Suspicious Activity Detection from Surveillance Video Using Deep Learning

**SIDDHARTH CHOUDHURY**

**IDNO: 2020A7PS0028U**

**ASHWIN SHIBU**

**IDNO: 2020A7PS0030U**

**Under the supervision of**

**Mr. Tamizharasan Periyasamy**

****

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**DUBAI CAMPUS, DUBAI UAE**

# INTRODUCTION

# Human behavior recognition in real-world environments, such as intelligent video surveillance, is crucial for ensuring security in various settings. Manual monitoring of CCTV footage is impractical, and automated systems are needed to analyze abnormal events. Artificial Intelligence, Machine Learning, and Deep Learning, supported by powerful architectures like Convolutional Neural Networks and Long Short-Term Memory models, play a significant role in this domain. By automatically extracting features and recognizing patterns, these models enable efficient event detection and human behavior recognition. Implementing such systems in campuses and other sensitive areas can prevent theft, vandalism, terrorism, and ensure the safety of individuals. Training these surveillance systems involves data preparation, model training, and inference.

# REAL WORLD APPLICATION

Recognizing suspicious human activities from video surveillance is crucial to prevent theft, terrorist attacks involving abandoned objects or explosives, vandalism, fights, personal attacks, and fires in highly sensitive areas such as banks, hospitals, malls, transportation hubs, airports, refineries, nuclear power plants, educational institutions, and borders.

Video surveillance in university campuses and academic institutions ensures the safety of assets from theft and vandalism. It also helps prevent inappropriate student behavior, fights, and monitors the perimeter for the safety of students and faculty. During exams, video surveillance can monitor for suspicious activity in examination halls.

Ultimately, video surveillance enables maintaining order and security in academic institutions with fewer guards required.

# APPROACH

The proposed approach involves utilizing CCTV camera footage to monitor students' activities in a campus setting and notifying the relevant authority when suspicious events occur. The architecture consists of several phases, including video capture, video pre-processing, and class prediction. The system classifies the videos into three categories:

* Suspicious class: Students fighting in the campus.
* Normal class: Walking and running activities. By implementing this system, any instances of student fighting can be promptly identified and appropriate action can be taken. Additionally, normal activities such as walking and running are classified as non-suspicious.

## Video capture

The first stage of implementing a video surveillance system involves the installation of CCTV cameras and the subsequent monitoring of the captured footage. Multiple cameras are strategically positioned to cover the entire surveillance area, resulting in the collection of diverse video recordings.

## Video Pre-processing

## During the pre-processing stage, we extract 30 frames from each recorded video. These frames are evenly spaced apart in terms of time intervals. To facilitate further analysis, we resize each of the extracted frames to dimensions of 64 x 64 and convert them into a numpy array with a shape of (64 x 64 x 3), where the three dimensions represent the image width, height, and RGB channels, respectively. This conversion is done using the OpenCV library in Python.

## Next, we normalize the values in each frame by dividing them by 255. This normalization process scales the pixel values to a range between 0 and 1, facilitating consistent processing and analysis.

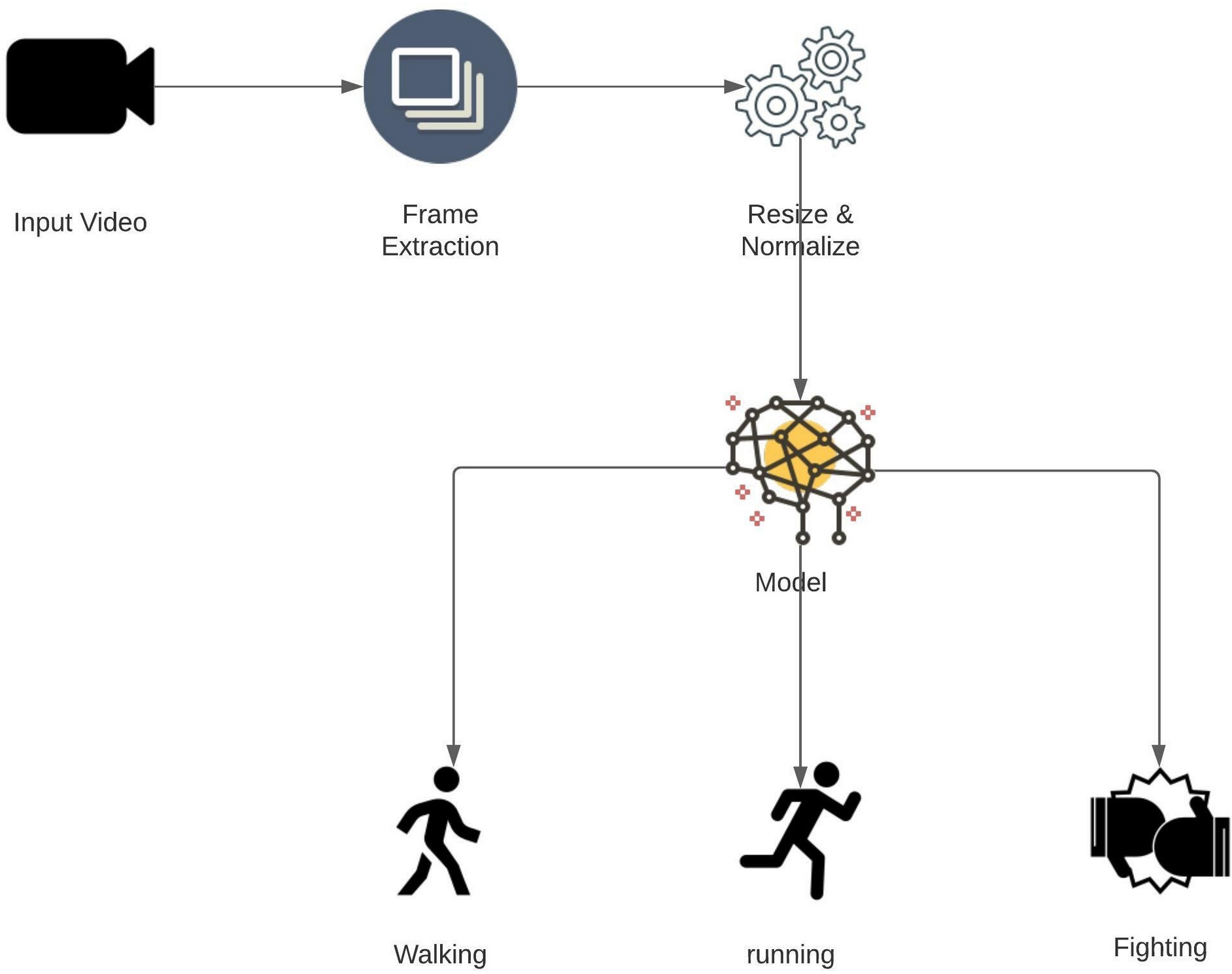
## Finally, we store all the 30 normalized frames from each video as a sequence in a numpy array with a shape of 30 x 64 x 64 x 3. This array structure allows for efficient handling and manipulation of the video frames during subsequent stages of the pipeline.

## Class Prediction

The model takes a numpy array as input, which represents a video, and makes predictions about the class or category of that video.

# PROPOSED SYSTEM AND DESIGN

Our proposed system utilizes the LRCN (Long-term Recurrent Convolutional Network) approach to detect anomalous behavior. Recognizing the temporal data within videos is crucial for effectively classifying anomalous activities. In current practice, CNN (Convolutional Neural Network) is widely employed to extract key features from individual frames of the video. To ensure successful classification of the input, it is imperative that CNN is capable of understanding and extracting the relevant features from video frames. We extract a sequence of 30 frames from the video and pass them to the LRCN Model for further processing and analysis.



# IMPLEMENTATION OF OUR UPGRADED 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 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).

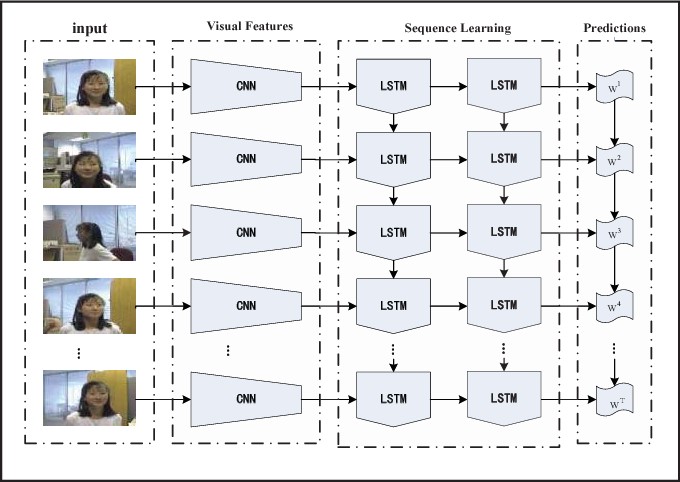
### Data Pre-processing

1. **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.
2. **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.
3. **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.
4. **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.
5. **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 using in our proposed system for suspicious activity detection from video surveillance.

*LRCN Model*

In 2016, a group of authors proposed a class of architectures called LRCN, which enables end-to-end training for visual recognition and description tasks. The core concept behind LRCN involves combining convolutional neural networks (CNNs) with long short-term memory (LSTM) networks. This combination allows for the learning of visual features from video frames using CNNs, and then utilizing LSTMs to transform a sequence of image embeddings into various outputs, such as class labels, sentences, or probabilities.

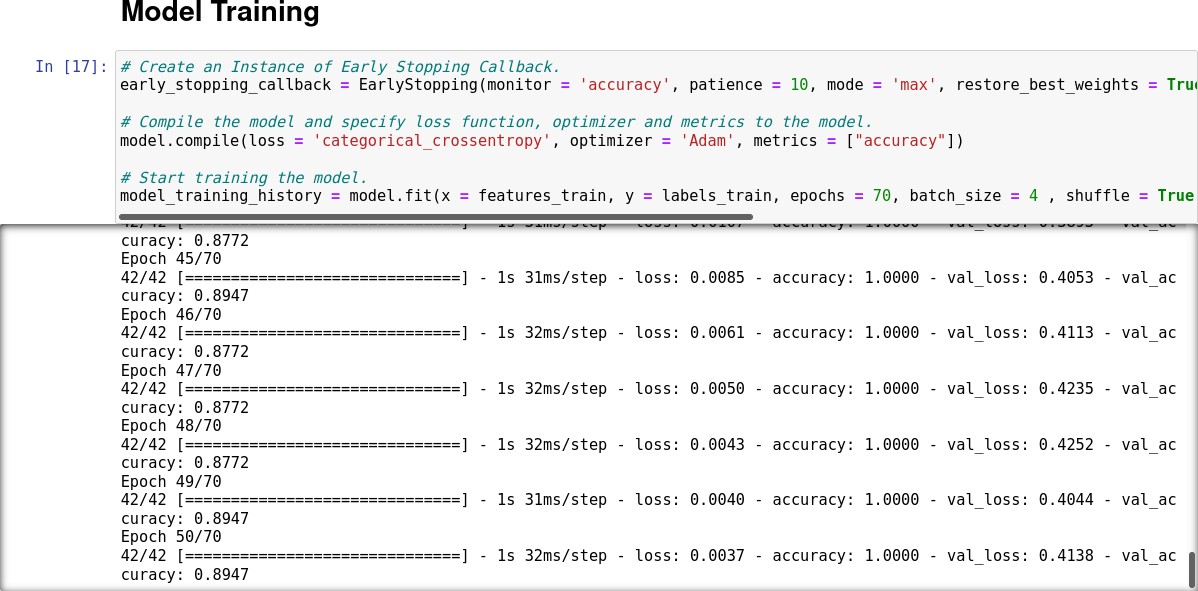
The primary advantage of using LSTM networks is their suitability for processing and classifying time series data. They are capable of handling situations where there may be unpredictable delays between significant events in the time series. LSTMs were specifically designed to address the problem of vanishing gradients, which can arise when training conventional recurrent neural networks (RNNs).

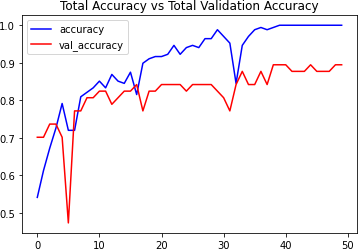
### Model Training

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

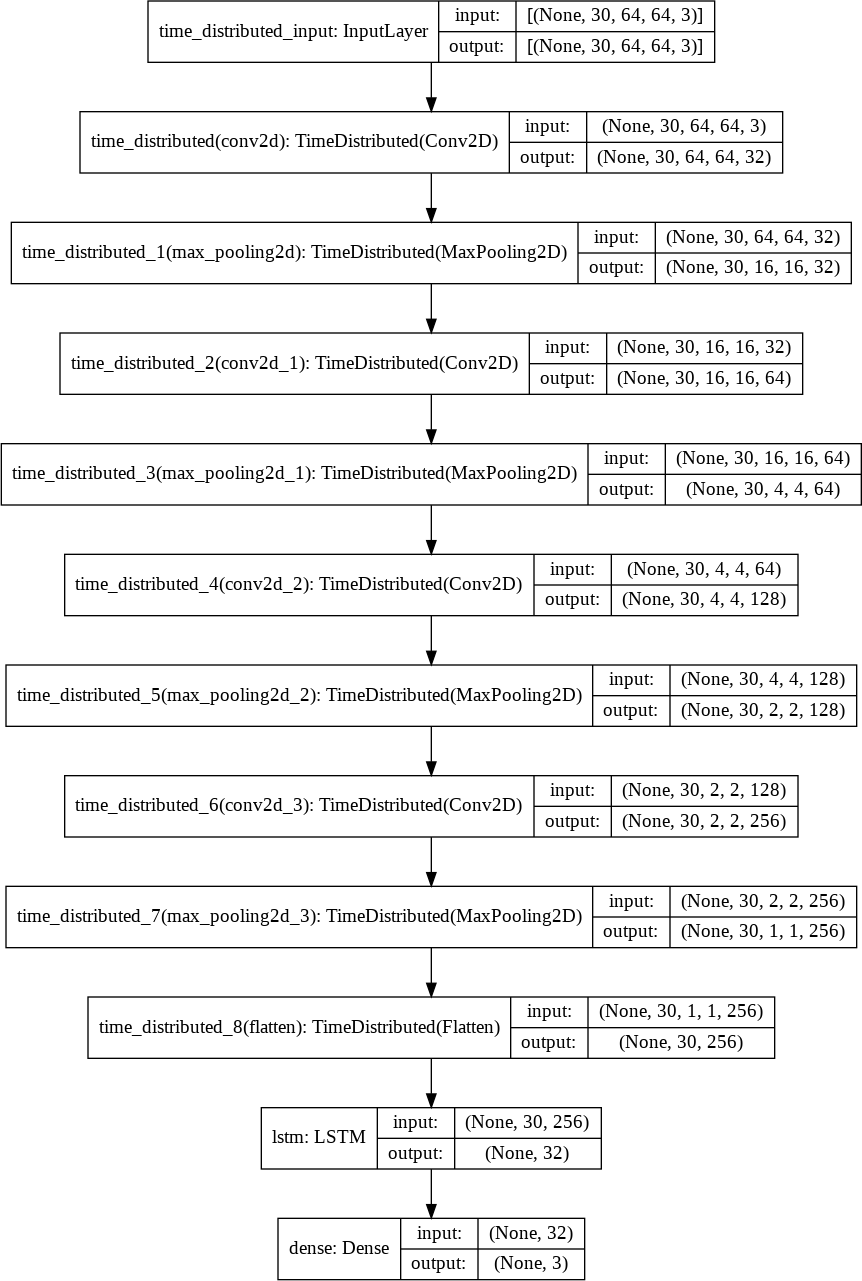


A picture containing screenshot, plot, line, diagram

Description automatically generatedModel Training graphs :

Loss vs Validation Loss Accuracy vs Validation Accuracy

# MODEL LAYER DIAGRAMS



**UP-GRADATION**

Our previous Model VGG-16 with LSTM, had the following shortcomings -

1. Took too long to train even on high end devices as it had over 19 CNN Layers.
2. Long prediction time.
3. As the model required 224 x 224 frames it was nearly impossible to load that many frames in the Numpy array and the Dataset had to be shrinked by 55%.
4. Totally Not suitable for real life scenarios.

Upgradations we did to our model -

1. Created a custom LRCN Model which has only 12 Layers and does not take too long to train.
2. Short Prediction times.
3. The Frames are now resized to 64 x 64 which makes it possible to load the whole dataset even on low end devices.
4. Suitable for real life scenarios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MODEL | DATASET | FRAME SIZE | ACCURACY | NEAR REAL- TIME |
| VGG-16+ LSTM | 45 videos/Class | 224\*244px | 85% | No |
| LRCN | 100  videos/Class | 64\*64px | 82% | Yes |

# RESULT

Our proposed model aims to detect anomalous behavior in videos and has achieved an accuracy of 82% on our custom dataset. To address the limitations of our previous model, which used the VGG-16 architecture with 16 layers and was time-consuming, we transitioned to the LRCN model with 11 layers. This reduced the computational time, enabling real-time detection.

In order to save memory space, we resized the frames from 224px to 64px. Additionally, we augmented our dataset by including more videos, which helped improve the overall accuracy of the model. The dataset consists of videos showcasing anomalous behavior, such as fighting, as well as videos exhibiting normal behavior, such as walking and running.

Below are some images displaying the results obtained from the proposed model.

