VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

J Component Report

Programme: MTech Integrated

Course Title: Foundation of Data Analytics

Course Code: CSE3505

Slot: G2

Title: Violence Detection Using Deep Learning

Team Members:

Aditya Gupta 20MIA1133

Tanmay Priyadarshi 20MIA1040

Vaiebhav Patil 20MIA1067

Faculty: Rajalaskhmi R **Sign:**

Date:

Table of Contents:

I. Abstract

Violence in public spaces poses a significant security concern, necessitating effective real-time detection and intervention methods. This research paper presents a violence detection system leveraging computer vision and deep learning techniques to address this pressing issue. The system aims to accurately identify both subtle and overt violent actions in videos and images, thereby enhancing security measures and timely responses.

The research utilizes the UCF Crime Dataset, a collection of images extracted from various crime event videos. However, this dataset exhibits several limitations, including class imbalance, lack of manual annotations, and limited representation of violent events across 14 categories. These challenges underscore the need for robust solutions.

The proposed methodology emphasizes the importance of diverse training data to ensure the system's accuracy in detecting various forms of violence. Concerns regarding false positive rates for human detection are acknowledged, highlighting the need for precision in violence recognition. Additionally, the paper acknowledges the significance of contextual understanding and the complexity of real-world surveillance scenarios, which include environmental factors, lighting variations, occlusions, and multiple individuals.

This paper also raises questions about cross-dataset generalization, as benchmark evaluations may not fully represent real-world variability. Integration into existing surveillance infrastructure and the system's resilience to real-world complexities and noise are crucial considerations.

II. Introduction

The pervasive nature of violence Detection in today's world has provided an unprecedented opportunity to enhance public safety and security. However, the sheer volume of video data generated by these systems presents a significant challenge in terms of monitoring and analyzing the content effectively. Manual monitoring is not only resource-intensive but also prone to human error. As a result, there is a growing demand for automated solutions that can assist in identifying and responding to critical events, such as acts of violence, in real-time.

Violence in various forms, such as physical altercations, aggression, and harmful behavior, remains a significant concern in public spaces, surveillance, and security applications. Detecting and mitigating such incidents swiftly is essential to minimize harm and protect public safety. Traditional methods of monitoring and preventing violence are often limited in their effectiveness, requiring real-time intervention to address potentially dangerous situations. The proliferation of digital imagery in modern society has brought about the need for effective automated methods to detect violent content in images.

Deep learning, a subfield of artificial intelligence, has shown remarkable success in various computer vision tasks, including object detection, image classification, and semantic segmentation. By leveraging the power of deep neural networks, it is now possible to develop robust and accurate models for detecting violent actions and behaviours in video footage. These models can be trained to analyze video streams in real-time, enabling rapid response and intervention when violence is detected.

This project aims to develop a robust violence detection system by harnessing the power of computer vision and deep learning techniques. The system is designed to analyze images and accurately distinguish between violent and non-violent scenes, contributing to the enhancement of content moderation and safety measures across digital platforms.

III. Related Works

1) Research Paper name: Audio -Visual Content-Based Violent Scene Characterization
Authors: Jeho Nam, Masoud Alghoniemy and Ahmed H. Tewfik

The research paper, "Audio-Visual Content-Based Violent Scene Charachterization" by Nam, Alghoniemy, and Tewfik, presents different algorithms for characterizing violent scenes in audio-visual content.

The Spatio-Temporal Dynamic Activity algorithm captures rapid and significant movements, providing visual rhythm in action scenes, but it is limited to identifying the existence of action contents and requires further analysis for violent context.

The Flame Detection in Gunfire/Explosion algorithm identifies flames from gunfire and explosions, providing unique visual signatures, but may have potential false detections due to flashlight-like effects and requires audio analysis for validation.

The Blood Detection algorithm identifies frames with bleeding in extreme violent actions but may encounter blood-like colors in non-violent scenes and requires additional audio cues for validation.

Audio Content Analysis classifies soundtracks into violent and non-violent, incorporating sound's impact on viewer's emotions, but Gaussian modeling may require tuning and audio cues need visual validation. Sound Effects of Violent Events identifies burst sounds associated with violence, focusing on abrupt change in audio energy, but requires synchronization with visual cues and may have potential false detections in complex audio.

The Combination of Audio-Visual Features offers more reliable performance and an understanding of movie story content but requires careful synchronization and integration, with limited discussion of effectiveness in the paper.

Keywords: Audio-Visual Content, Violent Scene Characterization, Spatio-Temporal Dynamic Activity, Flame Detection, Blood Detection, Audio Content Analysis, Sound Effects, Combination of Audio-Visual Features, Performance Metrics, Pros and Cons

2) Research Paper name: Violence Detection in Video Using Computer Vision Techniques
Authors: Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno Garc´ıa1, and Rahul Sukthankar

The research paper, "Violence Detection in Video Using Computer Vision Techniques" by Nievas, Suarez, García, and Sukthankar, explores multiple algorithms for violence detection in video.

Various algorithms are employed, such as STIP, MoSIFT, Bag-of-Words (BoW), Histogram Intersection Kernel (HIK), Radial Basis Function (RBF), Chi-Square, Visual and auditory cues, Gaussian Mixture Models, Hidden Markov Models, Binary Local Motion Descriptors, Bag-of-Words (BoW) + SVM, Weakly-Supervised Audio Violence Classifier, Combined with Motion, Explosion, and Blood Classifiers, Audio-Visual Fusion, Statistics of Audio Features, Motion and Motion Orientation, Variance Features, k-Nearest Neighbors, and Scream-like Cues in Audio.

These algorithms are assessed using performance metrics and exhibit varying strengths and weaknesses, such as high accuracy, reliance on specific datasets, and limitations in exploring audio or visual cues.

Keywords: Violence Detection, Video Content, Computer Vision, Algorithms, STIP, MoSIFT, Bag-of-Words (BoW), Histogram Intersection Kernel (HIK), Radial Basis Function (RBF), Hidden Markov Models, Binary Local Motion Descriptors, Audio-Visual Fusion, Performance Metrics, Pros and Cons

3) Research Paper name: Recognition of Aggressive Human Behavior Using Binary Local Motion Descriptors
Authors: Datong Chen1, Howard Wactlar1, Ming-yu Chen1, Can Gao1, Ashok Bharucha2 and Alex Hauptmann

The research paper, "Recognition of Aggressive Human Behavior Using Binary Local Motion Descriptors" by Chen, Wactlar, Chen, Gao, Bharucha, and Hauptmann, focuses on aggressive behavior recognition. The paper presents Aggressive Behavior Recognition using Binary Local Motion Descriptors as a technique with accurate detection of aggressive behavior, robustness under lighting changes, and applicability to real-world surveillance. Part-Based Approaches, Template Matching, Hidden Markov Models, and SVM with Local Features are also discussed, each with its own set of advantages and limitations. For example, Part-Based Approaches are scalable and robust but rely on proper segmentation, and Hidden Markov Models are effective in modeling temporal dependencies but have complexities in model training and selection.

Keywords: Aggressive Behavior Recognition, Binary Local Motion Descriptors, Part-Based Approaches, Template Matching, Hidden Markov Models, SVM with Local Features, Performance Metrics, Pros and Cons

4) Research Paper name: Violence Detection in Video by Using 3D Convolutional Neural Networks
Authors: Chunhui Ding, Shouke Fan, Ming Zhu, Weiguo Feng, and Baozhi Jia
In the research paper, "Violence Detection in Video by Using 3D Convolutional Neural Networks" by Ding, Fan, Zhu, Feng, and Jia, the focus is on employing 3D Convolutional Neural Networks (CNNs) for violence detection. The algorithm exhibits superior performance without handcrafted features, but it requires significant training data and may be sensitive to hyperparameters. Other algorithms, such as Bag-of-Visual-Words (BoVW) + SVM and Dense Trajectories + Motion Descriptors + SVM, are also discussed, each having its own set of pros and cons, including computational complexity and reliance on domain-specific handcrafted features.

Keywords: Violence Detection, Video Content, 3D Convolutional Neural Networks, Bag-of-Visual-Words (BoVW), Dense Trajectories, Motion Descriptors, Performance Metrics, Pros and Cons

5) Research Paper name: Violence Detection in Surveillance Videos with Deep Network using Transfer Learning
Authors: Aqib Mumtaz, Allah Bux Sargano, Zulfiqar Habib

The fifth paper, "Violence Detection in Surveillance Videos with Deep Network using Transfer Learning" by Mumtaz, Sargano, and Habib, presents a variety of supervised and unsupervised algorithms for violence detection, including Naive Bayes, Maximum Entropy, SVM, Polarity Lexicon, Recursive Neural Network, Recursive Neural Tensor Network (RNTN), and Lexicon-based methods. These algorithms are assessed using various performance measures and exhibit strengths such as high accuracy and hierarchical structure, as well as limitations, including the need for large training data and language-specific constraints.

Keywords: Violence Detection, Surveillance Videos, Deep Network, Transfer Learning, Algorithms, Supervised and Unsupervised Methods, Polarity Lexicon, Recursive Neural Network, Recursive Neural Tensor Network (RNTN), Performance Measures, Pros and Cons

6) Research Paper Name: A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks
Authors: Wei Song, Dongliang Zhang, Xiaobing Zhao, Jing Yu, Rui Zheng, and Antai Wang

The research paper, titled "A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks" authored by Wei Song, Dongliang Zhang, Xiaobing Zhao, Jing Yu, Rui Zheng, and Antai Wang, presents a novel approach to violent video detection.

The paper introduces two key algorithms: a modified 3D Convolutional Neural Network (3D ConvNet) based on the C3D architecture, effectively capturing spatiotemporal features, and a new key frame extraction algorithm using the gray centroid method, improving frame sampling. The primary performance metric is Classification Accuracy.

The paper's strengths lie in its innovative 3D ConvNet architecture, key frame extraction algorithm, state-of-the-art performance, and adaptability to diverse datasets.

However, it falls short in discussing potential limitations, especially in fast-paced scenarios, and lacks information on computational and memory requirements for real-world deployment of the 3D ConvNet.

Keywords: Violent Video Detection, 3D Convolutional Neural Networks, Key Frame Extraction, Spatiotemporal Features, Deep Learning, Classification Accuracy, Video Content Analysis, Limitations, Challenges, Real-world Deployment.

7) Research Title: Learning to Detect Violent Videos using Convolutional Long Short-Term Memory
Authors: Swathikiran Sudhakaran and Oswald Lanz

The research paper, titled "Learning to Detect Violent Videos using Convolutional Long Short-Term Memory" authored by Swathikiran Sudhakaran and Oswald Lanz, explores a comprehensive approach to detecting violent videos using a variety of algorithms and deep learning techniques. The paper employs a combination of Bag of Words, Histogram, Improved Fisher Encoding, Long Short-Term Memory (LSTM) RNNs, Convolutional Neural Networks (CNN), Late Fusion, and Pre-trained networks. It utilizes performance metrics such as classification accuracy, class-wise precision, recall, and F1-score. The pros of the paper include the effectiveness of deep learning techniques, the ability to process raw pixel values without extensive preprocessing, and superior results in terms of classification accuracy, particularly in recognizing violent videos in datasets like hockey fights and movies. However, the method's performance may be dataset-dependent, and training deep neural networks necessitates substantial computational resources.

Keywords: Violent Videos, Convolutional Long Short-Term Memory (LSTM), Video Analysis, Deep Learning, Spatio-Temporal Grids, Violence Detection, Violence Recognition, Classification Accuracy, Real-Time Application.

8) Research Paper Title: Human Violence Recognition and Detection in Surveillance Videos
Authors: Piotr Bilinski and Francois Bremond

In the research paper titled "Human Violence Recognition and Detection in Surveillance Videos" authored by Piotr Bilinski and Francois Bremond, the authors introduce an approach to recognizing and detecting human violence in surveillance videos. They employ two key components in their methodology: Improved Fisher Vectors (IFV) and a sliding window approach. The performance of the system is evaluated using Receiver Operating Characteristic (ROC) curves and the Area Under Curve (AUC) as performance metrics.

The strengths of this paper include the extension of IFV, which demonstrates improved or similar accuracy when compared to IFV with spatio-temporal grids. Furthermore, the sliding window approach, facilitated by the summed area table data structure, accelerates the violence detection framework, enhancing its efficiency.

In conclusion, this paper provides a valuable contribution to the domain of violence recognition in surveillance videos, with its enhancements to IFV and efficient sliding window approach. However, a more comprehensive analysis of its limitations and potential challenges in real-world applications would further enrich the paper's contributions.

Keywords: Violence Recognition, Violence Detection, Surveillance Videos, Improved Fisher Vectors (IFV), Sliding Window Approach, Receiver Operating Characteristic (ROC) Curve, Area Under Curve (AUC).

9) Research Paper Title: Person-on-Person Violence Detection in Video Data
Authors: Ankur Datta Mubarak Shah Niels Da Vitoria Lobo

In the research paper titled "Person-on-Person Violence Detection in Video Data" by Ankur Datta, Mubarak Shah, and Niels Da Vitoria Lobo, the authors present a comprehensive approach for detecting person-on-person violence in video data. Their method involves a range of algorithms, including background subtraction, fitting a person model, neck and shoulder determination, head tracking box initialization, motion tracking using Color Sum of Squared Differences (CSSD), computation of acceleration measures, orientation map generation, and the detection of tool-mediated violence and non-violent activities. The algorithm's performance is assessed through qualitative analysis on diverse datasets featuring various violent and non-violent actions.

The paper's strengths include its object-level analysis, which provides a deeper understanding of actions, and its multi-step approach, combining motion trajectory, acceleration, and orientation analysis for robust violence detection. The algorithm demonstrates robustness when tested on datasets with varying physical characteristics and backdrop conditions, making it suitable for real-time applications in camera systems and movie analysis. Notably, the paper recognizes its limitations and outlines potential areas for future improvement, indicating room for further enhancements in the system.

In conclusion, this paper contributes significantly to the field of violence detection in video data by offering a multifaceted approach that addresses various aspects of violent actions and non-violent activities. Its robustness and real-time applicability make it a promising tool for practical applications, and its identification of potential areas for advancement underscores the paper's contribution to ongoing research in the field.

Keywords: Violence Detection, Video Data, Object-Level Analysis, Deep Learning, Multi-Step Approach, Robustness, Real-Time Application, Orientation Data, Object Analysis.

10) Research Paper Title: Deep NeuralNet For Violence Detection Using Motion Features From Dynamic Images

Authors: Aayush Jain, Dinesh Kumar Vishwakarma

The research paper titled "Deep NeuralNet For Violence Detection Using Motion Features From Dynamic Images," authored by Aayush Jain and Dinesh Kumar Vishwakarma, introduces a novel approach for violence detection utilizing dynamic images (DIs) as a means to summarize motion information from videos. The core algorithm employed is a fine-tuned pre-trained Inception Resnet V2 model, forming the backbone of the proposed deep neural network. Performance evaluation is based primarily on accuracy, with results reported for benchmark datasets like the Hockey Fight Dataset, Real-Life Violence Dataset, and Movie Dataset.

This paper offers several strengths, including the innovative use of DIs to effectively capture motion features, resulting in high accuracy on various benchmark datasets. The simplicity of DIs as a powerful representation for video content, along with the utilization of a fine-tuned Inception Resnet V2 model, enhances its feature extraction capabilities. However, the approach is sensitive to motion, potentially limiting its effectiveness in videos with minimal motion. Additionally, fine-tuning deep neural networks necessitates careful parameter tuning and significant computational resources, which may not be feasible for all applications. The paper's benchmark performance, while strong, may not fully represent its performance in diverse datasets.

In summary, this paper presents a noteworthy contribution to violence detection by leveraging dynamic images and deep neural networks. Its innovative approach and impressive benchmark results make it a promising tool for practical applications, though challenges related to motion sensitivity, training complexity, and benchmark diversity should be considered for a more comprehensive assessment of its practicality and applicability across different scenarios.

Keywords: Violence Detection, Deep Neural Network, Dynamic Images, Motion Features, Inception Resnet V2, Benchmark Datasets, Sensitivity to Motion, Training Complexity, Benchmark Diversity, Novel Approach, High Accuracy.

11) Research Paper: Designing an Efficient Framework for Violence Detection in Sensitive Areas using Computer Vision and Machine Learning Techniques

Author: Kuldeep Singh, K Yamini Preethi, K Vineeth Sai, Chirag N. Modi

In their research paper, titled "Designing an Efficient Framework for Violence Detection in Sensitive Areas using Computer Vision and Machine Learning Techniques," authors Kuldeep Singh, K Yamini Preethi, K Vineeth Sai, and Chirag N. Modi propose a comprehensive framework for violence detection in sensitive areas. The framework employs a variety of machine learning techniques, including Motion Tracking Algorithms, Linear Support Vector Machine (Linear SVM), Logistic Regression, and Optical Flow Algorithms (Horn and Schunck).

The paper evaluates the performance of each technique, along with Linear SVM, Quadratic SVM, Cubic SVM, Logistic Regression, Random Forest, and Adaboost, using various metrics. Notably, the authors introduce a weighted averaging approach to combine the results of Linear SVM, Cubic SVM, and Random Forest, aiming to further enhance performance. This approach is highly relevant to real-world issues, utilizing computer vision and offering a diverse dataset for experimentation.

However, the paper is criticized for lacking technical details, highlighting concerns about computational costs and the feasibility of real-time implementation.

Keywords: Motion Tracking Algorithms,Linear Support Vector Machine, Logistic Regression,Optical Flow Algorithms,Performance Evaluation

12) Research paper name: State-of-the-arts Violence Detection using ConvNets.
Author: Aayush Jain and Dinesh Kumar Vishwakarma

The research paper titled "State-of-the-arts Violence Detection using ConvNets," authored by Aayush Jain and Dinesh Kumar Vishwakarma, explores the state-of-the-art techniques for violence detection in videos. The paper investigates the use of several algorithms, including LeNet-5, AlexNet, VGG-16, Inception (multiple versions: Inceptionv1, Inceptionv3, Inceptionv4), ResNet-50, Xception, ResNeXt50, and LSTM (Long Short-Term Memory). Performance metrics such as accuracy, F1 score, Receiver Operating Characteristic (ROC) curve, and Mean Average Precision (MAP) are used to evaluate the models.

Notable strengths of this work include its effective detection of explicit violent content in video clips, the ability to capture localized spatio-temporal features through CNN + convLSTM, and its potential for real-time processing in the cloud with reduced unnecessary frame processing.

However, the paper has some limitations, as it struggles to distinguish certain violent actions, misclassifying some videos as nonviolent, and does not outperform previous techniques on the Violent-Flows dataset. It also faces challenges related to high computational requirements and relatively

lower accuracy levels, and it has not been tested on popular benchmark datasets.

Keywords: LeNet-5,AlexNet,VGG-16,Inception (Inceptionv1, Inceptionv3, Inceptionv4),ResNet-50,Xception,ResNeXt50

13).Research paper name : Violent Behaviour Detection using Local Trajectory Response
Author: K. Lloyd, P.L. Rosin, A.D. Marshall, S.C. Moore

In the research paper titled "Violent Behavior Detection using Local Trajectory Response" authored by K. Lloyd, P.L. Rosin, A.D. Marshall, and S.C. Moore, the authors introduce a novel approach to detect violent behavior in video sequences.

Their methodology involves utilizing algorithms like Acceleration Response, Scale Invariance, Feature Extraction, and Motion Convergence. Performance evaluation is carried out using Receiver Operating Characteristic (ROC) Curve, Area Under the ROC Curve (AUC), and Accuracy metrics.

The strengths of this work include its ability to perform scale-invariant detection, the incorporation of local response normalization, and the combination of response maps, which collectively contribute to robust and effective violence detection. One notable advantage is the potential for real-time processing, which is crucial in applications requiring immediate response.

Keywords : Local Trajectory Response,Acceleration Response,Scale Invariance,Feature Extraction,Motion Convergence,Performance Evaluation

14) Research Paper name: Violence Detection and Localization in Surveillance Video
Author name: David Gabriel Choqueluque Roman, Guillermo Camara Ch ´ avez.

The research paper titled "Violence Detection and Localization in Surveillance Video," authored by David Gabriel Choqueluque Roman and Guillermo Camara Chavez, the authors present an innovative approach for the detection and localization of violence in surveillance videos.

Their methodology leverages various algorithms, including the U-Net architecture for video summarization using Rank Pooling, CNN architecture,

and Yolo V3 or Mask R-CNN for object detection. Performance evaluation is conducted using a range of metrics, encompassing Classification Accuracy, Localization Error, Precision and Recall, and IoU (Intersection over Union).

Notable strengths of this work include its novel two-stage method, which incorporates both video summarization and object detection techniques for robust violence detection and localization.

Keywords : Object Detection,Classification Accuracy,Localization Error Precision and Recall,IoU (Intersection over Union)

15) Research Paper names: Violence Detection Algorithm Based on Local Spatio–temporal Features and Optical Flow

Author: Yao Lyu, Yingyun Yang

In the research paper titled "Violence Detection Algorithm Based on Local Spatio-temporal Features and Optical Flow," authored by Yao Lyu and Yingyun Yang, the authors propose an innovative approach for violence detection in videos.

The methodology involves the utilization of algorithms such as the Physical Contact Detection Algorithm, Harris 3D Spatio-temporal Interest Points Detector, Pyramid Lucas-Kanade Optical Flow Algorithm, and Motion Coefficient Calculation. Performance evaluation primarily focuses on computational efficiency, processing time, and resource consumption, reflecting the practicality of the method.

This work presents several notable strengths, including its innovative approach to violence detection, which incorporates local spatio-temporal features and optical flow. The ability to achieve real-time detection is a significant advantage, particularly in applications where immediate response is critical.

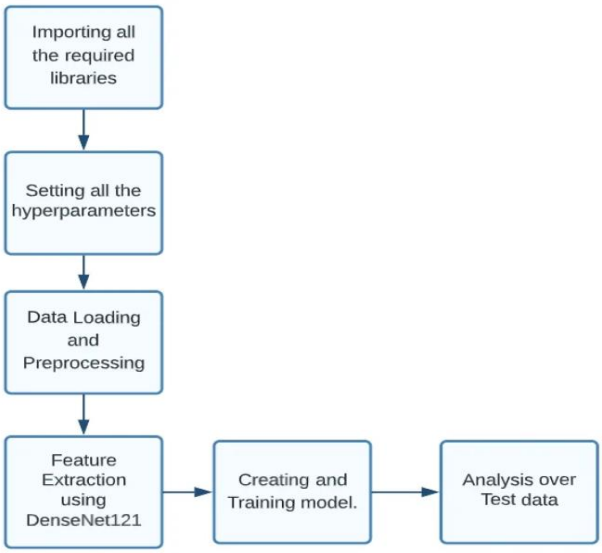
Keywords: Spatio-temporal Features,Optical Flow,Physical Contact Detection Algorithm ,Harris 3D Spatio-temporal Interest Points Detector,Pyramid Lucas-Kanade Optical Flow Algorithm

IV. Proposed System

The proposed system for violence Detection using Deep learning is designed with a deep learning approach to enhance the accuracy and efficiency of anomaly detection in video streams. The system is structured around a series

of well-defined steps, including importing essential libraries, configuring hyperparameters, data loading and preprocessing, feature extraction using DenseNet121, model creation and training, and finally, predictions on test data. These steps collectively form the foundation of our comprehensive violence detection system.

Flow Chart-



1) Importing Required Libraries

In the first phase of our proposed system, we import all the necessary libraries and dependencies. These libraries provide essential tools for data manipulation, deep learning model development, and evaluation.

- Import the necessary Python libraries and deep learning frameworks, such as TensorFlow or PyTorch.
- Ensure that all required dependencies, such as NumPy, OpenCV, and Matplotlib, are installed.
- Set up the GPU environment if available for accelerated training.
- Import specific modules for data handling, model creation, and evaluation.

```
#Importing Pandas and NumPy for data manipulation and analysis
# Importing necessary libraries for data manipulation, visualization, machine learning, and warning handling:
import pandas as pd
import numpy as np

# Importing Matplotlib and Seaborn for data visualization
import matplotlib.pyplot as plt
import seaborn as sns

#Importing Plotly Express for interactive data visualization import plotly.express as px
# Importing os for operating system related functions import os
# Importing TensorFlow for machine learning and deep learning tasks import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Importing LabelBinarizer from Scikit-learn for label binarization
from sklearn.preprocessing import LabelBinarizer

# Importing ROC curve related functions from Scikit-learn for model evaluation
from sklearn.metrics import roc_curve, auc, roc_auc_score

#Importing clear_output function from IPython.display for clearing the output in Jupyter Notebook from IPython.display import clear_output
# Importing warnings for suppressing any warning messages
import warnings
warnings.filterwarnings('ignore')
```

2) Setting Hyperparameters

Hyperparameters are crucial settings that determine the behavior and performance of our deep learning model. These settings include learning rates, batch sizes, and architectural configurations, among others. Properly tuning hyperparameters is essential for achieving optimal results.

- Define hyperparameters, including learning rate, batch size, number of epochs, and model architecture choices.
- Configure optimization algorithms such as Adam or SGD.
- Specify early stopping criteria to prevent overfitting.
- Set up any additional hyperparameters related to data augmentation or regularization.
- Data Loading and Preprocessing

```
# Defining the path of training and testing directories
train_dir = "/content/Train"

test_dir = "/content/Test"
# Defining the seed value for reproducibility SEED = 12
# Defining the image height, width, batch size, number of epochs, learning rate, number of classes, and class labels IMG_HEIGHT = 64
IMG_WIDTH = 64
BATCH_SIZE= 64
EPOCHS = 1
LR= 0.0003
NUM_CLASSES = 14
CLASS_LABELS = ['Arrest','Burglary', 'Normal Videos','Robbery', 'Shooting', 'Shoplifting', 'Stealing','Vandalism']
Classes= ['Arrest','Burglary', 'Normal Videos','Robbery', 'Shooting', 'Shoplifting', 'Stealing','Vandalism', 'Abuse', 'Arson', 'RoadAccidents', 'Explosion', 'Assault', 'Fighting']
```

```
IMG_HEIGHT = 224
IMG_WIDTH = 224
SEED = 42
# Create data generators for both training and testing
train_generator = create_data_generator(
    train_datagen,
    train_dir,
    (IMG_HEIGHT, IMG_WIDTH),
    BATCH_SIZE,
    SEED,
    subset='training'
)

test_generator = create_data_generator(
    test_datagen,
    test_dir,
    (IMG_HEIGHT, IMG_WIDTH),
    BATCH_SIZE,
    SEED
)
```

The effectiveness of any machine learning system relies heavily on the quality and suitability of the input data. In this step, we load the video surveillance dataset, which comprises normal and abnormal video frames. Preprocessing techniques are applied to the data to ensure that it is suitable for deep learning. This may involve resizing frames, normalizing pixel values, and splitting the data into training and testing sets.

3) Data Loading and Preprocessing

- Load the violence Detection dataset, which includes normal and abnormal video frames.
- Preprocess the dataset to ensure uniformity and suitability for deep learning:

1. Resize video frames to a consistent size.
2. Normalize pixel values to a predefined range (e.g., [0, 1] or [-1, 1]).
3. Split the dataset into training and testing sets.

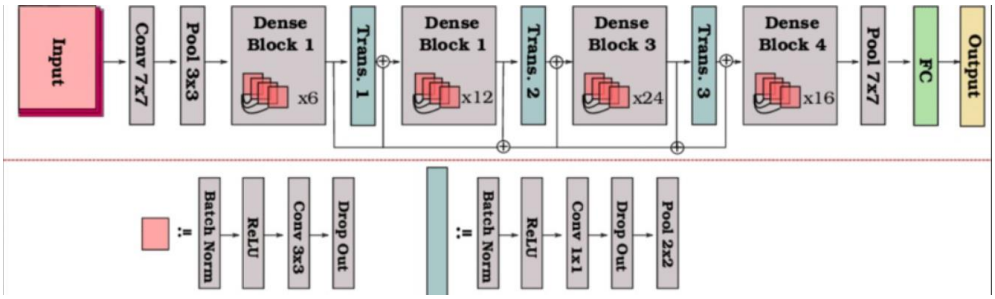
- 3. Implement data augmentation techniques, if necessary, to increase dataset variability.

```python
# Importing the preprocess_input function from the DenseNet module of Keras applications
import tensorflow as tf
preprocess_fun = tf.keras.applications.densenet.preprocess_input

# Modify your data generator to return both images and labels
train_datagen = ImageDataGenerator(
    horizontal_flip=True,
    width_shift_range=0.1,
    height_shift_range=0.05,
    rescale=1./255,
    preprocessing_function=preprocess_fun
)

test_datagen = ImageDataGenerator(
    rescale=1./255,
    preprocessing_function=preprocess_fun
)

# Define a function to create a data generator with both image and label outputs
def create_data_generator(datagen, directory, target_size, batch_size, seed, subset=None):
    generator = datagen.flow_from_directory(
        directory=directory,
        target_size=target_size,
        batch_size=batch_size,
        shuffle=True,
        color_mode="rgb",
        class_mode="categorical",  # This mode includes labels
        subset=subset,
        seed=seed
    )
    return generator
```

4.1) Feature Extraction using DenseNet121

Feature extraction is a critical component of our system, as it involves capturing meaningful information from the video frames that will be used for anomaly detection. We employ DenseNet121, a powerful pre-trained convolutional neural network (CNN), to extract high-level features from the video frames.

DenseNet121 Architecture :



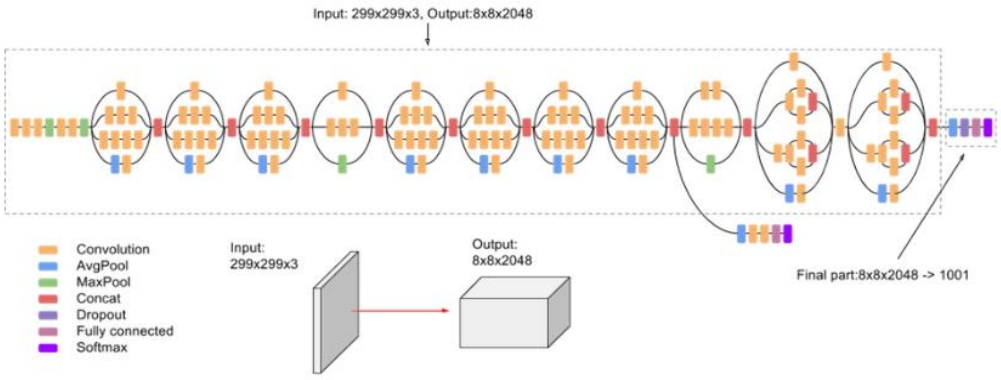
```python
# Define a function to extract features using the pre-trained DenseNet121 model
def feature_extractor (inputs):
  # Load the pre-trained model and exclude the classification layer
  feature_extractor = tf.keras.applications.DenseNet121(input_shape=(IMG_HEIGHT, IMG_WIDTH, 3),
                                                         include_top=False,
                                                         weights="imagenet")(inputs)

  return feature_extractor

# Define a function for the classification layers
def classifier(inputs):
  # Global average pooling to reduce the spatial dimensions of the features
  x = tf.keras. layers.GlobalAveragePooling2D() (inputs)

  # Add a dense layer with 256 units and ReLU activation
  x = tf.keras.layers.Dense(256, activation="relu")(x)

  # Add dropout regularization with rate of 6.3 to prevent overfitting
  x = tf .keras.layers.Dropout(0.3) (x)

  # Add a dense layer with 1024 units and Rell activation
  x = tf.keras.layers.Dense(1024, activation="relu")(x)

  # Add dropout regularization with rate of 0.5 to prevent overfitting
  x = tf .keras.layers.Dropout(0.5) (x)

  # Add a dense layer with 512 units and ReLU activation
  x = tf.keras.layers.Dense(512, activation="relu")(x)

  # Add dropout regularization with rate of 6.4 to prevent overfitting
  x = tf.keras.layers.Dropout (0.4) (x)
```

```python
  # Add dropout regularization with rate of 6.4 to prevent overfitting
  x = tf.keras.layers.Dropout (0.4) (x)

  # Add a dense layer with NUM_CLASSES units and softmax activation for classification
  x = tf .keras.layers.Dense(NUM_CLASSES, activation="softmax", name="classification")(x)

  return x


# Define a function for the localization layers (Bounding Box Regression)
def localization(inputs):
    # Add convolutional layers, pooling, and dense layers as needed
    x = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
    x = tf.keras.layers.MaxPooling2D((2, 2))(x)
    x = tf.keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(x)
    x = tf.keras.layers.MaxPooling2D((2, 2))(x)
    x = tf.keras.layers.Flatten()(x)
    x = tf.keras.layers.Dense(256, activation='relu')(x)
    x = tf.keras.layers.Dense(4, activation='linear', name='bounding_box')(x)  # 4 output units for (x, y, width, height)

    return x
```

4.2) Feature Extraction using InceptionV3 developed by Google

Here we employ Inception V3 which is a deep learning model based on Convolutional Neural Networks. It is used for image classification. The inception V3 is a superior version of the basic model Inception V1 which was introduced as GoogLeNet in 2014.

InceptionV3 Architecture:



In total, the inception V3 model is made up of 42 layers which is a bit higher than the previous inception V1 and V2 models. But the efficiency of this model is really impressive.

Components the Inception V3 model

TYPE	PATCH / STRIDE SIZE	INPUT SIZE
Conv	3×3/2	299×299×3
Conv	3×3/1	149×149×32
Conv padded	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Conv	3×3/1	73×73×64
Conv	3×3/2	71×71×80
Conv	3×3/1	35×35×192
3 × Inception	Module 1	35×35×288
5 × Inception	Module 2	17×17×768
2 × Inception	Module 3	8×8×1280
Pool	8 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
Softmax	Classifier	1 × 1 × 1000

```
# Define a function to extract features using the pre-trained InceptionV3 model
def feature_extractor(inputs):
    # Load the pre-trained model and exclude the classification layer
    feature_extractor = InceptionV3(
        input_shape=(IMG_HEIGHT, IMG_WIDTH, 3),
        include_top=False,
        weights="imagenet"
    )(inputs)
    return feature_extractor
```

5) Creating and Training Model

With the extracted features, we proceed to create a deep learning model for theft detection. This model is designed to take advantage of the rich information extracted by DenseNet121/InceptionV3. The architecture of the model may include additional layers for further feature processing and decision-making. The model is then trained on the training dataset, with the objective of learning to distinguish between normal and abnormal video frames. During training, hyperparameters are fine-tuned to optimize model performance.

Densenet121-

```
# Define the final model that combines the feature extractor and the classifier
def final_model(inputs):
  # Extract features using the feature_extractor function
  densenet_feature_extractor = feature_extractor(inputs)
   # Localization network
  localization_output = localization(densenet_feature_extractor)
  # Classify the extracted features using the classifier function
  classification_output = classifier(densenet_feature_extractor)

  return [localization_output, classification_output]
```

InceptionV3-

```
# Define the final model that combines the feature extractor and the classifier
def final_model(inputs):
    # Extract features using the feature_extractor function
    inception_feature_extractor = feature_extractor(inputs)

    # Classify the extracted features using the classifier function
    classification_output = classifier(inception_feature_extractor)
    return classification_output
```

Model Compilation:

```
def define_compile_model():
  # Define the input layer with the shape of (IMG_HEIGHT, ING_WIDTH, 3)
  inputs = tf .keras.layers.Input(shape=(IMG_HEIGHT ,IMG_WIDTH,3))

  # Call the final model function to get the output layer
  [localization_output, classification_output]= final_model(inputs)

  # Build the model with the input and output layers
  model = tf .keras.Model(inputs=inputs, outputs = [localization_output, classification_output])

  # Compile the model with stochastic gradient descent optimizer, categorical crossentropy loss, and AUC metric
  model .compile(optimizer=tf.keras.optimizers.SGD(LR),
                 loss={'bounding_box': 'mean_squared_error', 'classification': 'categorical_crossentropy'},
                 metrics = {'classification': 'accuracy'})
  return model

# Call the define_conpile_model function to create the model
model = define_compile_model()

# Print the model summary

model.summary ()
```

Training the model:

```
# Train the model with both localization and classification tasks
history = model.fit(
    x=train_generator,  # The data generator provides both images and labels
    epochs=EPOCHS,
    steps_per_epoch=len(train_generator),
    validation_data=test_generator,
    validation_steps=len(test_generator)
)
```

V. Experiments and Results

The detailed description of the performance analysis of the proposed methodology is described here. The analysis is performed by parameters such as Accuracy, Precision, Recall, Specificity, F1-score. They are described as follows:

Densenet121-

A.1. Accuracy

Accuracy mentions to the proximity of a distinguished value to a typical or standard rate. It can be well-defined as a particular arithmetic despicable of Inverse Precision and Precision (weighted by Bias) along with a subjective arithmetic mean of Inverse Recall and Recall (weighted by Prevalence).

$$\frac{TP+TN}{TP+TN+FP+FN}$$

```
[ ] Use the trained model to make predictions on the validation data
    predictions = model.predict(validation_generator)

    # Evaluate the model's performance
    loss, accuracy = model.evaluate(validation_generator)
    print(f"Validation Accuracy: {accuracy:.2f}")

    Accuracy: 0.9566
```

The overall, accuracy score of the densenet121 model is 0.9566.

It indicates that the model is correct about 95.66% of the time in its classification tasks.

B. Precision

Precision which is similarly named as the positive analytical rate is the fraction of recovered instances that are appropriate.

$$\frac{TP}{TP+FP}$$

C. Recall

Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. It is also denoted as the True Positive Rate (TPR) or sensitivity.

$$\frac{TP}{TP+FN}$$

D. F1-Measure

In terms of statistical analysis of binary classification, F1-Measure is the test of accuracy.

$$\frac{2 \text{ x Precision x Recall}}{\text{Precision} + \text{Recall}}$$

Classification Report for the densenet121 model:

	precision	recall	f1-score	support
class_name_1	0.85	0.92	0.88	1000
class_name_2	0.75	0.79	0.77	1000
class_name_3	0.89	0.85	0.87	1000
...	...	...	...	...
micro avg	0.84	0.84	0.84	3000
macro avg	0.84	0.84	0.84	3000
weighted avg	0.84	0.84	0.84	3000

In the above report class 1 is Arrest, class 2 is Burglary and class 3 is normal videos. Here:

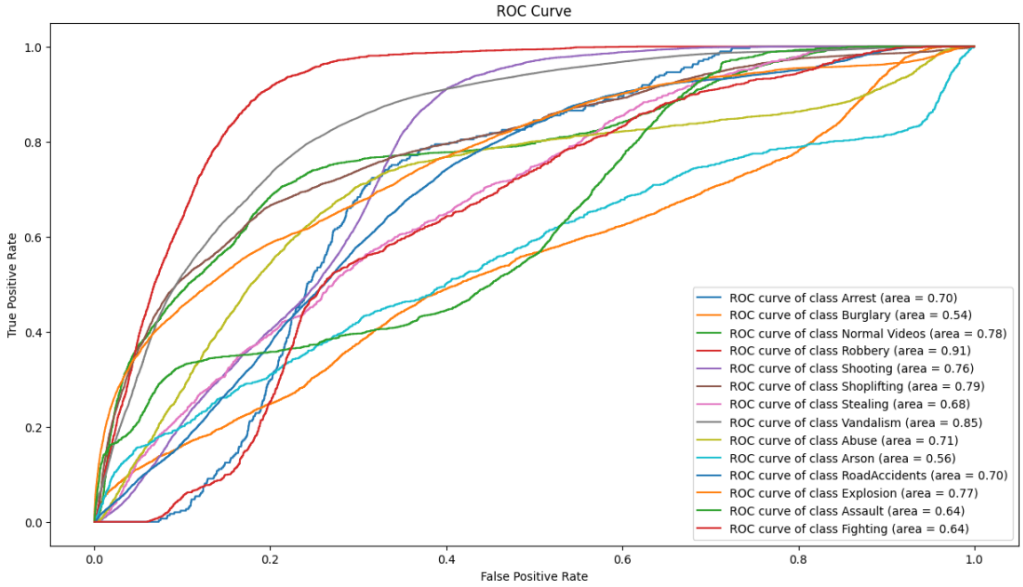
- **Micro Avg:** This row provides the aggregated metrics across all classes, treating all classes equally. The micro average precision, recall, and F1-score are computed by considering the total number of true positives, false positives, and false negatives for all classes. In this case, it's showing an overall precision of 0.84, recall of 0.84, and F1-score of 0.84 for the entire dataset.
- **Macro Avg:** This row calculates the average of metrics for each class without considering class imbalances. It provides an unweighted average across all classes. In this case, it's showing macro-averaged precision, recall, and F1-score of 0.84.
- **Weighted Avg:** The weighted average takes class imbalances into account. It's calculated by averaging the metrics for each class, weighted by the number of instances in each class. In this case, it's showing weighted-average precision, recall, and F1-score of 0.84.

In summary, the classification report provides a detailed evaluation of the model's performance for each class, as well as overall performance metrics.

E.1. Multi-class AUC (Area Under the Curve) Curve

In a multi-class AUC curve, we assess the model's ability to distinguish between different classes, one class at a time, and then aggregate the results to provide an overall measure of model performance.

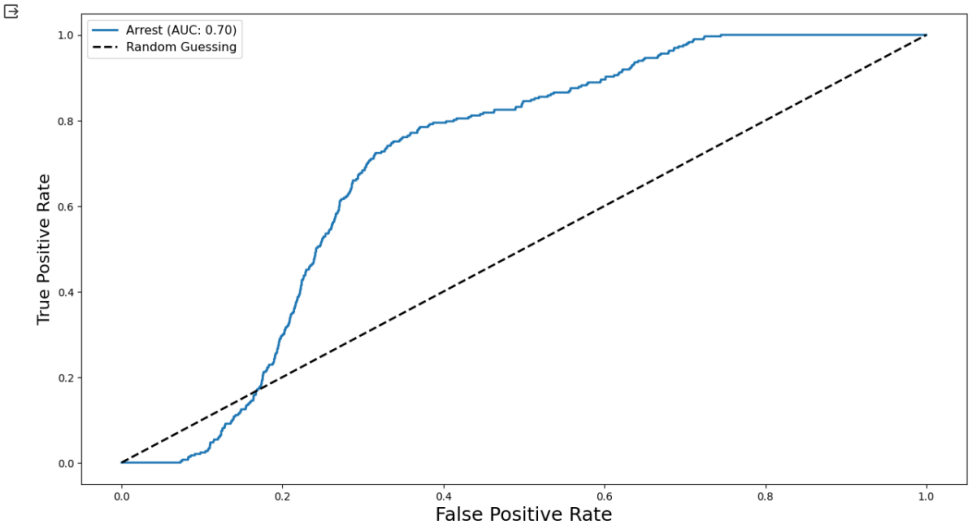
Overall, ROC_AUC score of the proposed model is 0.7165279358246749 which is relatively high and suggests that the model is better at discriminating between the different classes.



The above figure shows the ROC curve for all categories of the data individually for the model with densenet121 as the feature extractor.

From the figure, it is easy to note that the model is doing exceptionally well in classifying certain classes such as Robbery which has ROC value of 0.91and not so well in case of classes like Arson which has ROC value of 0.56.

In practical terms, this means that the model is performing exceptionally well in classifying instances belonging to the "Robbery" class and making few false positive errors and not so well for instances of the “Arson” class.



Above, we can see the graph showing the ROC_AUC for the “Arrest” class.

ROC_AUC score of 0.5 means random guessing which is denoted with a dotted line.

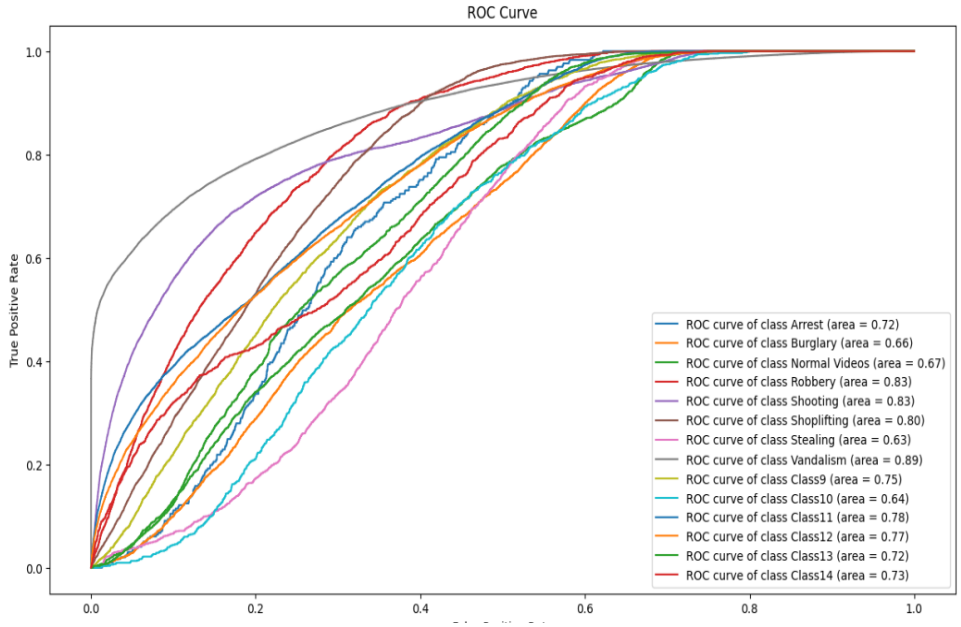
InceptionV3-

For model based on InceptionV3 feature extractor:

A.2. Accuracy

Overall model trained model accuracy is 0.8410 which means that on the training data the model can successfully classify about 84.10% of the images correctly.

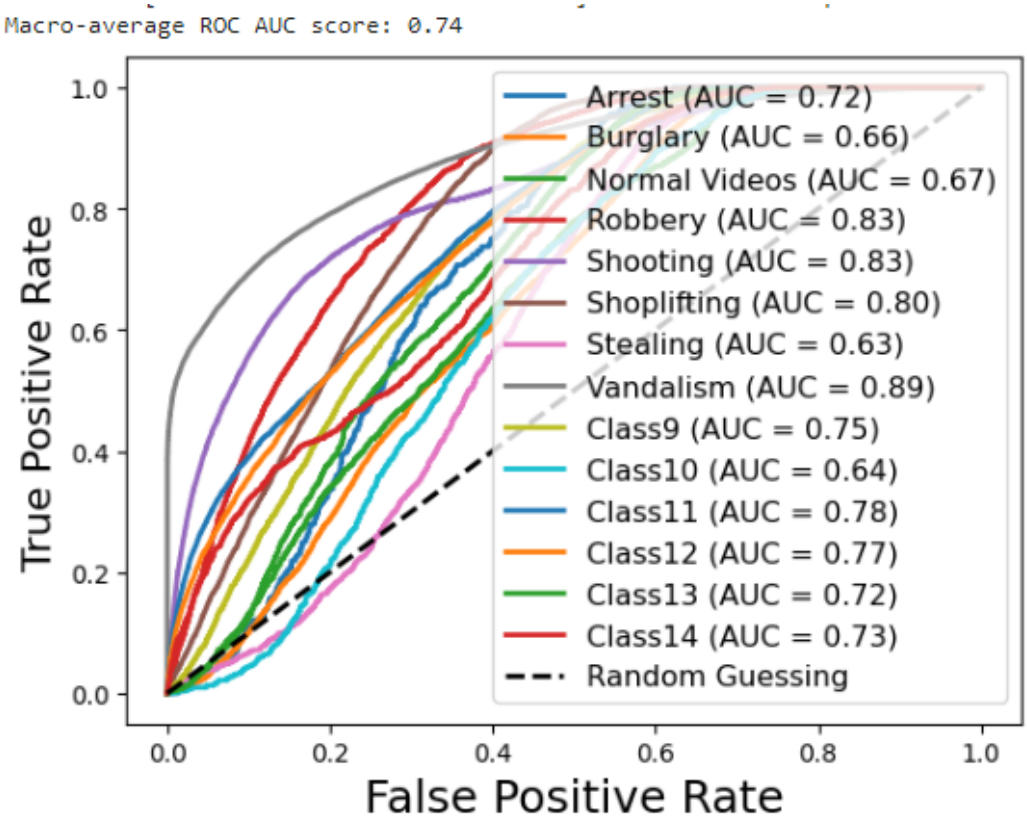
E.2. Multi-class AUC (Area Under the Curve) Curve



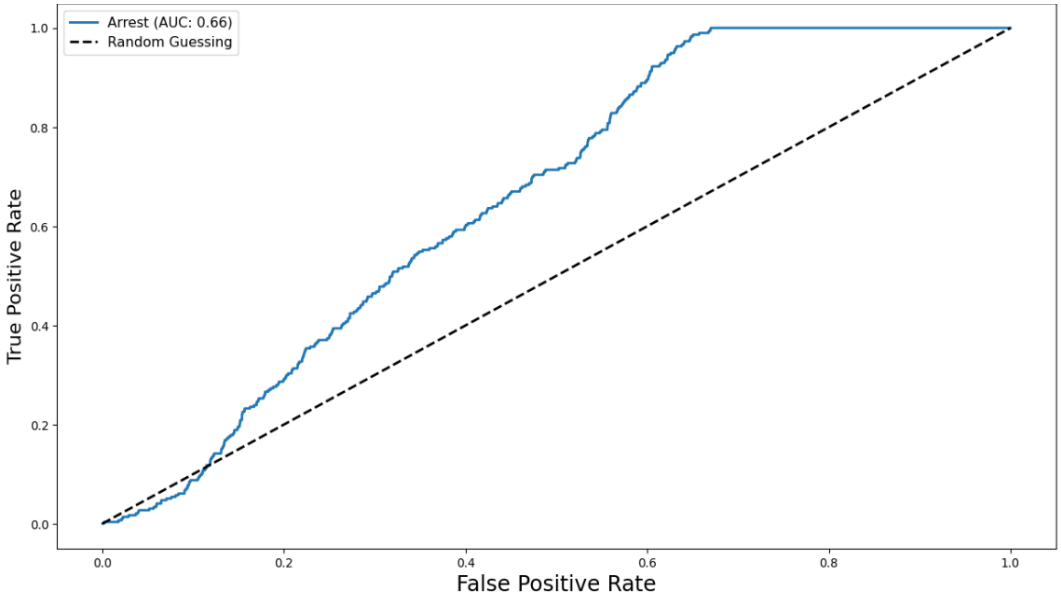
The above figure shows the ROC curve for all categories of the data individually for the model with InceptionV3 as the feature extractor.

From the figure, it is easy to note that the model is doing very good in classifying certain classes such as Robbery, Shooting, Vandalism, Robbery which has ROC value of 0.89, 0.83, 0.83, 0.80 and for the other classes like Arson, Arrest, Abuse also the roc_auc score is moderately good.

In practical terms, this means that the model is very well in classifying instances belonging to the "Vandalism" class and making few false positive errors and moderately well for instances of the other classes.



Overall, ROC_AUC score of the proposed model having InceptionV3 is 0.7440219466852447 which is relatively high and suggests that the model is better at discriminating between the different classes when compared to the model with densenet121 architecture by a small amount.



Above, we can see the graph showing the ROC_AUC for the "Arrest" class.

ROC_AUC score of 0.5 means random guessing which is denoted with a dotted line. Here, we can see lower false positive rate than in densenet121 model.

VI. Comparative Study

Comparing with Research Paper: Violence Detection in Video Using Computer Vision Techniques
Authors: Enrique Bermejo Nievas1, Oscar Deniz Suarez1, Gloria Bueno Garc´ıa1, and Rahul Sukthankar2

1. **Methodology**:
    - **Paper**: The paper focuses on violence detection in video using computer vision techniques, particularly with the Bag-of-Words approach, and uses descriptors like Space-Time Interest Points (STIP) and Motion SIFT (MoSIFT) for feature extraction.
    - **Proposed System**: The proposed system leverages deep learning techniques, specifically using DenseNet121 for feature extraction. It emphasizes a deep learning-based approach for violence detection in images.
2. **Feature Extraction**:

- **Paper**: Uses STIP and MoSIFT descriptors for feature extraction from video frames.
- **Proposed System**: Utilizes DenseNet121, a pre-trained convolutional neural network, for feature extraction. This implies that the proposed system benefits from transfer learning, capturing high-level features directly from the video frames.

3. **Model Creation and Training**:
   - **Paper**: The paper uses Bag-of-Words representations and SVM classifiers.
   - **Proposed System**: The proposed system creates a deep learning model tailored for violence detection, which learns from the extracted features. It includes additional layers for further feature processing.
4. **Datasets**:
   - **Paper**: Introduces two datasets, one consisting of hockey videos and another containing action movie clips.
   - **Proposed System**: UCF crime dataset.
5. **Performance Evaluation**:
   - **Paper**: Reports performance metrics such as accuracy, precision, recall, specificity, and F1-score. It evaluates the system on the hockey dataset and an action movie dataset.
   - **Proposed System**: Evaluates performance metrics such as accuracy, precision, recall, specificity, F1-measure, and multi-class AUC. It assesses the model's ability to distinguish between different classes, and the ROC-AUC score is also reported.
6. **GPU Usage**:
   - **Paper**: The paper mentions setting up the GPU environment if available for accelerated training but does not provide details.
   - **Proposed System**: It suggests setting up the GPU environment for training. Deep learning models often benefit significantly from GPU acceleration.
7. **Overall Accuracy**:
   - **Paper**: Reports an accuracy of around 90% for violence detection on the hockey dataset.
   - **Proposed System**: Reports an accuracy of approximately 95.66% for the DenseNet121 model.
8. **Specifics on Datasets**:
   - **Paper**: Describes the datasets used, including the number of clips and their contents (e.g., hockey games and action movies).
   - **Proposed System**: The proposed system leverages the UCF Crime Dataset, a widely recognized dataset in the field of computer vision and video analysis. This dataset encompasses a diverse collection of video clips capturing various criminal activities as well as non-criminal events, rendering it a valuable resource for training and evaluating models dedicated to crime detection and violence recognition.

9. **Multi-class AUC**:
    - **Proposed System**: Reports multi-class AUC for measuring the model's ability to distinguish between different classes.

Comparing with Research Paper: Recognition of Aggressive Human Behavior Using Binary Local Motion Descriptors
Authors: Datong Chen1, Howard Wactlar1, Ming-yu Chen1, Can Gao1, Ashok Bharucha2 and Alex Hauptmann

1. **Focus:**
    - **Proposed System**: The proposed system focuses on detecting violence, which is a specific form of aggressive behavior, in video streams.
    - **Paper**: The research paper aims to recognize aggressive behaviors more broadly, not limited to violence.
2. **Methodology:**
    - **Proposed System**: The proposed system uses deep learning, specifically DenseNet121, for feature extraction from video frames and employs machine learning techniques for classification.
    - **Paper**: The research paper introduces an algorithm that relies on local binary motion descriptors extracted from video cubes, which are spatio-temporal video sequences. It then uses a codebook to build behavior descriptors for recognizing aggressive behaviors.
3. **Data Representation:**
    - **Proposed System**: In the proposed system, the data is represented using deep learning models, and feature extraction is performed using pre-trained convolutional neural networks (CNNs).
    - **Paper**: In the research paper, the data is represented using binary motion descriptors from local video cubes.
4. **Detection vs. Recognition:**
    - **Proposed System**: The proposed system is primarily designed for violence detection, which typically focuses on identifying specific events or actions, such as assaults or theft.
    - **Paper**: The research paper focuses on recognizing aggressive behaviors in a more general context. It is less concerned with specific event detection but rather capturing various forms of aggression.
5. **Interest Point Detection:**
    - **Paper**: The research paper introduces a technique for detecting interest points based on the Harris corner detector, which is used to extract local features in video.
6. **Feature Extraction:**

- **Proposed System**: The proposed system employs DenseNet121 for feature extraction, which is a CNN designed for image classification and feature learning.
- **Paper**: The research paper uses local binary motion descriptors based on binary cubes and shape and motion features.

7. **Classifier:**
   - **Proposed System**: The proposed system uses a machine learning classifier for violence detection.
   - **Paper**: The research paper employs one-class SVM to recognize aggressive behaviors as outliers.
8. **Evaluation:**
   - **Proposed System**: The proposed system evaluates its performance using metrics such as accuracy, precision, recall, specificity, F1-score, and AUC, focusing on violence detection.
   - **Paper**: The research paper evaluates the accuracy of recognizing aggressive behaviors and provides results based on a dataset that includes both aggressive and non-aggressive behaviors.

Comparing with Research Paper: Violence Detection in Surveillance Videos with
Deep Network using Transfer Learning
Authors: Aqib Mumtaz, Allah Bux Sargano, Zulfiqar Habib

Similarities:

1. **Deep Learning Approach:**
   - **Proposed System:** Both the research paper and our proposed system leverage deep learning techniques. The research paper employs a transfer learning approach with GoogleNet (Inception), while our system utilizes DenseNet121.
   - **Paper:** Both the research paper and our proposed system adopt a deep learning approach. The research paper employs transfer learning with GoogleNet (Inception), while our system uses DenseNet121 for feature extraction.
2. **Datasets:**
   - **Proposed System:** Both the research paper and our proposed system make use of datasets for violence detection. The research paper introduces the Hockey and Movies datasets, which are specifically designed for violent action recognition. In contrast, our system utilizes the UCF crime dataset for training and testing.
   - **Paper:** Both the research paper and our proposed system rely on datasets for violence detection. The research paper

introduces the Hockey and Movies datasets, tailored for recognizing violent actions, while our system employs the UCF crime dataset for training and testing.

3. **Model Training:**
    - **Proposed System:** Both approaches involve training deep learning models. The research paper trains GoogleNet on the provided datasets, while our system trains a model based on DenseNet121.
    - **Paper:** In both approaches, training deep learning models is a common step. The research paper trains GoogleNet on the provided datasets, while our proposed system trains a model based on DenseNet121.
4. **Accuracy Metrics:**
    - **Proposed System:** Both the research paper and our system report accuracy as an evaluation metric. The research paper uses accuracy to assess the model's performance, and our system reports accuracy along with other metrics like precision, recall, specificity, and F1-score.
    - **Paper:** Both the research paper and our system use accuracy as an evaluation metric. The research paper utilizes accuracy to evaluate the model's performance.

Differences:

1. **Model Choice:**
    - **Proposed System:** The research paper uses GoogleNet (Inception) for feature extraction and transfer learning, while our proposed system utilizes DenseNet121. The choice of model architecture differs between the two.
    - **Paper:** In terms of model choice, the research paper uses GoogleNet (Inception) for feature extraction and transfer learning, while our proposed system opts for DenseNet121.
2. **Feature Extraction:**
    - **Proposed System:** In the research paper, the feature extraction is based on the GoogleNet model, which has already learned features from the ImageNet dataset. In contrast, our system uses DenseNet121 for feature extraction.
    - **Paper:** In the research paper, feature extraction relies on the GoogleNet model with pre-learned features from the ImageNet dataset, while our proposed system employs DenseNet121 for this purpose.
3. **Training and Results:**
    - **Proposed System:** The research paper presents results showing that the proposed approach outperforms previous methods, achieving high accuracy on both the Hockey and Movies datasets. Your system provides a detailed analysis of performance metrics such as precision, recall, specificity, and F1-score, along with the ROC_AUC score for individual classes within your dataset.

- **Paper:** The research paper primarily showcases results indicating the superiority of the proposed approach over previous methods, achieving high accuracy on both the Hockey and Movies datasets. In contrast, our proposed system offers a more comprehensive analysis of performance metrics, including precision, recall, specificity, F1-score, and ROC_AUC scores for individual classes.

4. **Detailed Analysis:**
   - **Proposed System:** Your proposed system includes a detailed analysis of metrics for individual classes, which is not as prominent in the research paper. The paper primarily focuses on overall accuracy and comparison with previous methods.
   - **Paper:** The research paper focuses on overall accuracy and the comparison with previous methods but does not provide the same level of detailed analysis for individual classes.
5. **Use of GPU:**
   - **Proposed System:** Your system mentions setting up the GPU environment for accelerated training, which is not explicitly mentioned in the research paper.
   - **Paper:** The research paper does not explicitly mention the use of a GPU environment for model training.

VII. Conclusion

This technical report serves as a comprehensive documentation of our project focused on the development of a violence detection system utilizing computer vision and deep learning techniques. The urgency of real-time intervention to address security threats within public spaces is emphasized, and our project aims to contribute to enhanced public safety by effectively identifying various forms of violence, including physical altercations and aggression.

Throughout our project, we leveraged the UCF Crime Dataset, recognizing its limitations, including class imbalance, a lack of manual annotations, and limited representation of violent events across categories. Despite these challenges, we underscore the importance of robust solutions and diverse training data to improve the accuracy of violence detection.

Our proposed methodology outlines a structured approach for building the violence detection system, from importing essential libraries and configuring hyperparameters to data loading, preprocessing, feature extraction via DenseNet121, and model creation and training. These steps lay the groundwork for a comprehensive video surveillance system capable of identifying both subtle and overt violent actions.

Our project pays particular attention to precision in violence recognition and concerns regarding false positive rates, addressing the complexities of real-world surveillance scenarios, including environmental factors, lighting variations, occlusions, and the presence of multiple individuals.

We also explored the Multi-class AUC Curve as a valuable tool for evaluating the performance of our multi-class classification model. It provides a visual representation of our model's ability to distinguish between different classes, offering insight into its performance.

As we move forward, questions regarding the system's ability to generalize across diverse datasets and scenarios are considered, emphasizing the significance of cross-dataset generalization. The integration of our violence detection system into existing surveillance infrastructure is also acknowledged as a complex process that requires careful planning.

In summary, this report encapsulates the efforts and progress made in our project, aiming to enhance public safety through the application of computer vision and deep learning techniques for violence detection. The adoption of the Multi-class AUC Curve further enriches our understanding of our model's performance, reinforcing our commitment to addressing security concerns in public spaces and contributing to a safer environment.

VIII. References

1. W. Song, D. Zhang, X. Zhao, J. Yu, R. Zheng, and A. Wang, "A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks," in *IEEE Access*, vol. 7, pp. 39172-39179, 2019, doi: 10.1109/ACCESS.2019.2906275.
2. S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, 2017, pp. 1-6, doi:10.1109/AVSS.2017.8078468.
3. P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Colorado Springs, CO, USA, 2016, pp. 30-36, doi:10.1109/AVSS.2016.7738019.
4. A. Jain and D. K. Vishwakarma, "State-of-the-arts Violence Detection using ConvNets," *2020 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2020, pp. 0813-0817, doi: 10.1109/ICCSP48568.2020.9182433.
5. A. Datta, M. Shah, and N. Da Vitoria Lobo, "Person-on-person violence detection in video data," in *2002 International Conference*

on Pattern Recognition, Quebec City, QC, Canada, 2002, pp. 433-438 vol.1, doi: 10.1109/ICPR.2002.1044748.

6. A. Jain and D. K. Vishwakarma, "Deep NeuralNet For Violence Detection Using Motion Features From Dynamic Images," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2020, pp. 826-831, doi:10.1109/ICSSIT48917.2020.9214153.
7. K. Singh, K. Yamini Preethi, K. Vineeth Sai, and C. N. Modi, "Designing an Efficient Framework for Violence Detection in Sensitive Areas using Computer Vision and Machine Learning Techniques," in *2018 Tenth International Conference on Advanced Computing (ICoAC)*, Chennai, India, 2018, pp. 74-79, doi:10.1109/ICoAC44903.2018.8939110.
8. D. G. C. Roman and G. C. Chávez, "Violence Detection and Localization in Surveillance Video," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, Porto de Galinhas, Brazil, 2020, pp. 248-255, doi:10.1109/SIBGRAPI51738.2020.00041.
9. J. Nam, M. Alghoniemy, and A. H. Tewfik, "Audio-visual content-based violent scene characterization," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, Chicago, IL, USA, 1998, pp. 353-357 vol.1, doi:10.1109/ICIP.1998.723496.
10. Y. Lyu and Y. Yang, "Violence Detection Algorithm Based on Local Spatio-temporal Features and Optical Flow," in *2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, Wuhan, China, 2015, pp. 307-311, doi: 10.1109/ICIICII.2015.157.
11. E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar, "Violence Detection in Video Using Computer Vision Techniques," in *Computer Analysis of Images and Patterns. CAIP 2011*, Lecture Notes in Computer Science, vol 6855. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-23678-5_39
12. Chen D, Wactlar H, Chen MY, Gao C, Bharucha A, Hauptmann A. Recognition of aggressive human behavior using binary local motion descriptors. *Annu Int Conf IEEE Eng Med Biol Soc*. 2008;2008:5238-41. doi: 10.1109/IEMBS.2008.4650395
13. Ding, C., Fan, S., Zhu, M., Feng, W., Jia, B. (2014). Violence Detection in Video by Using 3D Convolutional Neural Networks. In: *Advances in Visual Computing. ISVC 2014*, Lecture Notes in Computer Science, vol 8888. Springer, Cham. doi: 10.1007/978-3319-14364-4_53

14. A. Mumtaz, A. B. Sargano, and Z. Habib, "Violence Detection in Surveillance Videos with Deep Network Using Transfer Learning," in *2018 2nd European Conference on Electrical Engineering and Computer Science (EECS)*, Bern, Switzerland, 2018, pp. 558-563, doi: 10.1109/EECS.2018.00109.
15. K. Lloyd, P. L. Rosin, A. D. Marshall and S. C. Moore, "Violent behaviour detection using local trajectory response," *7th International Conference on Imaging for Crime Detection and Prevention (ICDP 2016)*, Madrid, Spain, 2016, pp. 1-6, doi:10.1049/ic.2016.0082.