iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IX **Month of publication:** September 2023

DOI: https://doi.org/10.22214/ijraset.2023.55877

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IX Sep 2023- Available at www.ijraset.com

Assessing the Performance of Video Classification Models in Identifying Violent Actions

Divya Patel[1], Jash Shah[2], Pradhyuman Pandey[3], Neha Katre[4], Prachi Tawde[5]
Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Maharashtra, India

Abstract: ***Manual video surveillance takes time and is prone to human error. Thanks to machine learning and deep learning, video analytics automates these processes. Video classification is an important aspect of video analytics. It is critical to detect violent behavior in recordings for surveillance, crime prevention, and public safety. Video classification algorithms are capable of detecting violent acts in video data. This paper introduces readers to the many approaches to video classification and describes the underlying network structure of the 3DCNN, ConvLSTM, and LRCN models, which are commonly used for video classification. Additionally, the models' implementation results were compared in order to conduct a comparative performance analysis of the models for the task of violent action classification. When it comes to classification, F1-Score, AUC score, and accuracy are useful metrics for evaluating models, and they were compared. We discuss some of the difficulties in violent action classification, as well as some potential future opportunities and new perspectives on how to address them and improve the system.***

Keywords: Video classification, Deep Learning, Violent action detection, spatiotemporal features, long short-term memory.

I. INTRODUCTION

A video is simply a series of multiple still images (called frames) that are updated very quickly to create the illusion of motion. Manual video surveillance to extract relevant information is a laborious and time-consuming process with a high risk of missing vital information due to human error. These processes have been automated through the use of video analytics. Video analytics is used in a variety of industries, including surveillance, transportation, healthcare, sports and security [5]. It makes use of the concepts of computer vision and pattern analysis of video footage. One key aspect of video analytics is Video classification i.e. the machine learning task of identifying what a video represents. For image classification, it has become established that CNN-based methods perform better than the vast majority of cutting-edge handmade features, but it is not yet clear whether the same is true for video classification [3]. In order to train a video classification model, a dataset must be provided that contains examples of each class of interest, such as actions or motions. The main distinction between images and videos is that videos have a temporal structure in addition to the spatial structure found in images there by making videos nothing but a sequence of images that operate at a specific temporal resolution, i.e. frames per second [4]. This means that information in a video is encoded not only spatially (in the objects or people in the video), but also sequentially and in a specific order, for example, closing a door vs opening a door, sitting down vs standing up [2]. This additional information is what makes classifying videos both interesting and challenging. Due to its many real-world uses in surveillance, crime prevention, and public safety, the detection of violent behaviours in recordings has grown in importance. As computer vision and deep learning techniques have improved, video classification algorithms have demonstrated significant promise for identifying violent acts in video data. The classification of violent acts is a crucial area of research for a number of reasons. Law Enforcement represents one of the most direct applications of violent action classification. By correctly classifying violent acts, law enforcement agencies can learn more about the root causes of criminal behaviour and create more efficient plans for reducing crime and bringing offenders to justice. Identifying and classifying violent acts can also contribute to the improvement of public safety. Communities can take measures to reduce the likelihood of violent incidents by analysing patterns of violence and identifying high-risk areas. The dataset, the characteristics retrieved, and the model architecture employed can all affect how well these techniques work. Hence, a comparative study of different video classification algorithms for violent action detection is essential to identify the most effective approach.

II. VIDEO CLASSIFICATION TECHNIQUES

A. Single Frame Classification

Within a single frame CNN Each video frame is treated as a separate image and fed into a convolutional neural network (CNN) model for classification. The CNN model is made up of several filter layers that extract features from the input frames. These features are then passed through fully connected layers to generate a video class prediction [6].

B. Late Fusion

In practice, the Late Fusion method closely resembles the Single-Frame CNN method, but is slightly more complex. The only difference is that in the Single-Frame CNN approach, averaging across all predicted probabilities is done after the network has completed its work, whereas in the Late Fusion approach, averaging (or some other fusion technique) is built into the network itself [7]. Thus, the sequence of frames' temporal structure is also considered. The results of multiple networks trained on different frames at different times can be combined using a Fusion layer. Maximum pooling, average pooling, or flattening are the three most common methods used. This method enables the model to learn both spatial and temporal information regarding the appearance and movement of scene objects [2]. Each stream independently performs image classification, and the final fusion layer combines the predicted scores. [8, 9].

C. Early Fusion

This approach is the inverse of late fusion in that the temporal and channel (RGB) dimensions of the video are fused at the outset before passing it to the model, enabling the first layer to operate over frames and learn to identify local pixel motions between adjacent frames [8, 9]. A video of shape (T x 3 x H x W) with a temporal dimension, three RGB channel dimensions, and two spatial dimensions H and W is fused into a shape tensor after fusion (3T x H x W) [2].

D. 3D-CNN

This approach uses a 3D convolution network that allows you to process temporal information and spatial by using a 3-Dimensional CNN [10,12]. Unlike Early and Late fusion, this method fuses the temporal and spatial information slowly at each CNN layer throughout the entire network such that higher layers get access to progressively more global information in both spatial and temporal dimensions [13]. A four-dimensional tensor (two spatial dimensions, one channel dimension and one temporal dimension) of shape H W C T is passed through the model [11], allowing it to easily learn all types of temporal interactions between adjacent frames.

E. CNN with LSTM

The idea of this approach is to Combine two powerful models - Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. It involves extracting spatial features from individual video frames with a CNN and then capturing temporal dynamics across multiple frames with an LSTM. The basic process of the CNN+LSTM approach for action recognition begins with pre-processing the input video to extract individual frames by passing each frame through CNN to extract spatial features. This is followed by Sequence formation, in which spatial features are combined to form a sequence of feature vectors, each of which corresponds to a single frame of video. Following that is LSTM modelling, which involves feeding a sequence of feature vectors into an LSTM network to capture the temporal dynamics across multiple frames [14]. The LSTM network learns to recognize motion patterns and changes in spatial features over time [15,16]. Finally, the output of the LSTM network is passed through a fully connected layer to generate a class label that corresponds to the action being performed in the video [17].

III. NETWORK ARCHITECURE OF IMPLEMENTED METHODOLOGIES

A. 3D CNN's

The proposed model has three 3D convolutional layers, these layers perform convolutions on the input data using 3D kernels of size (3,3,3). Each convolutional layer has multiple filters that learn during the training process to extract different features from the input data [10]. An activation layer (Rectified Linear Unit (ReLU)) is then added after each convolutional layer to avoid the vanishing gradient problem [20]. Max-pooling layers are used as a dimensionality reduction technique to reduce the spatial dimension of the convolved feature maps [21]. The size of the strides in the pooling layers is set as (2,2,2) and a dropout layer is added after each pooling layer for regularization to prevent overfitting by ensuring that no units are co-dependent with one another [22]. Next, a flattening layer is stacked to transform the input feature space into a tensor of constant length [17]. Subsequently, to densely connect all the activations of the previous layer, we used two consecutive dense layers followed by dropout layers to ensure the regularization of the training algorithm and improve the generalization ability of the model [22]. Adam optimizer was used during the training of the model for batch normalization to reduce overfitting. For the output layer sigmoid activation function is used for binary classification.

B. ConvLSTM

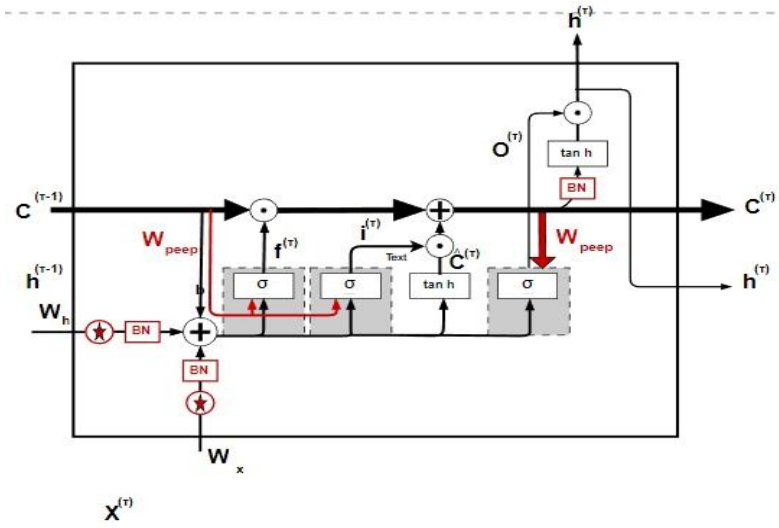


Figure 1 : ConvLSTM Cell

Like the LSTM, a ConvLSTM is a Recurrent layer, but internal matrix multiplications are replaced with convolution operations. As a result, rather than being a 1D vector with features, the data that flows through the ConvLSTM cells retains the input dimension (3D in our case) [19, 23]. Fig. 1 depicts a ConvLSTM cell. The ConvLSTM cell architecture is comprised of four primary elements: the input gate, the forget gate, the output gate, and the cell state. Each of these components includes a convolutional layer that processes both the input data and the previous hidden state. In addition, there are distinct convolutional layers for the gates that determine how much information to remember and how much to forget. The input shape of the model is (length of sequence, height of image, width of image, 3), which is a 4D tensor representing a sequence of RGB images. The first layer of the model is a ConvLSTM2D layer with 4 filters, a kernel size of (3, 3), and a tanh activation function. This layer also has recurrent dropout of 0.2 and returns sequences. The data format is set to "channels_last". The first layer's output is fed into a MaxPooling3D layer with a pool size of (1, 2, 2) and padding set to "same". This layer reduces the data's spatial dimensions while preserving the sequence length [21]. The MaxPooling3D layer's output is then passed through a TimeDistributed layer with a Dropout of 0.2. This layer independently applies dropout regularization to each time step of the sequence with the aim to reduce overfitting [24]. The following two layers are similar to the first, with ConvLSTM2D layers containing 8 and 14 filters, respectively. Both layers have a 0.2 recurrent dropout and return sequences. As before, these two layers are followed by MaxPooling3D layers and TimeDistributed Dropout layers. The final ConvLSTM2D layer consists of 16 filters and return sequences. This layer is followed by a MaxPooling3D layer with "same" padding. After this layer, there is no TimeDistributed Dropout layer. Final probabilities for each output class are obtained by flattening the output of the MaxPooling3D layer and feeding it into a fully connected Dense layer with a SoftMax activation function. The network is trained using end-to-end training, which involves learning the weights of all layers at the same time in order to minimize classification loss.

C. Long Recurrent Convolutional Network

Long-term Recurrent Convolutional Network (LRCN), which combines the CNN and LSTM layers into a single model. In order to model the temporal sequence, the spatial data from the frames are sent to the LSTM layer(s) at each time step using the convolutional layers. In this manner, a robust model is produced as the network directly learns spatiotemporal properties in an end-to-end training [25]. The model is able to independently apply the same layer to each frame of the video by making use of a TimeDistributed wrapper layer. This is very helpful because it allows the entire video to be input into the model in a single shot. In order to construct the LRCN model, the proposed method makes use of time-distributed Conv2D layers, which are then followed by MaxPooling2D and Dropout layers. The feature that was retrieved from the Conv2D layers and flattened by the Flatten layer is then passed to an LSTM layer. The Dense layer with Softmax activation will then use the LSTM layer's output to make a prediction about the subsequent action [24].

IV. DATASET

The model was trained on Violence Detection dataset from Kaggle [18] which consisted of two classes, violent and non-violent. The 180 videos used are divided into two groups: the training/validation set and the testing set. Eighty percent of the sequences are used in the training/validation set, while the remaining twenty percent are used in the testing set. The dataset consisted of 5 subjects performing violent and non-violent activities. Each video was divided into 20 frame sequences to reduce memory and hardware requirements while not losing temporal continuity. The spatial resolution of each input frame was rescaled to 64×64 pixels.

V. IMPLEMENTATION

In this research, a comparison analysis of several architectures is carried out on various model types for the task of classifying videos as violent or non-violent. Training has been performed on the dataset referred to as ViolenceDetection [18]. The models that will be compared are the ones that were built with 3D-CNN, ConvLSTM, and LRCN architectures. The dataset went through some preliminary processing before it was put to use in the training. The video files that make up the dataset are read as part of the process known as pre-processing, and then the frames are scaled to a specific width and height value. Normalization, which entails dividing pixel values with 255 in order to decrease calculations and normalize data to [0,1], speeds up convergence while a network is being trained. This can be accomplished by dividing pixel values with 255. To ensure that only a certain number of frames with an even distribution are added to the feature list, a set sequence length has been defined. This is done in order to cut down on the amount of time needed for calculation while keeping the temporal connectedness intact. After the frames have been pre-processed and retrieved, they are used as input into the models, and further assessments are carried out. Python is used to write the code for the implementation, and the models are validated using a Ryzen 5600H central processing unit and an NVIDIA GeForce GTX 1650 graphics processing unit. For the purposes of testing, a variety of clips from YouTube have been utilized.

VI. EVALUATION PARAMETERS

Given that the problem statement is a classification problem the following parameters have been chosen as the evaluation metrics to obtain a more nuanced understanding of the model's strengths and weaknesses in various aspects of the classification task. This can assist in identifying areas for improvement and fine-tuning the model for improved performance.

A. Precision

A model's precision is the rate at which it correctly identifies violent events, relative to the total number of events that are given that label. A high precision score indicates that the model can detect violent actions accurately while avoiding false positives. Precision is the number of true positives divided by the sum of true positives and false positives.

Precision = T P/ (T P + F P) = T P/All Detections

B. Recall

Recall measures the proportion of correctly identified violent actions out of all actual violent actions in the dataset. A high recall score indicates that the model can identify the majority of the violent actions in the dataset. Recall is the number of true positives divided by the sum of true positives and false negatives.

Recall = T P/ (T P + F N) = T P/All Ground Truths

C. Precision x Recall Curve

For various classification thresholds, the precision values are plotted on the y-axis and the recall values on the x-axis. The resulting curve depicts how the precision and recall values change as the classification threshold is increased or decreased. The curve is a great tool for evaluating the model's performance. If the action classifier accuracy stays high as recall rises, it is said to be an excellent model for the task.

D. F1-Score

F1 score is a measure of a model's precision and recall, and it provides a single score that balances both measures. It is calculated as the harmonic mean of precision and recall, where precision is the proportion of true positive predictions out of all positive predictions, and recall is the proportion of true positive predictions out of all actual positive instances.

F1 score = 2 * (precision * recall) / (precision + recall)

E. Area Under Curve (AUC) Score

The total accuracy of the model's predictions is indicated by the AUC, which measures the model's capacity to distinguish between positive and negative classifications. The AUC ranges from 0 to 1, with 1 denoting a perfect classifier and 0.5 denoting a random classifier.

VII. EVALUATION BASED ON VIOLENCE DETECTION DATASET

For evaluation comparison has been done using three models, 3D CNN, ConvLSTM and LRCN trained on the ViolenceDetection dataset [18].

Table I : Comparison of models based on Precision Recall and F1 score for violent action recognition

Model	Precision	Recall	F1-Score
3D-CNN	0.80	0.71	0.7523
ConvLSTM	0.83	0.77	0.7988
LRCN	0.83	0.74	0.7824

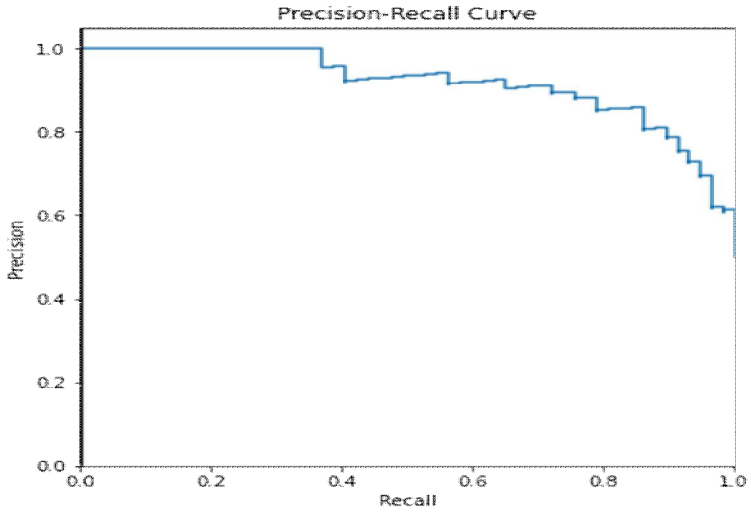


Figure 2 : Precision vs Recall curve for 3D-CNN Model

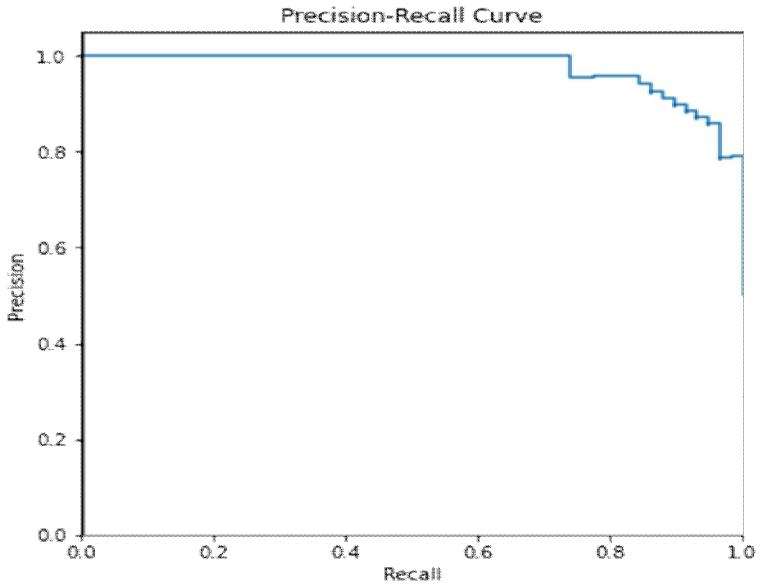


Figure 3 : Precision vs Recall curve for ConvLSTM Model

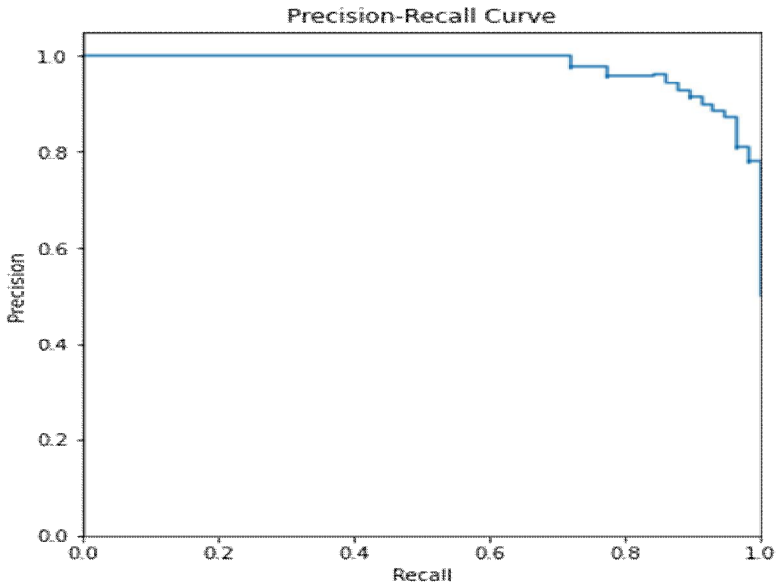


Figure 4 : Precision vs Recall Curve for LRCN Model

Table II : Comparison of accuracy and AUC score of the models for violent action classification

Model	Accuracy	AUC-Score
3D- CNN	85.96	90.61
ConvLSTM	89.47	96.95
LRCN	91.23	97.88

Table III : Ranking of violent action classification models based on Accuracy, AUC score and F1 Score

Rank	Accuracy	AUC-Score	F1-Score
1	LRCN	LRCN	ConvLSTM
2	ConvLSTM	ConvLSTM	LRCN
3	3D- CNN	3D- CNN	3D- CNN

VIII. PERFORMANCE ANALYSIS

Figure 5 : Violent action output

Figure 6 : Non-violent action output

A. 3D CNN

The 3DCNN model has a good accuracy and AUC score, but it is lower than the other two models when trained for the same amount of time. As shown in Fig. 2, the precision value decreases dramatically as the recall value increases, indicating that the model's predictions are becoming increasingly inclusive, i.e., a large number of positive predictions are being made at the expense of more false positive predictions. This indicates that the 3DCNN model is not the optimal model for the task. As can be seen in Table III, the 3DCNN model ranks last across a variety of metrics, which is additional evidence for the forestated.

B. ConvLSTM

The ConvLSTM model is suitable for the task. It outperforms the LRCN model in terms of F1 score, but falls short in terms of accuracy and AUC score. Figure 3 depicts the precision versus recall curve for the ConvLSTM model, which demonstrates that it maintains a high precision value despite a rising recall value, and is therefore a suitable model for the problem under consideration.

C. LRCN

The LRCN model works well for the given task. The model has the highest AUC score and accuracy of the three. It trails the ConvLSTM model by a small margin and ranks second in terms of F1 score. Figure 4 depicts the precision versus recall curve for the LRCN model, which indicates that the model is in good health. The graph is also marginally superior to the graph of the ConvLSTM model and vastly superior to the graph of the 3DCNN model. Overall, the LRCN model is an effective classification system for violent actions.

IX. CHALLENGES FACED

One of the most difficult tasks involved was pre-processing volumetric data or 3D video. Resampling, normalization, and augmentation were all steps in the pre-processing stage that took a lot of time and required a lot of computing power. 3D CNN, LRCN, and ConvLSTM models require high computational resources to train and predict. The high dimensionality of the input data (3D video or volumetric data) increases the number of parameters that need to be learned, which resulted in longer training times and higher computational requirements. Due to the high complexity of models and small size of the training dataset, they were overfitted to the training data, resulting in poor performance on unseen data. Another challenge is the lack of video datasets for violent action classification.

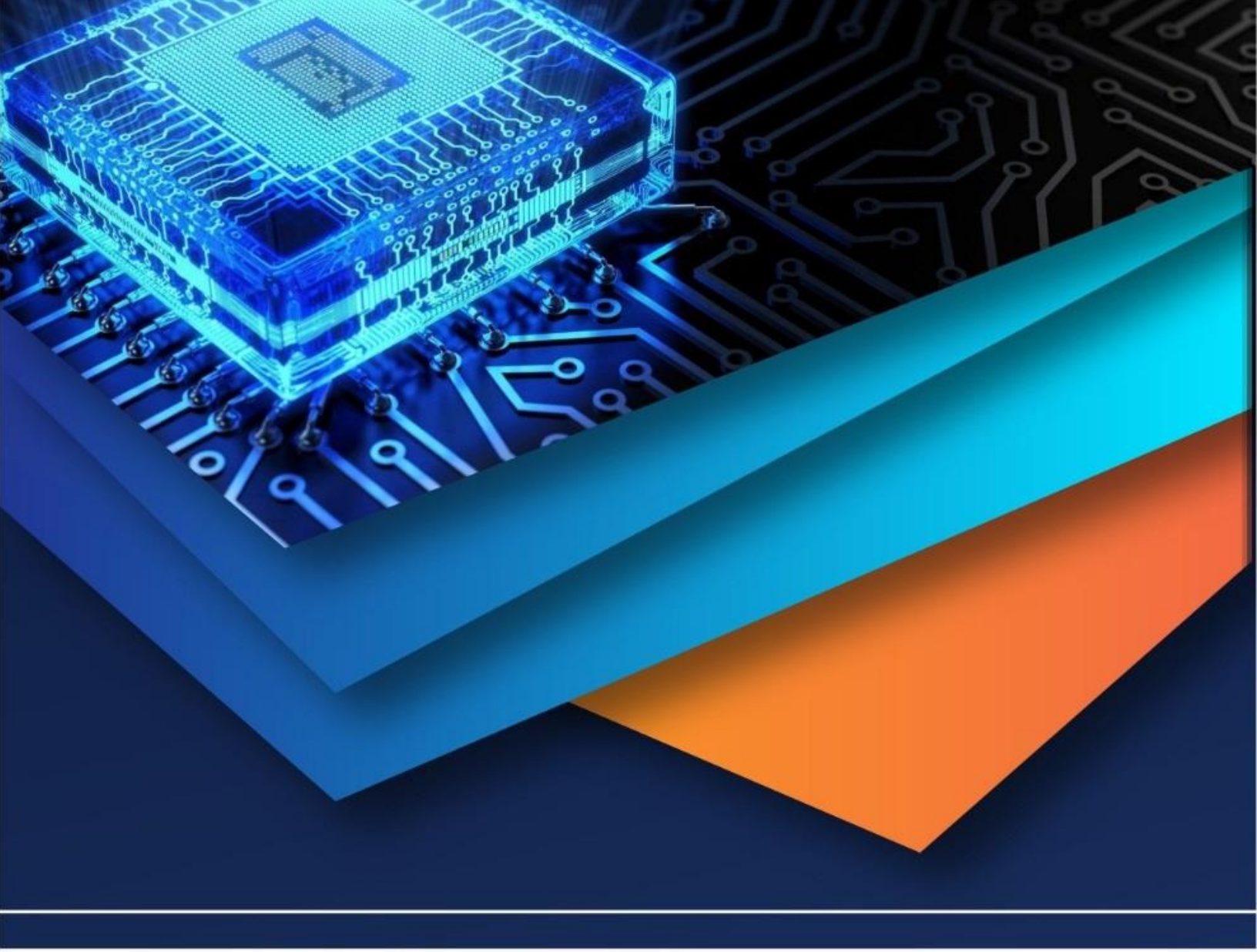
X. CONCLUSION

This paper presented an overview of various video classification methodologies. In addition, a comparative analysis of the 3DCNN, ConvLSTM, and LRCN models used to classify violent action in videos is presented. The discussions lead us to the conclusion that the 3DCNN model is the least suited for the task, whereas the ConvLSTM and LRCN models are suitable but have room for improvement.

Creating an extensive custom dataset of short video clips of various fight action classes, including punching, kicking, etc., that can be used for training is a significant task that can be undertaken. This would significantly improve the model's ability to learn and also improve model accuracy on unseen data.

REFERENCES

[1] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).
[2] Anwar T. (2021) Introduction to video classification and human activity recognition, LearnOpenCV. (BleedAI.com). Available at: https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/
[3] Sharif Razavian, A., Azizpour, H., Sullivan, J. and Carlsson, S., 2014. CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 806-813).
[4] Willems, G., Tuytelaars, T. and Van Gool, L., 2008. An efficient dense and scale-invariant spatio-temporal interest point detector. In Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II 10 (pp. 650-663). Springer Berlin Heidelberg.
[5] Jagad, C., Chokshi, I., Chokshi, I., Jain, C., Katre, N., Narvekar, M. and Mukhopadhyay, D., 2022, May. A Study on Video Analytics and Their Performance Analysis for Various Object Detection Algorithms. In 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET) (pp. 1095-1100). IEEE.
[6] Zha, S., Luisier, F., Andrews, W., Srivastava, N. and Salakhutdinov, R., 2015. Exploiting image-trained CNN architectures for unconstrained video classification. arXiv preprint arXiv:1503.04144.
[7] Liu, D., Lai, K.T., Ye, G., Chen, M.S. and Chang, S.F., 2013. Sample-specific late fusion for visual category recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 803-810).
[8] Snoek, C.G., Worring, M. and Smeulders, A.W., 2005, November. Early versus late fusion in semantic video analysis. In Proceedings of the 13th annual ACM international conference on Multimedia (pp. 399-402).
[9] Boulahia, S.Y., Amamra, A., Madi, M.R. and Daikh, S., 2021. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. Machine Vision and Applications, 32(6), p.121.
[10] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).
[11] Song, W., Zhang, D., Zhao, X., Yu, J., Zheng, R. and Wang, A., 2019. A novel violent video detection scheme based on modified 3D convolutional neural networks. IEEE Access, 7, pp.39172-39179.
[12] Ji, S., Xu, W., Yang, M. and Yu, K., 2012. 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1), pp.221-231.
[13] Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z. and Shi, W., 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4778-4787).
[14] Orozco, C.I., Buemi, M.E. and Berlles, J.J., 2019. Cnn-lstm architecture for action recognition in videos. In I Simposio Argentino de Imágenes y Visión (SAIV 2019)-JAIIO 48 (Salta).
[15] Tee, W.Z., Dave, R., Seliya, J. and Vanamala, M., 2022, May. A Close Look into Human Activity Recognition Models using Deep Learning. In 2022 3rd International Conference on Computing, Networks and Internet of Things (CNIOT) (pp. 201-206). IEEE.
[16] Wang, X., Gao, L., Song, J. and Shen, H., 2016. Beyond frame-level CNN: saliency-aware 3-D CNN with LSTM for video action recognition. IEEE signal processing letters, 24(4), pp.510-514.
[17] Bayoudh, K., Hamdaoui, F. and Mtibaa, A., 2022, January. An Attention-based Hybrid 2D/3D CNN-LSTM for Human Action Recognition. In 2022 2nd International Conference on Computing and Information Technology (ICCIT) (pp. 97-103). IEEE.
[18] Alhaidari, M. (2021) Violencedetection, Kaggle. Available at: https://www.kaggle.com/datasets/mahdialhaidari/violencedetection.
[19] Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K. and Woo, W.C., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.
[20] Tan, H.H. and Lim, K.H., 2019, June. Vanishing gradient mitigation with deep learning neural network optimization. In 2019 7th international conference on smart computing & communications (ICSCC) (pp. 1-4). IEEE.
[21] Wang, P., Liu, L., Shen, C. and Shen, H.T., 2019. Order-aware convolutional pooling for video based action recognition. Pattern Recognition, 91, pp.357-365.
[22] Kavikuil, K. and Amudha, J., 2019. Leveraging deep learning for anomaly detection in video surveillance. In First International Conference on Artificial Intelligence and Cognitive Computing: AICC 2018 (pp. 239-247). Springer Singapore.
[23] Marsiano, A.F.D., Soesanti, I. and Ardiyanto, I., 2019, September. Deep Learning-Based Anomaly Detection on Surveillance Videos: Recent Advances. In 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA) (pp
[24] Anwar, T., Naeem, R., Anjum, M., and Taha Anwar, R. N., & and Rizwan Naeem, T. A. (2021, September 24). Human Activity Recognition using TensorFlow (CNN + LSTM) | Bleed AI. Bleed AI.Available at: https://bleedai.com/human-activity-recognition-using-tensorflow-cnn-lstm/
[25] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY