Suspicious Activity Detection from Surveillance Video

- Made by : Kunal

B.Tech

IIITDM Jabalpur

2019085@iiitdmj.ac.in

INTRODUCTION

Human behavior recognition in the real-world environment finds plenty of applications including intelligent video surveillance, shopping behavior analysis. Video surveillance has vast application areas especially for indoor outdoor and places. Surveillance is an integral part of security. Today security camera becomes part of life for the safety and security purposes.

Today, manual monitoring of all the events on the *CCTV (Closed Circuit Television)* camera is impossible. Even if the event had already happened, searching manually the same event in the recorded video wastes a lot of time. Analyzing abnormal events from video is an emerging topic in the domain of *automated video surveillance systems*.

Human behavior detection in video surveillance system is an automated way of intelligently detecting any suspicious activity. Number of efficient algorithms is available for the automatic detection of human behavior in public areas like airports, railway stations, banks, offices, examination halls etc

Video surveillance is the emerging area in the application of *Artificial Intelligence, Machine Learning and Deep Learning*. Artificial intelligence helps the computer to think like human. In machine learning, important components are learning from the training data and make prediction on future data. Nowadays *GPU (Graphics Processing Unit)* processors and huge datasets are available, so the concept of deep learning is used.

Deep Neural Networks is one of the best architectures used to perform difficult learning tasks. Deep Learning models automatically extract features and builds high level representation of image data. This is more generic because the process of feature extraction is fully automated. From the image pixels, *convolutional neural network (CNN)* can learn visual patterns directly. In the case of video stream, *long short-term memory (LSTM)* models are capable of learning long term dependencies. LSTM network has the ability to remember things.

The proposed system will use footage obtained from CCTV camera for monitoring the human behavior in a campus and gently warn when any suspicious event occurs. The major components in intelligent video monitoring are event detection and human behavior recognition.

The entire process of training a surveillance system can be summarized in to three phases: data preparation, training the model and inference.

MOTIVATION

Importance of the suspicious human activities recognition from video surveillance is to prevent the theft cases, leaving abandoned objects for the explosive attacks by terrorists, vandalism, fighting and personal attacks and fire in the different highly sensitive areas such as banks, hospitals, malls, parking lots, bus and railway stations, airports, refineries, nuclear power plants, schools, university campuses, borders etc.

Video surveillance can be used in *university campuses and other academic institutions* to monitor the activities of students for the safety of assets from theft and vandalism. It will also help to prevent the inappropriate behavior of the students and fighting among the students. It will monitor the perimeter of the university campus, school and academic institutions for the safety of the students and faculties. Video surveillance can be used at the time of examination to monitor the suspicious activity of the students in the examination hall.

At last, Video Surveillance can be used *to maintain rule and order* in the academic institutions at the cost of lesser guards.

LITERATURE SURVEY

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

[1] *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. 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.

[2] 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 into living or non-living using Haar like algorithm. Objects tracking were done using Real-Time blob matching algorithm. Fire detection was also detected in this paper.

[3] The unusual events in video footage could be detected by tracking of 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.

[4] A real time violence detection system using deep learning was developed to prevent the violence behavior of crowd 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 stadium.

***Number inside [ ] are linked to same number of research paper in the references section*

APPROACH

The proposed approach will use footage obtained from CCTV camera for monitoring students' activities in a campus and send message to the corresponding authority when any suspicious event occurs.

The architecture has different phases like Video capture, Video pre-processing, Class Prediction. The system classifies the videos into three classes:

- Students fighting in campus- Suspicious class
- Walking, running- Normal class

A. 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.

B. 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 224 x 224 and read in a numpy array of dimension (224 x 224 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 224 x 224 x 3.

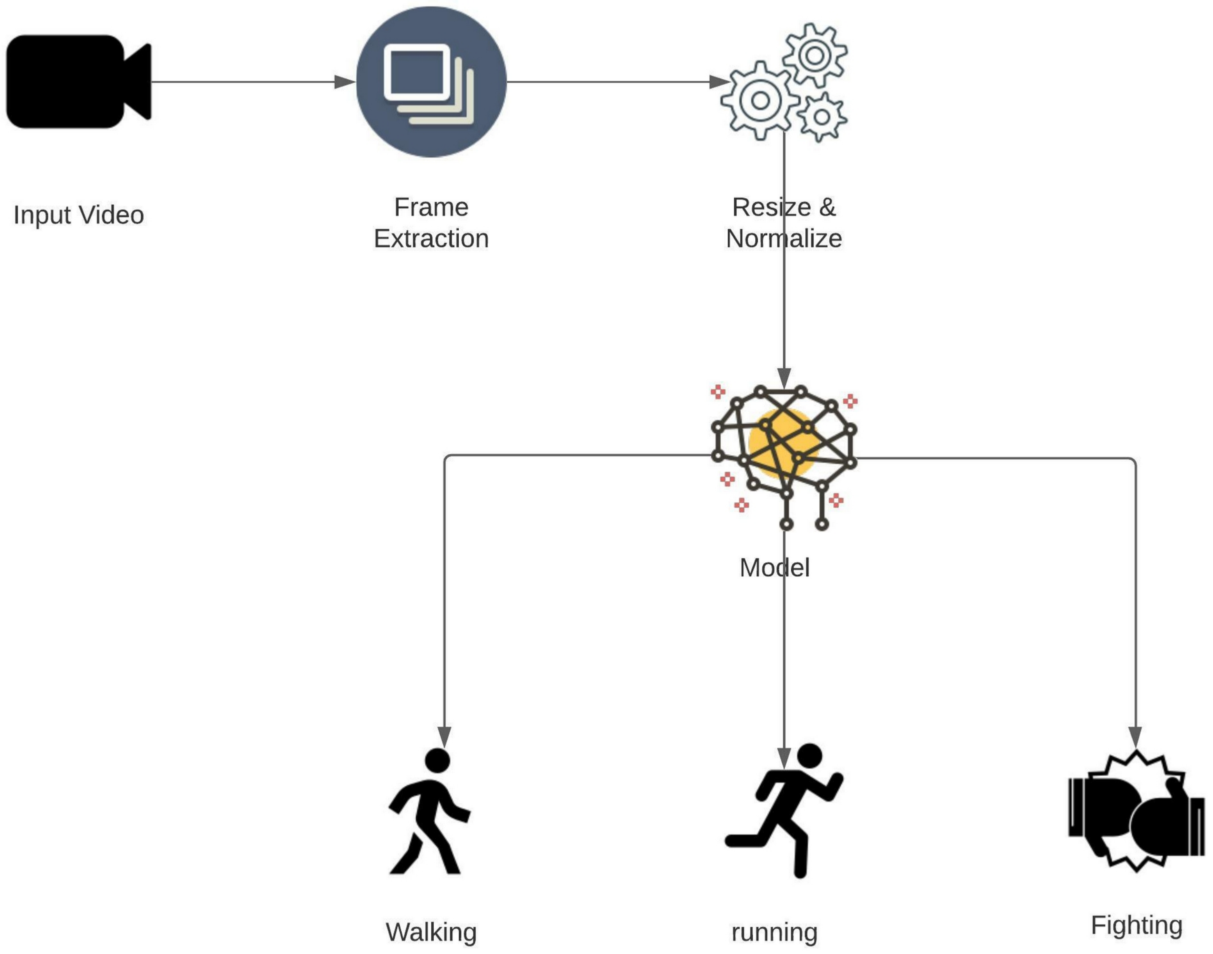
C. Class Prediction

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

PROPOSED SYSTEM AND DESIGN

In our proposed system, for detecting anomalous behavior, VGG-16 CNN (convolution neural network) has been used. For effectively 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.

Features from 30 frames of the video are extracted and passed as a sequence to LSTM Layer for Video's class prediction.



IMPLEMENTATION

1. 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 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. The model is trained on this dataset only with 45 sequences of each class due to hardware Limitations for normal behavior. (running and walking).

Kaggle Dataset, over 100 videos taken from movies and YouTube videos can be used for training suspicious behavior (fighting). Only 45 of the 100 videos are taken due to hardware limitation.

2. 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: 224 px and height: 224px 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.

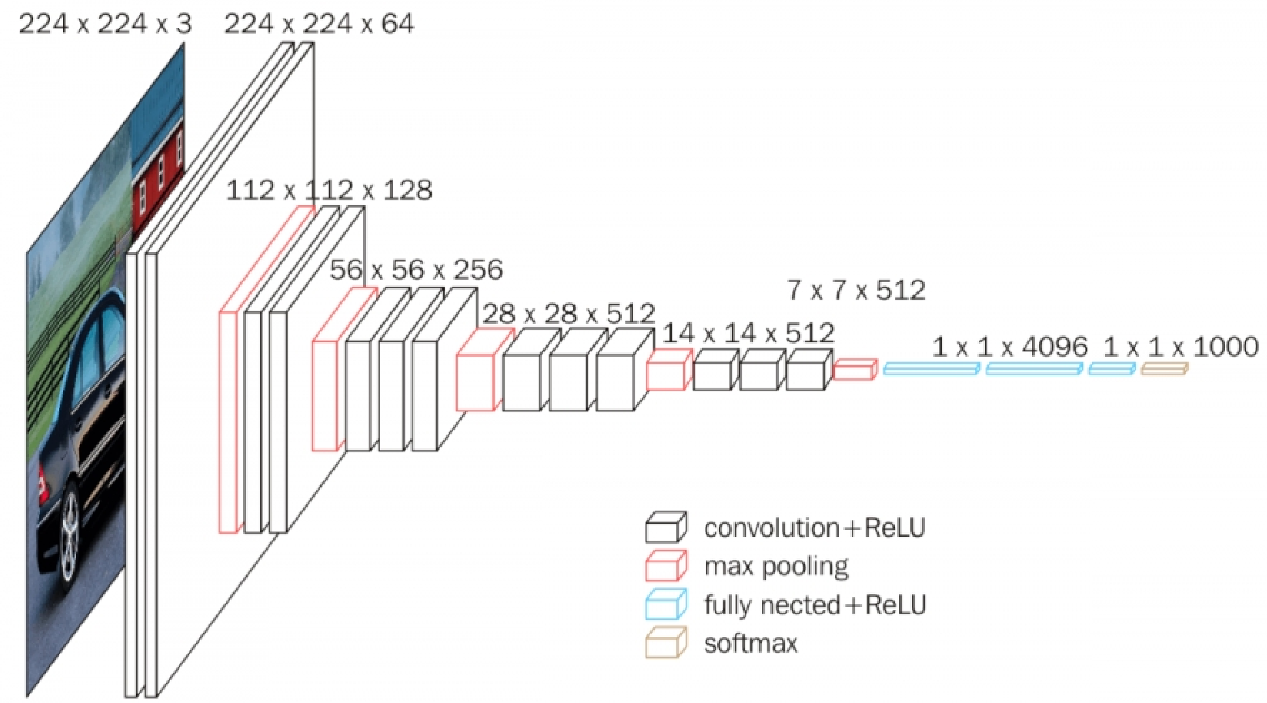
3. Train Test Split Data

75% of the data is used for Training

25% of the data is used for Testing

4. Model Creation

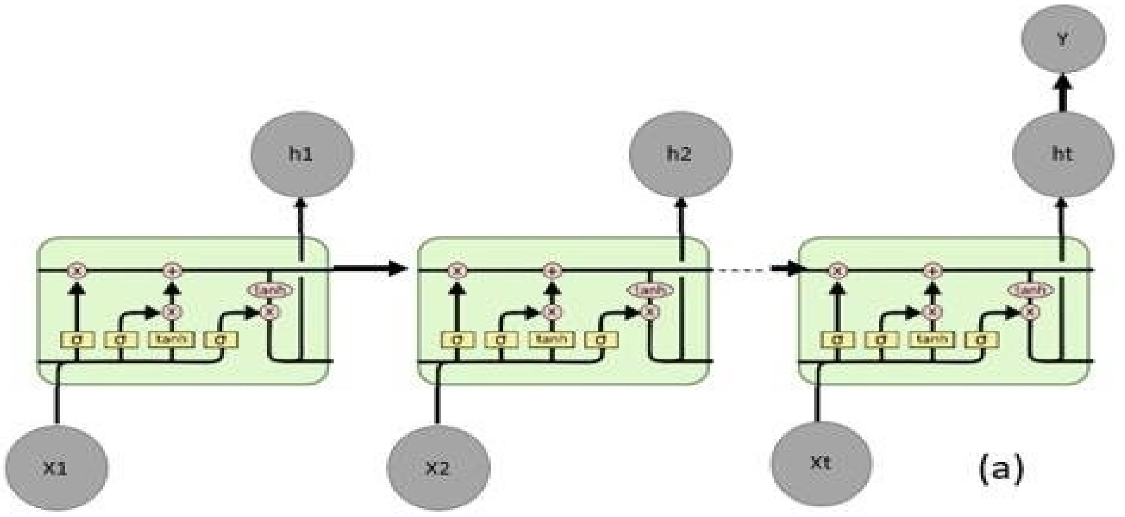
A deep learning network, VGG-16 is using in our proposed system for suspicious activity detection from video surveillance.



VGG-16 Model

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

The Last 4 Layers of the VGG-16 Model are removed and VGG-16 model is set over TimeDistributed Layer, followed by LSTM and two Dense Layers respectively.



LSTM Layer - Multi Input Single Output

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.

5. Model Training

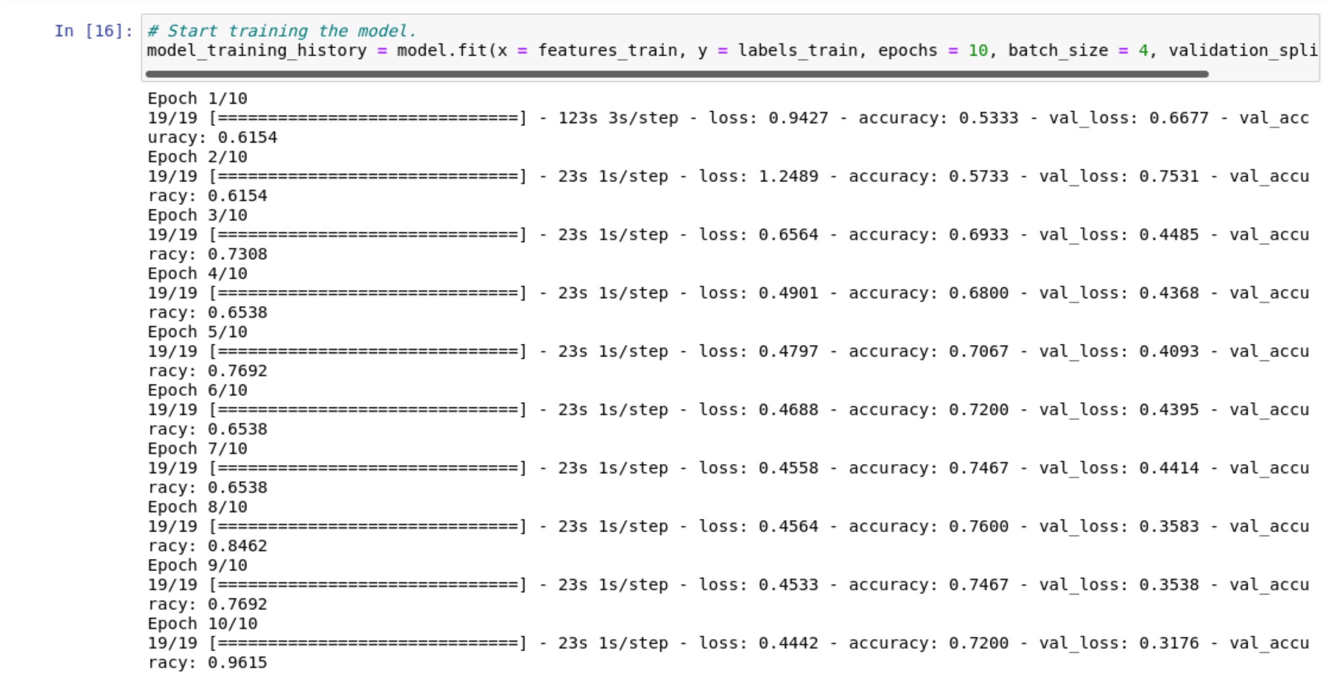
The model is trained to predict over 3 classes – walking, running and fighting

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

a) epochs = 10

b) batch_size = 4

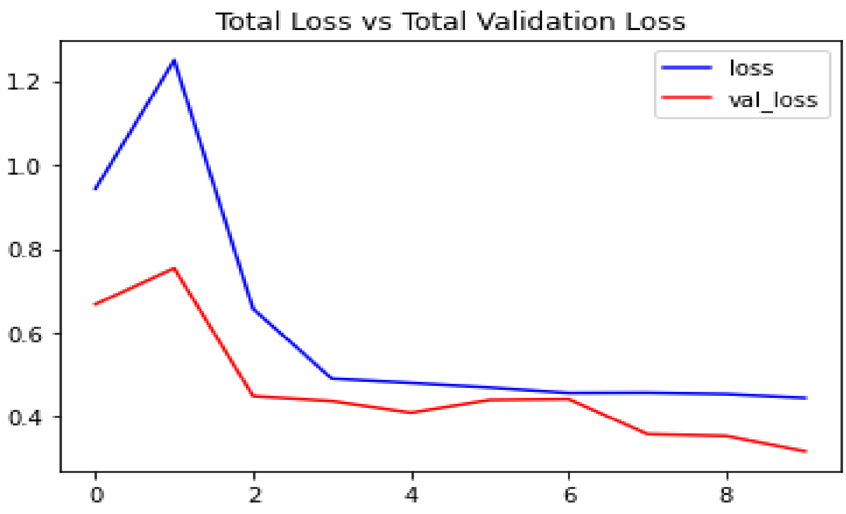
c) validation_split = 0.25



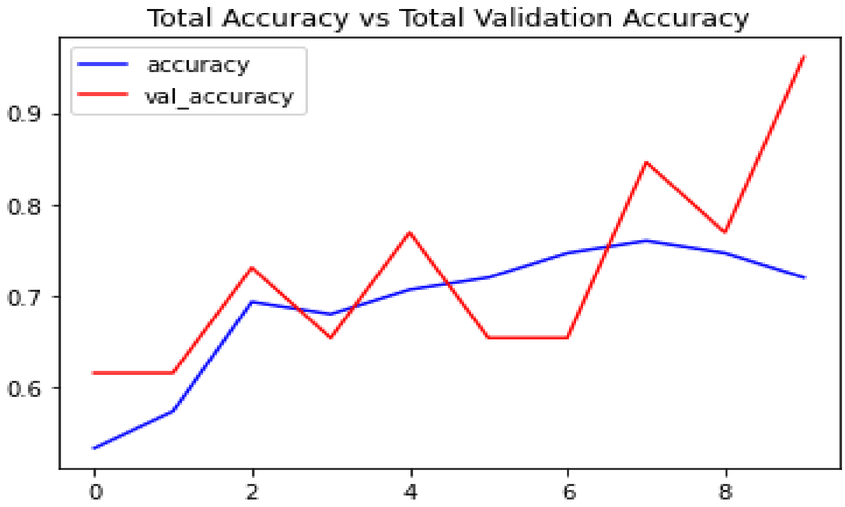
```
In [16]: # Start training the model.
         model_training_history = model.fit(x = features_train, y = labels_train, epochs = 10, batch_size = 4, validation_spli

         Epoch 1/10
         19/19 [==============================] - 123s 3s/step - loss: 0.9427 - accuracy: 0.5333 - val_loss: 0.6677 - val_acc
         uracy: 0.6154
         Epoch 2/10
         19/19 [==============================] - 23s 1s/step - loss: 1.2489 - accuracy: 0.5733 - val_loss: 0.7531 - val_accu
         racy: 0.6154
         Epoch 3/10
         19/19 [==============================] - 23s 1s/step - loss: 0.6564 - accuracy: 0.6933 - val_loss: 0.4485 - val_accu
         racy: 0.7308
         Epoch 4/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4901 - accuracy: 0.6800 - val_loss: 0.4368 - val_accu
         racy: 0.6538
         Epoch 5/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4797 - accuracy: 0.7067 - val_loss: 0.4093 - val_accu
         racy: 0.7692
         Epoch 6/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4688 - accuracy: 0.7200 - val_loss: 0.4395 - val_accu
         racy: 0.6538
         Epoch 7/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4558 - accuracy: 0.7467 - val_loss: 0.4414 - val_accu
         racy: 0.6538
         Epoch 8/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4564 - accuracy: 0.7600 - val_loss: 0.3583 - val_accu
         racy: 0.8462
         Epoch 9/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4533 - accuracy: 0.7467 - val_loss: 0.3538 - val_accu
         racy: 0.7692
         Epoch 10/10
         19/19 [==============================] - 23s 1s/step - loss: 0.4442 - accuracy: 0.7200 - val_loss: 0.3176 - val_accu
         racy: 0.9615
```

Model Training graphs :

Loss vs Validation Loss

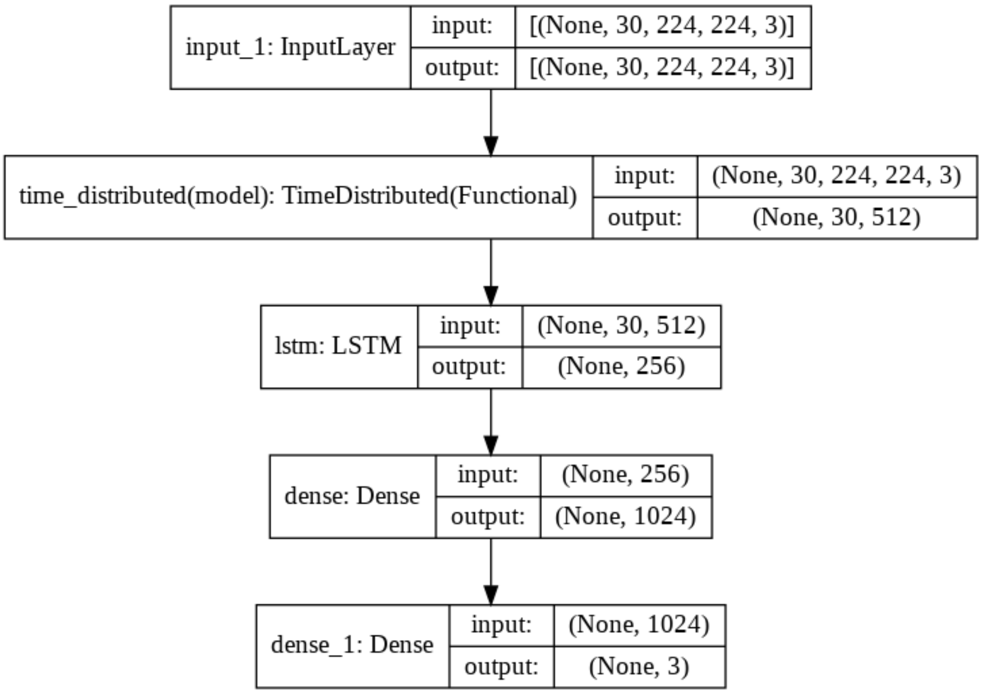


Accuracy vs Validation Accuracy

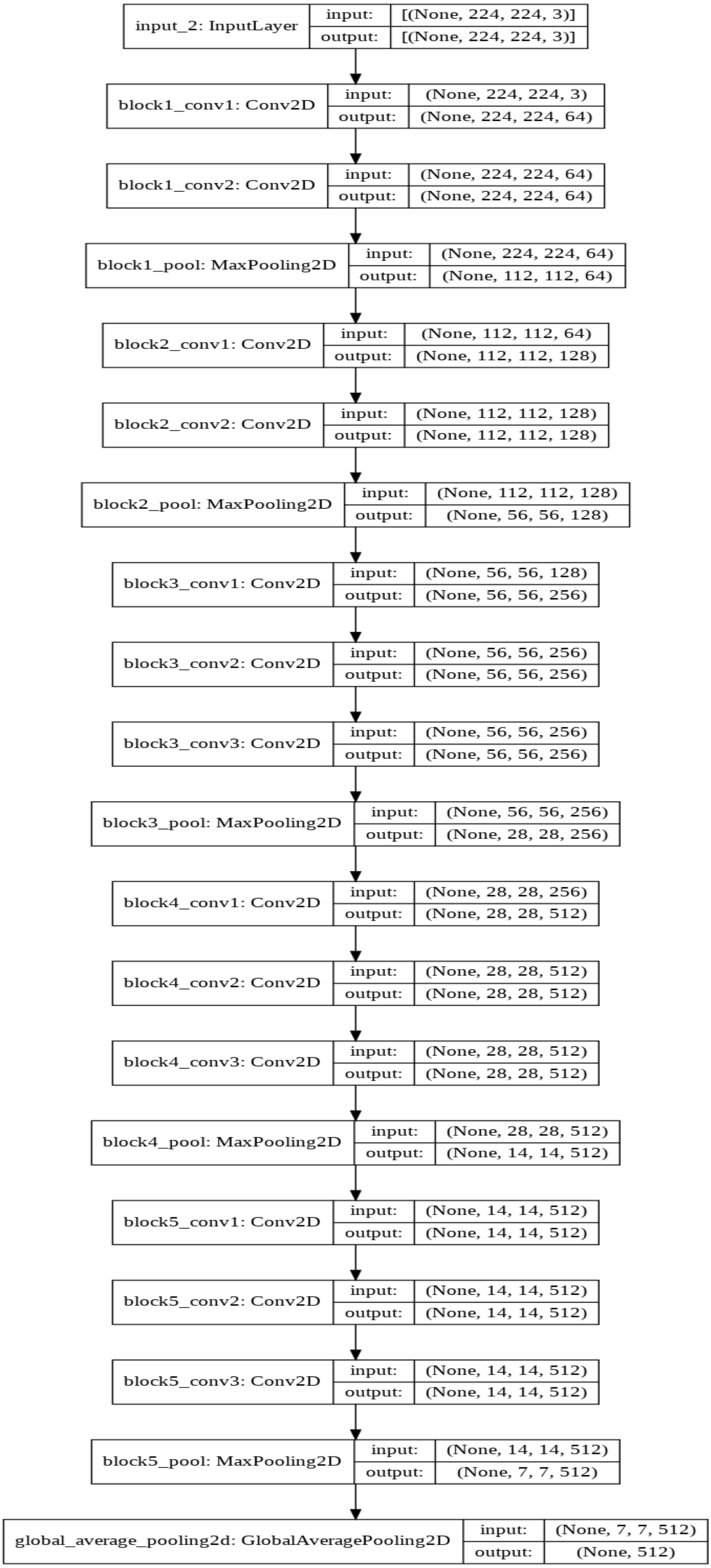


MODEL LAYER DIAGRAMS

1. Final Model

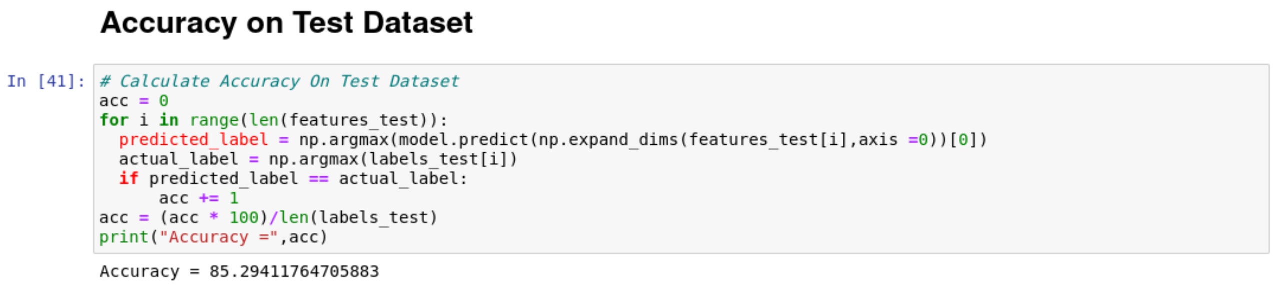


2. Layer 2 of Final Model Expansion (VGG-16 Model)


input_2: InputLayer | input: | [(None, 224, 224, 3)] | output: | [(None, 224, 224, 3)]
block1_conv1: Conv2D | input: | (None, 224, 224, 3) | output: | (None, 224, 224, 64)
block1_conv2: Conv2D | input: | (None, 224, 224, 64) | output: | (None, 224, 224, 64)
block1_pool: MaxPooling2D | input: | (None, 224, 224, 64) | output: | (None, 112, 112, 64)
block2_conv1: Conv2D | input: | (None, 112, 112, 64) | output: | (None, 112, 112, 128)
block2_conv2: Conv2D | input: | (None, 112, 112, 128) | output: | (None, 112, 112, 128)
block2_pool: MaxPooling2D | input: | (None, 112, 112, 128) | output: | (None, 56, 56, 128)
block3_conv1: Conv2D | input: | (None, 56, 56, 128) | output: | (None, 56, 56, 256)
block3_conv2: Conv2D | input: | (None, 56, 56, 256) | output: | (None, 56, 56, 256)
block3_conv3: Conv2D | input: | (None, 56, 56, 256) | output: | (None, 56, 56, 256)
block3_pool: MaxPooling2D | input: | (None, 56, 56, 256) | output: | (None, 28, 28, 256)
block4_conv1: Conv2D | input: | (None, 28, 28, 256) | output: | (None, 28, 28, 512)
block4_conv2: Conv2D | input: | (None, 28, 28, 512) | output: | (None, 28, 28, 512)
block4_conv3: Conv2D | input: | (None, 28, 28, 512) | output: | (None, 28, 28, 512)
block4_pool: MaxPooling2D | input: | (None, 28, 28, 512) | output: | (None, 14, 14, 512)
block5_conv1: Conv2D | input: | (None, 14, 14, 512) | output: | (None, 14, 14, 512)
block5_conv2: Conv2D | input: | (None, 14, 14, 512) | output: | (None, 14, 14, 512)
block5_conv3: Conv2D | input: | (None, 14, 14, 512) | output: | (None, 14, 14, 512)
block5_pool: MaxPooling2D | input: | (None, 14, 14, 512) | output: | (None, 7, 7, 512)
global_average_pooling2d: GlobalAveragePooling2D | input: | (None, 7, 7, 512) | output: | (None, 512)

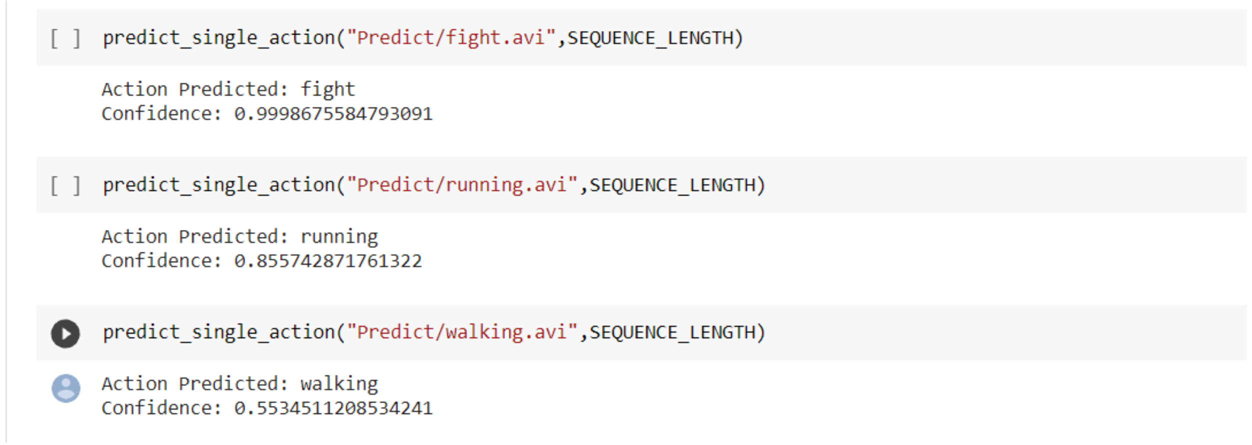
RESULT

Our proposed model aims to detect the anomalous behavior happening in the video and the system is achieving the accuracy of 85% on our created data set. 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 result of the proposed model.

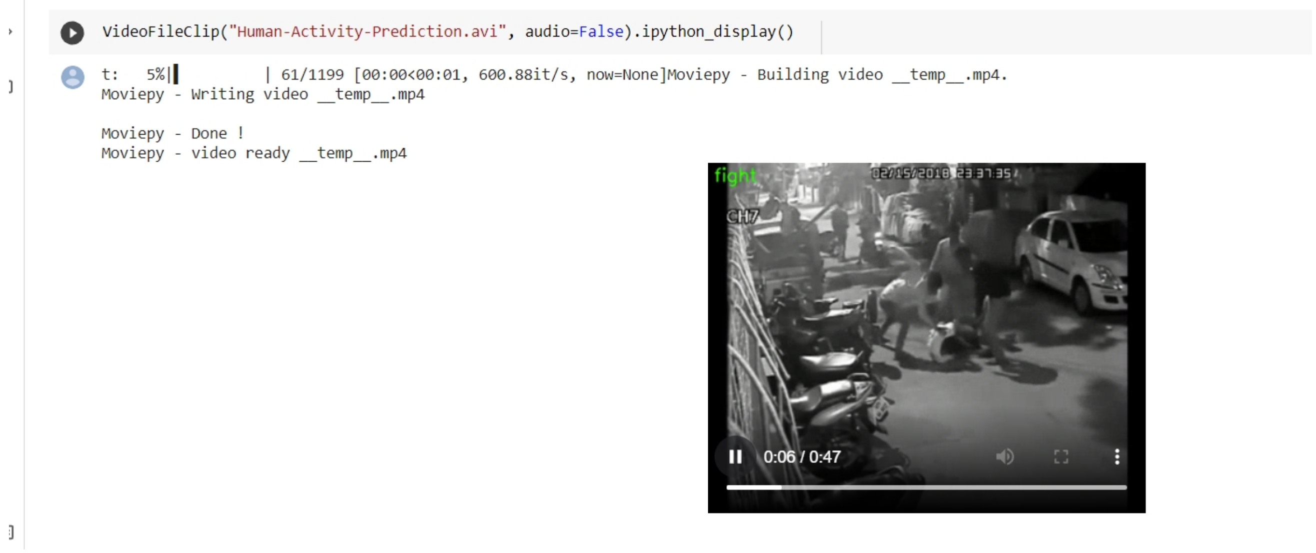


Accuracy on Test Dataset

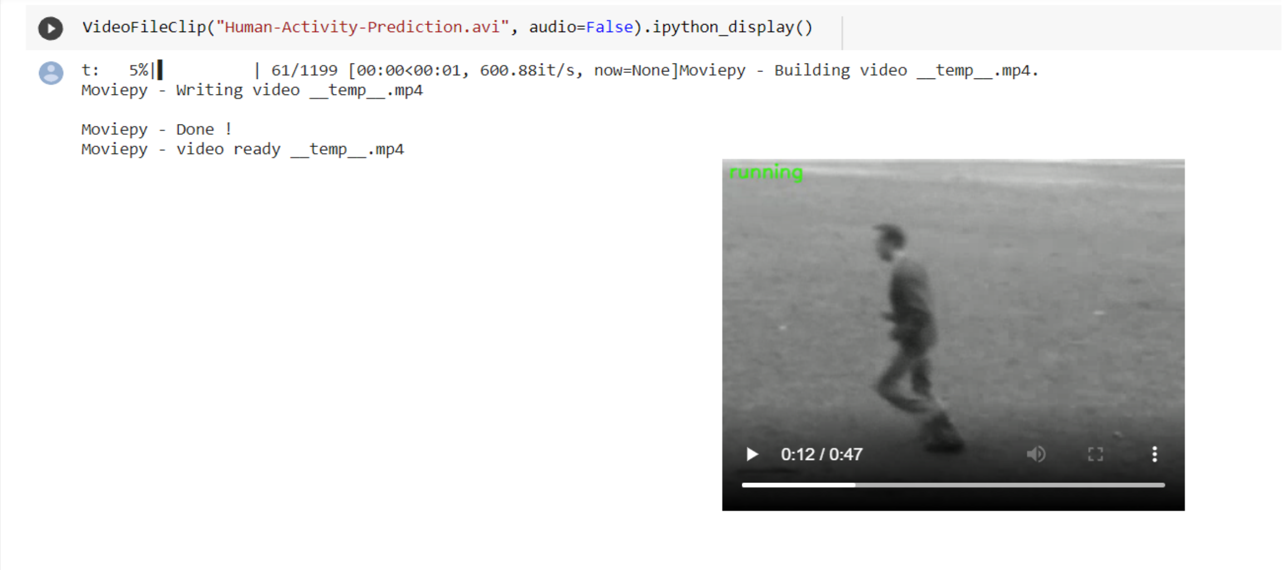
```
In [41]: # Calculate Accuracy On Test Dataset
         acc = 0
         for i in range(len(features_test)):
           predicted_label = np.argmax(model.predict(np.expand_dims(features_test[i],axis =0))[0])
           actual_label = np.argmax(labels_test[i])
           if predicted_label == actual_label:
               acc += 1
         acc = (acc * 100)/len(labels_test)
         print("Accuracy =",acc)

         Accuracy = 85.29411764705883
```



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

     Action Predicted: fight
     Confidence: 0.9998675584793091

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

     Action Predicted: running
     Confidence: 0.8557428717761322

     predict_single_action("Predict/walking.avi",SEQUENCE_LENGTH)

     Action Predicted: walking
     Confidence: 0.5534511208534241
```

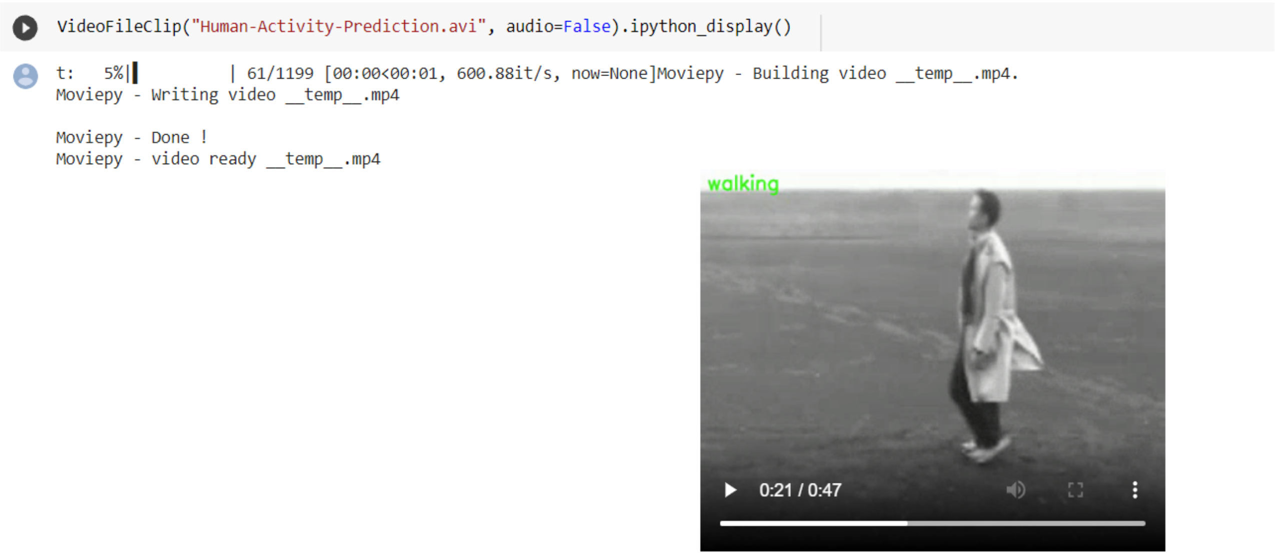


```
VideoFileClip("Human-Activity-Prediction.avi", audio=False).ipython_display()

t:   5%|▌         | 61/1199 [00:00<00:01, 600.88it/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:   5%|▌          | 61/1199 [00:00<00:01, 600.88it/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:   5%|▌          | 61/1199 [00:00<00:01, 600.88it/s, now=None]Moviepy - Building video __temp__.mp4.
Moviepy - Writing video __temp__.mp4

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

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] Dinesh Jackson Samuel R,Fenil E, Gunasekaran Manogaran, Vivekananda G.N, Thanjaivadivel T , Jeeva S , Ahilan A, *“Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM”*,The International Journal of Computer and Telecommunications Networking,2019.