VIOLENCE DETECTION USING DEEP LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by

RAHNA RASHEED (TKM19MCA019)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirments for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2022

DECLARATION

I undersigned hereby declare that the project report on''VIOLENCE DETECTION USING DEEP LEARNING TECHNIQUES" , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Dr.Nadera Beevi S. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

22/05/2022

Rahna Rasheed

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

C E R T I F I C A T E

This is to certify that, the project report entitled **"VIOLENCE DETECTION USING DEEP LEARNING TECHNIQUES. "** submitted by **RAHNA RASHEED (TKM19MCA019)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications, is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor | Head of the Department | External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia.M.Shamsudeen**, Head of the Department, for providing me with best facilities.

I would like to thank my project guide **Dr. Nadera Beevi S**, Department of Computer Applications, who motivated me throughout the work of my project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

Rahna Rasheed

ABSTRACT

Violence recognition is challenging since recognition must be performed on videos acquired by a lot of surveillance cameras at any time or place. It should make reliable detections in real time and inform surveillance personnel promptly when violent crimes take place. Therefore,this focus on efficient violence recognition for real-time and on-device operation, for easy expansion into a surveillance system with numerous cameras. In this work, we propose a novel violence detection pipeline that can be combined with the conventional 2-dimensional Convolutional Neural Networks (2D CNNs). In particular, frame-grouping is proposed to give the 2D CNNs the ability to learn spatio-temporal representations in videos. It is a simple processing method to average the channels of input frames and group three consecutive channel-averaged frames as an input of the 2D CNNs.

Furthermore, spatial and temporal attention modules are included that are lightweight but consistently improve the performance of violence recognition. The proposed pipeline brings significant performance improvements compared to the 2D CNNs followed by the Long Short-Term Memory (LSTM) and much less computational complexity than existing 3D-CNN-based methods. In particular, MobileNetV3 and EfficientNet-B0 with our proposed modules achieved state-of-the-art performance on six different violence datasets.

Contents

List of Figures

Chapter 1

Introduction

In video surveillance, to critically assure public safety hundreds and thousands of surveillance cameras are deployed within cities, but it is almost impossible now a day to manually monitor all cameras to keep an eye on violent activities. Rather, there is a significant requirement for developing automated video surveillance systems to automatically track and monitor such activities. Thereby, in case of emergencies alarming the controlling authorities to take appropriate measures against detected violence. Violence recognition is a key step towards developing automated security surveillance systems, to distinguish normal human activities from abnormal/violent actions. Normal human activities are often categorized as routine life interactive behaviours, such as walking, jogging, running, hand waving. However, violence is subjected to unusual furious actions, such as fight activity happening between two or more people. In last few years, the task of human action recognition has received much attention of the researcher community, to detect normal day human activities through video analysis. However, little attention has paid to the problem of human violent action detection, until the availability of violent sequences for fight activities. This system have created two datasets specifically for fight activities detections, to distinguish violent/fight incidents from normal events. Before the availability of this dataset, most of the available datasets were particularly concerned about general human activities. CNN (Convolutional Neural Network) followed by LSTM (Long-short term memory) has been proved to be the best architecture when data are small and low computing power available for the task in hand.

1.1 Existing System

Violence Detection has been performed using various techniques, including data mining, machine learning, and hybrid technologies.These techniques enable and support companies in identifying, predicting, and detecting violences. In video surveillance, to critically assure public safety hundreds and thousands of surveillance cameras are deployed within cities, but it is almost impossible now a day to manually monitor all cameras to keep an eye on violent activities. Rather, there is a significant requirement for developing automated video surveillance systems to automatically track and monitor such activities. Most of the existing systems uses several handcrafted techniques and thus have a higher response time.Complex network structures often decreases the reliability of the project.

- For violence detection, most of the existing approaches relay on hand-defined features descriptors, to distinguish fight sequences from normal ones, a scheme often used in human action recognition domain.
- Thereby, since the introduction of the violent/fight specific two datasets, most of the techniques are dependent on formulating hand-crafted feature representations for violence identification.

1.2 Proposed System

Today modern cities tend to grow rapidly. The increased population density brings new challenges in term of public safety. Crime and violence are hard to be detected and managed especially in specific crowd environments like music concerts, sport events or public meetings. To overcome this issue the city administration should implement monitoring systems capable of detecting and analysing such situations. This approach uses deep learning techniques. They demonstrate outstanding performances in image and actions classification based on a prior learning process. Thus the system succeed to obtain a real and cost-effective solution for surveillance networks. Figure 1.1 below shows the proposed model for violence detection.

The features of the system can be summarized as follows:

- A violence detector based on deep-learning methods.
- The proposed model consists of CNN as a spatial feature extractor and LSTM as temporal relation learning method.
- Publicly available dataset is used for violence detection in the input video.
- The system efficiently classify the input video as either violent or non violent video.

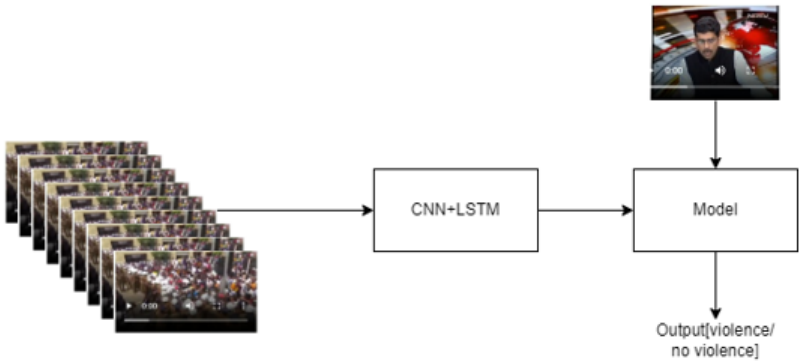

Figure 1.1: Proposed model of violence detection

1.3 Objectives

The main objectives of the project are as follows:

- The aim of this work is to develop a system, that can detect violence present in the input video.
- Training the deep learning model with data set using pre-trained VGG19 architecture.
- Testing both models and find its accuracy.

Chapter 2

Literature Survey

Literature review is the comprehensive study and interpretation of literature that relates to a particular topic. When one uses literature review research questions are identified, then one seek to answer this research questions by searching for and analyzing relevant literature. Some importance of literature reviews is that new insights can be developed by the re-analyzing the results of the study. A literature review is both a summary and explanation of the complete and current state of knowledge on a topic as found in academic books and journal articles.

2.1 Related Works

Some work done related to Violence Detection is explained.

Pin Wang et al. proposed a Violence detection and face recognition based on deep learning method. The purpose of this article is to study methods of brute force detection and face recognition based on deep learning. Aiming at the problem of abnormal behavior detection, especially the low efficiency and low accuracy of brute force detection, a brute force detection method based on the combination of Convolutional Neural Network and trajectory is proposed[1].

I.P. Febin et al. proposed a method that the surveillance videos are checked through a movement filtering algorithm based on temporal derivative and avoid most of the nonviolent actions from going through feature extraction. Only the filtered frames may allow going through feature

extraction. In addition to scale-invariant feature transform (SIFT) and histogram of optical flow feature, motion boundary histogram is also extracted and combined to form MoBSIFT descriptor[2].

P. Wu, J. Liu et al. proposed Not only Look, but also Listen: Learning Multimodal Violence Detection under Weak Supervision. Previous works were either superficial like classification of short-clips, single scenario, or undersupplied like the single modality, and hand-crafted features based multimodality. To address this problem, a large-scale and multi-scene dataset was introduced. Then a neural network was proposed containing three parallel branches to capture different relations among video snippets and integrate features[3].

Samee Ullah Khan et al. proposed a violence detection scheme for movies that is comprised of three steps. First, the entire movie is segmented into shots, and then a representative frame from each shot is selected based on the level of saliency. Next, these selected frames are passed from a light-weight deep learning model, which is fine-tuned using a transfer learning approach to classify violence and non-violence shots in a movie. Finally, all the non-violence scenes are merged in a sequence to generate a violence-free movie that can be watched by children and as well violence paranoid people[4].

Javad Mahmoodi et al. introduced a new feature descriptor named Histogram of Optical flow Magnitude and Orientation (HOMO). The proposed method converts input frames to the grayscale format and then it computes the optical flow between two consequence frames, then the optical flow magnitude and orientation of each pixel in each frame are compared separately with its predecessor frame and six binary indicators are created.Finally, these binary indicators are analyzed to get the HOMO descriptor which is used to train a SVM classifier[5].

Oswald Lanz et al. proposed a deep neural network for the purpose of recognizing violent videos. A convolutional neural network is used to extract frame level features from a video. The frame level features are aggregated using a variant of the long short term memory that uses convolutional gates. The convolutional neural network along with the convolutional long short term

memory is capable of capturing localized spatio-temporal features which enables the analysis of local motion taking place in the video.[6].

Enrique Bermejo Nievas et al. proposed Violence Detection in Video Using Computer Vision Techniques. The action recognition community has focused mostly on detecting simple actions like clapping, walking or jogging, the detection of fights or general aggressive behaviours has been comparatively less studied. Such capability may be extremely useful in some video surveillance scenarios like in prisons, psychiatric or elderly centres or even in camera phones. After an analysis of previous approaches, well-known Bag-of-Words framework used for action recognition in the specific problem of fight detection, along with two of the best action descriptors currently available: STIP and MoSIFT[7].

Ali Sharif Razavian et al. proposed CNN Features off-the-shelf: an Astounding Baseline for Recognition.The system has a series of experiments conducted for different recognition tasks using the publicly available code and model of the over feat network which was trained to perform object classification on ILSVRC13. It use features extracted from the over feat network as a generic image representation to tackle the diverse range of recognition tasks of object image classification, scene recognition, fine grained recognition, attribute detection and image retrieval applied to a diverse set of datasets[8].

Aqib Mumtaz et al. proposed Violence Detection in Surveillance Videos with Deep Network Using Transfer Learning. Violent action recognition has significant importance in developing automated video surveillance systems. Over last few years deep representation based approaches have been successfully used in image recognition and human action detection tasks. This paper proposed deep representation based model using concept of transfer learning for violent scenes detection to identify aggressive human behaviors[9].

Chapter 3

Methodology

Violence Detection using Deep Learning Techniques, this project mainly aims to identify the existence of any violent activities in videos. This approach uses machine learning model to improve the classification accuracy of violence detection. For building a prediction model we have used different models, Convolutional Neural Network and Long Short-Term Memory . In this project a Hockey dataset is used to train and test the prediction models.

3.1 System Architecture

The main purpose of the project is the detection of violence. So when given video clips, the proposed algorithm can detect whether the video contains violence or not. This study uses a well-known Convolutional Neural network model VGG19 for detection of violence. This demonstrates that using pre trained models for image classification is an effective way for violence detection. Figure 3.1 below shows system architecture. The proposed system consist of four major phases:

- Data Pre-processing and Augmentation.
- Training the data set.
- Evaluation of the model.
- Deploying the model.

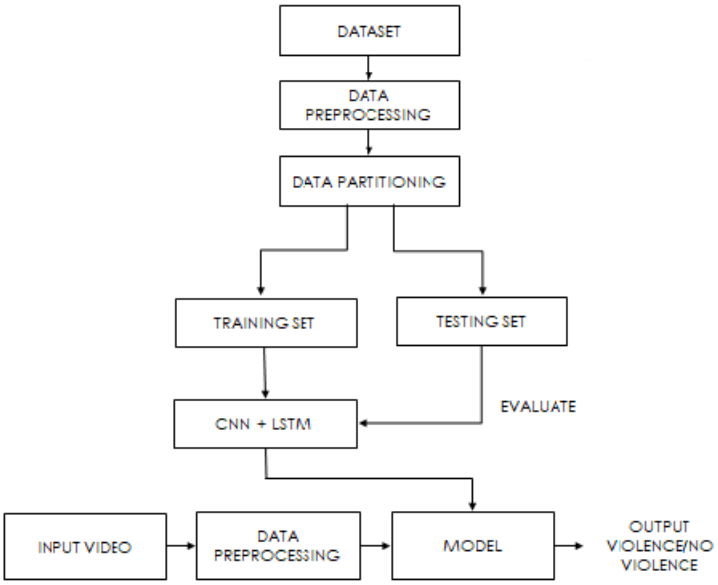

Figure 3.1: System Architecture

3.1.1 Data set

Data collection is the process of gathering and measuring information from countless different sources. Usually the dataset has very large number of features, examples and incomplete values. In this project, use publicly available dataset named Hockey dataset. Hockey Fight Dataset is a 2000 videos Dataset with 1000 fight and 1000 no fight videos of Hockey. Clips last approximately 2 seconds and consist of approximately 41 frames with a resolution of 360x288. The details of the sequences are quite similar.

3.1.2 Data Preprocessing

After loading the images, they resized to 224, 224 size because the images are of different length and widths. Also normalize the image pixels by dividing by 225. Data Augmentation techniques

are extensively used to enhance the performance of deep learning algorithms. Data augmentation is primarily used to improve the performance of the convolutional neural network model, used to classify the image frames in video. Data Augmentation is implemented on the training data, which are the images in this case, to enhance and augment the quality, adaptness, and size of the data. Numerous operations are carried out to increase the size of the data and help the Deep Learning model capture multiple nuances in the training images. Convolutional Neural Networks are fed with the augmented data to prevent over fitting of the data i.e, to optimize the performance of the model. Over fitting is an undesirable condition where a machine learning or deep learning model performs well on the training set but yields undesirable results on the testing and validation sets. Thus, over fitting of the model is not advisable and should not occur during the implementation. Increasing the training dataset enables the Deep Learning and Computer Vision algorithms to fit the models more efficiently. There are various data augmentation techniques, which have been used to train the model. This practically does not change the original dataset and is only implemented during the run time, without any unnecessary disk space being utilized to store the modified and augmented images. Data augmentation performed in this are flipping, rotation and shearing.

3.1.3 Training the model

Another performance enhancing method in deep models especially in CNNs which is named as transfer learning. Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones. There are three different transfer learning approaches in CNNs. These are feature extractor, fine-tuning and pre-trained models. The study, used fine tuning approach which is motivated by observing in early layers of CNNs have more generic features such as edges, colors. The models used in the proposed system are VGG19and LSTM. Both VGG19 and LSTM models are trained separately and these models are saved as H5 files. VGG19 is added to the top of the network, and then, other layers like Time Distributed layer, Flatten layer, Dropout layer, Dense layer, etc, are added as the rest of the layers with required parameters. After network designed, apply training set to the input layer of the neural network for training.

3.1.4 Model Creation and Prediction

After training completed, testing is done for accuracy. After each iteration completed, testing the model using test dataset. Finally, save the model for new prediction. Evaluation is performed for different classifiers by applying testing set. Calculate accuracy to analyse the performance of the classifier. Analyse the evaluation results. If the model has acceptable accuracy, the model can be used for future prediction.

3.2 Deep Learning

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also, known as deep neural learning or deep neural network. Deep learning occurs when decisions are made on unstructured data without supervision. Object recognition, speech recognition, and language translation are some of the tasks performed through deep learning. Deep learning uses hierarchical neural networks to analyze data. Neuron codes are linked together within these hierarchical neural networks, similar to the human brain. Unlike other traditional linear programs in machines, the hierarchical structure of deep learning allows it to take a nonlinear approach, processing data across a series of layers which each will integrate subsequent tiers of additional information.

3.2.1 Convolutional Neural Networks

CNN's have proven to perform remarkably well on classifying images, and as such, are a great fit for this problem. It is worth noting that CNNs typically suffer from overfitting, which occurs when a model adapts too well on trained data, but performs poorly on new data, and thus is said to generalize poorly. CNN rely on the idea that local understanding of an image is good enough, with the practical benefit of having fewer parameters, consequently reducing the computation time and data required to train the model. CNNs only have enough weights to look at small parts of the image at a time. The process typically involves a convolution layer, followed by a pooling and an

activation function, but not necessarily in that exact order. These three separate operations can be applied as separate layers to the original image, typically multiple times. Finally, a fully connected layer (or several) is added at the end to classify an image accordingly. With the highly variable number of combinations and permutations, it can be difficult to find the exact one that gives the optimal performance. CNN designs are driven by the community and research who thankfully have made some successful results, and made their CNNs publicly available for others to use and improve upon. Figure 3.2 shows the CNN Network.

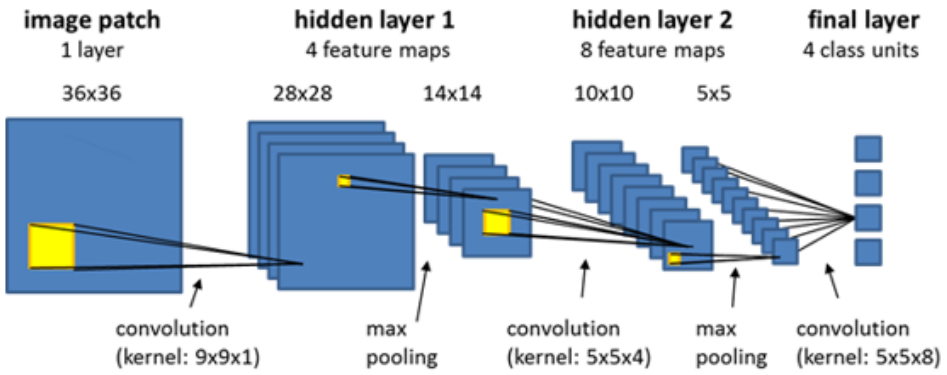

Figure 3.2: CNN Network

VGG19

VGG-19 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 19 layers(16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer) deep and can classify images into 1000 object categories, such as a keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. There are other variants of VGG like VGG11, VGG16 and others. VGG19 has 19.6 billion FLOPs. The figure 3.3 below shows different layers of VGG19 architecture.

Architecture

- A fixed size of RGB image was given as input to this network.
- The only pre-processing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set.
- It uses kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image.
- Spatial padding was used to preserve the spatial resolution of the image.
- Max pooling was performed over a 2 * 2-pixel windows with stride 2.
- This was followed by sigmoid or Rectified linear unit (ReLu) to introduce non-linearity to make better model classification and also to improve the computational time.

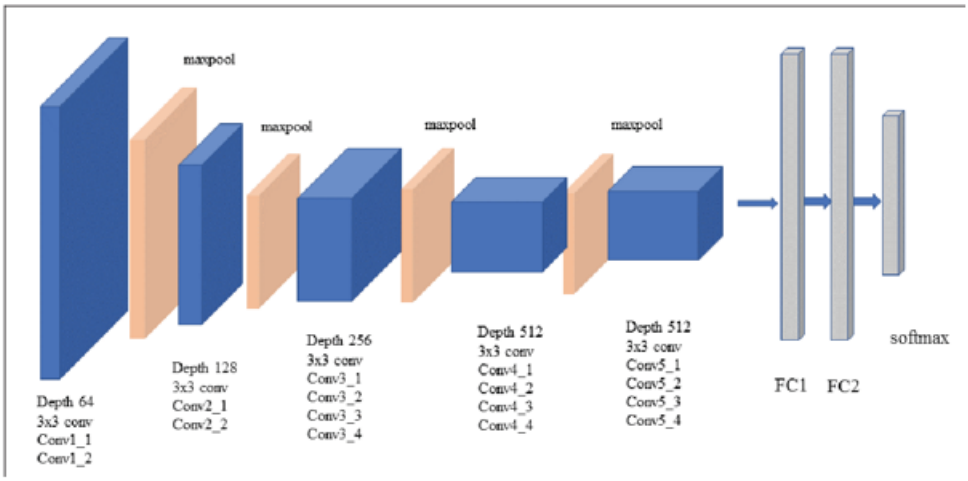

Figure 3.3: VGG19 Architecture

LSTM

Long Short-Term Memory is an improvement of Recurrent Neural Networks (RNNs). It proposes memory blocks instead of RNN units in solving the vanishing and exploding gradient problem. It then adds a cell state to save long-term states. An LSTM network can remember and connect previous information to data obtained in the present. LSTM is combined with three gates, such as an input gate, a "forget" gate, and an output gate, where refers to the current input; and denote the new and previous cell states, respectively.The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got the only single neural net layer of tanh, LSTMs comprises of three logistic sigmoid gates and one tanh layer.They determine which part of the information will be needed by the next cell and which part to be discarded. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'. The figure 3.4 below shows different layers of LSTM architecture.

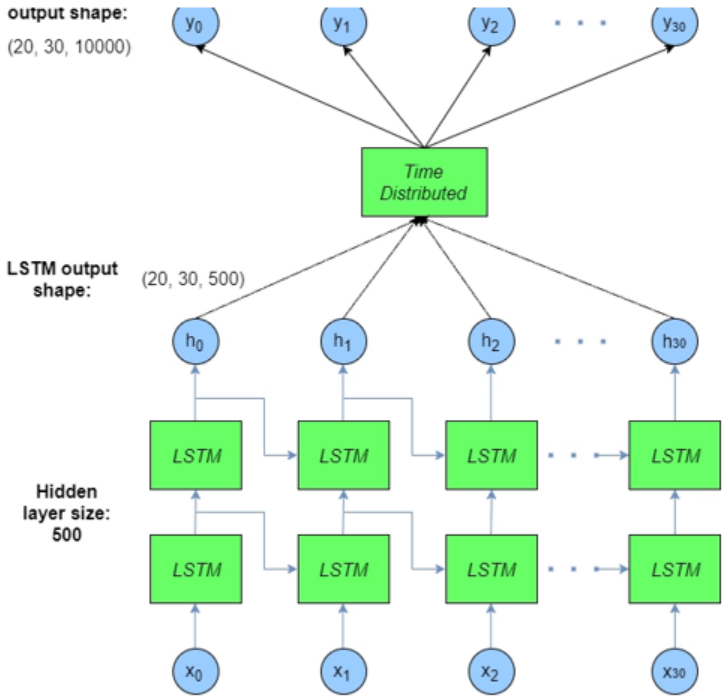

Figure 3.4: LSTM Architecture

3.3 Software Requirements and Specification

- Program logic - Python 3
- IDE - VS Code
- Browser - Chrome/ Edge
- Designing tool - HTML, CSS, Bootstrap, Javascript
- Web framework - Django
- Database - MySQL
- Packages - Tensorflow, Keras

3.3.1 Python

Python is an object-oriented programming language created by Guido Rossum in 1989. It's ideally designed for fast prototyping of complicated applications. It has interfaces to several OS system calls and libraries and is protractile to C or C++. Several massive corporations use the Python programming language embody NASA, Google, YouTube, BitTorrent, etc. Python programming is widely utilized in AI, natural language Generation, Neural Networks and other advanced fields of computer science.Reading and writing codes in Python is far like reading and writing regular English statements. As a result, they're not written in the machine-readable language, Python programs got to be processed before machines can run them. This implies that each time a program is run, its interpreter runs through the code and interprets it into machine-readable byte code.Everything in Python is, in fact, top-notch. All objects, data types, functions, methods, and classes take an equal position in Python. Programming languages are created to satisfy the requirements of programmers and users for an efficient tool to develop applications that impact lives, lifestyles, economy, and society. they assist build lives better by increasing productivity, enhancing communication, and rising potency. Python programming language artificial language that has stood the test of time and has remained relevant across industries and businesses and among programmers, and individual users.

3.3.2 Python Standard Library

The Python Standard Library is a collection of exact syntax, token, and semantics of Python. It comes bundled with core Python distribution. We mentioned this when we began with an introduction. It is written in C, and handles functionality like I/O and other core modules. All this functionality together makes Python the language it is. More than 200 core modules sit at the heart of the standard library. This library ships with Python. But in addition to this library, you can also access a growing collection of several thousand components from the Python Package Index (PyPI).

Pandas

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Like we have said before, Pandas is a must for data-science. It provides fast, expressive, and flexible data structures to easily (and intuitively) work with structured (tabular, multidimensional, potentially heterogeneous) and time-series data.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It has advanced math functions and a rudimentary scientific computing package.

SciPy

SciPy, a scientific library for Python is an open source, BSD-licensed library for mathematics, science and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array manipulation. The main reason for building the SciPy library is that, it should work with NumPy arrays. One of the libraries we have been talking so much about. It has a number of user-friendly and efficient numerical routines. These include routines for optimization and numerical integration.

Scikit

Scikit-learn is a free machine learning library for Python. It features various algorithms like support

vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

3.3.3 MySQL

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history. MySQL database is owned, developed and supported by Sun Microsystems, one of the world's largest contributors to open source software. MySQL was originally found and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius. The best and the most-used database in the world for online applications.With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use My SQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack.

3.3.4 Django

Django is a high-level python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a non-profit. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "plugability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data

models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

- Ridiculously fast.
- Reassuringly secure.
- Exceedingly scalable.

3.3.5 HTML, CSS

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

3.3.6 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is among the most starred projects on GitHub, with more than 142,000 stars, behind freeCodeCamp (almost 312,000 stars) and marginally behind Vue.js framework.

3.3.7 Tensorflow

TensorFlow is an end-to-end open source platform for machine learning. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 in 2015. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google.

TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs.

3.3.8 Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear actionable error messages. It also has extensive documentation and developer guides.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

Chapter 4

Result And Discussion

Convolutional Neural Network(CNN) has been quite successful in violence detection.Also by using pre trained CNN models on large scale datasets, important features can be learned which can be used in image classification. The efficiency of the model is rated high. After the model is constructed, compile the neural net using a categorical cross entropy selected as loss function.Different approaches are used for overfitting. Firstly, the dropout method was used after fully connected layers. Secondly, early stopping is used for avoiding overfitting.

4.1 Training and Validation Results

Training gives optimized results with high accuracy of 99.14 percentage for CNN and a corresponding loss of 0.029 for training and 98 percentage accuracy was found for validation.Figure 4.1 below shows the model's training and testing accuracy.

```
started at  1542478315557
Train on 700 samples, validate on 300 samples
Epoch 1/10
700/700 [==============================] - 200s 286ms/step - loss: 0.3792 - acc: 0.8643 - val_loss: 0.2605 - val_acc: 0.9233
Epoch 2/10
700/700 [==============================] - 201s 287ms/step - loss: 0.2232 - acc: 0.9243 - val_loss: 0.1652 - val_acc: 0.9467
Epoch 3/10
700/700 [==============================] - 201s 288ms/step - loss: 0.1790 - acc: 0.9500 - val_loss: 0.1335 - val_acc: 0.9533
Epoch 4/10
700/700 [==============================] - 200s 286ms/step - loss: 0.1319 - acc: 0.9600 - val_loss: 0.1083 - val_acc: 0.9733
Epoch 5/10
700/700 [==============================] - 199s 284ms/step - loss: 0.1051 - acc: 0.9750 - val_loss: 0.1849 - val_acc: 0.9350
Epoch 6/10
700/700 [==============================] - 199s 284ms/step - loss: 0.0871 - acc: 0.9736 - val_loss: 0.1137 - val_acc: 0.9583

Epoch 00006: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
Epoch 7/10
700/700 [==============================] - 202s 289ms/step - loss: 0.0586 - acc: 0.9843 - val_loss: 0.1131 - val_acc: 0.9650

Epoch 00007: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
Epoch 8/10
700/700 [==============================] - 200s 286ms/step - loss: 0.0459 - acc: 0.9879 - val_loss: 0.0857 - val_acc: 0.9800
Epoch 9/10
700/700 [==============================] - 200s 286ms/step - loss: 0.0354 - acc: 0.9886 - val_loss: 0.0855 - val_acc: 0.9800
Epoch 10/10
700/700 [==============================] - 200s 285ms/step - loss: 0.0299 - acc: 0.9914 - val_loss: 0.0853 - val_acc: 0.9800
<tensorflow.python.keras._impl.keras.callbacks.History at 0x20c3e7a1940>
```

Figure 4.1: Accuracy after training

ROC Curve

ROC curve, also known as Receiver Operating Characteristics Curve, is a metric used to measure the performance of a classifier model. The ROC curve depicts the rate of true positives with respect to the rate of false positives, therefore highlighting the sensitivity of the classifier model. The ROC is also known as a relative operating characteristic curve, as it is a comparison of two operating characteristics, the True Positive Rate and the False Positive Rate, as the criterion changes. An ideal classifier will have a ROC where the graph would hit a true positive rate of 100 percentage with zero false positives. Figure 4.2 shows the ROC Curve for the system.

Figure 4.2: ROC Curve

4.2 Confusion matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix. Figure 4.3 represents the model's training and testing accuracy.
The important terms included in confusion matrix are as following:

- True Positive (TP) :Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.
  The figure 4.4 indicates the confusion matrix based on the classification results.

		Predicted	
		0	1
Actual	0	True Negative	False Negative
	1	False Positive	True Positive

Figure 4.3: Representation of Confusion matrix

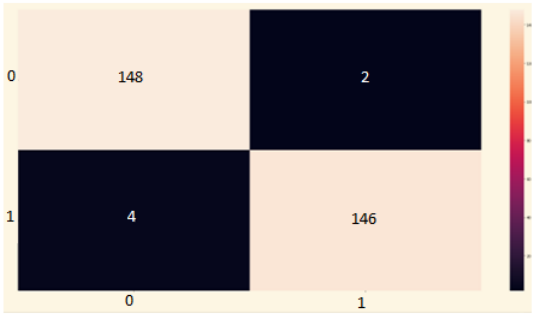

Figure 4.4: Confusion matrix

4.3 Performance Metrics

The below table indicates the performance metrics to calculate the accuracy based on classification results.Figure 4.5 shows the precision matrix of the project.

- Precision

Precision is the ability of a classifier not to label an instance positive, that is actually negative. Precision is the fraction of relevant instances among the retrieved instances of classification technique.
Precision = TP / (TP + FP)

- Recall

Recall is one of the most common performance metrics to estimate the classification model. The recall is the fraction of the total amount of relevant instances that were actually the retrieved instance of the classification algorithm.
Recall =TP / (TP + FN)

- f1 Score

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0.The weighted average of f1 should be used to compare classifier models, not global accuracy.
f1Score =2 (Recall * precision)/(Recall + Precision)

- Support

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or re-balancing. Support doesn't change between models but instead diagnoses the evaluation process.

- Accuracy

Accuracy is a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

Accuracy=(TP + TN) / (TP + TN + FP + FN)

	Precision	Recall	F1-Score	Support
No Violence	0.97	0.99	0.98	150
Violence	0.99	0.97	0.98	150
Accuracy			0.98	300
Macro Avg	0.98	0.98	0.98	300
Weighted Avg	0.98	0.98	0.98	300

Figure 4.5: Performance Metrics of CNN

Chapter 5

Conclusion

Therefore, it is concluded that the deep learning model proposed detects the violence present in the videos in a very accurate manner.The loss of model is minimised while training and the accuracy simultaneously increases through each epoch stages in order to yield distinct results.The preprocessing stage helps to ensure that the performance of Convolutional Neural Network and deep learning networks is not subjected to overfitting, thus the results obtained will always remain coherent. The proposed base model is evaluated on a benchmark dataset and resulted in improved performance compared to the previous works for that dataset. With a smaller number of convolutional layers the model predicts adroitly whether a given image contains violence or not. This is immensly helpful.

5.1 Future Enhancement

This project has an immense scope in law enforcement, public safety ensuring,crowd monitoring and thus can be continued for other insightfull innovations. There is still a place for improvement and to explore or create a new well-balanced large data set with different video sources for violence detection. For future work, other datasets are introduced in order to benchmark more databases and techniques.

References

[1] Pin Wang, Peng Wang, En Fan, ”Violence detection and face recognition based on deep learning”,IEEE 2021.

[2] I. P. Febin, K. Jayasree, Preetha Theresa Joy, ”Violence detection in videos for an intelligent surveillance system using MoBSIFT and movement filtering algorithm”,IEEE 2020.

[3] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang,”Not onlylook, but also listen: Learning multimodal violence detection under weak supervision”, IEEE 2020.

[4] Samee Ullah Khan,Ijaz Ul Haq, Seungmin Rho,Sung Wook Baik,, ”Cover the Violence: A Novel Deep-Learning-Based Approach Towards Violence-Detection in Movies”,IEEE 2020.

[5] Javad Mahmoodi, Afsane Salajeghe, ”A classification method based on optical flow for violence detection”,IEEE 2019.

[6] Swathikiran Sudhakaran, Oswald Lanz, ”Learning to Detect Violent Videos using Convolutional Long Short-Term Memory”, IEEE 2019.

[7] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García Rahul Sukthankar, ”Violence Detection in Video Using Computer Vision Techniques, IEEE 2019.

[8] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, Stefan Carlsson, ”CNN Features off-the-shelf: an Astounding Baseline for Recognition”,IEEE 2019.

[9] Aqib Mumtaz, Allah Bux Sargano, Zulfiqar Habib, ”Violence Detection in Surveillance Videos with Deep Network Using Transfer Learning”, IEEE 2019.

[10] I. S. Gracia, O. D. Suarez, G. B. Garcia, and T. K. Kim, “Fast fight detection,” IEEE 2018.

[11] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, “Spatio-temporal attention networks for action recognition and detection,” IEEE Trans. Multimedia, vol. 22, no. 11, pp. 2990–3001, Nov. 2020.

[12] O. Deniz, I. Serrano, G. Bueno, and T.-K. Kim, ”Fast violence detection in video. In International Conference on Computer Vision Theory and Applications (VISAPP), 2019.

APPENDICES

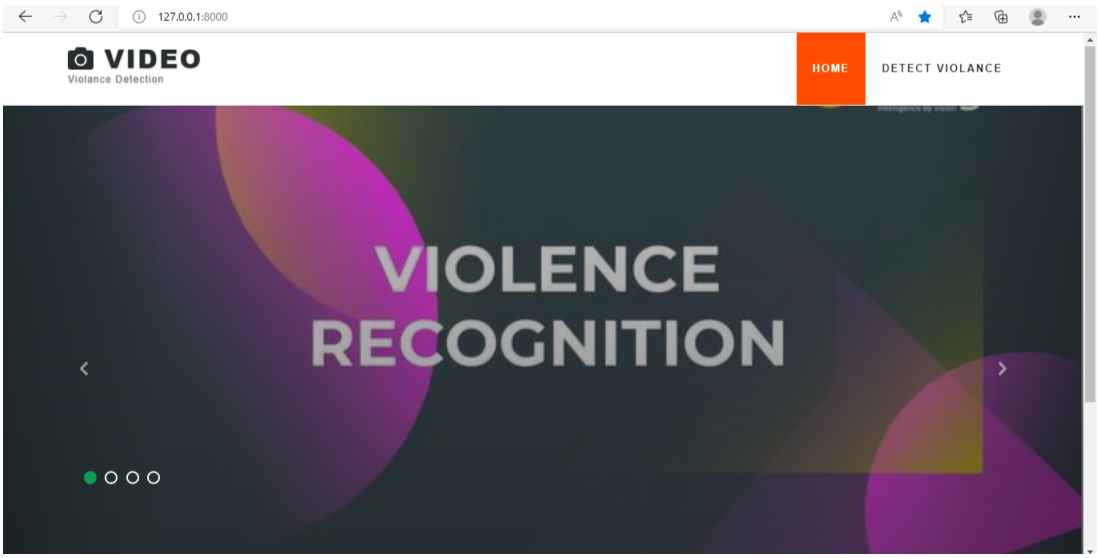

Figure A.1: login page

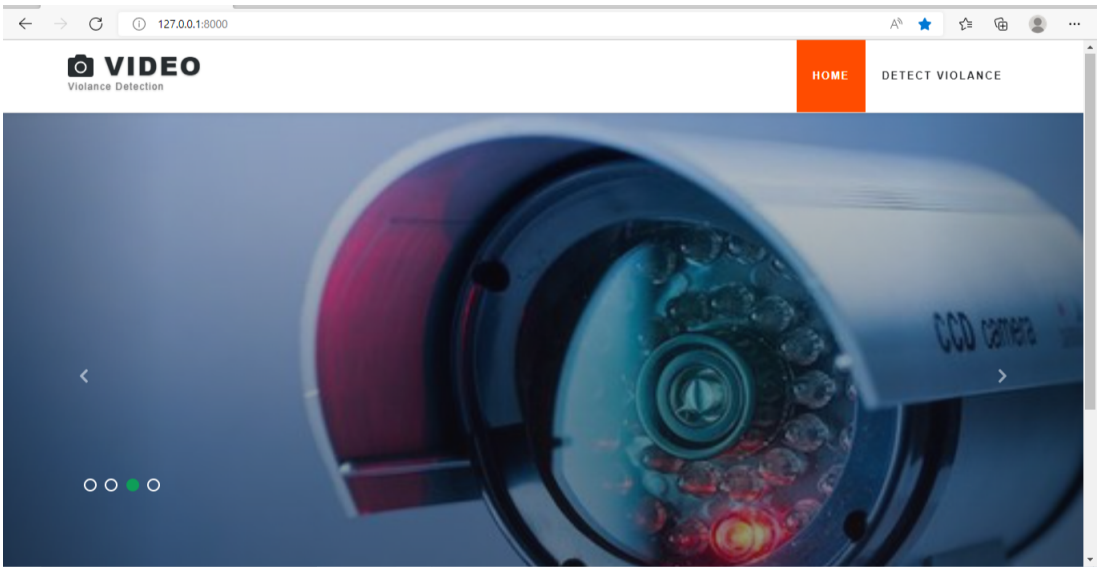

Figure A.2: login page

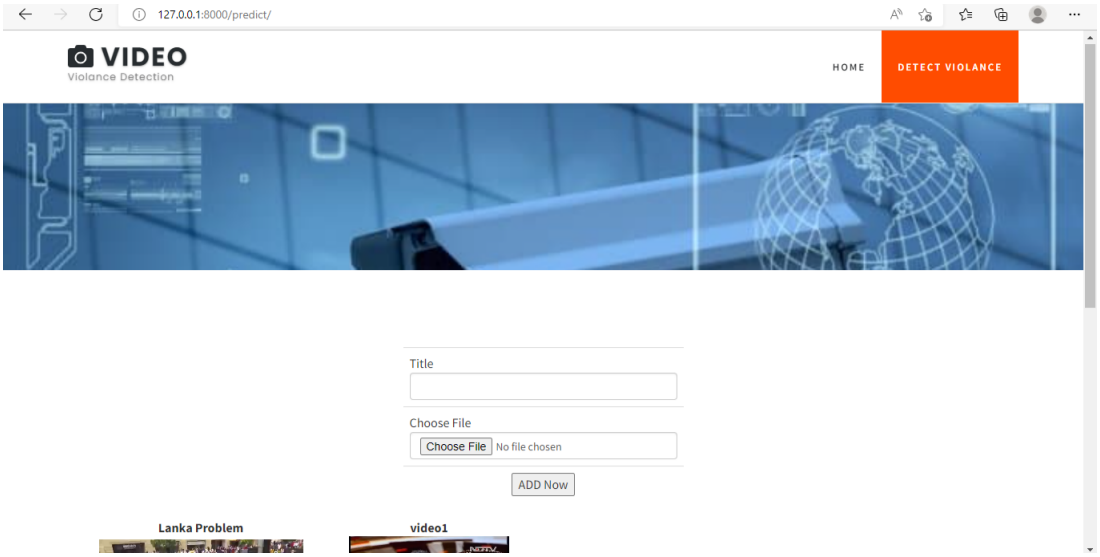

Figure A.3: Video Uploading

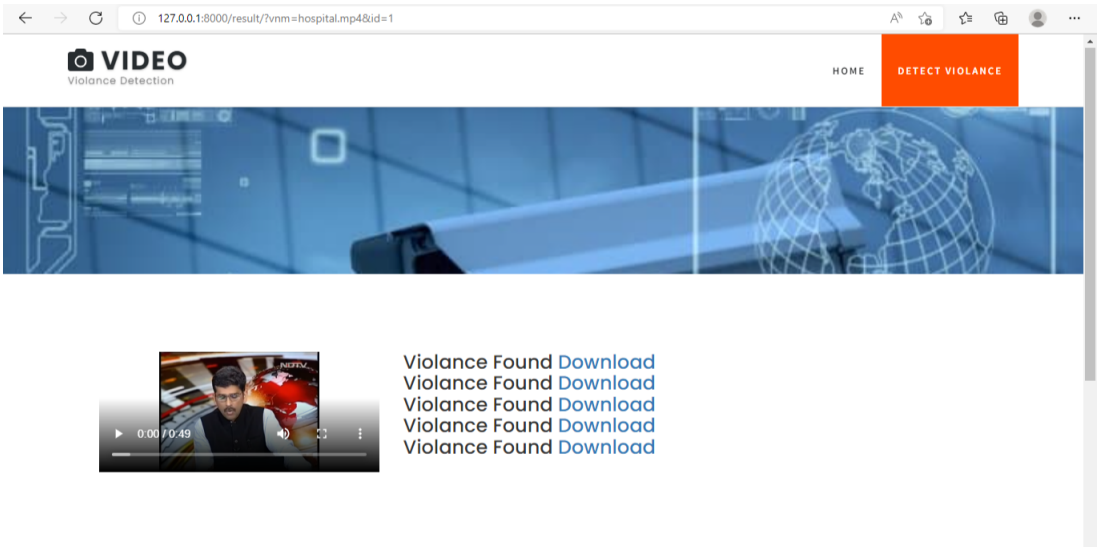

Figure A.4: Prediction result