IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Real Time Crime Detection By Captioning Video Surveillance Using Deep Learning

Nagesh Nayak, Shlesha Odhekar, Sapna Patwa, Sukanya Roychowdhury

Student Student Student Professor

Information Technology,

VESIT ,Mumbai, India

Abstract: CCTV are commonly used for surveillance almost everywhere in the world. Despite that, there are a large number of crimes happening due to the lack of a proper system to detect crimes or such type of behaviour and control rates of illegal activities. Our project not only deals with detecting crimes from the video data received at real time from CCTV, but also solves problems beyond this. Storing video data is a huge problem , and also it is less secure. Our application solves this problem by eliminating the need to store the videos itself. Instead, a better solution is to store only the accurate captions of the events happening in the video along with the respective timestamp. Crimes are detected based on the captions generated by detecting certain crime related keywords like knife, thief, fire, assault etc. we store these captions in text files along with the timestamp so we can even search for a particular event in the log. It is also more secure than storing CCTV captured videos as we will actually be storing a text file and it can be encrypted using good encryption algorithms. Video, photos, and notifications of crime in real-time are sent to a human supervisor to act in a responsible manner

Index Terms **- Neural Networks, Deep Learning, Image Captioning, Real Time Video Processing**

I. INTRODUCTION

With the increasing instrumentation of our metropolitan areas, law enforcement agencies continue to invest in developments that ensure increased attention to crime and plans to computerise the detection and response to crime. The number of criminal occurrences reported in a single day is skyrocketing. Only the most serious instances are mentioned among these. However, both significant and little accidents should be investigated, as even minor incidents might have disastrous implications in the future. Robbery, vandalism, assault, murder, kidnapping, and other crimes are examples of crime events. Pretty much entire cities can now be monitored thanks to the ever-increasing installation of advanced CCTV infrastructure, though the primary aim is simply demonstrative. The current surveillance monitoring procedure is a labor-intensive manual task that is very infeasible. Manual observation and retrieval of information from video footage is a more time-consuming method. We develop an application that can locate suspicious actions in a video, specifically CCTV data, in order to simplify the manual procedure that is required when investigating crimes and providing resources to speed up the emergency response. By assessing human actions based on behaviours and reporting any violent or suspicious activity to the responsible authorities, the technology automates the video surveillance monitoring process. For ongoing (or about to happen) errors and crimes, it's only sensible to anticipate an alert or warning system. An intelligent monitoring pro-gram that uses "Image Captioning" with DL and data analytics to dramatically improve the pre-existing surveillance system for smart crime detection. Our software can execute ImageCaptioning on several CCTV clips and save the captions,as well as the capture time, in a handy log. By adding afew keywords, the file of preserved captions can be used tolook for incidences from any point in time. The requestedCCTV footage can then be obtained using the camera numberand time period returned. This might also be used to detectcrime by detecting threats (such as firearms) and performingpredictive analysis on crime patterns.

II. OBJECTIVES

The following are the objectives of this project that have been implemented:

• Compression of video files

• Extracting frames from videos and storing them as images

• Captioning of images to identify threats like weapons, fire, violence etc.

• Search features like timeframe, keywords etc

• Data encryption of the generated caption file.

III. LITERATURE SURVEY

Multiple papers were studied and their findings are being summarised in this section. This section illustrates papers studied before and during the development of the project. The papers helped in gaining insight into existing solutions, possible ways to optimize algorithms and facilitate the selection of algorithms based on their performance.

IV. TECHNICAL DEFINITIONS

A. Convolutional Neural Network[7]

A deep learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.

Fig. 1. Comparison of 5 papers[1-5]

SR	Link	Topic	Author
1	https://sci-hub.se/https://ieeexplore.ieee.org/document/8698097	Video Captioning using Deep Learning: An Overview of Methods, Datasets and Metrics	M. Amaresh and S. Chitrakala
2	https://sci-hub.se/https://ieeexplore.ieee.org/document/8821168	Crime Intention Detection System Using Deep Learning	Umadevi V Navalgund Computer Science and Engineering KLE Technological University Hubballi, India
3	https://sci-hub.se/https://ieeexplore.ieee.org/document/8113405	Using Machine Learning to Assist Crime Prevention	Ying-Lung Lin; Tenge-Yang Chen; Liang-Chih Yu
4	https://sci-hub.se/https://ieeexplore.ieee.org/document/9155832	Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications	Harsh Jain, Aditya Vikram, Mohana, Ankit Kashyap, Ayush Jain
5	https://sci-hub.se/https://ieeexplore.ieee.org/document/9230901	A Proposed Architecture to Suspect and Trace Criminal Activity Using Surveillance Cameras	Tanjila Naurin, S., Saha, A., Akter, K., & Ahmed, S.

While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

B. Recurrent Neural Network[8]

A class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable-length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition, or speech recognition.

C. VGG model[9]

VGG models are a type of CNN Architecture proposed by Karen Simonyan Andrew Zisserman of Visual Geometry Group (VGG), Oxford University, which brought remarkable results for the ImageNet Challenge. They experiment with 6 models, with different numbers of trainable layers. Based on the number of models, the two most popular models are VGG16 and VGG19.

D. LSTM model[10]

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

V. PROPOSED SOLUTION

The proposed solution is a Deep Learning application capable of running Video Captioning on multiple CCTV footage and storing the captions along with the time of capture in a convenient log. The file of saved captions can then be used to look up incidents from any instant of time just by entering a few keywords. The returned camera number and time slot can then be used to obtain the required CCTV footage. This could further be used to detect crime by identifying threats(e.g. weapons) and carry out predictive analysis of the crime patterns.

A. Stakeholders

The stakeholders of the system are basically users that would interact with the system in its entirety. The main purpose of this system is supervision of criminal activity. A plethora of surveillance devices are being used by the Defense Services for supervision and monitoring. Also companies storing valuable information, products, etc. and banks too need good quality surveillance.

B. Software Environment

The entire application is web-based. The framework and libraries used are as follows:

- UI: The framework used is Streamlit to build the user interface of the machine learning model.
- Machine Learning Libraries: Tensorflow, Keras

C. Workflow Of the System

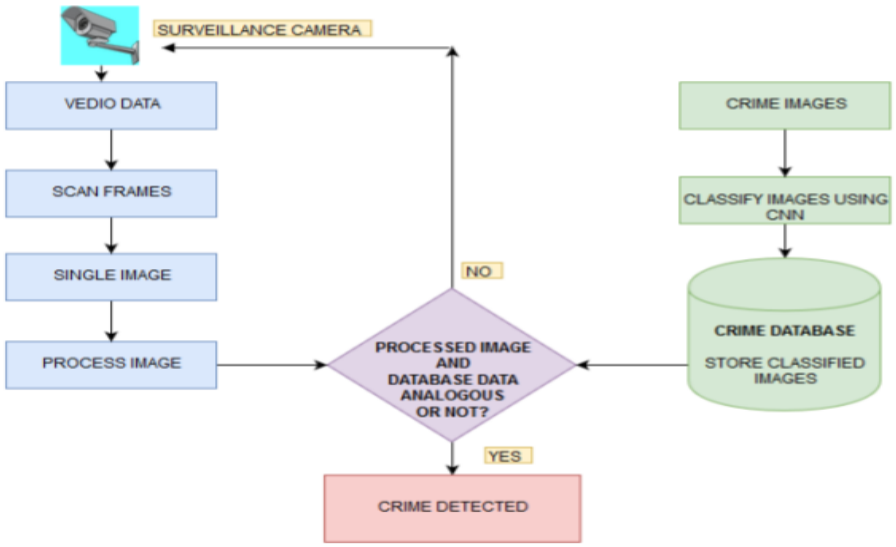

Fig. 2. shows system workflow[5]

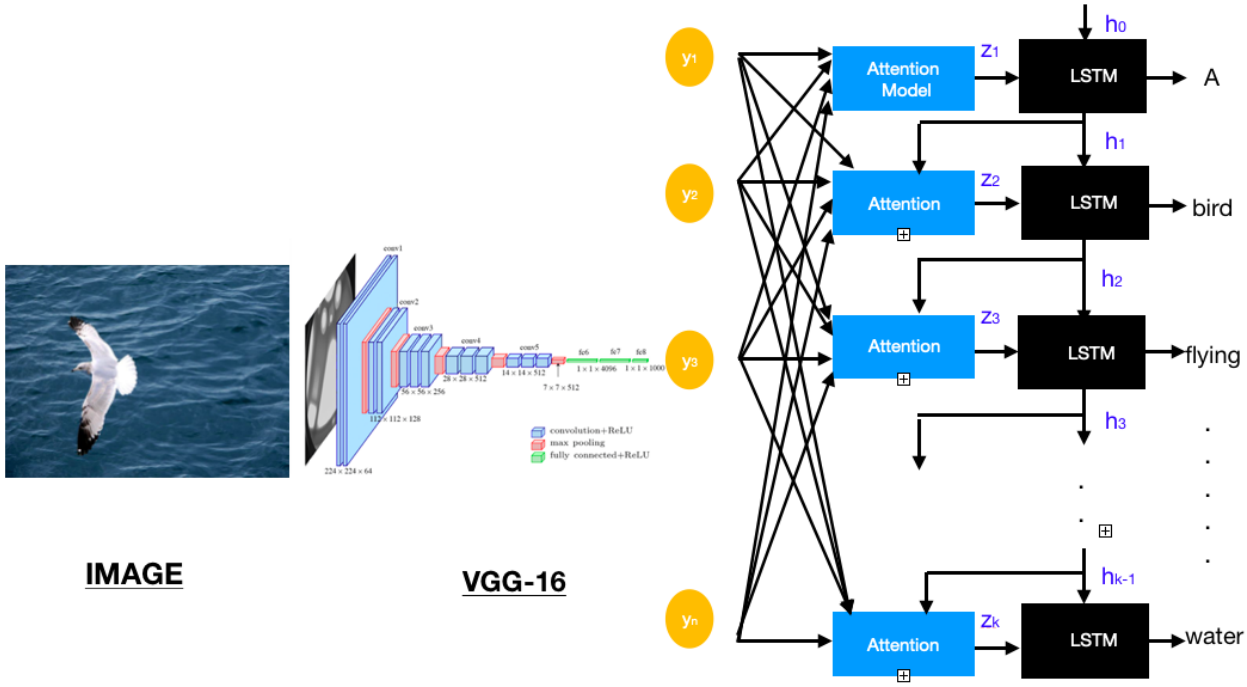

Fig. 3. Image captioning framework

D. Data Collection and Preprocessing

Data Collection:

For the purpose of this study, we have used the MSVD data set by Microsoft and UCF crime dataset[6]. This data set contains 1450 short YouTube clips that have been manually labelled for training and 100 videos for testing. Each video has been assigned a unique ID and each ID has about 15–20 captions. While the UCF Dataset[6] has 1900 long videos from which we chose some according to the scenario , for e.g. arson , assault ,etc.

Understanding the dataset:

On downloading the data set there are two folders training- data and testing data. Each of the folders contain a video sub folder which contains the videos that will be used for training as well as testing. These folders also contain a feat sub folder which is short for features. The feat folders contain the features of the video. There is also a training label and testing label json files. These json files contain the captions for each ID. We can read the json files as shown in Figure 4. Thus for each video id there are many alternative captions.

```
def preprocessing(self):
    """
    Preprocessing the data
    dumps values of the json file into a list
    """
    TRAIN_LABEL_PATH = os.path.join(self.train_path, 'training_label.json')
    with open(TRAIN_LABEL_PATH) as data_file:
        y_data = json.load(data_file)
    train_list = []
    vocab_list = []
```

Fig. 4. Code snapshot for data preprocessing

```
{
    "caption": [
        "A man dressed in commando clothes is firing from two guns held in both of his hands.",
        "A soldier is shooting two guns.",
        "A soldier is firing two machine guns simultaneously with his hands.",
        "A soldier shoots off two guns at once.",
        "A soldier is firing two squad automatic weapons.",
        "A soldier fires two huge machine guns at the same time.",
        "A soldier shoots two rifles simultaneously.",
        "A soldier is firing two weapons.",
        "A soldier standing in an open barren land is firing bullets holding a rifle in each hand.",
        "A soldier firing two large guns.",
        "The soldier shot his weapons.",
        "The soldier is shooting two machine guns.",
        "A soldier is firing a machine gun.",
        "The soldier fired two machine guns."
    ],
    "id": "gCra4qOrjFw_1_17.avi"
},
```


Fig. 5. Training data json file

Extracting video features:

Video Captioning is a two part project. In the first part the features of the video are extracted. As we can say a video is a list of images, so for a video in the data set each and every image called frame is extracted from the video. Since the length of videos are different, the number of frames extracted is also going to be different. So for the sake of simplicity only 80 frames are taken from each video. Each of the 80 frames is passed through a pre-trained VGG16 and 4096 features are extracted from each frame. These features are stacked to form a (80, 4096) shaped array. 80 being the number of frames and 4096 is the number of extracted features from each frame.

Features are stored in files with .npy extension

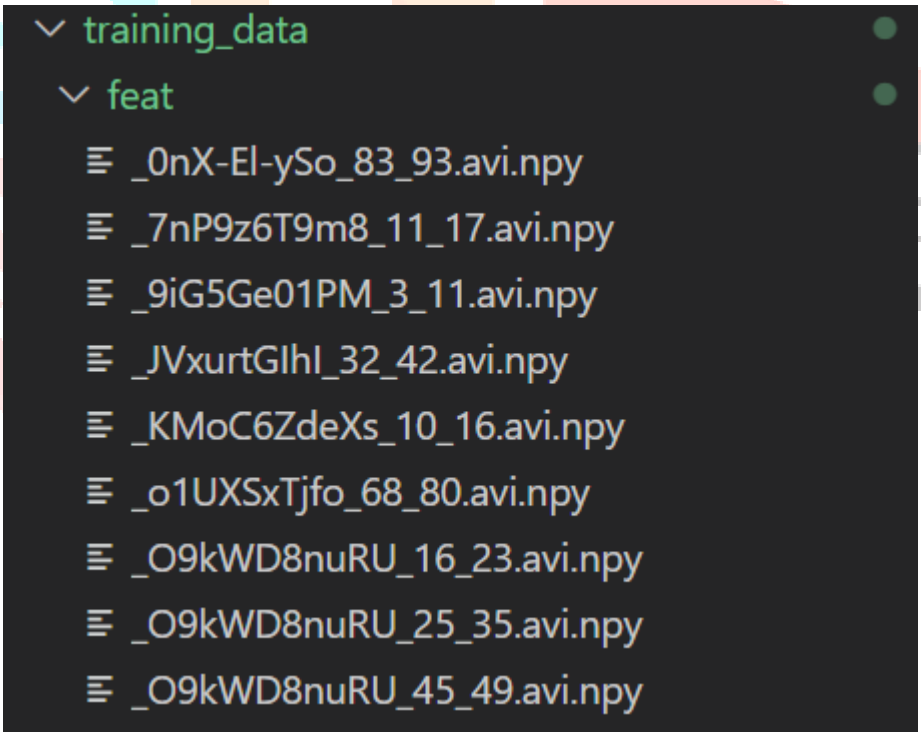

Fig. 6. Shows how features of video file are stored

E. Neural Network Architecture

Mostly for problems related to text generation, the preferred model is an encoder-decoder architecture[11]. Here in our problem statement since the text has to be generated we will also use this sequence-to-sequence architecture. One thing to know in this architecture is that the final state of the encoder cell always acts as the initial state of the decoder cell. In our problem we will use the encoder to input the video features and the decoder will be fed the captions. As earlier said a video is a sequence of images. For anything related to sequence we always prefer using RNNs or LSTMs. In our case we will use an LSTM. Now that we will use LSTM for encoder let us look into the decoder. The decoder will generate captions. Captions are basically a sequence of words so we will use LSTMs in the decoder as well.

Sequence to Sequence Model:

The model consists of 3 parts: encoder, intermediate (en- coder) vector and decoder. Here, in the Figure the features of the first frame are fed into the 1st LSTM cell of the encoder. This is followed by the features of the second frame and this goes on till the 80th frame. For this problem we are interested only in the final state of the encoder so all the other outputs from the encoder are discarded. Now the final state of the encoder LSTM acts as the initial state for the decoder LSTM. The time steps for the encoder is the number of LSTM cells we will use for the encoder which is equal to 80. Encoder tokens is the number of features from video which is 4096 in our case. Time step for the decoder is the number of LSTM cells for the decoder which is 10 and the number of tokens is the length of vocabulary which is 1500[11].

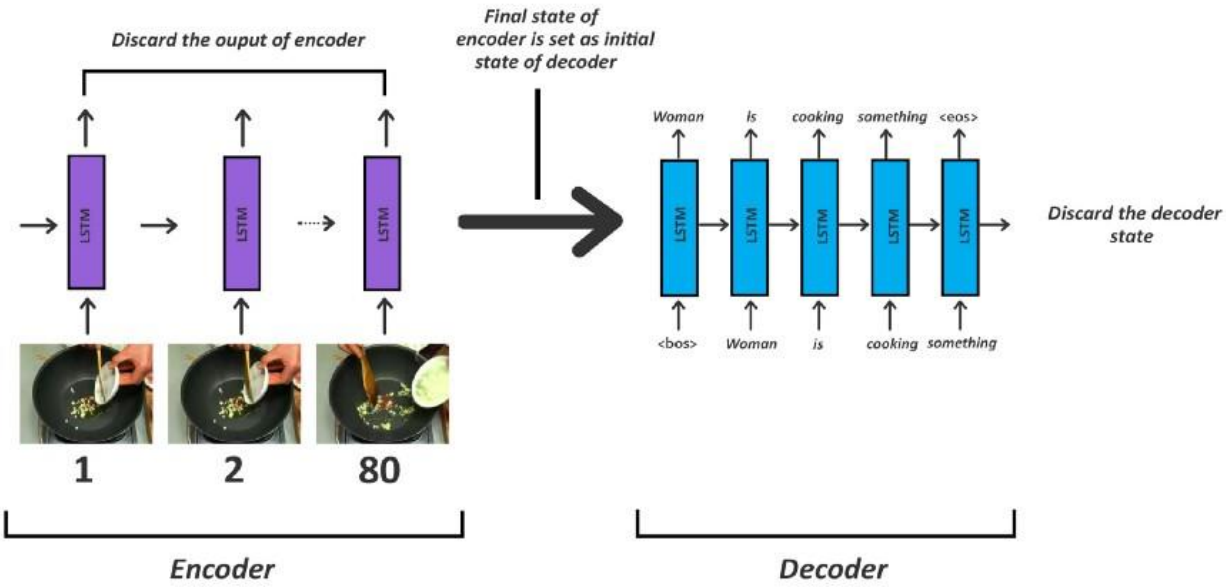

Fig. 7. Encoder Decoder model used for Training[11]

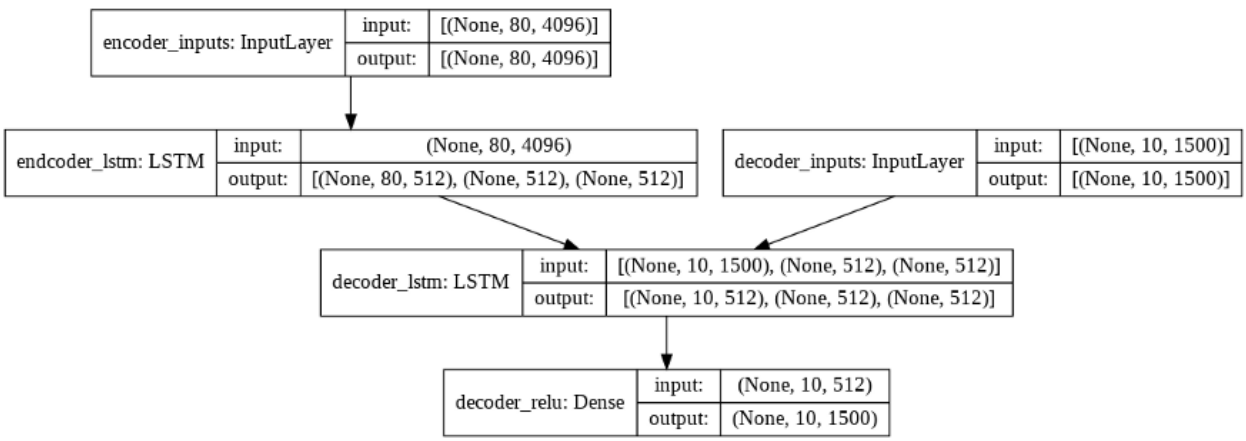

Fig. 8. Inter-layer Architecture of RNN[11]

F. Data Encryption

The captions generated after the processing of extracted video data features in the Sequence to Sequence model are stored in a file called 'results.csv' . We have also implemented a feature to encrypt the generated captions to make this application more secure. So, in the application the results.csv file is encrypted with the help of the Fernet (symmetric encryption) using Cryptography module in Python. The fernet module of the cryptography package has inbuilt functions for the generation of the key, encryption of plaintext into ciphertext, and decryption of ciphertext into plaintext. For this purpose, we have the encrypt and decrypt methods.

G. Application interface

On opening the application, the user will see the homepage. Here, the user will upload the video file. In the application, the user can also stream real time videos and get captions simultaneously , So the user can choose this option as well.

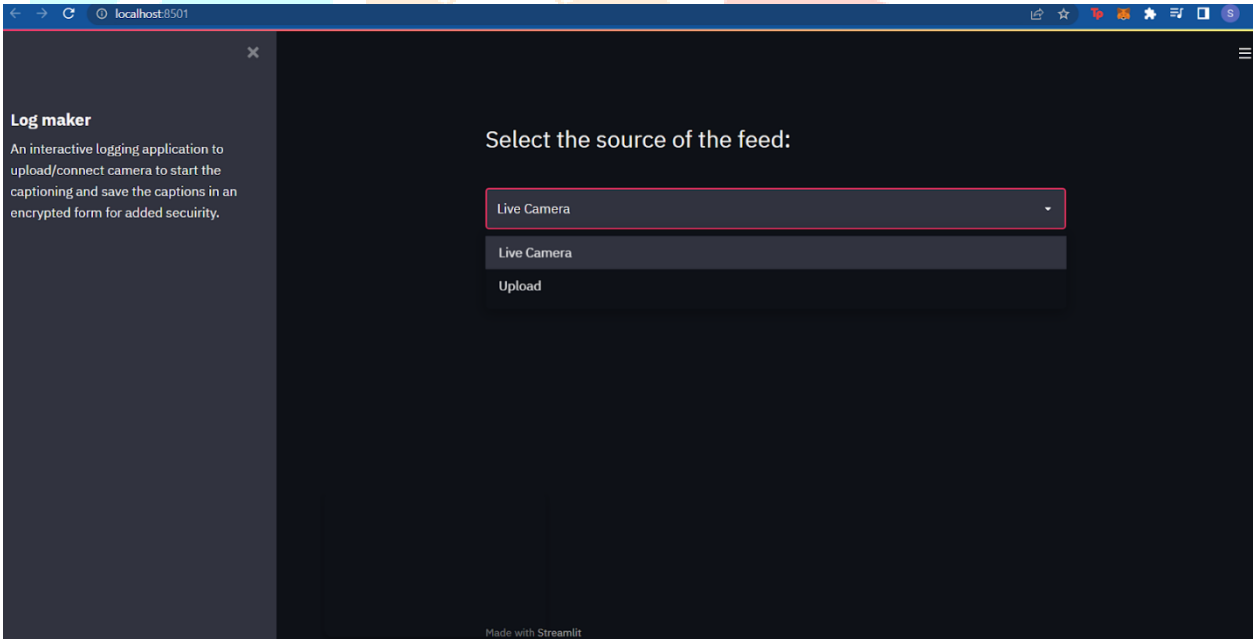

Fig. 9. Homepage to upload video files

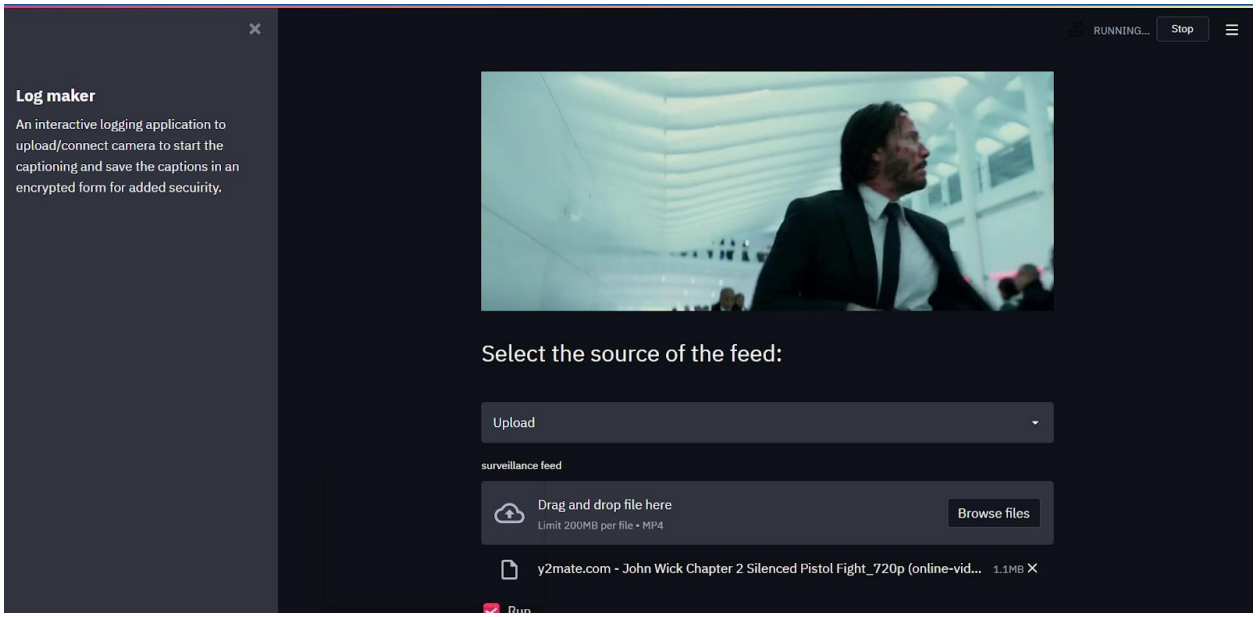

Fig. 10. Generated captions

As shown in Figure 10, we can see the captions generated in the terminal as we run the application. These captions are stored in a csv file which is later used in the search feature for searching keywords in the file. If any suspicious activity is detected through the cctv cameras or in the video file, then a notification will be shown in the homepage as shown in Figure 10. The notification shows the timestamp of the event.

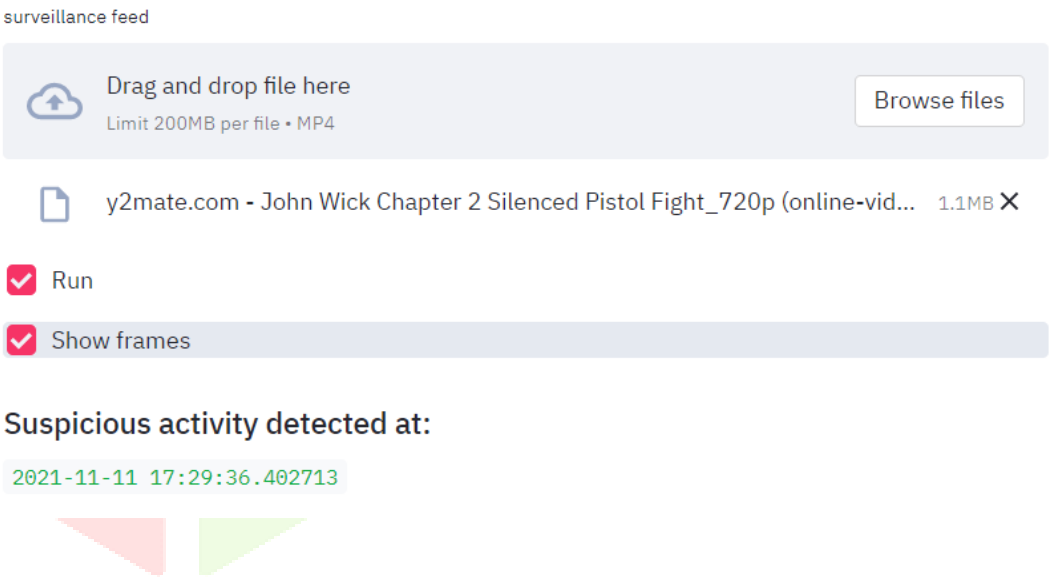

Fig. 11. Detects criminal activity and gives response on UI

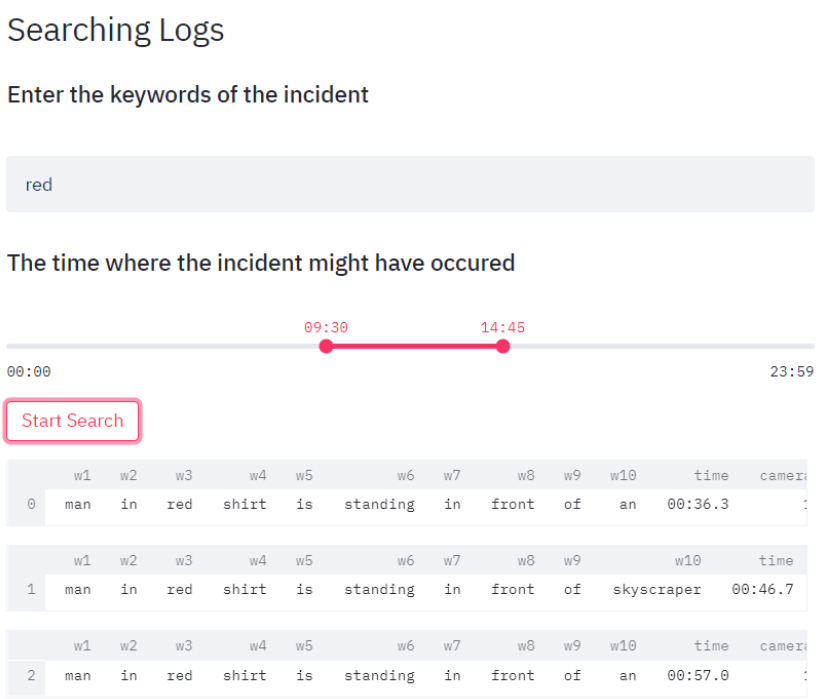

Fig. 12. Search Feature

In the search feature, the user has to load the caption file that was generated earlier as shown in Figure 12. Now, if the user needs to search for a specific event , person, thing that went missing etc, then the user has to enter the keyword associated, and as a result, the application will show the timestamp of the event occurred by searching for the keyword in the caption file.

VI. INFERENCE

For this project, we have used the MSVD data set by Microsoft and UCF crime dataset[6]. Our data set contains 1450 short YouTube clips that have been manually labelled for training and 100 videos for testing. Each video has been assigned a unique ID and each ID has about 15–20 captions. While the UCF Dataset has 1900 long videos from which we chose some according to the scenario , for e.g. arson , assault ,etc. The OpenCv library was used to capture frames from the video files. For the sake of simplicity, we extracted 80 frames from the training data and passed through the VGG16 model. Although the VGG16 model was slow to train, it is one of the best performing pre-trained model and gave the least classification error. The encoder-decoder sequence to sequence model was used for captioning of video data. The main purpose and advantage of using this model was that the inputs and outputs are not correlated and also their lengths can differ. As you can see from the explanation from above that the input data consisted of extracted features from the images (video frames) and the output was captions, which were of variable length.

VII. CONCLUSION AND FUTURE SCOPE

The system is capable of producing captions in real time. The Proposed Crime Intention Detection System is an automated system which controls occurrence of crimes by detecting gun and knife in hands of a person using VGGNET 16 pre-trained model and LSTM . If a person is detected with weapons like gun and knife then our system sends the ‘crime intention’ detection security alert message along with the time frame of when the suspicious activity occurred on the user interface. Proposed system gives good results compared to other existing approaches for crime detection. The Designed

Fig. 13. Evaluation of Model Accuracy in last 5 epochs

Fig. 14. Comparison of Train- Test loss

Fig. 15. Comparison of Train- Test accuracy

Crime Intention Detection System features can be embedded to CCTV to detect the crime scenes like detection of gun, knife in hands of persons, if crime is to occur the added feature makes the CCTV to automatically send the crime intention security messages to registered number. Although the system implemented is an optimal solution to the current established problem, there is a ton of room for improving the performance and managing multi-user interaction onto the system. Following are the future goals :

• Adding attention blocks and pre-trained embeddings like gloves so that the model understands sentences better.

Fig. 16. encoder-decoder model

• Using other pretrained models to extract features specially ones made for understanding videos like I3D

• Visualization of crime hotspots like banks , ATMs ,jewellers, etc using multiple CCTV footage at the same time.

REFERENCES

[1] Amaresh M, Chitrakala S. (2019). Video Captioning using Deep Learning: An Overview of Methods, Datasets and Metrics. 2019 International Conference on Communication and Signal Processing (ICCSP).

[2] Navalgund, U. V. K, P. (2018). Crime Intention Detection System Using Deep Learning. 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSD).

[3] Lin, Y.-L., Chen, T.-Y., Yu, L.-C. (2017). Using Machine Learning to Assist Crime Prevention. 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI).

[4] Harsh Jain, Aditya Vikram, Mohana, Ankit Kashyap, Ayush Jain.Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications.International Conference on Electronics and Sustainable Communication Systems (ICESC 2020).

[5] Tanjila Naurin, S., Saha, A., Akter, K., Ahmed, S. (2020). A Proposed Architecture to Suspect and Trace Criminal Activity Using Surveillance Cameras. 2020 IEEE Region 10 Symposium.

[6] Real-world Anomaly Detection in Surveillance Videos https://www.crcv.ucf.edu/projects/real-world/

[7] A Comprehensive Guide to Convolutional Neural Networks the ELI5 way https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[8] Recurrent neural network https://en.wikipedia.org

[9] How to use a pre-trained model (VGG) for image classification https://towardsdatascience.com/how-to-use-a-pre-trained-model-vgg-for-image-classification-8dd7c4a4a517

[10] A Gentle Introduction to Long Short-Term Memory Networks by the Experts https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts

[11] Video Captioning with Keras https://medium.com/analytics-vidhya/video-captioning-with-keras-511984a2cfff For this study secondary data has been collected. From the website of KSE the monthly stock prices for the sample firms are obtained from Jan 2010 to Dec 2014. And from the website of SBP the data for the macroeconomic variables are collected for the period of five years. The time series monthly data is collected on stock prices for sample firms and relative macroeconomic variables for the period of 5 years. The data collection period is ranging from January 2010 to Dec 2014. Monthly prices of KSE -100 Index is taken from yahoo finance.