,3 , Lars Andersson3, "Encouraging LSTMs to Anticipate Actions Very Early BY,ICCV,2017

[2] S. Ma, L. Sigal and S. Sclaroff, "Learning Activity Progression in LSTMs for Activity Detection and Early Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016

[3] Singh, Gurkirt & Saha, Suman & Sapienza, Michael & Torr, Philip & Cuzzolin, Fabio. (2017). Online Real-Time Multiple Spatiotemporal Action Localisation and Prediction. 3657-3666. 10.1109/ICCV.2017.393.

[4] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing

[5] Activities from streaming videos," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 1036-1043, doi:10.1109/ICCV.2011.612 6349.

[6] J. Weng, X. Jiang, W. Zheng and J. Yuan, "Early Action Recognition with Category Exclusion using Policy-based Reinforcement Learning," in IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2020.2976789.

[7] Wang, Dong, Yuan Yuan, and Qi Wang. "Early Action Prediction With Generative Adversarial Networks." IEEE Access 7 (2019): 35795–35804.

[8] J. Hu, W. Zheng, L. Ma, G. Wang, J. Lai and J. Zhang, "Early Action Prediction by Soft Regression," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 11, pp. 2568-2583, 1 Nov. 2019, doi: 10.1109/TPAMI.2018.2863279.

[9] Chenkai Guo, Dengrong Huang, Jianwen Zhang, Jing Xu , Guangdong Bai , Naipeng Dong "Early prediction for mode anomaly in generative adversarial network training: An empirical study". Elsevier,Information Sciences, Volume 534, September 2020, Pages 117-138.

[10] Ahmed Ali Hammam, Mona M. Soliman, Aboul Ella Hassanien "Real-Time Multiple Spatiotemporal Action Localization and Prediction Approach using Deep Learning". Neural Networks Volume 128, August 2020, Pages 331-344

[11] VictoriaBloom, VasileiosArgyriou, DimitriosMakris, "Linear Latent Low Dimensional Space for Online Early Action Recognition and Prediction" Pattern Recognition ,Volume 72, December 2017, Pages 532-547

[12] Mukherjee, Subham & Ghosh, Spandan & Ghosh, Souvik & Kumar, Pradeep & Roy, Partha. Predicting Video-frames Using Encoder-convlstm Combination. 2027-2031. 10.1109/ICASSP.2019.8682158, IEEE 2019.

[13] Wei Henglai, Xiaochuan Yin and Penghong Lin, "Novel Video Prediction For Large-Scale Scene Using Optical Flow".arXiv: abs/1805.12243 (2018).

[14] Liang, Xiaodan, Lisa Lee, Wei Dai and Eric P. Xing, "Dual Motion GAN for Future-Flow Embedded Video Prediction," 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017.

[15] Xu Jingwei, Ni Bingbing, Yang Xiaokang," Video Prediction via Selective Sampling", Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, Pages 1712—1722.

[16] Wang, Yunbo and Long, Mingsheng and Wang, Jianmin and Gao, Zhifeng and Yu, Philip, PredRNN: Recurrent Neural Networks for Predictive Learning using Spatiotemporal LSTMs,volume:30, 2017.