Soft Computing

Analysis and Comparison of Various Deep Learning Models to Implement Suspicious Activity Recognition in CCTV Surveillance

--Manuscript Draft--

Manuscript Number:	SOCO-D-23-04068
Full Title:	Analysis and Comparison of Various Deep Learning Models to Implement Suspicious Activity Recognition in CCTV Surveillance
Article Type:	S.I. : Soft computing-based IDSS and its use in solving real-world issues
Keywords:	Deep learning; machine learning; DenseNet121; Activity recognition; Convolutional neural network; Video processing
Corresponding Author:	Dhruv Saluja, B.Tech Bharati Vidyapeeth's College of Engineering New Delhi New Delhi, Delhi INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Bharati Vidyapeeth's College of Engineering New Delhi
First Author:	Dhruv Saluja, B.Tech
First Author Secondary Information:	
Order of Authors:	Dhruv Saluja, B.Tech
	Harsh Kukreja, B.Tech
	Akash Saini, B.Tech
	Devanshi Tegwal, B.Tech
	Preeti Nagrath
Funding Information:	
Abstract:	The project aims to develop a Suspicious Activity Recognition (SAR) system for closed-circuit television (CCTV) surveillance using deep learning (DL) algorithms. Automated systems for detecting and classifying suspicious activities are crucial as technology's role in safety and security expands. This project addresses these challenges by creating a robust SAR system using machine learning techniques. It analyzes and compares evaluation metrics such as Precision, Recall, F1 Score, and Accuracy using various deep learning methods (convolutional neural network (CNN), Long short-term memory (LSTM) - Visual Geometry Group 16 (VGG16), LSTM - ResNet50, LSTM - EfficientNetB0, LSTM InceptionNetV3, LSTM - DenseNet121, and Long-term Recurrent Convolutional Network (LRCN)). The proposed system improves threat identification, vandalism deterrence, fight prevention, and video surveillance. It aids emergency response by accurately classifying suspicious activities from CCTV footage, reducing reliance on human security personnel and addressing limitations in manual monitoring. The objectives of the project include analyzing existing works, extracting features from CCTV videos, training robust deep learning models, evaluating algorithms, and improving accuracy. The conclusion highlights the superior performance of the LSTM-DenseNet121 algorithm, achieving an overall accuracy of 91.17% in detecting suspicious activities. This enhances security monitoring capabilities and reduces response time. Limitations of the system include subjectivity, contextual understanding, occlusion, false alarms, and privacy concerns. Future improvements involve real-time object tracking, collaboration with law enforcement agencies, and performance optimization. Ongoing research is necessary to overcome limitations and enhance the effectiveness of CCTV surveillance.

Analysis and Comparison of Various Deep Learning Models to Implement Suspicious Activity Recognition in CCTV Surveillance

Dhruv Saluja[1*†], Harsh Kukreja[1†], Akash Saini[1†], Devanshi Tegwal[1†], Preeti Nagrath[1†]

[1*]Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, A - 4, Paschim Vihar, New Delhi, 110063, Delhi, India.

*Corresponding author(s). E-mail(s): dhruvsaluja4@gmail.com;
Contributing authors: harsh25kukreja@gmail.com; akashmm861@gmail.com;
devyanshitegwal8130@gmail.com; preeti.nagrath@bharatividyapeeth.edu;
†These authors contributed equally to this work.

Abstract

The project aims to develop a Suspicious Activity Recognition (SAR) system for closed-circuit television (CCTV) surveillance using deep learning (DL) algorithms. Automated systems for detecting and classifying suspicious activities are crucial as technology's role in safety and security expands. This project addresses these challenges by creating a robust SAR system using machine learning techniques. It analyzes and compares evaluation metrics such as Precision, Recall, F1 Score, and Accuracy using various deep learning methods (convolutional neural network (CNN), Long short-term memory (LSTM) - Visual Geometry Group 16 (VGG16), LSTM - ResNet50, LSTM - EfficientNetB0, LSTM - InceptionNetV3, LSTM - DenseNet121, and Long-term Recurrent Convolutional Network (LRCN)). The proposed system improves threat identification, vandalism deterrence, fight prevention, and video surveillance. It aids emergency response by accurately classifying suspicious activities from CCTV footage, reducing reliance on human security personnel and addressing limitations in manual monitoring. The objectives of the project include analyzing existing works, extracting features from CCTV videos, training robust deep learning models, evaluating algorithms, and improving accuracy. The conclusion highlights the superior performance of the LSTM-DenseNet121 algorithm, achieving an overall accuracy of 91.17% in detecting suspicious activities. This enhances security monitoring capabilities and reduces response time. Limitations of the system include subjectivity, contextual understanding, occlusion, false alarms, and privacy concerns. Future improvements involve real-time object tracking, collaboration with law enforcement agencies, and performance optimization. Ongoing research is necessary to overcome limitations and enhance the effectiveness of CCTV surveillance.

Keywords: Deep learning, Machine learning, DenseNet121, Activity recognition, Convolutional neural network, Video processing

1 Introduction

As the world becomes increasingly digitized, the reliance on technology for safety and security has grown. In particular, The utilization of Closed Circuit Television (CCTV) cameras to observe public spaces in order to detect suspicious behavior has become prevalent. However, (Popoola et al. 2012) there is a pressing need for automated systems that can effectively detect suspicious activity in real-time. The objective of this project is to develop a suspicious activity recognition (SAR) system using machine learning for CCTV surveillance. Manual monitoring is time-consuming and prone to errors, making it crucial to develop a robust SAR system using machine learning algorithms.

1.1 Motivation

- **Prevention of Theft:** Video surveillance helps in deterring and preventing theft cases by providing continuous monitoring of public areas. Suspicious activities can be detected in real-time, allowing for immediate response and intervention to prevent theft incidents.
- **Identification of Potential Threats:** Video surveillance enables the identification of potential threats posed by abandoned objects. By monitoring public spaces, video surveillance systems can alert security personnel to any suspicious objects that could potentially be used for explosive attacks. Prompt action can be taken to neutralize the threat and ensure public safety.
- **Deterrence of Vandalism and Personal Attacks:** The presence of video surveillance cameras acts as a deterrent against acts of vandalism and personal attacks. Knowing that their actions are being recorded increases the likelihood of individuals refraining from engaging in such behaviors. In case of any incidents, video footage can be used as evidence for investigations and legal proceedings.
- **Prevention of Fighting Incidents:** Video surveillance systems play a crucial role in monitoring public areas prone to fighting incidents, such as entertainment venues, transportation hubs, and crowded spaces. By identifying signs of aggression and monitoring crowd behavior, security personnel can intervene to prevent fights and maintain a peaceful environment.
- **Cost-effective Security Solutions:** Implementing video surveillance systems provides a cost-effective solution for maintaining security in public areas. It reduces the reliance on physical security personnel by automating the detection and monitoring process. This leads to more efficient resource allocation and reduces the overall security costs for institutions and organizations.
- **Emergency Response and Disaster Management:** (Kyrkou et al. 2020) Video surveillance systems play a crucial role in emergency response and disaster management. They provide real-time situational awareness, assist in coordinating response efforts, and aid in the rapid deployment of resources during critical situations. Video surveillance projects contribute to effective emergency preparedness, response, and recovery, saving lives and minimizing damage in times of crisis.

1.2 Problem Statement

The problem addressed in this project is the challenge of accurately identifying and classifying suspicious activities captured by CCTV cameras in public areas. Suspicious activity can be diverse and difficult to define, often occurring in various forms. Manual monitoring is not only labor-intensive but also susceptible to human biases and limitations. Therefore, the project aims to overcome these challenges by developing an automated SAR system that can efficiently detect and classify suspicious activities, reducing the reliance on physical security personnel.

1.3 Aim and Objective

- Develop a machine learning-based SAR system for monitoring public areas, detecting suspicious activities, and alleviating the workload of security personnel.
- To summarize and analyze the results of previously published works on the technologies for Suspicious Activity Detection in CCTV Video Surveillance
- To extract frames and identify the features from the CCTV video using a pre-trained CNN.
- Train robust DL model using labeled data to accurately detect suspicious activity from CCTV footage.

- To estimate the evaluation criteria and efficiency of various algorithms and compare their results with the state of art methodologies.
- To improve the accuracy of models and add more classes.

1.4 Scope of Project

The scope of this project entails designing and implementing a SAR system utilizing deep learning algorithms. It involves collecting a diverse dataset of labeled suspicious activities and training the model to classify these activities accurately. When the system detects suspicious activities, it reduces response time to potential threats. However, the project does not encompass the physical installation of CCTV cameras or the management of the overall surveillance infrastructure. By focusing on these objectives and scope, the project aims to enhance security monitoring capabilities, provide a more efficient and reliable surveillance system, and contribute to public safety.

1.5 Organization

This research paper comprises several sections to investigate the problem at hand. The paper begins with an abstract providing a concise overview of the study. Following the abstract is the introduction that sets the context for the research as shown in Section 1, related works given in Section 2 provide an overview of the existing knowledge on the topic. Section 3 explains the approach and the research methodology used in this study. Results and analysis present the findings of the study which is in Section 4, and Section 5 concludes the paper and summarizes the key outcomes.

2 Related Works

The challenges encountered in the detection of suspicious actions are not novel. One approach, known as Tube Convolutional Neural Network (T-CNN), has been developed to recognize and locate actions using three-dimensional (3D) convolution features (Hou et al. 2017). However, it remains unclear how well this T-CNN architecture can be generalized to different domains. Greg Mori (De Geest et al. 2018) employed an LSTM approach capable of modeling short-term and long-term patterns. Unfortunately, applying LSTM alone for action detection did not yield significant improvements compared to traditional methods. In the case of Spatio-Temporal Activity Detection (Gkountakos et al. 2021) which utilizes a 3D Convolutional deep learning architecture to handle spatio-temporal boundaries as multi-labeled classifications, accurately detecting and recognizing the boundaries of each activity posed challenges due to the varying length of co-occurring activities.

Yeung et al. (2016) introduced an end-to-end approach for action detection that directly reasons about the temporal bounds of actions, using the THUMOS'14 and ActivityNet datasets. This approach only observes a small fraction (2% or less) of the video frames. The extracted features from CNN are then fed into a Discrimination Deep Belief Network (DDBN) (Scariaa et al. 2017) for detecting suspicious activities. However, the accuracy of this methodology depends on the scene's complexity, and it demands substantial computational resources compared to other approaches. Ehud Rivlin (Adam et al. 2008) presented an algorithm based on statistical monitoring of low-level observations at multiple spatial locations, which could have been highly beneficial. However, it is limited in its ability to detect events characterized by an unusual sequence of short-term normal actions.

Boudihir et al. (2012) proposed the system architecture of an Intelligent video surveillance system (IVSS), based on the Support Vector Machine (SVM) framework, but it did not support sudden frame deformations. Tripathi et al. (2018), utilizing the PETS 2006 dataset, proposed a series of algorithms for activity detection but failed to provide a discussion on the accuracy, false positive rate, and false negative rate, thereby lacking quantitative analysis and evaluation measures, making it challenging to assess the performance of the proposed algorithms.

The Sequential Convolutional Neural Network (SCNN) (Yang et al. 2017) model, which directly feeds two-dimensional convolved feature maps into the recurrent model, demonstrated better performance than UCF-101 and Human Motion Database 51 (HMDB-51) (Wang et al. 2017). However, it suffered from conventional issues such as dataset bias, computational complexity, and

overfitting (Yao et al. 2018). Reviewed CNN-based (Chéron et al. 2015) action recognition and highlighted the algorithm's limitations in terms of robustness, suggesting the need for a more efficient and robust CNN architecture. The combination of CNN and deep bidirectional LSTM networks (Ullah et al. 2018) explored action detection performance on a limited dataset. However, this approach demanded significant computational resources and lacked interpretability, as deep learning models are often considered black boxes. VideoCapsuleNet (Zhao L et al. 2021) presented a novel approach using a simplified network based on the capsule network for action detection. It simultaneously performed pixel-wise action segmentation and action classification. However, this approach faced challenges in accurately localizing the action.

3 Research Methodology

The methodology employed in this research is divided into five main phases: the first phase involves the capturing of the videos; the second phase involves the extraction of frames from the captured videos by pre-processing them; the third phase involves crafting various deep learning models that detect and recognize actions; in the fourth phase, we train the models using the acquired datasets and validate their performance on the designated test split; the fifth phase involves class prediction utilizing the deep learning models developed for action detection and recognition purposes.

The proposed method employed footage captured by CCTV cameras to monitor the activities of individuals and triggered alerts in the event of any suspicious incidents. Fig. 1 shows the workflow of the entire process. This consisted of different phases such as video capture, video pre-processing, model creation, train test split data and class prediction. The videos were classified into three classes as follows: people fighting (suspicious class), walking and running (conventional/normal class).

3.1 Video capturing

The primary phase of the video surveillance system involved the monitoring and analysis of the recorded closed-circuit television (CCTV) footage.

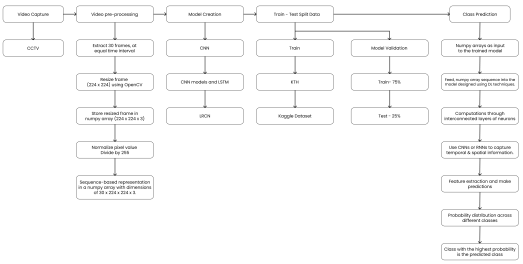

Fig. 1: Methodology

The collected CCTV footage served as a crucial dataset for our research endeavor, providing the necessary foundation for training and testing the performance of our developed model. The acquisition of video data was accomplished through the utilization of CCTV cameras, as illustrated in Fig. 1.

3.2 Video Pre-processing

In video preprocessing for machine learning tasks, an essential step involves the extraction of frames from the captured videos. The flow of video preprocessing is shown in Fig. 1. In this particular process, 30 frames were extracted, ensuring an equal time interval between each frame. These frames were used as representative snapshots of the video content. To facilitate further analysis, the extracted frames were then resized to a standardized dimension of 224 x 224 pixels using Open Source Computer Vision Library (OpenCV), a popular computer vision library in Python.

The resized frames were stored in a numpy array, which had a dimension of (224 x 224 x 3), representing the image width, height, and RGB channels, respectively. This format enabled efficient storage and manipulation of the frames within the machine learning pipeline. Additionally, to ensure consistent data representation, each value in the frame was normalized by dividing it by 255, which scaled the pixel values to a range between 0 and 1.

To capture the temporal dynamics of the video, all the 30 normalized frames from each video were stored as a sequence in a numpy array. This resulted in a final array with dimensions of 30 x 224 x 224 x 3, encapsulating the temporal evolution of the video frames. This sequence-based representation facilitated the training and analysis

of deep learning models, enabling them to leverage the sequential information present in the video data.

3.3 Model Creation

In model creation, we build models that extract patterns from complex data. These models are trained on large datasets to improve their performance. Here we explored different deep learning algorithms and architectures which are as follows:

3.3.1 CNN

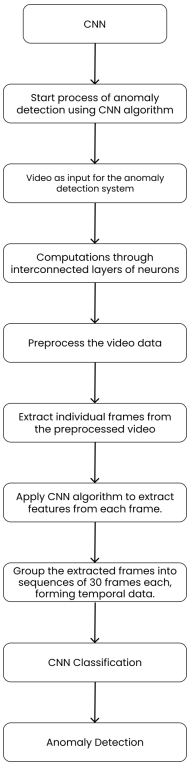

Fig. 2: Anomaly Detection using CNN

In the initial approach, the Convolutional Neural Network algorithm was employed to detect anomalous behavior. Fig. 2 shows the process of detecting an anomaly using CNN. To ensure the effective classification of these anomalies, the capturing, and analysis of the temporal data present in the video were considered essential. CNNs had emerged as a popular technique for extracting significant features from individual video frames. To successfully classify the input, it was crucial for the CNN to accurately identify and extract the required features from each frame. Sequences of 30 frames were extracted from the video and directly fed into the CNN model for classification. This approach eliminated the need for recurrent connections and focused solely on the capabilities of the CNN algorithm.

By utilizing convolutional layers, which apply a set of learnable filters to input images, local patterns and spatial relationships present in the data can be captured by CNNs. Through this process of hierarchical feature extraction, complex patterns and objects can be detected at different levels of abstraction.

3.3.2 LSTM with CNN architectures (VGG16, ResNet50, EffecientNetB0, InceptionV3, DenseNet121)

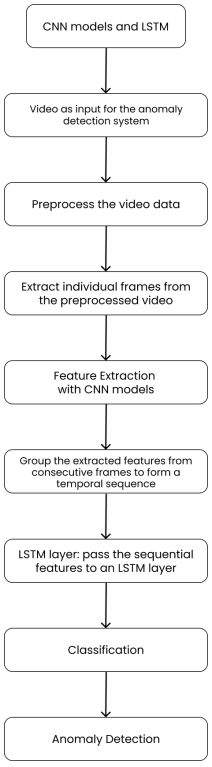

Fig. 3: Anomaly Detection using CNN models and LSTM

The subsequent approach now utilized different CNN architectures (VGG16, ResNet50, EfficientNetB0, InceptionV3, DenseNet) for the detection of anomalous behavior as compared to the previous approach above which only used a base CNN model. Fig. 3 shows the process of detecting an anomaly using different CNN models and Long short-term memory (LSTM). In the approach, features were extracted from 30 frames of the video and then passed as a sequence to an LSTM layer, enabling the prediction of the video's class. The following are the various CNN pre-trained models and all of them are typically trained using stochastic gradient descent (SGD) optimization.

- **VGG16 (Visual Geometry Group 16):** VGG-16, a widely recognized convolutional neural network (CNN) architecture, is characterized by its deep structure consisting of 16 layers, hence the name. The network primarily consists of convolutional layers with 3x3 filters and a stride of 1, followed by max-pooling layers with a 2x2 window and a stride of 2 for downsampling. This architecture allows VGG-16 to learn hierarchical representations of input images, capturing features at different levels of abstraction. The model also incorporates employing techniques like dropout for regularization.
- **ResNet50 (Residual Network 50):** ResNet-50, a prominent CNN architecture, introduces the concept of residual learning to address the challenges of training deeper networks. It consists of 50 layers, including residual blocks that allow for the direct flow of information from one layer to another. Each residual block contains multiple convolutional layers, followed by skip connections that bypass the intermediate layers. This enables the network to learn residual mappings, focusing on the differences between the input and the desired output. ResNet-50 is typically trained with SGD along with cross-entropy loss and regularization techniques such as dropout or weight decay.
- **EfficientNetB0:** EfficientNetB0 is a state-of-the-art CNN architecture that aims to achieve superior performance while being computationally efficient. It employs a novel approach called compound scaling, which systematically scales the depth, width, and resolution of the network. By striking a balance between these dimensions, achieves optimal resource allocation and improves both accuracy and efficiency. It incorporates a combination of depthwise separable convolutions and squeeze-and-excitation blocks, which reduce the number of parameters and computational complexity without compromising the model's representational power. It is also trained using SGD with adaptive learning rate schedules and weight decay regularization.
- **InceptionV3:** InceptionV3 is a notable CNN architecture that addresses the challenge of balancing model depth and computational efficiency. It introduces the concept of "Inception" modules, which consist of multiple parallel convolutional operations with different filter sizes and pooling operations. This allows the network to capture information at different scales and resolutions, enabling richer feature representations. InceptionV3 also incorporates additional techniques such as batch normalization and factorized convolutions to further enhance model efficiency. The architecture is typically trained using SGD with momentum and weight decay regularization.
- **DenseNet121:** DenseNet121 is a CNN architecture that introduces a dense connectivity pattern among its layers, aiming to enhance feature propagation and information flow. Unlike traditional CNNs, where layers are connected in a sequential manner, DenseNet121 establishes direct connections between all layers in a block, resulting in dense interconnections. This design enables each layer to receive feature maps from all preceding layers, allowing for efficient information sharing and promoting feature reuse. By leveraging this dense connectivity, DenseNet121 encourages gradient flow, reduces the vanishing gradient problem, and enables better feature representation. The training techniques used in this architecture are the same as ResNet50.

The proposed system utilized a CNN model architecture for the purpose of detecting suspicious activity in video surveillance. Various CNN models with LSTM were employed, as CNN models had demonstrated impressive performance in large-scale image recognition tasks. To adapt the CNN model for video analysis, certain modifications were made to the architecture. The last four layers of the CNN network were removed and

incorporated into a TimeDistributed layer. This allowed the processing of each frame of the video separately while retaining the convolutional neural network architecture. Following the TimeDistributed layer, an LSTM layer and two dense layers were added. The incorporation of LSTM (Long Short-Term Memory) was particularly effective for handling time series data, as it enabled the capture of dependencies and the making of predictions based on temporal patterns. LSTMs were specifically designed to address the vanishing gradient problem commonly encountered in training traditional recurrent neural networks.

In summary, the approach leveraged a CNN model architecture to detect suspicious activity in videos. By integrating this model with a TimeDistributed layer, LSTM, and dense layers, the system was able to effectively process and classify time-dependent information, thereby enabling accurate detection of anomalous events in video surveillance.

3.3.3 LRCN

In the concluding approach, the model incorporates the utilization of LRCN (Long-term Recurrent Convolutional Network) for detecting anomalous behavior, relying on CNNs to extract pivotal features from individual frames and analyzing the temporal data within the video. Fig. 4 shows the process of detecting an anomaly using LRCN. Our proposed system for suspicious activity recognition from video surveillance is being utilized by a deep-learning network called LRCN. The main concept behind LRCN involves the utilization of a combination of CNNs to extract visual features from video frames and LSTMs to convert a sequence of image embeddings into various outputs such as class labels, sentences, probabilities, or any desired format. Consequently, the raw visual input is processed using a CNN, and the resulting outputs are fed into a stack of recurrent sequence models. LSTM networks are especially suitable for the classification, processing, and prediction tasks based on time series data. This is because there may exist time gaps of unknown duration between significant events in a time series. LSTMs were developed to address the vanishing gradient problem, which can arise when training traditional RNNs.

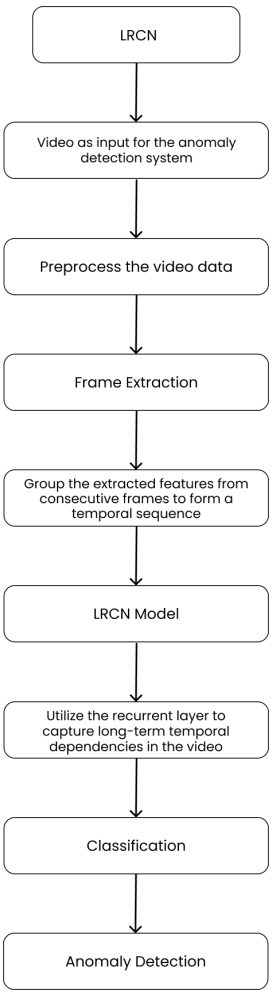

Fig. 4: Anomaly Detection using LRCN

3.4 Train Test Split Data

For the purpose of training we are using the following datasets and they are available as KTH dataset (Schuldt et al. 2004) for detection of Running and Walking, https://www.csc.kth.se/cvap/actions/ and the Kaggle dataset for fight detection, https://www.kaggle.com/naveenk903/movies-fight-detection-dataset.

The KTH dataset is a widely recognized benchmark for action recognition, with six distinct actions. Each action class has 100 sequences, approximately 600 frames per sequence, and a frame rate of 25 frames per second. Due to hardware limitations, the model trained on a reduced subset of 45 sequences per class, focusing on

running and walking. This approach optimizes resource usage and improves accuracy in classifying normal behaviors. The dataset's rich temporal information, diverse action classes, and controlled environment provide a solid foundation for training action recognition models. Future investigation could explore the model's performance on unseen data and its potential for generalization beyond the dataset's limitations. The Kaggle Dataset offers a valuable resource for training models to detect suspicious behavior, specifically focusing on instances of fighting. This dataset comprises over 100 videos sourced from various movies and YouTube videos, providing a diverse and extensive collection of examples.

By leveraging these datasets, deep learning algorithms can be trained to accurately identify and classify instances of suspicious behavior, enabling proactive measures for maintaining security and public safety. The inclusion of videos from different sources contributes to the dataset's richness and ensures a broader representation of real-world scenarios. The combined dataset provides sufficient training data to develop effective models capable of recognizing and distinguishing instances of fighting accurately.

The train-test split is considered a crucial step in machine learning, involving the division of a dataset into two subsets: the training set and the test set, as illustrated in Fig. 1. The training set is utilized for training the model, enabling it to learn patterns and relationships within the data. The test set is employed to assess the model's performance on unseen data. The split is typically performed randomly, with a larger portion being allocated to the training set (around 70-80%) and the remaining portion reserved for the test set (approximately 20-30%). By acting as a benchmark, the test set facilitates the assessment of the model's ability to generalize to new data, aiding in informed decisions regarding model selection and performance enhancement. It is important to note that the test set should not be used for training or parameter tuning to avoid bias. Instead, a separate validation set can be utilized. In summary, the train-test split plays an essential role in evaluating model performance on unseen data and guiding the development of the model. We used 75% of the data for training and 25% of the data is used for testing.

3.5 Class Prediction

In the field of deep learning, the utilization of numpy arrays as input to models has been a common practice. Numpy arrays, with their efficient and versatile nature, have been used as a suitable data structure for representing video data. By feeding the numpy array as input to a trained model, the power of deep learning algorithms can be harnessed to make predictions on the class of the given video. The model has undergone extensive training on labeled video data to learn the underlying patterns and features that differentiate different classes or categories. Through this training process, the model has developed a deep understanding of the data and has been able to generalize its knowledge to new, unseen videos. When the numpy array is passed into the model, it undergoes a series of computations through layers of interconnected neurons. These computations involve matrix multiplications, non-linear activation functions, and parameter optimizations to transform the input data into meaningful representations. The model's architecture, such as CNNs, has played a crucial role in capturing temporal and spatial information within the video. As the numpy array has propagated through the model, the network's learned weights and biases have been applied to extract relevant features and make predictions. The final output of the model has been a probability distribution across different classes, indicating the likelihood of the video belonging to each category. By selecting the class with the highest probability, the predicted class of the given video could be determined. It's worth noting that the accuracy of the model's predictions has heavily relied on the quality and diversity of the training data, the model's architecture and hyperparameter configuration, as well as the availability of computational resources for training and inference. Overall, the process of feeding numpy arrays into a model and obtaining class predictions as depicted in Fig. 1 has showcased the power of deep learning in video analysis and classification tasks, enabling applications in various domains such as surveillance, activity detection, and more.

4 Results and analysis

4.1 Total accuracy vs Total validation accuracy

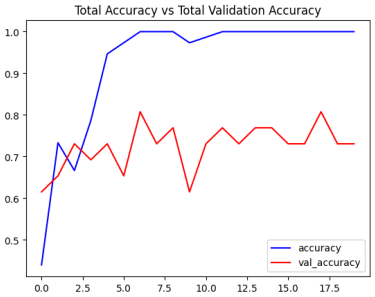

Fig. 5: Total Accuracy vs Total Validation Accuracy using CNN

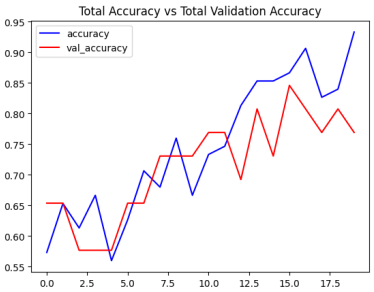

Fig. 6: Total Accuracy vs Total Validation Accuracy using LSTM - VGG16

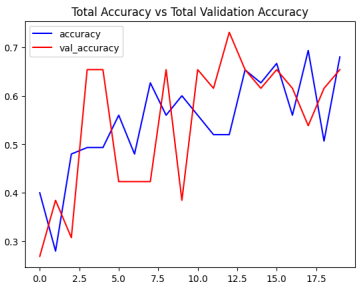

Fig. 7: Total Accuracy vs Total Validation Accuracy using LSTM - ResNet50

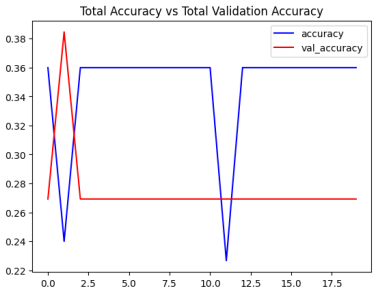

Fig. 8: Total Accuracy vs Total Validation Accuracy using LSTM - EfficientNetB0

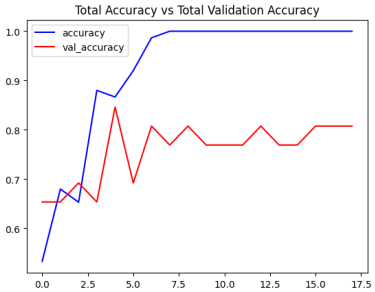

Fig. 9: Total Accuracy vs Total Validation Accuracy using LSTM - InceptionNetV3

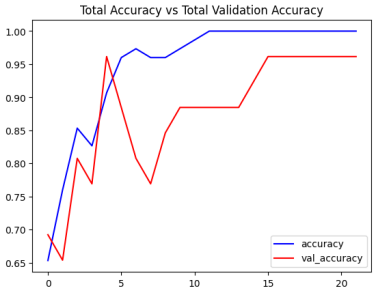

Fig. 10: Total Accuracy vs Total Validation Accuracy using LSTM - DenseNet121

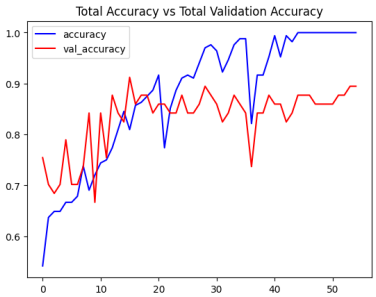

Fig. 11: Total Accuracy vs Total Validation Accuracy using LRCN

The "total accuracy vs total validation accuracy" graph compares the performance of a machine learning model on the training data (total accuracy) and the validation data (total validation accuracy). The total accuracy represents the accuracy of the model's predictions on the entire training dataset, while the total validation accuracy represents the accuracy on the separate validation dataset. The graphs of total accuracy vs total validation accuracy are shown in Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, Fig. 11 using CNN, LSTM - VGG16, LSTM - ResNet50, LSTM - EfficientNetB0, LSTM - InceptionNetV3, LSTM - DenseNet121 and LRCN respectively.

The graph shows how the two accuracies change over time or epochs during the training process. It provides insights into how well the model generalizes to unseen data.

Ideally, both accuracies should increase together initially, indicating that the model is learning from the training data and also performing well on the validation data. However, if the total accuracy continues to improve while the total validation accuracy plateaus or starts to decrease, it suggests that the model might be overfitting the training data and failing to generalize well to new data. Monitoring and analyzing this graph helps in assessing the model's performance and detecting issues like overfitting.

4.2 Total loss vs Total validation loss

The "total loss vs total validation loss" graph compares the loss of a machine learning model on the training data (total loss) and the validation data (total validation loss). The total loss represents the average loss of the model's predictions on the entire training dataset, while the total validation loss represents the average loss on the separate validation dataset. The graphs of total loss vs total validation loss are shown in Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18 using CNN, LSTM - VGG16, LSTM - ResNet50, LSTM - EfficientNetB0, LSTM - InceptionNetV3, LSTM - DenseNet121 and LRCN respectively. The graph shows how the two losses change over time or epochs during the training process. It provides insights into how well the model is fitting the training data and generalizing to new data.

Ideally, both losses should decrease together initially, indicating that the model is learning and minimizing errors. However, if the total loss continues to decrease while the total validation loss starts to increase, it suggests that the model is overfitting the training data and failing to generalize well to new data. Monitoring and analyzing this graph helps in assessing the model's performance and detecting overfitting issue.

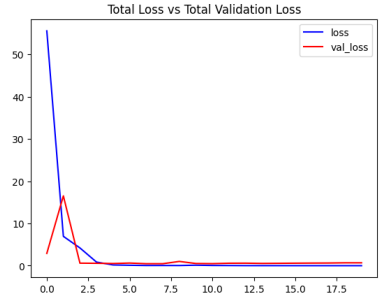

Fig. 12: Total Loss vs Total Validation Loss using CNN

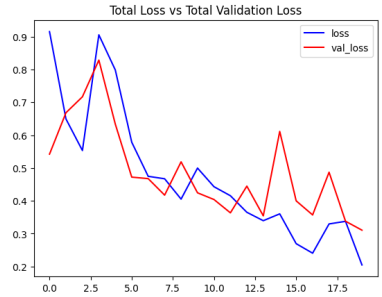

Fig. 13: Total Loss vs Total Validation Loss using LSTM - VGG16

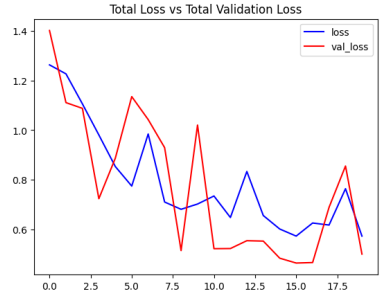

Fig. 14: Total Loss vs Total Validation Loss using LSTM - ResNet50

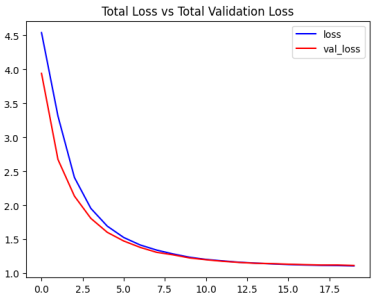

Fig. 15: Total Loss vs Total Validation Loss using LSTM - EfficientNetB0

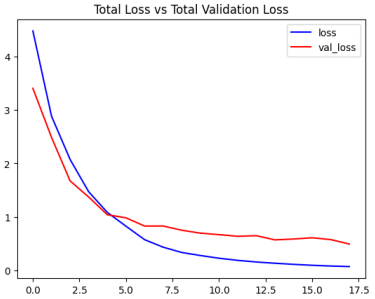

Fig. 16: Total Loss vs Total Validation Loss using LSTM - InceptionNetV3

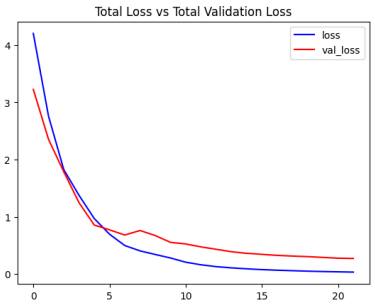

Fig. 17: Total Loss vs Total Validation Loss using LSTM - DenseNet121

4.3 Confusion matrix

A confusion matrix is a graphical representation of the performance of a classification model, specifically for a classifier with three class labels. It provides a detailed summary of the model's predictions and the actual class labels. The matrix

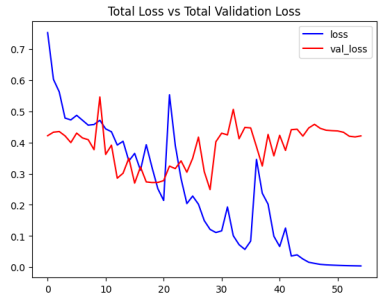

Fig. 18: Total Loss vs Total Validation Loss using LRCN

consists of a grid where the rows represent the actual class labels, and the columns represent the predicted class labels. Each cell in the matrix shows the number of instances that were classified into a particular combination of actual and predicted labels. The confusion matrix allows for a comprehensive analysis of the model's performance. It provides information about true positives, true negatives, false positives, and false negatives, enabling the calculation of various evaluation metrics such as accuracy, precision, recall, and F1 score.

Analyzing the confusion matrix helps identify patterns of misclassifications and understand the strengths and weaknesses of the model for each class label. It aids in evaluating the model's overall performance and provides valuable insights for refining the classification model. Label 0 - Walking Label 1 - Fighting Label 2 - Running

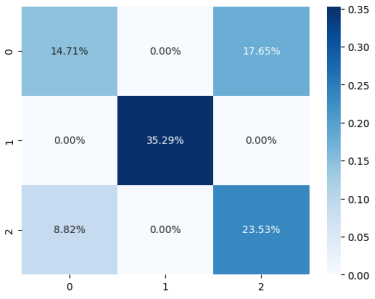

Fig. 19: Confusion matrix using CNN

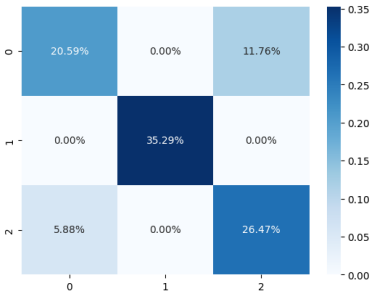

Fig. 20: Confusion matrix using LSTM - VGG16

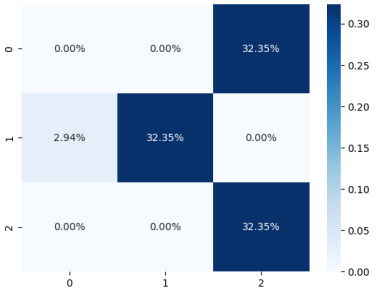

Fig. 21: Confusion matrix using LSTM - VGG16

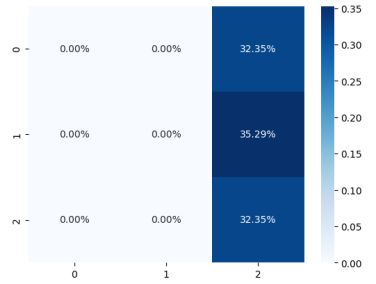

Fig. 22: Confusion matrix using LSTM - EfficientB0

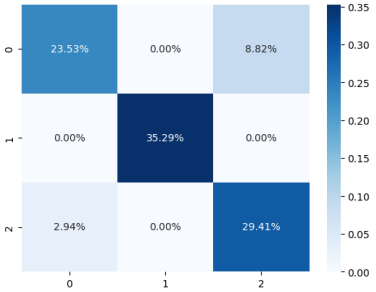

Fig. 23: Confusion matrix using LSTM - InceptionNetV3

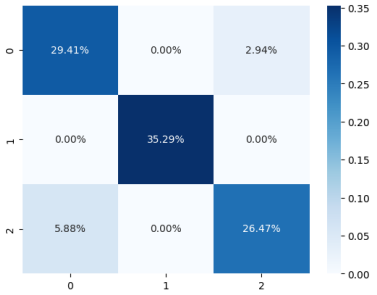

Fig. 24: Confusion matrix using LSTM - DenseNet121

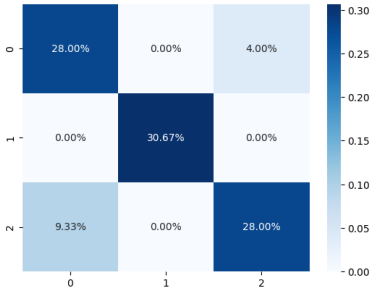

Fig. 25: Confusion matrix using LRCN

The performance of the model in classifying suspicious activity (specifically fighting) compared to conventional activities (such as running and walking) demonstrates a commendable level of accuracy. However, it is worth noting that the model's classification between running and walking shows room for improvement, as it is evident by the presence of numerous false positives and false negatives in the confusion matrix. The confusion matrix created using CNN is shown in Fig. 19. LSTM-VGG16 has demonstrated superior performance compared to the Convolutional Neural Network (CNN) approach. The model's accuracy is notably enhanced, indicated by the significant increase in true positives within the confusion matrix. This model is able to classify between the class labels better than the CNN model. The confusion matrix created using LSTM - VGG16 is shown in Fig. 20. Contrary to our expectations, this model does not exhibit any noticeable improvement in comparison to previous models. In fact, the confusion metrics highlight the model's inability to accurately classify instances into the

specific label of "walking." Moreover, the accuracy of the model diminishes when compared to its predecessors, as evidenced by the escalating number of false positives and negatives. The confusion matrix created using LSTM - ResNet50 is shown in Fig. 21. Among all the classification models considered, this particular model exhibits the poorest performance. It demonstrates a significant drawback by indiscriminately assigning all data points to a single class label, namely "running." The model's inability to differentiate and classify instances into multiple labels highlights a critical area for improvement. Addressing this limitation is paramount to ensure reliable and accurate classification results. The confusion matrix created using LSTM - EfficientB0 is shown in Fig. 22. The LSTM-InceptionNet model demonstrates a remarkable and substantial improvement in performance compared to all previous models. Notably, there is a significant reduction in the occurrence of false positives and false negatives, thereby enhancing the model's accuracy and reliability. This noteworthy advancement positions the model as a pivotal component for our final approach. The model exhibits exceptional proficiency in accurately classifying the data points, further reinforcing its effectiveness and value in our research study. The confusion matrix created using LSTM - InceptionNetV3 is shown in Fig. 23. The LSTM-DenseNet121 model emerges as the frontrunner in terms of performance when compared to its counterparts. It distinguishes itself by exhibiting the most favorable outcomes, characterized by the minimal occurrence of false positives and false negatives. As a result, it establishes itself as the optimal model to adopt for our final approach. Moreover, the model showcases a remarkable boost in accuracy compared to the alternative models under consideration. The confusion matrix created using LSTM - DenseNet121 is shown in Fig. 24.

The LRCN exhibits noteworthy performance, showcasing its competence in accurately classifying diverse class labels. This model demonstrates exceptional proficiency by achieving minimal instances of false positives and negatives. Consequently, it emerges as a crucial contender in the selection of our final approach. The confusion matrix created using LRCN is shown in Fig. 25.

4.4 Findings

Fig. 26: Conventional activity : walking

Fig. 27: Conventional activity : Running

Fig. 28: Suspicious activity : Fight

Fig. 26 and Fig. 27 show the conventional activity which is walking and running respectively while Fig. 28 shows suspicious activity that is fight.

A high precision value indicates that the model has a minimal rate of false positives. This means that when the model predicts a positive instance, it is highly likely to be correct. Precision holds significant value in scenarios where false positives carry significant consequences, such as medical diagnosis or spam detection. A high recall value indicates that the model has a minimal rate of false negatives. This implies that the model can effectively detect the majority of positive instances. Recall holds significant importance in situations where the omission of positive instances can have significant consequences, such as disease detection or fraud identification. With a range of 0 to 1, where 1 signifies perfect precision and recall, a higher F1 score indicates a more favorable balance between the two. A higher F1 score implies that the model exhibits strong overall performance. Although accuracy is a valuable metric, it can be deceptive when dealing with imbalanced classes. For instance, in scenarios where the negative class dominates, a model that consistently predicts negative will yield a high accuracy, even if it fails to identify any positive instances. Consequently, the interpretation of accuracy should be approached cautiously, taking into account the class distribution.

In conclusion, after thorough experimentation and analysis, it was determined that the LSTM-DenseNet121 algorithm employed in this research paper consistently demonstrated superior performance compared to other algorithms evaluated. The rigorous evaluation process involved comparing multiple machine learning algorithms using various metrics such as accuracy, precision, recall, and F1 score (Table 1). The selection of this algorithm as the primary model was not arbitrary but rather a well-informed decision based on empirical evidence. Its superior performance and robustness in handling various forms of suspicious activities provided a strong foundation for the development of an effective suspicious activity recognition (SAR) system. By adopting this algorithm as the main DL model for our project on the basis of overall accuracy (Table 2), we can confidently rely on its capabilities to monitor public areas, detect and classify suspicious activities, and generate timely warning messages. The utilization of this algorithm will contribute significantly to enhancing the overall security monitoring capabilities, reducing response time, and ensuring public safety. It is important to note that while this algorithm has demonstrated exceptional performance, continuous monitoring and evaluation will be necessary to adapt and improve the SAR system as new data and challenges emerge. Nonetheless, the evidence presented in this research paper supports the conclusion that this algorithm stands as the optimal choice for achieving our objectives and addressing the challenges of accurate suspicious activity recognition in CCTV surveillance.

4.5 Proposed Architecture

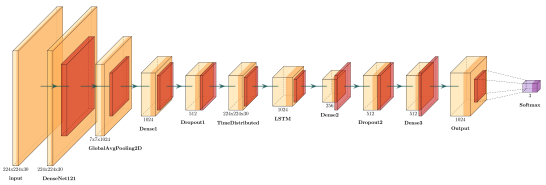

Fig. 29: Architecture of DenseNet121-LSTM

The model is designed to process videos consisting of 30 frames, each with an image height and width of 224 pixels and 3 color channels red, green, blue(RGB). The architecture as shown in Fig. 29 begins with an input layer that receives the video input. The base CNN layer, DenseNet121, is then employed as a feature extractor, initialized with pre-trained weights from the ImageNet dataset. A GlobalAveragePooling2D layer follows, reducing the spatial dimensions of the extracted features while preserving their depth. Next, several dense layers are added on top of the pooled features. A dense layer with 512 units, ReLU activation, and L2 regularization is incorporated, followed by a dropout layer to mitigate overfitting. The encoded frames are then processed through a TimeDistributed layer, which applies the DenseNet121 CNN to each frame independently, maintaining the sequence dimension. To capture temporal dependencies, an LSTM layer is introduced, which takes in the encoded frames and produces an encoded sequence. This sequence is subsequently passed through additional dense layers with 512 units, ReLU activation, and L2 regularization, accompanied by dropout layers to further regularize the model. Furthermore, a hidden layer with 1024 units and ReLU activation,

Table 1: Classification Results

Algorithms →		CNN	LSTM-VGG16	LSTM-ResNet50	LSTM-EfficientNetB0	LSTM-InceptionNetV3	LSTM-DenseNet121	LRCN
Label 0	Precision	0.62	0.77	0.0	0.0	0.88	0.83	0.75
	Recall	0.45	0.63	0.0	0.0	0.72	0.90	0.87
	F1 Score	0.52	0.70	0.0	0.0	0.79	0.86	0.80
	Accuracy	0.73	0.82	0.64	0.67	0.88	0.91	0.86
Label 1	Precision	1.0	1.0	1.0	0.0	1.0	1.0	1.0
	Recall	1.0	1.0	0.91	0.0	1.0	1.0	1.0
	F1 Score	1.0	1.0	0.95	0.0	1.0	1.0	1.0
	Accuracy	1.0	1.0	0.97	0.64	1.0	1.0	1.0
Label 2	Precision	0.57	0.69	0.50	0.32	0.76	0.9	0.87
	Recall	0.72	0.81	1.0	1.0	0.90	0.81	0.75
	F1 Score	0.64	0.75	0.66	0.48	0.83	0.85	0.81
	Accuracy	0.73	0.82	0.67	0.32	0.88	0.91	0.86

Table 2: Overall accuracy result

Algorithms	CNN	LSTM-VGG16	LSTM-ResNet50	LSTM-EffecientNetB0	LSTM-InceptionNetV3	LSTM-DenseNet121	LRCN
Accuracy(%)	73.53	82.35	64.70	32.35	88.23	91.17	86.67

combined with L2 regularization, enhances the model's representation capacity. The final output layer employs a softmax activation function to predict the class probabilities for the given video. The number of units in this layer corresponds to the length of the CLASSES_LIST, which is 3 in this instance. The model is compiled with a categorical cross-entropy loss and the Adam optimizer. Overall, this model architecture effectively combines the strengths of DenseNet121 for frame-level feature extraction with LSTM layers for capturing temporal information in videos. It is trained to classify videos into different classes and offers flexibility for further customization and fine-tuning based on specific video classification tasks.

To summarize, the architecture starts by processing each frame of the input video through a DenseNet121 base CNN to extract features. The features are then pooled, followed by additional dense layers with dropout regularization. The TimeDistributed layer applies the CNN to each frame independently, and the LSTM layer processes the sequence of features. Finally, the output is obtained by passing the LSTM output through dense layers, including the output layer that predicts the class probabilities.

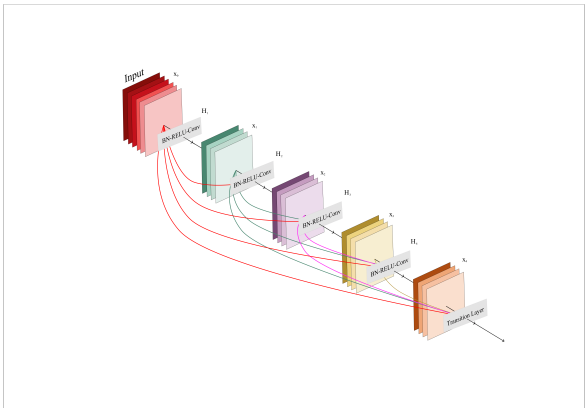

Fig. 30: Internal layers of DenseNet121

5 Conclusion

After thorough experimentation and analysis, the LSTM-DenseNet121 algorithm emerged as the top performer in this research paper, surpassing other algorithms in terms of accuracy, precision, recall, and F1 score. Its selection as the primary model was based on empirical evidence, showcasing its superior performance and robustness in detecting suspicious activities. By employing this algorithm, the project aims to enhance security monitoring capabilities, reduce response time,

and ensure public safety. Continuous monitoring and evaluation will be vital for adapting and improving the system. Overall, the accuracy evidence of 91.17% supports the conclusion that the LSTM-DenseNet121 algorithm is the optimal choice for accurate suspicious activity recognition in CCTV surveillance. In conclusion, this research paper presented the development of a suspicious activity recognition (SAR) system using machine learning for CCTV surveillance. The introduction highlighted the growing reliance on technology for safety and security, particularly the use of CCTV cameras in public areas. The motivation behind the project was to address the need for an automated system that can effectively detect suspicious activity in real-time, reducing the limitations and biases associated with manual monitoring. The problem statement emphasized the challenge of accurately identifying and classifying suspicious activities captured by CCTV cameras, highlighting the diverse nature of the suspicious activity and the limitations of manual monitoring. The aim and objectives of the project were defined, focusing on the development of a machine learning-based SAR system that can monitor public areas, detect suspicious activities, and alleviate the workload of physical security personnel. The scope of the project outlined the design and implementation of the SAR system, including the collection of labeled datasets, training of the machine learning model, and integration of real-time monitoring capabilities. It was made clear that the project did not cover the physical installation of CCTV cameras or the management of the surveillance infrastructure. In summary, this research paper aimed to enhance security monitoring capabilities and contribute to public safety by developing a robust SAR system. By leveraging machine learning algorithms, the system can accurately detect and classify various forms of activities such as conventional activities (walking, running) and suspicious activity (fighting), generate warning messages, and reduce response time to potential threats. The conclusion sets the stage for the subsequent sections of the paper, which will delve into the literature review, research methodology, results, and analysis.

5.1 Limitations

While suspicious activity recognition in CCTV surveillance has demonstrated its benefits, it also faces several limitations and challenges. One of the shortcomings is the ambiguity and subjectivity associated with identifying suspicious activity. The interpretation of suspicious behavior can vary among human operators or predefined rules, leading to inconsistent detection results. Additionally, CCTV cameras often have a limited field of view and lack contextual understanding, which can result in false positives or false negatives due to the absence of relevant contextual information. Occlusion and environmental factors pose challenges to accurate suspicious activity recognition, as objects or individuals can be hidden from the camera's view, and lighting conditions, shadows, reflections, and other environmental factors can impact algorithm accuracy. Real-world scenarios with crowded scenes, diverse activities, and varying environmental conditions make it difficult for algorithms to detect subtle or nuanced suspicious activities. High false alarm rates overwhelm operators, reducing trust and effectiveness. Privacy concerns arise with continuous monitoring and analysis of CCTV footage, necessitating a balance between effective surveillance and privacy protection. Scarcity of classification labels limits the ability to accurately detect diverse suspicious behaviors. Availability of diverse labeled data is crucial for reliable training, necessitating continuous expansion of the dataset with new examples of suspicious activities. Addressing these shortcomings requires ongoing research and development efforts in the field of video analytics. Advancements in computer vision, machine learning, and data fusion techniques are necessary to improve the understanding of complex scenes, refine anomaly detection algorithms, and consider contextual information. These areas of focus aim to enhance the accuracy and reliability of suspicious activity recognition in CCTV surveillance systems.

5.2 Future Scope

The project has several potential future scope areas for development and enhancement. These include integrating real-time object tracking to analyze the movement of suspicious objects or individuals across multiple CCTV camera feeds.

Multimodal data fusion, incorporating audio and sensor data, can enhance the system's capability to detect and classify suspicious activities. Anomaly detection techniques can be explored to identify novel or previously unseen suspicious activities, ensuring adaptability against emerging threats. Advanced human behavior analysis techniques such as gait recognition, crowd behavior analysis, or gesture recognition can help detect abnormal behavioral patterns more accurately. Integration with edge computing and IoT devices can enhance scalability and enable distributed surveillance capabilities. Collaboration with law enforcement agencies can refine system performance by incorporating real-time face recognition and matching against criminal databases. Continuous performance optimization in terms of speed, computational efficiency, and memory usage is essential for practical deployment. Finally, integrating a warning system based on monitoring individuals' time in specific areas can provide timely alerts if thresholds are exceeded. These future scope areas provide a starting point for further research and development, aiming to enhance the effectiveness, adaptability, and applicability of the suspicious activity recognition system in CCTV surveillance.

Declarations

- **Funding** The author(s) received no financial support for the research, authorship, and/or publication of this manuscript.
- **Competing interests** The author(s) declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article. In addition, the authors have no relevant financial or non-financial interests to disclose.
- **Ethics approval** Throughout the project, we have taken great care to adhere to ethical principles, including respect for privacy, confidentiality, and the responsible use of the open-source dataset. All data handling, storage, and analysis procedures were conducted in compliance with relevant regulations and guidelines.
- **Consent to Participate** The dataset used consisted of pre-existing videos, there were no individual participants involved, and no direct interaction or informed consent process was required.The individuals featured in the video dataset utilized for this project were obtained from an open-source video dataset. As a result, there is no identifiable personal information associated with the videos used in this project.
- **Consent for publication** The human subjects featured in the video dataset used for this project were sourced from an open-source video dataset. Therefore, there were no individual participants involved, and no direct interaction or informed consent process was required. We acknowledge that the dataset used in our research does not contain any personally identifiable information or sensitive data. Our research findings are solely based on the analysis and comparison of the dataset
- **Availability of data and materials** Data used in this research are available upon request
- **Code availability** Code used in this research is available upon request.
- **Authors' contributions** Dhruv Saluja involved in methodology, validation, formal analysis, writing—review & editing, Harsh Kukreja took part in methodology, writing—original draft, writing—review & editing. Akash Saini participated in writing—original draft, conceptualization, writing—review & editing. Devanshi Tegwal participated in writing—review & editing. Preeti Nagrath guided and reviewed the research work and research paper.

References

A. Adam, E. Rivlin, I. Shimshoni and D. Reinitz, "Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555-560, March 2008, doi: 10.1109/TPAMI.2007.70825.

Boudihir, Elarbi. (2012). INTELLIGENT VIDEO SURVEILLANCE SYSTEM ARCHITECTURE FOR ABNORMAL ACTIVITY DETECTION. The International Conference on Informatics and Applications (ICIA2012). 102-111.

G. Chéron, I. Laptev and C. Schmid, "P-CNN: Pose-Based CNN Features for Action Recognition," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 3218-3226, doi: 10.1109/ICCV.2015.368.

R. De Geest and T. Tuytelaars, "Modeling Temporal Structure with LSTM for Online Action Detection," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 2018, pp. 1549-1557, doi: 10.1109/WACV.2018.00173.

Konstantinos Gkountakos, Despoina Touska, Konstantinos Ioannidis, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2021. Spatio-Temporal Activity Detection and Recognition in Untrimmed Surveillance Videos. In Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21). Association for Computing Machinery, New York, NY, USA, 451–455. https://doi.org/10.1145/3460426.3463591

C. Kyrkou and T. Theocharides, "EmergencyNet: Efficient Aerial Image Classification for Drone-Based Emergency Monitoring Using Atrous Convolutional Feature Fusion," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 1687-1699, 2020, doi: 10.1109/JSTARS.2020.2969809.

Hou, Rui & Chen, Chen & Shah, Mubarak. (2017). Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos. 5823-5832. 10.1109/ICCV.2017.620.

Zhao L and Huang L. An Investigation on Sparsity of CapsNets for Adversarial Robustness. Proceedings of the 1st International Workshop on Adversarial Learning for Multimedia. (55-61). https://doi.org/10.1145/3475724.3483609

O. P. Popoola and K. Wang, "Video-Based Abnormal Human Behavior Recognition—A Review," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 865-878, Nov. 2012, doi: 10.1109/TSMCC.2011.2178594.

Elizabeth Scaria, Aby Abahai T, Elizabeth Isaac "Suspicious Activity Detection in Surveillance Video using Discriminative Deep Belief Network." (2017).

@INPROCEEDINGS1334462,
author=Schuldt, C. and Laptev, I. and Caputo, B., booktitle=Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., title=Recognizing human actions: a local SVM approach, year=2004, volume=3, number=, pages=32-36 Vol.3, doi=10.1109/ICPR.2004.1334462

Tripathi, Rajesh & Jalal, Anand & Agrawal, Subhash. (2018). Suspicious human activity recognition: a review. Artificial Intelligence Review. 50.10.1007/s10462-017-9545-7.

A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," in IEEE Access, vol. 6, pp. 1155-1166, 2018, doi: 10.1109/ACCESS.2017.2778011.

X. Wang, L. Gao, J. Song and H. Shen, "Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition," in IEEE Signal Processing Letters, vol. 24, no. 4, pp. 510-514, April 2017, doi: 10.1109/LSP.2016.2611485.

H. Yang, C. Yuan, J. Xing and W. Hu, "SCNN: Sequential convolutional neural network for human action recognition in videos," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 355-359, doi: 10.1109/ICIP.2017.8296302.

Yao, Guangle & Lei, Tao & Zhong, Jiandan. (2018). A Review of Convolutional-Neural-Network-Based Action Recognition. Pattern Recognition Letters. 118. 10.1016/j.patrec.2018.05.018.

S. Yeung, O. Russakovsky, G. Mori and L. Fei-Fei, "End-to-End Learning of Action Detection from Frame Glimpses in Videos," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 2678-2687, doi: 10.1109/CVPR.2016.293

Figure

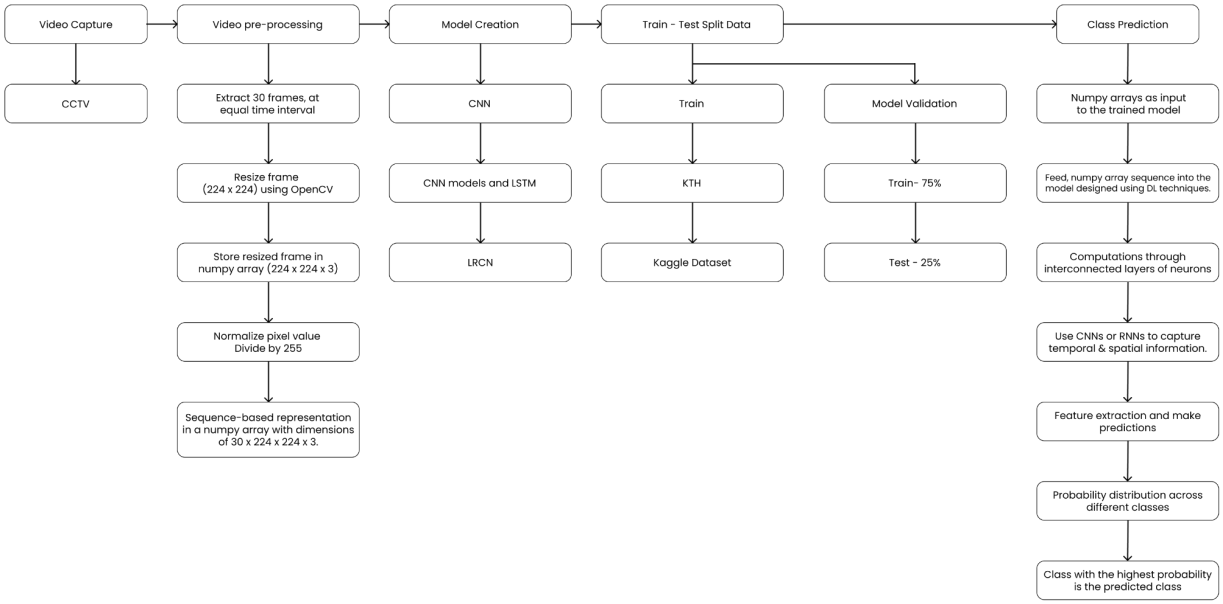

Fig. 1: Methodology

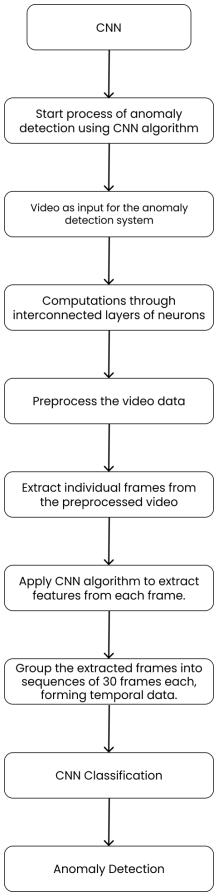

Fig. 2: Anomaly Detection using CNN

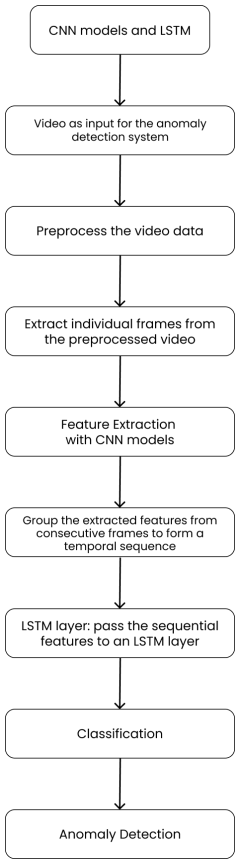

Fig. 3: Anomaly Detection using CNN models and LSTM

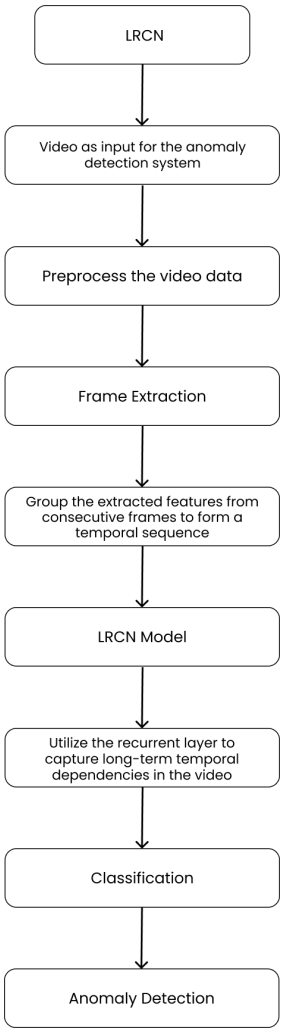

Fig. 4: Anomaly Detection using LRCN

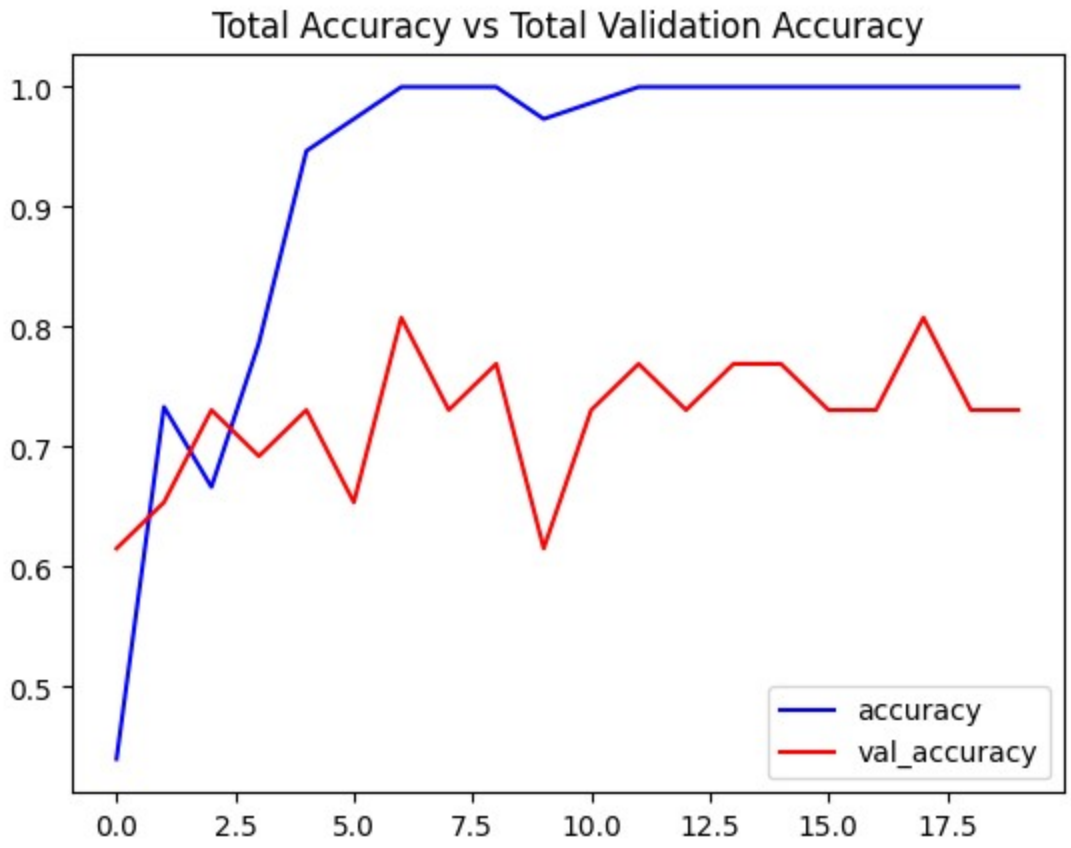

Fig. 5: Total Accuracy vs Total Validation Accuracy using CNN

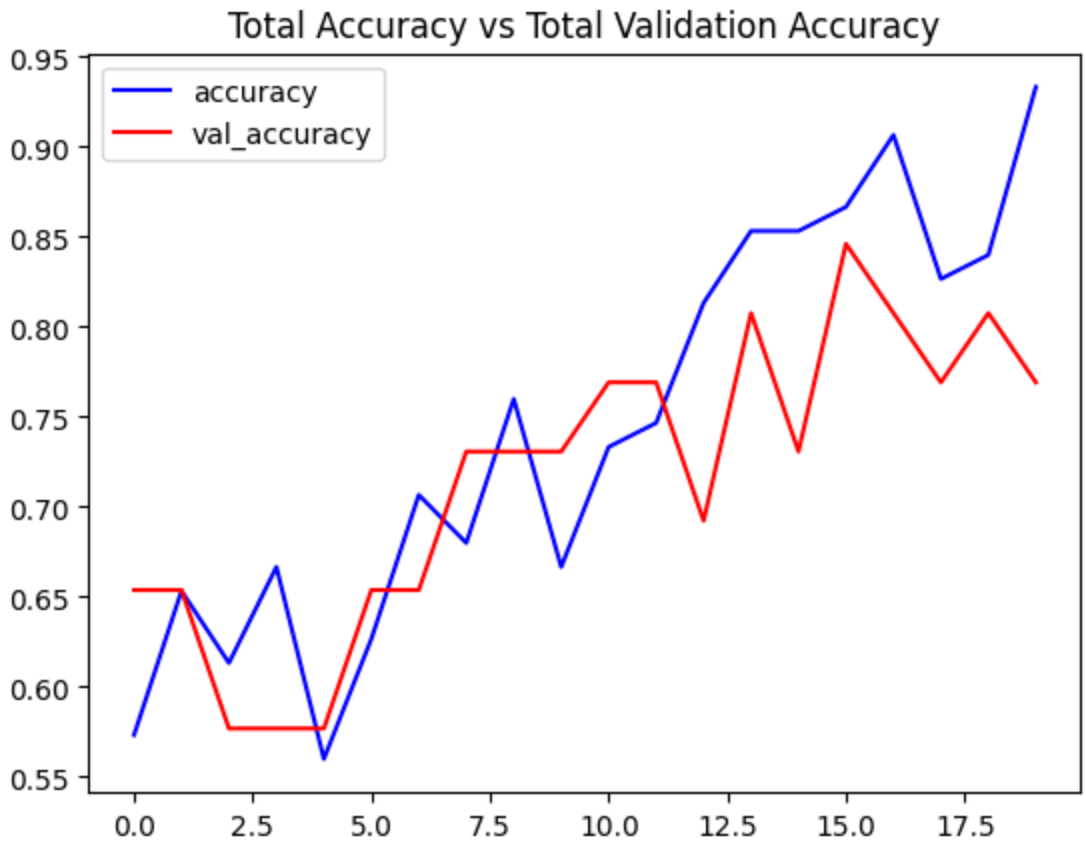

Fig. 6: Total Accuracy vs Total Validation Accuracy using LSTM - VGG16

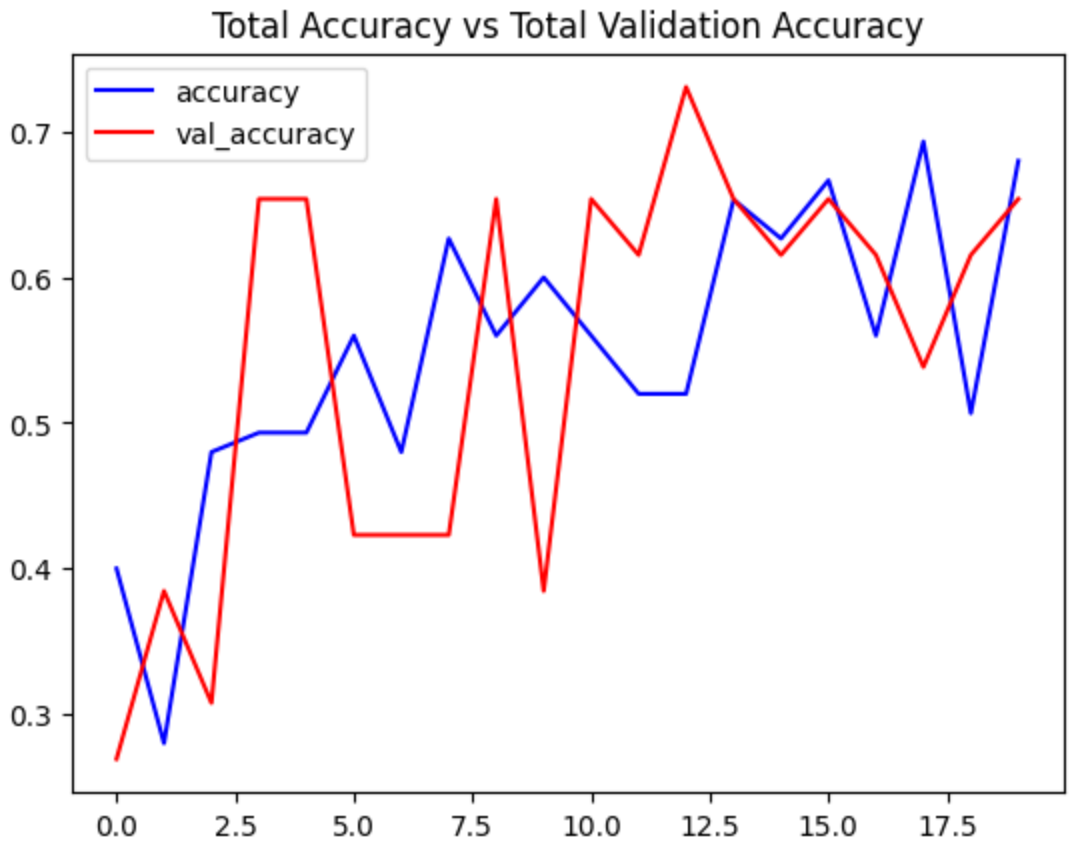

Fig. 7: Total Accuracy vs Total Validation Accuracy using LSTM - ResNet50

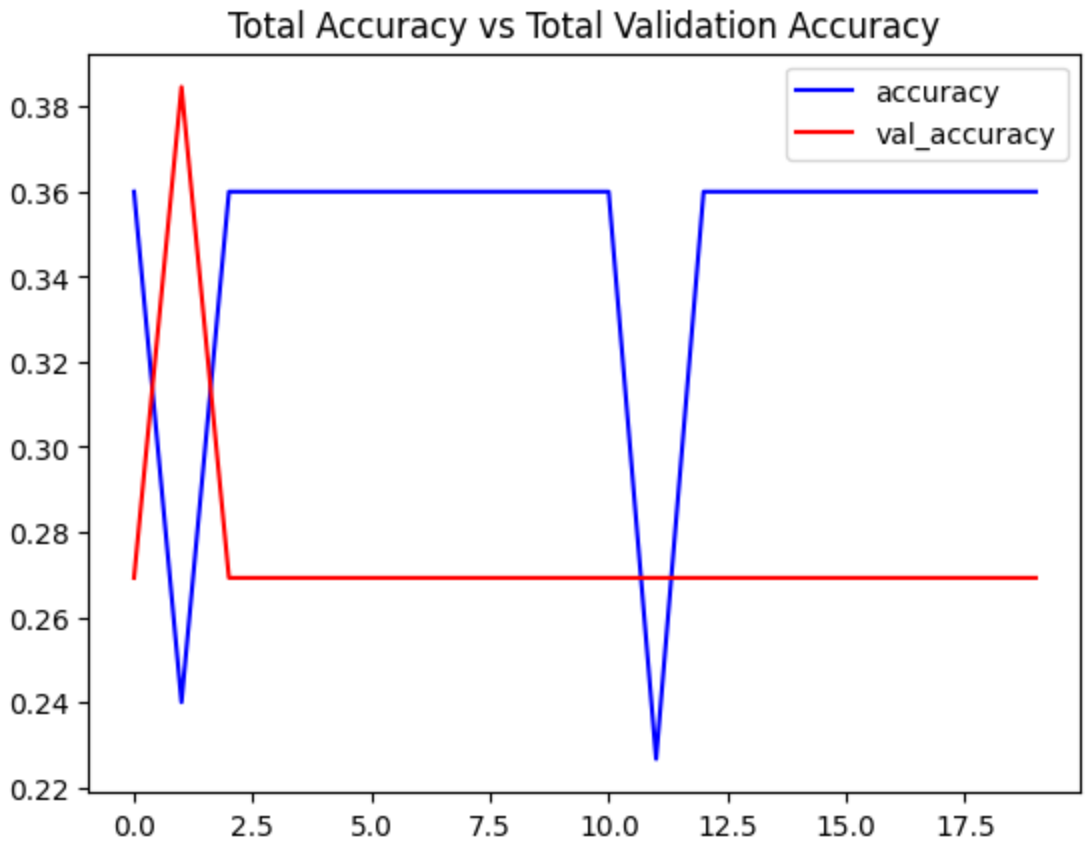

Fig. 8: Total Accuracy vs Total Validation Accuracy using LSTM - EfficientNetB0

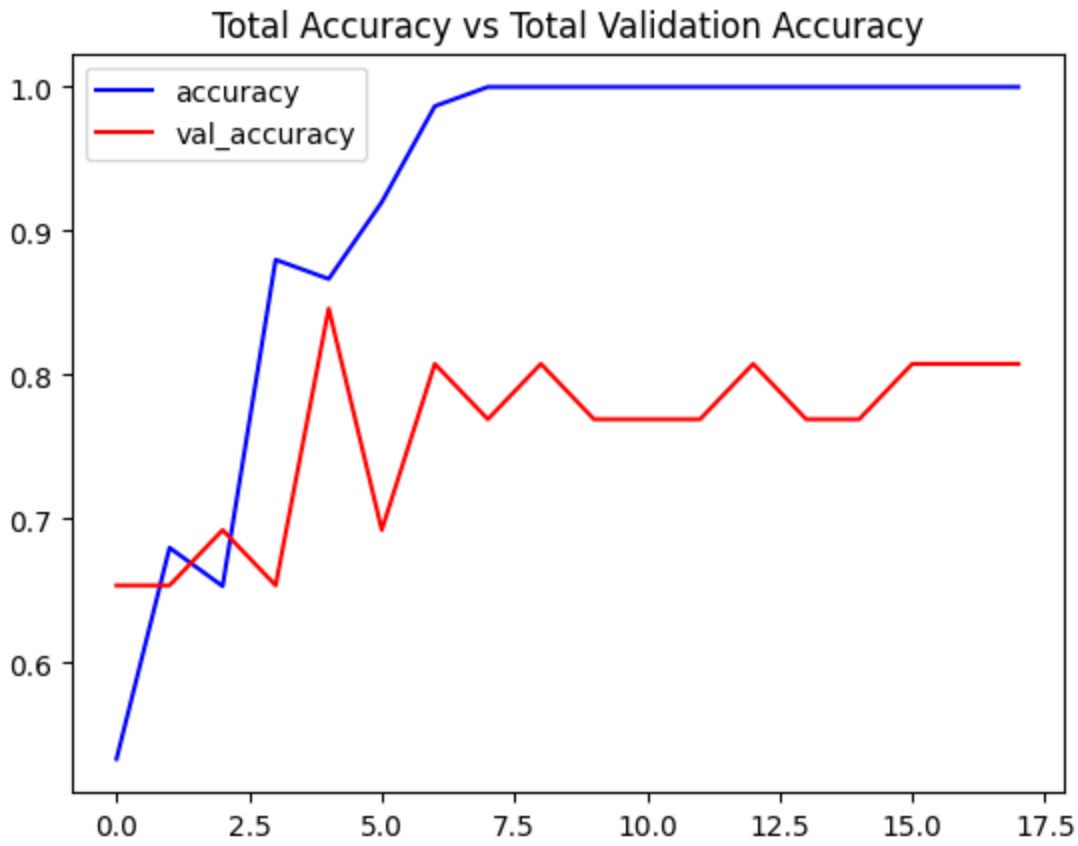

Fig. 9: Total Accuracy vs Total Validation Accuracy using LSTM - InceptionNetV3

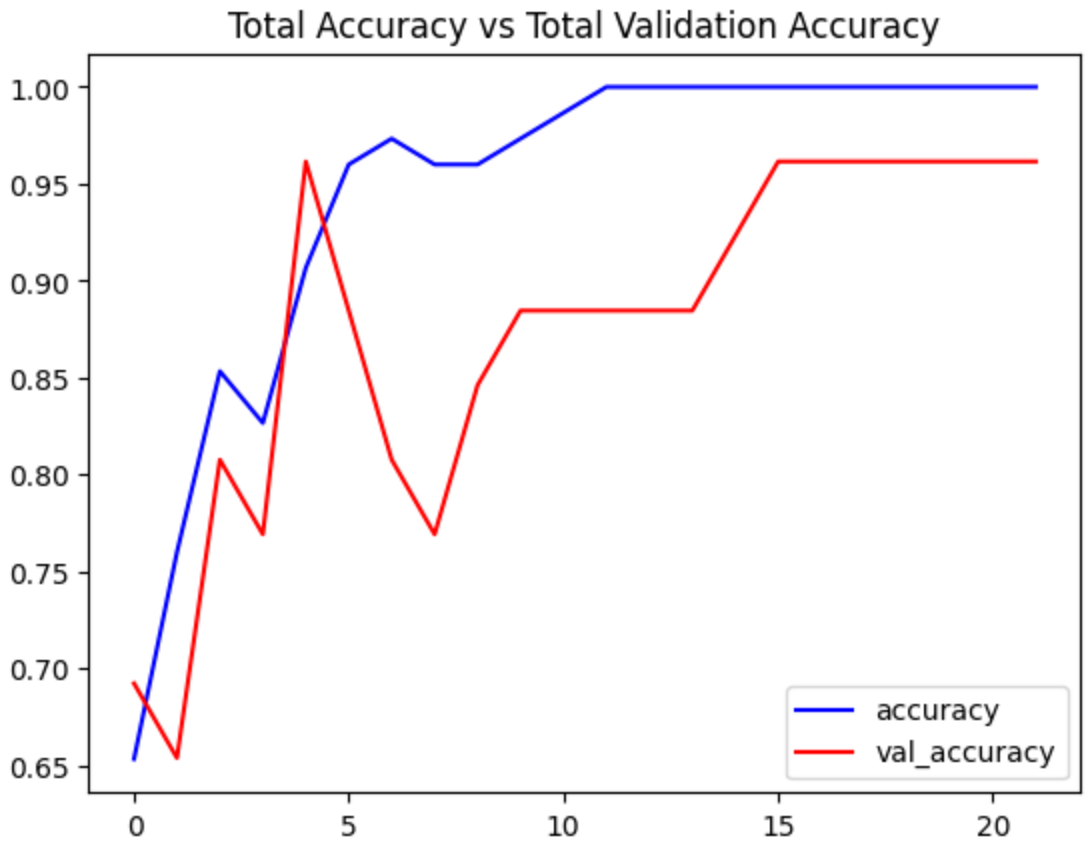

Fig. 10: Total Accuracy vs Total Validation Accuracy using LSTM - DenseNet121

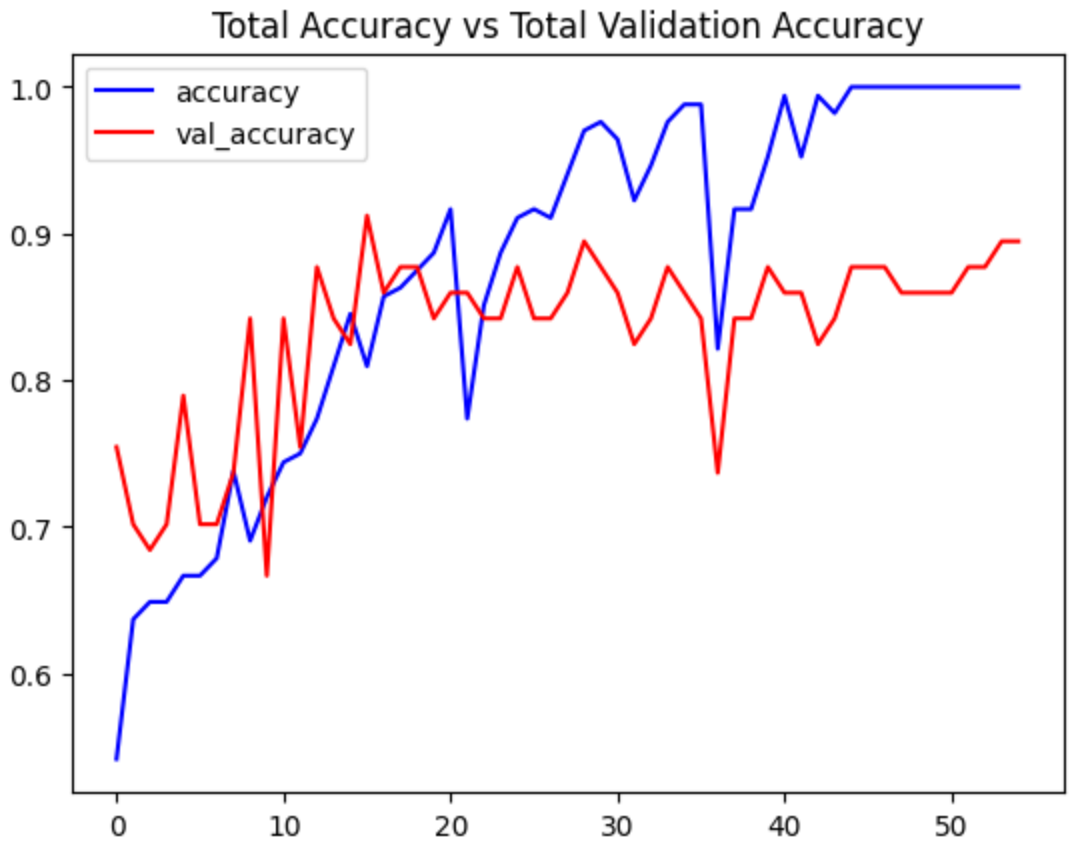

Fig. 11: Total Accuracy vs Total Validation Accuracy using LRCN

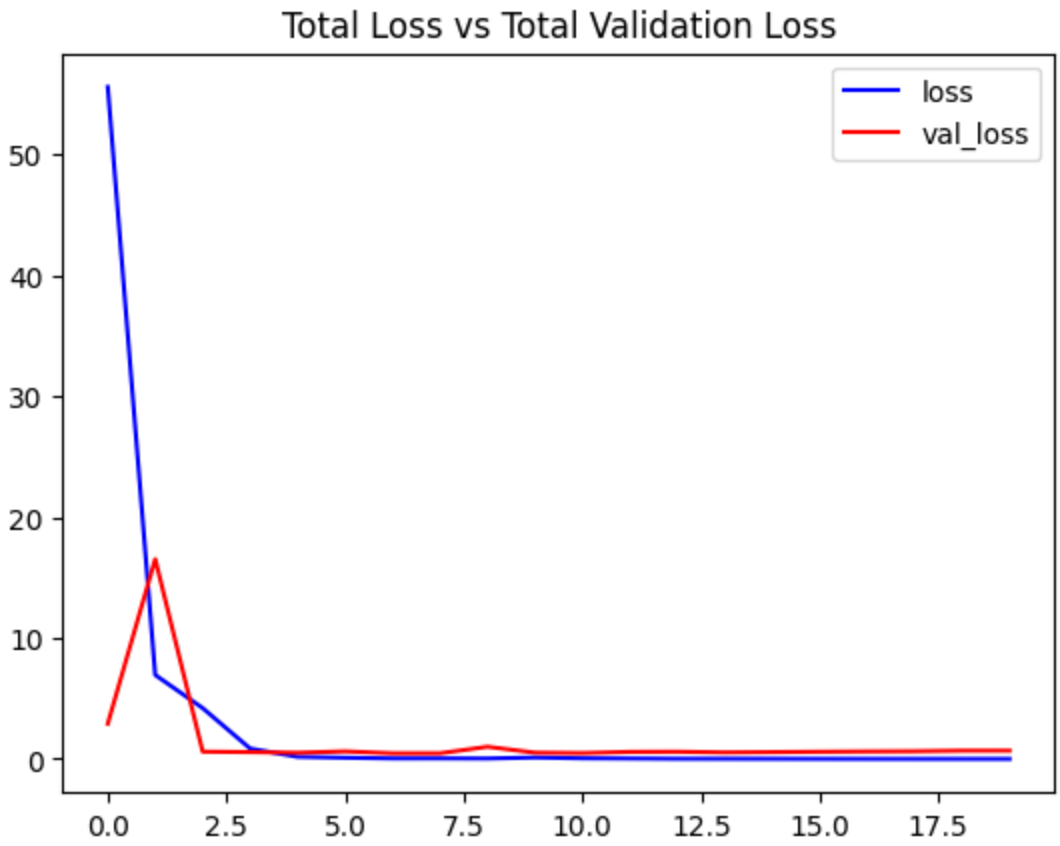

Fig. 12: Total Loss vs Total Validation Loss using CNN

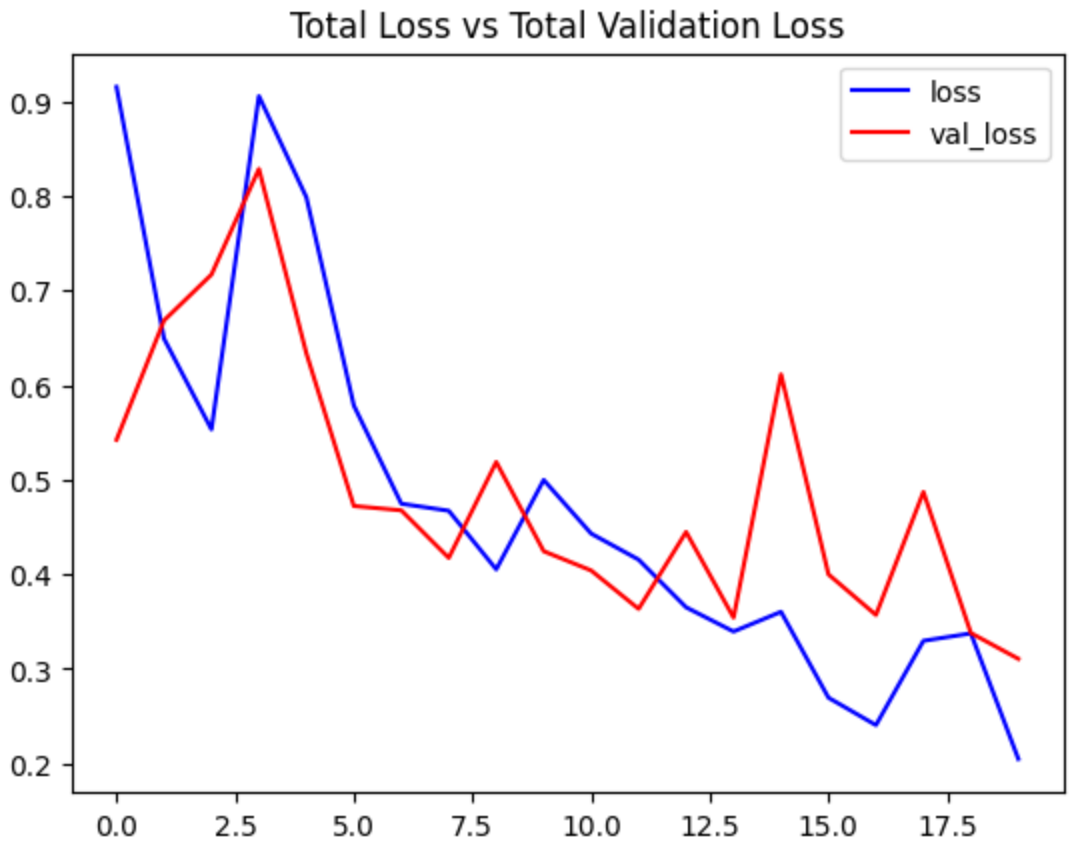

Fig. 13: Total Loss vs Total Validation Loss
using LSTM - VGG16

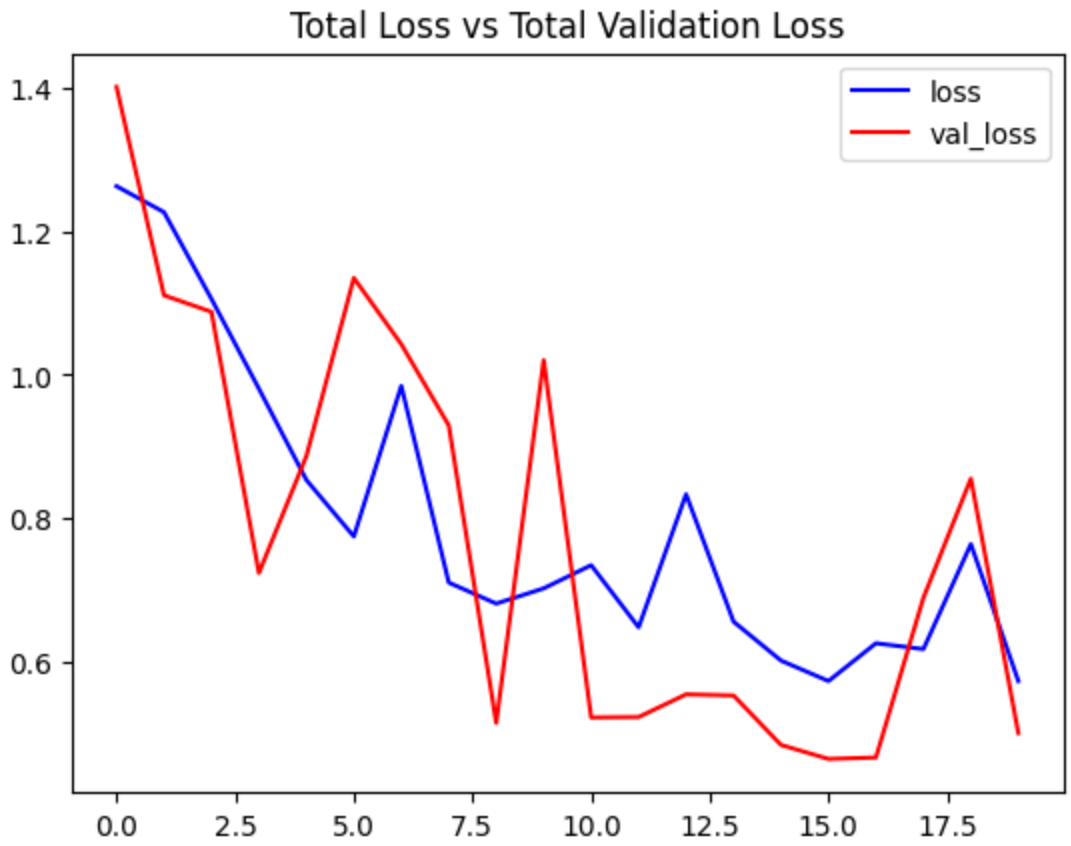

Fig. 14: Total Loss vs Total Validation Loss using LSTM - ResNet50

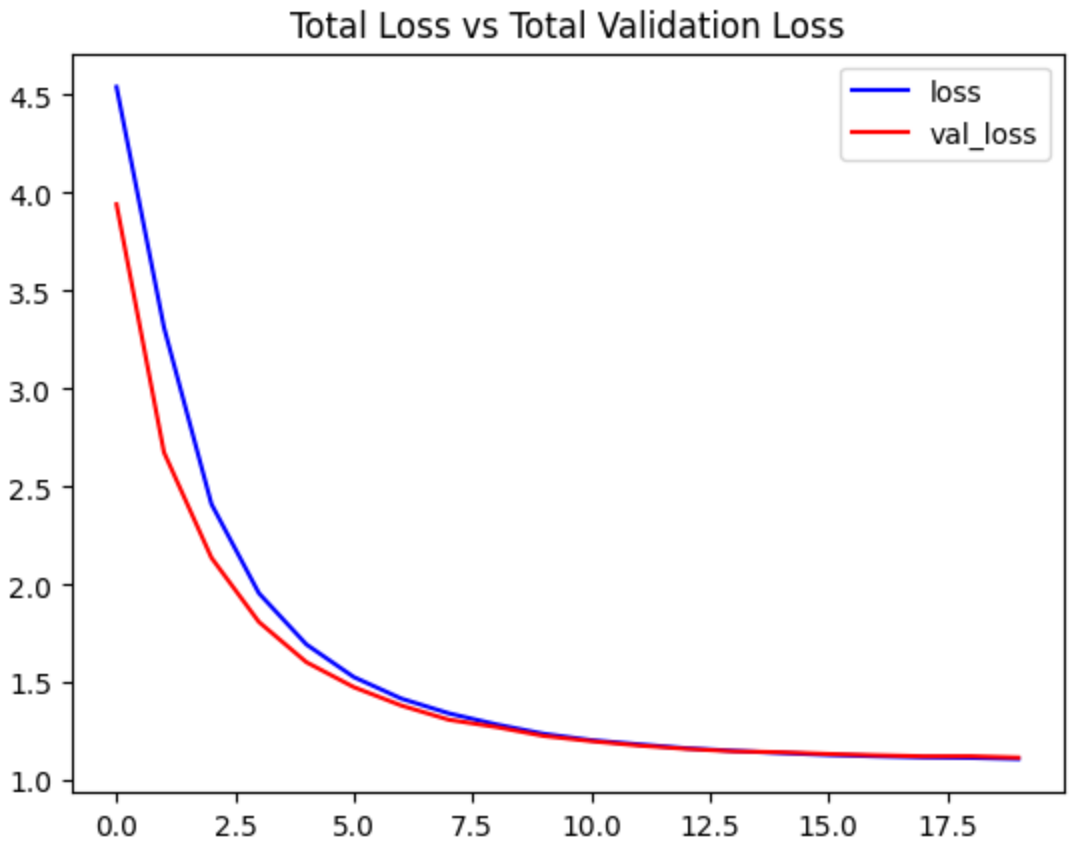

Fig. 15: Total Loss vs Total Validation Loss using LSTM - EfficientNetB0

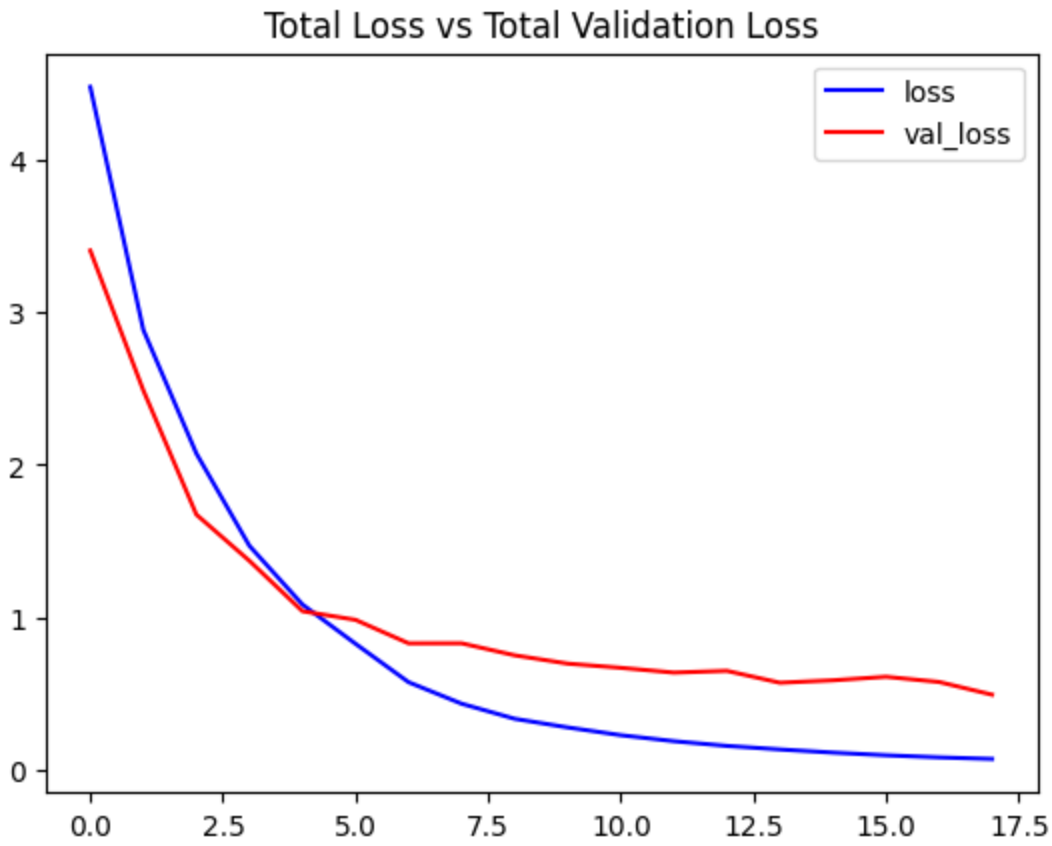

Fig. 16: Total Loss vs Total Validation Loss using LSTM - InceptionNetV3

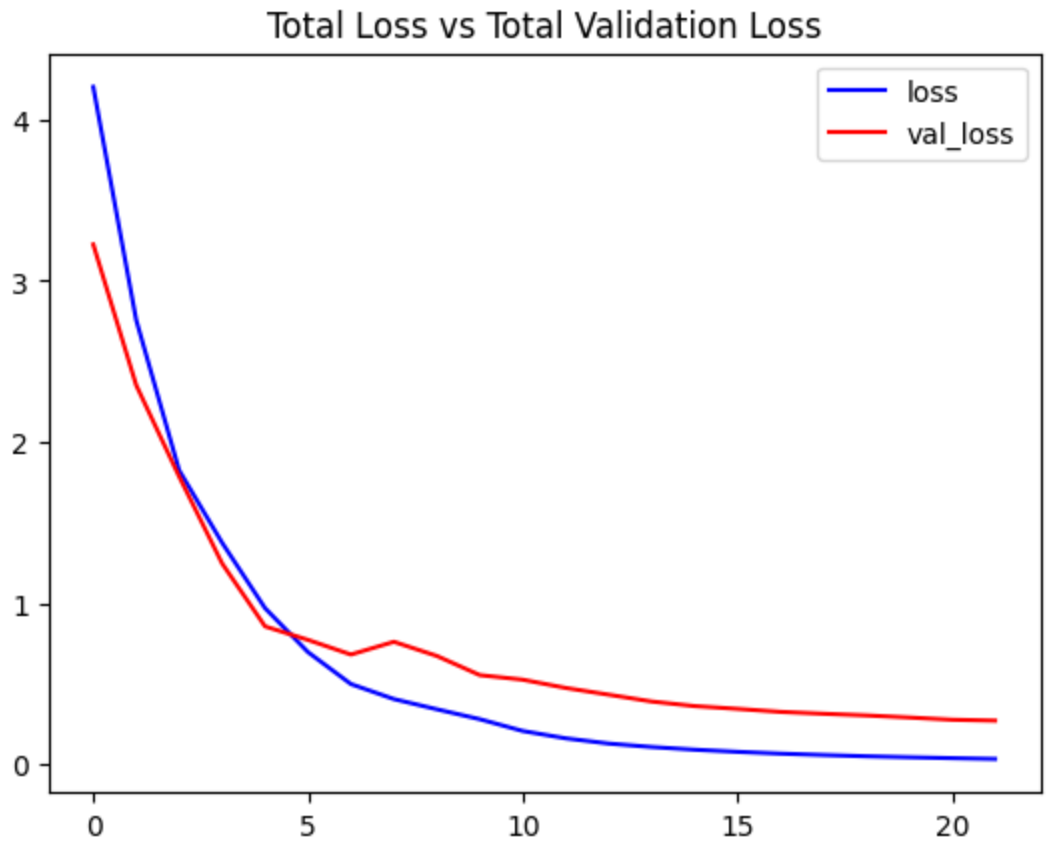

Fig. 17: Total Loss vs Total Validation Loss using LSTM - DenseNet121

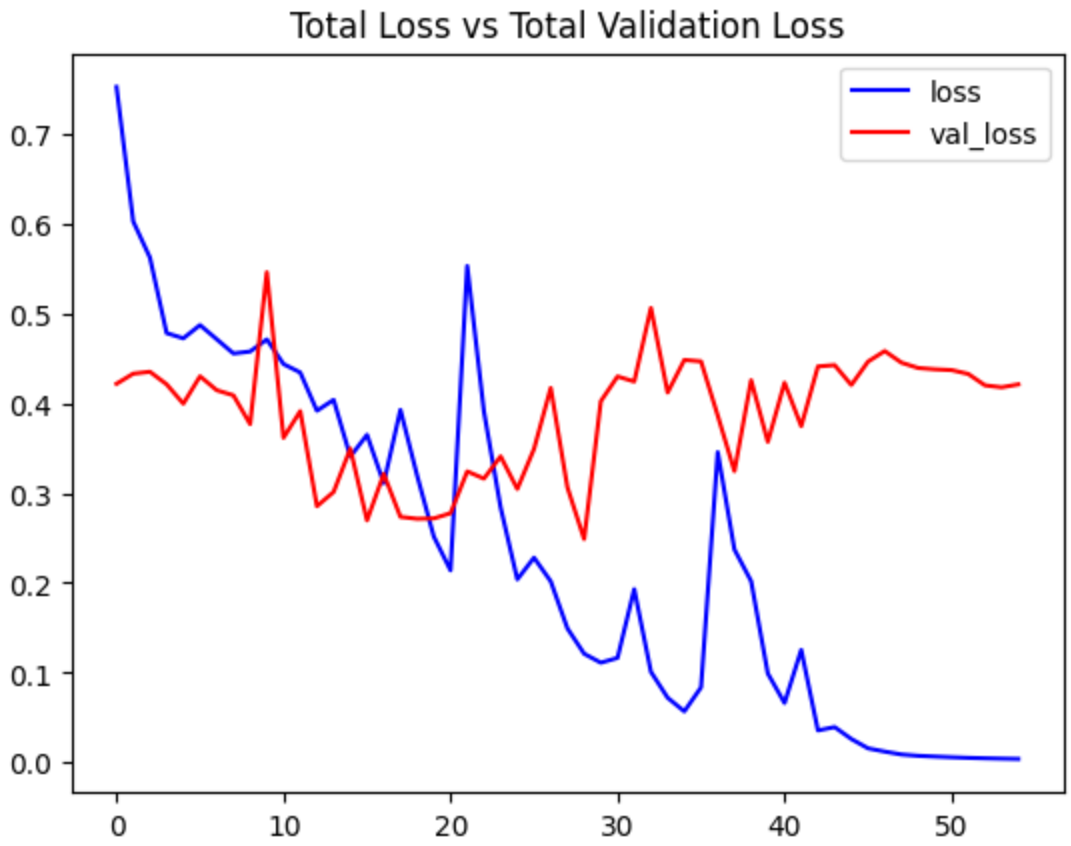

Fig. 18: Total Loss vs Total Validation Loss using LRCN

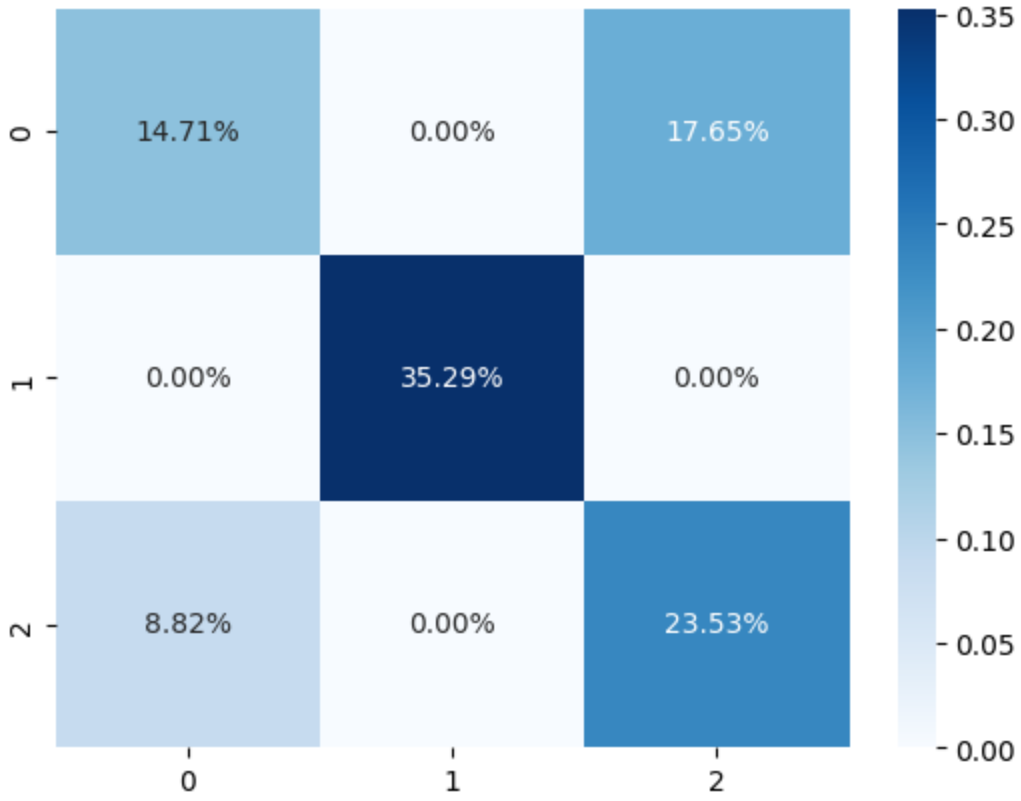

Fig. 19: Confusion matrix using CNN

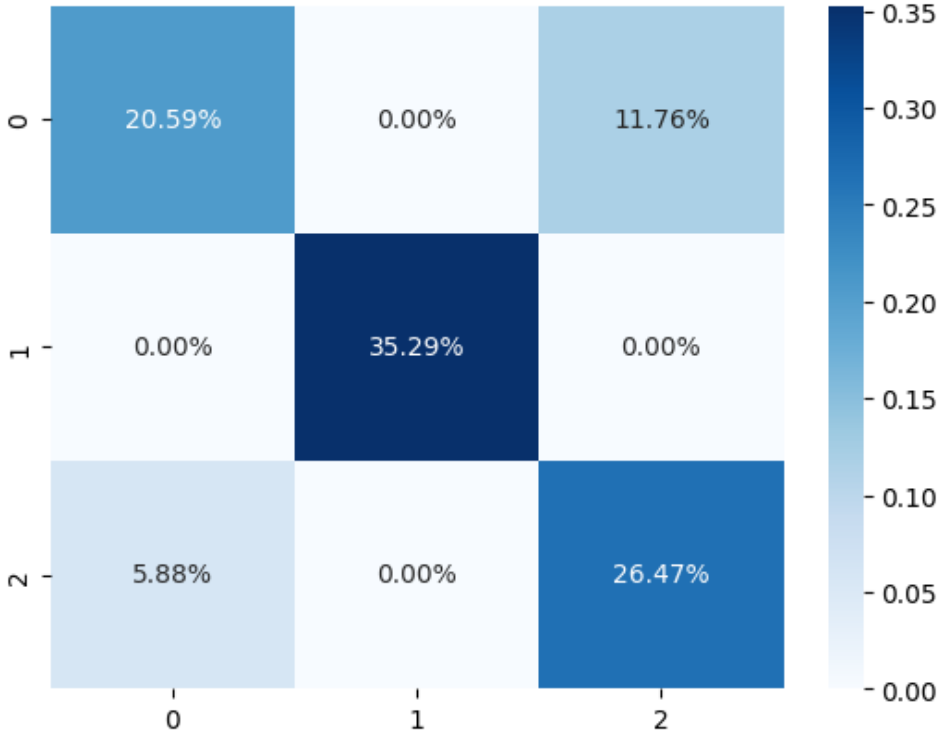

Fig. 20: Confusion matrix using LSTM - VGG16

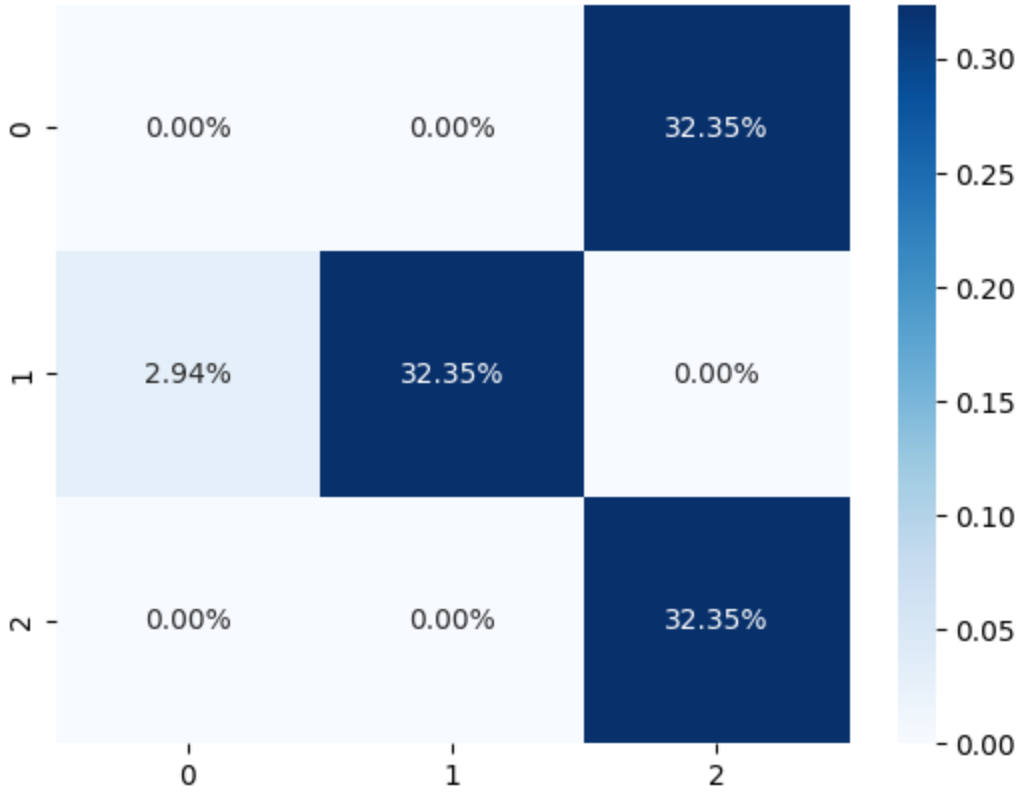

Fig. 21: Confusion matrix using LSTM - VGG16

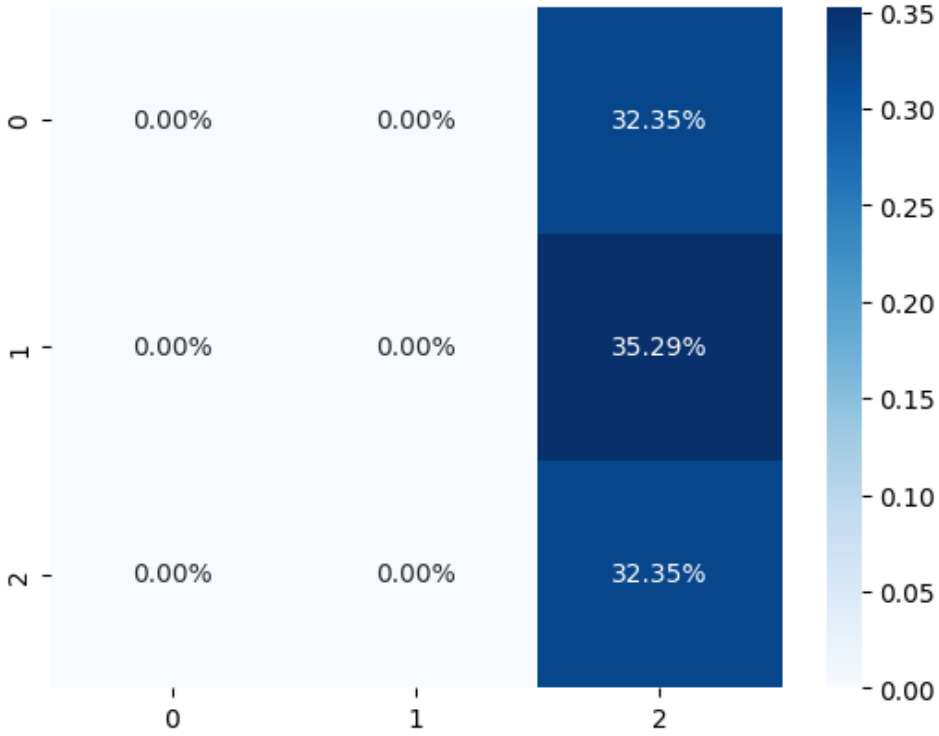

Fig. 22: Confusion matrix using LSTM - EfficientB0

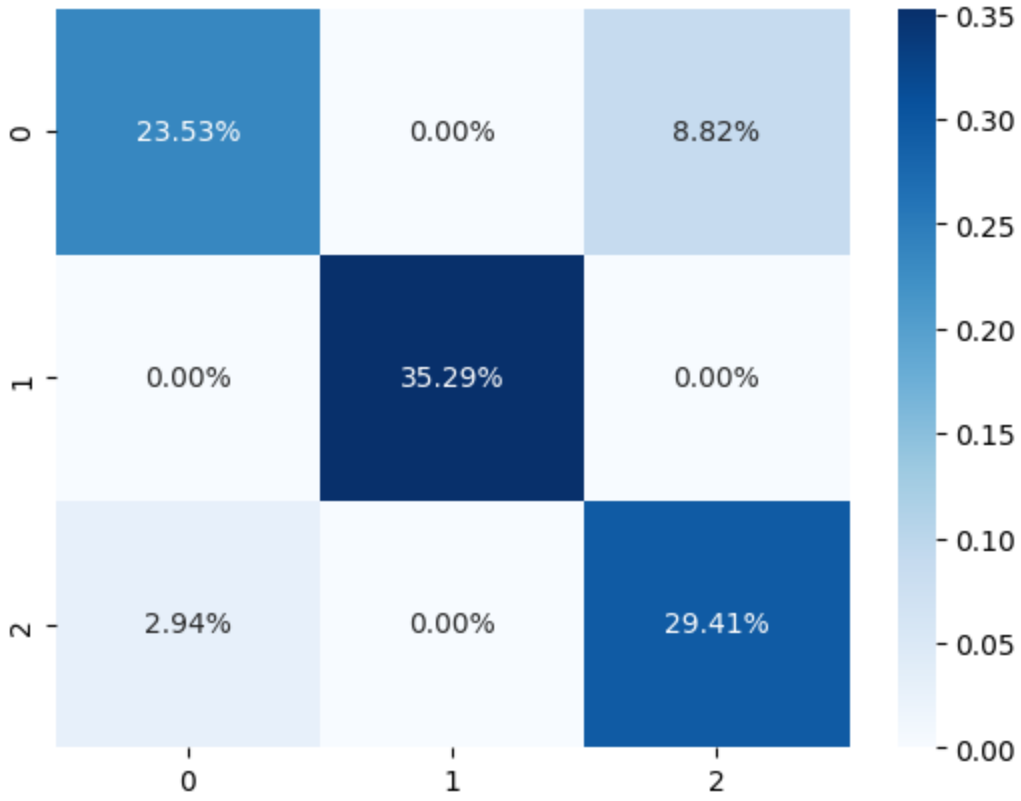

Fig. 23: Confusion matrix using LSTM - InceptionNetV3

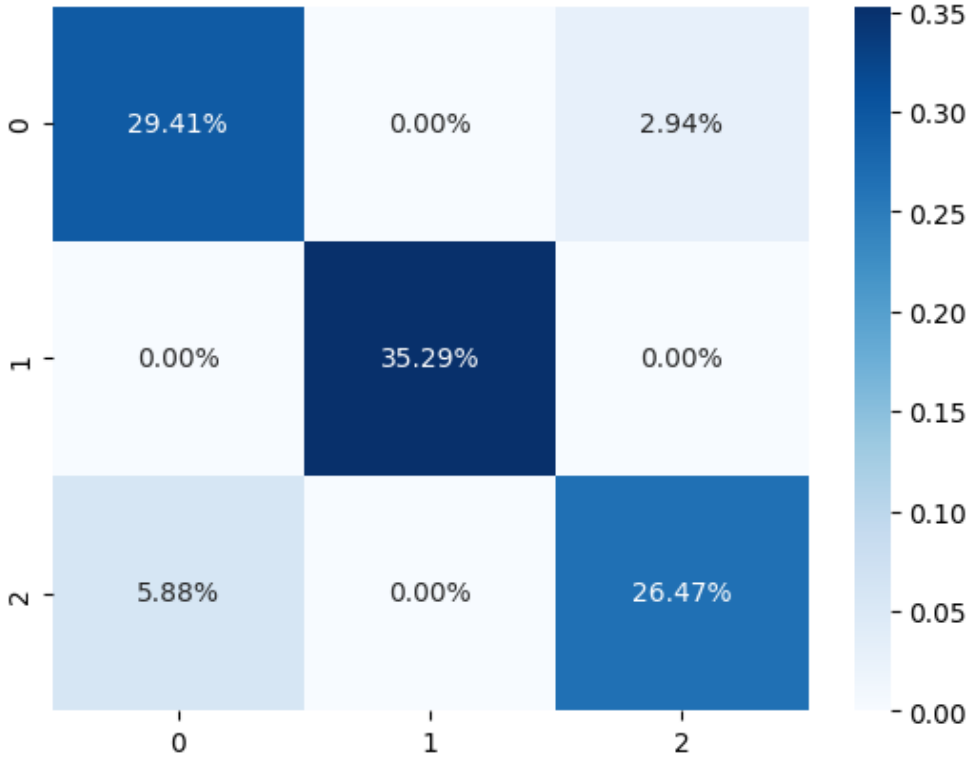

Fig. 24: Confusion matrix using LSTM - DenseNet121

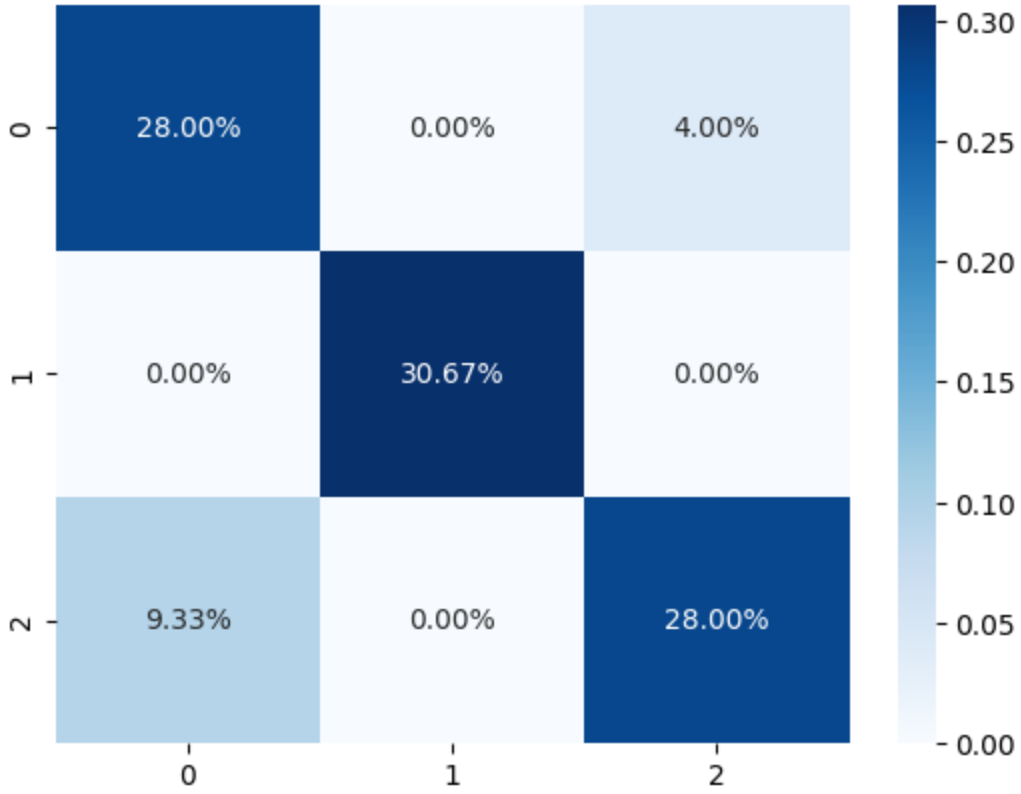

Fig. 25: Confusion matrix using LRCN

Fig. 26: Conventional activity : walking

Fig. 27: Conventional activity : Running

Fig. 28: Suspicious activity : Fight

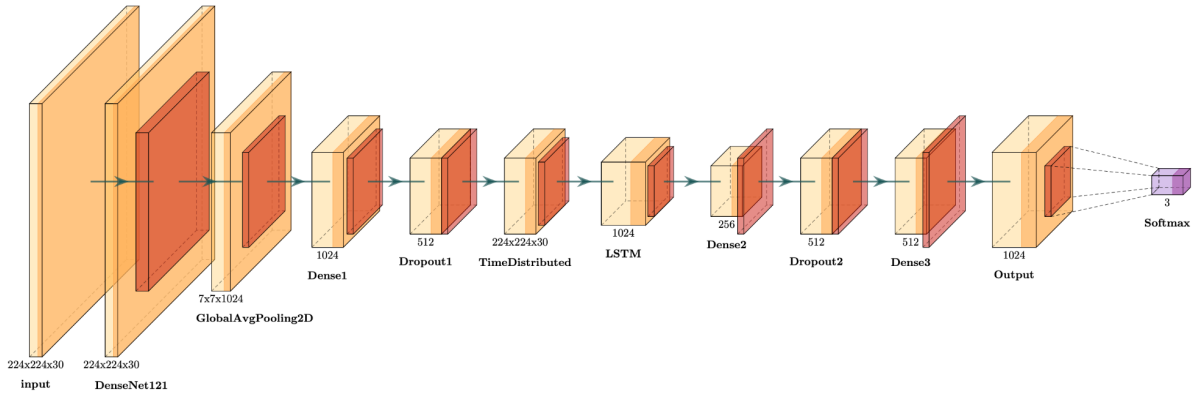

Fig. 29: Architecture of DenseNet121-LSTM

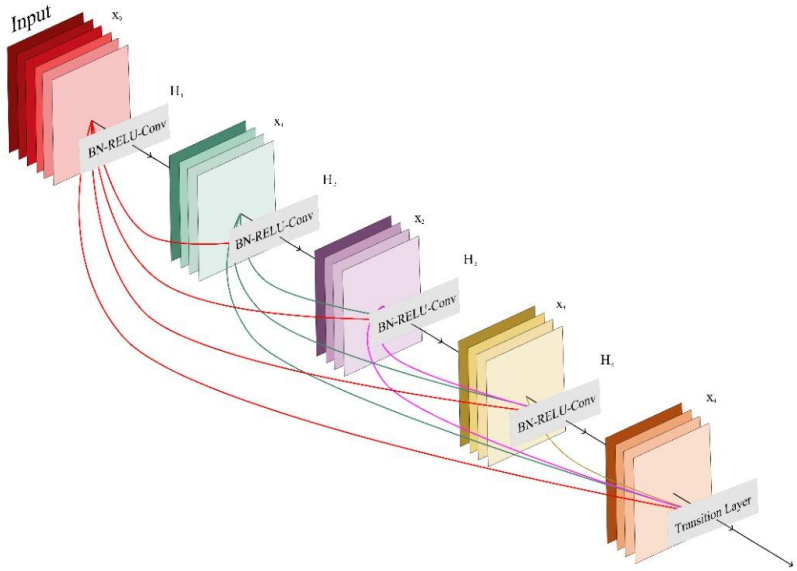

Fig. 30: Internal layers of DenseNet121

Table

Table 1: Classification Results

Algorithms →		CNN	LSTM-VGG16	LSTM-ResNet50	LSTM-EfficientNetB0	LSTM-InceptionNetV3	LSTM-DenseNet121	LRCN
Label 0	Precision	0.62	0.77	0.0	0.0	0.88	0.83	0.75
	Recall	0.45	0.63	0.0	0.0	0.72	0.90	0.87
	F1 Score	0.52	0.70	0.0	0.0	0.79	0.86	0.80
	Accuracy	0.73	0.82	0.64	0.67	0.88	0.91	0.86
Label 1	Precision	1.0	1.0	1.0	0.0	1.0	1.0	1.0
	Recall	1.0	1.0	0.91	0.0	1.0	1.0	1.0
	F1 Score	1.0	1.0	0.95	0.0	1.0	1.0	1.0
	Accuracy	1.0	1.0	0.97	0.64	1.0	1.0	1.0
Label 2	Precision	0.57	0.69	0.50	0.32	0.76	0.9	0.87
	Recall	0.72	0.81	1.0	1.0	0.90	0.81	0.75
	F1 Score	0.64	0.75	0.66	0.48	0.83	0.85	0.81
	Accuracy	0.73	0.82	0.67	0.32	0.88	0.91	0.86

Table 2: Overall accuracy result

Algorithms	CNN	LSTM-VGG16	LSTM-ResNet50	LSTM-EffecientNetB0	LSTM-InceptionNetV3	LSTM-DenseNet121	LRCN
Accuracy(%)	73.53	82.35	64.70	32.35	88.23	91.17	86.67