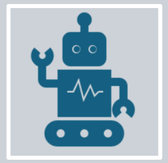
Real-time Violence Detection (RTVD)

Samsung Innovation Campus

Introduction

SCVD leverages cutting-edge artificial intelligence technologies to enhance urban safety by enabling real-time detection of violent incidents captured by CCTV systems.



AI algorithms can analyze vast amounts of video data instantaneously, significantly improving response times and operational efficiency for security agencies.

Challenges with Traditional CCTV Surveillance

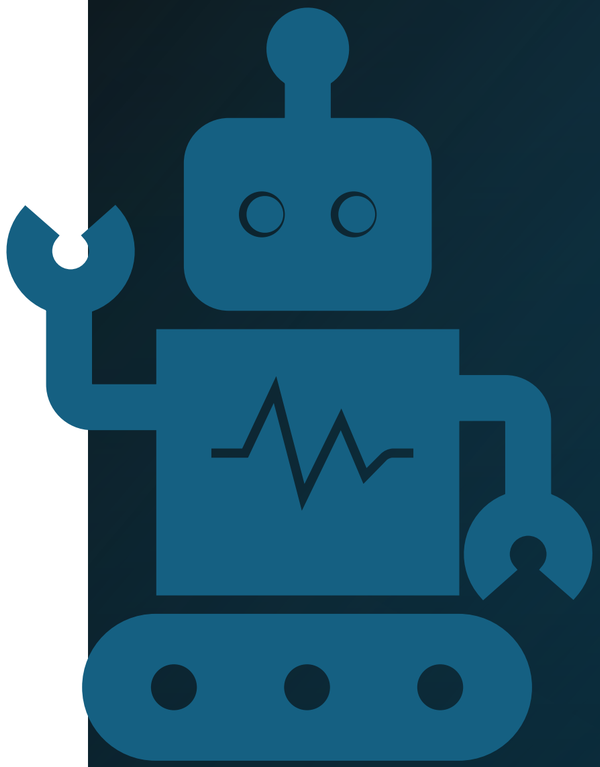
- Slow Response Times:
  Delays in human analysis can lead to escalating situations before law enforcement is notified.

- Human Error:
  Fatigue and distractions can impair the ability of human operators to detect threats.

The Role of AI in Security

- Deep Learning Capabilities:
  Utilizes Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for real-time video analysis.

- Immediate Alerts:
  Automated alerts to law enforcement when potential violence is detected, ensuring rapid response.



Market Analysis

Samsung Innovation Campus

Global Market for Smart Surveillance

Market Growth Statistics:
The global video surveillance market size was estimated at USD 56.90 billion in 2022 and is expected to grow at a compound annual growth rate (CAGR) of 12.4% from 2023 to 2030.

Samsung Innovation Car

Key Growth Factors

- Urbanization Trends:
  Over 55% of the world’s population lives in urban areas, a proportion that is expected to increase to 68% by 2050, increasing the demand for effective security solutions.
- Rising Crime Rates:
  A surge in violent crime in many urban areas necessitates advanced surveillance systems.
- Technological Advancements:
  Innovations in AI and machine learning are making advanced surveillance technologies more accessible and affordable.

Industry Competitors

- Axis Communications AB
- Bosch Security Systems Incorporated
- Honeywell Security Group
- Samsung Group
- Panasonic Corporation

In this project, We worked on two datasets with two different models due to lake of resources like (ram crashing)

- Real Life Violence Situations Dataset
- Smart-City CCTV Violence Detection Dataset (SCVD)

	Real Life Violence Situations Dataset	Smart-City CCTV Violence Detection Dataset (SCVD)
Frame size	(1080, 1920, 3)	(720, 1280, 3)
Number of videos	1800	1950,477

Technical Overview

Video Preprocessing & Deep Learning

Samsung Innovation Campus

Video Preprocessing Overview

Objective:
Enhance video data quality for accurate action recognition.

Main Steps:
1. Load video frames
2. Resize and normalize frames
3. Prepare data for model input

Reading Frames from Videos

- **Function:**
  load_video(path, nframes=FRAME_COUNT, size=(IMG_SIZE, IMG_SIZE))
- **Key Steps:**
  Total Frame Count: Retrieve total frames in the video.
  Frame Skipping: Calculate the number of frames to skip to get a subset of nframes.
  Frame Processing:
  - Resize to (IMG_SIZE, IMG_SIZE)
  - Convert BGR to RGB
  - Normalize pixel values to [0, 1]

  Output: List of processed frames returned as a numpy array.

```
ideo(path, nframes=FRAME_COUNT, size=(IMG_SIZE, IMG_SIZE)):
 = []
cv2.VideoCapture(path)


frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT)) # Get the total number of frames in
rames = max(1, (total_frames // nframes) -1)  # Ensure at least 1 frame is skipped

in range(nframes):
t, frame = cap.read()  # Read the current frame
 not ret:
  break

Process the current frame (resize, convert, normalize)
ame = cv2.resize(frame, size)
ame = frame[:, :, [2, 1, 0]]  # BGR to RGB
ame = frame / 255.0
ames.append(frame)

Skip the specified number of frames
r _ in range(skip_frames):
  cap.grab()  # Advance to the next frame without reading it

lease()
 frames
```

Extracting Frames for Training

- **Function:** extract_frames(main_directory)
- **Process:**
  Loop through each class directory and load videos.
  Use the load_video function to extract frames.
  Append valid frame sequences and corresponding labels to lists.
- **Output:** Numpy arrays for features and labels.

```
xtract_frames(main_directory):
eatures = []
abels = []
or c in scvd_classes:
    class_dir = os.path.join(main_directory,c)
    for i,video in enumerate(os.listdir(class_dir)):
        if i < 650:
            if video.endswith(('.mp4', '.avi', '.mov')):
                video_path = os.path.join(class_dir, video

                # Load and process the video
                frames = load_video(video_path)

                if len(frames) == FRAME_COUNT:
                    features.append(frames)
                    labels.append(c)
eturn np.asarray(features),np.array(labels)
```

Data Preparation for Model Training

Encoding	Label Encoding: Convert string labels into integers for model training.
Training and Testing	Training and Testing Split: Prepare datasets for training and validation.

Deep Learning Model-1 Architecture

- **Base Model:** MobileNetV2 for feature extraction (weights from ImageNet).
- **Architecture:**
  -TimeDistributed layers for processing video frames.

  -LSTM layers for capturing temporal dependencies in sequences.

  -Dense layers for final classification output.

```
model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(128, 128, 3))


ayer in base_model.layers:
ayer.trainable = False


 = Sequential()
.add(TimeDistributed(base_model, input_shape=(15, 128, 128, 3)))
.add(TimeDistributed(GlobalAveragePooling2D()))
.add(TimeDistributed(Dropout(0.5)))
.add(TimeDistributed(BatchNormalization()))

.add(LSTM(128, return_sequences=True))
.add(LSTM(64, return_sequences=False))

.add(Dense(256, activation='relu'))
.add(BatchNormalization())
.add(Dropout(0.3))
.add(Dense(128, activation='relu'))
.add(BatchNormalization())
.add(Dropout(0.3))
.add(Dense(64, activation='relu'))
.add(Dropout(0.3))
.add(Dense(1, activation='sigmoid'))

.compile(optimizer=Adam(learning_rate=1e-3), loss='binary_crossentropy', metrics=['accuracy'

.summary()
```

Model Training and Evaluation

- **Early Stopping:** Monitor validation loss to prevent overfitting.
- **Model Checkpointing:** Save the best model based on validation loss.
- **Training Process:**
- Fit the model on training data with validation split.
- Number of epochs and callback functions for training monitoring.

```
= EarlyStopping(monitor='val_loss', patience=5, restore_best
delCheckpoint(
rking/cctvmodel.keras',
l_loss',
nly=True,


.fit(


split=0.2,
early_stopping,checkpoint],
```

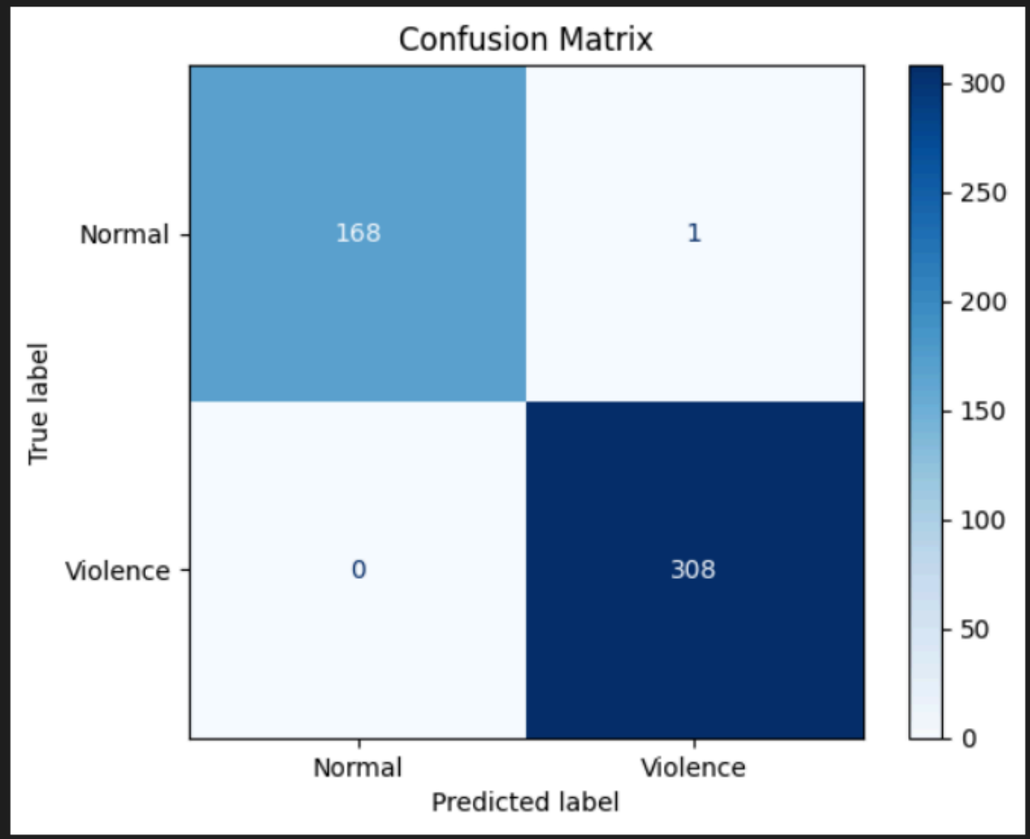
Performance Metrics Interpretation

Samsung Innovation

```
15/15 ━━━━━━━━━━━━━━━━━━━━ 37s 1s/step
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       169
           1       1.00      1.00      1.00       308

    accuracy                           1.00       477
   macro avg       1.00      1.00      1.00       477
weighted avg       1.00      1.00      1.00       477
```



Deep Learning Model-2 Architecture

- **Key Differences from Model 1:**

1. **More Complex LSTM Layers:**
    1. Model 2 includes several **Bidirectional LSTMs**, which allow the network to capture temporal dependencies more effectively in both forward and backward directions.
    2. **More layers and higher units** in the LSTM layers (starting from 256, and moving down to 32 units).
2. **Deeper Fully Connected Layers:**
    1. Model 2 has a more complex fully connected network with **1024, 512, 256, 128, and 64 units**, compared to Model 1, which stops at 256 units.
    2. More **BatchNormalization** and **Dropout layers** are used, which can help the model generalize better and prevent overfitting.
3. **Higher Capacity:**
    1. Model 2 is designed to **handle more complex patterns and relationships,** which can lead to better performance on larger, more varied datasets. However, it will also require more computational power and potentially more training data.

```
= MobileNetV2(weights='imagenet', include_top=False, input_shape=(128, 128, 3

 base_model.layers:
rainable = False


uential()
imeDistributed(base_model, input_shape=(15, 128, 128, 3)))
imeDistributed(GlobalAveragePooling2D()))
imeDistributed(Dropout(0.5)))
imeDistributed(BatchNormalization()))

TM(256, return_sequences=True))
atchNormalization())
idirectional(LSTM(128, return_sequences=True)))
idirectional(LSTM(64, return_sequences=True)))
idirectional(LSTM(32, return_sequences=False)))

ense(1024, activation='relu'))
atchNormalization())
ropout(0.3))

ense(512, activation='relu'))
atchNormalization())
ropout(0.3))

ense(256, activation='relu'))
atchNormalization())
ropout(0.3))

ense(128, activation='relu'))
atchNormalization())
ropout(0.3))

ense(64, activation='relu'))
atchNormalization())
ropout(0.3))

ense(1, activation='sigmoid'))

le(optimizer=Adam(learning_rate=1e-3), loss='binary_crossentropy', metrics=['

ry()
```

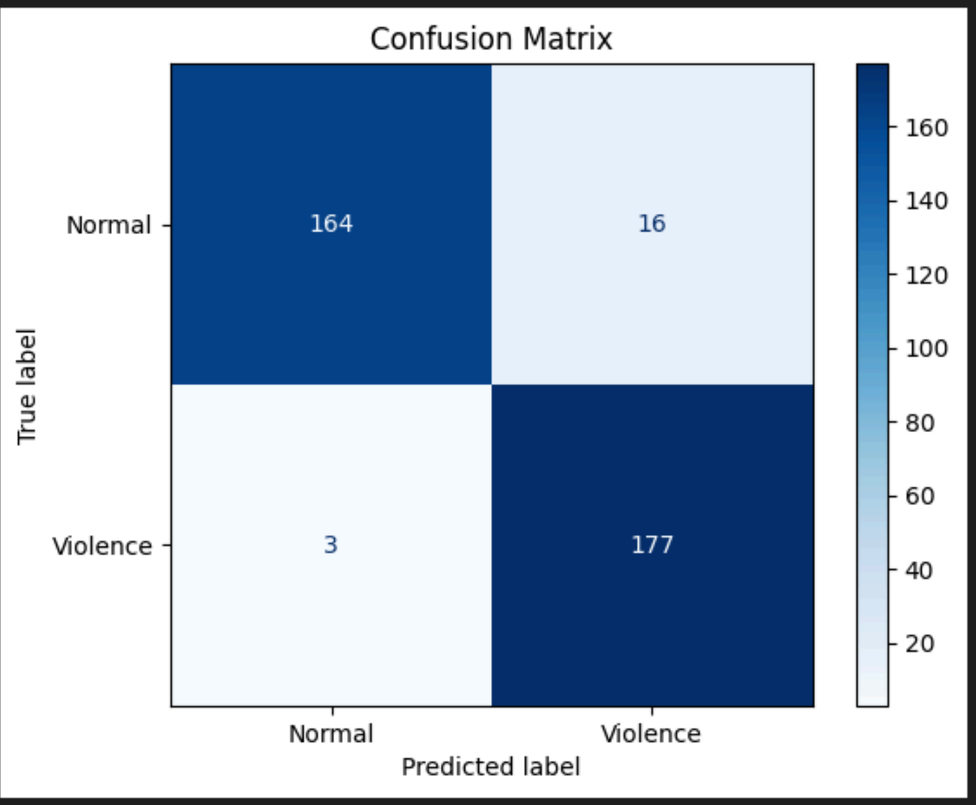
Performance Metrics Interpretation

Samsung Innovation

```
12/12 ━━━━━━━━━━━━━━━━━━━━ 35s 2s/step
              precision    recall  f1-score   support

           0       0.98      0.91      0.95       180
           1       0.92      0.98      0.95       180

    accuracy                           0.95       360
   macro avg       0.95      0.95      0.95       360
weighted avg       0.95      0.95      0.95       360
```



Deployment

BEST SECURITY SERVICES

Safe & Secure Company for your Customer

Get Quote Contact Us

Video Classification Deployment

Upload a video for classification and analysis.

Choose File | V_1000.mp4

Apply Video Classification | Display Video

Live Camera Streaming

Start Streaming

conclusion

SCVD represents a significant leap in urban security, harnessing the power of AI for real-time violence detection.