International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 04, April-2023

Security Surveillance System using Computer Vison

Abhinav Naidu Chintala (Lead)
Dept. of Computer Science Engineering
Gitam University
Vizag, India

Nurzahan Mohammad
Dept. of Computer Science Engineering
Gitam University
Vizag, India

Ashish Addepalli
Dept. of Computer Science Engineering
Gitam University
Vizag, India

***Abstract*— Security is a fundamental word/issue in today's world. Nowadays, many surveillances, CCTV cameras, and other types of monitoring cameras are used in various major and minor ways. Visual tracking by human beings is becoming a complex job as there are many ways to simultaneously inflow data from various sources. In this project, we are trying to eliminate that constraint by using an intelligent security surveillance system using computer vision. Here, we are going to detect the activity going on in a suspected video and going to predict the action involved in it. The suspect video is selected when an object's movement is detected.**

We are using the UCF-50 dataset. It contains different categories of videos which were extracted from YouTube. There are 50 categories of videos and within each video category there are 25 groups. If two videos are in same group then they have similar view point of camera and some other common features like same background and same person, etc. We trained this model using this UCF-50 dataset.

Here the model works by breaking the video into each frame, then classifying it, and then predicting the activity of the video by classifying each frame by CNN and then predicting using LSTM.

Keywords—UCF-50, LSTM, RNN, CNN, 2D ConvNet

I. INTRODUCTION

Our Objective is to find the activity going on in a video and forecast the action that will be taken in that video. The suspect video is selected, when an object movement is detected..If any violent action is detected, it is immediately alerted.

We are going to build a model that detects the action involved in the given video

We are using:

1. Long Short Term Memory (LSTM)
2. Conventional Neural Network (CNN) algorithms for detection of action involved in the video.

We are using the UCF-50 dataset provided by the University of Central Florida. The dataset consists of 50 action categories, some real-world videos extracted from YouTube. Each video consists of 25 groups, and a group has at least more than four videos. The video clips in the same group may share some standard features. The size of the dataset is 3712 videos & 3.5 GB of memory.

II. LITERATURE REVIEW

Here in this section we described the different methodologies adopted for the literature review. In this project we also represent the contribution and learnings take from them:

[1] Ann, Ong Chin, and Lau Bee Theng described that the ability to understand human body motion or gesture using sensors and identify human activity or action is known as "human activity recognition." Most of the human everyday duties can be reduced or automated if they can be detected by HAR system. HAR systems are often either supervised or unsupervised.

[2] Ambeshwar Kumar and Dr. Rajesh T.M proposed that, using feature extraction technique and algorithm, we match the features of the object from the saved database object and then recognise the object. Video is captured from the database, changed into a number of frames using Mat Lab, and the extracted frames are going to be saved in a database.

[3] In their study of deep architectures for gesture detection in video, Lionel Pigou, Aaron van den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre offer a new end-to-end trainable neural network architecture that includes temporal convolutions and bidirectional recurrence.

[4] Late fusion was proposed by Yunke Zhang, Lixue Gong, Lubin Fan, Peiran Ren, Qixing Huang, Hujun Bao, and Weiwei Xu. They claimed that the network can implicitly produce trimaps without user interaction and that this makes it simple for beginners who are unfamiliar with digital matting to use. Experimental results reveal that our network can create high-quality alpha mattes for many types of objects and outperform the state-of-the-art CNN-based image matting algorithms on the human picture matting test.

[5] Given that the strategy shown in Figure 3 is an early fusion technique that aggregates data using the preprocessed raw signals at the lowest level of abstraction, it was developed by Patrick Thiam, Peter Bellmann, Hans A. Kestler, and Friedhelm Schwenker. A 2D representation of the input data that are generated by concatenating the multiple physiological modalities along the temporal axis, likely to result in a tensor with the dimensionality 3 × 1152 × 1. Following that, the obtained data are fed into a network made up of 2D convolutional layers.

[6] A homogenous architecture with modest 3×3×3 convolution kernels in all layers is among the top performing architectures for 3D ConvNets, according to the research paper "Learning Spatiotemporal Characteristics with 3D Convolutional Networks."

[7] Hasim Sak, Andrew Senior, and Francoise Beaufays released an article by Google, introducing the first distributed training of LSTM RNNs on a large cluster of machines utilizing asynchronous stochastic gradient descent optimization. We demonstrate that a two-layer deep LSTM RNN with a linear recurrent projection layer in each LSTM layer outperforms state-of-the-art voice recognition performance.
[8] E. Bermejo1 , O. , G. Deniz 1. R. and Bueno 1. Sukthankar2 proposed that the performance of modern action recognition algorithms for the recognition of fights in videos, movies or video-surveillance recordings.
[9] Kishore K. Reddy, Mubarak Shah demonstrated how the motion descriptors lose their discriminative power as the number of categories rises. They also showed that the suggested scene context descriptor is more discriminative, and when appropriately merged with motion descriptors offers 15% and 4% improvement on UCF50. Our method does not need stabilizing videos, removing motion or static elements, or detecting and following people.

III. PROBLEM IDENTIFICATION AND OBJECTIVES

The objective is to find the activity going on in a video and forecast the action that will be taken in that video.
The model breaks the video into frames and generates an output for each frame using CNN. LSTM is then used to forecast the activity of the video using a sequence of CNN outputs. Each video consists of a certain number of frames.
The dataset UCF50 is a data set for action recognition that includes 50 categories of action and real-world videos which were taken from YouTube. This data set is an advanced version of data set (UCF11) which comprises 11 action category types.
The majority of the data sets used for action recognition today are staged actor performances and are not realistic. Our main goal in gathering data and use this data to predict the action involved in the video, which will closely act as realistic videos from surveillance cameras that can be used for action recognition. Due to wide variations in camera motion, item look and posture, scale of the object, viewpoint, cluttered background, illumination conditions, etc., our data set is quite difficult to work with. The videos are given in form of 25 groups for each of the 50 categories, with each group containing at least 4 videos.

IV. SYSTEM METHODOLGY

As we Know that the feed comes from the CCTV's and also some other various sources. If we consider a particular feed, at first it watches for any object movement in the video. If there is any it will start saving the video until for certain time. Then it sends the saved sample to our model, there it will predicts the activity in video and detects the output for the video. If any violent activity is detected then it will immediately send an alert the concerned people, which helps them to act immediately to the incident. If its not an violent activity , video can be ignored.

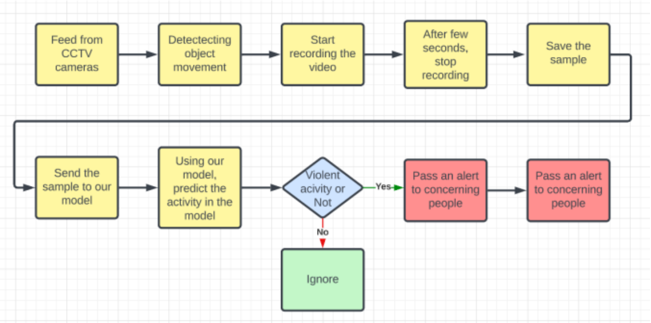
Feed from CCTV cameras
Detectecting object movement
Start recording the video
After few seconds, stop recording
Save the sample
Send the sample to our model
Using our model, predict the activity in the model
Violent acivity or Not
Yes
Pass an alert to concerning people
Pass an alert to concerning people
No
Ignore

Fig 4.1 shows working of our project

Actually the model will be executed by following certain steps:

At first importing of libraries is done, and fetched the dataset UCF-50 which was provided by University of central Florida.
Now, we performed Exploratory data analysis on the dataset to know more about the dataset. In this we just printed the first frame of video and it corresponding class.
After the above process, the collected dataset is went into preprocessing phase where we normalized the video and resized it and then extracted the frames from it. Then we performed one hot encoding and splitting operation is performed on that generated numerical data where the generated dataset is divided with a ratio of 8:2 which is 80%-training and 20%-testing. After we got the dataset, we constructed 2-models LRCN and ConvLstm models.

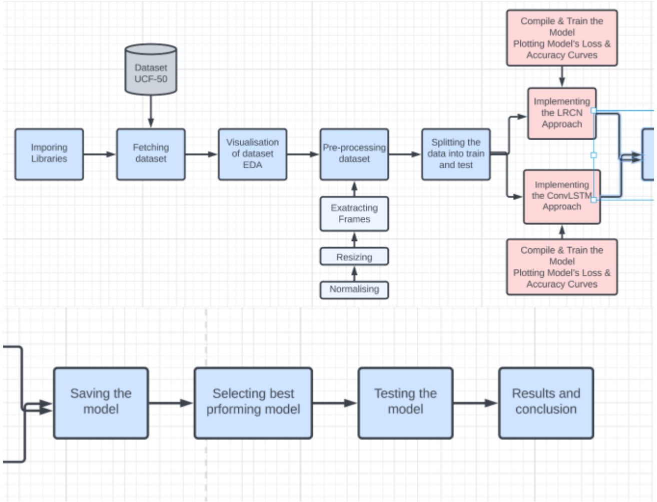
Dataset UCF-50
Imporing Libraries
Fetching dataset
Visualisation of dataset EDA
Pre-processing dataset
Exatracting Frames
Resizing
Normalising
Splitting the data into train and test
Compile & Train the Model Plotting Model's Loss & Accuracy Curves
Implementing the LRCN Approach
Implementing the ConvLSTM Approach
Compile & Train the Model Plotting Model's Loss & Accuracy Curves
Saving the model
Selecting best prforming model
Testing the model
Results and conclusion

Fig 4.2 shows execution of our project

V. OVERVIEW OF TECHNOLOGIES

We used following algorithims in the model:

A. CNN

A subset of deep learning algorithms called convolutional neural networks (CNNs) are particularly good at processing and identifying pictures. Among the layers that make up this structure are convolutional layers, pooling layers, and totally linked layers[10].

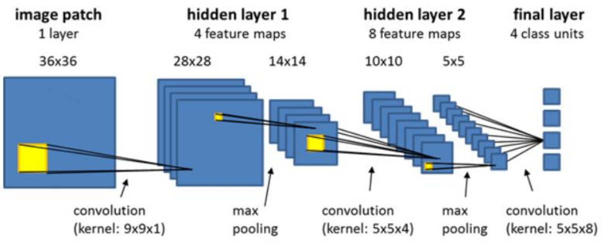

Fig 5.1. CNN structure

The key part of a CNN is its convolutional layers, where filters are used to extract characteristics like edges, textures, and forms from a given input image. And output of the convolutional layers is then sent through pooling layers, which are employed to down-sample the feature maps and retain the most crucial data while lowering the spatial dimensions. The output of the pooling layers is then transmitted through one or more fully connected layers, which are utilized to produce a prediction or classify the image[10].

B. RNN

Recurrent neural networks, or RNNs, we can describe them as subset of artificial neural networks that can analyse sequential input, identify patterns, and forecast results. The reason why this neural network is referred to be recurrent is because it may repeatedly carry out the same action on a series of inputs. An RNN's internal memory enables it to retain or memorise the details of the input it received, aiding the system in understanding the context. Consequently, an RNN will be a suitable fit to handle sequential data, such as a time series.[11]

A CNN or a feed-forward neural network cannot accomplish this since they are unable to sort the correlation between one input and the next. The idea behind an RNN is to store a layer's output and feed it back to the input in order to anticipate that layer's output.

Here's a straightforward illustration of how to change a feed-forward neural network into a an RNN [11].

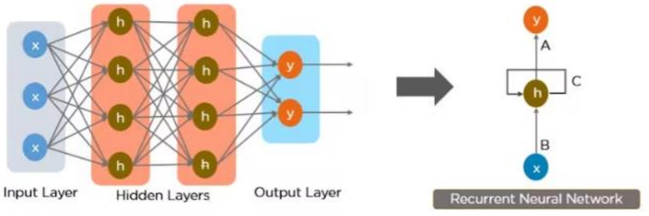

Fig 5.2. RNN structure

C. LSTM

LSTM networks are an example of a recurrent neural network (RNN) that may develop long-term relationships between the time steps of sequence data. A sequence input layer and an LSTM layer are the two main parts of an LSTM network. A sequence input layer feeds the network with data from time series or sequences. The long-term correlations between the time steps in sequence data are learned by an LSTM layer[7].

The fundamental topology of an LSTM network used for classification is shown in this diagram. An LSTM layer precedes a sequence input layer in the network. Class labels may be predicted by the fully connected, softmax, and classification output layers of the network.

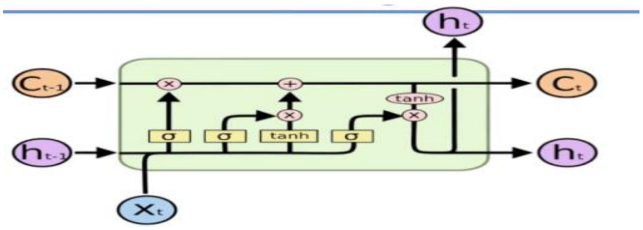

Fig 5.3. LSTM structure

VI. IMPLEMENTATION

A. At first we have imported all the necessary libraries and the downloaded the dataset and loaded. Then we visualized the dataset. Which given output as:

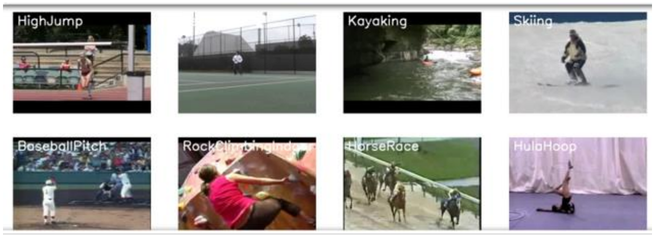

Fig 6.1.1. EDA of dataset

B. Pre-processing the dataset

Then we preprcessed data by following steps:

a) We'll be writing a function called frames_extraction(), that helps us extract frames. The video's path is taken as an input to the function, creating a list of the video's scaled and normalized frames. We cannot put each frame into a separate list. So we are going to do this in batches. The sequence length is provided by us, which decides the number of frames per batch and divides the frames equally by reading the video frame by frames

b) After that, we created a function create dataset() helps us to iterates by the class list which was provided by the user in the name CLASSES LIST constant and then calls the frame extraction() function on each video file and extracts the frames from it and returns those extracted frames along with the file path which is the class name.

c) Now we will utilize the above created function create dataset() to extract the data of the classes and a proper dataset is built with their class names.

d) We will convert those labels (which represents class indexes) into vectors which are one-hot encoded.

e) So now we obtained the dataset where all the necessary features are converted into NumPy array and we performed one hot encoding on it. Which can now feed this data into our models. Before doing that we have to divide our data into training and testing. In order to eliminate bias we are shuffled our date to in increase the accuracy and we divided our dataset into 80-20 training and testing datasets.

C. *Creating ConvLstm Model*

We are now ready to use Keras ConvLSTM2D recurrent layers to build our model. Convolutional operations require a certain number of filters and a certain size kernel, which the ConvLSTM2D layer also accounts for. By using softmax activation function, the output from each layer is fed into danse layer which helps us to find the probability for each action category.Also, we are going to add Dropout layers to avoid overfitting to the data and drop will helps us to increase the accuracy. It leaves the 20% of random output, which helps to give correct predictions and MaxPooling3D layers to reduce frame dimensions and eliminate pointless calculations. The design is straightforward and just includes a few trainable parameters. Some other defaults values which good for the model.

D. *Creating LRCN Model*

In LRCN model we are going to implement both LSTM and CNN in single model. Where LSTM predicts based on the outputs generated by CNN. As we implementing both Convolution and LSTM layers, we will apply the LRCN Approach in this step. A single pre-trained model that can be adjusted for the issue which the can derive spatial features which were extracted by video's frame data using the CNN model. The action being done in the video can then be predicted by the LSTM model using the features extracted by CNN.

VII. RESULTS

Here are accuracies of the three models:

MODEL	ACCURACIES
LRCN	82.36%
ConvLSTM	78.60%

Table 7.1. Accuracies of our models

After performing training for the two models, the accuracies are represented above for recognition.

Analysis of accuracy and loss for LRCN with respective to accuracy and loss.

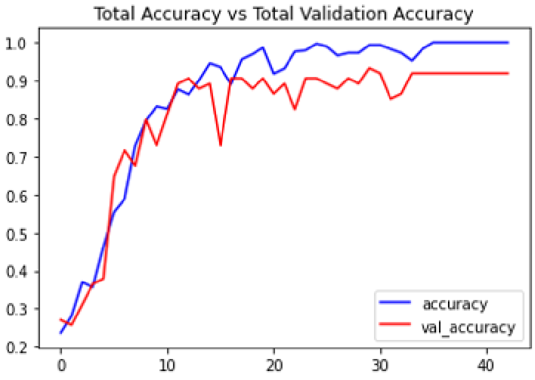

Fig. 7..1.1 shows accuracies against training and testing data of LRCN Model

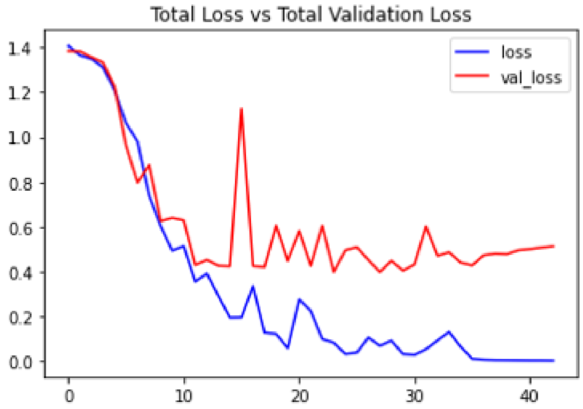

Fig. 7.1.2 shows loss against training and testing data of LRCN Model

Analysis of accuracy and loss for ConvLSTM with respective to accuracy and loss:

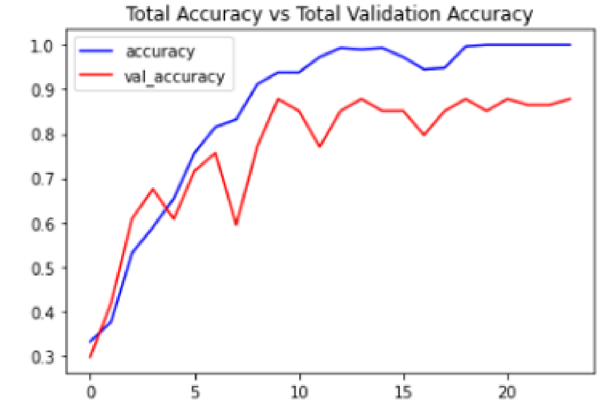

Fig. 7.2.1 shows accuracies against training and testing data of ConvLSTM Model

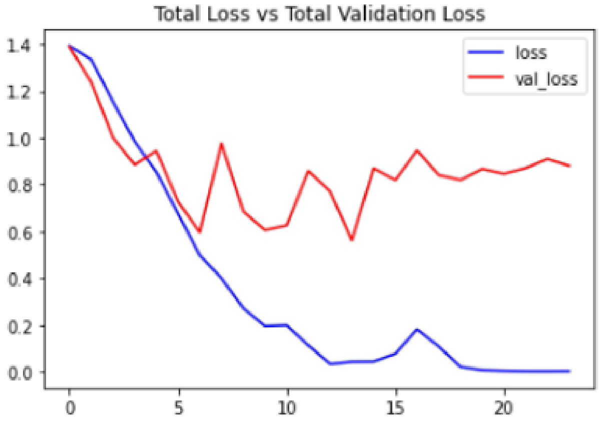

Fig. 7.2.2 shows loss against training and testing data of ConvLSTM Model

From the figure above, we could see that the accuracy graph increases rapidly. For the dataset we have assumed it is very clear that LRCN model works with more accuracy. And our model successfully predicting the output.

VIII. CONCLUSION AND FUTURE SCOPE

Above, it is clear that LRCN is the best-performing algorithm. So far, we have built our model with the limited resources available. Because of advancements in GPU technology, the execution time will decrease exponentially in the future, allowing us to execute these models with ease and have them perform well in real-world scenarios.

The Future scope is, with the advancements in technology in the future. The problem of crimes will drastically decrease by adopting this technology in real-world scenarios.

The field of human action recognition is still rapidly evolving, and there are several promising areas of research that could shape the future of the field.

It can evolve in the fields like explainable AI, where it helps to make certain predictions, improving transparency and trust. It can help identify important features for predicting actions, identify incorrect or biased predictions, and improve trust and acceptance of AI systems. Also there are other sources of information that could be used to improve human action recognition, such as audio, depth, and skeletal data. Combining these different modalities of data using fusion techniques could lead to more accurate and robust action recognition models. By also other ways like Transfer learning, Continual learning, Human-robot interactions, etc.

REFERENCES

[1] Ann, Ong Chin, and Lau Bee Theng. "Human activity recognition: A review." 2014 IEEE international conference on control system, computing and engineering (ICCSCE 2014). IEEE, 2014.

[2] Rajesh, Ambeshwar Kumar1 Dr. "A Moving Object Recognition using Video Analytics."

[3] Pigou, Lionel, et al. "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video." International Journal of Computer Vision 126 (2018): 430-439

[4] Zhang, Yunke, et al. "A late fusion cnn for digital matting." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[5] Thiam, Patrick, et al. "Exploring deep physiological models for nociceptive pain recognition." Sensors 19.20 (2019): 4503.

[6] Tran, Du, et al. "Learning spatiotemporal features with 3d convolutional networks." Proceedings of the IEEE international conference on computer vision. 2015.

[7] Sak, Hasim, Andrew W. Senior, and Françoise Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." (2014).

[8] Bermejo Nievas, Enrique, et al. "Violence detection in video using computer vision techniques." Computer Analysis of Images and Patterns: 14th International Conference, CAIP 2011, Seville, Spain, August 29-31, 2011, Proceedings, Part II 14. Springer Berlin Heidelberg, 2011.

[9] Reddy, Kishore K., and Mubarak Shah. "Recognizing 50 human action categories of web videos." Machine vision and applications 24.5 (2013): 971-981.

[10] O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.

[11] Sherstinsky, Alex. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena. 404. 132306. 10.1016/j.physd.2019.132306.