International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Human Crime Based Intrusion Detection by Semantic Features Using LSTM with Inception Deep Learning Approach

Garima Bohra[1*], Dr. Chandra Kumar Jha[2], Dr. Neelam Shama[3]

Submitted: 15/10/2023 **Revised**: 06/12/2023 **Accepted**: 16/12/2023

Abstract: A technique that is based on Deep Learning (DL) is presented here in order to categorize human activities based on the video data. In the area of computer vision, the use of image and video categorization has recently shown considerable progress thanks to the use of convolutional neural networks (CNN). CNN conducts research and analysis on new developments in its network architecture. A method for the classification of human activities is proposed, and its foundations are the structural characteristics of CNNs that have been investigated, as well as the levels of accuracy attained by various architectural configurations during the Image Large Scale Visual Recognition Challenge (ILSVRC). In addition to the spatial correlation that is seen in 2D pictures, the correlation that is seen in the temporal domain is also owned by video data. Incept LSTM is the name of the suggested approach, and it is built on both Inception and LSTM. The approach that has been suggested is capable of accurately recognizing human actions. In addition, the significance of hyper-parameter adjustment has been investigated and applied. The data from the UCF-Crime dataset was used to train and verify the approach that is being suggested. The findings of the experiments provide evidence that the suggested technique is capable of accurately identifying human activities in movies.

Keywords*: *Intrusion Detection, Crime-based Intrusion Detection, Semantic Features, Deep Learning.

1. Introduction

The performance of DL models is very reliant on a number of different hyper-parameters [1]. When compared to more conventional machine learning techniques, deep neural networks have a greater dependence on fine-tuning its hyper-parameters. During the process of learning, the weights of neurons are both initialised and then modified. On the other hand, some hyper-parameters are not capable of being estimated using any data learning technique. Before beginning the training, it is necessary to establish these conditions. The hyper-parameters are the parameters that are used to construct the DL model with or to indicate the technique that is used to minimise the loss function. Hyper-parameters are also known as hyper-parameter settings. Optimizers and activation functions play an important part in reducing the amount of loss that occurs [3, 4]. In order to construct a model that is optimised, it is necessary to investigate a wide variety of potential ranges of solutions. Hyper-parameter tuning refers to the process of discovering the optimal combination of hyper-parameters that will enable a model to provide the highest possible level of performance [5, 6].DL models have higher dependency on hyper-parameter tuning as compared to traditional ML models. Because DL models have more hyper-parameters to tune and also the performance depends on the configuration of hyper-parameters. Literature reports that the DL model accuracy fluctuates from 30-90% due to different selection of hyper-parameters [7, 8, 9]

1.1. Motivation

There are two types of parameters in DL models. First, model parameters which are learned during the training process and second, hyperparameters which are adjustable in nature. In this research paper an empirical approach is used to optimize Incept_LSTM by adjusting the values of these hyperparameters [12, 13]. The error (after minimizing loss) is computed and used for bias-variance adjustment.

2. Proposed Motion Flow Based Incept LSTM Architecture

Deep convolutional model outperforms the traditional approach of feature extraction. However, it has certain limitation [2] like

1) deep models are generally designed by trial-and-error process, which requires large amount of labelled training data [14, 17].
2) Also, large number of neuron connections will result into large computational expense [22, 23].

[1*]*Banasthali Vidyapith University, Jaipur, Rajasthan– 304022, INDIA ORCID ID: 0000-3343-7165-777X*
[2]*Banasthali Vidyapith University, Jaipur, Rajasthan– 304022, INDIA ORCID ID: 0000-3343-7165-777X*
[3]*Banasthali Vidyapith University, Jaipur, Rajasthan– 304022, INDIA, ORCID ID: 0000-3343-7165-777X*
**Corresponding Author: garimajakhar29@gmail.com*

In this part of the tutorial, Incept LSTM will be enhanced and tuned so that it can assess aberrant and normal behaviour in real-world scenarios. The motion flow based Incept LSTM that has been suggested retrieves the spatial information that is included within the RGB frames and also gathers the motion information from the input data that has been provided. The appearance characteristics [34, 33] are stored in RGB frames and must be retrieved by hand as part of the pre-processing operation. The Lucas Kanade optical flow method is used in order to determine the motion flow [18, 19]. Through combined learning of a person's motion and appearance attributes, the motion flow based Incept LSTM that was suggested is able to distinguish between normal and aberrant patterns of behaviour shown by a person. Following the extraction of these characteristics from the Inception v3 model, a data fusion procedure is carried out. Transfer learning allows for the undertaking of such a combination of parameter fusion and combination.

Transfer learning [24, 25] refers to a method in which information gained from completing one job may be transferred to completing another. Image networks that have been pre-trained may be used to successfully do this job. It may be done in two distinct ways depending on your preference. The first method is called the fine-tuning strategy, and it involves using a pre-trained network while simultaneously updating all of the model's parameters in order to complete a new job. Second, an update is made to the topmost layer, which serves as the basis for future forecasts. The process of obtaining features using transfer learning is called feature extraction.

CNN that has already been trained is used as the feature extractor, and the final layer is modified according to the classification problem at hand. Following the collection of all of the features, the following step is to provide the LSTM network with those features. It takes sequential data and extracts the temporal information from that data.

2.1. Feature Extraction using Pre-trained Model

DL models plays a significant role in extracting the features from image/video data. CNN has the capability of learning deep features from the static images. It has ability to learn spatial features present in the individual image frames. Training a DL model for the image representation requires a large volume of training data. Further, requires high computational resources to handle huge data. To resolve this limitation, the proposed model leveraged the concept of transfer learning. In the proposed model, the weights of pre-trained Inception v3 model is used to accomplish the task. Inception v3 is pre- trained model and has shown excellence in the area of image classification. Various other pre-trained model has shown remarkable results in this area. Due to its deep structure and top-5 accuracy score and reduced error rate, it has been chosen for the feature extraction process.

2.2. Motion Information

To understand the human activities from videos, it is required to analyse given data in spatial and temporal domain. As Video is a collection of static images operating in specific temporal range. Further, information in video data is not only processed spatially but sequence of frames is also considered for the correct understanding of the event/scene. This extra bit of motion information makes the problem more challenging.

There is a numerous increase in the number of parameters while considering video data. Therefore, it is a very challenging to design algorithm for detecting temporal structure in the given data with large number of parameters. This task can be accomplished by converting 2D networks to 3D networks to inherent the motion information. But this approach is not computationally effective. Videos are treated as the collection of separate frames and most methods learn features from the image frame only. And most of the classification is also performed frame wise. Maximum voting for one label is considered as the final classification for the video files which is not always true. For e.g., a person throwing a ball in one frame may be misinterpreted by catching a ball in another frame. To understand the importance of the representation, it is required to estimate the motion in between consecutive frames.

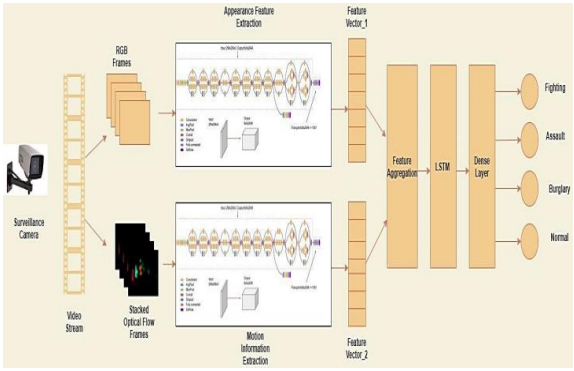


Fig. 1. Schematic Representation of Motion flow based Incept LSTM

To understand and estimate the motion information, optical flow is used. It is widely used in classifying videos at very low computational cost. Optical flow shows the direction of motion of the object in corresponding image frame. It works on the phenomenon of estimating the pixel brightness across the screen over time.

Optical flow estimation, in the proposed model, relies on following two assumptions [41]:

1) Pixel intensity doesn't 't change along the motion trajectory.

2) Motion appears locally as a translation or neighbouring pixel have similar motions.

Variational method is one of the earliest methods of estimating the optical flow. It works on estimating the brightness over consecutive frames. It was the most simple and effective approach given by Horn Schunck (Savian, Elahi and Tillo, 2020). Another approach is given by Lucas Kanade (LK) (Mliki, Bouhlel and Hammami,2020), which works on considering consecutive frames and dividing them into patches of fixed sizes.

2.3. Feature Fusion

Motion and appearance features are fused together using feature level fusion to form a single feature vector. Aggregated feature vector is then passed further for processing. Independent feature vectors are combined together using late fusion (Xu, Yan, Ricci and Sebe, 2017) strategy to form a strong feature vector. The feature level fusion helps model to learn more effectively and jointly using two different or similar kind of modalities. Sequential Learning Using LSTM After extraction of these consecutive time-varying features, LSTM is used to accumulate the dynamic behaviour.

3. Experimental Results and Analysis

In proposed motion flow based Incept__LSTM, Experiments are conducted on segmented videos taken from UCF-Crime dataset. Data collected is pre-processed for further processing. Augmentation techniques are incorporated to enhance the size of the dataset. This method is really helpful when the size of the data is limited. Further, in order to train the model, experiments are carried out to select best hyper-parameters values. With the careful selection of hyper-parameters, model is trained for the designated task. Finally, testing is performed to validate the model.

3.1. Settings in Training and Testing Phase

- Pre-processing

Data is pre-processed before feeding into the model for training. Pre-processing is the essential step for training pro- cess. Not all the categories of UCF-crime dataset are selected rather, due to computational resource limitation this work focuses on only 4 categories. Optical frames are generated using Lucas Kanade algorithm for individual category. RGB and optical frames are used for the training purpose. These extracted frames were then combined in a group of 16 frames each to create one image of dimension 2048 X 128.

Fig. 2. Optical Flow Computation Using Lucas Kanade Method

Fig. 3. Optical Flow Computation Using Lucas Kanade Method

- Hardware and Software Requirements

For training and validation purpose following resources are selected:

Software Resources

- TensorFlow 2.0: TensorFlow 2.0

(https://www.tensorflow. org) is an open-source, free library for developing DL models more efficiently. It is a tight integration of TensorFlow and Keras. It provides a high level API tf.keras to build neural networks and other ML models.

- Python 3.8: Python

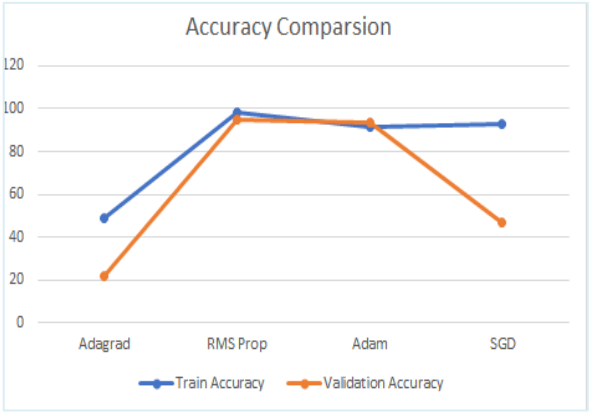
(https://www.python.org/downloads/re lease/python-370/) offers a simple syntax for writing the pro- gram, which makes more comfortable reading and understand-ing of the code and also cost of maintenance is also reduced. It is an interactive language which has capability of interpreting and object-oriented programming concepts. High-level data structures are built-in functionality, supports dynamic linking which helps programmers to build rapid applications.

Other than these two major software tools, some more packages are used to fulfil the need of the problem. OpenCV, sklearn, numpy and pandas are used to accomplish the task.

Hardware Requirements

For this exploratory study, training and validation use avail- able hardware resource with following specifications:

- Intel Core i7 9th Generation. Nvidia
- GTX
- Windows 10 Operating System 16 GB Ram
- Model Training



The training of the model is done under close observation at all times. Each layer that is a part of the architecture has some weight and bias, both of which are initialised with values that are chosen at random. I gave this model a few runs through the training process in order to determine the optimal values for its hyper-parameters. At each iteration of the process, the values of the parameters are varied and the results are monitored. A categorical accuracy metric is used in order to do the accuracy measurement. A loss function is determined by using the cross entropy. The difference between the actual distribution and the probability distribution of a particular random variable may be measured with its help. In this body of work, the cross entropy loss function is used to determine the degree of deviation that exists between the actual value and the value that was anticipated. To do this, throughout the training phase, the value of the weight or parameter will also be modified. The goal is to reduce the effectiveness of the loss function. In the training phase, the batch size is configured to be 4, and the epochs are configured to be 15. In addition, a modest learning rate of 1e-6 has been assigned in order to improve the capability of learning and fitting.

3.2. Analysis

The value of learning rate is set to 1e-6, Leaky ReLU was picked as the activation function, and RMSProp was chosen as the optimizer for the proposed motion flow-based Incept LSTM. These hyperparameter parameters were determined by the optimizer. The influence that Hyper-Parameters have on Model Performance When it comes to the classification model's overall performance, hyper-parameters have a considerable impact. Hyper-parameters that have a major influence on model performance include the kind of optimizer that was employed, the learning rate, and the type of activation function. Altering the settings of the model's hyper-parameters allows for the experiment to be carried out on the UCF-Crime dataset, and the performance of the model can then be assessed.

As a result of the optimizer: The purpose of the calculation is to determine the value of the weights that will result in the lowest possible cost function. Calculating and maintaining an up-to-date version of the network's parameters is the job of the optimizer. It has an impact on the training as well as the output of the model. As a result, it is strongly recommended to choose the optimizer for the training of deep neural networks with great care. A few examples of common optimizers include Adagrad, SGD, Adam, and RMSprop. Table 1 presents the results of experiments conducted to verify the optimizers' performance. Figure 4 illustrates the effects and results of a number of different optimizers that were chosen for experimentation. image 5.4 demonstrates that the model that was trained by the RMSProp optimizer has an excellent fitting effect. This can be noticed by looking at the image. In addition, the slope of the fall is not changing and is instead providing the best stable curve. Because of this, the RMSProp optimizer has been chosen as the one that will be used to train the Incept LSTM model.

Table 1 Performance Comparison on Various Optimizers

Optimizer	*Train Accuracy*	*Train Loss*	*Validation Accuracy*	*Validation Loss*
Adagrad	48.8	1.202	21.66	1.449
RMS	98.2	0.2	94.57	0.3891
Prop Adam	91.8	0.4197	93.37	0.5630
SGD	92.9	0.4793	46.8	0.4835

Effect of Activation Function: An activation function in neural network is responsible for transforming the weighted sum of the inputs to output node. These functions are attached to each neuron in the network and it determines whether it should be activated or not. The decision is made on analysing the neuron ‘s current input, and sent further if it is helpful in predicting the results. Table.2 shows the comparison of using various activation functions. Figure4 and figure 5 shows the accuracy and loss comparison with different activation functions.

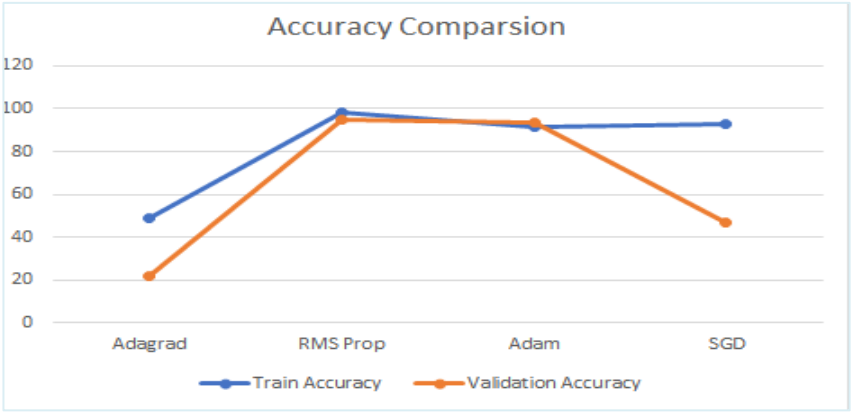


Fig. 4. Accuracy Comparison with Various Optimizers

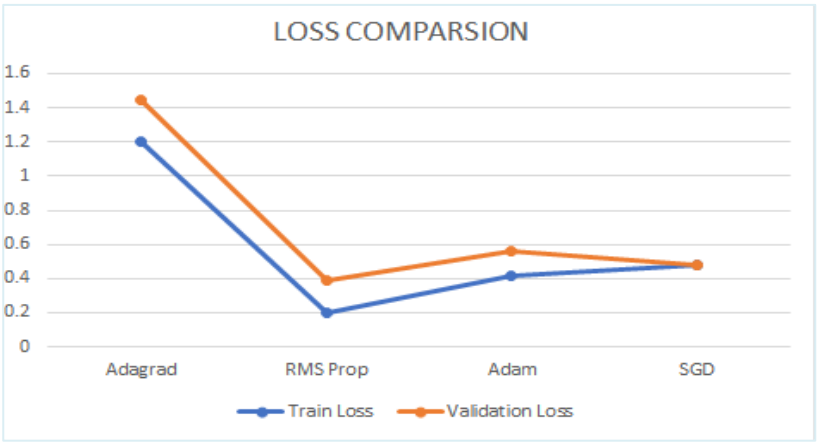


Fig. 5. Loss Comparison with Various Optimizers

Table 2 Performance Comparison on Various Activation Function

Activation Function	*Accuracy*	*Loss*
ReLu	97.6	0.3581
Leaky ReLu	98.2	0.2
Softmax	89.3	0.4547

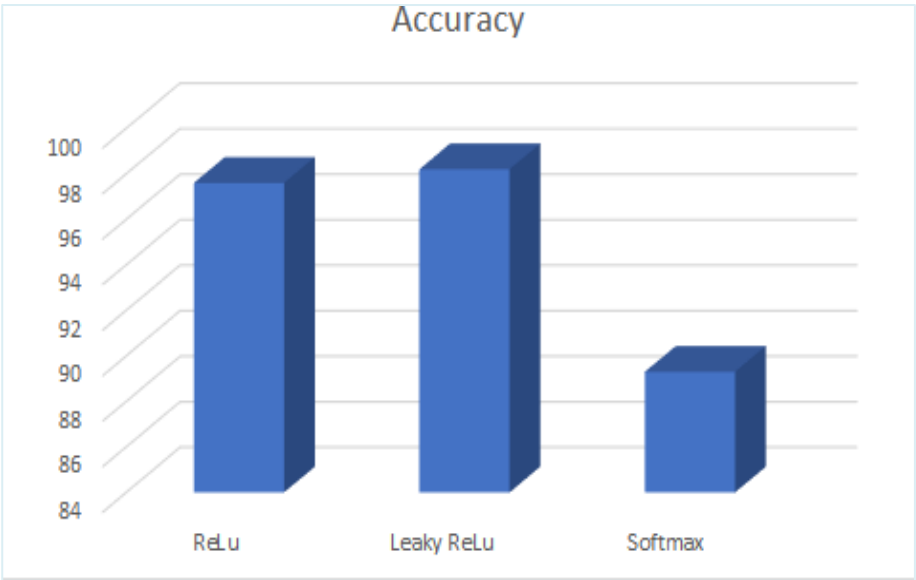


Fig. 6. Accuracy Comparison on Various Activation Function

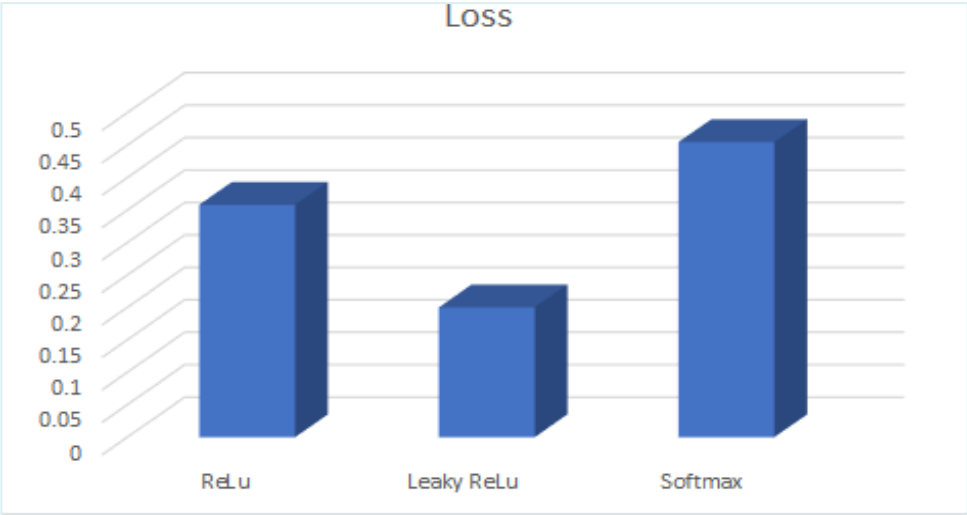


Fig. 7. Loss Comparison on Various Activation Function

Effect of Learning Rate: Learning rate is defined as the step size for parameter update in training process. It is a hyper- parameter that controls the adjustment of the weights with respect to gradient loss. The value of learning rate lies in the range of 0.0 to 1.0. Smaller learning rates requires more training epochs for any update and it may happen that process may stick in between. Whereas, the larger value of learning rate converges at faster speed and produces suboptimal results with a smaller number of epochs. It is very challenging for DL models for the careful selection of the learning rates.

Hyper-parameter tuning is performed to improve the training and testing loss. Instead of using fixed value for learning rate, it is suggested that with each iteration, the value of learning rate needs to be revised until an optimal solution is achieved.

For optimizing the performance of the proposed method, learning rate is tuned and analysed in table 5.3 On empirically testing, the model was able to learn the problem well with learning rates 1e-4 and 1e-6. Selection of too high (1e-2) and too low (1e-8) value of learning rate gives comparatively low model performance on train and test sets. On analysing, it is observed that 1e-6 gives the best possible outcome on train and test sets. Training and validation accuracy is achieved 98.2 and 94.57 respectively, also train and test loss incurred is minimum as compared to other empirically tested values. Figure 5.7 and 5.8 shows the accuracy and loss comparison on empirically set values of learning rates.

Table 3 Performance Comparison on Various Learning Rate

Learning Rate	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
1e-2	85.2	0.5471	32.6	0.975
1e-4	92.15	0.4078	58.21	0.586
1e-6	98.2	0.2	94.57	0.389
1e-8	85.45	0.328	86.18	0.418

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

In this section, the performance of the proposed model motion flow based Incept_LSTM is compared with state of-the-art-methods mainly on UCF-Crime dataset. In recent times, DL models are performing very well, however, with its deep structure, the storage and computational requirement is also increasing.

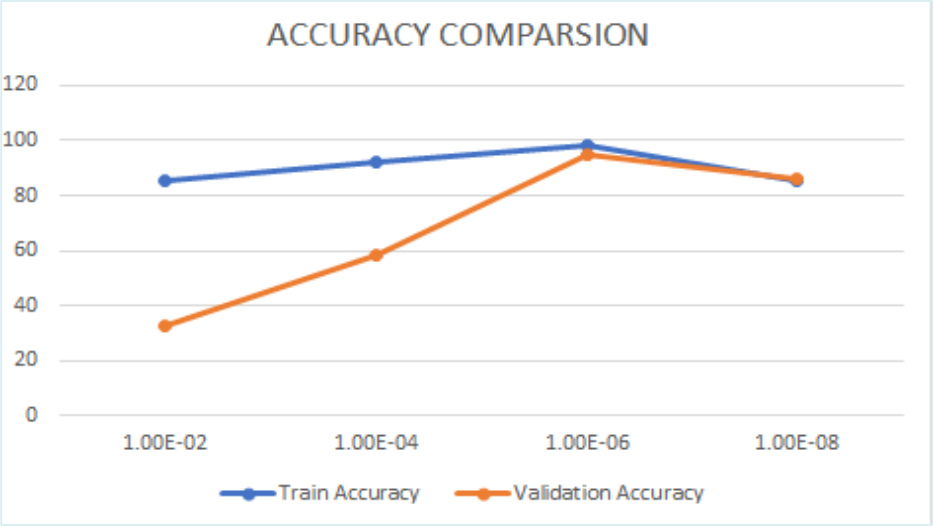


Fig. 8. Accuracy Comparison on Various Learning Rates

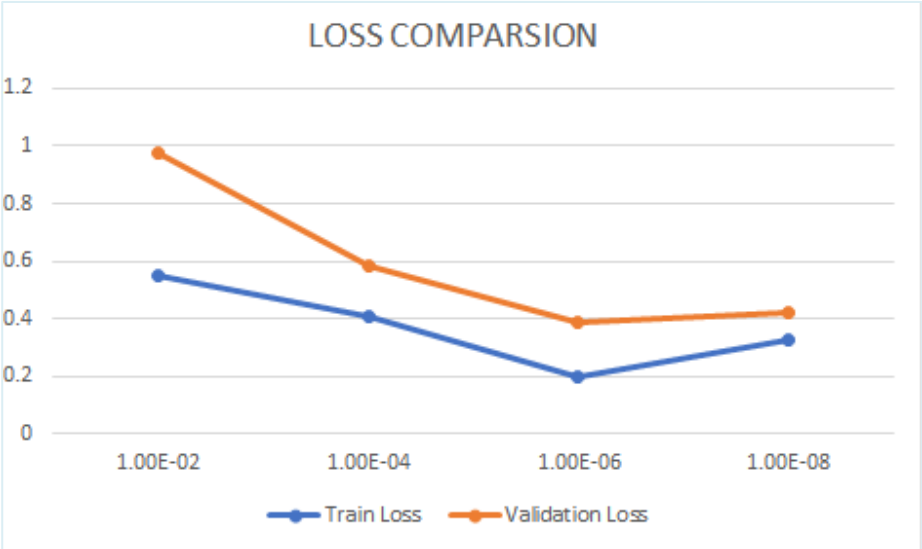


Fig. 9. Loss Comparison on Various Learning Rates

Table 4 Motion Flow-Based Incept LSTM Results

Data set	Epochs	Learning rate	Optimizer	Training accuracy (%)	Training Loss (%)	Validation Accuracy (%)	Validation Loss (%)	No. of output classes
UCF Crime dataset	15	1e-6	RMSProp	98.2	0.2	94.57	0.38	4

In anomaly recognition, a delayed response can cause the loss of human being, and important assets. Therefore, model selection for feature extraction is important for anomaly detection. Table 5 presents the accuracy comparison of the proposed model with the state-of-the-art methods. In the proposed model, Motion flow based Incept LSTM, the number of parameters is 23 million, with a 1.5 GB model size. The time taken to process a sequence of 16 frames is 0.08 seconds with a total training time of 17hrs 13minutes presents the comparison on the number of parameters and processing time taken by the individual model to process a sequence of frames for activity recognition.

Table 5 Accuracy Comparison with State-of-the-Art Methods

Method used	**Accuracy**	**Dataset Used**
DEARESt based on VGG19 + FlowNet for motion flow [17]	76.66%	UCF-Crime dataset
VGGNet+Bi-	81%	UCF-Crime dataset
directional LSTM [43]		
VGG16 +BD- LSTM [30]	82/87.5%	UCF-Crime dataset/UCFCrime2Local
Inception v3+BD- LSTM [30]	80/88%	UCF-Crime dataset/UCFCrime2Local
Resnet50+BD- LSTM [30]	85.53/89.05%	UCF-Crime dataset/UCFCrime2Local
Inception v3+LSTM [32]	88.37%	KTH
CNN(MobileNet)+Atte	78.30%	UCF-Crime Dataset
Proposed Method	**94.57%**	**UCF-Crime Dataset**

Table 6 Comparison of the Proposed Method with State-of-the-Art terms of the Parameters and Time Complexity

Method used	**No. of parameters (in millions)**	**Time Complexity/Per Sequence (in seconds)**
DEARESt based on VGG19 + FlowNet for motion flow [17]	305.49	-
VGG16+BD- LSTM [30]	143	0.22
Inception v3+BD- LSTM [30]	23	-
Resnet50+BD- LSTM [30]	25	0.2
Proposed Method	**23**	**0.08**

5. Conclusion

In this research paper, the baseline Incept LSTM is optimized. To improve the recognition accuracy, the importance of motion flow with video data is analysed and incorporated. The input to baseline model has changed and provided with the more construct, required for the video data. Further, impact of hyper-parameters is studied and analysed. An empirical driven approach is used to optimize Incept LSTM by adjusting the values of these parameters. The impact of learning rate, activation function and optimizers are carefully observed. And empirically observed results are used for the final training purpose. To establish the validity of the results, the final computed results are compared with state-of-the-art accuracies.

References

[1] X. Luo, H. Li, D. Cao, Y. Yu, X. Yang, and T. Huang, "Towards efficient and objective work sampling: Recognizing workers'activities in site surveillance videos with two-stream convolutional networks," *Automation in Construction*, vol. 94, pp. 360–370, 2018.

[2] N. Sarma, S. Chakraborty, and D. S. Banerjee, "Learning and Annotating Activities for Home Automation using LSTM," *11th International Conference on Communica- tion Systems & Networks (COMSNETS)*, pp. 631–636, 2019.

[3] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, chal- lenges, and applications," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 2259–2322, 2021.

[4] N. N. Pandey, N. B. Muppalaneni, S. Peng, H. Huang, Chen, L. Zhang, and W. Fang, "Temporal and spatial feature- based approaches in drowsiness detection using deep learning technique," *Neurocomputing*, vol. 411, pp. 9–19, 2020.

[5] H. A. Imran, U. Latif, J. M. Tomczak, and P. Forre´, "HHARNet: Taking inspiration from Inception and Dense Networks for Human Activity

Recognition using Inertial Sensors," *2020 IEEE 17th International Confer- ence on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, pp. 4555–4562, 2020.

[6] D. Arifoglu and A. Bouchachia, "Activity recognition and abnormal behaviour detection with recurrent neural networks," *Procedia Computer Science*, vol. 110, pp. 86– 93, 2017.

[7] A. J. Suresh and J. Visumathi, "Inception ResNet deep transfer learning model for human action recognition using LSTM," *Materials Today: Proceedings*, 2020.

[8] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, pp. 107 561–107 561, 2020.

[9] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," *Proceedings of the IEEE conference on computer vision and pattern recog- nition*, pp. 6479–6488, 2018.

[10] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," *2016 International Joint Confer- ence on Neural Networks (IJCNN)*, pp. 381–388, 2016.

[11] T. Yu and H. Zhu, 2020.

[12] W. Ahmad, B. M. Kazmi, and H. Ali, "Human activity recognition using multi-head CNN followed by LSTM," *2019 15th international conference on emerging tech- nologies (ICET)*, pp. 1–6, 2019.

[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[14] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," *Proceedings of the IEEE international conference on computer vision workshops*, pp. 142–150, 2015.

[15] C. A. Ronao and S. B. Cho, pp. 235–244, 2016.

[16] L. G. Shapiro, "Computer vision: the last 50 years," *In- ternational Journal of Parallel, Emergent and Distributed Systems*, vol. 35, no. 2, pp. 112–117, 2020.

[17] K. Biradar, S. Dube, and S. K. Vipparthi, "DEARESt: Deep Convolutional aberrant behavior detection in real- world scenarios," *2018 IEEE 13th International Confer- ence on Industrial and Information Systems (ICIIS)*, pp. 163–167, 2018.

[18] N. Sarma, S. Chakraborty, and D. S. Banerjee, "Activity recognition through feature learning and annotations us- ing LSTM," *11th International Conference on Communi- cation Systems & Networks (COMSNETS)*, pp. 444–447, 2019.

[19] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architec- ture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, 2020.

[20] B. Jahne and H. Haubecker, *Computer Vision and Ap- plications: A Guide for students and practioners*. San Dieago: Academic Press, 2000.

[21] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F. Y. Wang, "Driver activity recognition for intelligent vehicles: A deep learning approach," *IEEE transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5379–5390, 2019.

[22] C. Y. Ma, M. H. Chen, Z. Kira, and G. Alregib, "TS- LSTM and temporal-inception: Exploiting spatiotempo- ral dynamics for activity recognition," *Signal Processing: Image Communication*, vol. 71, pp. 76–87, 2019.

[23] S. Dubey, A. Boragule, J. Gwak, and M. Jeon, "Anoma- lous Event Recognition in Videos Based on Joint Learn- ing of Motion and Appearance with Multiple Ranking Measures," *Applied Sciences*, vol. 11, no. 3, pp. 1344– 1344, 2021.

[24] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, pp. 938–938, 2018.

[25] C. Shorten, T. M. Khoshgoftaar, R. C. Staudemeyer, and E. R. Morris, "A survey on image data augmentation for deep learning," *Understanding LSTM–a tutorial into Long Short-Term Memory Recurrent Neural Networks*, vol. 6, pp. 1–48, 2019.

[26] A. Murad and J. Y. Pyun, "Deep recurrent neural net- works for human activity recognition," *Sensors*, vol. 17, no. 11, pp. 2556–2556, 2017.

[27] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, *Human activity recognition using recurrent neural networks. In International cross- domain conference for machine learning and knowledge extraction*. Cham: Springer, 2017.

[28] "Video Surveillance Market Size, Share, Growth — In- dustry Trends, 2026 -

fortunebusinessinsights.com, last accessed on," *Fortune Business Insights*.

[29] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, 2018.

[30] W. Ullah, A. Ullah, T. Hussain, Z. A. Khan, and S. W. Baik, "An Efficient Anomaly Recognition Framework Using an Attention Residual LSTM in Surveillance Videos," *Sensors*, vol. 21, no. 8, pp. 2811–2811, 2021.

[31] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, pp. 9893– 9902, 2019.

[32] S. Begampure and P. Jadhav, "Intelligent video analytics for human action detection: a deep learning approach with transfer learning," *International Journal of Com- puting and Digital System*, 2021.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[34] A. K. Chowdhury, D. Tjondronegoro, V. Chandran, and S. G. Trost, "Physical activity recognition using posterior-adapted class-based fusion of multiaccelerom- eter data," *IEEE journal of biomedical and health infor- matics*, vol. 22, no. 3, pp. 678–685, 2017.

[35] Q. Xu, M. Zhang, Z. Gu, G. Pan, D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Computer Vision and Image Understanding*, vol. 328, pp. 117–127, 2017.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[37] S. Ha, J. M. Yun, and S. Choi, "Multi-modal convo- lutional neural networks for activity recognition," *2015 IEEE International conference on systems, man, and cybernetics*, pp. 3017–3022, 2015.

[38] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamar´ıa, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[39] A. Aghaei, A. Nazari, and M. E. Moghaddam, "Sparse Deep LSTMs with Convolutional Attention for Human Action Recognition," *SN Computer Science*, vol. 2, no. 3, pp. 1–14, 2021.

[40] F. M. Noori, B. Wallace, M. Z. Uddin, and J. Torresen, "A robust human activity recognition approach using openpose, motion features, and deep recurrent neural network," in *Scandinavian conference on image analysis*. Springer, 2019, pp. 299–310.

[41] H. Ye, Z. Wu, R. W. Zhao, X. Wang, Y. G. Jiang, and Xue, "Evaluating two-stream CNN for video classi- fication," *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 435–442, 2015.

[42] B. V. K. Rao, K. S. Gopikrishna, M. Sukrita, and L. Parameswaran, "Activity Recognition Using LSTM and Inception Network," in *Soft Computing and Signal Processing*. Springer, 2021, pp. 119–128.

[43] B. Venkata, K. Raok, Gopikrishnam, and Parameswaran, "Activity Recognition Using LSTM and Inception Network," *Soft Computing and Signal Processing*, pp. 119–128, 2021.

[44] T. Mustafa, S. Dhavale, and M. M. Kuber, "Performance Analysis of Inception-v2 and Yolov3-Based Human Ac- tivity Recognition in Videos," *SN Computer Science*, vol. 1, no. 3, pp. 1–7, 2020.

[45] F. J. Ordo´n˜ez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, pp. 115– 115, 2016.

[46] Z. Weng, W. Li, and Z. Jin, "Human activity prediction using saliency- aware motion enhancement and weighted LSTM network," *EURASIP Journal on Image and Video Processing*, no. 1, pp. 1–23, 2021.

[47] A. B. Sargano, P. Angelov, and Z. Habib, pp. 110–110, 2017.

[48] T. Georgiou, Y. Liu, W. Chen, and M. Lew, "A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision," *Inter- national Journal of Multimedia Information Retrieval*, vol. 9, no. 3, pp. 135–170, 2020.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735– 1780, 1997.

[50] M. Zeng, H. Gao, T. Yu, O. J. Mengshoel, H. Langseth, I. Lane, and X. Liu, "Understanding and improving recurrent networks for human activity recognition by continuous attention," *Proceedings*

of the 2018 ACM international symposium on wearable computers, pp. 56– 63, 2018.

[51] S. Savian, M. Elahi, and T. Tillo, "Optical flow estimation with deep learning, a survey on recent advances," in *Deep biometrics*. Springer, 2020, pp. 257–287.

[52] H. Mliki, F. Bouhlel, and M. Hammami, "Human activity recognition from UAV-captured video sequences," *Pat- tern Recognition*, vol. 100, pp. 107 140–107 140, 2020.

[53] N. Nasaruddin, K. Muchtar, A. Afdhal, and A. P. J. Dwiyantoro, "Deep anomaly detection through visual attention in surveillance videos," *Journal of Big Data*, vol. 7, no. 1, pp. 1–17, 2020.

[54] P. Girdhar, P. Johri, and D. Virmani, "Incept LSTM: Accession for human activity concession in automatic surveillance," *Journal of Discrete Mathematical Sciences and Cryptography*, pp. 1–15, 2020.

[55] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad, M. Saj- jad, and S. W. Baik, "CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks," *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16 979–16 995, 2021.

[56] X. He, K. Zhao, and X. Chu, pp. 106 622–106 622, 2021.

[57] Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.