Vol. 8, No. 1 | Jan – Jun 2024

Comparative Analysis of Pre-trained based CNN-RNN Deep Learning Models on Anomaly-5 Dataset for Action Recognition

Fayaz Ahmed[1], Pardeep Kumar[1], Imtiaz Ali Halepoto[1],Farida Memon[2], Rahila Tallal[3], Danish Munir Arain[4]

Abstract:

Action recognition in videos is one of the essential, challenging and active area of research in the field of computer vision that adopted in various applications including automated surveillance systems, security systems and human computer interaction. In this paper, we present an in-depth comparative analysis of five CNN-RNN models based on pre-trained networks such as InceptionV3, VGG16, MobileNetV2, ResNet152V2 and InceptionResNetV2 with recurrent LSTM units for action recognition on Anomaly-5 dataset. The performance of these models is analyzed and compared in terms of accuracy, precision, recall & F1-scores and computational efficiency. The CNN-RNN architectures we considered for analysis in this paper, the ResNet152V2 based CNN-RNN model exhibits better performance and achieved highest accuracy, precision, recall and F1-score equal to 92.20% due to its ability to capture more complex spatial features. This comparative analysis may guide the researchers in selecting appropriate models for real-world applications for action recognition. In addition of this, a new dataset is developed called Anomaly-5 that can helps as a valuable resource for training and evaluating action recognition algorithms.

Keywords: *Action recognition; Convolutional Netural Network; Recurrent Neural Network; Recision; Recall; Transfer learning*

1. Introduction

Action recognition refers to the process of automatic identification and interpretation of human actions or activities within video frames. This is one of the essential tasks of computer vision that that plays a vital role in many applications such as early detection of suspicious activities in security and surveillance systems[1], [2], monitoring and analyzing patient movements in health care systems[3], [4] , autonomous vehicles for safety of the people and reducing traffic congestion[5], [6], sports analysis for performance evaluation[7],and human-computer interaction for intuitive gesture-based controls[8]. The use of action recognition is extended in robotics to understand human actions by the robots and respond appropriately to human gestures and movements and to detect potential hazardous situations based on human actions[9], [10].

[1]Software Engineering Department, QUEST, Nawabshah, Sindh, Pakistan
[2] Electronic Engineering Department, MUET, Jamshoro, Sindh, Pakistan
[3] Shaheed Zulfiqar Ali Bhutto University of Law, Karachi, Pakistan
[4] Ceptua IT inc., Birmingham, USA

Action recognition is also in educational institutes such as in schools, colleges and universities for monitoring the activities of students and others in terms of their movements, behavior and facial expressions[11].

In these days, deep learning based techniques are widely used for detection and recognition of human actions[12], [13], [14]. This is contrast to conventional machine learning based techniques which heavily depend on manual extraction of features from input data. In deep learning based techniques, the features are automatically extracted from raw data and are directly mapped to desired output without relying on handcrafted features. These techniques not only improve the efficiency and accuracy of action recognition systems but also strengthen their ability on assorted and intricate datasets. However, it is a difficult and a challenging task to design action recognition system that can accurately recognize and classify actions due to variability of human actions, occlusions and environmental factors including lightening conditions. Furthermore, we also need enough quantity and quality of training datasets and proper handling of biases in these datasets when training models.

Convolutional Neural Network (CNN)[15] and Recurrent Neural Network (RNN)[16] are the two popular and most widely used deep learning networks for developing an effective and robust action recognition system. The 2DCNNs are employed to extract features such as edges, colors, textures and other patterns in video frames through convolutional, pooling and activation layers that are crucial for recognizing actions. These nets are frequently employed for the classification of images and can also be employed for classifying human actions in videos by treating each frame individually and independently.

In order to capture temporal relationship between video frames we can use 3DCNN instead of 2D CNN. These networks operate on sequence of frames rather than individual frames as in case of 2DCNNs. The 3DCNNs also used for video based tasks like action recognition due to their ability of capturing dynamics of actions over time. On the other hand, the RNNs are employed for handling sequential or time series data such as voice, video, natural language and other activities.

For action recognition tasks, these networks learn the temporal features between consecutive frames over time to provide perspective and continuousness in the recognition process. A type of RNN called Long Short-Term Memory (LSTM)[17]network is usually used for modeling long term temporal dependencies to enhance the understanding of dynamic aspects in action recognition.

A common and simple approach used for developing action recognition systems nowadays employs a combination of CNN and RNN networks called CNN-RNN. In this approach, CNNs are used for extracting spatial features from individual video frames and then these features are fed into RNNs for modeling temporal dependencies across frames. One of the approach for classification of actions uses two separately trained CNN and RNN models. A CNN model is trained either from scratch or by using transfer learning that allows us to reuse a pre-trained network than has been trained on large datasets for a new task rather than building a network from scratch. Transfer learning has proven to be a valuable approach which not only enhance classification performance of action recognition models as well as it reduces computational resources for training CNN models. The transfer learning also helps to address data shortage issues by training models with limited labeled samples. CNN-RNN models usually beat conventional methods including feature based methods and traditional machine learning models due to their ability of learning hierarchical representation automatically and modeling temporal dependencies from raw video data.

In this paper, we present a comparative analysis of five CNN-RNN models based on pre-trained networks; InceptionV3, VGG16, MobileNetV2, ResNet152V2 and InceptionResNetV2. These networks are chosen due to various factors such as their popularity, robust architectures, widely acceptance for their performance and their proven effectiveness in various image classification tasks. The performance of CNN-RNN models is measured using various metrics such accuracy, precision, recall, F1-score and ROC curves. Several CNN-RNN models based on pre-trained networks can be developed and trained using transfer learning on dataset for action recognition. We selected five models which allows us in depth analysis of the performance of each model on our own dataset containing anomalous actions. This comparative analysis of five selected models can sets a foundation for future research and can be extended by including additional models. Our dataset is specifically developed to capture five anomalous action scenes including road accident and shooting scenes which are often missing in the existing datasets. This offers a crucial resource for advancing research particularly in action recognition and anomaly detection. Furthermore, the comparative analysis in this paper, may guide the researchers in selecting appropriate models for real-world applications for action recognition. The main contributions of this paper are summarized as under.

i. A new dataset called Anomaly-5 is developed for action recognition.

ii. Developed and trained various CNN-RNN models based on pre-trained networks on Anomaly-5 dataset.

iii. Analyzed and compared the performance of developed CNN-RNN models.

2. Literature Review

In this section we discuss some approaches developed by researchers in recent years for action recognition tasks.

In recent studies, the various CNN-RNN architectures are explored for action recognition tasks in-order to improve recognition accuracy and their robustness on temporal variations in video data. One of the innovative works in this arena is the use of two-stream networks, in which CNNs are used for extracting spatial features from individual video frames, and RNNs are employed for modeling temporal dependencies across frames. The earliest two-stream networks proposed by Simonyan andZisserman[18]in 2014 which consist of spatial and temporal streams. This two stream network for action recognition attained innovative prediction results on two HMDB-51 and UCF-101 benchmark action recognition datasets. The Temporal Segment Networks (TSN) was presented by Wang et al.[19]in 2016 that improved action recognition accuracy by exploiting both spatial and temporal attention mechanisms. Using this technique an input video is divided into segments and information from multiple segments is aggregated in-order to capture long-term temporal dependencies. Later on, Zhou et al. [20]in 2018 presented an innovative temporal relation network (TRN) that effectively captures temporal context by modeling pairwise interactions between video frames. This network is not only computationally efficient and also attained significant classification results on various benchmark datasets.

Feichtenhofer et al.[21]in 2019, developed SlowFast networks that operates on two paths; slow and fast for capturing both spatial and motion information at various frame rates. The developed architecture enhances the modeling of motion dynamics in fast moving actions and preserves spatial resolution. Donahue et al.[22]proposed the use of LSTM cells to model temporal dependencies in video frames. These cells are integrated with CNN-RNN architectures to capture long range temporal information between video frames for action recognition tasks.

The recent techniques for action recognition also employed attention mechanisms that selectively focus on information and temporal regions within video frames. These techniques[23], [24], [25]enhance the discriminative power of CNN-RNN models by attending to relevant features. Furthermore, recent studies [26], [27]also explored transfer based architectures to capture long-range dependencies in video sequences and have shown promising results in various action recognition benchmarks.

Kumar and Biswas[28] proposed an approach based on CNN-RNN with fuzzy logic for abnormal human activity detection. The proposed approach recognizes anomalies in video by firstly extracting key-frames from video frames using fuzzy logic. These key-frames are then passed to pre-trained based CNN model to extract spatial features. Finally, these spatial features are fed into an LSTM network to recognize anomalies. The proposed model outperformed on two benchmark UCF-50 and UCF-crime datasets and achieved 95.04% and 49.04% accuracies respectively.

In short, the combination of CNNs and RNNs has achieved significant advancements in action recognition, to develop robust and effective models for analyzing video data. Future research directions may involve exploring novel CNN-RNN architectures, exploiting attention mechanisms, and incorporating combined information for more in-depth understanding of actions in videos.

3. Methodology

The proposed methodology for comparative analysis of five CNN-RNN models for classification of actions based on five pre-train networks is shown in Figure 1. In the initial and crucial preprocessing phase of our project, the dataset is prepared by performing various operations such as splitting dataset, creating frame datasets by extraction of frames and resizing the images into uniform image size according to standard acceptable image size by the pre-trained network. In this step we also apply data augmentation techniques such as random cropping and flipping to enhance the model performance and robustness. After preprocessing, the CNN-RNN model is developed by firstly training CNN model based on pre-trained networks using transfer learning on Anomaly-5 dataset. After training CNN model, we train RNN model on spatial features obtained by applying each of the video in training and validation sets to extract spatial features. The models i.e. CNNs & RNNs; are trained on Google Colab platform which provides us not only a python executable environment as well it provides us free access of GPU for fast training. After training CNN-RNN models, we evaluate the performance of models on test set using several performance metrics such as accuracy, precision, recall, F1-score, confusion matrix and ROC graphs.

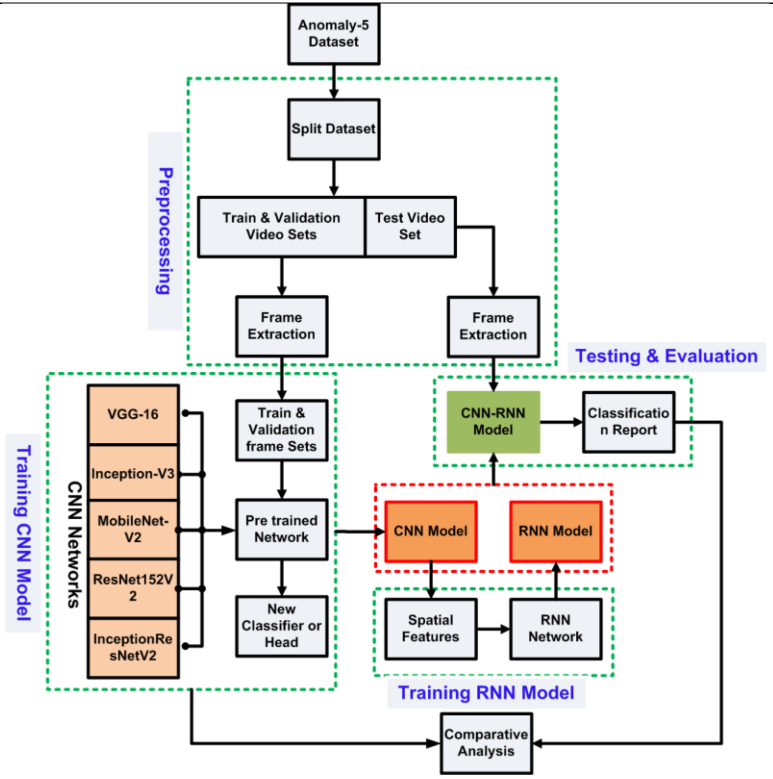


Fig. 1. *Proposed Methodology*

3.1. Description of Anomaly-5 Dataset

The Anomaly-5 dataset contains videos that focuses on anomalies within five distinct categories or classes labeled as Fighting, Fire, Crowd, Road Accident and Shooting Scenes. These classes of scenes involve a diverse range of human activities and incidents, captured from various sources including YouTube, self-recorded and CCTV footages. Each class of this dataset contains 90 videos which vary in size, reflecting the natural variability in duration and quality inherent in real-world recordings. This dataset may serves as a valuable resource for training and evaluating action recognition algorithms, facilitating research and development in the field of computer vision and artificial intelligence. The sample extracted images/frames of each class of dataset are shown in Figure 2.

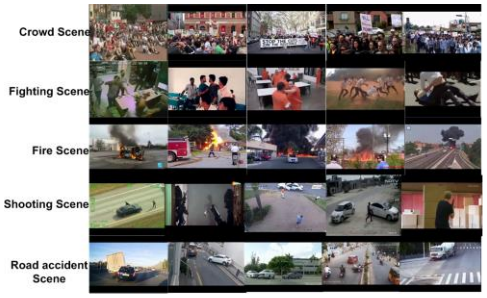


Fig. 2. *Sample Extracted Images of Anomaly-5 Dataset*

3.2. Training CNN-RNN Models

In order to train CNN models on Anomal-5 dataset, we first randomly split this video dataset into training, validation and testing sets by a ratio of 60:20:20 respectively as shown in Figure 3. This split helps us in evaluating the model's generalization and performance accurately. After splitting the dataset, we obtain the frame datasets of these sets by extracting 15 frames/images of each of the videos in these video sets to preserve the temporal information while keeping computational requirements in check.

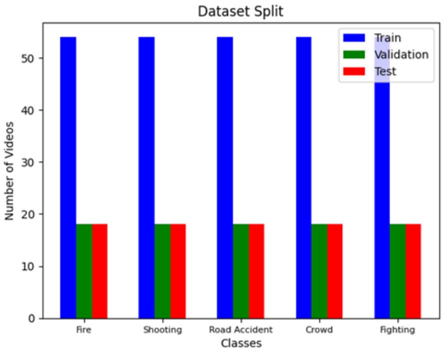


Fig. 3. *Dataset Spilt*

The CNN models are developed by adding custom layers on top of the pre-trained models which include a global average pooling layer, a dense layers of 512 units with ReLUactivation function, a dropout layer with dropout equal to 0.3 to prevent overfitting, and an output layer of 5 units with softmax activation function. The CNN models are then trained using Adam optimizer with learning rate equal to 0.001, categorical cross-entropy loss function, batch size of 32 and 30 training epochs on training and validation frame sets consisting of 2700 and 900 images respectively. The architecture of InceptionResNetV2 based CNN model in terms of number of layers, their types and their connections is shown in Figure 4.

The RNN models developed for learning spatial sequences are based on single LSTM layer with 128 units followed by dense layer and a dropout layer. These models are then trained on Nxfx1024x1024 training and Kxfx1024x1024 validation spatial features where K is the number of videos in training set, N is the number of videos in validation set and f is the number of frames for each video (in our case N=270, K=90 and f=15). The spatial features are obtained by feeding each of the videos in training and validation sets to the CNN models. We use almost same values for hyperparameters as mentioned above for CNN models except number of training epochs equal to 100 for training RNN models. The model structure of RNN model in terms of number of layers, their types and their connections is shown in Figure 5.

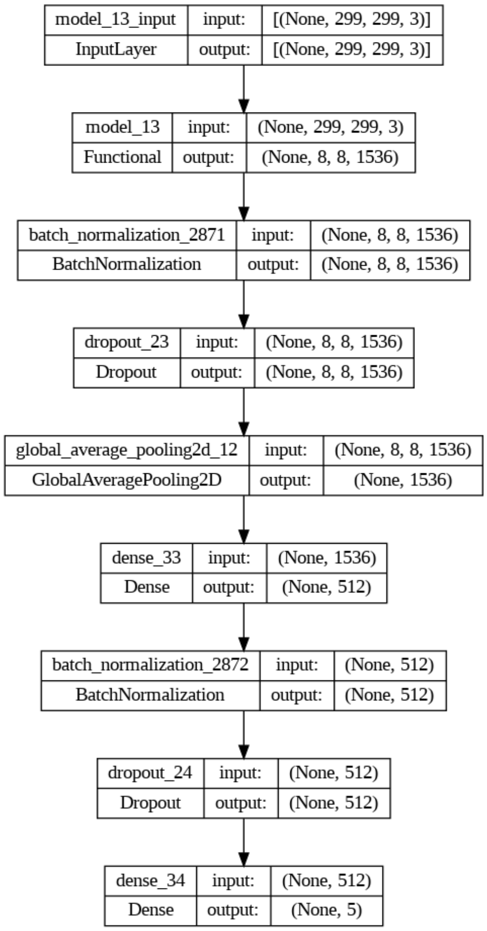

model_13_input | input: | [(None, 299, 299, 3)]
InputLayer | output: | [(None, 299, 299, 3)]
model_13 | input: | (None, 299, 299, 3)
Functional | output: | (None, 8, 8, 1536)
batch_normalization_2871 | input: | (None, 8, 8, 1536)
BatchNormalization | output: | (None, 8, 8, 1536)
dropout_23 | input: | (None, 8, 8, 1536)
Dropout | output: | (None, 8, 8, 1536)
global_average_pooling2d_12 | input: | (None, 8, 8, 1536)
GlobalAveragePooling2D | output: | (None, 1536)
dense_33 | input: | (None, 1536)
Dense | output: | (None, 512)
batch_normalization_2872 | input: | (None, 512)
BatchNormalization | output: | (None, 512)
dropout_24 | input: | (None, 512)
Dropout | output: | (None, 512)
dense_34 | input: | (None, 512)
Dense | output: | (None, 5)

Fig.4. *Model Architecture for InceptionResNetV2 based CNN Model*

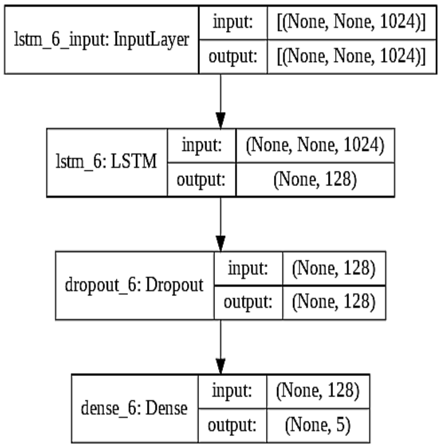


Fig. 5. *Model Architecture for RNN Model*

4. Results and Discussions

The performance and convergence patterns of five CNN models are closely examined through training graphs as shown in Figure 6. These graphs offer valuable insights into how models adopt and learn from the data. All models showed a prompt drop in training loss and steady rise in training accuracy as the training progresses suggesting that the models learnt the training data effectively by capturing the underlying patterns in the data without overfitting and achieved around 99% training accuracy. VGG16 based CNN model is faster and lighter Model and it takes fewer number of parameters as compared to other models, however this model learns slower and exhibits sign of overfitting because its validation loss diverges from training loss as training progresses which indicate a drop of validation accuracy hence achieved lowest validation accuracy.

ResNet152V2 CNN model takes longer training time equal to around 65 mints as compared to other CNN models because this model has many parameters and is larger in size. However this model learns faster as compared to other CNN models. The rising validation accuracies of all models except VGG16 indicate that the models classification or prediction ability is improving as training progresses. In the Table 1, we have provided a comprehensive overview of these five CNN models, showcasing their training times, model sizes, achieved test accuracies, and total parameter counts.

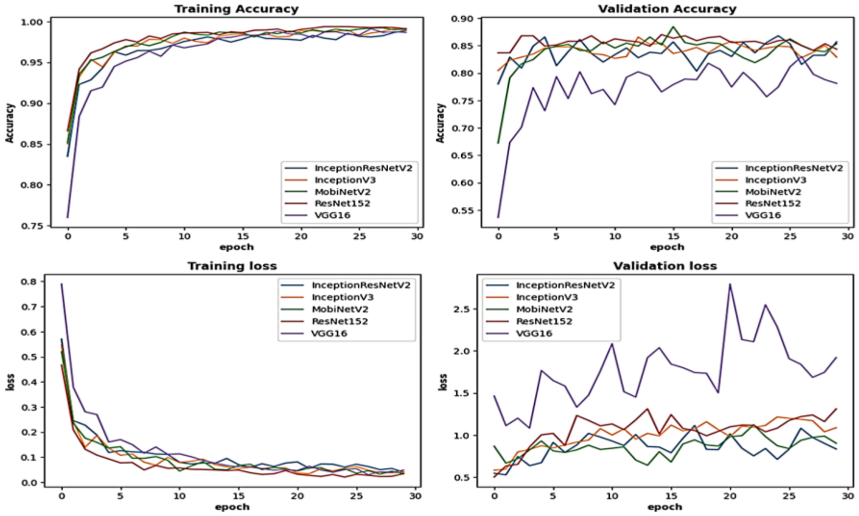


Fig. 6. *Accuracy & Loss of CNN Models*

TABLE. 1. Analysis of CNN Models

CNN Models	Total Training & Non-training Parameters	Training Time (Mints)	Image_Size	Model_ Size (MBs)	Accuracy (%)
VGG-16	15251269	26.0	224x224	69	82.20
Inception-V3	23918373	37.0	299x299	100	82.20
InceptionResNet-V2	55925989	53.0	299x299	221	87.80
ResNet152-V2	60447237	63.0	299x299	240	77.80
MobileNet-V2	3584069	34.0	224x224	119	86.70

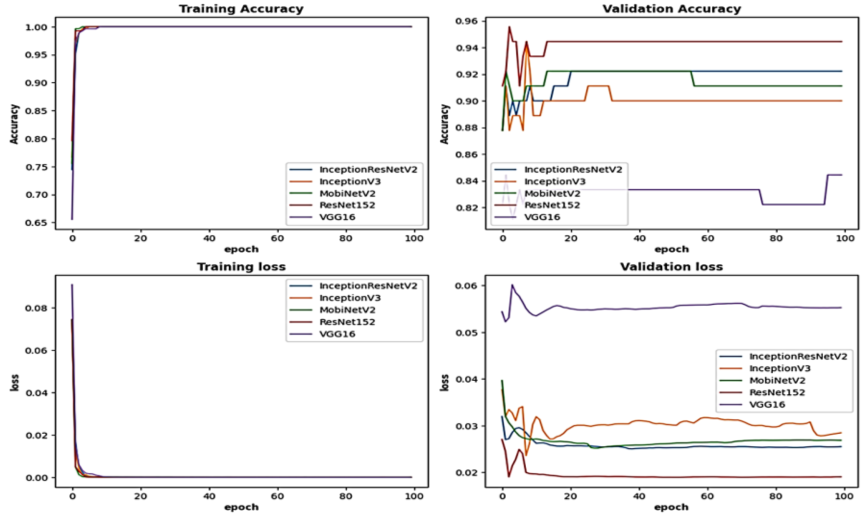


Fig. 7. *Accuracy & Loss of RNN Models*

After comparing the performance of five CNN models based on pre-trained networks, we will examine the performance and training dynamics of five CNN-RNN models. These models are trained on different Nxfx1024x1024 training and Kxfx1024x1024 validation spatial features extracted by using five CNN models as discussed earlier. Figure7 depicts the training graphs of CNN-RNN models presenting a visual representation of their training progress and performance trends. From the Figure7 it can be seen that, the training graphs for all CNN-RNN models show stability in both training loss and training accuracy throughout the training process. Furthermore, the validation accuracies and losses remain mostly steady throughout training process, except at the beginning there are apparent variations in validation accuracies. This suggests that the CNN-RNN models are robust and consistently maintaintheir performance, with only minor fluctuations observed in the validation metrics. The ResNet152V2 based CNN-RNN model attained maximum validation accuracy near to 96 % and InceptionV3 based CNN-RNN model attained lowest validation accuracy that is equal to 91%.

In order to check how effectively CNN-RNN models classify and distinguish different classes within the dataset, we also visualize the performance on test dataset by examining their normalized confusion matrices and classification reports as shown in Figure 8 to Figure 12.The diagonal of these normalized matrices represent the percentage of correct classifications for each class. In classification report, the precision tells us how accurate the model is when it predicts a positive class whereas recall tell us how well model predicts all the positive instances and F1 score balances precision and recall. F1score is a very useful metric when we need to consider both true positive and false negatives in evaluation of the model. The performance of ResNet152V2 based CNN-RNN model is shown in Figure 8. From Figure8 it can be seen that, the crowd scene class standout with a perfect accuracy rate of 100%, indicating the model's exceptional ability to perfectly recognize the samples belonging to this class. For Fire and Road accident scenes classes, the model also achieved significant accuracy equal to 94%. However, on the other hand, for Shooting scene and Fighting scene classes, the model achieved lower accuracies, due to resemblance in these classes. The accuracies in these classes can further be improved by collecting and providing more training samples of these classes. The model attained overall significant accuracy that is equal to 92.2%. Besides this, the model achieved higher precision, recall and F1 score values which show that the model is performing well in terms of both accuracy and its ability of correctly identifying positive samples. The performance of this model is also assessed using ROC plot as shown in Figure 13. The ROC curves in this plot shows that this model exhibit outstanding classification performance and 99% AUC values of most of the classes indicate that this model is highly effective in distinguishing between positive and negative classes. The higher AUC values of the classes and closeness of the curves to the top left corner also suggests that the model has a strong capability to discriminate between true and false positives which means that the model makes very few misclassifications and has a high degree of accuracy. The ROC plots for remaining four CNN-RNN models are shown in Figure 14. The higher AUC values and closeness of the curves to the top left corner in these plots suggest the better performance of these models.

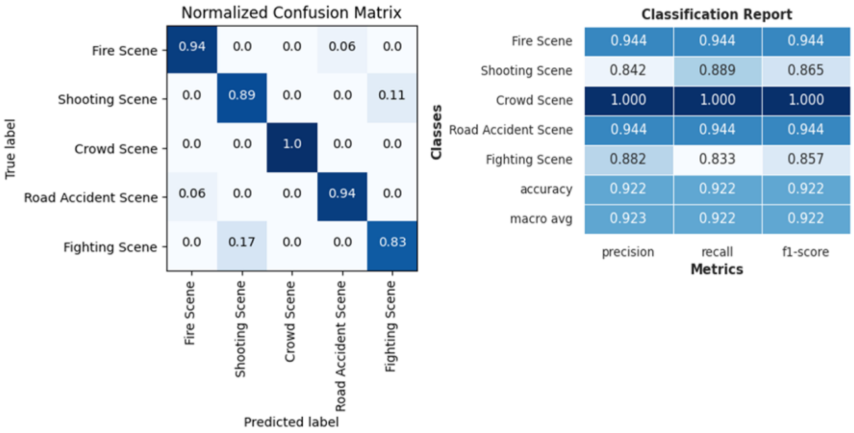


Fig.8.*Performance Analysis of RestNet152V2 based CNN-RNN Model*

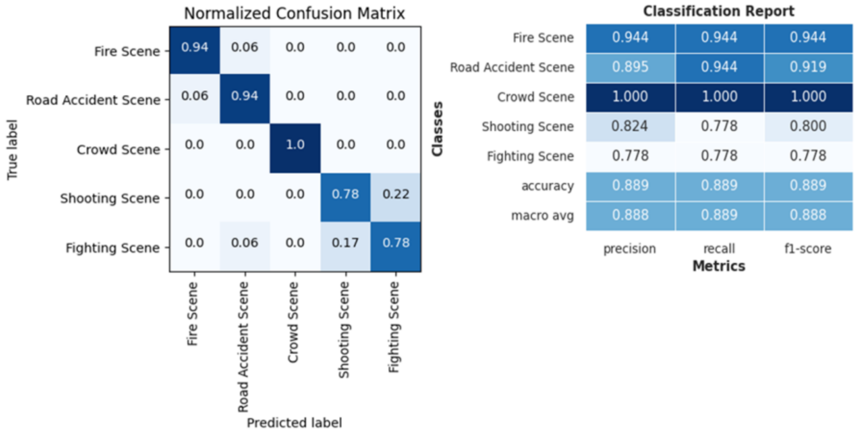


Fig.9.*Performance Analysis of IncepResNetV2 CNN-RNN Model*

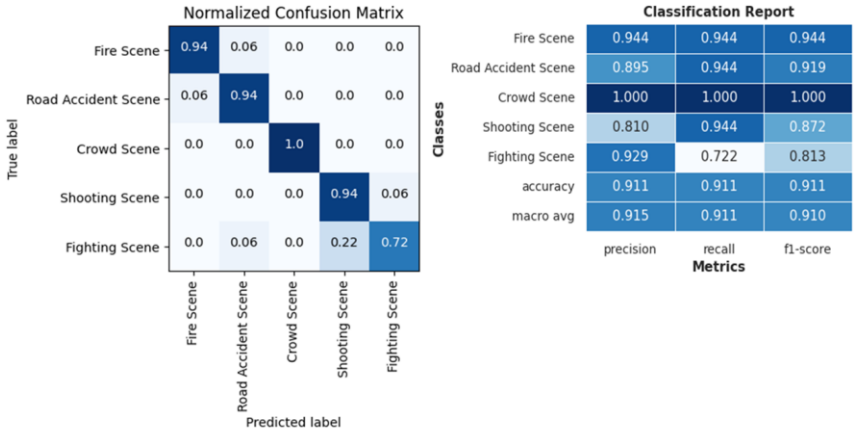


Fig.10.*Performance Analysis of InceptionV3 based CNN-RNN Model*

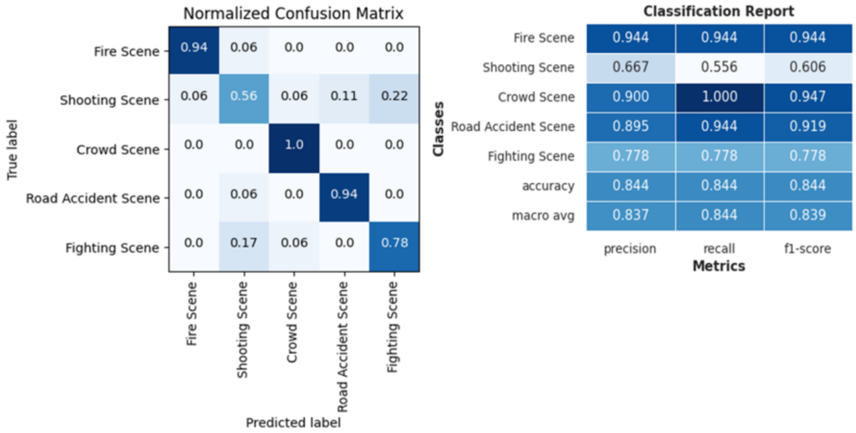
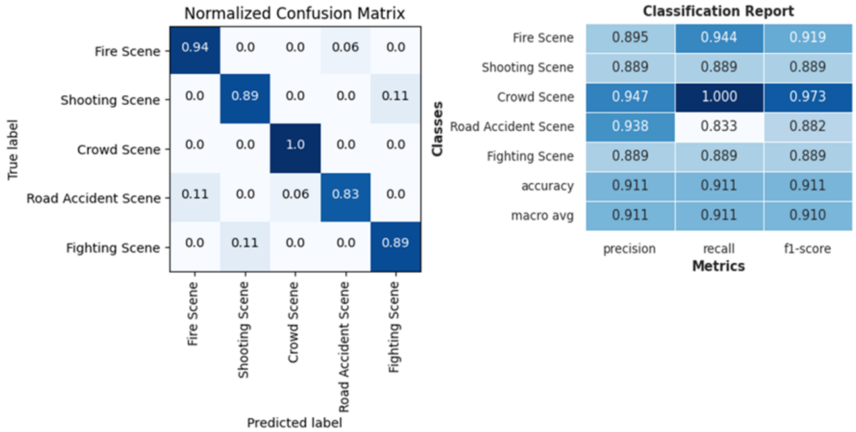


Fig.11.*Performance Analysis of VGG16 CNN-RNN Model*



Normalized Confusion Matrix

True label \ Predicted label	Fire Scene	Shooting Scene	Crowd Scene	Road Accident Scene	Fighting Scene
Fire Scene	0.94	0.0	0.0	0.06	0.0
Shooting Scene	0.0	0.89	0.0	0.0	0.11
Crowd Scene	0.0	0.0	1.0	0.0	0.0
Road Accident Scene	0.11	0.0	0.06	0.83	0.0
Fighting Scene	0.0	0.11	0.0	0.0	0.89

Classification Report

Classes	precision	recall	f1-score
Fire Scene	0.895	0.944	0.919
Shooting Scene	0.889	0.889	0.889
Crowd Scene	0.947	1.000	0.973
Road Accident Scene	0.938	0.833	0.882
Fighting Scene	0.889	0.889	0.889
accuracy	0.911	0.911	0.911
macro avg	0.911	0.911	0.910

Fig. 12. *Performance Analysis of MobileNetV2 CNN-RNN Model*

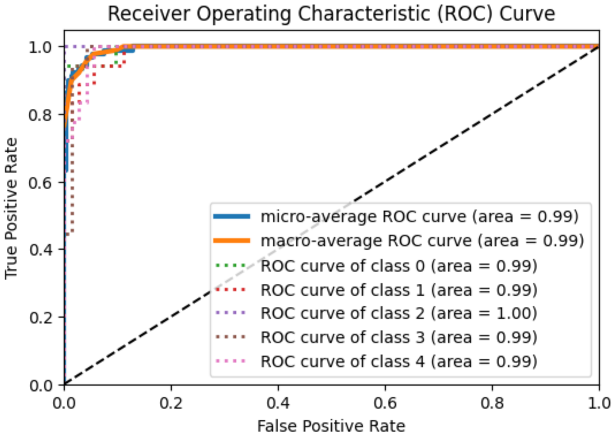
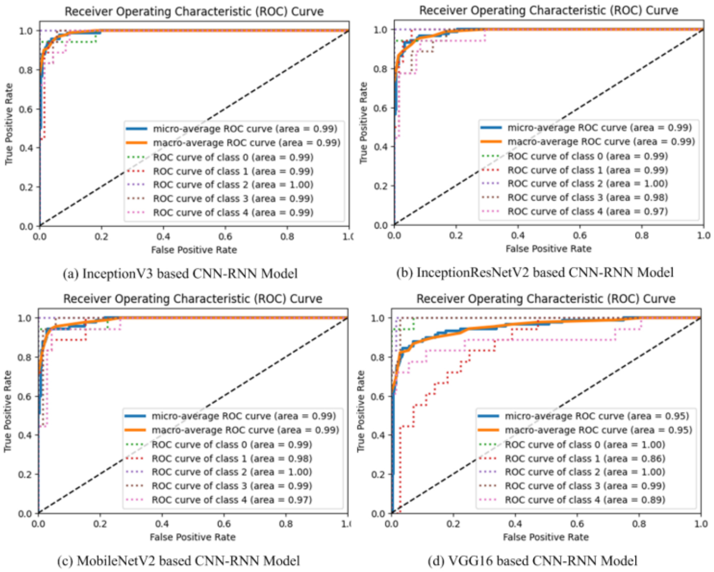


Fig. 13. *ROC curve of ResNet152V2 based CNN-RNN architecture*



(a) InceptionV3 based CNN-RNN Model

(b) InceptionResNetV2 based CNN-RNN Model

(c) MobileNetV2 based CNN-RNN Model

(d) VGG16 based CNN-RNN Model

Fig. 14. *ROC Plots*

4.1 Comparative Analysis of CNN-RNN Models

In this paper, the performance of five CNN-RNN models have been comprehensively evaluated through various performance metrics such as training graphs, confusion matrices, ROC curves and classification reports. These performance metrics give us valuable insights into how well each model performs on Anomaly-5 dataset in different aspects of classification. However, in order to make decision about which model performs well amongst the model, we need to compare the models through multiple key metrics. Therefore, in this final analysis, we emphasis on assessing test accuracies, precision, recall and F1scores of the models to identify which model tops in terms of accuracy, class wise performance and the balance between precision and recall. This comparative analysis is presented in Table 2.

TABLE. 2. Comparative Analysis

Models	Accuracy	Precision	Recall	F1Score
InceptionV3CNN-RNN	91.10	91.10	91.10	91.10
InceptionResNetV2CNN-RNN	88.90	88.90	88.90	88.90
ResNet152V2CNN-RNN	92.20	92.20	92.20	92.20
VGG16CNN-RNN	84.40	84.40	84.40	84.40
MobileNetV2CNN-RNN	91.10	91.10	91.10	91.10

ResNet152V2 based CNN-RNN model achieved highest test accuracy of 92.20%, indicating that this model has classified or predicted highest overall correct predictions on the test dataset. This model also outperformed the other models by achieving highest F1 score of 92.20%, suggesting that the model has also better balance between true positives and false positives. The InceptionV3 and MobiNetV2 based CNN-RNN models showed slightly lower performance but still performed well in terms of accuracy, precision, recall and F1-score. The VGG16 based CNN-RNN achieved lowest classification results as compared to other CNN-RNN models.

5. Conclusion

In this paper, we analyzed and evaluated the performance of various pre-trained CNN-RNN models for action recognition on the Anomaly-5 dataset. We observed several key findings through experimental results and their analysis. Firstly, we found that CNN-RNN architectures by combining CNNs with RNNs consistently outperformed than single-stream CNN architectures which indicate the significance of capturing both spatial and temporal information for accurate action recognition. Secondly, among the CNN-RNN architectures considered, ResNet152V2 based CNN-RNN model exhibits better performance due to its ability to capture more complex spatial features. This architecture offers higher testing recognition accuracy than others; however it often comes at the cost of increased computational complexity, highlighting the trade-off between performance and efficiency. In future, we also aiming to extend Anomaly-5 dataset by introducing new anomaly classes and in each class including more action videos.

References

[1] M. Zahrawi and K. Shaalan, “Improving video surveillance systems in banks using deep learning techniques,” *Sci. Rep.*, vol. 13, no. 1, Art. no. 1, May 2023.

[2] M. A. Khan *et al.*, “Human action recognition using fusion of multiview and deep features: an application to video surveillance,” *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 14885–14911, Feb. 2024.

[3] N. D. Kathamuthu *et al.*, “A deep transfer learning-based convolution neural network model for COVID-19 detection using computed tomography scan images for medical applications,” *Adv. Eng. Softw.*, vol. 175, p. 103317, Jan. 2023.

[4] I. Hussain, S. Khan, and M. B. Nazir, “Empowering Healthcare: AI, ML, and Deep Learning Innovations for Brain and Heart Health,” *Int. J. Adv. Eng. Technol. Innov.*, vol. 1, no. 4, Art. no. 4, May 2024.

[5] J. D. Choi and M. Y. Kim, “A sensor fusion system with thermal infrared camera and LiDAR for autonomous vehicles and deep learning based object detection,” *ICT Express*, vol. 9, no. 2, pp. 222–227, Apr. 2023.

[6] W. Sun, L. Pan, J. Xu, W. Wan, and Y. Wang, “Automatic Driving Lane Change Safety Prediction Model Based on LSTM,” in *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, Shanghai, China: IEEE, Mar. 2024, pp. 1138–1142.

[7] M. M. Afsar *et al.*, “Body-Worn Sensors for Recognizing Physical Sports Activities in Exergaming via Deep Learning Model,” *IEEE Access*, vol. 11, pp. 12460–12473, 2023.

[8] W. Alsabhan, “Human–Computer Interaction with a Real-Time Speech Emotion Recognition with Ensembling Techniques 1D Convolution Neural Network and Attention,” *Sensors*, vol. 23, no. 3, p. 1386, Jan. 2023.

[9] K. You, C. Zhou, and L. Ding, “Deep learning technology for construction machinery and robotics,” *Autom. Constr.*, vol. 150, p. 104852, Jun. 2023.

[10] D. Moutinho, L. F. Rocha, C. M. Costa, L. F. Teixeira, and G. Veiga, “Deep learning-based human action recognition to leverage context awareness in collaborative assembly,” *Robot. Comput.-Integr. Manuf.*, vol. 80, p. 102449, Apr. 2023.

[11] D. Dukić and A. Sovic Krzic, “Real-Time Facial Expression Recognition Using Deep Learning with Application in the Active Classroom Environment,” *Electronics*, vol. 11, no. 8, Art. no. 8, Jan. 2022.

[12] J. Arunnehru, G. Chamundeeswari, and S. P. Bharathi, “Human Action Recognition using 3D Convolutional

Neural Networks with 3D Motion Cuboids in Surveillance Videos," *Procedia Comput. Sci.*, vol. 133, pp. 471–477, Jan. 2018.

[13] V. A. Athavale, S. C. Gupta, D. Kumar, and Savita, "Human Action Recognition Using CNN-SVM Model," *Adv. Sci. Technol.*, vol. 105, pp. 282–290, 2021.

[14] P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug. 2016, pp. 30–36.

[15] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, pp. 533–536, Oct. 1986.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, in NIPS'14. Cambridge, MA, USA: MIT Press, Dec. 2014, pp. 568–576.

[19] L. Wang *et al.*, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 20–36.

[20] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal Relational Reasoning in Videos," Jul. 24, 2018, *arXiv*: arXiv:1711.08496.

[21] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6202–6211.

[22] J. Donahue *et al.*, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.

[23] C. Dai, X. Liu, and J. Lai, "Human action recognition using two-stream attention based LSTM networks," *Appl. Soft Comput.*, vol. 86, p. 105820, Jan. 2020.

[24] L. Chen, X. Zeng, and D. Li, "A two-stream attention mechanisms network," in *2022 International Conference on Artificial Intelligence, Information Processing and Cloud Computing (AIIPCC)*, Aug. 2022, pp. 226–230.

[25] W. Ullah, T. Hussain, F. U. M. Ullah, M. Y. Lee, and S. W. Baik, "TransCNN: Hybrid CNN and transformer mechanism for surveillance anomaly detection," *Eng. Appl. Artif. Intell.*, vol. 123, p. 106173, Aug. 2023.

[26] R. Pramanik, R. Sikdar, and R. Sarkar, "Transformer-based deep reverse attention network for multi-sensory human activity recognition," *Eng. Appl. Artif. Intell.*, vol. 122, p. 106150, Jun. 2023.

[27] M. Yang *et al.*, "Transformer-based deep learning model and video dataset for unsafe action identification in construction projects," *Autom. Constr.*, vol. 146, p. 104703, Feb. 2023.

[28] M. Kumar and M. Biswas, "Abnormal human activity detection by convolutional recurrent neural network using fuzzy logic," *Multimed. Tools Appl.*, vol. 83, no. 22, pp. 61843–61859, Jul. 2024.