Violent Activity Detection through Surveillance Camera using Deep Learning

by

Parvez Miah
19101339
Abrar Ahbabul Haque
19301040
Abdullah Al Imran
19101482
MD. Radip Hassan
19101524
Rafiur Rahman
19101461

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

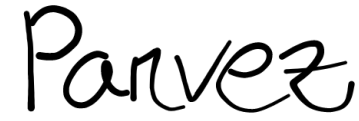
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2023

Declaration

It is hereby declared that

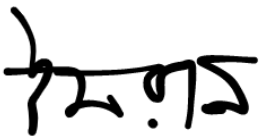
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



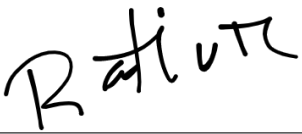
Parvez Miah
19101339

Abrar Ahbabul Haque
19301040



Abdullah Al Imran
19101482

MD. Radip Hassan
19101524



Rafiur Rahman
19101461

Approval

The thesis/project titled "Violent Activity Detection through Surveillance Camera Using Deep Learning" submitted by

1. Parvez Miah (19101339)
2. Abrar Ahbabul Haque (19301040)
3. Abdullah Al Imran (19101482)
4. MD. Radip Hassan (19101524)
5. Rafiur Rahman (19101461)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2023.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

We officially declare that this thesis is supported by our study findings. This research work is free of plagiarism at all research levels. All other information sources have been acknowledged in the text. This thesis has not been submitted, in whole or in part, to any other university or institution for the purpose of degree-granting.

Abstract

Surveillance camera systems have been implemented in most parts of the world to combat the rising rate of criminal activities. In the hopes of making public places safer for everyone, computer vision has also aided in making these systems more sophisticated but reliable and efficient. However, we are yet to make them better. While modern systems are able to record incidents, they often do not do so intelligently in order to make it easier for law reinforcements to respond quickly enough to aid victims or stop more crimes from occurring. Hence for our thesis project, we intend to use computer vision on a surveillance system so that it is able to identify crimes such as physical altercation, harassment, hijacking, snatching, etc. In this model, an action recognition system will be used, where we will be using extracted images from video feeds from multiple sources, and all those sources (cameras) will be centrally connected to a server. The server will be connected to databases containing information about violent activities. Based on the feeds, a signal will be sent to the respective system if a particular activity is detected. This system is mainly based on image processing concepts using different neural networks like MobileNet-V2, ResNet50, and LSTM to match live images with the existing trained system. This model will specifically use to detect criminal activities such as punching, kicking, slapping, and weapon violence, and all these pieces of information will be previously stored in the database. We have also implemented Grad-CAM in an effort to apply model explain ability.

Keywords: Deep Learning; Machine Learning; MobileNet-V2; ResNet50, Surveillance camera; Security; Suspicious Activity; CNN; LSTM; Grad-CAM

Dedication

Every difficult task requires personal effort and encouragement from our elders, especially those closest to our hearts. We dedicate our humble efforts to our loving parents, whose love, devotion, motivation, and nightly prayers have made us deserving of this achievement and honor, as well as all of the outstanding academics we met and learned from while pursuing our Bachelor's degrees, and especially our beloved supervisor, Dr. Md. Golam Rabiul Alam.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our Supervisor Dr. Md. Golam Rabiul Alam sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

$AMQP$ The Advanced Message Queuing Protocol

$AVSS$ Automated Vital Statistics System

$CCTV$ Closed Circuit Television

CNN Convolutional Neural Network

$COAP$ Constrained Application Protocol

$DCNN$ Deep Convolutional Neural Network

DDS Data Distribution Service

DL Deep Learning

$EDCAR$ Elements and Descriptors of Context and Action Representations

GPU Graphics Processing Unit

$Grad - CAM$ Gradient-weighted Class Activation Mapping

HDL Hierarchical Deep Learning

$HTTP$ HyperText Transfer Protocol

$LBPH$ Local Binary Pattern Histogram

$LSTM$ Long Short Term Memory

ML Machine Learning

$MQTT$ MQ Telemetry Transport

NN Neural Networks

$OpenCV$ Open Source Computer Vision

PCA Principal Component Analysis

$RELU$ Rectified Linear Unit

$ResNet$ Residual Neural Network

RNN Recurrent Neural Network

$SMTP$ Simple Mail Transfer Protocol

$SSGrad - CAM$ Spatial Sensitive Gradient-weighted Class Activation Mapping

SVM Support Vector Machine

$XMPP$ Extensible Messaging Presence Protocol

$YOLO$ You Only Look Once

Chapter 1

Introduction

1.1 Background

Over the last century, the human population has grown exponentially, and our infrastructures have become unimaginably sophisticated. While this is, for the most part, beneficial to us as a society, there has also been a significant increase [1] in unfortunate, deprived individuals who choose to commit crimes. Whenever defense and precautions against these crimes are upgraded, criminals find a way to outsmart them. In the modern world, one of these precautions is surveillance camera systems that are designed to record criminal activities in public and private places. While they are valuable tools, they are not utilized to their maximum potential without artificial intelligence. That is where the implementation of computer vision comes in. With it, an automated system can detect an individual's identity or an ongoing crime, suspicious/threatening objects being carried, etc. and if necessary, report it to the authorities immediately. Quick responsiveness to crimes being committed can reduce the number of crimes drastically. Aside from real-time suspect identification and crime detection, CCTV footage is also requested by investigators post-event, where they may need to sift through a massive volume of data to extract any evidence that may help a case. Over the years, the number of instances where investigators have requested CCTV footage for crime identification has increased [5]. Deep learning is becoming more and more popular for monitoring over the years. It was born at the same time as video recording and surveillance. Deep learning is a more advanced type of machine learning that uses layers of algorithms to process data, mimic thinking processes, and build abstractions. It is typically used to visually distinguish objects and understand human speech. Each layer uses the previous layer's output as input to the next layer and passes the information to the next layer. The input layer is the first layer in a network, whereas the output layer is the final. Depending on the problem that needs to be solved, several different types of deep learning algorithms can be used. Some examples include Self-driving cars, Natural Language Processing and Speech Recognition, Computer Vision, Machine Translation, Medical Image Analysis, Big Data, and Data Mining, Video Games, Real-Time Predictive Analytics, Computer Vision and Classification Tasks, etc. The best thing about deep learning is that it can be used in different fields, so it can be used in different applications. The use of computer vision, along with deep learning, can reduce the time required to prevent crimes.

1.2 Problem Statement

We all know the apparent consequences of crimes that occur in public places, especially the ones that remain unsolved due to a lack of information. Suspects may be captured by surveillance cameras spread out across urban landscapes. Information about their whereabouts may be crucial to investigations. Action recognition using classifiers is already a step taken by some developed countries to help solve this. An ongoing public crime may need an immediate response. There are also widespread crimes of rampage and harassment that especially go unnoticed or unattended in our country. We may have the tools to capture these, but we do not have an efficient dealing process. One answer is to use deep learning algorithms to detect indications of an ongoing crime so that at least response personnel could be on the alert. Until recently, machine learning applications for video analysis products have been minimized mainly because of their high complexity and high demand for resources, making such products too expensive for mainstream deployments. However, in recent years, research and progress surrounding the field of machine learning, called deep learning, have increased significantly. Much of the recent interest in deep learning is because of the advancement in graphics processing unit or GPU technology, where vast improvements in parallel processing have allowed computers to efficiently train and test deep learning classifiers. As a result, the research conducted by the scientific community was completed much quicker, which brought them to a point where these algorithms' performances exceed that of most of the traditional machine learning algorithms across several categories. Without AI on surveillance cameras, crimes will not be met with immediate response, more crimes will go unnoticed, and as a result, proper justice will not be served. It only makes our society less safe for everyone.

1.3 Research Contribution

For this research, our goal was to identify possible violent actions from video feeds from a surveillance camera system and detect any identified crimes. We have implemented three models for detecting violent activities; those are MobileNet-v2, ResNet50, and LSTM. Then we compared the results produced by them with each other. We have found multiple research projects on action detection with deep learning algorithms. Compared to the other papers with a similar type of dataset, the highest accuracy that was found was 92.7%, produced by VGG-16. The second-best accuracy was 91% using MobileNet-v2. Although our datasets are non-identical, both our data consist of violent actions like kicking, punching, slapping, etc., as we have seen in the image samples in their paper. We have observed an accuracy of 95% with MobileNetv2, which is higher than both of those results. We have also implemented an explainable AI, Grad-CAM, in our dataset to generate heat maps that show violent activities.

Chapter 2

Literature Review

The first case of using facial recognition for law enforcement was in 2002 [9]. Since then, endeavors have been made to improve it along with several other features that improved surveillance tech has brought over the years. For example, machine learning algorithms have been used in face and motion detection [5]. In table 8 of [5], the authors have referenced several methods to extract facial features, such as SVM, CHM, MCT-based Adaboost face detector, and more. This may be useful for image preprocessing. Each of these uses different properties of the image. The paper also mentions several other moving object and facial component detection methods. While some of the classifiers mentioned are very fast, deep learning algorithms are not mentioned.

In the related works, multiple approaches have been used to detect suspicious behavior from surveillance cameras. The objective of the work was to develop a system that detects any unusual or suspicious occurrences in a video surveillance system and responds quickly.

Deep learning approach along with Convolutional Neural Network (CNN) was used to detect suspicious activity from surveillance footage [1]. Modeling environments, motion detection, categorization of moving objects, tracking, behavior interpretation, description, and synthesis of input from multiple cameras are all steps of computer vision technologies. Deep Neural Networks are one of the most successful designs for solving difficult learning challenges. Deep Learning models automatically extract features and produce high-level representations of image data. This is broader since the feature extraction technique is completely automated. CNN can be trained to directly remember image properties and patterns from input frames. LSTM, or Long Short-Term memory models, can be trained to detect long-term dependencies for video streams. The LSTM network can remember information. This model's accuracy increases with the number of iterations it is applied, so fewer iterations may result in lower accuracies. Moreover, this system was proposed for limited private places but can be used in any scenario consisting of suspicious activity.

Automated face recognition was proposed to identify criminal activity from CCTV footage [4]. The system proposes Criminal Detection by Face Recognition in this project. Facial recognition can be used for two different types of tasks: verifying and

identifying. Face verification means the application matches a person's face with a stored image of his/her face to verify that person's identity. It makes a 1:1 comparison. Face identification, on the other hand, is a 1: N issue that compares two query face images. The application maintains track of photos from various sides of the face here. Based on this record number, the algorithm downloads the suspect's record, which compares and finds a match of more than 90 percent, indicating that a crime has been discovered.

Many security recordings from the event location were utilized in the training dataset. A frame rate of one frame per second is used to convert video to frames. Keyframes from the films are extracted and kept in files labeled Suspicious-Violent and Non-Suspicious. This serves as the model's training data. The Training dataset has a total of 251 frames. Similarly, a Validation dataset with Suspicious-Violent and non-suspicious category frames is constructed. There are 109 photos in the validation set. CNN model is trained using this dataset, which contains 70 percent Training and 30 percent Validation. Convolution is a technique for extracting characteristics from photos. To create convolution maps, the input picture is processed through many filters/kernels of 3×3 size in the first step. 32 filters were employed in this strategy. The convolutional layer is given an [W1*H1] input picture and a filter that eliminates certain properties from pictures, preferably [3*3]. Convolution is achieved by multiplying a two-dimensional array of weights with an array of input data (the image) (the filter or kernel). RELU employs an activation function in the convolutional layer, such as the max(0, x) function, to overcome element-wise non-linearity. The CNN model guesses which category each frame belongs to by splitting the movie into frames (suspect or not). Suspicious frames comprise activities/actions such as robbery, theft, and so on, and they are recorded in a directory. To build a video summary, these suspicious frames contain priority data and are played at a rate of 2 frames per second. The produced short video is sent to the host via the SMTP protocol. To build an SMTP session in Python, use the smtplib module. The SMTP object has been defined and may send mail to any host. When tested on various films, the proposed algorithm shows exceptional accuracy and compresses long videos to a large amount, resulting in a summary video as output [3].

Gait analysis has been used for criminal activity identification based on motion capture [8]. There are many sorts of gaits, including walking, crawling, jogging, running, and skipping. These human gaits can be divided into gait cycles, and these gait cycles consist of three tasks, they are weight acceptance, single limb support, and limb advancement. The proposed methodology performs identification in four different steps. First, M is the motion sequence for each identity (samples and suspects). Each of these motion sequences may take a different path from the others. As a result, all M motion sequences require a normalization transformation to be tested, matched and confirmed under the same condition. Translation normalization and rotation normalization are the two steps of normalization. M(norm), the normalized motion sequence, is still high-dimensional data challenging to examine (curse of dimensionality). The multidimensionality of the data is reduced using Principal Component Analysis (PCA). Only the most important data components are evaluated when employing the PCA approach, while the less important ones are

ignored. The Euclidean distance is a method for calculating the distance between two locations in space. The similarity of the two motion sequences may then be checked. When the Euclidean distance between the suspect data and the sample data is the smallest, it is considered matched. A threshold value defines the matched or unmatched comparison between two data sets. According to the research, ankle movement is the most important pattern that may differentiate a person from others. Work on the specific motion gait that reveals the most important pattern in assessing the joint position, angles, and velocity can be done in the future.

In [2], the authors have tried to improve face detection using Haar classifiers by training the model to detect and identify faces at different angles, which a generic Haar classifier cannot do. Since it is a very fast algorithm, it can be used in real-time. However, this paper does not include the tracking of identified suspects, which we could add. Once a suspect is identified, a group of cameras available in the same vicinity will try to track the suspect's movement. We could also try to use deep learning models instead of Haar classifier and compare accuracy and speed.

The research work [10] proposes efficient spatio-temporal modeling methods for real-time violence recognition. Rather than 3D-CNN, this model uses 2D-CNN with C3D and I3D, adding a time axis to 2D-CNNs. Here frame grouping is proposed, which gives spatio-temporal representation in the video to 2D-CNNs. In this model, MSM captures desired regions of the feature map, and the T-SE block highlights the period. Though 3D-CNNs can do the exact thing to some extent, in this model, the computational complexity, cost, and need for space are much less. And it showed the state of the art performance in Mobile Net-V3 and Efficient Net-B0.

The research work [11] reviews 220 journal-based publications that tried to demonstrate different trends, datasets, methods, and frameworks used in Intelligence Video Surveillance. Here, they talked about different indoor and outdoor situations where the kyungroul framework, smart surveillance frames, Rise frameworks, EDCAR frameworks, etc., were used for identifying crimes, accidents, traffic see-over, motion detection, activity analysis, classifying vehicles and humans by using different datasets like AVSS, BEHAVE, CalTech, CAVIAR, ViSOR, etc. Most of these models used deep learning, support vector machines, and fuzzy logic. Through this work, we can learn that even with our current research ability, there are problems with detection in many luminous regions, weather change, and shadow. And the camera's movement can cause problems in detecting an individual in a group. And using a fast and accurate method for applying segmentation techniques can help in minimizing the mentioned problems.

This research work [12] is a surveillance system with face recognition features where the authors used a system development model and RND methodology. This model uses three face recognition algorithms - LBPH, FisherFace, and Eigenface. Among all three, LBPH showed most accuracy rate, which is 95.92 percent. The Validation process was performed following ISO 25010-2015, and IT experts from Philippine National Police rated it 4.82, which can be considered acceptable. This system has a scope for redevelopment, such as skin color variation, facial expression, occlusion, and human face recognition.

The research work [13] tries to present different approaches which can fool surveillance systems. First, it shows how a small patch can fool the system into not recognizing the system. Here the authors used YOLOv2. Though the system was trained for millions of data (CNNs R-CNNs), it failed at different stages. The Goal was to generate printable adversarial patches, and they used optimization processes on image pixels.

This research paper [14] presented videos with or without incidents through tagging and matching. The first step is manually generated tags. A technology that recognizes the incident and behavioral pattern from the given data and matches different tagged moments or individuals. And matching can be done by following two types of re-identification methods. First, the automatic alert system will give alerts if good matches are found, and it will only notify the system when any action is required. Second, a semi-automated search doesn't tag matches like the previous one but compares all images and results through the user. Here, the results are decided by continuously sorting the decreasing similarities.

This research paper [15] is about the choice of application layer protocols for next-generation video surveillance using the internet of video things. In recent years the cost of CCTV has become less than the cost of monitoring them. So, it can be reduced using the genetic architecture of an IoVT-based VSS. Different application layer protocols are used, such as MQTT, HTTP, COAP, AMQP, DDS, and XMPP. To experiment, an IoVT source node was used to capture the image and an IoVT sink node to send them. The source node is used as an edge or fog, and the sink node works as fog or cloud. The parameters for the experiment are latency, throughput, CPU and memory usage, packet loss, average bandwidth consumption, overhead, and energy consumption. Firstly, the experiment was performed in single-node image data. Secondly, it's experimented with single-node video data and finally on multi-node latency. After seeing the performance of all the protocols, MQTT outperformed all the other protocols, and it is used in a 320×240 resolution as the standard dataset.

In [6], the authors have proposed more modern approaches to detect unscrupulous human behavior. In order to do this, they first extract specific properties from scenes to reduce variability. They include basic properties like detecting people and identifying specific actions using the 3-D skeleton, object classification, gender recognition, etc. However, the more important ones (important because they are closely related to our intentions) were hand gesture detection and face detection. In the paper, hand gesture recognition may be used for a wide range of symbols produced by human hands, but for our purpose, we only need our model to be trained to recognize a few gestures pointed at the camera. [8] has covered hand gestures in detail. Although the authors did not mention using deep learning algorithms like CNN for face detection, they make a good point about the case where a person is wearing a mask to hide their identity. This could be useful to us because, in post-events, the footage could be used to gather clues, such as apparent height, clothing, skin color, etc.

[7] has used Convolutional Neural Networks along with several other deep learning classifiers in their model to detect criminal activities such as car theft and physical assault. The accuracy of this model is high across large datasets. Hence it works in large urban areas to detect even minor crimes. Not only that but because of the use of DCNN and HDL, this system is said to improve over time.

In[17], the authors have tried to solve the spatial sensitivity issue of Grad-CAM. Here the authors proposed using Spatial Sensitive Grad-CAM (SSGrad-CAM) for generating heatmap for object detection. By computing space maps by normalizing the gradient magnitude, SSGrad-CAM alters the heatmap produced by Grad-CAM. This way, spatial sensitivity may be included in SSGrad-CAM, emphasizing the value of both features and space. They demonstrate through studies that SSGrad-CAM can provide suitable heatmaps for detection results and that it can also generate when models detect objects by paying close attention to their periphery areas.

Chapter 3

Methodology

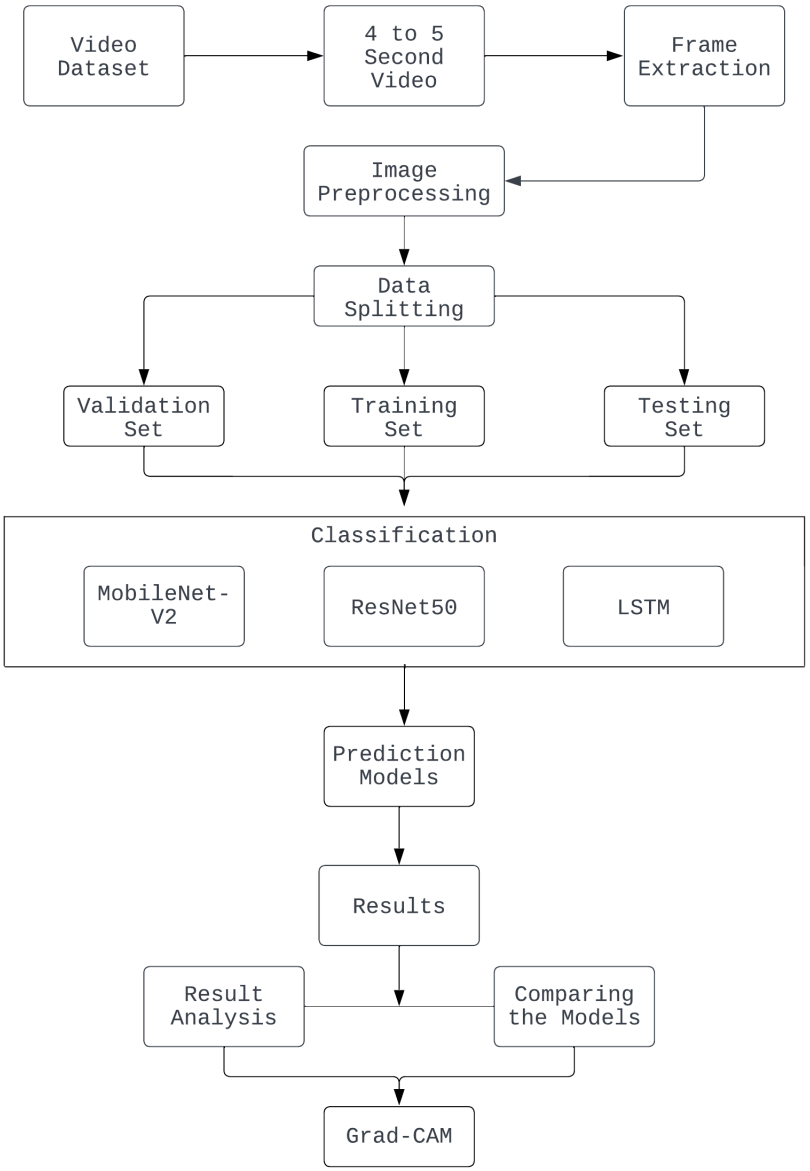


Figure 3.1: Top Level Overview of the Proposed Methodology

The primary purpose of our project is to correctly predict from an external video feed whether any action in the videos is violent or nonviolent, so we have two classes. We have used a dataset that we found online. There are one set of violent activities and another set of nonviolent activities. Violent activities include kicking, punching, wrestling, etc., whereas nonviolent ones have regular activities such as running, hugging, verbal interactions, and so on. Each video is around 5 seconds long. We have decided to train and test 3 models and compare their results. But for that, the images are first extracted from the videos and then resized to a fixed size. The image data is split into training,validation and testing sets. Training set is fed into the model several times to improve its accuracy, and the testing set consists of unseen data and is used to measure the performance of the model after each epoch of training. The test accuracy determines how good a model is at classifying violent and nonviolent actions for real-world applications. The results for each model are gathered and compared with each other. We have also applied Grad-CAM for model explain-ability.

3.1 Dataset Description

We primarily focused on collecting data for violent and nonviolent activity. These data were collected from internet resources such as kaggle (Elesawy, M. (2019, April 27). Real life violence situations dataset. Kaggle. Retrieved January 23, 2023, from https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset). Our dataset consists of 600 Videos, of which 300 have violent actions, and the other 300 are nonviolent in nature. The average duration of each clip was approximately 5 seconds, indicating that this was the only time frame containing genuine action. Each video is split into images for training the model.

3.1.1 Data Sample

Figure 3.2: Violent Data Sample

Figure 3.3: Non-violent Data Sample

The pictures shown in figure 3.2 show samples of three instances of violent activities. Each picture was taken from an approximately 5-second long video. The pictures in figure 3.3 similarly show nonviolent actions taken from three videos. We have worked with such violent videos and nonviolent videos. For this project, we have two classes of data, as shown in the sample figures above. After extracting frames from the videos and storing them as images, we got our desired form for the implementation. After training our datasets, we can find that the system accurately identifies two classes. In the images in figure 3.2, a physical altercation between two human beings is identified as a violent act. On the other hand, the images in figure 3.3 depict regular human interactions as nonviolent. These are images of people talking or playing nonviolent games such as chess. So when these datasets are fed to the system, it will identify and label them as violent or safe.

3.1.2 Training Set

A training set is a portion of a data set used to fit a model for predicting or classifying values that are known in the training set but unknown in other future data. The step in which examples tagged with machine learning algorithm results or output labels is fed into the system. This algorithm enables machines to solve problems based on past observations. The great thing about ML models is that they improve over time as they're exposed to relevant training data. The data training process can be divided into multiple steps, first needing to feed the training input data to

the algorithm. Then tag training data with the desired output. Next, the model transforms the training data into numbers representing data features. Finally, algorithms are trained to associate feature vectors with tags based on manually tagged samples, then learn to make predictions when processing unseen data.

3.1.3 Testing Set

In machine learning, a test set is a secondary or tertiary data set used to test a program learned on a training set of data. As the algorithm converges to increase performance, it may pick up on some training set characteristics utilizing a series of real-world examples. Positive results will boost the algorithm's credibility in the real world for an unidentified test collection. In addition, it offers a neutral measure of the final model's performance in precision, accuracy, etc.

3.1.4 Validation Set

The validation set is used to test a particular model. However, this is done frequently. Machine learning engineers use this information to adjust the model hyperparameters. The model occasionally encounters this input but never "Learns" from it. Instead, we update higher-level hyperparameters using the findings from the validation set. So a model is indirectly influenced by the validation set. The Development Set or the Dev Set is another name for the Validation Set. Given that this dataset aids in the development phase of the model, this makes it logical.

3.2 Data Pre-processing

Figure 3.1, shows a summary of the methodology. The videos are first converted to frames for preprocessing and feeding into the model. Next, the dataset is split into training and testing sets. The model is then trained with the training set to identify violent activities and validated with the testing set. The model is now ready to take in new videos for violent activity detection. Any new video needs to be divided into frames for feeding into the test model to get the results of the models.

3.2.1 Frame Extraction

For image extraction, we used the read() method from the VideoCapture class of the OpenCV library. Each video is around 5 seconds long, with a framerate of 30fps. Not every frame was extracted; only every seventh frame was taken from each video as part of preprocessing to avoid the possibility of duplicate images.

3.2.2 Image Resizing

In our work, we will be using pre-trained models. Hence, Each image from the extracted frames needs to be resized to a fixed size of 128 x 128 during the network training process. We used TensorFlow. TensorFlow is a complete open-source machine learning framework. With its rich ecosystem of tools, libraries, and community resources, machine learning allows researchers to push the state of the art forward. Simultaneously, programmers can build and release programs that make

use of ML. The Google Brain team, comprised of researchers and engineers from Google's Machine Intelligence Research division, developed TensorFlow in order to do research in machine learning and deep neural networks. The applicability of this system extends to many more fields. TensorFlow offers guides, illustrations, and other resources to hasten model development and produce scalable ML solutions.

Figure 3.4: Resized Images 128 x 128

3.3 Model Specification

3.3.1 MobileNet-V2

MobileNet-V2 is a version of CNN designed for mobile devices, so it is less resource intensive. It is built on an inverted residual structure where residual connections connect the bottleneck layers. In addition, in the intermediate expansion layer, lightweight depthwise convolutions are used as a source of non-linearity to filter features. The architecture of MobileNet-V2 includes a 32-filter initial fully convolution layer and 19 additional bottleneck layers. MobileNet-V2 used three convolutional layers in the block. The final two layers consist of depth-wise convolution filters for the inputs and 1x1 point-wise convolution. Nonetheless, this 1x1 layer now serves a different purpose. It decreases the number of channels. This layer is currently referred to as the projection layer. The new block is the first layer. Likewise, it is a 1x1 convolution. It increases the number of data channels before passing them to the subsequent layer. This layer is called the expansion layer. This layer is the inverse of the projection layer.

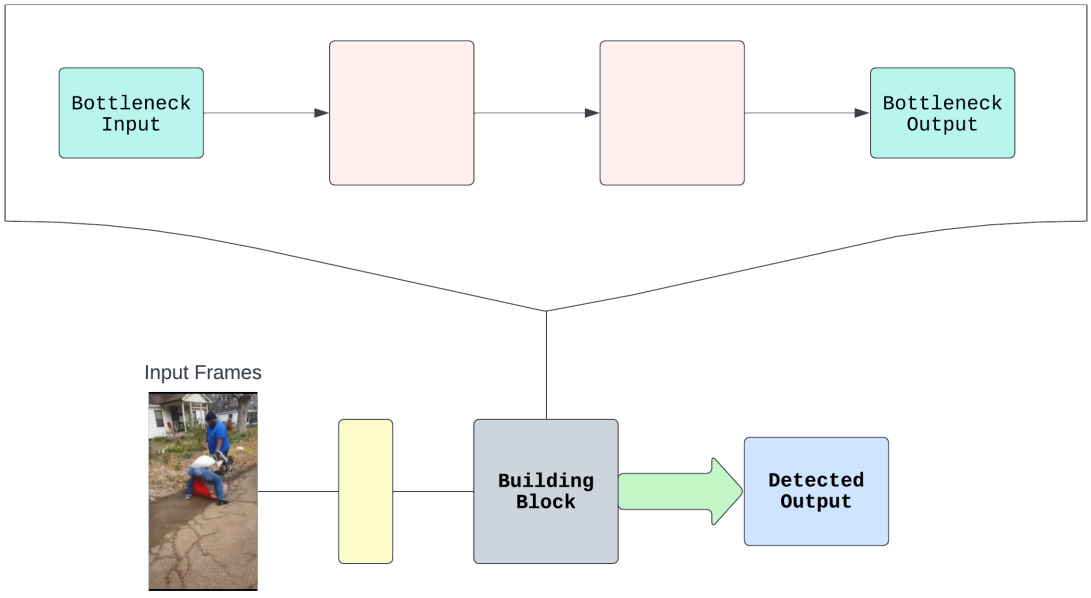


Figure 3.5: MobileNet-V2 Architecture

3.3.2 ResNet50

A deep learning model called Residual Network (ResNet) is applied in computer vision applications. It is an architecture for a convolutional neural network (CNN) that can accommodate hundreds or thousands of convolutional layers. Previous CNN designs could only support a few layers, which had a negative impact on performance. However, researchers encountered the "vanishing gradient" issue when they added more layers. In order to solve this problem of the vanishing gradient, this Resnet Architecture is introduced. It is called Residual Blocks. Gradient descent is used in the backpropagation technique to train neural networks, which lowers the loss function

and identifies the weights that minimize it. Performance saturates or declines with each additional layer if there are too many layers, and repeated multiplications will eventually cause the gradient to "disappear." ResNet's "skip connections" function provides an innovative approach to resolving the vanishing gradient problem. We stack, skip, and reuse activations from the previous layer to create multiple identity mappings (convolutional layers that initially do nothing; ResNet). Skipping speeds up first-pass training since it reduces the network's size by eliminating layers. After the network has been constrained, all layers are stretched, allowing the remnant sections to further investigate the input image's feature space. Most ResNet models skip two or three layers simultaneously, with batch normalization and nonlinearity in between. HighwayNets, a more sophisticated ResNet architecture, can learn "skip weights," which dynamically decide how many layers to skip.

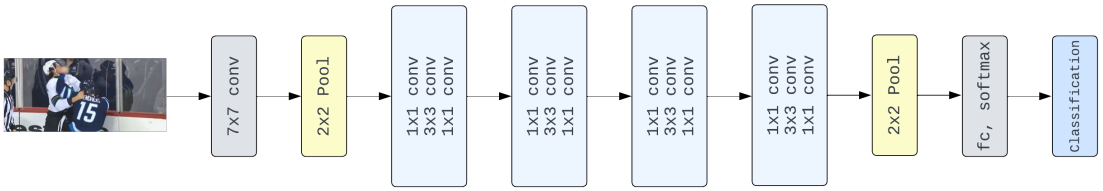


Figure 3.6: ResNet50 Architecture

3.3.3 LSTM

A recurrent neural network is one of the effective techniques for real-time contextual data (RNN). RNNs may handle input sequences of any length since they have feedback connections between nodes and layers, in contrast to feedforward neural networks. However, training the simple RNN can be difficult. The creation of "Long Short-Term Memory" has been shown to solve the disappearing or exploding gradient difficulties caused by the gradient-based weight update techniques employed in RNNs (LSTM). A unique variety of RNN, known as LSTM, has multiplicative gates and internal memory. Prediction, pattern classification, various sorts of recognition, analysis, and even sequence synthesis are just a few of the activities that LSTM neural networks may carry out.

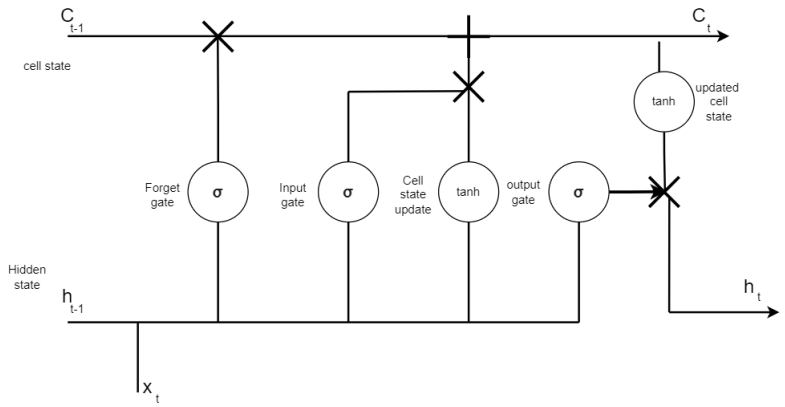


Figure 3.7: LSTM Architecture

3.4 Experimental Setup

Through our research, we have preprocessed images extracted from violent and nonviolent videos and split the preprocessed data into three sets- training, validation, and testing sets. The images were set to a size of 128 by 128 and then fed into the three models for 40 iterations of training. The performance metrics(precision, recall, and F1) from each model were analyzed and compared with each other. To ensure the fairness of this comparison, we ran all the training sessions on the same setup. We have used Google Colab's GPU runtime, which featured a NVIDIA Tesla T4 GPU, Intel Xeon CPU @ 2.00 GHz, 12.68 GB of RAM, and a 64-bit version of Linux.

Chapter 4

Result and Discussion

4.1 Implementing Proposed Architecture

In our architecture, we feed preprocessed data to already-trained architecture. In this section, the input dataset shape was identical for all pre-trained models with which we worked. We received a result containing values that required reformatting. Therefore, we standardized them for our model. We calculated accuracy, Confusion matrix, precision, recall, and F1 score for each pre-trained model based on the model's final output.

4.2 Performance Metrics

Confusion Matrix:
A confusion matrix summarizes the outcomes of predictions made for categorization issues. Confusion matrix criteria were used in our study to evaluate the performance of MobileNet-V2. Classification accuracy can be used to gauge effectiveness as long as the dataset contains an equal number of samples from each class. When addressing an unequal dataset, a more extensive set of performance indicators should be used. In this case, confusion matrices are utilized. A table that lists accurate and inaccurate classifications is called a confusion matrix. The confusion matrix values are used to construct a number of metrics that quantify the classifier's performance. Classification accuracy is measured by the ratio of correctly identified samples to the total amount of data.
Precision:
Precision, or the success rate of the model's predictions, is one metric for measuring a machine learning model's efficacy. Precision is determined by dividing the total number of positively identified samples by the proportion of accurately classified positive samples (True Positive) (either correctly or incorrectly).

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.1}$$

The model's capacity to identify positive samples is measured by recall. More positive samples are found when recall is higher. Recall is calculated as the ratio of positive samples correctly classified as positive to the total number of positive samples. Recall measures the model's ability to detect positive samples.

Recall:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.2}$$

Comparing precision and recall demonstrates the trade-off between the two at various thresholds. High scores for both indicate that the classifier is producing accurate (high precision) results and mostly positive (high recall) results. Compared to the training labels, the majority of the labels predicted by a system with a high recall but low precision provide various outcomes. In contrast, a system with high precision but low recall produces very few results, yet the majority of its projected labels correspond to the training labels. A system with perfect precision and recall will accurately classify a huge number of outcomes.

F1-score:

The f1-score is one of the most crucial machine learning assessment measures. Combining accuracy and recall, two metrics that would generally compete, it elegantly summarizes a model's prediction ability. It is used to score a binary classification system that classifies examples as 'positive' or 'negative.' The range is from 1 to 0. For models, higher f1 scores are generally better.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.3}$$

4.3 MobileNet-V2 Implementation and Result

The trained model is tested with two random unseen videos in order for us to see some qualitative results. At first, the 5-second-long violence video was input into the model. Then the model gives us all the frames in the video, each one labeled violent, which means it has correctly identified the class of the video. For example, one such frame is shown below in figure 4.1.

Figure 4.1: Violent Frame (MobileNet-V2)

Similarly, the second video, which is a nonviolent video, is input into the model. One of the frames that it has given us is shown below in figure 4.2. Each frame has been labeled "safe."

Figure 4.2: Nonviolent Frame (MobileNet-V2)

4.3.1 MobileNet-V2 Accuracy and Loss Curves

During the MobileNet-V2 model training period, the training curves were generated, which included both train and validation loss curves. Both types kept decreasing with passing epochs in figure 4.3.

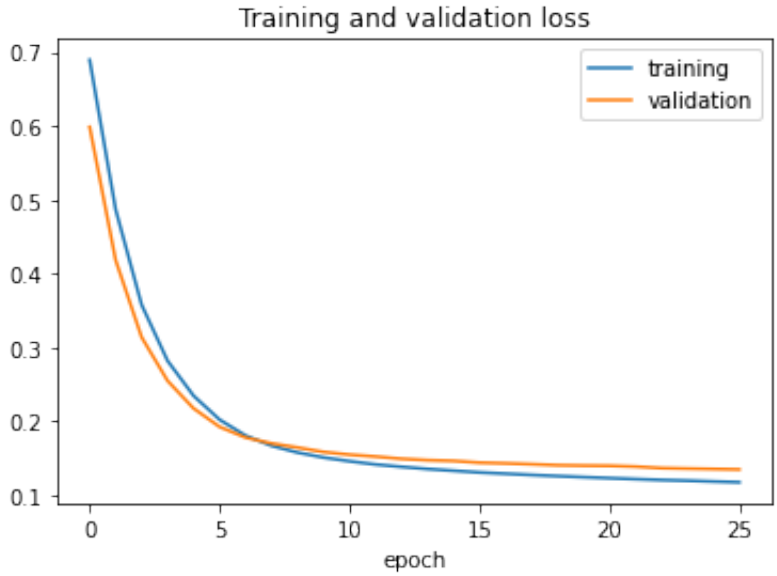


Figure 4.3: Training and Validation Loss of MobileNet-V2

And in figure 4.4, from curves of training and validation accuracy, we can see both keep increasing with the increasing number of epochs.

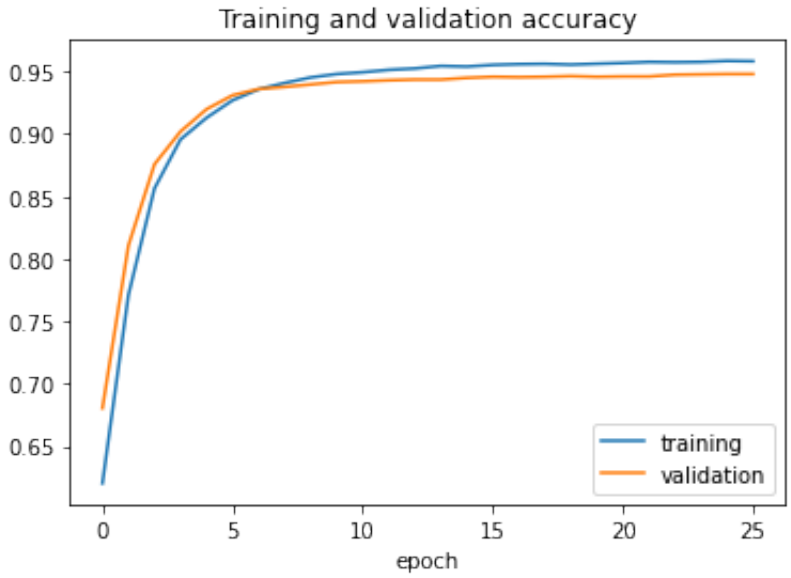


Figure 4.4: Training and Validation Accuracy of MobileNet-V2

It can be observed from figure 4.4 that the training and validation accuracy increases exponentially in the first five epochs, but after that, the rate of increase in accuracy decreases drastically. The curve plateaus at around the 10th epoch.

4.3.2 Confusion Matrix

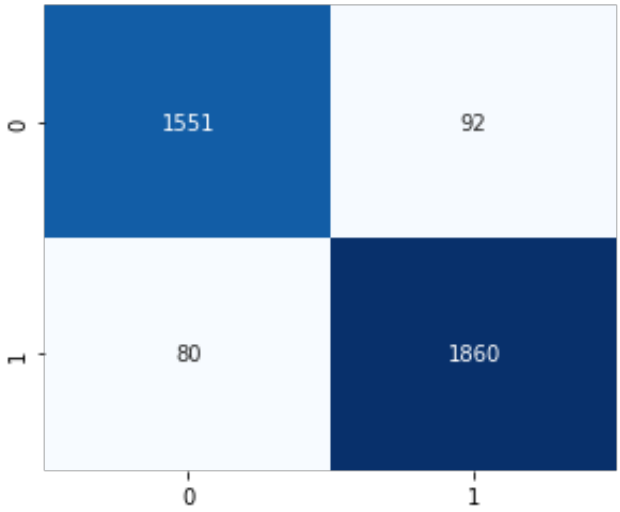


Figure 4.5: Confusion Matrix of MobileNet-V2

In our confusion matrix in figure 4.5, class '0' represents nonviolence, and '1' represents violence. Out of 1952 violent frames, the model was able to predict 1860 frames of violent images accurately (true positives) but made 92 wrong predictions (or false positives), i.e., 92 frames were detected as nonviolent. Similarly, out of 1631 nonviolent frames, 80 were incorrectly detected as violent, while the remaining 1551 were correctly predicted.

4.3.3 Precision, Recall & F1 Score Interpretation

Precision:

Table 4.1: MobileNet-V2 Precision

Violent	**Non-Violent**
0.95	0.95

Table 4.1 shows that the precision rate is 0.95 for both Non-violent and Violent frames.

Recall:

Table 4.2: MobileNet-V2 Recall

Violent	**Non-Violent**
0.96	0.94

The ratio for non-violent predicted frames is 0.94, and the ratio is 0.96 for violent predicted frames. The average them is 0.95.
F1-score:

Table 4.3: MobileNet-V2 F1-score

Violent	**Non-Violent**
0.96	0.95

Table 4.3 shows that the f1 score for non-violent frame type is 0.95, and the score is 0.96 for violent predicted frames.

4.4 ResNet50 Implementation and Result

In order to obtain quality findings for ResNet50, we evaluated two random sets of videos. The model then provides all violently classified frames from the video, indicating that it properly identified the video's category. An example of such a frame is displayed below in figure 4.6.

Figure 4.6: Violent Frame (ResNet50)

Conversely, the second video, which is nonviolent, is fed into the model. One of the frames it has given us is shown in figure 4.7 below. Each frame is marked as "safe."

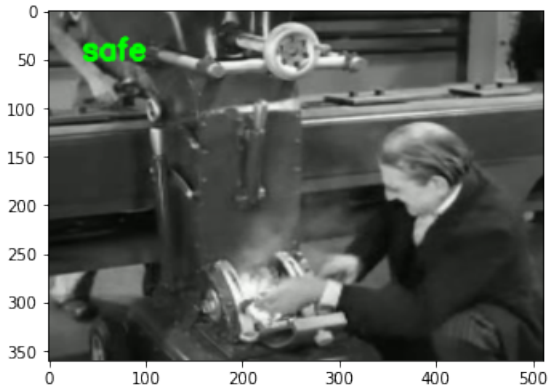


Figure 4.7: Nonviolent Frame (ResNet50)

4.4.1 ResNet50 Accuracy and Loss Curves

During the training phase of the ResNet50 model, training curves comprising both training and validation loss curves were generated. Both categories continued to reduce as epochs passed in figure 4.8.

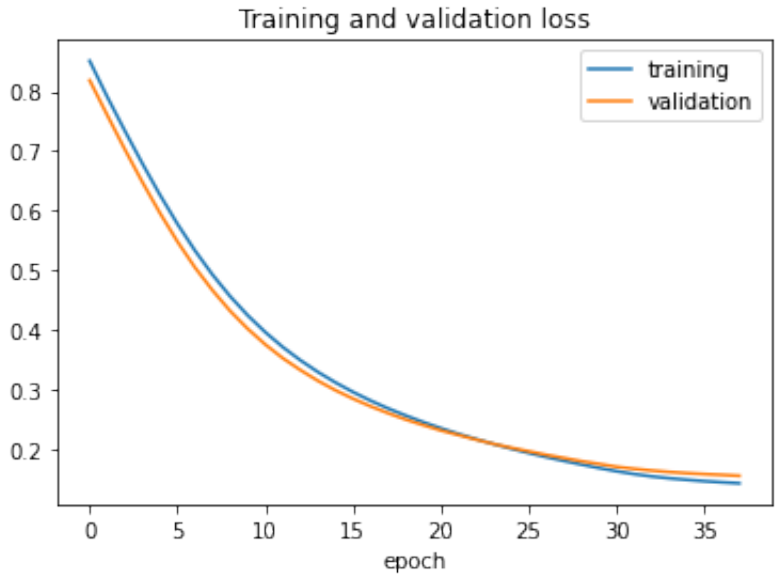


Figure 4.8: Training and Validation Loss of ResNet50

As seen in figure 4.9, the training and validation accuracy curves continue to rise as the number of epochs increases.

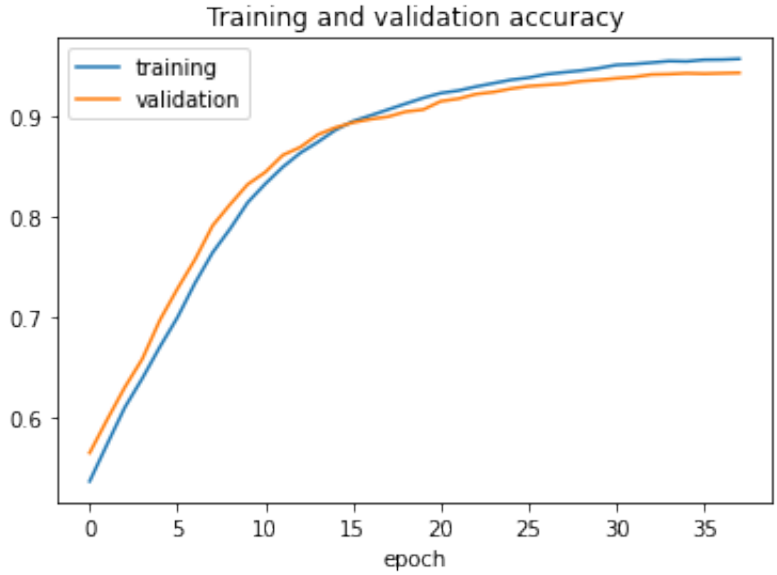


Figure 4.9: Training and Validation Accuracy of ResNet50

4.4.2 Confusion Matrix

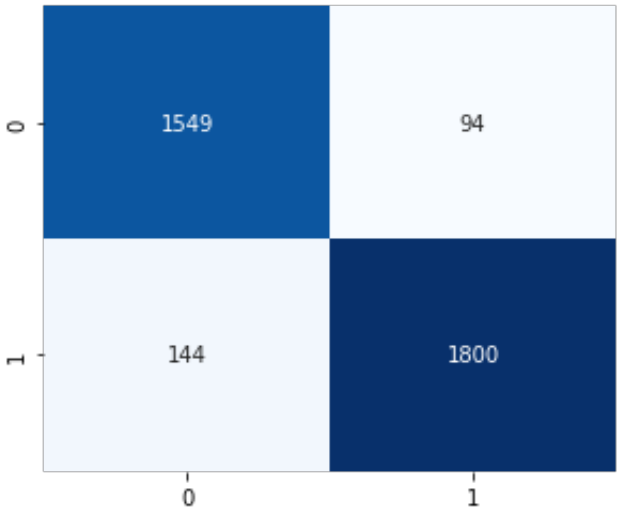


Figure 4.10: Confusion Matrix of ResNet50

In our confusion matrix figure 4.10, class '0' represents nonviolent, and '1' represents violent. Out of 1894 violent frames, the model was able to accurately predict 1800 frames of violent actions, but 94 wrong predictions, i.e., 94 frames were detected as nonviolent. Similarly, out of 1693 nonviolent frames, 144 were incorrectly detected as violent, while the remaining 1549 were correctly predicted.

4.4.3 Precision, Recall & F1 Score Interpretation

Precision:

Table 4.4: ResNet50 Precision

Violent	**Non-Violent**
0.95	0.91

Table 4.4 shows that the precision rate is 0.95 for Violent frames & 0.91 for Non-Violent frames.

Recall:

Table 4.5: ResNet50 Recall

Violent	**Non-Violent**
0.93	0.94

The ratio for non-violent predicted frames is 0.94, and the ratio is 0.93 for violent predicted frames, as shown in Table 4.5.

F1-score:

Table 4.6: ResNet50 F1-score

Violent	Non-Violent
0.94	0.93

Table 4.6 shows that the f1 score for the non-violent frame type is 0.93, and the score is 0.94 for violent predicted frames.

4.5 LSTM Implementation and Result

To acquire consistent results for LSTM, we analyzed two random sets of videos. The model then gives all violently labeled video frames, demonstrating that it accurately determined the video's categorization. Below is an example of such a frame in figure 4.11.

Figure 4.11: Violent Frame (LSTM)

The second video, which is nonviolent, on the other hand, is fed into the model. The frame it provided us with is depicted in figure 4.12 below. Each frame has a "safe" label on it.

Figure 4.12: Nonviolent Frame (LSTM)

4.5.1 LSTM Accuracy and Loss Curves

During the training phase of the LSTM model, training curves comprising both training and validation loss curves were generated. Both training and validation losses continue to reduce with each passing epoch in figure 4.13.

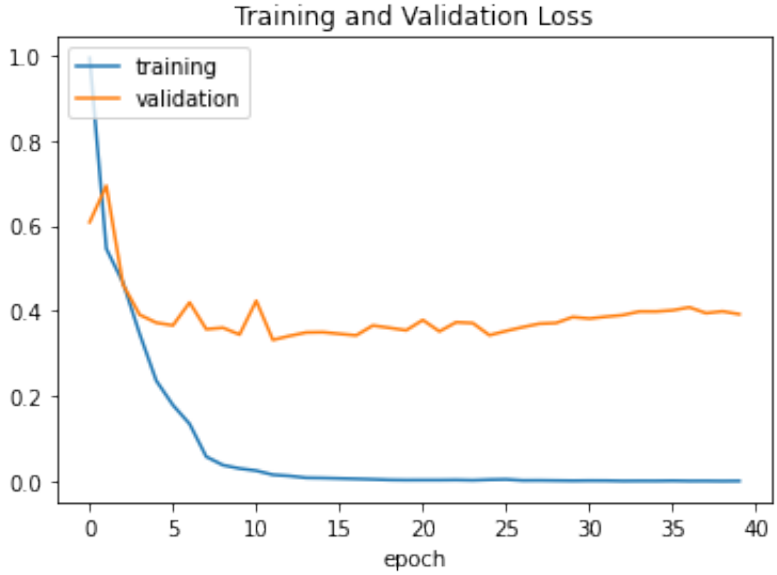


Figure 4.13: Training and Validation Loss of LSTM

Unlike ResNet50 and MobileNetv2, the training and validation accuracy graph of LSTM shown in figure 4.14, is very inconsistent overall. The training accuracy reached 100% after about 7 epochs, which is not characteristic of an unbiased model, but the validation accuracy is very inconsistent from the starting epoch. Not only does it follow a zig-zag pattern at the beginning and fluctuates afterwards, but it is also much lower than the training accuracy. These observations suggest that the model is overfitting.

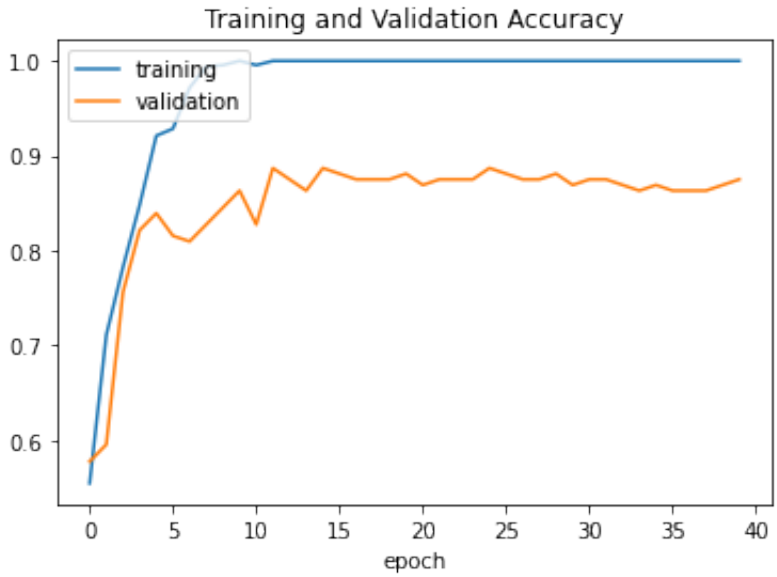


Figure 4.14: Training and Validation Accuracy of LSTM

4.5.2 Confusion Matrix

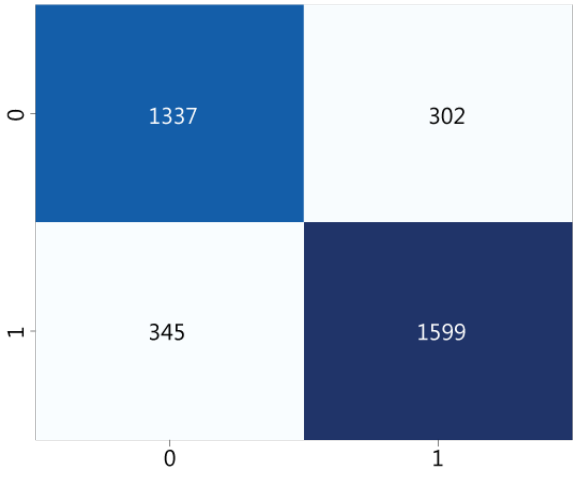


Figure 4.15: Confusion Matrix of LSTM

In our confusion matrix figure 4.15, class '0' represents nonviolent, and '1' represents violent. Out of 1901 violent frames, the model was able to accurately predict 1599 frames of violent actions but made 302 wrong predictions, i.e., 302 frames were detected as nonviolent. Similarly, out of 1682 nonviolent frames, 345 were incorrectly detected as violent, while the remaining 1337 were correctly predicted.

4.5.3 Precision, Recall, F1 Score & Accuracy Interpretation

Precision:

Table 4.7: LSTM Precision

Violent	**Non-Violent**
0.84	0.79

Table 4.7 shows that the precision rate is 0.84 for Violent frames & 0.79 for Non-Violent frames.

Recall:

Table 4.8: LSTM Recall

Violent	**Non-Violent**
0.79	0.85

The ratio for non-violent predicted frames is 0.85, and the ratio is 0.79 for violent predicted frames, as shown in Table 4.8.

F1-score:

Table 4.9: LSTM F1-score

Violent	**Non-Violent**
0.81	0.82

Table 4.9 shows that the f1 score for non-violent frame type is 0.82, and the score is 0.81 for violent predicted frames.

4.6 Comparison Between Different Models

Table 4.10: Comparison Table

Model	**Precision**	**Recall**	**F1-score**
MobileNet-V2	0.95	0.96	0.96
ResNet50	0.95	0.93	0.94
LSTM	0.84	0.79	0.81

These results have been taken from the confusion matrices after training each model. For each precision, recall, and f1 score, MobileNet-V2 showed the best results, whereas LSTM had the least scores. Precision is a metric that tells us the percentage of true positives detected from the test set. MobileNet-V2 and ResNet50

have precisions of 95%, but LSTM has much lower at 84%. Recall tells us how well the model is able to identify true positives, and here we see that MobileNet-V2 has the best recall out of the three models. F1 score is the weighted average of precision and recall, so again, MobileNet-V2 also has the highest score at 96%, while LSTM has the least at 81%.

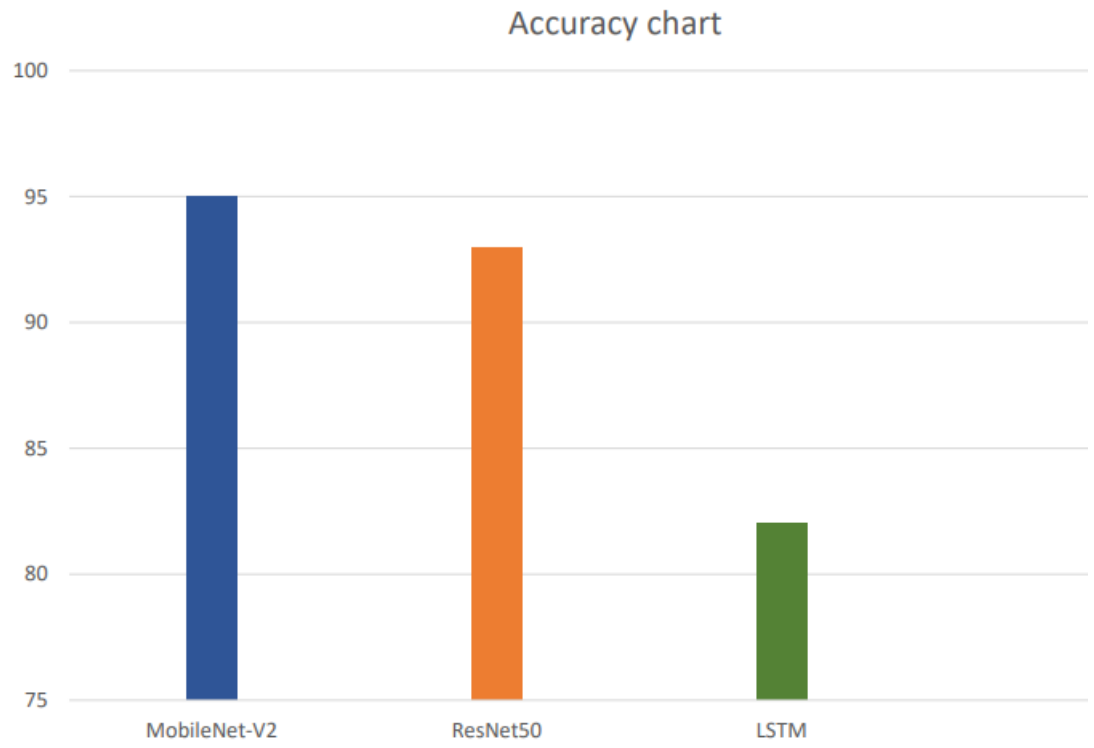


Figure 4.16: Accuracy Comparison

Here the testing accuracies of each architecture have been compared. It is observed that MobileNet-V2 has the highest accuracy at 95%, while LSTM's prediction accuracy was the least at 82%. ResNet50 had a test accuracy of 93%. We can see that MobileNet-V2 and ResNet50 provide a high accuracy than LSTM for our image dataset. While using fewer computing resources and memory storage than the other two models, MobileNet-V2 offers greater accuracy. Because it effectively learns all the parameters from early activations deeper in the network, ResNet50 consumes more memory storage but performs better than LSTM. LSTM has shortcomings, as its Temporal Dependencies and Limited Context Window Size give lower accuracy than the other two.

Chapter 5

Explainable AI: Grad-CAM

5.1 Grad-CAM

Grad-CAM, short for Gradient-weighted Class Activation Mapping, is a technique for visualizing which regions of an image a deep learning model is paying attention to when making a prediction. It is used to understand the decision made by the model by visualizing which part of the image the model used to make the prediction. It is called explainable AI because it makes the model's decision-making process more transparent and interpretable by highlighting the image regions the model uses to make a prediction. This can be useful for debugging and understanding the model's performance, as well as for building trust with users of the model. Frequently, a heat map may be used to demonstrate how the model came to decide on a particular class. We qualitatively analyzed the networks' identified regions of interest by showing heatmaps at various levels of three proposed networks and Grad-CAM activation maps at network-indicated regions of interest. The original picture and heat map are overlaid to identify the essential regions in the image to foresee the pathological condition, model interpretability, and conduct additional analysis. Although the last convolutional neural network layer is frequently utilized in Grad-CAM literature, we viewed all the network levels to examine the learning of the models since the last layer provides high-level information. Here, we used the Grad-CAM method to display and analyze a model's various layers to determine the model's decision-making processes and pinpoint the network segments that have the most influence on classification, To compare the activation maps of the final convolutional layer of three CNN models to clarify all model performances.

5.2 Implementation of Grad-CAM in ResNet50

The process of creating a heat map using Grad-CAM with a ResNet50 model involves the following steps:

1. Select a target class to create the heat map.

2. Forward propagate the image through the ResNet50 model and obtain the output of the last convolutional layer.

3. Compute the gradient of the output of the last convolutional layer for the target class.

4. Average the gradients over the spatial dimensions to obtain a weight for each feature map in the last convolutional layer.

5. Multiply the weight with the feature map to obtain the importance of each feature map in the last convolutional layer for the target class.

6. Up-sample the feature map to the original image size and use it as a heat map to highlight the regions the model is paying attention to.

It is important to note that the Grad-CAM only highlights the regions the model is looking at to make a decision; the model is not trained to look for other areas or regions of the image. Here, figure 5.1,5.2,5.3 shows the heat map acquired from Grad-CAM.

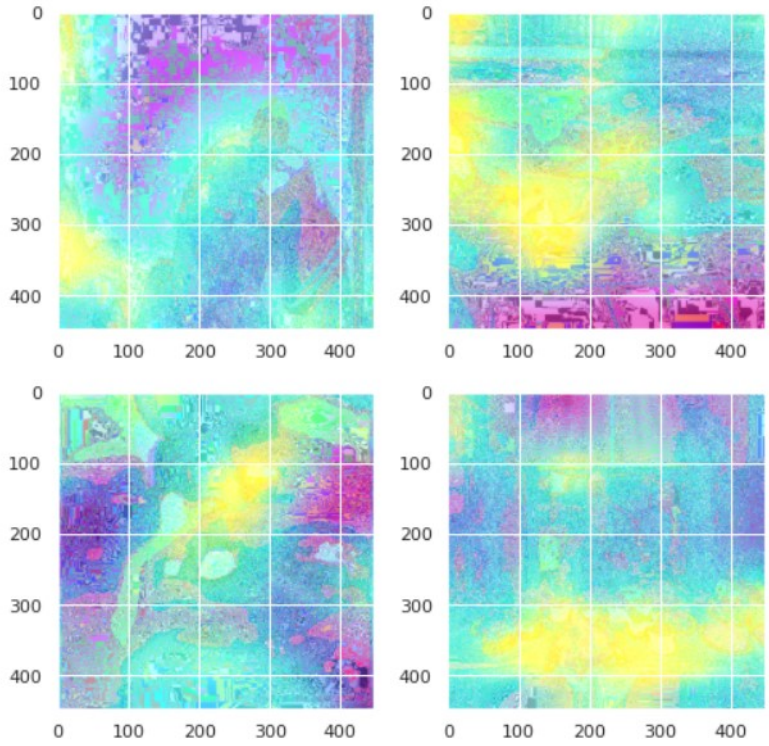


Figure 5.1: Heat map generated by Grad CAM(a)

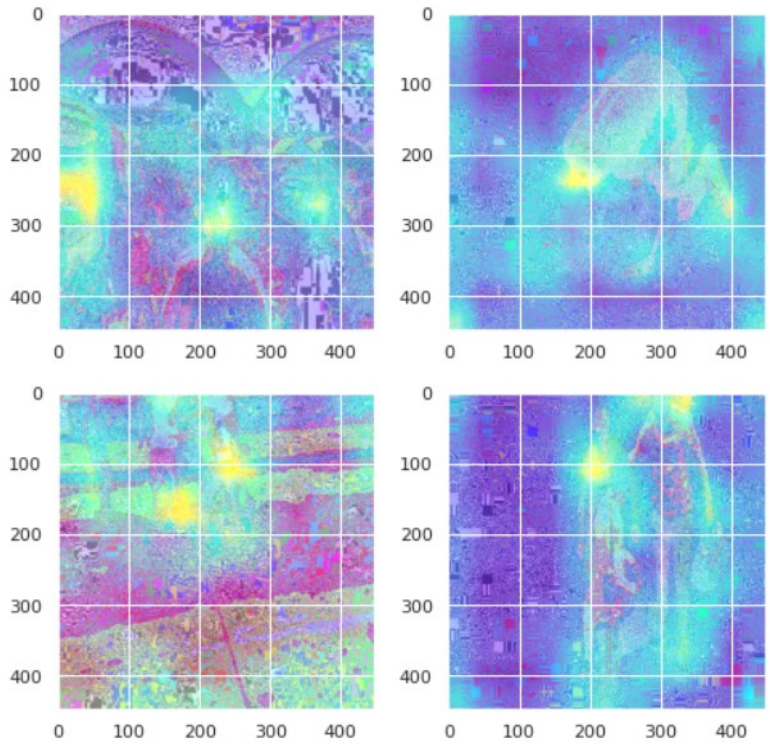


Figure 5.2: Heat map generated by Grad CAM(b)

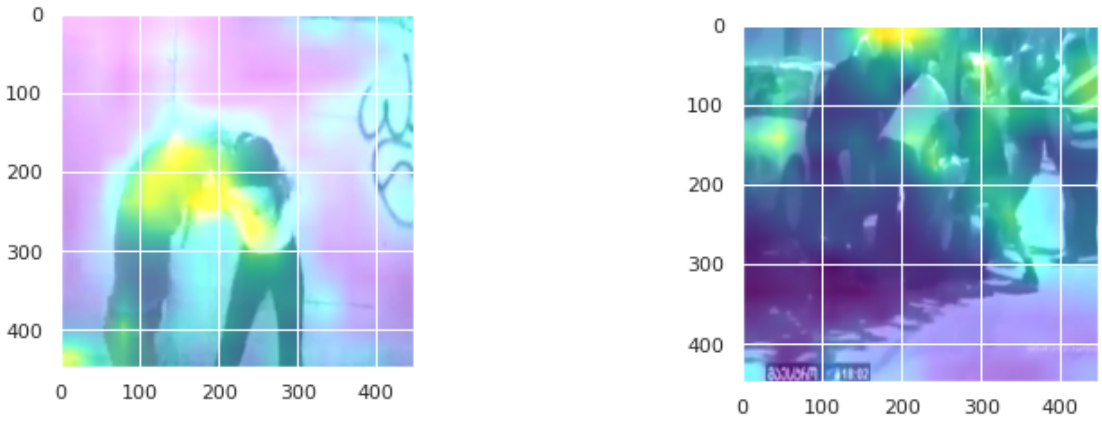


Figure 5.3: Heat map generated by Grad CAM(c)

The Grad-CAM activation map highlights the areas of the image that the neural network "sees" to determine its class. This helps us observe a crucial part of a network: whether it is determining a class based on the correct parts of the image or not. The entire image is colored in blue, and the yellow parts show the parts that the neural network observes for classification. In Figure 5.3, we can see that the neural network looks at the person's arm and the person getting hit to determine the entire image to be of the violent class. We can visually determine here that the model is able to detect the relevant parts of the image. However, for the maps in Figure 5.1, the highlighted parts are located in several places that do not have visible violent actions, indicating that, although the classification prediction was correct,

the network is looking at the wrong places. This is not acceptable, as in real-world scenarios, wrong predictions could be made. This also helps determine whether the model is ready for deployment or not.

Chapter 6

Conclusion and Future Work

The use of deep learning in surveillance systems could save lawful authorities plenty of time and help them solve crimes more effectively. Identifying violent activity through our proposed system can help us minimize the crime rate and help us make a better stable society. We have used three different types of models and tested each of their prediction performances with some random, unseen videos. After running all the models, we received the best accuracy in one of the models (MobileNet-V2), which can detect violent activity the best. Moreover, it is also the least resource intensive, so that it can be used in a wide range of systems. As an effort to include explainable AI in our project, we also used Grad-CAM to generate the heat map when the violent action was taking place, and this helped us visually and qualitatively identify which part of the images our models were detecting as violent actions. We have used one specific open-source dataset for our research. In the future, this could be applied to other datasets with different data. We also want to train models for classifying different types of violent activities individually in an effort to further increase prediction accuracies. We also want to test other pre-trained models and even try to work on a combined model in order to extend further our understanding of the subject of complex deep-learning models. Hopefully, all of our work and the work of future researchers will strive for a better and more peaceful tomorrow.

References

[1] Amrutha C.V, C. Jyotsna, Amudha J. Deep Learning Approach for Suspicious Activity Detection from Surveillance Video, IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1, March 2020.

[2] P. Apoorva, H.C. Impana, S.L. Siri, M.R. Varshitha and Ramesh.B, "Automated Criminal Identification by Face Recognition Using Open Computer Vision Classifiers," in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pp. 775-778, 2019

[3] Sharnagat Singh, Priyanshu Kumar, Animesh Kumar, Shaily Priya Singh, AUTOMATED CRIMINAL IDENTIFICATION SYSTEM USING FACE RECOGNITION, Volume 5——Issue 12——February 2021——ISSN (Online) 2456-0774.

[4] Mrunal Malekar, Detecting Criminal Activities of Surveillance Videos using Deep Learning, doi : https://doi.org/10.32628/CSEIT217111, Volume 7, Issue 1, February 2021.

[5] Sikandar, T., Ghazali, K.H. Rabbi, M.F. ATM crime detection using image processing integrated video surveillance: a systematic review. Multimedia Systems 25, 229–251 (2019). https://doi.org/10.1007/s00530-018-0599-4

[6] Nikolay Teslya, Igor Ryabchikov, Evgeniy Lipkin, "The Concept of the Deviant Behavior Detection System via Surveillance Cameras", Computational Science and Its Applications – ICCSA 2019, vol.11624, pp.169, 2019.

[7] S. Chackravarthy, S. Schmitt and L. Yang, "Intelligent Crime Anomaly Detection in Smart Cities Using Deep Learning," 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), 2018, pp. 399-404, doi: 10.1109/CIC.2018.00060.

[8] Nandwana, B., Tazi, S., Trivedi, S., Kumar, D., Vipparthi, S.K.: A survey paper on hand gesture recognition. In: 7th International Conference on Communication Systems and Network Technologies (CSNT), pp. 147–152. IEEE (2017). https://doi.org/10.1109/csnt. 2017.8418527

[9] S. T. Ratnaparkhi, A. Tandasi and S. Saraswat, "Face Detection and Recognition for Criminal Identification System," 2021 11th International Conference on Cloud Computing, Data Science Engineering (Confluence), 2021, pp. 773-777, doi: 10.1109/Confluence51648.2021.9377205.

[10] M. -S. Kang, R. -H. Park and H. -M. Park, "Efficient Spatio-Temporal Modeling Methods for Real-Time Violence Recognition," in IEEE Access, vol. 9, pp. 76270-76285, 2021, doi: 10.1109/ACCESS.2021.3083273.

[11] G. F. Shidik, E. Noersasongko, A. Nugraha, P. N. Andono, J. Jumanto and E. J. Kusuma, "A Systematic Review of Intelligence Video Surveillance: Trends, Techniques, Frameworks, and Datasets," in IEEE Access, vol. 7, pp. 170457-170473, 2019, doi: 10.1109/ACCESS.2019.2955387.

[12] Lumaban, Michael Bryan. (2020). CCTV-Based Surveillance System with Face Recognition Feature. International Journal of Advanced Trends in Computer Science and Engineering. 9. 349-355. 10.30534/ijatcse/2020/5491.32020.

[13] Simen Thys, Wiebe Van Ranst, Toon Goedeme; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019, pp. 0-0

[14] Henri Bouma, Jan Baan, Gertjan J. Burghouts et al., "Automatic detection of suspicious behavior of pickpockets with track-based features in a shopping mall", Optics and Photonics for Counterterrorism, Crime Fighting, and Defense X; and Optical Materials and Biomaterials in Security and Defense Systems Technology XI 9253, pg., (2014); doi:10.1117/12.2066851

[15] T. Sultana and K. A. Wahid, "Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things," in IEEE Access, vol. 7, pp. 41607-41624, 2019, doi: 10.1109/ACCESS.2019.2907525.

[16] Nor Shahidayah Razali, Azizah Abdul Manaf, "Gait Analysis for Criminal Identification Based on Motion Capture", 2010.

[17] T. Yamauchi and M. Ishikawa, "Spatial Sensitive GRAD-CAM: Visual Explanations for Object Detection by Incorporating Spatial Sensitivity," 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 256-260, doi: 10.1109/ICIP46576.2022.9897350.