Veil - The Smart Surveillance Solution

Divya S
Dept. of Computer Science
PES University
Bengaluru, India

***Abstract*—In this paper, we attempt to build a smart video surveillance solution based on a Machine Learning algorithm to identify abnormal activities in CCTV surveillance and alert authorities so as to prevent escalation of violent scenes and theft. We propose a model that runs an algorithm on surveillance footage as a batch of video clippings to monitor loss values of footage frames to trigger the appropriate responses when encountering frames with unusual activities. The concerned authorities are intimated with and without requirement of the internet to ensure immediate action.**

Anomalous events detection in real-world video scenes is a challenging problem due to the complexity of "anomaly" as well as the cluttered backgrounds, objects and motions in the scenes. Most existing methods use hand-crafted features in local spatial regions to identify anomalies. In this paper, we propose a model called Spatio-Temporal AutoEncoder (ST AutoEncoder or STAE), which utilizes deep neural networks to learn video representation automatically and extracts features from both spatial and temporal dimensions by performing 3-dimensional convolutions.

In addition to the reconstruction loss used in existing typical autoencoders, we introduce a weight-decreasing prediction loss for generating future frames, which enhances the motion feature learning in videos. Since most anomaly detection datasets are restricted to appearance anomalies or unnatural motion anomalies, we collected a new challenging dataset comprising a set of real-world traffic surveillance videos. Several experiments are performed on both the public benchmarks and our traffic dataset, which show that our proposed method remarkably outperforms the state-of-the-art approaches.

Keywords—surveillance, machine learning, AI model, safety, CCTV cameras, footage

I. INTRODUCTION

Showing a 1.6% annual increase in the registration of cases (50,74,635 cases), the crime rate per 100,000 population has increased from 383.5 in 2018 to 385.5 in 2019. More than a fifth of all registered crime (10,50,945) were classified as offences affecting the human body, which included violent acts such as murder, kidnapping, assault and death by negligence.

Many of these crimes and acts of violence take place in broad daylight in under-staffed or unmonitored places. It is highly impractical for organizations of all scales to hire personnel to patrol the building and neighborhood, or even stay put in a surveillance room watching CCTV footage for hours on end every day.

Video Surveillance has become an indispensable component for ensuring public safety in the modern world. Sophisticated video object tracking techniques specially designed for surveillance applications are of increasing importance for analyzing and understanding numerous surveillance videos in an effective manner.

A large majority of video surveillance applications are concerned with monitoring activities within structured environments, such as indoor environments, surrounding areas of buildings, highways, traffic junctions, etc., the structures of which are often static and known to the surveillance personnel. One important characteristic of moving objects in these applications is that the motions of objects are constrained by the structure of the environment under surveillance.

In this project, we present a machine-learning algorithm based on spatio-temporal autoencoders, utilizing long short term memory, that identifies the abnormal events by computing the reconstruction loss using Euclidean distance between original and reconstructed batch.

Videos can be understood as a series of individual images and therefore, many deep learning practitioners would be quick to treat video classification as performing image classification a total of N times, where N is the total number of frames in a video. The problem with this approach is that video classification is more than just simple image classification — with video we can typically make the assumption that subsequent frames in a video are correlated with respect to their semantic contents.

If we are able to take advantage of the temporal nature of videos, we can improve our actual video classification results. Neural network architectures such as Long short-term memory (LSTMs) and Recurrent Neural Networks (RNNs) are suited for such time series data.

II. EASE OF USE

A. *Easy Setup*

This program structure when packaged into a product requires the cctv camera footage simply be directed as input to the algorithm. No formatting of data is required as it is included in the product. The algorithm can build a usable model with low resolution images and test live streaming data with this specific model. In this manner, the model performance does not deteriorate drastically over video quality of the surveillance camera.

B. *Personalized Model Training*

This model is trained specifically for each camera setup. This reduces false positive classification of video frames, thereby increasing accuracy and avoiding false alarms.

The user also has the option to alter the threshold for losses so as to personalize the model to the location, angle and atmosphere of the camera. This enables the model to adapt to various circumstances using input data from the very same CCTV camera. The footage input can be taken in several ways. It can be taken via Amazon Kinesis Streaming to a cluster and the algorithm can be run on an EMR cluster with data on S3, or alternatively, the data can be taken in from the video display already set up for CCTV camera surveillance. This is achieved locally.

C. *Effective Alert System*

This product triggers three different actions when an abnormal activity is detected. Consumers using the product will be requested to create an account on Veil's website and mobile application, through which they can be notified with an image of the site of abnormality in real time. The user also gets a message on social media, namely WhatsApp, and on SMS.

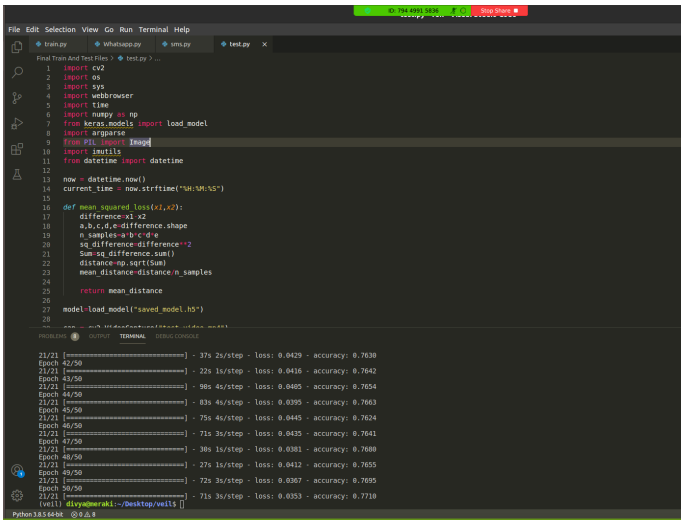
III. TECHNOLOGY STACK

A. *Software*

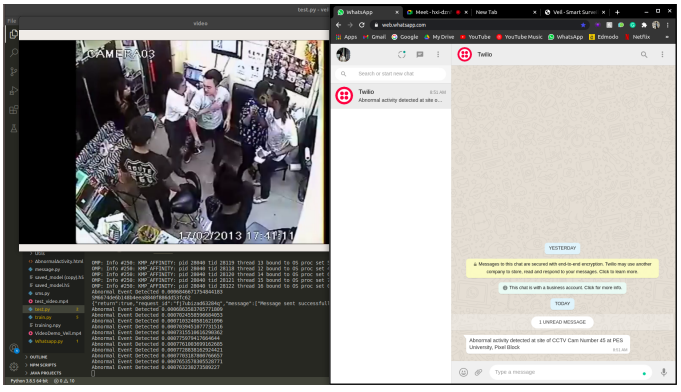
The software is split into two parts - Training and Testing. The training model uses libraries such as Keras, Tensorflow, Numpy, OpenCV etc. It first takes the input video files from the training dataset and breaks each video down to integral frames at the rate of 26 fps and stores these frames in a folder. These images are then resized and normalized. Next, a sequential model is built with 7 layers - two 3D convolutional layers with 128 and 64 filters respectively, three convolutional 2D LSTM layers with 64, 32, and 64 layers respectively, and two 3D convolutional transpose layers with 128 filters and 1 filter respectively. The most constricted layer is termed the "bottleneck layer".

The model is compiled using an Adam Optimizer, using Mean Squared Loss and Accuracy metrics. As a demonstrative model, we have trained this model for 60 epochs, achieving 81.7% accuracy. Once trained, this model is saved to test.
With extensive training data and increased iterations, this accuracy will improve proportionately.

Testing uses the OpenCV library pyQT to utilise video display to play the surveillance footage on an independent window which immediately pops up when testing begins. Frames are captured serially and resized and processed to grayscale. Then the reconstruction loss is calculated for each frame, and should a frame exceed the threshold loss, it is classified as abnormal.



Training the model for 50 epochs



Training the model and depiction of WhatsApp trigger

Actions triggered on detecting abnormality:

1. WhatsApp Message:

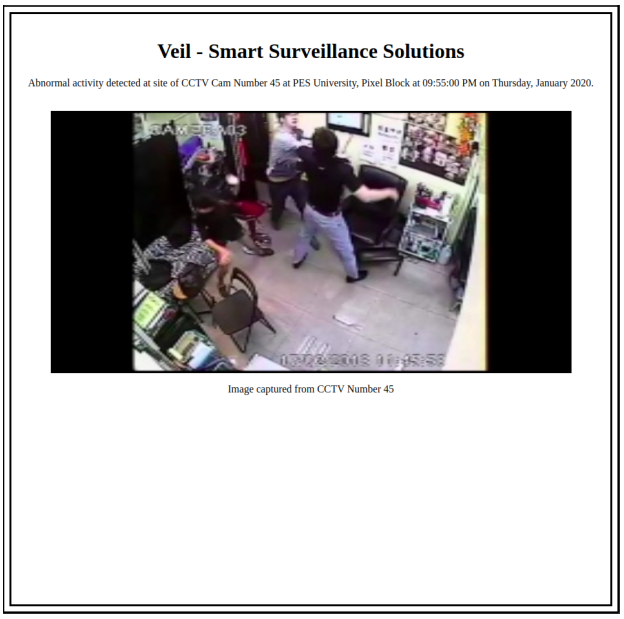
A third party API - Twilio has been registered with to orchestrate information passing from the testing module to the consumer. This app requires two keys - account_sid and auth_token which are provided on the Twilio website for registered users. The message sent to the consumer is in the format that conveys the site of abnormality, and datetime stamp.

2. SMS:

Third party API Fast2sms is used to send SMS to single or multiple users with the alert message. Product admin registers with this service and adds recipients contact information post encryption.

3. An HTML page:

A screenshot of the abnormal frame is displayed on an HTML page, indicating the site of the event and the CCTV camera number that captures the irregularity.



HTML pop-up page sample

III. NETWORK ARCHITECTURE

This project extends to deep neural networks to 3-dimensional for learning spatio-temporal features of the video feed.

Here we introduce a spatio temporal autoencoder, which is based on a 3D convolution network. The encoder part extracts the spatial and temporal information, and then the decoder reconstructs the frames. The abnormal events are identified by computing the reconstruction loss using Euclidean distance between original and reconstructed batch.

Choice of Optimizer:

For this model we have used the Adam optimizer. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.

According to Kingma et al 2014, the method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters".

A. Keras Dependencies

- Sequential API

A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. It allows us to create models layer-by-layer for most problems.

- ModelCheckpoint Callback

It is used in conjunction with training using model.fit() to save a model or weights (in a checkpoint file) at some interval, so the model or weights can be loaded later to continue the training from the state saved.

- EarlyStopping

A method that allows us to specify an arbitrary large number of training epochs and stop training once the model performance stops improving on a hold out validation dataset.

B. Model specifications

1. Layers

The model comprises 7 layers in total, with one input layer, one output layer and 5 hidden layers.

- Two 3-D Convolutional layers
- Three convolutional 2-D LSTM layers
- Two 3-D convolutional transpose layers

2. Frame per Second(FPS): 26 FPS

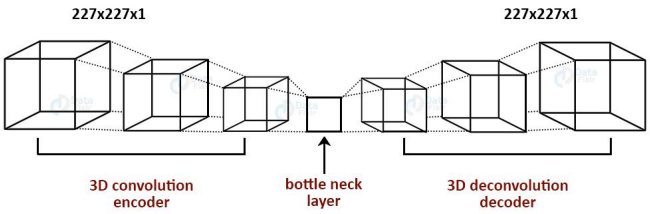
For smooth video quality, the frame rate is usually 20 FPS or more. However, due to the subtle lag in our training dataset, our training model scans the frames at 26 FPS.

3. Activation Function: Tanh

Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated (“fired”) or not, based on whether each neuron's input is relevant for the model's prediction.

Activation function used in our model is tanh, which is a shifted version of the sigmoid function. It almost always performs better than the sigmoid function.

While a sigmoid function will map input values to be between 0 and 1, a tanh function ensures that the values stay between -1 and 1, thus regulating the output of the neural network.



4. Loss Function: Mean Squared Error (MSE)

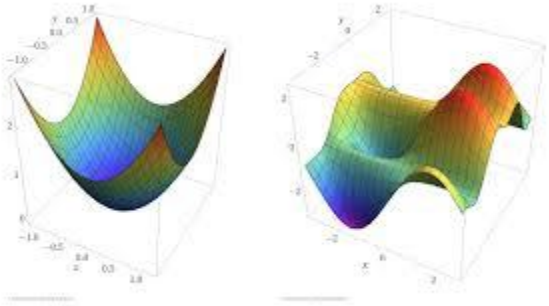
Loss is a prediction error of the Neural Net, while the method to calculate the loss is called Loss function.

Loss is used to calculate the gradients, which in turn are used to update the weights of the Neural Network.

Our model uses the mean squared error loss function that is responsible for calculating the loss between subsequent frames and is an essential part of our model-building and enhancement.

5. Optimizer: Adam

Adam is a stochastic gradient descent algorithm method that computes individual adaptive learning rates for different parameters from estimates of first- and second-order moments of the gradients.



Adam optimization on sparse tensors

C. Key Takeaways

- Building a machine-learning model highly relies on the dataset we procure.
- It is important to fathom real-time usage of the products we build, and work towards keeping them relevant, reliable and practical.
- AI and Machine Learning are here to stay, and these approaches will be the answer to bigger problems in the future.

D. Further Work

- Host the HTML page on a server so that it can be viewed by the authorized user.
- Upload the screen capture of the abnormal event on our dedicated website and directly communicate alerts to customers.
- A simple Flutter application to notify users instead of using third-party API.
- Install it in our University campus and stream real-time footage from the CCTV cameras.

E. Common Mistakes

Some common mistakes we have come across during our research on this project’s previous work are choosing irregular datasets, overfitting or overtraining the ML models, or choosing sub-optimal values of fps, loss threshold etc. These errors can cause the model to omit critical data and as a result, fail to classify important scenes correctly as abnormal or otherwise. This, we have found, can be severely catastrophic in some cases. Although the project is not 100% reliable, it highly depends on adopting appropriate values for tuning variables.

IV. Conclusions and Inferences

This paper presented a Deep Learning solution to the growing advancements in video surveillance. Recent Deep Learning methodologies for the detection of humans and tracking deserved a dedicated state-of-the-art survey. The problem of behavior analysis has been approached with different methods. The reviewed works highlight the approach to incorporating futuristic methods for detecting abnormal activities, violence and theft.

Deep learning together with classical Machine Learning models has high levels of accuracy. This method requires less computation time with respect to the existing features and classification.

A. Abbreviations and Acronyms

- STAE - Spatio-Temporal AutoEncoder
- LSTM - Long Short Term Memory
- RNN - Recurrent Neural Network
- CNN - Convoluted Neural Network
- API - Application Programming Interface
- AWS - Amazon Web Services
- Tanh - Hyperbolic Tangent Function
- FPS - Frames Per Second
- CCTV - Closed Circuit Television
- MSE - Mean-Squared Error
- SMS - Short Message Service
- HTML - HyperText Markup Language

REFERENCES

[1] https://ieeexplore.ieee.org/document/8995084
[2] https://ieeexplore.ieee.org/document/8827932