Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.12 No.1, 2023, pp.35-40
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajcst-2023.12.1.3630

Human Anomaly Detection Using Deep Learning

Lambu Yashaswi[1], Sreenivas Mekala[2] and Maktala Durga Prasad[3]
[1&3]Student, [2]Associate Professor,
Department of Information Technology, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India
E-mail: lambuyashaswi@gmail.com, msreenivas@sreenidhi.edu.in, 19311a12d6@sreenidhi.edu.in
(Received 19 May 2023; Revised 27 May 2023; Accepted 30 May 2023; Available online 3 June 2023)

***Abstract* - Human Anomaly Detection can be used in order to identify thefts, terrorist attacks, fighting, and fires in susceptible areas including banks, parking areas, hospitals, shopping malls, universities, colleges, schools, borders, airports, bus and railway stations, etc. Video surveillance can be used in crowded areas to identify anomalies and analyse human behaviour to detect theft and vandalism. It will also help to prevent inappropriate behaviour such as fighting among humans by monitoring the perimeter of the location, for the safety of people. It can be used to monitor the suspicious activity of humans in crowded places.**
***Keywords:* Human Anomaly Detection, Deep Learning, LRCN, LSTM, CNN**

I. INTRODUCTION

Human Anomaly Detection can be used in plenty of real-time applications. It includes shopping behaviour analysis, intelligent video surveillance and much more. Surveillance is now becoming vital in terms of security. Even this can be used to identify emergency situations in the public places. CCTV (Closed Circuit Television) monitoring manually takes a lot of time and effort.

It is not efficient to get past information and analyse them. Identifying abnormal events and analysing videos is now an emerging concept of automated video surveillance systems. The intelligent way of detecting any suspicious activity can be automated to detect human behaviour.

The learning tasks can be performed efficiently using Deep Neural Networks. Deep Learning models have the capability of extracting the features and also representing high-level image data. CNN (Convolutional Neural Network) can identify the patterns from images, but in order to extract the long-term dependencies from videos the LSTM can be used as it is the capability to remember things.

This proposed technology uses CCTV camera footage to monitor human behaviour in crowded places. Here event detection and also human behaviour recognition are the main aspects of intelligent video monitoring. The whole process involves three main processes, data preparation, model and inference.

II. METHODOLOGY

This approach makes the use of footage which is obtained by using CCTV cameras for identifying suspicious events. The architecture involves capturing, pre-processing, and prediction. Three classes were identified by the system.

1. Fighting - Suspicious
2. Running and Walking - Normal

A. Video Capture

CCTV camera is installed, the footage monitoring was the basic step in surveillance software. The whole area of surveillance is covered by using different kinds of videos.

B. Video Pre-Processing

30 frames are identified from each of videos, frames are separated on equal time intervals. 30 extracted frames are resized to 64 x 64 and read in a NumPy array of dimensions (64 x 64 x 3) using OpenCV Library in Python.

Each Value in the frame is then Normalized by dividing it by 255. All the 30 Normalized frames from each video are stored as a sequence in a NumPy array with dimensions 30 x 64 x 64 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.

III. PROPOSED SYSTEM AND DESIGN

LRCN can be used to recognize suspicious human activity. Temporal data is used for the effective classification of anomalous activities. CNN extracts the features from every frame of the video. The given input is successfully classified, and features are extracted from CNN. Sequences of 30 frames present in the video are extracted and given to the LRCN Model.

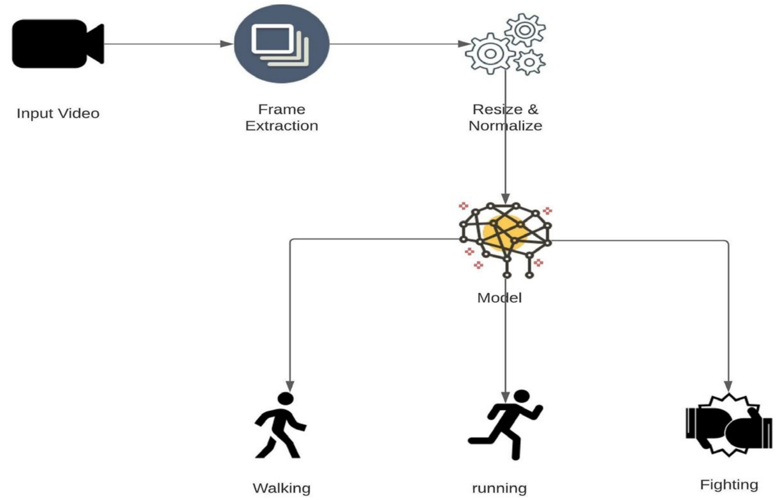

Fig. 1 Architecture of the Proposed System

A. Dataset Description

For detecting Running and Walking KTH database is used. KTH dataset is the standard one consisting of collection expressing 6 various actions, every action has 100 steps. Every step has maximum of frames 600 and video was shot for 25 fps. The Kaggle has of, at least 100 videos which were taken from YouTube videos, movies that can also be employed for the process of training anomalies.

B. Data Processing

To identify the class from its respective Class folder and the Class label from NumPy array, OpenCV library was used. Video is taken as input using OpenCV, 30 frames are identified to get a sequence of 30 images.

Resizing of an image was done to have fixed number of the pixels. The frames are resized width: 68px and height: 68px to establish uniformity of the input images present in the model.

Normalization is used to learn the data faster and extract necessary features from images. The frame which is resized is normalized and is dividing it by 255. The uniformity value of the pixel is maintained between 0 to 1. Resized and as well as Normalized frames are then stored into NumPy arrays which can be used as an input for the model.

C. Data

75% - Training Data
25% - Testing Data

D. Creating Model

LRCN is used to recognise the anomalous activity from video surveillance.

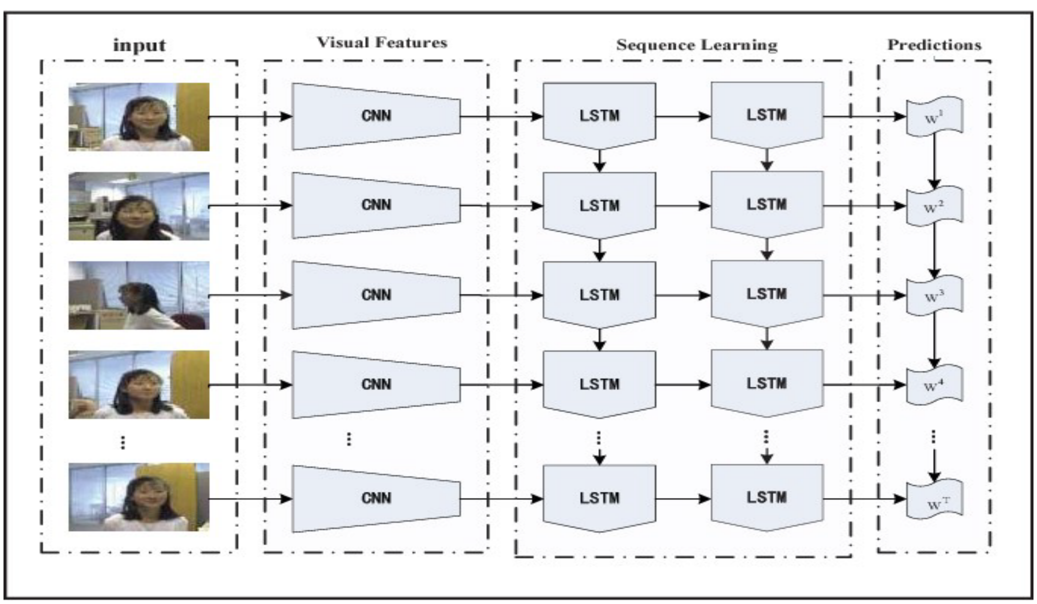

Fig. 2 Model Creation

LRCN is to mainly used for combination of the CNNs in order to identify the features among the frames. LSTMs are used to transform sequence as a valid class label, probabilities and sentence. The input is then processed using CNN, where the outputs are given for a stack of recurring models. It involves classifying, processing, and predictions on data. LSTMs are mainly used to deal vanishing gradient problem that can be extracted when training RNNs.

E. Training the Model

The model learns from the labelled data and also has the capability to predict 3 classes – Running, Fighting and Walking. The hyper parameters used are:

1. validation split = 0.25
2. epochs = 70
3. batch size = 4

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
```

Fig. 3 Training of the Model

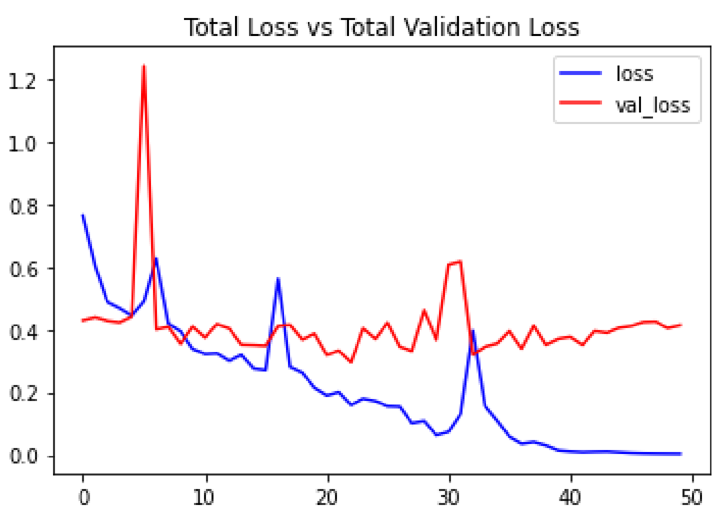

Fig. 4 Graph to identify the total loss and validation loss

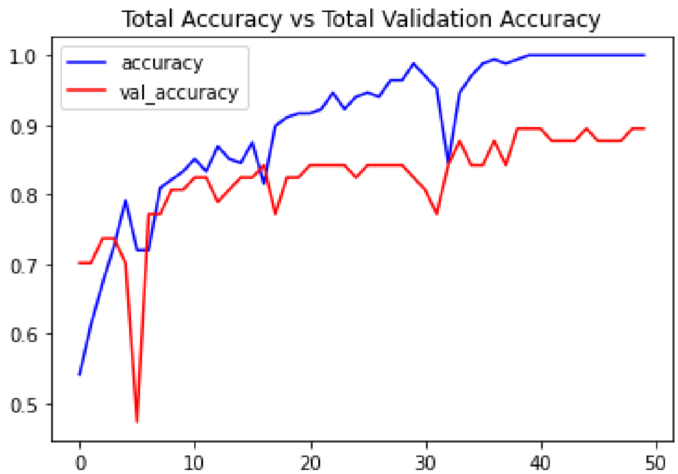

Fig. 5 Graph to identify the total accuracy and total validation accuracy

IV. MODEL LAYER DIAGRAMS

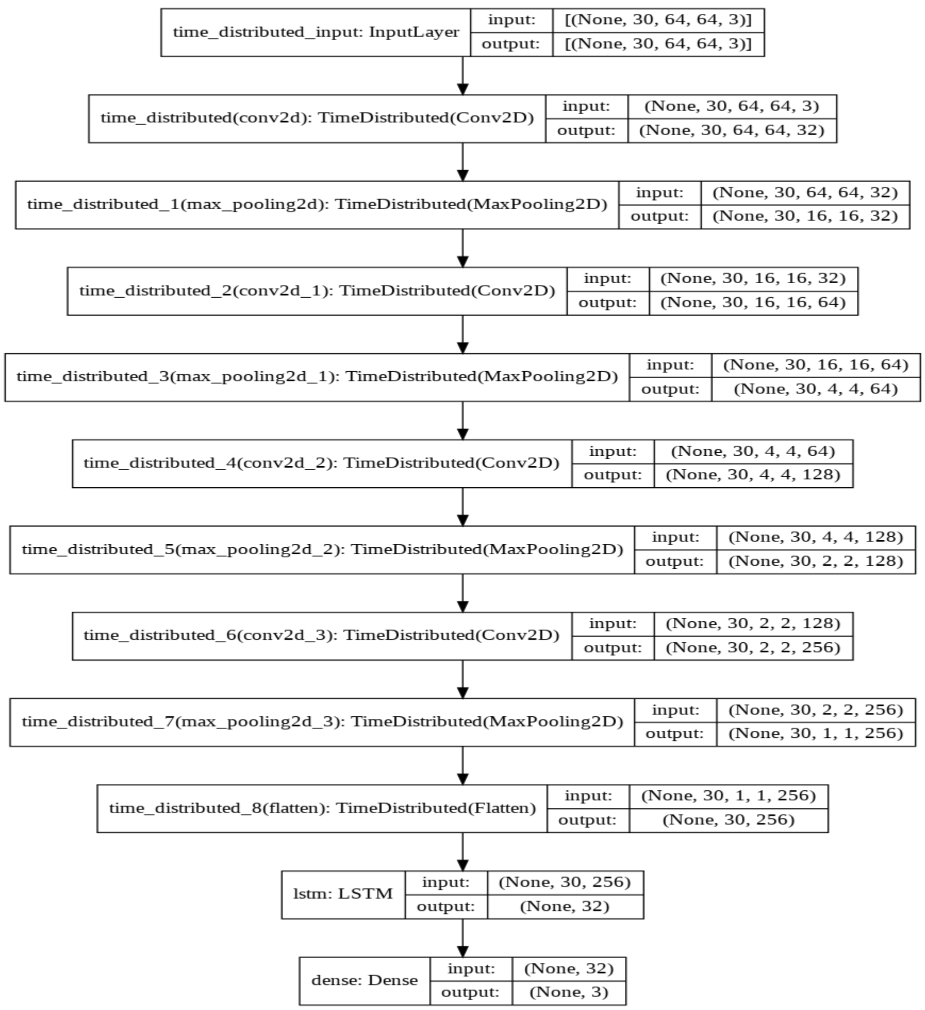

Fig. 6 Model Layer Diagram of LSTM

TABLE I COMPARATIVE ANALYSIS OF VGG-16+LSTM AND LRCN MODELS

Model	Dataset-Class	Frame Size(px)	Accuracy	Realtime
VGG-16+ LSTM	45	224*244	85%	No
LRCN	100	64*64	82%	Yes

V. RESULTS AND DISCUSSION

The Proposed model aims to detect the anomalous behavior in the videos and has an accuracy of around 82% on our data. VGG-16 model consists of 16 layers and is not efficient as it was time-consuming and could not be used in detecting real life situations. In order to overcome this drawback LRCN model was used, size of layers is downgraded by 5 and has less time complexity and can be applicable in real time object detection. The frames are then resized to 64px to have more memory as well as many videos can be added to increase efficiency. Following are images of the 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
```

Fig. 8 Accuracy of the model and individual accuracy for predicting the single action

```
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
```

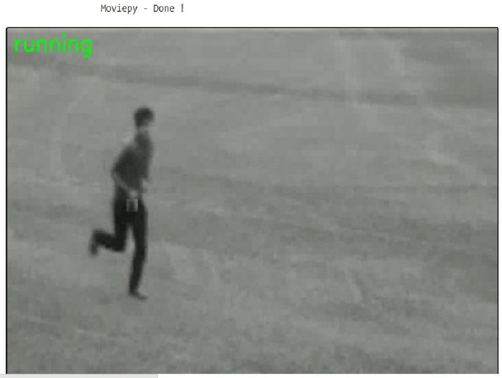

Fig. 8,9,10 Predictions of fighting, running, and walking by model

VI. CONCLUSION

LRCN is to mainly used for combination of the CNNs in order to identify the features among the frames. LSTMs are used to transform sequence as a valid class label, probabilities and sentence. The input is then processed using CNN, where the outputs are given for a stack of recurring models. It involves classifying, processing, and predictions on data. LSTMs are mainly used to deal vanishing gradient problem that can be extracted when training RNNs. We conclude that LRCN is efficient in comparison to the other models and is much more effective.

REFERENCES

[1] P. Bhagya Divya, S. Shalini, R. Deepa and Baddeli Sravya Reddy, “Inspection of suspicious human activity in the crowdsourced areas captured in surveillance cameras,” *International Research Journal of Engineering and Technology (IRJET),* Vol. 04, No. 12, December 2017. [Online]. Available: https://qu.edu.iq/journalcm/index.php/journalcm/article/view/858

[2] Prof. Jitendra Musale, Miss. Akshata Gavhane, Mr. Liyakat Shaikh, Miss. Pournima Hagwane and Miss. Snehalata Tadge, “Suspicious Movement Detection and Tracking of Human Behavior and Object with Fire Detection using Closed Circuit TV (CCTV) cameras,” *International Journal for Research in Applied Science and Engineering Technology (IJRASET),* Vol. 5, No. 12, pp. 2013-2018.

[3] R. Dinesh Jackson Samuel, E. Fenil, Gunasekaran Manogaran, G. N. Vivekananda, T. Thanjaivadivel, S. Jeeva and A. Ahilan, “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, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1389128618308521.