OPERATING SYSTEM PROJECT

VIDEO SURVEILLANCE

Yash Rajoria: 2020UEA6570
Vaibhav Paliwal: 2020UEA6571
Rohit Kumar: 2020UEA6573
Ritik: 2020UEA6575

Submitted to:- Somiya Rani

Table of Contents:

ABSTRACT

Nowadays, as CCTV cameras are placed all over the places for surveillance and security. But surveillance security is a very tedious and time-consuming job. We have people watching over various CCTV cameras, but even they might miss some things due to human error and slow reaction time.
To improve upon this as the number of cameras is increasing, our objective is to make this process fast and automatic.

We present Slow Fast networks for video recognition. Our model involves
(i) A Slow pathway, operating at low frame rate, to capture spatial semantics, and
(ii) A Fast pathway, operating at high frame rate, to capture motion at fine temporal resolution.
The Fast pathway can be made very lightweight by reducing its channel capacity yet can learn useful temporal information for video recognition. Our models achieve strong performance for both action classification and detection in video, and large improvements are pinpointed as contributions by our Slow Fast concept.

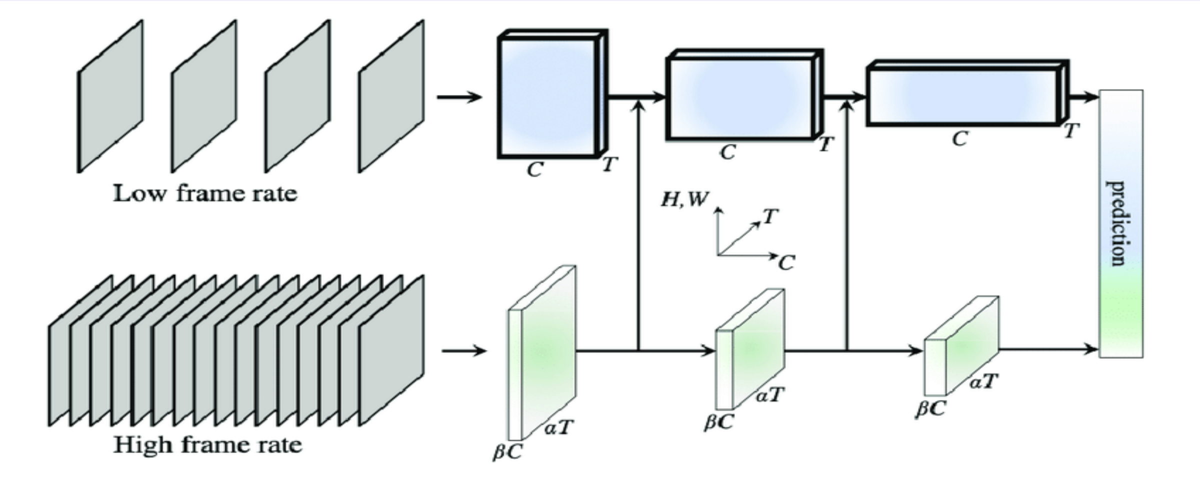
INTRODUCTION

Detecting objects in images and categorizing them is one of the more well-known Computer Vision tasks, popularized by the 2010 ImageNet dataset and challenge. While much progress has been achieved on ImageNet, a still vexing task is video understanding — analysing a video segment and explaining what's happening inside of it. Despite some recent progress on solving video understanding, contemporary algorithms are still far from human-level results.

In this project we try to work on the data collected by the surveillance systems and classify them into various classes depending upon what is happening in the videos.

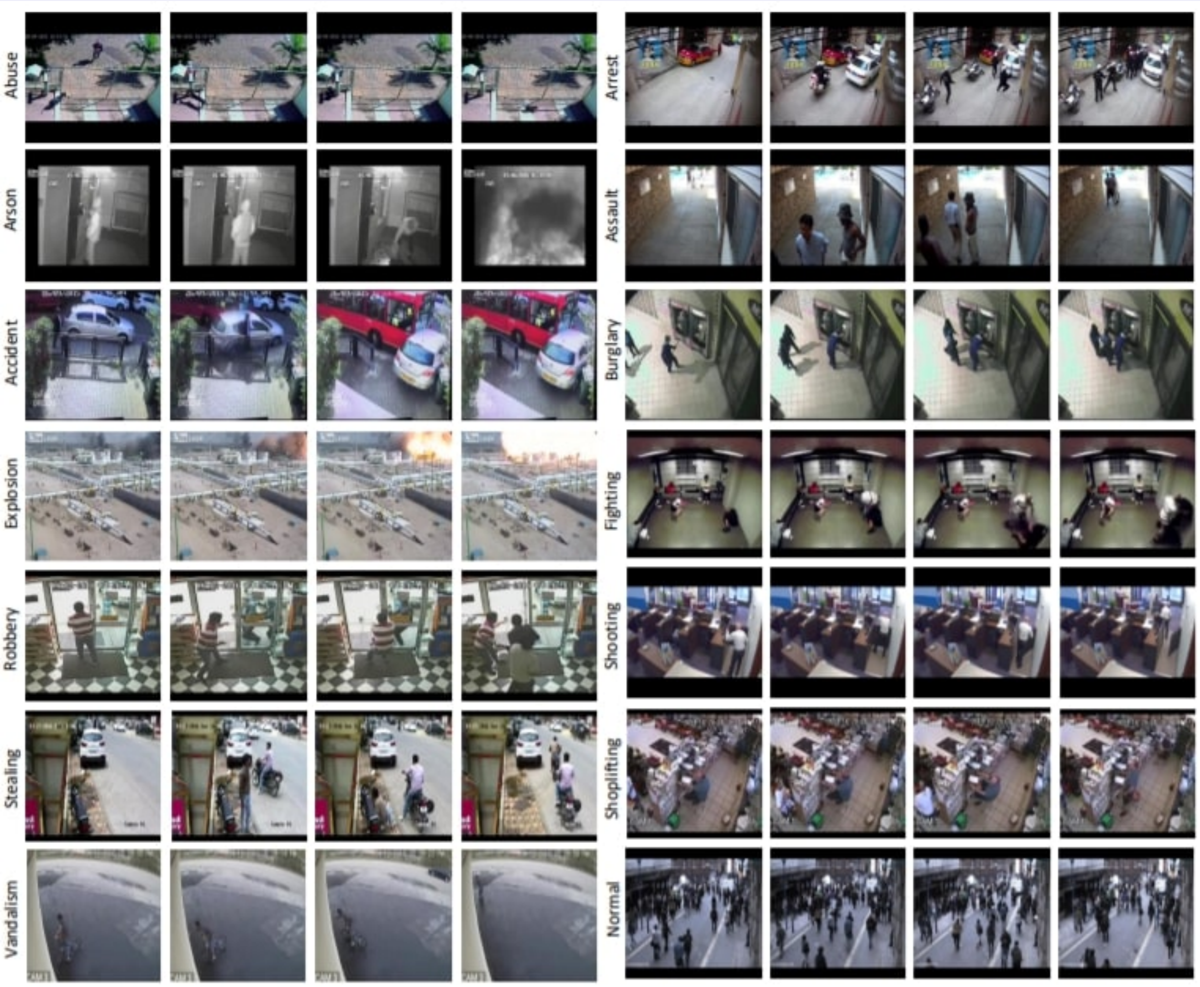
We use SlowFast which presents a novel method to analyse the contents of a video segment. At the heart of the method is the use of two parallel convolution neural networks (CNNs) on the same video segment — a Slow pathway and a Fast pathway.

Slow Fast networks



SlowFast uses a slow, high-definition CNN (Fast pathway) to analyse the static content of a video while running in parallel a fast, low-definition CNN (Slow pathway) whose goal is to analyse the dynamic content of a video. The technique is partially inspired by the retinal ganglion in primates, in which 80% of the cells (P-cells) operate at low temporal frequency and recognize fine details, and ~20% of the cells (M-cells) operate at high temporal frequency and are responsive to swift changes. Similarly, in SlowFast the compute cost of the Slow pathway is 4x larger than that of the Fast pathway.

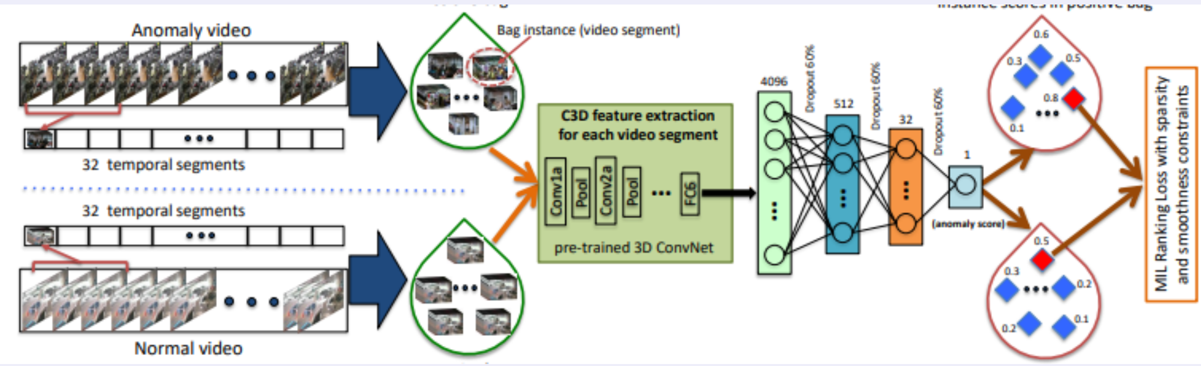
Dataset



We need a huge amount of video data for training our model. Thus, we are using UCF-Crime dataset.

The UCF-Crime dataset is a large-scale dataset of 128 hours of videos. It consists of 1900 long and untrimmed real-world surveillance videos, with 13 realistic anomalies including Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. These anomalies are selected because they have a significant impact on public safety.

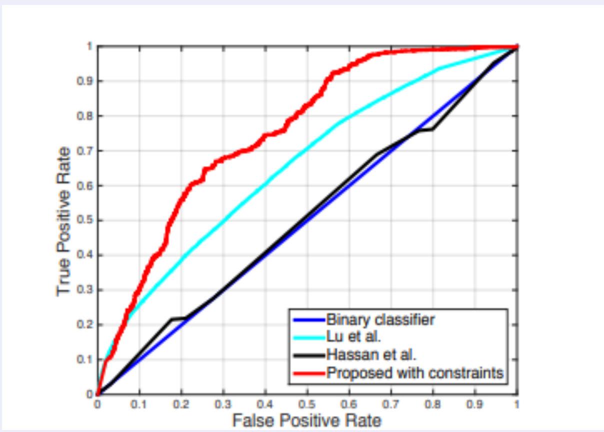
Methodology



We first load data from dataset, divide video data by using Slow Fast Network, then train it using Neural networks.

Given the positive (containing anomaly somewhere) and negative (containing no anomaly) videos, we divide each of them into multiple temporal video segments. After extracting Conv3D features for video segments, we train a fully connected neural network by utilizing a categorical cross-entropy loss function which computes the entropy loss between the highest scored instances.

After minimising the loss function, we evaluate the model using Region Operating Characteristics (ROC).



MODELS

Overall, we are using SlowFast model, along with a combination of 3D Convolution Neural network layers and Dense hidden layers.

SlowFast model: Slow and Fast pathways use a 3D ResNet model, capturing several frames at once and running 3D convolution operations on them.

3D CNN Layers: A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

The convolutional layers are the key component of a CNN, where filters are applied to the input image to extract features such as edges, textures, and shapes. The output of the convolutional layers is then passed through pooling layers, which are used to down-sample the feature maps, reducing the spatial dimensions while retaining the most important information.

Dense Layers: The output of the pooling layers is then passed through one or more fully connected layers, which are used to make a prediction or classify the image.

Conventional model summary:

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1 (Conv3D)              (None, 32, 160, 160, 16)  1312

 pool1 (MaxPooling3D)        (None, 32, 80, 80, 16)    0

 conv2 (Conv3D)              (None, 32, 80, 80, 16)    6928

 pool2 (MaxPooling3D)        (None, 16, 40, 40, 16)    0

 conv3a (Conv3D)             (None, 16, 40, 40, 32)    13856

 conv3b (Conv3D)             (None, 16, 40, 40, 32)    27680

 pool3 (MaxPooling3D)        (None, 8, 20, 20, 32)     0

 zero_padding3d (ZeroPadding  (None, 8, 22, 22, 32)    0
 3D)

 pool5 (MaxPooling3D)        (None, 4, 11, 11, 32)     0

 flatten (Flatten)           (None, 15488)             0

 fc6 (Dense)                 (None, 1024)              15860736

 dropout (Dropout)           (None, 1024)              0

 fc8 (Dense)                 (None, 1024)              1049600

 dropout_1 (Dropout)         (None, 1024)              0

 dense (Dense)               (None, 8)                 8200

=================================================================
Total params: 16,968,312
Trainable params: 16,968,312
Non-trainable params: 0
```

SlowFast model summary:

```
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 32, 160, 160 0
__________________________________________________________________________________________________
input_1 (InputLayer)            [(None, 4, 160, 160, 0
__________________________________________________________________________________________________
conv1 (Conv3D)                  (None, 32, 160, 160, 1312        input_2[0][0]
__________________________________________________________________________________________________
conv1s (Conv3D)                 (None, 4, 160, 160,  1312        input_1[0][0]
__________________________________________________________________________________________________
pool1 (MaxPooling3D)            (None, 32, 80, 80, 1 0           conv1[0][0]
__________________________________________________________________________________________________
pool1s (MaxPooling3D)           (None, 4, 80, 80, 16 0           conv1s[0][0]
__________________________________________________________________________________________________
conv2 (Conv3D)                  (None, 32, 80, 80, 1 6928        pool1[0][0]
__________________________________________________________________________________________________
conv2s (Conv3D)                 (None, 4, 80, 80, 16 6928        pool1s[0][0]
__________________________________________________________________________________________________
pool2 (MaxPooling3D)            (None, 16, 40, 40, 1 0           conv2[0][0]
__________________________________________________________________________________________________
pool2s (MaxPooling3D)           (None, 3, 40, 40, 16 0           conv2s[0][0]
__________________________________________________________________________________________________
conv3a (Conv3D)                 (None, 16, 40, 40, 3 13856       pool2[0][0]
__________________________________________________________________________________________________
conv3as (Conv3D)                (None, 3, 40, 40, 32 13856       pool2s[0][0]
__________________________________________________________________________________________________
pool3 (MaxPooling3D)            (None, 8, 20, 20, 32 0           conv3a[0][0]
__________________________________________________________________________________________________
pool3s (MaxPooling3D)           (None, 1, 20, 20, 32 0           conv3as[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 9, 20, 20, 32 0           pool3[0][0]
                                                                 pool3s[0][0]
__________________________________________________________________________________________________
zero_padding3d (ZeroPadding3D)  (None, 9, 22, 22, 32 0           concatenate[0][0]
__________________________________________________________________________________________________
pool5 (MaxPooling3D)            (None, 4, 11, 11, 32 0           zero_padding3d[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 15488)        0           pool5[0][0]
__________________________________________________________________________________________________
fc6 (Dense)                     (None, 1024)         15860736    flatten[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 1024)         0           fc6[0][0]
__________________________________________________________________________________________________
fc8 (Dense)                     (None, 1024)         1049600     dropout[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 1024)         0           fc8[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 8)            8200        dropout_1[0][0]
==================================================================================================
Total params: 16,962,728
Trainable params: 16,962,728
Non-trainable params: 0
```

IDE USED

- Jupyter Notebook- Free software, open standards, and web services for interactive computing across all programming languages.
- Spyder- Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts.

Predictions


```
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:34.633894: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:34.729791: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:34.826217: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:34.923927: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:35.022734: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:35.117980: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:35.215015: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
Assault
2023-04-12 00:12:35.314676: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start
aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for
placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
     [[{{node Placeholder/_0}}]]
```

```
Explosion
2023-04-12 00:16:37.670209: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:37.765526: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:37.859179: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:37.956674: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:38.048945: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:38.143671: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:38.234963: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
Explosion
2023-04-12 00:16:38.328756: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_0' with dtype float and shape [1,32,160,160,3]
	 [[{{node Placeholder/_0}}]]
```

Future Work

Right now, the model is detecting the action in the video footage which as of now is being trained on UCF Crime dataset, So to increase its accuracy more, we can implement CNN model using VGG19 instead of ResNet.

Instead of using traditional Convolution Layers, we can use Diluted or Deformable Convolution Layers.

We can further build on this model to identify the person itself.

Right now it requires high end devices to run, but later on we can reduce its resource dependencies.

Conclusion

In this project, we have successfully implemented the CNN based Video Surveillance Deep Learning Project using two completely different models. We wanted to achieve above 90% accuracy but not able to achieve that due resource constraints.