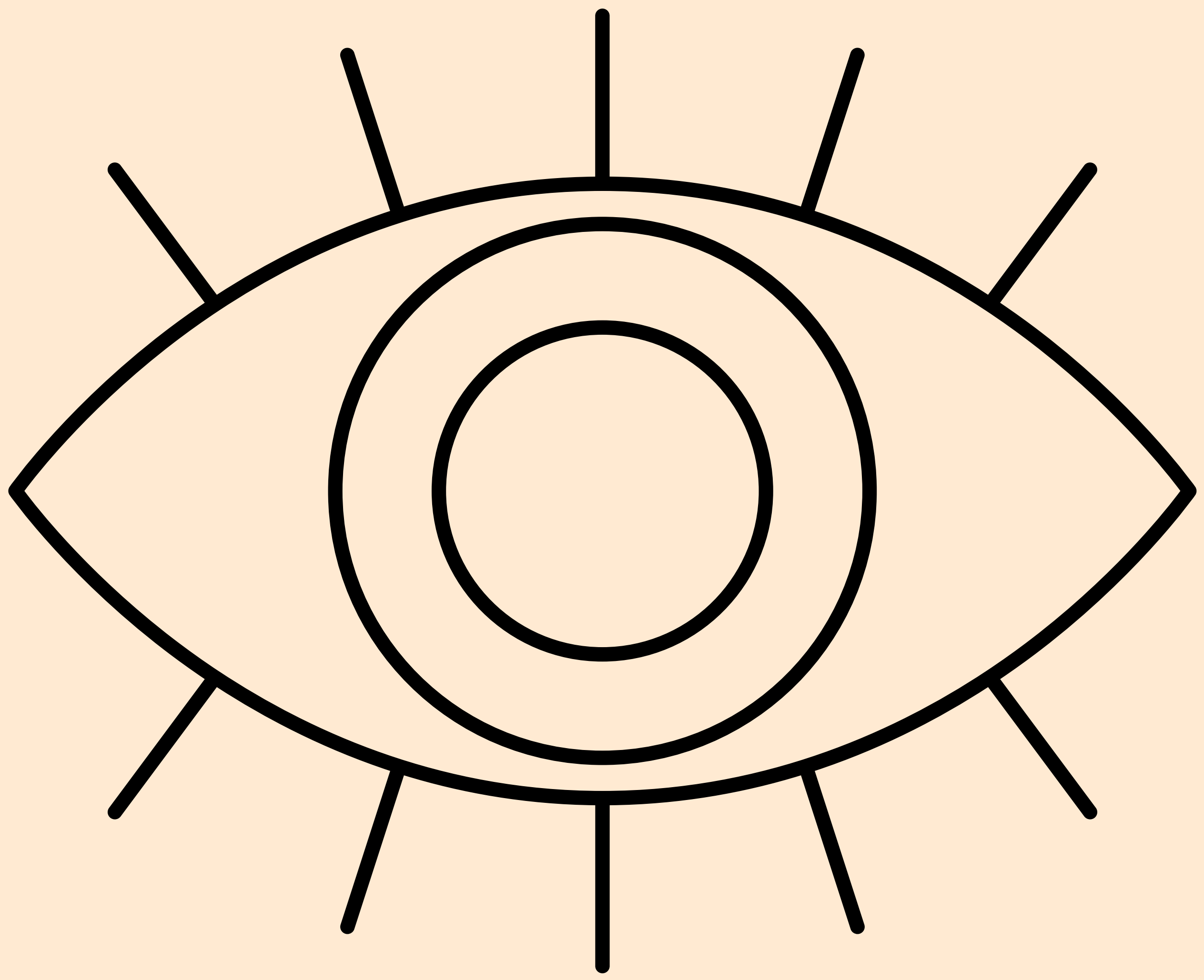
HUMAN ABNORMAL BEHAVIOUR DETECTION



Jaouhar Cherif & Hamza Faidi

2024

Content

INTRODUCTION

CONTEXT

Nowadays, security cameras are everywhere, and they can be really helpful. Sometimes, they capture abnormal scenes like fights, shootings, explosions, assaults etc ... The exciting part is, we can use automated Deep Learning algorithms to look at this footage and make fast and reliable decisions. It's like having a smart helper that can quickly analyze what's happening and help keep things safe.

IMPORTANCE OF DETECTING ABNORMAL BEHAVIOUR

PUBLIC SAFETY AND SECURITY

Identifying and responding to potential threats or suspicious activities helps prevent and mitigate security incidents.

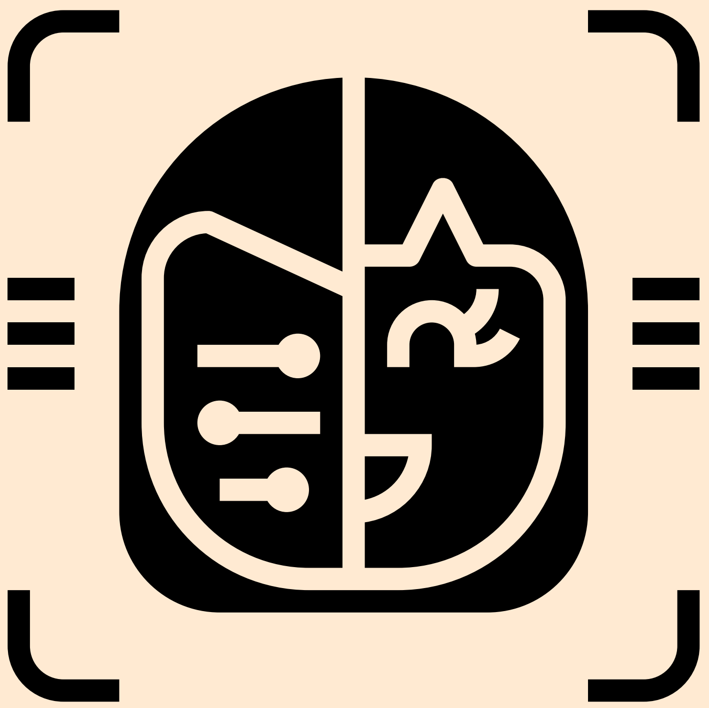
CRIME PREVENTION AND INVESTIGATION

Analyzing historical data to identify patterns of unusual activities can provide valuable insights and support law enforcement efforts.

CRITICAL INFRASTRUCTURE PROTECTION

Protecting critical infrastructure, such as power plants, transportation systems, and communication networks, relies on identifying abnormal behaviour

Leveraging the power of CV



IMPROVED ACCURACY AND PRECISION

Enhanced accuracy and precision in recognizing abnormal behaviors, reducing false positives and negatives.

BEHAVIORAL PATTERN ANALYSIS

By training models on diverse datasets, these systems can recognize subtle deviations from normal behavior, even in crowded or dynamic public environments

CONTINUOUS LEARNING AND ADAPTATION

Deep learning models can be designed for continuous learning, allowing them to adapt to evolving patterns of behavior over time.

DATA PRE-PROCESSING

DATASET

DESCRIPTION

This dataset contains videos based on the following 13 classes: Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. Each video is labeled as normal (0) or abnormal (1) according to its content.
There is a total of **16853** videos, where **9676** videos are labeled as **Normal** and **7177** as **abnormal.**
Reference

Data preprocessing

RESIZING

- Resizing the frames to a consistent size is important for deep learning models that expect fixed input dimensions

NORMALIZATION

- Normalization is a crucial preprocessing step that scales pixel values to a range between 0 and 1.
- It helps in stabilizing and accelerating the training process by ensuring that input features are within a similar numerical range.

FRAMES

- The videos were curated by selecting those with 30 frames, and for videos originally containing 60 frames, downsampling was applied by considering only every other frame. This preprocessing step was implemented to ensure uniformity in the input shape across all videos, aligning with the model's requirement for consistent input dimensions.

MODELING

Model Overview

DESCRIPTION

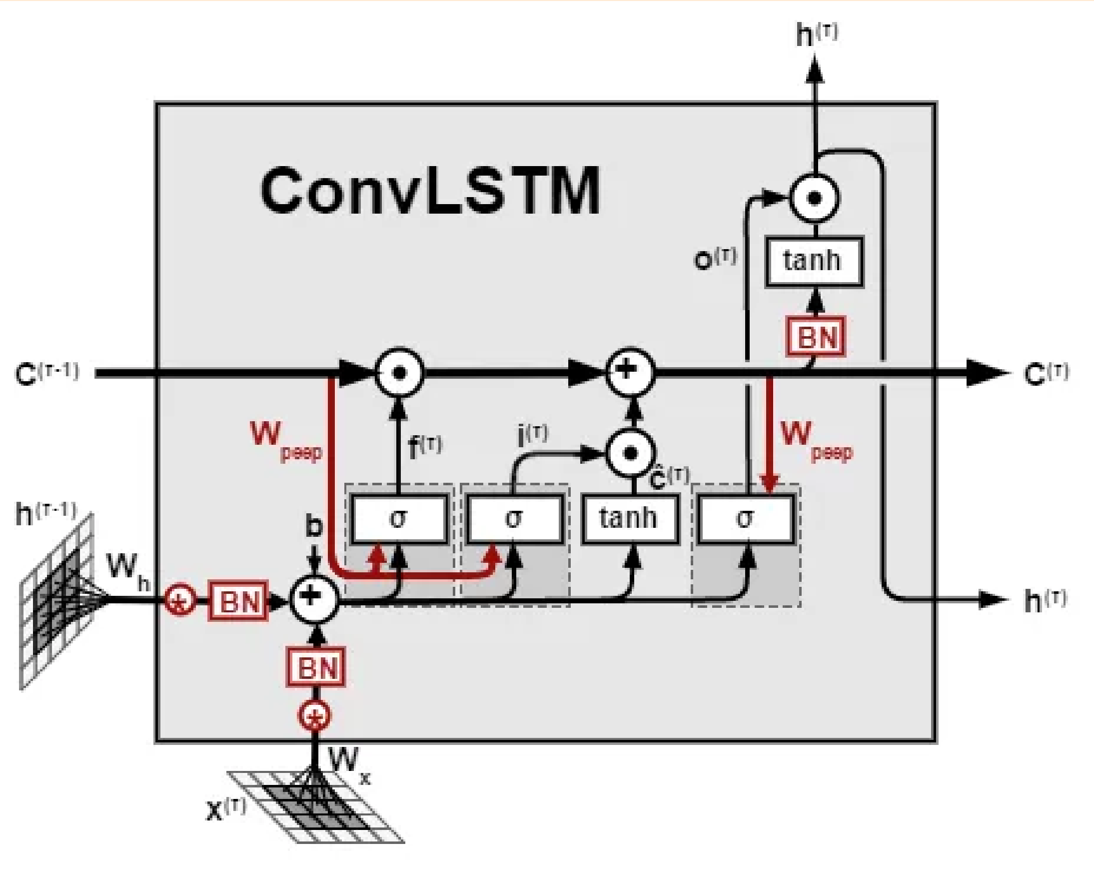
The RNN model utilizes a ConvLSTM2D layer as the input, enabling simultaneous extraction of spatiotemporal features in 3D sequences. It follows a repeated pattern of ConvLSTM2D, MaxPooling3D, and TimeDistributed layers for hierarchical pattern extraction. The model ends with a Flatten layer and a dense layer for class prediction,

ARCHITECTURE

ConvLSTM2D

DESCRIPTION

It is a Recurrent layer, just like the LSTM, but internal matrix multiplications are exchanged with convolution operations. As a result, the data that flows through the ConvLSTM2D cells keeps the input dimension (3D in our case) instead of being just a 1D vector with features.
source



A ConvLSTM cell

First Approach: Binary Classification

DESCRIPTION

In this phase, we developed a binary classification model. During training, we structured the data in the form of frames, with corresponding labels denoting **0** for **normal** behavior and **1** for **abnormal** behaviour. **Sigmoid** activation was employed in the **output layer**, facilitating binary classification. The loss function chosen for this task was **binary-cross-entropy**, which issuited for scenarios where the objective is to classify instances into two distinct classes.

NORMAL

VS

ABNORMAL

Moving to Multiclass Classification

DESCRIPTION

- In the second multiclass classification approach, we retained the same model architecture.
- We incorporated a **one-hot encoder** to map class names to unique integer identifiers.
- The input data for the model is now a sequence of image frames, each labeled with a unique integer representing the class.
- The activation function in the output layer was changed to **softmax** to accommodate multiclass classification.
- The loss function was updated to **categorical-cross-entropy** to align with the classification task.

ASSAULT

ARREST

ACCIDENT

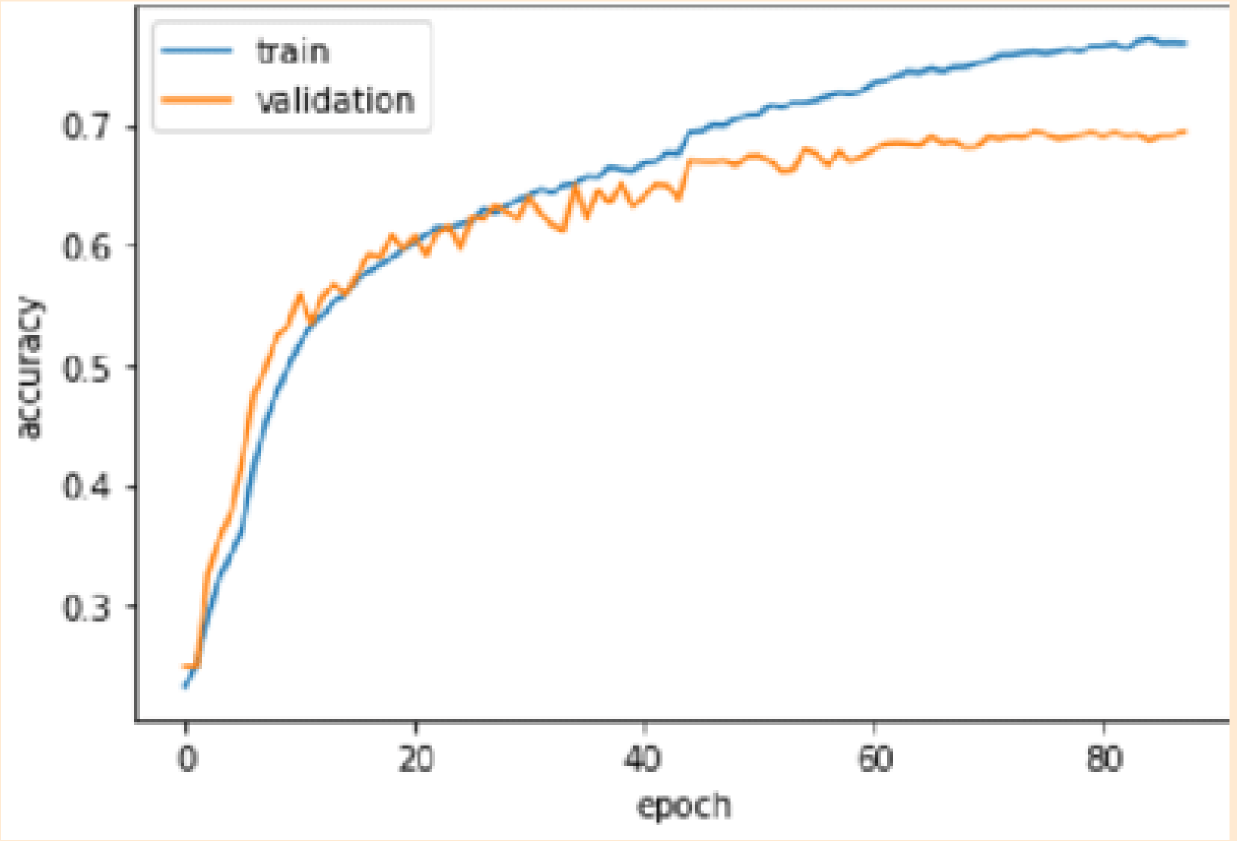
BURGLARY

EXPLOSION

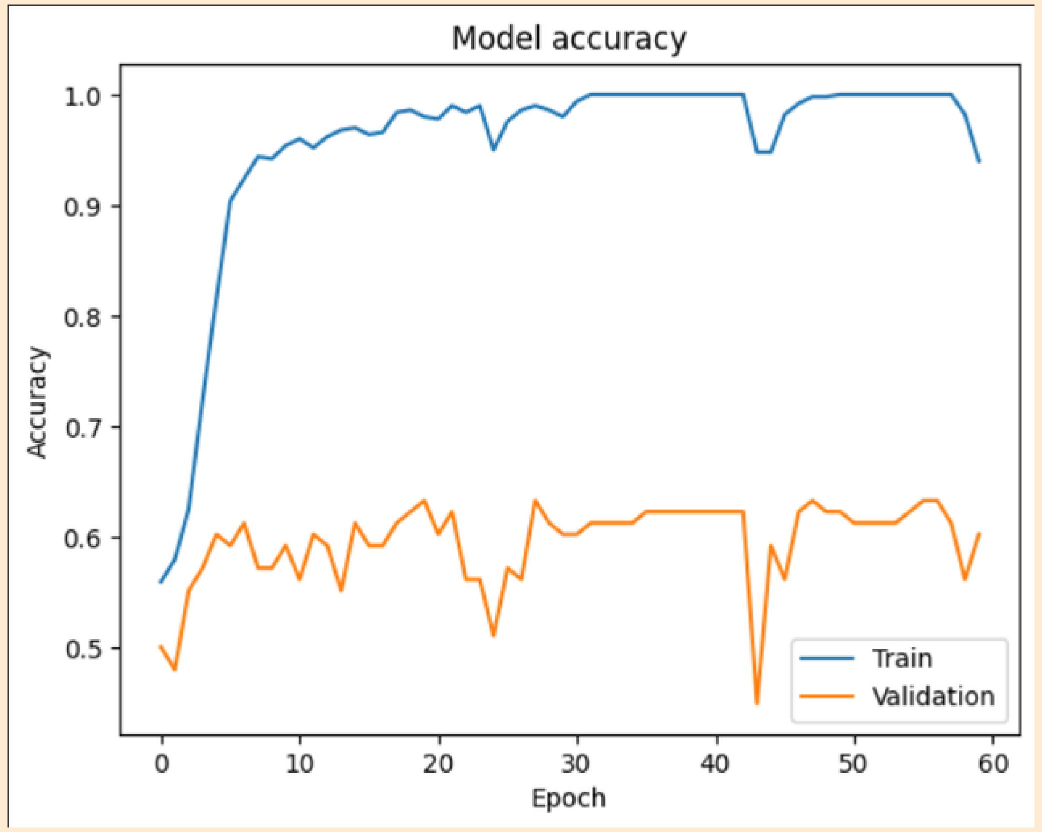
SHOOTING

+ 7...

Tuning



For Binary classification

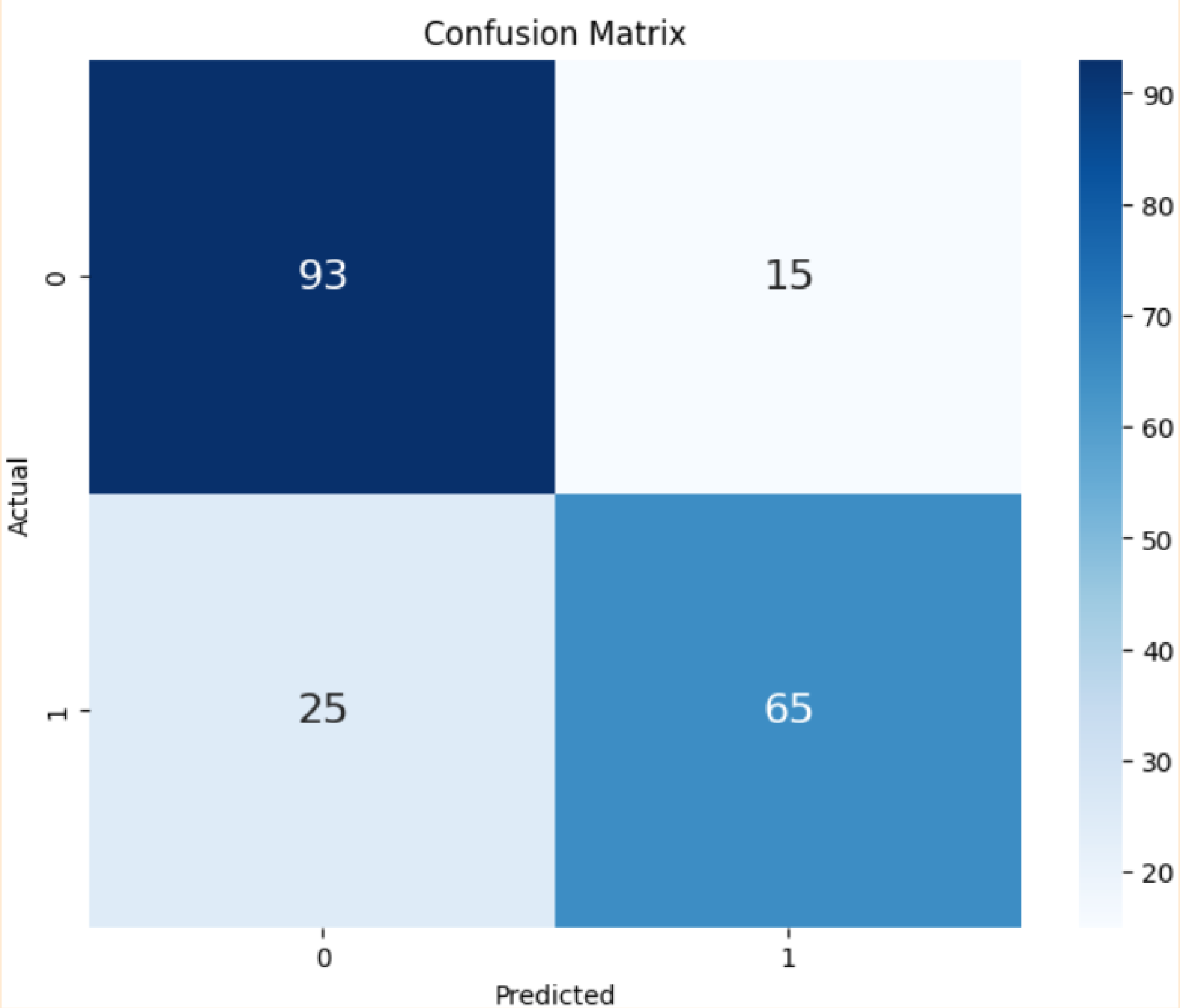


For Multi Class classification

MODEL EVALUATION

Binary classification

	precision	recall	f1-score	support
0	0.79	0.86	0.82	108
1	0.81	0.72	0.76	90
accuracy			0.80	198
macro avg	0.80	0.79	0.79	198
weighted avg	0.80	0.80	0.80	198



Multi Class classification

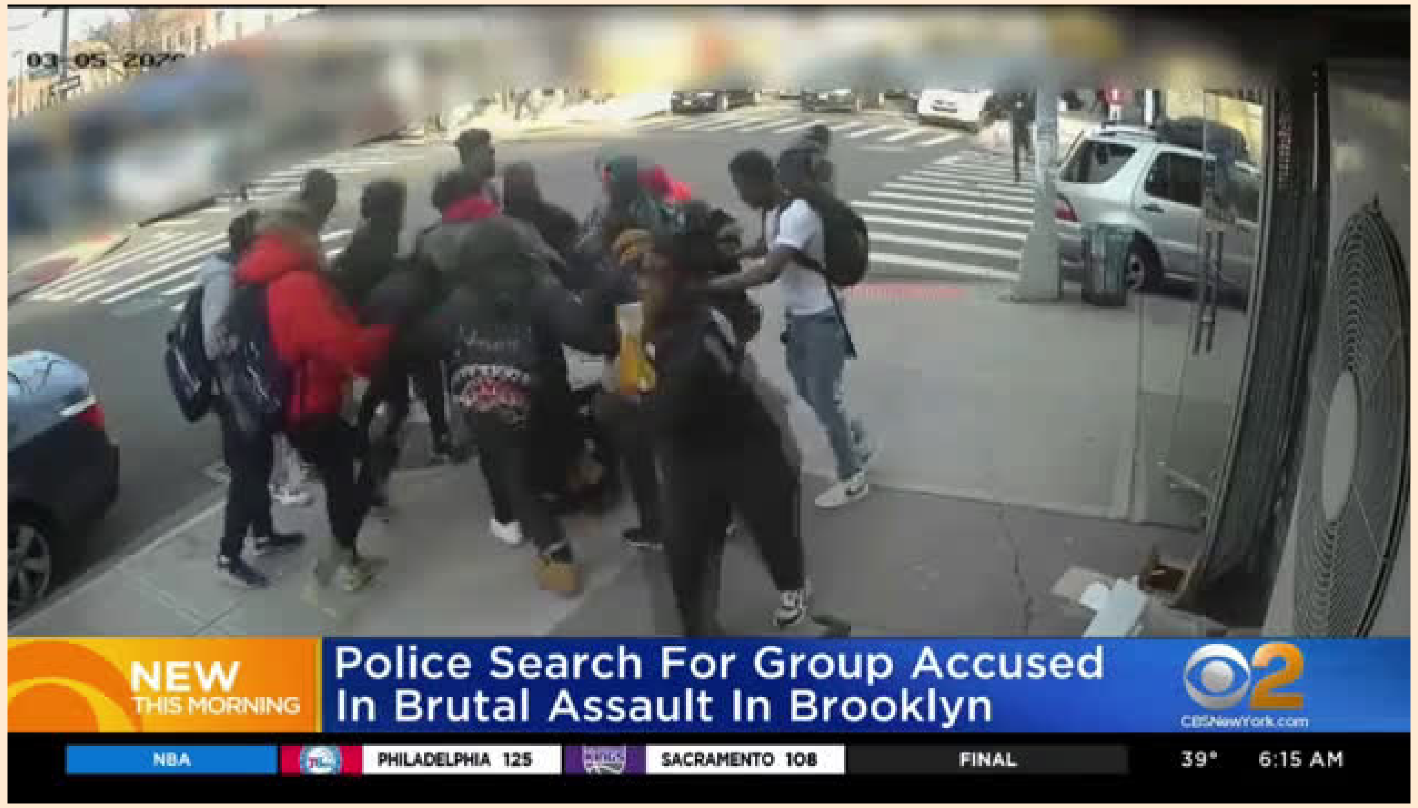
```
Classification Report:
              precision    recall  f1-score   support

           0       0.20      0.26      0.22       472
           2       0.00      0.00      0.00         0
           4       0.41      0.41      0.41      1009
           7       0.40      0.34      0.37      1019

    accuracy                           0.35      2500
   macro avg       0.25      0.25      0.25      2500
weighted avg       0.37      0.35      0.36      2500
```

REAL LIFE TESTING

Real life testing: Binary classification



RESULT

- The model predicts an abnormal behavior with the probability 77%

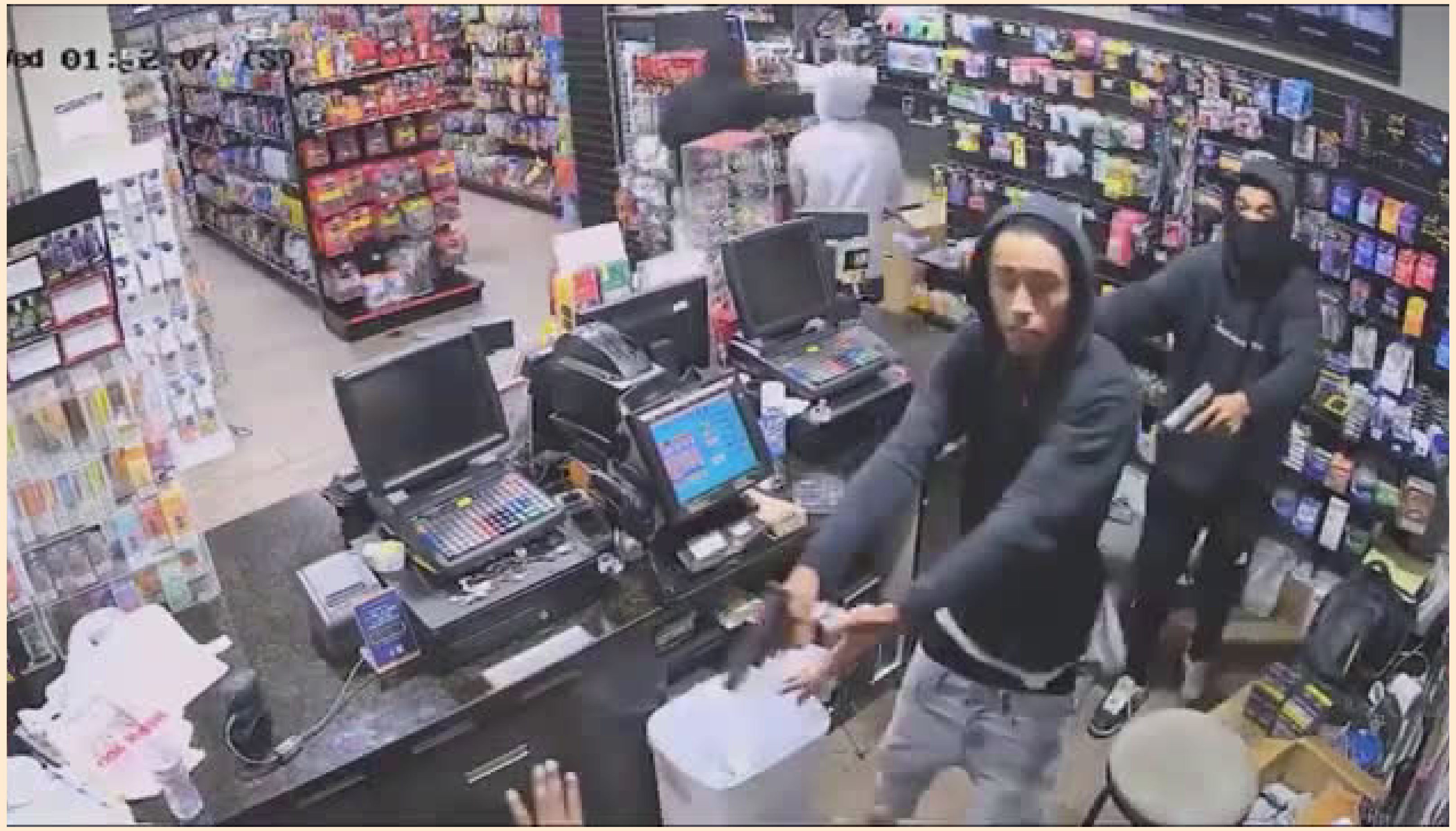
```
prediction

array([[0.61065847],
       [0.72576845],
       [0.7587639 ],
       [0.42758608],
       [0.8795326 ],
       [0.90544146],
       [0.9119626 ],
       [0.7753142 ],
       [0.7557923 ],
       [0.83062965],
       [0.95546865],
       [0.76655275],
       [0.8453363 ],
       [0.6694225 ],
       [0.90525246],
       [0.62625366]], dtype=float32)

sum(prediction)/len(prediction)

array([0.7718585], dtype=float32)
```

Real life testing: Multi Class classification



RESULT

- The model predicts a robbery with the probability of 60.5%

```
column_averages = np.mean(prediction, axis=0)

print(column_averages)
```

```
[5.6630705e-04 6.5027183e-05 1.1887921e-03 2.1362442e-03 7.3188632e-03
 2.2123150e-04 4.1088389e-04 6.0542059e-01 8.1641483e-04 2.2558886e-01
 7.7844004e-04 4.7926200e-04 1.5424687e-01 7.6212944e-04]
```

CONCLUSION

Conclusion

In this project, we developed two classification models—one for binary and the other for multiclass scenarios. Real-time testing demonstrated good performance for binary classification and acceptable results for multiclass classification. This acceptability is influenced by various factors, including the diversity of classes and variations in video formats, sizes, perspectives, and luminosity.

To further enhance the models, we can delve into discussing and refining the architecture. Additionally, expanding the dataset and incorporating continuous learning techniques can contribute to the models' adaptability and improved performance over time.