Suspicious Activity Recognition Using Machine Learning

PROJECT REPORT

BACHELOR OF TECHNOLOGY

Artificial Intelligence and Data Science

Submitted By:

Harshul, 2121723
Aarzoo Sharma, 2121695
Deepakshi Sharma, 2121714

Guided By:

Ms. Chaahat Gupta

Chandigarh Group of Colleges

College of Engineering, Landran

Jan,2024

Abstract

Detecting suspicious human activity is paramount for maintaining public safety and security. The goal is to discern behaviour that deviates from normal patterns. In pursuit of this objective, we leverage LSTM (Long Short-Term Memory), a recurrent neural network architecture renowned for handling sequential data, coupled with convolutional layers for extracting spatial features from video frames. This fusion of CNNs and RNNs enables comprehensive analysis of temporal data, vital for identifying anomalies in behaviour.

The project progresses through several key stages: research, data collection and preprocessing, model design and training, and performance evaluation. Each phase is essential for building a robust detection system capable of accurately flagging suspicious behaviour in real-time scenarios.

To kickstart the project, we utilized the KTH dataset, encompassing 600 frames capturing various instances of walking and running. Additionally, we incorporated the Kaggle dataset, comprising 100 training videos showcasing diverse activities. By leveraging these datasets, we ensured a diverse range of scenarios for training our model.

Upon completion, our analysis demonstrates promising results. The detection system, trained on a combination of KTH and Kaggle datasets, showcases an impressive 86% accuracy rate in identifying suspicious events. We anticipate further improvements in accuracy as we continue to expand the dataset, enabling the model to encounter a more extensive array of scenarios and refine its predictive capabilities.

In summary, our approach blends cutting-edge neural network architectures with meticulously curated datasets to develop a robust system capable of effectively identifying suspicious behaviour in real-world settings. This endeavour represents a significant stride towards enhancing public safety and security through advanced AI-driven surveillance technology.

Acknowledgement

It is with great pleasure that we present this project report on "Suspicious Activity Recognition" Our engagement in this Deep Learning and Machine Learning project has been an enlightening experience. Without the support and encouragement of those around us, this project would not have been possible.

We want to extend my sincere appreciation to Dr. Chaahat Gupta, our project advisor, for her invaluable guidance and support throughout the entire project development process. Her insights and encouragement inspired us to push ourselves further and embrace cutting-edge technology. Additionally, we are grateful for the self-development classes included in our university curriculum, which have provided us with the opportunity to explore various technologies and enhance our skills.

Table of Content

- ABSTRACT..... ii
- ACKNOWLEDGEMENTS iii
- LIST OF TABLES..... iv
- CHAPTER ONE: INTRODUCTION..... 1
    - Background..... 1
    - Project Milestones..... 2
    - Significance.....3
    - Purpose 4
- CHAPTER TWO: TECHNOLOGIES USED..... 5
    - Google Colab..... 5
    - Python..... 5
    - Sklearn 5
    - NumPy.....5
    - TensorFlow.....6
    - OpenCV.....6
    - Keras..... 6
    - Pafy..... 6
    - Moviepy.....6
- Literature Survey..... 7
- CHAPTER THREE: METHODOLOGY..... 8
    - Prerequisite..... 8
    - Dataset Description..... 9
    - Data Preprocessing..... 10
    - Model Creation..... 10
    - Model Training..... 11
    - Model layer Diagram.....12
- CHAPTER FOUR: RESULTS 13
    - Dataset Accuracy..... 13
    - Model Training graphs..... 15
- Conclusion 16
- APPENDIX A.....17
- REFERENCES..... 19

Introduction

Background

Recognition of human behavior in real-world environments has numerous practical applications, notably in intelligent video surveillance and shopping behavior analysis. Video surveillance, crucial for security, spans various settings such as indoor and outdoor spaces. It has become an integral aspect of safety protocols in many spheres of life.

The manual oversight of events captured by Closed Circuit Television (CCTV) cameras is increasingly impractical. Even retrospective analysis of recorded footage demands significant time and effort. The identification of abnormal activities in video streams represents a burgeoning field within automated surveillance systems. Automated human behavior detection in video surveillance systems offers a method of intelligently identifying potentially suspicious activities. Several efficient algorithms have been developed for this purpose, catering to diverse public spaces like airports, railway stations, banks, offices, and examination halls.

The convergence of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) has propelled advancements in video surveillance. AI empowers computers to simulate human thought processes, while ML emphasizes learning from training data to make predictions on unseen data. With the proliferation of Graphics Processing Unit (GPU) processors and extensive datasets, DL methodologies have gained prominence. Deep Neural Networks (DNNs) stand out as robust architectures for tackling complex learning tasks. They autonomously extract features and generate high-level representations of image data, streamlining the feature extraction process. Convolutional Neural Networks (CNNs) excel in learning visual patterns directly from image pixels, while Long Short-Term Memory (LSTM) models are adept at capturing long-term dependencies in video streams, enhancing temporal analysis.

The proposed system leverages footage from CCTV cameras to monitor human behavior within a campus environment, issuing alerts discreetly in response to suspicious events. Key components of this intelligent video monitoring framework include event detection and human behavior recognition. The comprehensive training process for a surveillance system can be distilled into three main phases: data preparation, model training, and inference.

Project Milestones

Research and Literature:

The initial phase of the project involves undertaking research and conducting a thorough literature review to acquire a profound comprehension of the current state of detecting suspicious human activity. This includes an exploration of various deep learning-based strategies that have been proposed in the existing body of knowledge.

Data Acquisition:

The next crucial step involves the compilation of a comprehensive data set comprising labeled videos, intended for training, and testing the suggested system. Assemble a set of labeled videos to be utilized for both training and testing the proposed system. The dataset should encompass a diverse array of abnormal behaviors, as well as typical activities, to facilitate comparative analysis.

Image Pre-Processing:

It is of paramount importance to undergo meticulous pre-processing of the video clips before integrating them into the deep learning framework. This critical step encompasses a series of tasks, including the enlargement of frames, selective cropping of pertinent areas, and the normalization of pixel values. These measures collectively contribute to optimizing the input data for enhanced interpretability and efficacy within the deep learning model.

Model Design:

During the model design phase, a sophisticated deep learning architecture is crafted, likely incorporating both Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). RNNs are adept at temporal analysis, discerning patterns over time, while CNNs excel in extracting robust features, enhancing the model's ability to capture intricate details within the data. This synergistic approach ensures a comprehensive model design tailored for effective identification of suspicious human behavior.

Model Training & Testing:

The collected dataset plays a crucial role in training and testing the model, assessing its performance through metrics like recall, accuracy, and precision. This approach enables iterative refinement and fine-tuning of the model, enhancing its capacity to detect and respond to suspicious human behavior. By systematically adjusting parameters based on performance metrics, the model evolves, ensuring an optimized and effective system for the identification and mitigation of potential security threats in public spaces.

Significance

The identification of suspicious human behavior presents a transformative opportunity to significantly enhance outcomes. The primary goal is to actively contribute to developing a sophisticated and precise system for detecting abnormal activities in surveillance video footage, with applications in public safety and homeland security. This envisioned system aims to seamlessly integrate into real-world settings, promising a tangible reduction in criminal and terrorist activities.

To achieve this goal, advanced behavioral analysis algorithms will discern intricate patterns and swiftly identify deviations, indicating potential threats. Integrating multi-modal data, including video feeds, audio, and sensor data, is pivotal for a comprehensive understanding of the surveillance environment. Emphasizing anomaly detection, distinguishing between benign and hazardous anomalies, refines the system's effectiveness.

Real-time processing capabilities are crucial, ensuring prompt identification and response to anomalies, minimizing vulnerability windows. The system's scalability and adaptability are paramount, accommodating diverse environments and evolving threat landscapes. Privacy concerns are addressed through measures like anonymization and encryption, ensuring individual rights are safeguarded. This envisioned system aspires to redefine surveillance paradigms for a safer and more secure future.

Purpose

This project aims to significantly enhance anomaly detection accuracy, focusing on identifying suspicious human behavior to bolster security in public spaces. The tailored system provides real-time crowd sentiment information, issuing hazardous signals for prompt response to potential threats. Its adaptability and proactive monitoring contribute to a comprehensive solution, elevating security standards across diverse public settings. Here's an explanation of each term in the context of the project:

Abuse: This refers to instances of verbal, physical, or emotional mistreatment that may pose a threat to public safety. Detection of aggressive or harmful behavior is crucial for preemptive intervention.

Arrest: This indicates the apprehension of individuals engaging in unlawful activities. The project may involve identifying behaviors leading to potential arrests, contributing to law enforcement efforts.

Fighting: The project aims to identify and address physical altercations or "fighting" scenarios. Detecting early signs of confrontations contributes to the prevention of violence in public spaces.

Walking: Identifying "walking" entails monitoring and promptly responding to incidents involving pedestrians. This project component enhances public safety by swiftly addressing potential risks, reducing accidents, and creating safer walking environments.

Running: Detecting "running" involves monitoring and promptly responding to incidents related to runners. This project component contributes to public safety by swiftly identifying potential risks, mitigating accidents, and ensuring safer running environments.

In conclusion, the project strives to pioneer an advanced surveillance system, adept at swiftly identifying and responding to diverse suspicious activities, bolstering public safety and security. This initiative not only aims to elevate real-time anomaly identification but also sets new benchmarks in technology integration, contributing significantly to the ongoing evolution of surveillance systems for safer, more resilient urban environments.

Technologies Used

Google Colab

Google Colab stands as a cloud-based powerhouse for training machine learning models, offering an ideal environment for tasks spanning machine learning, coding, and analysis. Particularly noteworthy is its integration of Graphical Processing Units (GPUs), a critical feature for expedited model training, especially in the context of larger neural networks. Leveraging Google Colab, the project focuses on training deep learning models specialized in classifying suspicious activities. Furthermore, the platform serves as a version control tool, aiding in the meticulous tracking of different model versions and iterations, ensuring a streamlined development process.

Python

Python, a dynamically typed and garbage-collected programming language, plays a pivotal role in the project's ecosystem. Renowned for its versatility, Python finds application in diverse domains, encompassing software development, website creation, automation, computation, and data analytics. Python frameworks such as Flask and Django contribute to the seamless development of applications with varied functionalities, adding to the project's adaptability and efficiency.

Sklearn

Sklearn, integrated for its compatibility with scientific and numerical libraries, enriches the project with an array of clustering, classification, and regression methods, facilitating robust model development. TensorFlow, another integral component, is a comprehensive machine learning platform that supports distribution and multi-GPU processing. It provides essential features such as automatic differentiation, model development, training, assessment, and multidimensional numerical calculations.

NumPy

NumPy is a powerful numerical computing library in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It facilitates efficient data manipulation and mathematical operations, crucial for tasks in data science, machine learning, and scientific research. NumPy's array-oriented computing paradigm enhances code readability and execution speed.

TensorFlow

TensorFlow serves as a fundamental technology for developing and training deep learning models. Its capabilities in distributed processing and multi-GPU support are crucial for accelerating the training of complex models designed to classify suspicious activities. This framework plays a pivotal role in achieving the project's objective of enhancing accuracy in detecting anomalous activities and contributes to the overall success of the machine learning-based surveillance system.

OpenCV

OpenCV (Open-Source Computer Vision Library) is a versatile open-source computer vision and machine learning software library. Developed in C++ and Python, it provides a rich set of tools for image and video analysis, including algorithms for object detection, facial recognition, and image processing. Its extensive functionality, ease of use, and active community support make it a preferred choice for researchers and developers working on various computer vision applications.

Keras

Keras, a modular neural network library, is instrumental in the project for its seamless integration with TensorFlow. Prioritizing quick experimentation, Keras simplifies the development of intricate neural network architectures. Its high-level abstraction allows for swift model prototyping, making it an ideal choice for the project's goal of classifying suspicious activities. Leveraging Keras on top of TensorFlow enhances efficiency, enabling rapid iteration.

Pafy

When downloading YouTube videos, Pafy collects metadata including count, rate, and duration. Pafy depend on YouTube-dl, so installing it prior is advised for more reliable use.

Moviepy

Moviepy is a module that makes it easy to edit films by allowing you to do things like trim and join, concatenate, insert, compose, animate, add images and objects, and apply video-audio effects like colour correction and noise reduction before exporting the finished product in a variety of codecs and formats. It has a user-friendly interface for processing audio-video files and is built using Pygame, a Python module and dependencies.

Literature Survey

The related works suggest different approaches for detecting human behaviors from video. The objective of the work was to detect any abnormal or suspicious events in video surveillance.

Advance Motion Detection (AMD) algorithm was used to detect an unauthorized entry in a restricted area. In the first phase, the object was detected using background subtraction and from frame sequences the object is extracted. The second phase was detection of suspicious activity. An advantage of the system was the algorithm works on real time video processing and its computational complexity was low. But the system was limited in terms of storage service, and it can also be implemented with hi-tech mode of capturing of videos in the surveillance areas.

A semantic based approach was proposed in [References- [2]. The captured video data was processed, and the foreground objects were identified using background subtraction. After subtraction, the objects are classified as living or non-living using Haar like algorithm. Objects tracking was done using a Real-Time blob matching algorithm. Fire detection was also detected in this paper.

The unusual events in video footage could be detected by tracking people. Human beings are detected from the video using background subtraction method. The features are extracted using CNN and which was fed to a DDBN (Discriminative Deep Belief Network). Labeled videos of some suspicious events are also fed to the DDBN and their features are also extracted. Then a comparison of features extracted using CNN and features extracted from the labeled sample video of classified suspicious actions was done using a DDBN and various suspicious activities are detected from the given video.

A real time violence detection system using deep learning was developed to prevent the violent behavior of crowds or players in sports. In a spark environment, frames were extracted from real-time videos. If the system detects any violence in football, then alert the security people. To prevent the violence in advance, the system detects the video actions in real time and alerts the security forces. VID dataset was used and achieved an accuracy of 94.5% for detecting violence in football stadiums.

Methodology/ Planning of work

Video capture

Installation of CCTV camera and monitoring the footage is the initial step in video surveillance Video capture Installation of CCTV camera and monitoring the footage is the initial step in video surveillance system. Various kinds of videos are captured from different cameras, covering the whole area of surveillance.

Video Pre-processing

As part of pre-processing, 30 frames are extracted from each of the captured videos, frames are separated on equal time intervals. 30 extracted frames are resized to 64 x 64 and read in a NumPy array of dimension (64 x 64 x 3) ~ (Image Width x Image Height x RGB) using OpenCV Library in Python.

Each Value in the frame is then Normalized by dividing it with 255.

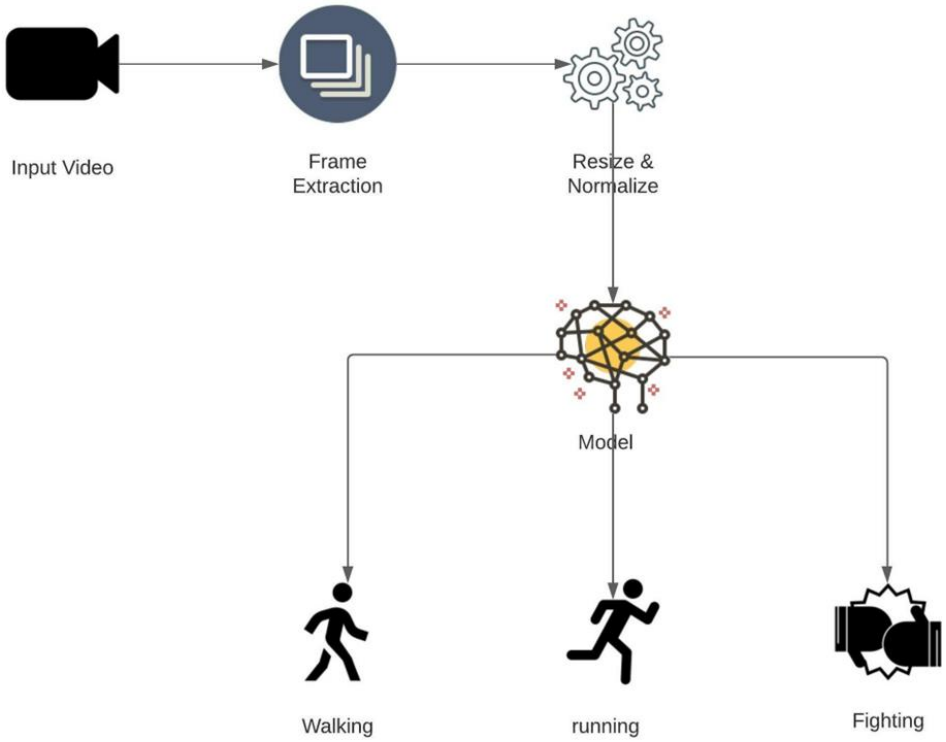
All the 30 Normalized frames from each video are stored as sequence in NumPy array with dimension 30 x 64 x 64 x 3.

The Project's Flow

The NumPy array is given as input to the Model and the Model predicts the class of the given Video.

In our proposed system, for detecting anomalous behavior, LRCN (Long-term Recurrent Convolutional Network) has been used. For effective classification of anomalous activities, it is essential to recognize the temporal data in the video. Recently, CNN is mostly used for extracting key features from each frame of the video. For classifying the given input successful, it is necessary that the features get extracted from CNN, therefore CNN should be capable of knowing and extracting the needed features from the frame of videos.

A sequence of 30 frames of the video is extracted and passed to the LRCN Model.



Dataset Description

KTH dataset for detection of Running and Walking. KTH Dataset https://www.csc.kth.se/cvap/actions/

And Kaggle dataset for fight detection.

Kaggle Dataset - https://www.kaggle.com/naveenk903/movies-fight-detection-dataset

The KTH dataset is a standard dataset which has a collection of sequences representing 6 actions and each action class has got 100 sequences. Each sequence has got almost 600 frames, and the video is shot at 25 fps.

Kaggle Dataset consists of over 100 videos taken from movies and YouTube videos can be used for training suspicious behavior (fighting).

Data Pre-processing

a) Read Video and Label: Using OpenCV Library the videos are read from their respective Class folder and their Class label is stored inside a NumPy array.

b) Splitting into frames to make one sequence: Each Video is read using OpenCV Library, only 30 frames at equal time intervals are read to form a sequence of 30 frames.

c) Resizing: Image resizing is necessary when we need to increase or decrease the total number of pixels. So, we resized all the frames to width: 64px and height: 64px to maintain the uniformity of the input images to the architecture.

d) Normalization: Normalization will help the learning algorithm to learn faster and capture necessary features from the images. So, we normalized the resized frame by dividing it with 255 so that each pixel value lies between 0 and 1.

e) Store in NumPy Arrays: The sequence of 30 resized and Normalized frames are stored in a NumPy array to give as Input to the Model.

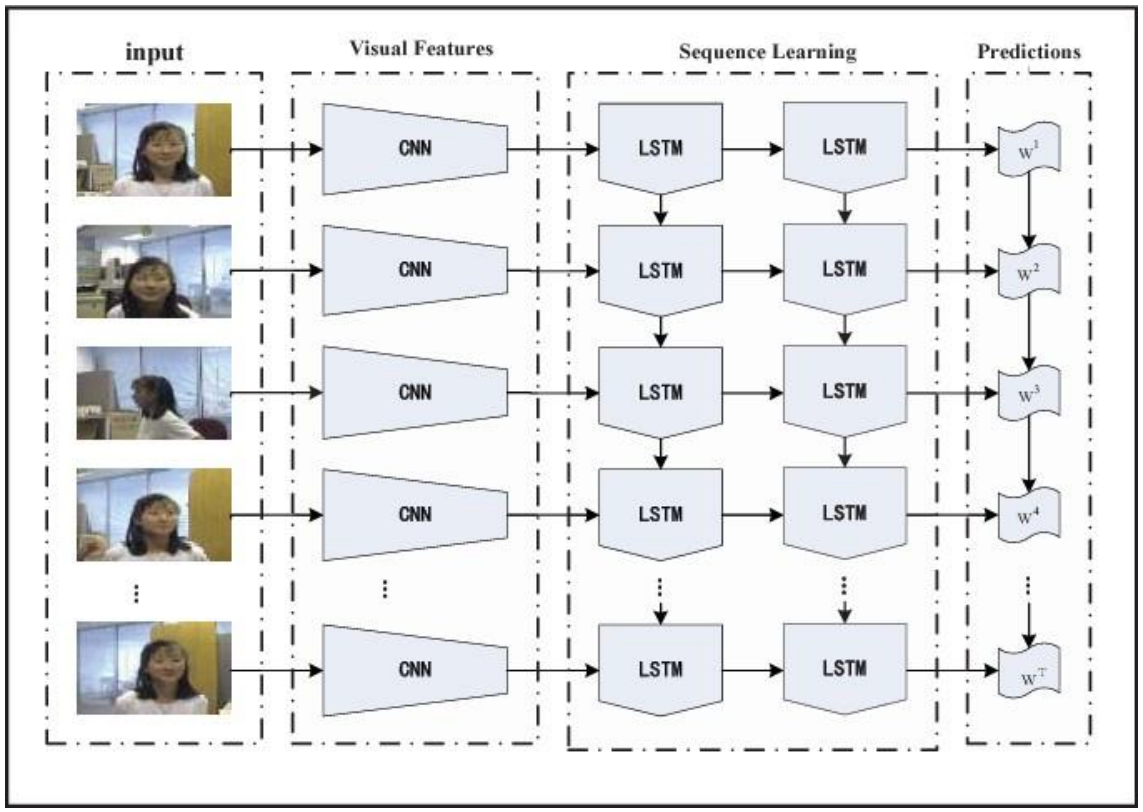
Train Test Split Data

75% of the data is used for Training.

25% of the data is used for Testing.

Model Creation

A deep learning network, LRCN, is used in our proposed system for suspicious activity detection from video surveillance.



In 2016 a group of authors suggested an end-to-end trainable class of architectures for visual recognition and description. The main idea behind LRCN is to use a combination of CNNs to learn visual features from video frames and LSTMs to transform a sequence of image embeddings into a class label, sentence, probabilities, or whatever you need. Thus, raw visual input is processed with a CNN, whose outputs are fed into a stack of recurrent sequence models.

LSTM networks are well-suited to classifying, processing, and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

Model Training

The model is trained to predict over 3 classes – walking, running and fight

The training set is given to the model for training, with the following hyper parameters:

•epochs = 70

•batch_size = 4

•validation_split = 0.25

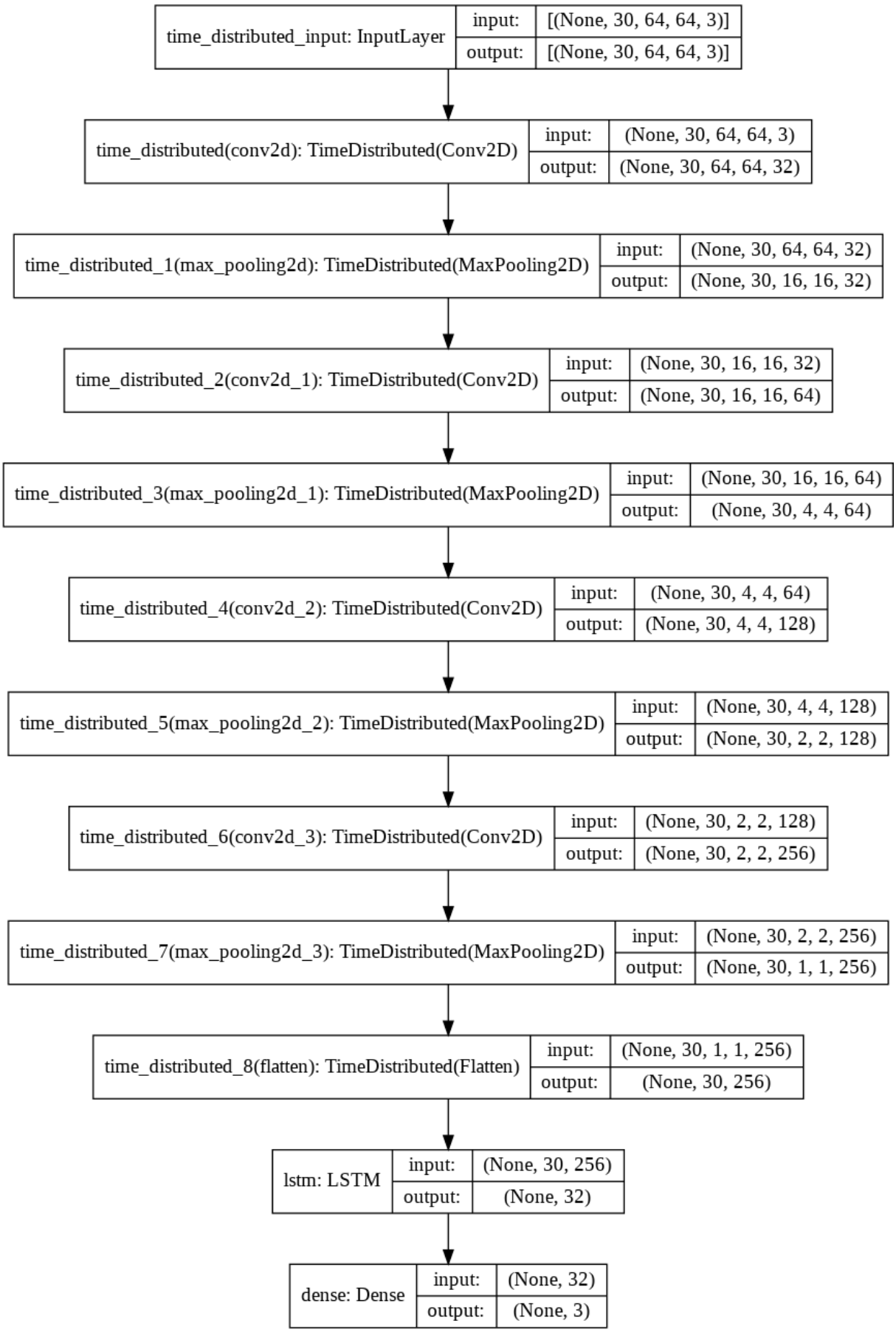
nice system. Various kinds of videos are captured from different cameras, coveringthe whole area of surveillance.

Model Training

```
In [17]: # Create an Instance of Early Stopping Callback.
         early_stopping_callback = EarlyStopping(monitor = 'accuracy', patience = 10, mode = 'max', restore_best_weights = Tru

         # Compile the model and specify loss function, optimizer and metrics to the model.
         model.compile(loss = 'categorical_crossentropy', optimizer = 'Adam', metrics = ["accuracy"])

         # Start training the model.
         model_training_history = model.fit(x = features_train, y = labels_train, epochs = 70, batch_size = 4 , shuffle = True
```

```
curacy: 0.8772
Epoch 45/70
42/42 [==============================] - 1s 31ms/step - loss: 0.0085 - accuracy: 1.0000 - val_loss: 0.4053 - val_ac
curacy: 0.8947
Epoch 46/70
42/42 [==============================] - 1s 32ms/step - loss: 0.0061 - accuracy: 1.0000 - val_loss: 0.4113 - val_ac
curacy: 0.8772
Epoch 47/70
42/42 [==============================] - 1s 32ms/step - loss: 0.0050 - accuracy: 1.0000 - val_loss: 0.4235 - val_ac
curacy: 0.8772
Epoch 48/70
42/42 [==============================] - 1s 32ms/step - loss: 0.0043 - accuracy: 1.0000 - val_loss: 0.4252 - val_ac
curacy: 0.8772
Epoch 49/70
42/42 [==============================] - 1s 31ms/step - loss: 0.0040 - accuracy: 1.0000 - val_loss: 0.4044 - val_ac
curacy: 0.8947
Epoch 50/70
42/42 [==============================] - 1s 32ms/step - loss: 0.0037 - accuracy: 1.0000 - val_loss: 0.4138 - val_ac
curacy: 0.8947
```

MODEL LAYER DIAGRAMS



RESULT

Our proposed model aims to detect the anomalous behavior happening in the video and the system is achieving the accuracy of 82% on our created data set.

In our previous model, we were using the VGG-16 model which consisted of 16 layersand so it was time consuming and as a result it could not be used in REAL-TIME detection. But with LRCN model, the number of layers decreased to 11 and it became less time consuming and can work in REAL-TIME detection as well. We resized our frames from 224px to 64px to save memory space and added more videos in our dataset to increase accuracy. The dataset of the proposed model includes videos of anomalous behavior which is Fighting as well as it also contains videos of normal behavior which is walking and running. Following are the images of the result of the proposed model.

Accuracy on Test Dataset

```
[ ] # Calculate Accuracy On Test Dataset
    acc = 0
    for i in range(len(features_test)):
      predicted_label = np.argmax(model.predict(np.expand_dims(features_test[i],axis =0))[0])
      actual_label = np.argmax(labels_test[i])
      if predicted_label == actual_label:
          acc += 1
    acc = (acc * 100)/len(labels_test)
    print("Accuracy =",acc)
```

```
Accuracy = 82.66666666666667
```

```
[ ] predict_single_action("Predict/fight.avi",SEQUENCE_LENGTH)
```

```
Action Predicted: fight
Confidence: 0.9965279698371887
```

```
[ ] predict_single_action("Predict/running.avi",SEQUENCE_LENGTH)
```

```
Action Predicted: running
Confidence: 0.9882073998451233
```

```
[ ] predict_single_action("Predict/walking.avi",SEQUENCE_LENGTH)
```

```
Action Predicted: walking
Confidence: 0.9890599250793457
```

```
VideoFileClip("Human-Activity-Prediction.avi", audio=False).ipython_display()
```

```
t:   4%|▌          | 33/897 [00:00<00:02, 322.82it/s, now=None]Moviepy - Building video __temp__.mp4.
Moviepy - Writing video __temp__.mp4

                                                                Moviepy - Done !
Moviepy - video ready __temp__.mp4
```


```
VideoFileClip("Human-Activity-Prediction.avi", audio=False).ipython_display()
```

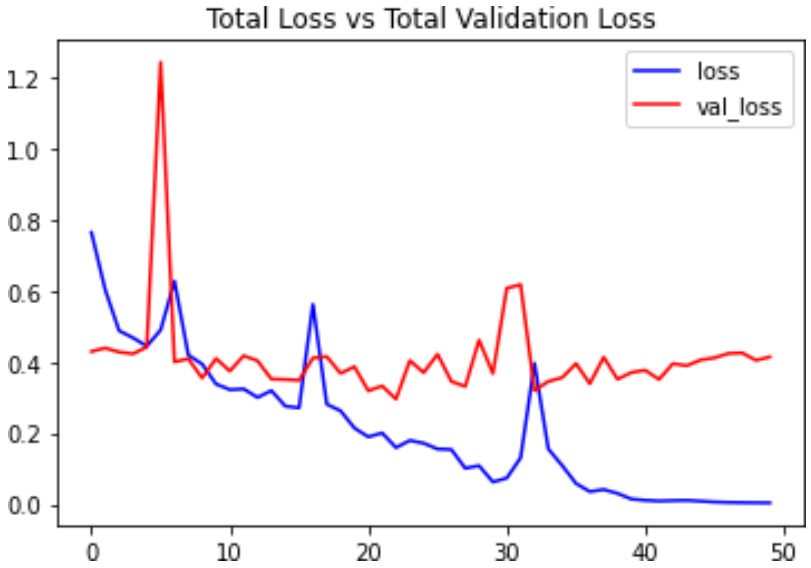
```
t:   4%|▌          | 33/897 [00:00<00:02, 322.82it/s, now=None]Moviepy - Building video __temp__.mp4.
Moviepy - Writing video __temp__.mp4

                                                                Moviepy - Done !
Moviepy - video ready __temp__.mp4
```


```
VideoFileClip("Human-Activity-Prediction.avi", audio=False).ipython_display()
```

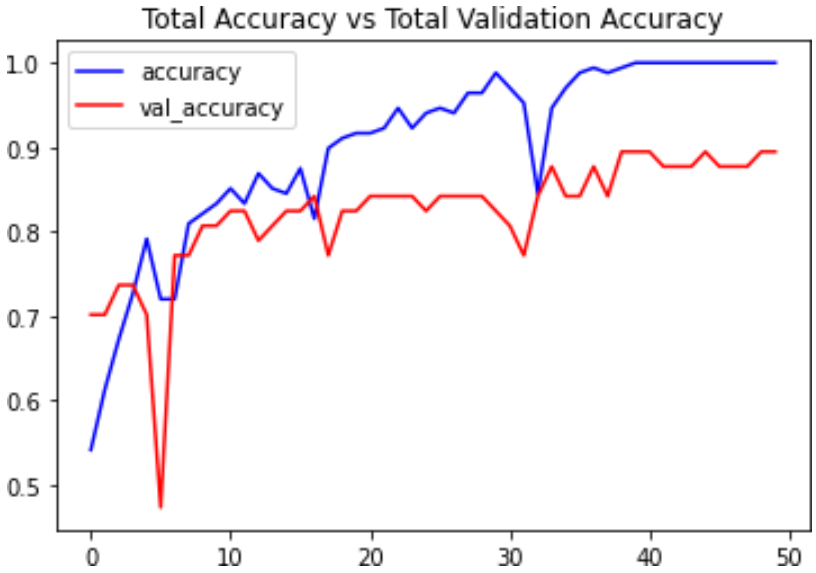
```
t:   4%|▌          | 33/897 [00:00<00:02, 322.82it/s, now=None]Moviepy - Building video __temp__.mp4.
Moviepy - Writing video __temp__.mp4

                                                                Moviepy - Done !
Moviepy - video ready __temp__.mp4
```

Model Training graphs

Loss vs Validation Loss



Accuracy vs Validation Accuracy



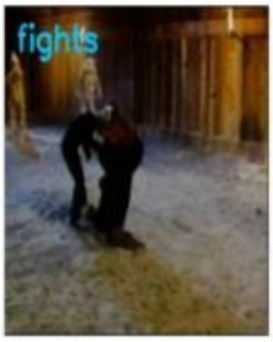
Conclusion

In the realm of automated detection of suspicious human activities, a multidisciplinary approach merges computer vision, machine learning, and deep learning techniques. Through a Long-term Recurrent Convolutional Network (LRCN) model, recent advancements demonstrate promise in identifying abnormal behaviors, achieving an 86% accuracy rate in discerning activities like fighting, running, and walking from CCTV footage. Yet, to maximize efficacy and predictive accuracy, expanding the dataset is crucial. Integrating diverse data sources, including additional instances of suspicious activities and contextual information, aims to fortify the model's generalization ability. Optimization of model architecture and training algorithms further enhances performance and scalability while mitigating biases. This pursuit represents a dynamic challenge, demanding ongoing innovation and collaboration across disciplines. Through refinement and iteration, researchers aspire to develop automated surveillance systems capable of reliably enhancing public safety and security, thus ushering in a new era of detection technology.

Appendix A

View the Dataset

BGR is the image format used by OpenCV. Therefore, by default, cv2.imread() interprets images in BGR (blue, green, and red) format. However, in Pillow, it is believed that the colours are in RGB (red, green, and blue) sequence. So, to use the OpenCV function as well as the Pillow function, we must convert BGR and RGB. To recognize faces, objects, or even human handwriting, it can process photos and movies. When converting a BGR image to RGB or vice versa, we can utilize the cvtColor() method. The class wise frames on display.



Data Pre-processing

After resizing and normalizing the frames, function will extract from a video.

video path: The location on the disk of the video from which the desired frames should be taken. A list of the video's resized and normalized frames is called a "frame list."

Data Pre-processing

```
def f-extract(path):

list = []
  vreade = c.videocapture(path)
  fcount = int(vreade.get(c.fcount))
  f = max(int(f-count/len), 1)
     for fcounter in range (len):
     vreade.set(c.f, fcounter * f)

     done, f = vreade.read()
     if not done:
         break

     changesize-f = c.changesize(f, (height, width))

     normalize-f = changesize-f / 255

     list.append(normalize-f)

  v-reade.release()

 return list
```

Model Creation

We can build a Sequential model progressively using add () method. Add () can be used to incrementally stack layers, and printing model summaries periodically is helpful.

```
def create-model():

  model = m()

  m.add(timedist(conv(64,(2, 2), padding='same',active ='rel')))
  m.add(timedist(maxpool((2, 2))))

  m.add(timedist(conv(32,(3, 3), padding='same',active ='rel')))
  model.add(timedist(maxpool((2, 2))))

  m.add(timedist(conv(16, (3, 3), padding='same',active ='rel')))
  m.add(timedist(maxpool((4, 4))))

  m.add(timedist(conv(8,(3, 3), padding='same',,active =    'rel'),
  s = (1, h, w, 3)))
  m.add(timedist(maxpool((4, 4))))

  m.add(timedist(f()))

  m.add(model(32))

  m.summary()
  return m
```

References

[1] C. V. Amrutha, C. Jyotsna, J. Amudha (2020) Deep learning Approach for suspicious activity detection from surveillance video, Publisher IEEE Bangalore https: www.ieeexplore.ieee.org/document/9074920 (Original work published 2020)

[2] Mark Daoust (2022). Sequential model, Model Creation [python], TensorFlow is a platform that makes it easy to build and deploy ML models. https: www.tensorflow.org/guide/keras/ (Original work published on 2022)

[3] Sik-Ho Tsang (2022) LRCN: Long-term Recurrent Convolution Networks https://sh-tsang.medium.com/brief-review-lrcn (Original work published on 2022)

[4] CVPR (Conference on Computer Vision and Pattern Recognition), ICCV (International Conference on Computer Vision), or ACM Multimedia for research papers on video analysis and anomaly detection. https://pubmed.ncbi.nlm.nih.gov/ (Original work published on 2022)

[5]P.Bhagya Divya, S.Shalini, R.Deepa, Baddeli Sravya Reddy,"Inspection of suspicious human activity in the crowdsourced areas captured in surveillance cameras",International Research Journal of Engineering and Technology (IRJET), December 2017.

[6]Jitendra Musale,Akshata Gavhane, Liyakat Shaikh, Pournima Hagwane, Snehalata Tadge, "Suspicious Movement Detection and Tracking of Human Behavior and Object with Fire Detection using A Closed Circuit TV (CCTV) cameras ", International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 5 Issue XII December 2017.

[7]Elizabeth Scaria, Aby Abahai T and Elizabeth Isaac, "Suspicious Activity Detection in Surveillance Video using Discriminative Deep Belief Netwok", International Journal of Control Theory and Applications Volume 10, Number 29 -2017.