**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

# GRADUATION THESIS

## Violence Detection in Surveillance Camera

**Tran Thi Thuy**
thuy.tt176059@sis.hust.edu.vn

**Major: Information Technology**
**Specialized major: Information Technology**

**Supervisor:** MSc. Nguyen Duc Tien _____

Supervisor's signature

**Department:** Computer Engineering

**School:** Information and Communication Technology

**HANOI, 08/2022**

# PLEDGE

Student name: Tran Thi Thuy
Tel: +84 966 282 794
Email: thuy.tt176059@sis.hust.edu.vn
Class: LTU17A
Program: SIE

We – *Tran Thi Thuy* and *Vo Sy Hung* – commit that the Graduation Project (GP) is our own research work under the guidance of *MSc. Nguyen Duc Tien*. The results stated in the GP are honest, our own work, not copied from any other works. All references in the GP—including images, tables, figures, and quotations—are clearly and fully documented in the bibliography. We take full responsibility for even one copy that violates the school's regulations.s regulations.

*Hanoi, August 8 2022*

Author

*Student name*

# ACKNOWLEDGEMENTS

During my study and graduation practice, I have always been concerned, guided, and received the dedicated help of teachers in the School of Information and Communication Technology.

First of all, I would like to express my deep gratitude to the Board of Directors, The brand name of Hanoi University of Science and Technology, the School of Information and Communication Technology, has supported me throughout my time at the school.

In particular, I would like to express my sincere gratitude to MSc. Nguyen Duc Tien for directly helping and guiding me to complete this project.

I would also like to express my deep gratitude to my family, relatives, friends, and colleagues who have helped and encouraged me to complete the thesis well.

Thank you very much!

# ABSTRACT

Violence, which causes numerous damage to society, is a major global problem. With the development of technologies, artificial intelligence algorithms have been combined with security cameras to create an end-to-end system for detecting and warning violent acts. Additionally, a majority of violent behaviors in Vietnam happen between a couple of people. Therefore, most benchmark datasets in this field, which are recorded from the crowd, sports games, or movies, are not suitable for our country's situation. In this project, we aimed to propose an approach for detecting violent acts which could be applied in Vietnam's environment. We developed a novel violence detection method fusing Deep Learning and optical flow on a benchmark dataset containing video of a small group of people (from 1 to 3) captured using surveillance cameras. Besides, we also extend the mentioned standard dataset to ensure its capability of evaluating the generalization of models. Experimental results show that our method outperforms in one test set and maintains the top 3 in accuracy in the remaining test set compared to experimented state-of-the-art violence detection methods.

**TABLE OF CONTENTS**

## LIST OF FIGURES

# LIST OF TABLES

## LIST OF TERMS AND ABBREVIATIONS

| Terms and abbreviations | Meaning |
|---|---|
| Centroid | An individual measurable property or characteristic of a phenomenon (In our case, video features are used to detect activities from surveillance videos |
| Computer Vision | An interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos |
| Dimension | A measurable extent of some kind, such as length, breadth, depth, or height |
| Feature | An individual measurable property or characteristic of a phenomenon (In our case, video features are used to detect activities from surveillance videos |
| Fusion | Integration of different information collected |
| Movement | An act of changing physical location or position by the object in the videos |
| Optical flow | The pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene |
| Spatio-temporal data | Data that relate to both space and time |
| Speed | Movement speed of the object |

# CHAPTER 1. INTRODUCTION

## 1.1 Problem

Violence is a global phenomenon that has caused lots of damage to people and property. It results in the deaths of more than 1.6 million people each year, making it one of the leading causes of death worldwide. While no country is untouched by violence, the vast majority of its resultant deaths occur in low- to middle-income countries, many of which are stricken with internal conflicts. However, it should be kept in mind that violent deaths cannot simply be attributed to war, and more than 80% of such deaths occur outside of armed conflicts. Violence has also shown to be an incredibly costly issue, and in 2015 alone the total impact of violence on the world economy was estimated at 13.6 trillion USD – a figure which is equivalent to 13.3% of world GDP. High levels of violence and crime in regions such as Southern Africa are often the symptoms of underlying social, economic, and political challenges such as social inequality, rapid urbanization, poverty, unemployment, and institutional shortcomings. The adverse effects of violence on a country are harmful not only to its citizens but to the well-being of the community and country as a whole. In many countries, the impact of violence has significantly and directly reduced economic growth and poses an obstacle to reducing poverty, while violence also causes profound psychological and physical trauma, reducing the quality of life for all of the society [1].

Nowadays, to contribute to the prevention of acts of violence, surveillance cameras are widely used. However, it is impossible to spend human resources to check captured videos 24/7, hence, this led to a problem: Automated violence detection system on surveillance cameras.

## 1.2 Current solutions and their limitations

Surveillance and anomaly detection have become more important as the quantity of video data has grown rapidly [2]. When compared to regular activity, such aberrant occurrences are uncommon. As a result, creating automated video surveillance systems for anomaly detection has become a need to reduce labour and time waste. Detecting abnormalities in videos is a difficult job since the term “anomaly” is often imprecise and poorly defined [3]. They differ greatly depending on the conditions and circumstances in which they occur. Bicycling on a standard route, for example, is a typical activity, but doing so in a walk-only lane should be noted as unusual. The uneven internal occlusion is a noteworthy, yet difficult to explain characteristic of the abnormal behaviour. Furthermore, owing to its large

dimensionality, resolution, noise, and rapidly changing events and interactions, video data encoding and modelling are more challenging. Other difficulties include lighting changes, perspective shifts, camera movements, and so on [4].

Violence detection is one of the most crucial elements of video-based anomaly detection (Khan et al., 2019). The usage of video cameras to monitor individuals has become essential due to the rise in security concerns across the globe, and early detection of these violent actions may significantly minimize the dangers. A violence detection system's primary goal is to identify some kind of aberrant behaviour that fits under the category of 'violence' [5]. If an event's conduct differs from what one anticipates, it is considered violent. A person striking, kicking, lifting the other person, and so on are examples of such anomalies [6]. Since human monitoring of the complete video stream is impractical owing to the repetitive nature of the work and the length of time required, automated identification of violent events in real-time is required to prevent such incidents [7].

Given the similarities, there seem to be confusion between the terms 'Action recognition' and 'Violence detection'. Action recognition is, simply put, a technology that can identify human actions. Human activities are categorized into four groups based on the intricacy of the acts and the number of bodily parts engaged in the action: Gestures, actions, interactions, and group activities. A gesture is a series of motions performed with the hands, head, or other body parts to convey a certain message. A single person's actions are a compilation of numerous gestures. Interactions are a set of human activities involving at least two people. When there are two actors are involved, one should be a human and the other may be a human or an object. When there are more than two participants and one or more interacting objects, group activities involve a mix of gestures, actions, or interactions [3].

On the other hand, violence detection is a specific issue within the larger topic of 'action recognition'. The goal of violence detection is to identify whether or not violence happens in a short amount of time automatically and efficiently. The usage of video cameras to monitor individuals has become essential due to the rise in security concerns across the globe, and early detection of these violent actions may significantly minimize the dangers. A violence detection system's primary goal is to identify some kind of aberrant behaviour that fits under the category of 'violence '[1]. If an event's conduct differs from what one anticipates, it is considered violent. A person striking, kicking, lifting the other person, and so on are examples of such anomalies [8]. Since human monitoring of the complete video stream is impractical owing to the repetitive nature of the work and the length of time required, automated identification of violent events in real-time is required

to prevent such incidents [2].

However, it should be noted that detecting abnormalities in videos is a difficult job since the term “anomaly” is often imprecise and poorly defined [9]. They differ greatly depending on the conditions and circumstances in which they occur. Bicycling on a standard route, for example, is a typical activity, but doing so in a walk-only lane should be noted as unusual. The uneven internal occlusion is a noteworthy, yet difficult to explain characteristic of the abnormal behaviour. Furthermore, owing to its large dimensionality, resolution, noise, and rapidly changing events and interactions, video data encoding and modelling are more challenging. Other difficulties include lighting changes, perspective shifts, camera movements, and so on [10]. Even so, despite the difficulties, creating automated video surveillance systems for anomaly detection has become a need to reduce labour and time waste.

From our survey to automatically detect violence in general, based on features extraction methods, there are 2 main approaches: handcrafted and deep learning. Overall, deep learning methods have achieved significantly higher accuracies and come in more flexibility compared to that handcrafted.

## 1.3 Goal and solution

The main purpose of this thesis is to develop an end-to-end system for automatically detecting violent behaviors from videos captured by surveillance cameras. Based on our survey from 1.2, we selected the deep learning approach because of its high accuracy and flexibility.

## 1.4 Thesis contributions

Our thesis has 3 main contributions as following:

1. Collecting more samples for test sets of the AICS - violence dataset [11].
2. Proposing candidate boxes extraction method to focus on human groups.
3. Proposing a new fusion method to combine features obtained by optical flow and 3D Convolutional Neural Network (3D CNN)

## 1.5 Thesis structure

The rest of your thesis are organized as follows:
Chapter 2 depicts theoretical background of our proposed method including machine learning, deep learning and motion estimation methods.
In chapter 3, our proposed violence detection approach are described.
Chapter 4 presents evaluation results of SOTA on violence detection as well as our proposed method on the AICS - violence and a standard benchmark dataset.
Finally, in chapter 5, we concludes the results of our work and suggest future

research directions.

# DIVISION OF WORK

| ID | Work | Assignee |
|---|---|---|
| 1 | Survey handcrafted and machine learning methods for violence detection | Thuy |
| 2 | Survey deep learning methods for violence detection | Hung |
| 3 | Collected more samples for test sets of the AICS - violence dataset | Hung, Thuy |
| 4 | Implement candidate box extraction | Hung |
| 5 | Implement and evaluate the thesis's proposed fusion method | Hung, Thuy |
| 6 | Evaluate baseline methods on the AICS - violence dataset | Hung, Thuy |
| 7 | Evaluate the thesis's proposed method on a standard dataset | Thuy |

# CHAPTER 2. THEORETICAL BACKGROUND

## 2.1 Problem context

As a result of violence being such a complex phenomenon, there is no clear definition for it. Therefore, it is often understood differently by different people in different contexts - such as those from different countries, cultures, or belief systems.

The World Health Organization (WHO) defines violence as "The intentional use of physical force or power, threatened or actual, against oneself, another person, or against a group or community, that either results in or has a high likelihood of resulting in injury, death, psychological harm, maldevelopment or deprivation" [8].

On the basis of the WHO's definition of violence, an elaborate "typology of violence" (as shown in Figure 2.1) has been developed that characterizes different categories and types of violence, as well as the links between them (allowing for a holistic approach to intervention).

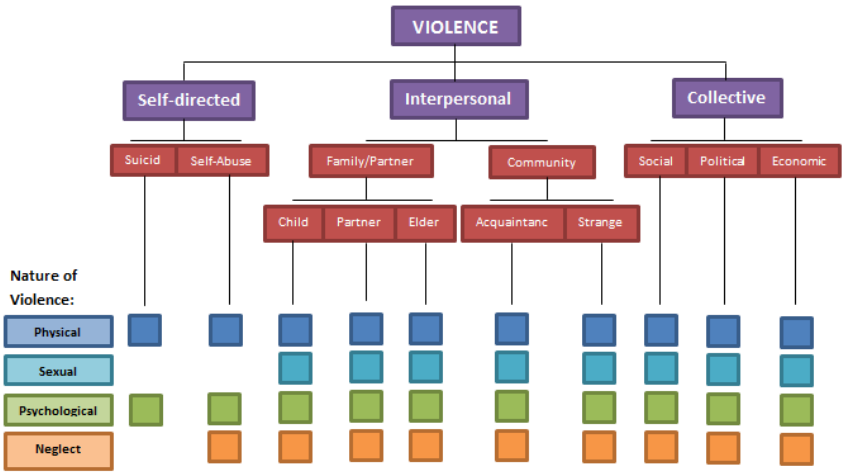



**Figure 2.1:** Typology of violence [8]

It divides violence into three broad categories according to who the perpetrators and victims are of violent acts:

- Self-Directed violence: Self-directed violence refers to violent acts a person inflicts upon him- or herself, and includes self-abuse (such as self-mutilation) and suicidal behaviour (including suicidal thoughts, as well as attempted and completed suicide).
- Interpersonal violence: Interpersonal violence refers to violence inflicted by another individual or by a small group of individuals. It can be further divided into two subcategories: Family and intimate partner violence and Community

violence.

- Collective violence: Collective violence can be defined as the instrumental use of violence by people who identify themselves as members of a group – whether this group is transitory or has a more permanent identity – against another group or set of individuals, in order to achieve political, economic or social objectives. This can manifest in a number of forms, such as genocide, repression, terrorism and organised violent crime.

By looking more closely at the nature of acts of violence, these three categories can be further divided into four, more specific, types of violence:

- Physical violence: Physical violence is the intentional use of physical force, used with the potential for causing harm, injury, disability or death. This includes, but is not limited to: scratching, pushing, shoving, grabbing, biting, choking, shaking, slapping, punching, hitting, burning, use of a weapon, and use of restraint or one's body against another person. This type of violence does not only lead to physical harm, but can also have severe negative psychological effects – for example, if a child is frequently a victim of physical violence at home, he or she can suffer from mental health problems and be traumatised as a consequence of this victimisation.
- Sexual violence: Sexual violence involves a sexual act being committed or attempted against a victim who has not freely given consent, or who is unable to consent or refuse. This includes, but is not limited to: forced, alcohol/drug-facilitated or unwanted penetration, sexual touching, or non-contact acts of a sexual nature. A perpetrator forcing or coercing a victim to engage in sexual acts with a third party also qualifies as sexual violence. This type of violence can also lead to physical harm, and in most cases has severe negative psychological effects too.
- Psychological violence: Psychological violence (also referred to as emotional or mental abuse) includes verbal and non-verbal communication used with the intent to harm another person mentally or emotionally, or to exert control over another person. The impact of psychological violence can be just as significant as that of other, more physical forms of violence, as the perpetrator subjects the victim to behaviour which may result in some form of psychological trauma, such as anxiety, depression or post-traumatic stress disorder. This includes, but is not limited to: expressive aggression (e.g., humiliating and degrading), coercive control (e.g., limiting access to things or people, and excessive monitoring of a person's whereabouts or communications), threats

of physical or sexual violence, control of reproductive or sexual health, and exploitation of a person's vulnerability (e.g., immigration status or disability).

- Neglect: Neglect, or deprivation, is a type of abuse which occurs when someone has the responsibility to provide care for an individual who is unable to care for him- or herself, but fails to do so, therefore depriving them of adequate care. Neglect may include the failure to provide sufficient supervision, nourishment, or medical care, or the failure to fulfil other needs for which the victim cannot provide themselves. Neglect can lead to many long-term side effects such as: physical injuries, low self-esteem, attention disorders, violent behaviour, physical and psychological illness, and can even result death.

These four types of violence can occur in each of the previously mentioned broad categories, and their subcategories (except for self-directed violence). The Figure 2.1 illustrates these links between types of violence and the nature of violent acts. Horizontally the graphic shows who is affected, while vertically it describes how they are potentially affected.

**Our thesis is solely directed towards detecting 'Physical violence', hence, from here onward, the word 'violence' indicates the phenomena of 'Physical violence' specifically.**

## 2.2 Related works

In this section, methods of violence detection are analysed to completely disassemble the present condition and anticipate the emerging trends of violence discovery research by providing a comprehensive assessment of the video violence detection methods that have been described in state-of-the-art researches. Current techniques, state-of-the-art violence detection techniques which were published between approximately from 2015 to 2021, into three categories based on their methodologies: conventional methods, end-to-end deep learning-based methods, and machine learning-based methods.

Scientists have presented various approaches and methods for detecting violent or unusual occurrences, citing the fast rise in crime rates as an example of the need for more efficient identification. Various methods for detecting violence have been developed in past few years. Based on the classifier employed, violence detection methods are divided into three categories: violence detection using machine learning, violence detection using SVM, and violence detection using deep learning [12]. A methodology for detecting objects and a method for extracting features are also discussed.

### 2.2.1 Overview of feature extractions

This section goes through the feature descriptors that violence detection papers utilized in their research as well as other recent state-of-the-art descriptors.

The fundamental components for detecting activity from the video are video features. The dataset and characteristics collected from video to evaluate the pattern of activity have a direct impact on the methodology's accuracy. For example, in combat situations, the movement of various objects increases faster. The movement of objects in a typical setting is normal and not too rapid. The direction of item movement in relation to time and space is also utilized to investigate unusual occurrences. Such features can be divided in to two main categories based on how they are collected:

- Handcrafted features: Manually engineered by the data scientist
- Learning features: Automatically obtained from a machine or deep learning algorithm

Suppose, for an image classification task, where the target is to classify cats from dogs, the developer is faced with a dilemma on how to input the data to the classifier. There are two options:

1. The raw pixel data (The issue with this approach is that the feature space is vast, which makes it hard for models to generalize)
2. Attempt to extract features from the image so that the feature space can be reduced. If the second option is chosen, two methods are available for implementation:
    - Manually define a set of features and extract them. Some examples include edge detection as in Figure 2.2, corner detection, histograms, etc. The problem with this approach is that nothing guarantees that the number of corners is a good descriptor for classifying cat and dog images.

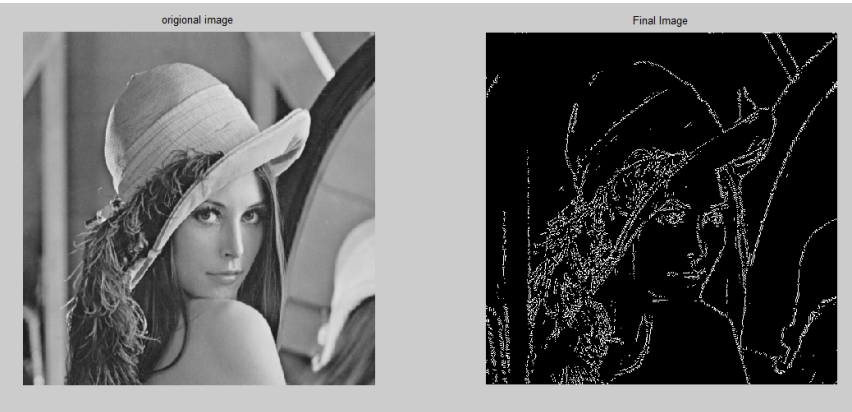



**Figure 2.2:** Edge detection [13]

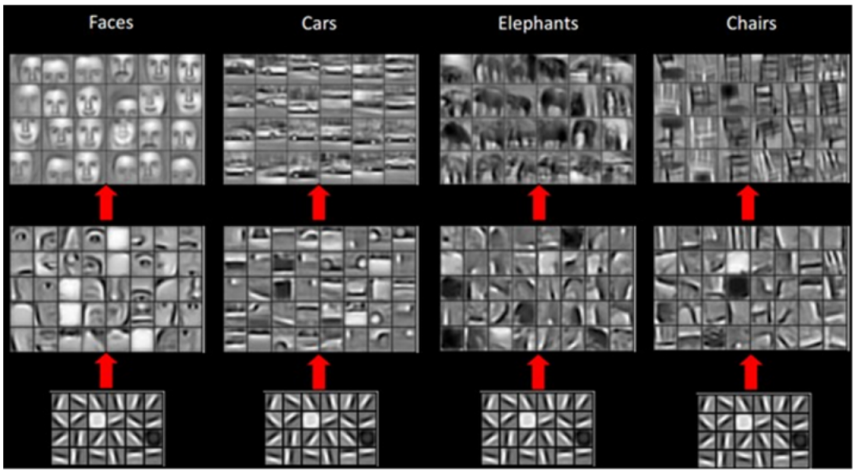



**Figure 2.3:** Feature extraction using Deep Learning [14]

- The alternative is to train a deep learning model to identify and extract useful features for this specific classification task. (This is exactly what a Convolutional Neural Network does as illustrated in Figure 2.3)

Traditionally, the first approach was used extensively in Machine Learning. However, that changed with the arrival of Deep Learning.

In a nutshell, since 'learned features' are extracted automatically to solve a specific task, they are extremely effective at it. In fact, deep learning models that perform feature extraction and classification outperform models that classify manually extracted features by a large margin. This is one of the reasons why deep learning is so popular. On the other hand, we have no control on what features the model will extract from the data. In many cases these features are only good for classifying the data and have no real-world interpretation. They are only good for the task that they were trained for.

A feature descriptor is an algorithm which takes an image and outputs feature descriptors/feature vectors. Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical "fingerprint" that can be used to differentiate one feature from another. Following are a few well-recognized feature descriptors

### a, Histogram of oriented gradients (HOG)

HOGs are feature descriptors for object identification and localization that can compete with DNN's performance [15]. The gradient direction distribution is utilized as a feature in HOG. Because the brightness of corners and edges vary greatly, calculating the gradient together with the directions may assist in the detection of this knowledge from the images.

### b, Histogram of optical flow (HOF)

A pattern of apparent motion of objects, surfaces, and edges is produced as a result of the relative motion between an observer and a scene. This process is called Optical Flow. The histogram of oriented optical flow (HOF) [16] is an optical flow characteristic that depicts the series of events at each point in time. It is scale-invariant and unaffected by motion direction.

### c, SPACE –time interest points

Laptev and Lindeberg and Laptev proposed the space–temporal interest point detector by expanding the Harris detector. A second-moment matrix is generated for each spatiotemporal interest point after removing points with high gradient magnitude using a 3D Harris corner detector [17], [18]. This descriptor's characteristics are used to describe the spatiotemporal, local motion, and appearance information in volumes.

### d, MoSIFT

MoSIFT [19] is an extension of the popular SIFT [20] image descriptor for video. The standard SIFT extracts histograms of oriented gradients in the image. The 256-dimensional MoSIFT descriptor consists of two portions: a standard SIFT image descriptor and an analogous HOF, which represents local motion. These descriptors are extracted only from regions of the image with sufficient motion. The MoSIFT descriptor has shown better performance in recognition accuracy than other state-of-the-art descriptors [19] but the approach is significantly more computationally expensive than STIP.

### e, Violence flow descriptor

The violence flow, which utilizes the frequencies of discrete values in a vectorized form, is an essential feature descriptor. This is different from other descriptors in that instead of assessing magnitudes of temporal information, the magnitudes are compared for each, resulting in much more meaningful measurements in terms of the previous frame [21]. Instead of looking at local appearances, the similarities between flow-magnitudes in terms of time are investigated.

### f, Bag-of-Words (BoW)

The Bag-of-Words (BoW) method, which originated in the text retrieval community [18] has lately gained popularity for a picture [22] and video comprehension [23]. Each video sequence is represented as a histogram over a collection of visual words in this method, which results in a fixed-dimensional encoding that can be analysed with a conventional classifier. The cluster centres produced via k-means clustering

across a large collection of sample low-level descriptors are usually described as the lexicon of visual words in a learning phase [24].

#### g, Motion boundary histograms (MBH)

By measuring derivatives independently for the horizontal and vertical components of the optical flow, Dalal et al. developed the MBH descriptor [16] for human detection. The relative motion between pixels is encoded by the descriptor. Because MBH depicts the gradient of the optical flow, information regarding changes in the flow field (i.e., motion boundaries) is preserved while locally constant camera motion is eliminated. MBH is more resistant to camera motion than optical flow, making it better at action detection.

### 2.2.2 Violence detection using hand-crafted and machine learning techniques

#### a, Fast Fight Detection

In the field of computer vision, action recognition has now become a relevant research area. However, the identification of particular events with immediate practical application, such as fighting or general violent conduct, has received much less attention. In certain situations, such as prisons, mental institutions, or even camera phones, video surveillance may be very helpful [25]. Given the circumstances, a new technique for detecting violent sequences was suggested by Gracia et al in the article 'Fast Fight Detection' [26]. To distinguish between fighting and non-fighting episodes, features derived from motion blobs are utilized.

As illustrated in Figure 2.4, Fast Fight Detection model [26] uses motion blobs for extracting features. The proposed method was assessed using three different datasets as 'Movies' dataset with 200 video clips [27], the 'Hockey fight' dataset that consists of 1000 video clips [28], and the 'UCF-101' dataset of realistic action videos collected from YouTube [25]. The proposed method was compared with other five related methods as Bag of Words (BoW) [29] using scale-invariant feature transform (MoSIFT) [30] and STIP [31] features, Violent Flows (ViF) method [32], Local Motion method [33], also variant v-1 and variant v-2 methods that applied KNN, AdaBoost, and Random Forest classifiers. Although the proposed technique falls short from a perspective of performance, it has a much quicker calculation time, making it suitable for practical uses [12], [26].

#### b, RIMOC (Rotation-Invariant Feature Modelling Motion Coherence)

Jerky and unstructured motion are often present in film with violent human behaviours due to the fact that aggressive occurrences are difficult to quantify owing to their unpredictability and sometimes need high-level interpretation. In order to capture its structure and distinguish the unstructured movements, a new

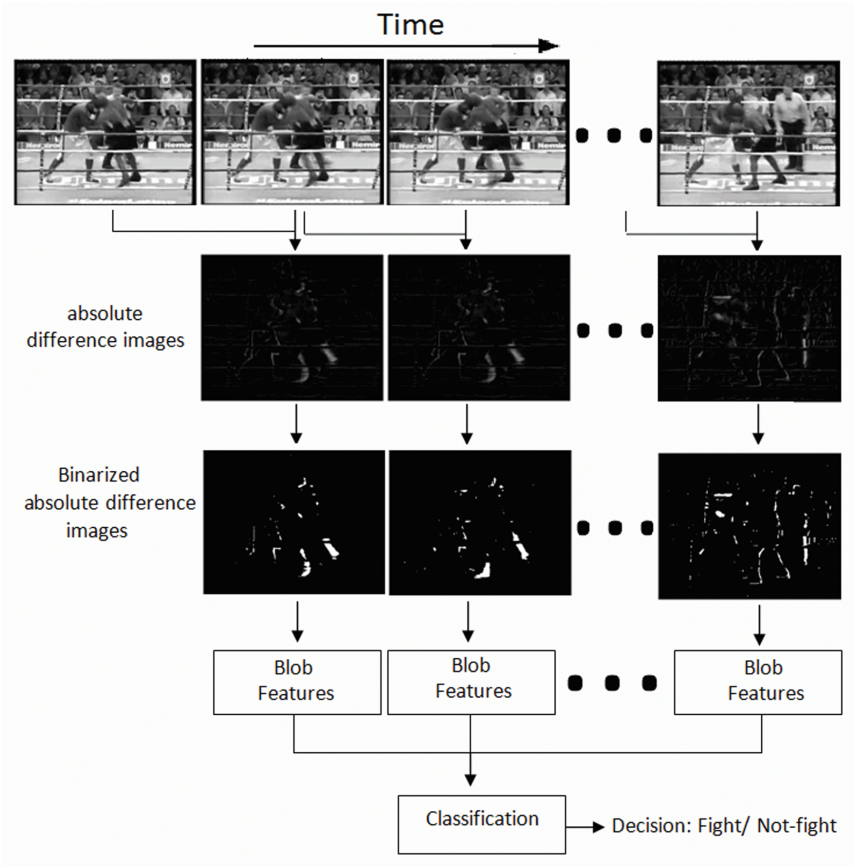



**Figure 2.4:** Fast Fight Detection architecture [26]

problem-specific ‘Rotation-Invariant feature modelling Motion Coherence’ (RIMOC) was suggested in 2016 [34].

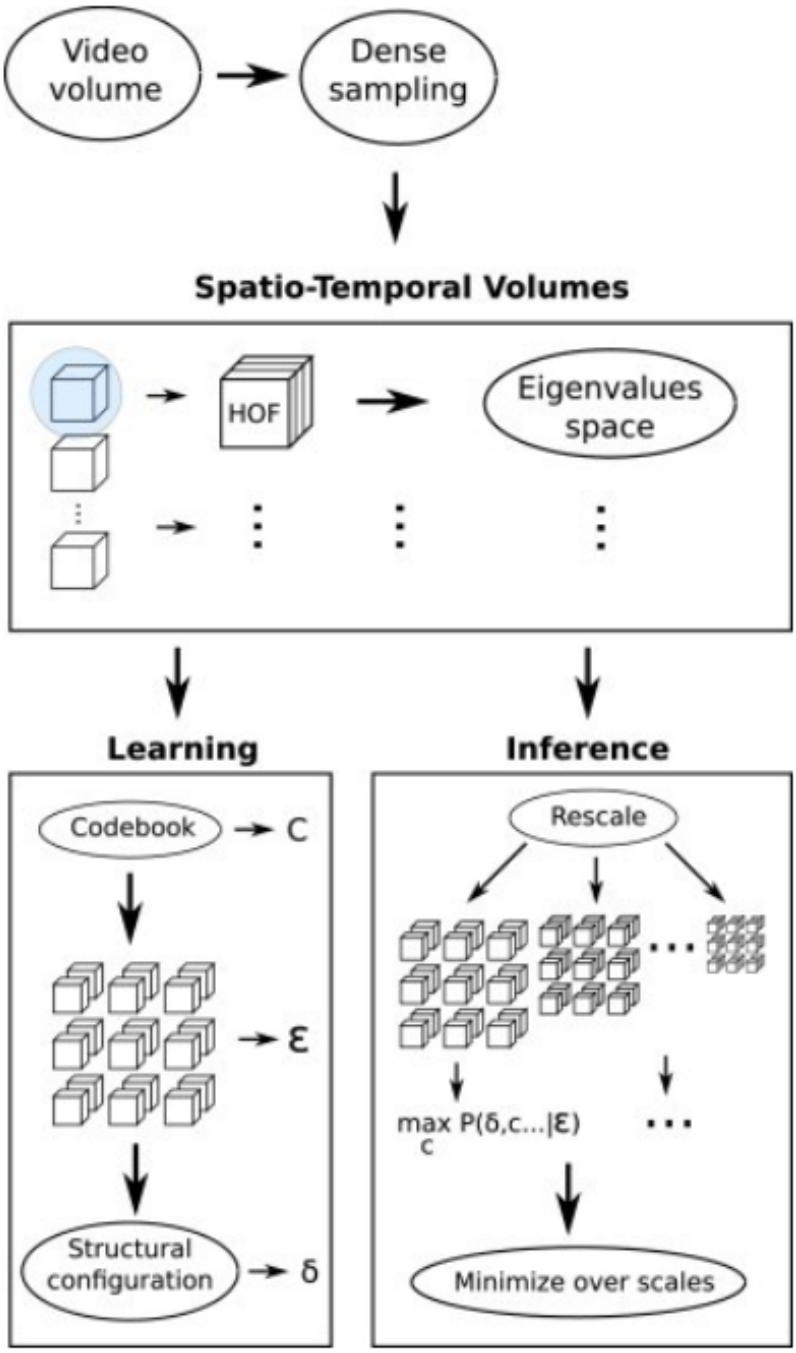



**Figure 2.5:** RIMOC architecture [34]

As shown in Figure 2.5, RIMOC is based on eigenvalues calculated locally and densely from second-order statistics of Histograms of Optical Flow vectors from successive temporal instants, then embedded into a spheric Riemannian manifold. In a poorly supervised way, the proposed RIMOC feature is utilized to develop statistical models of normal coherent movements. Events with irregular mobility may be identified in space and time using a multi-scale approach combined with an inference-based method, making them ideal candidates for aggressive events. There is no special dataset available for violence and aggressive behaviour detection. A big dataset is produced for this goal, which comprises of sequences from two

distinct sites: an in-lab fake train and a genuine underground railway line, real train, and then four datasets are formed: fake train, real train, real train station, and real-life settings. These datasets are used in the trials, and the findings indicate that the suggested approach outperforms all state-of-the-art methods in terms of ROC per frame and false-positive rate [34], [12].

### c, Motion Direction Inconsistency-Based Fight Detection for Multi-view Surveillance Videos

Yao et al. present a multiview fight detection technique based on optical flow statistical features and random forest [35]. This technique may provide fast and reliable information to cyber-physical monitoring systems. Motion Direction Inconsistency (MoDI) and Weighted Motion Direction Inconsistency (WMoDI), two new descriptors, are developed to enhance the performance of current techniques for films with various filming perspectives and to address misjudgement on nonfighting activities like jogging and chatting.

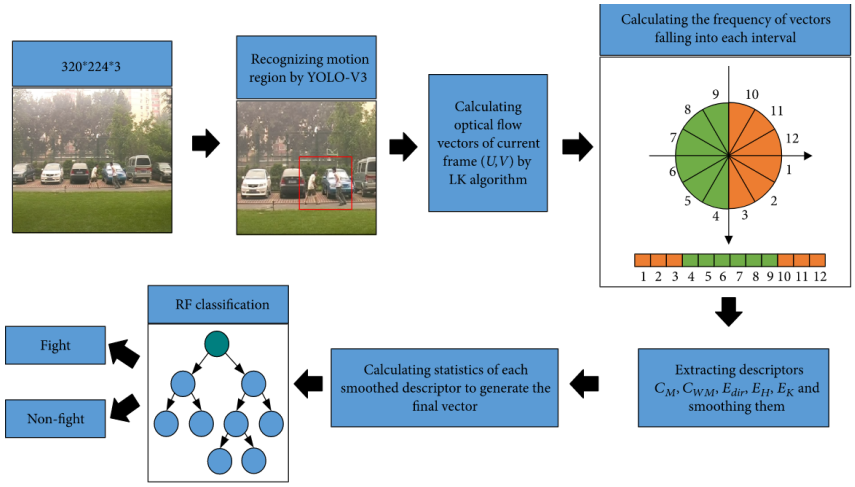



**Figure 2.6:** MoDI architecture [35]

As depicted in Figure 2.6, the motion regions are first marked using the YOLO V3 method, and then the optical flow is calculated to retrieve descriptors. Finally, Random Forest is utilized to classify data using statistical descriptor features. The experiments were performed using CASIA Action Dataset [36] and the UT-Interaction Dataset [37]. All films of fighting, as well as 15 additional videos in five categories, were chosen from the CASIA Action Dataset. The findings demonstrated that the proposed approach improves violence detection accuracy and reduces the incidence of missing and false alarms, and it is robust against films with various shooting perspectives [35], [12].

### d, Fast Face Detection

Fast Face Detection [38] is developed to accomplish the objective of identifying faces in violent videos to improve security measures.

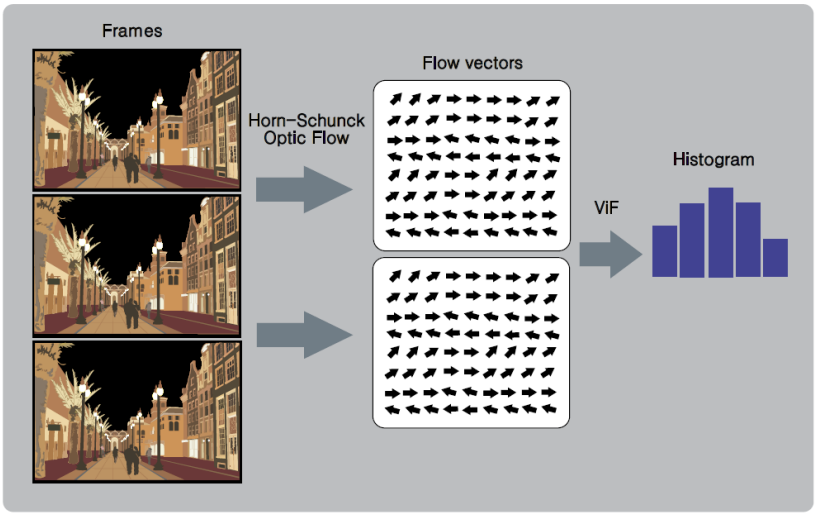



**Figure 2.7:** Fast Face Detection architecture [38]

As shown in Figure 2.7, for the initial step of violent scene identification, the authors utilized the ViF descriptor [15] in conjunction with Horn-Schunck [39] Then, to enhance the video quality, the non-adaptive interpolation super-resolution algorithm was used, followed by the firing of the Kanade-Lucas-Tomasi (KLT) face detector [16]. The authors used CUDA to parallelize the super-resolution and face detection algorithms in order to achieve a very fast processing time. The Boss Dataset [17] was utilized in the tests, as well as a violence dataset based on security camera footage. Face detection yields encouraging results in terms of area under the curve (AUC) and accuracy [12].

### e, Recognizing violent activity without decoding video streams

Most conventional activity identification techniques' motion target detection and tracking procedures are often complex, and their applicability is limited. To solve this problem, a fast method of violent activity recognition is introduced which is based on motion vectors [40].

First and foremost, the motion vectors were directly retrieved from compressed video segments. The motion vectors' characteristics in each frame and between frames were then evaluated, and the Region Motion Vectors (RMV) descriptor was produced. To classify the RMV to identify aggressive situations in movies, a SVM classifier with radial basis kernel function was used in the final step. In order to evaluate the proposed method, the authors created VVAR10 dataset that consists

of 296 positive samples and 277 negative samples by sorting video clips from UCF sports [40], UCF50 [41], HMDB51 [20] datasets. Experiments have shown that the proposed method can detect violent scenes with 96.1% accuracy in a short amount of time. That is why the proposed method can be used in embedded systems.

### f, MoBSIFT

Local spatiotemporal feature extractors have been explored in previous studies; nevertheless, they come with the overhead of complicated optical flow estimates. Despite the fact that the temporal derivative is a faster alternative to optical flow, it produces a low-accuracy and scale-dependent result when used alone. As a result, a cascaded approach of violence detection was suggested, based on motion boundary SIFT (MoBSIFT) and a movement filtering method [42].

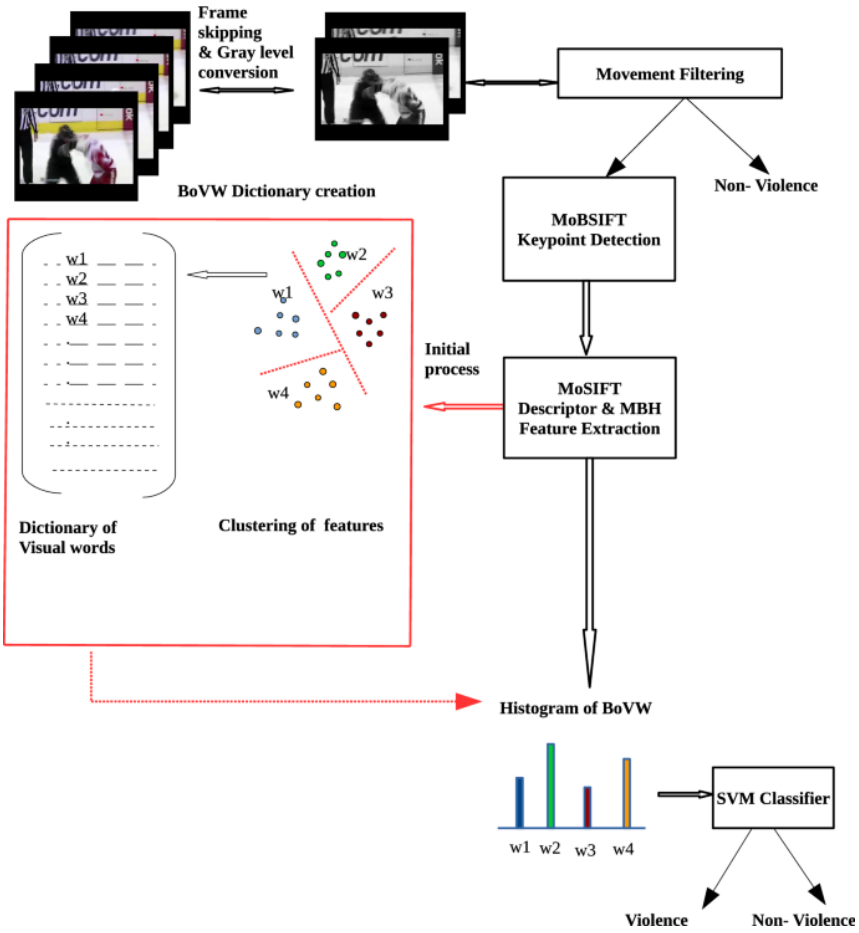


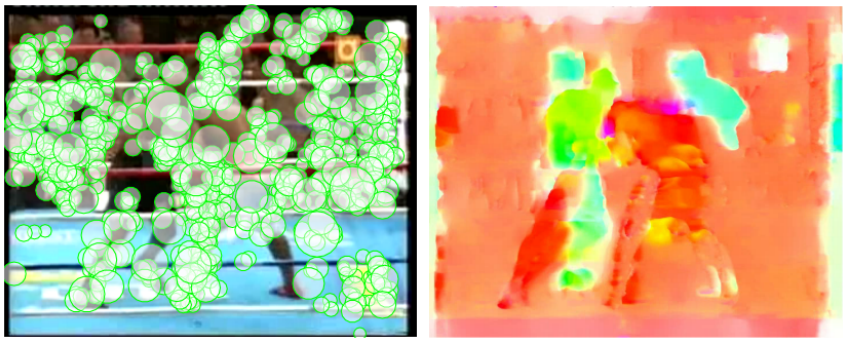

**Figure 2.8:** MoBSIFT architecture [42]

As illustrated in Figure 2.8, the surveillance films are examined using a movement filtering algorithm based on temporal derivatives in this approach, which avoids feature extraction for most peaceful activities. Only filtered frames may be suitable for feature extraction. Motion boundary histogram (MBH) is retrieved and merged with SIFT [20] and histogram of optical flow feature to create MoBSIFT descriptor.

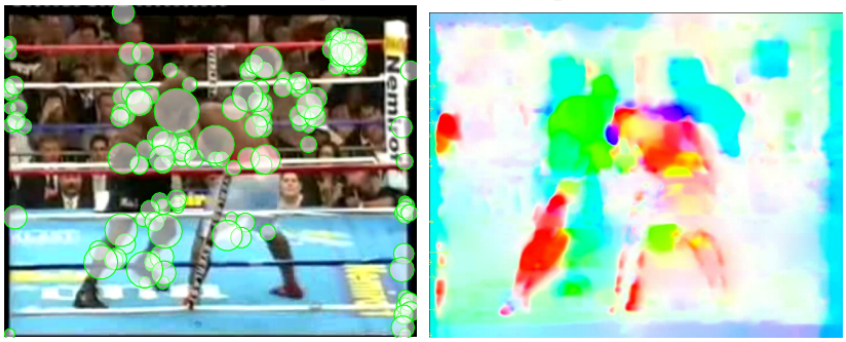
The models were trained using MoBSIFT and MPEG Flow (MF) [43] descriptors using AdaBoost, RF, and SVM classifiers. Because of its great tolerance to camera motions, the suggested MoBSIFT surpasses current techniques in terms of accuracy. The use of movement filtering in conjunction with MoBSIFT has also been shown to decrease time complexity [42].

### g, Crowd Violence Detection

In computer vision, Lagrangian theory [44] offers a comprehensive set of tools for evaluating non-local, long-term motion information. Authors propose a specialized Lagrangian method for the automatic identification of violent situations in video footage based on this theory [44]. The authors propose a new feature based on a spatiotemporal model that utilizes appearance, background motion correction, and long-term motion information and leverages Lagrangian direction fields as shown at 2.9. They use an expanded bag-of-words method in a late-fusion way as a classification strategy on a per-video basis to guarantee suitable spatial and temporal feature sizes.



**a)** without global motion compensation



**b)** using global motion compensation

**Figure 2.9:** Comparison of results with and without global motion compensations [44]

Experiments were conducted in three datasets as 'Hockey Fight' [28], 'Violence in Movies' [28], 'Violent Crowd' [45], and 'London Metropolitan Police (London Riots 2011)' [46] datasets. Multiple public benchmarks and non-public, real-world data from the London Metropolitan Police are used to verify the proposed system.

Experimental results demonstrated that the implementation of Lagrangian theory is a useful feature in aggressive action detection and the classification efficiency rose over the state-of-the-art techniques like two-stream convolutional neural network (CNN, ConvNet), ViF, HoF+BoW with STIP, HOG+BoW with STIP, etc. in terms of accuracy and ROC-AUC measure.

### h, A Video-Based DT–SVM School Violence Detecting Algorithm

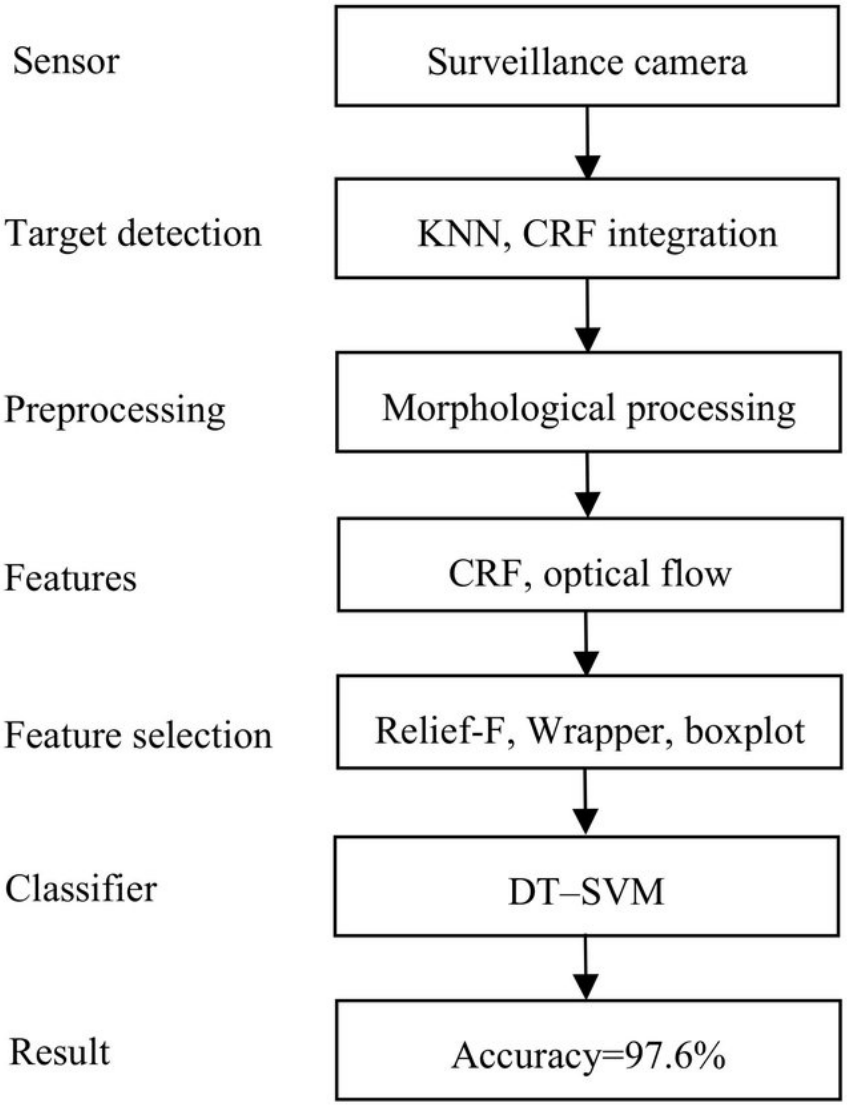



**Figure 2.10:** Video-based DT-SVM architecture [47]

A new method for identifying school violence was proposed by Ye et al. in 2020 [47]. As depicted in Figure 2.10, this technique uses the KNN algorithm to identify foreground moving objects and then uses morphological processing methods to pre-process the identified targets. Then, to optimize the circumscribed rectangular frame of moving objects, a circumscribed rectangular frame integrating technique was proposed. To explain the distinctions between school violence and everyday activities, rectangular frame characteristics and optical-flow features

were retrieved. To decrease the feature dimension, the Relief-F and Wrapper algorithms were applied. SVM is used as a classifier, and a 5-fold cross-validation was conducted. The results show 94.4 percent precision and 89.6 percent accuracy. In order to improve recognition performance, a DT–SVM two-layer classifier is created. Authors utilized boxplots to identify certain DT layer characteristics that can differentiate between everyday activities and physical violence. The SVM layer conducted categorization for the remaining activities. The accuracy of this DT–SVM classifier was 97.6 percent, while the precision was 97.2 percent, indicating a considerable increase [47], [12].

**i, Gaussian Model of Optical Flow (GMOF)**

At the time existing vision-based techniques focus mostly on detecting violence and make very little attempt to pinpoint its location. To tackle this problem, Zhange et al. presented a quick and robust method for identifying and localizing violence in surveillance situations to address this issue [48].

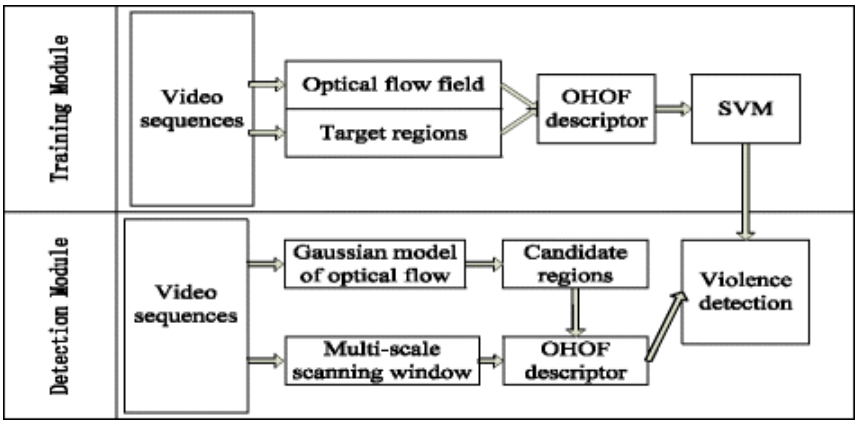



**Figure 2.11:** GMOF architecture [48]

A Gaussian Model of Optical Flow (GMOF) is suggested for this purpose in order to extract potential violent areas, which are adaptively modelled as a departure from the usual crowd behaviour seen in Figure 2.11. Following that, each video volume is subjected to violence detection by intensively sampling the potential violent areas. The authors also propose a new descriptor called the Orientation Histogram of Optical Flow (OHOF), which is input into a linear SVM for classification to differentiate violent events from peaceful ones. Experimental results on violence video datasets like 'Hockey fight' [28], 'BEHAVE' [49], 'CAVIAR' [50] have shown the superiority of the proposed methodology over the state-of-the-art descriptors like MoSIFT and SIFT, HOG, HOF, and Combination of HOG and HOF (HNF), in terms of detection accuracy, AUC-ROC, and processing performance, even in crowded scenes.

### j, Violence detection using Oriented VIolent Flows

In order to recognize violence in videos in a realistic manner, a novel feature extraction method named Oriented VIolent Flows (OViF) was proposed by Gao et al [51]. In 2016. In statistical motion orientations, the proposed method fully exploits the motion magnitude change information. The features are selected using AdaBoost, and the SVM classifier is subsequently trained on the features as illustrated in Figure 2.12.

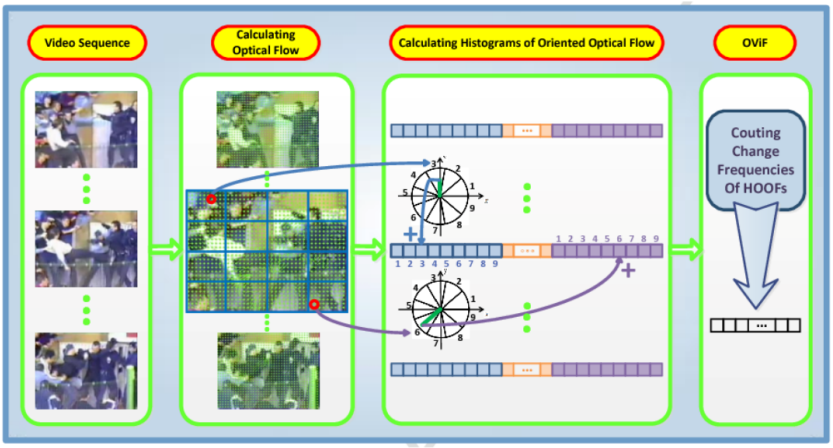



**Figure 2.12:** OViF architecture [51]

Experiments are carried out on the 'Hockey Fight' [28] and 'Violent-Flow' [52] datasets to assess the new approach's performance. The findings indicate that the suggested technique outperforms the baseline methods LTP and ViF in terms of accuracy and AUC. Furthermore, feature and multi-classifier combination methods have been shown to help improve the performance of the violence detector. The experiment results demonstrate that the combination of ViF and OViF using AdaBoost with a combination of Linear-SVM surpasses the state-of-the-art on the Violent-Flows database. The final best violence detection rates are 87.50% and 88.00% on 'Hockey Fight' and 'Violent-Flows' separately using ViF + OViF with Adaboost + SVM.

### k, Spatiotemporal Autocorrelation of Gradients (STACOG)

One of the most important stages in the development of machine learning applications is data representation. Data representation that is efficient aids in better classification across classes. Deepak et al. investigate Spatiotemporal Autocorrelation of Gradients (STACOG) as a handmade feature for extracting violent activity characteristics from surveillance camera videos [53] as described in Figure 2.13.

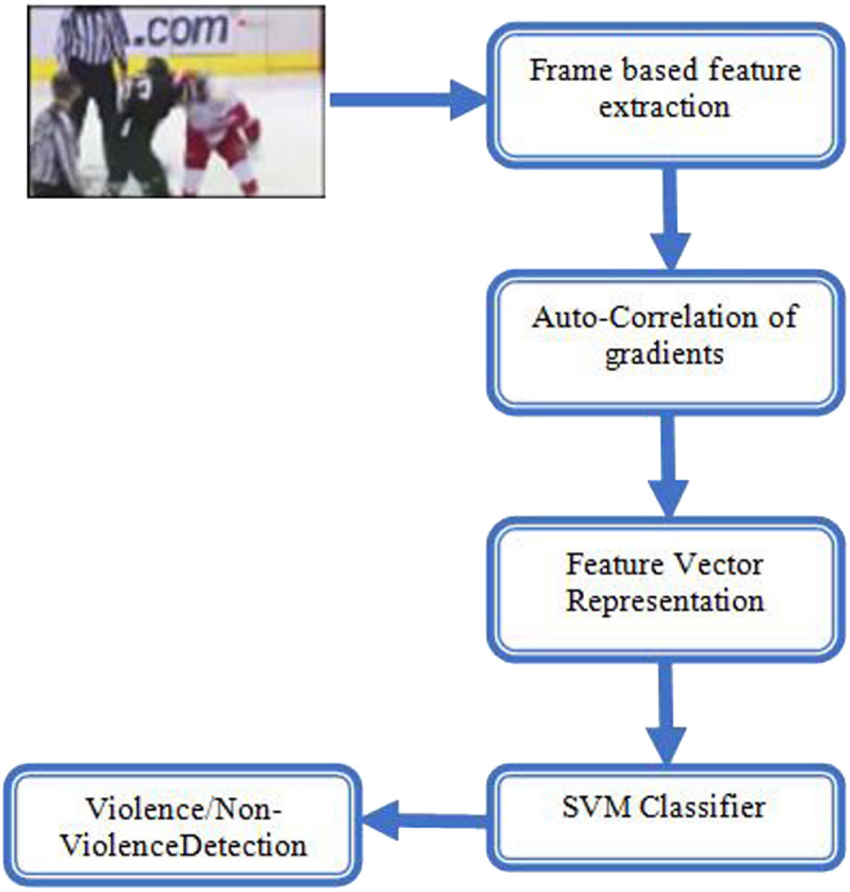



**Figure 2.13:** STACOG architecture [53]

The proposed strategy is divided into two stages:

1. Extraction of STACOG based Features
2. Discriminative learning of violent/non-violent behaviour's using an SVM Classifier

Two well-known datasets were used to test the proposed approach: 'Hockey fight' dataset [28] that contains 1000 video clips and the 'Crowd Violence' Dataset. The proposed 'STACOG features + SVM' model shown 91.38% accuracy in violence detection overcoming state-of-the-art methods like HOF+BoW, HNF+BoF, ViF+SVM, BiLSTM, GMOF, and others.

### l, Automatic real-time video-based surveillance system

In video processing, aggression detection is critical, and a surveillance system that can operate reliably in an academic environment has become a pressing requirement. To solve this problem, a novel framework for an automatic real-time video-based surveillance system is proposed [54].

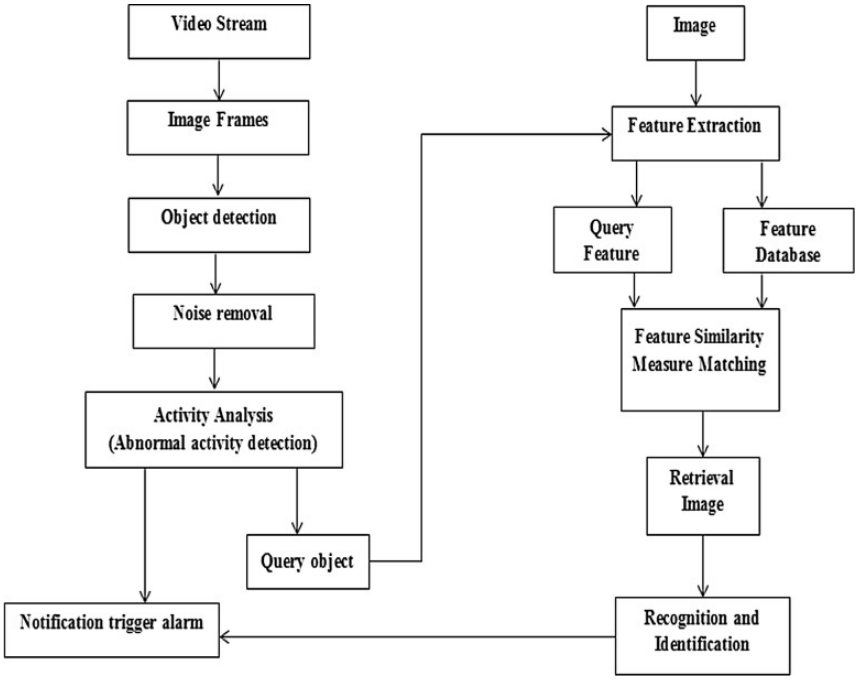



**Figure 2.14:** Automatic real-time video-based surveillance system architecture [54]

As depicted in Figure 2.14, the proposed system is divided into three phases during the development process. The first stage is pre-processing stage that includes abnormal human activity detection and content-based image retrieval (CBIR) in the event that the system identifies unusual student behavior. In the first stage, students are registered by entering their personal data including first name, second name, birthday, course, student id card, and photos. The entered data is stored in a central database for conducting a search when abnormal actions are detected. The video is then turned into frames in the second step. Motion objects are detected using a temporal-differencing method, and motion areas are identified using the Gaussian function. Furthermore, a form model based on the OMEGA equation is employed as a filter for identified items, whether human or non-human. SVM is used to classify human behaviours into normal and abnormal categories. When a person engages in abnormal behaviour, the system issues an automated warning. It also adds a method to get the identified item from the database using CBIR for object detection and verification. Finally, a software-based simulation using MATLAB is performed, and experimental findings indicate that the system performs simultaneous tracking, semantic scene learning, and abnormality detection in an academic setting without the need of humans.

### m, Framework for high-level activity analysis

Song, Kim Park, in 2018, proposed a new framework for high-level activity analysis based on late fusion and multi-independent temporal perception layers,

which is based on late fusion [55]. It is possible to manage the temporal variety of high-level activities using this approach. Multi-temporal analysis, multi-temporal perception layers, and late fusion are all part of the framework.

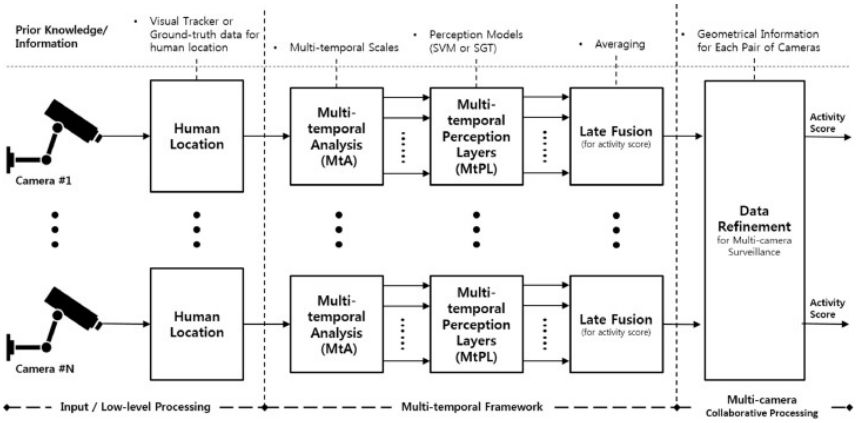



**Figure 2.15:** Framework for high-level activity analysis [55]

As shown in Figure 2.15, based on Situation Graph Trees (SGT) and SVM, authors create two kinds of perception layers (SVMs). Through a phase of late fusion, the data from the multi-temporal perception layers are fused into an activity score. To test the proposed method, the framework is applied to the detection of violent events by visual observation. The experiments are conducted applying three well-known databases: 'BEHAVE' [49], 'NUS–HGA' [56], and a number of YouTube videos depicting real-life situations. The tests yielded an accuracy of 70.2% (SVM), and 87.2% (SGT) in different datasets, demonstrating how the proposed multi-temporal technique outperforms single-temporal approaches. Vashistha, Bhatnagar Khan, in 2018 [57], utilized Linear SVM to categorize incoming video as violent or non-violent, extracting important characteristics like centroid, direction, velocity, and dimensions. Their approach took into account two feature vectors, i.e., ViF and the Local Binary Pattern (LBP). Because calculating LBP or ViF individually it takes less time than combining these feature vectors, their study found that combining LBP and ViF did not offer substantial direction for future development.

### 2.2.3 Violence detection using deep learning techniques

#### a, Violence detection using 3D CNN

To build complicated handmade characteristics from inputs, most techniques require domain expertise. Deep learning methods, on the other hand, may operate directly on raw inputs and extract necessary features automatically. As a result, Ding et al. created a new 3D ConvNets approach for video violence detection, which does not need any previous information in 2014 [58].

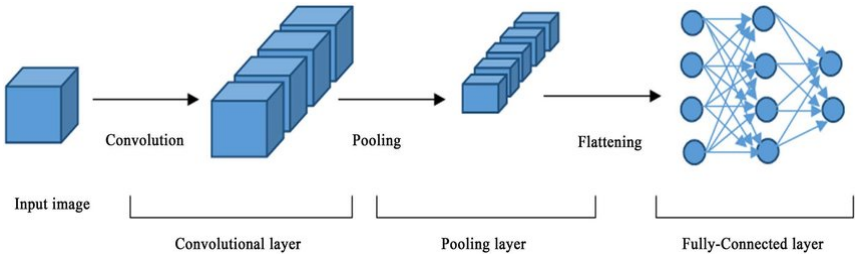



**Figure 2.16:** 3D CNN architecture [58]

As depicted in Figure 2.16, the convolution on the collection of video frames is computed using a 3D CNN, and therefore motion information is retrieved from the input data. The back-propagation technique is used to obtain gradients and the model has been trained to apply supervised learning. Experimental validation was carried out in the context of the 'Hockey fights' [28] dataset to assess the approach. The findings indicate that the approach outperforms manual features in terms of performance.

**b, Real time violence detection using bidirectional LSTM (BDLSTM)**

A real-time violence detection system is presented, which analyses large amounts of streaming data and recognizes aggression using a human intelligence simulation [59].

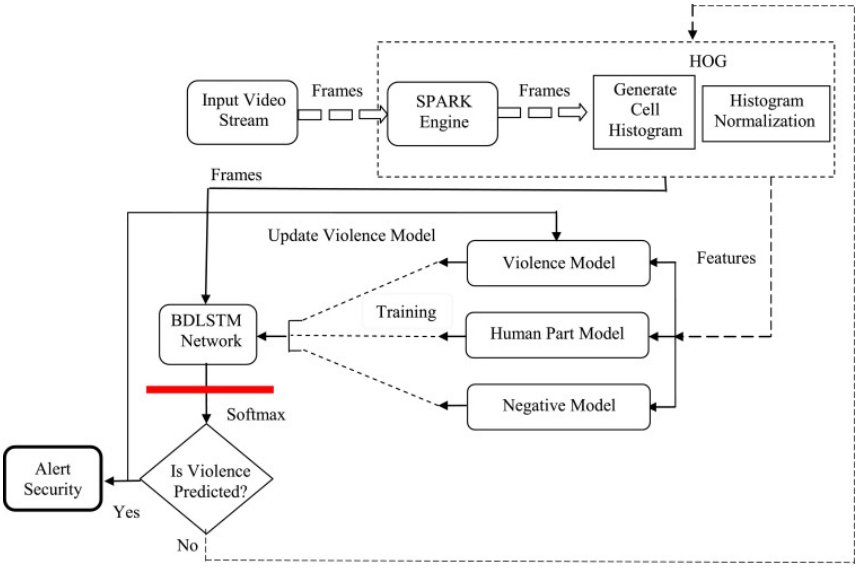



**Figure 2.17:** BDLSTM architecture [59]

As shown in Figure 2.17, the system's input is a massive quantity of real-time video that feeds from various sources, which are analysed using the Spark framework. The frames are split and the characteristics of individual frames are retrieved using the HOG function in the Spark framework. The frames are then labelled based on characteristics such as the violence model, human component

model, and negative model, which are trained using the BDLSTM network for violent scene detection. The data may be accessed in both directions via the bidirectional LSTM. As a result, the output is produced in the context of both past and future data. The violent interaction dataset (VID) is used to train the network, which contains 2314 movies with 1077 fights and 1237 no-fights. The authors also generated a dataset of 410 video episodes with neutral scenes and 409 video episodes with violence. The accuracy of 94.5% in detecting violent behaviour validates the model's performance and demonstrates the system's durability.

**c, Violent scene detection using CNN Deep Audio Features**

Violent scene detection system is proposed that uses CNN built on acoustic information from video clips [60]. CNN is applied in two ways: a as a classifier directly or as a deep acoustic feature extractor.

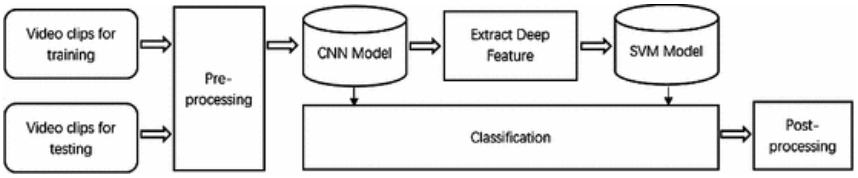



**Figure 2.18:** CNN Deep Audio Features architecture [60]

As illustrated in Figure 2.18, firstly, 40-dimensional Mel Filter-Bank (MFB) is utilized as the input feature to the CNN with their delta and delta-delta. Then the video is converted into short chunks. MFB features are divided into 3 feature channels to explore the local features. Then CNN is used for feature representation. CNN-based features are used to build SVM classifiers. Then the detection of violent scene is performed on each chunk of video. Then the detection is produced by max or min pooling on the segment-level detections. Experiments are performed via 'MediaEval' dataset [61] and results show that the proposed method performs better than the baseline methods: audio only, visual only and audio learned fusion and visual in terms of average precision.

**d, Detect violent videos using Convolutional Long Short-term Memory (ConvLSTM)**

Sudhakaran Lanz (2017) proposed a deep neural network for detecting violent scenes in videos [62].

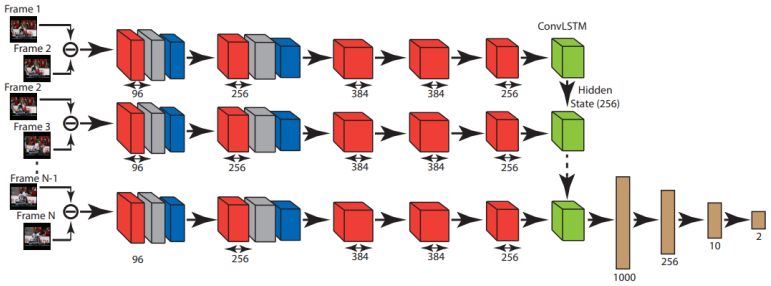



**Figure 2.19:** ConvLSTM architecture [62]

As described in Figure 2.19, to extract frame-level characteristics from a video, a CNN is applied. The frame-level characteristics are then accumulated using LSTM that uses a convolutional gate. The CNN, in combination with the ConvLSTM, can capture localized spatiotemporal characteristics, allowing for the analysis of local motion in the video. The paper also proposed feeding the model neighbouring frame differences as input, pushing it to encode the changes in the video. In terms of recognition accuracy, the presented feature extraction process is tested on three common benchmark datasets as 'Hockey Fight' [28], 'Movie Fight' [28], and 'Violent-Flows' [45]. Findings were compared to those produced using state-of-the-art methods. It was discovered that the suggested method had a promising capacity for identifying violent films prevailing state-of-the-art methods as three streams + LSTM, ViF, and ViF+OViF [62].

#### e, Mask RCNN + LSTM

To identify violent behaviours of a single person, an ensemble model of the Mask RCNN and LSTM was proposed [63]. Initially, human key points and masks were extracted, and then temporal information was captured. Experiments have been performed on datasets such as, 'Weizmann' [64], 'KTH' [65]. The results demonstrated that the proposed model outperforms individual models showing a violence detection accuracy rate of 93.4% in its best result. The proposed approach is more relevant to the industry, which is beneficial to society in terms of security.

#### f, Detecting human violent behaviour by integrating trajectory and deep CNN

Typical violence detection approaches depend on hand-crafted features, which may be insufficiently discriminative for the job of recognizing violent actions. Inspired by the performance of deep models for the recognition of human action, an innovative method for the detection of human violent behaviour by combining the trajectory and deep CNN is proposed that takes advantage of both handcrafted features and deep-learned features [66].

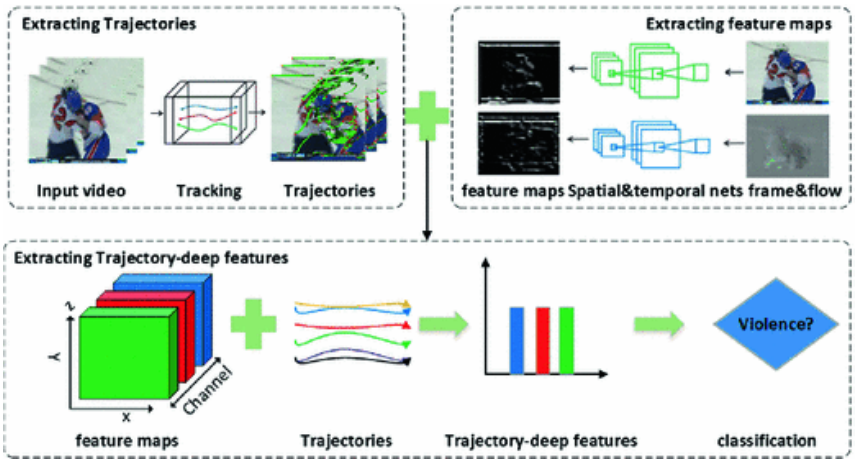



**Figure 2.20:** Integrating trajectory and deep CNN architecture [66]

To assess the proposed method, tests on two distinct violence datasets are performed: 'Hockey Fights' [28] and 'Crowd Violence' dataset. On these datasets, the findings show that the proposed approach outperforms state-of-the-art methods like HOG, HOF, ViF, and others.

### g, Violence detection using spatiotemporal features with 3D CNN

In smart cities, schools, hospitals, and other surveillance domains, an improved security system is required for the identification of violent or aberrant actions in order to prevent any casualties that may result in social, economic, or environmental harm. For this purpose, a three-staged end-to-end deep learning violence detection system is presented

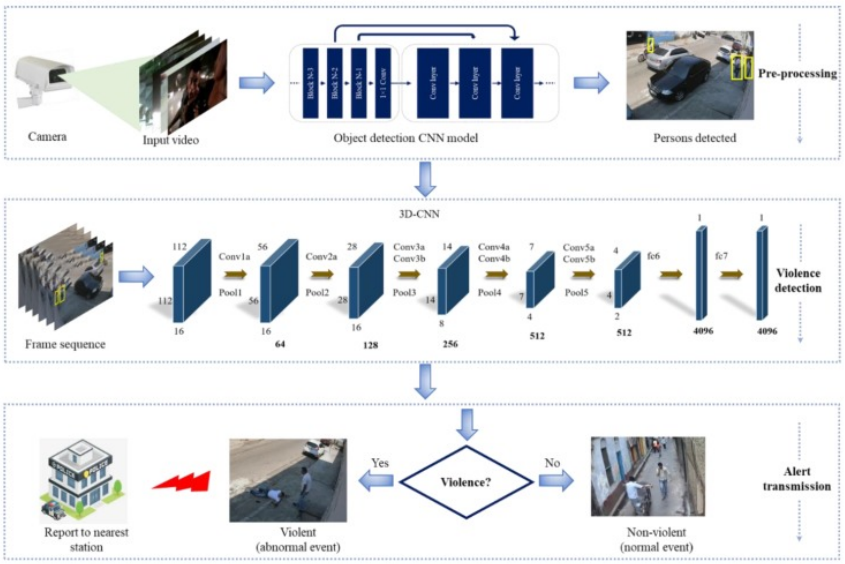



**Figure 2.21:** Spatiotemporal features with 3D CNN architecture [67]

As depicted in Figure 2.21, to minimize and overcome the excessive processing of useless frames, people are first identified in the surveillance video stream using

a lightweight CNN model. Secondly, a 16-frame sequence containing identified people is sent to 3D CNN, which extracts the spatiotemporal characteristics of the sequences and feeds them to the Softmax classifier. The authors also used open visual inference and neural networks optimization tools created by Intel to optimize the 3D CNN model, which transforms the training model into intermediate representation and modifies it for optimum execution at the end platform for the ultimate prediction of violent behaviour. When violent behaviour is detected, an alarm is sent to the closest police station or security agency so that immediate preventative measures may be taken. The datasets 'Violent Crowd [45], 'Hockey' [28], and 'Movies' [28] are used in the experiments. The experimental findings show that the proposed approach outperforms state-of-the-art algorithms such as ViF, AdaBoost, SVM, Hough Forest, and 2D CNN, sHOT, and others in terms of accuracy, precision, recall, and AUC.

## 2.3 Artificial Intelligence (AI)

Artificial Intelligence, Machine Learning, Deep Learning, Data Science are popular terms in this era. Knowing what they are and the differences between those areas is more crucial than ever.

Humans have long been obsessed with creating AI ever since the question, "Can machines think?", was posed by Alan Turing in 1950. AI enables the machine to think, that is without any human intervention the machine will be able to take its own decision. It is a broad area of computer science that makes machines seem like they have human intelligence. So, it's not only programming a computer to drive a car by obeying traffic signals but it's when that program also learns to exhibit the signs of human-like road rage [68]. Artificial Intelligence can be divided in to sub-fields based on the learning techniques they deploy as follows:

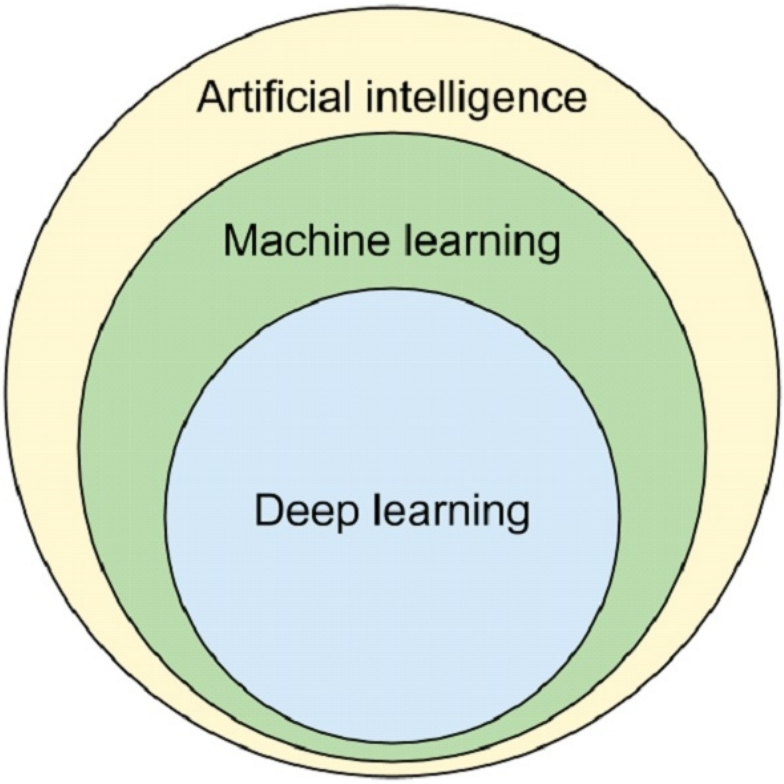



**Figure 2.22:** Sub-fields of Artificial Intelligence [68]

### 2.3.1 Machine Learning

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a subset of artificial intelligence. Machine learning algorithms build a model based on sample data, known as 'training data', in order to make predictions on unseen data or decisions without being explicitly programmed to do so [69]. The accuracy of such predictions can be improved by adding more data samples to the training set, scaling the data, Hyper-parameter-tuning, using dimensionality reduction methods, etc. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

At a broader level, machine learning can be classified into three types: Supervised learning, Unsupervised learning and Reinforcement learning.

#### a, Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labelled datasets to train algorithms that to classify data or predict outcomes accurately. As

input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross-validation process. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox [70]. Supervised learning can be separated into two types of problems when mining data: Classification and Regression.

Classification Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labelled or defined. Supervised learning models can be used to build and advance a number of business applications, such as, image- and object-recognition, customer sentiment analysis, spam detection, etc. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbours, and random forest, which are described in more detail below.

- K-Nearest Neighbours (K-NN):

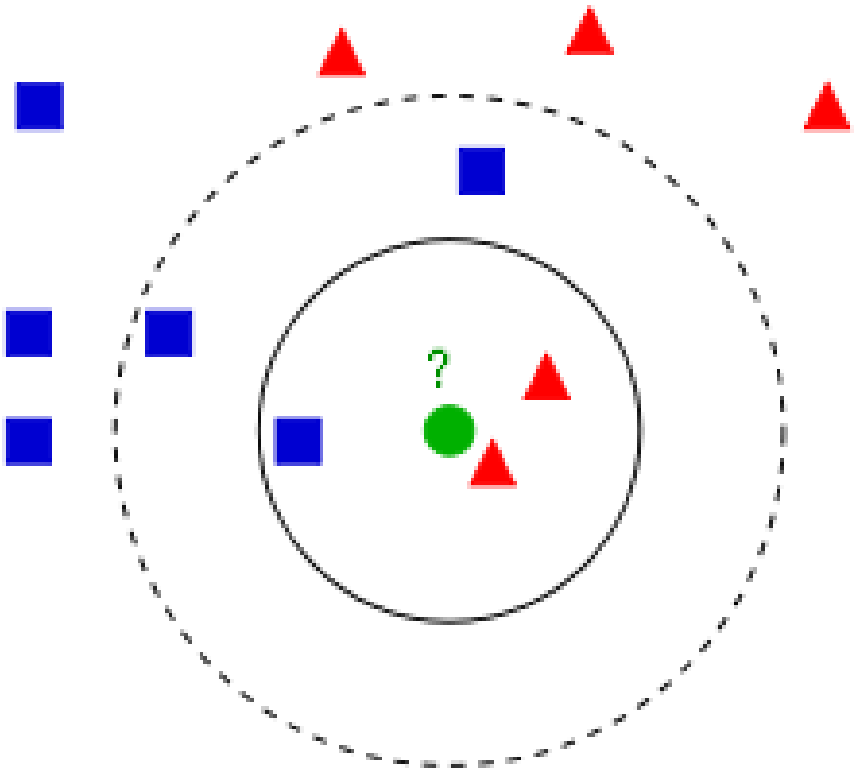



**Figure 2.23:** Example of k-NN classification [71]

K-NN algorithm is a non-parametric algorithm that classifies data points based on their proximity and association to other available data. This algorithm assumes that similar data points can be found near each other. As a result, it

seeks to calculate the distance between data points, usually through Euclidean distance, and then it assigns a category based on the most frequent category or average.

- Logistic regression:

  Logistic Regression is used when the dependent variable is categorical, meaning they have binary outputs, such as ”true” and ”false” or ”yes” and ”no.” The corresponding probability of the value labelled ”1” can vary between 0 (certainly the value ”0”) and 1 (certainly the value ”1”), hence the labelling; the function that converts log-odds to probability is the logistic function, hence the name. Spam identification is one real life application of logistic regression.

Regression

- Linear regression:

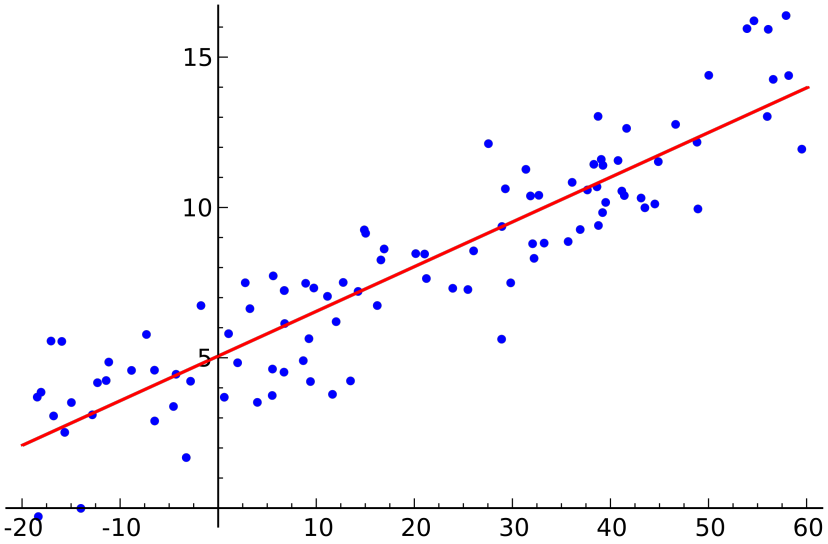



**Figure 2.24:** Illustration of Linear Regression model [72]

Linear Regression is used to identify the relationship between a dependent variable and one or more independent variables and is typically leveraged to make predictions about future outcomes. When there is only one independent variable and one dependent variable, it is known as simple linear regression. As the number of independent variables increases, it is referred to as multiple linear regression. For each type of linear regression, it seeks to plot a line of best fit, which is calculated through the method of least squares. However, unlike other regression models, this line is straight when plotted on a graph.

### b, Unsupervised Learning

Unsupervised Learning uses machine learning algorithms to analyse and cluster unlabelled data sets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition. Unsupervised learning models are used for three main tasks: clustering, association and dimensionality reduction.

Clustering

Clustering is a data mining technique for grouping unlabelled data based on their similarities or differences. In simple words, the aim is to segregate groups with similar traits and assign them into clusters. This technique is helpful for market segmentation, image compression, etc.

- K-means Clustering:

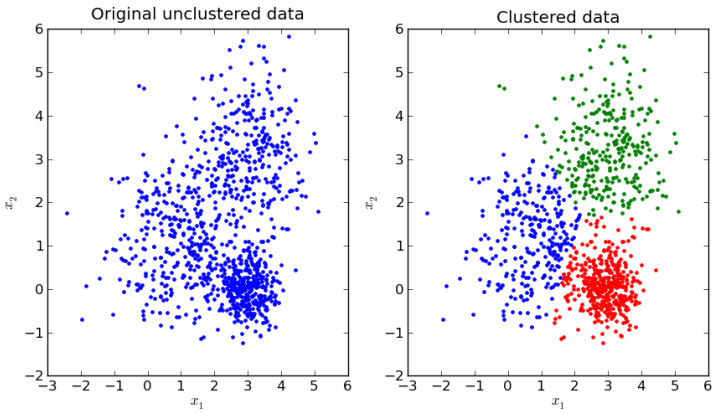



**Figure 2.25:** Original Vs. Clustered data [72]

K-means Clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centres or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

- Hierarchical Clustering:

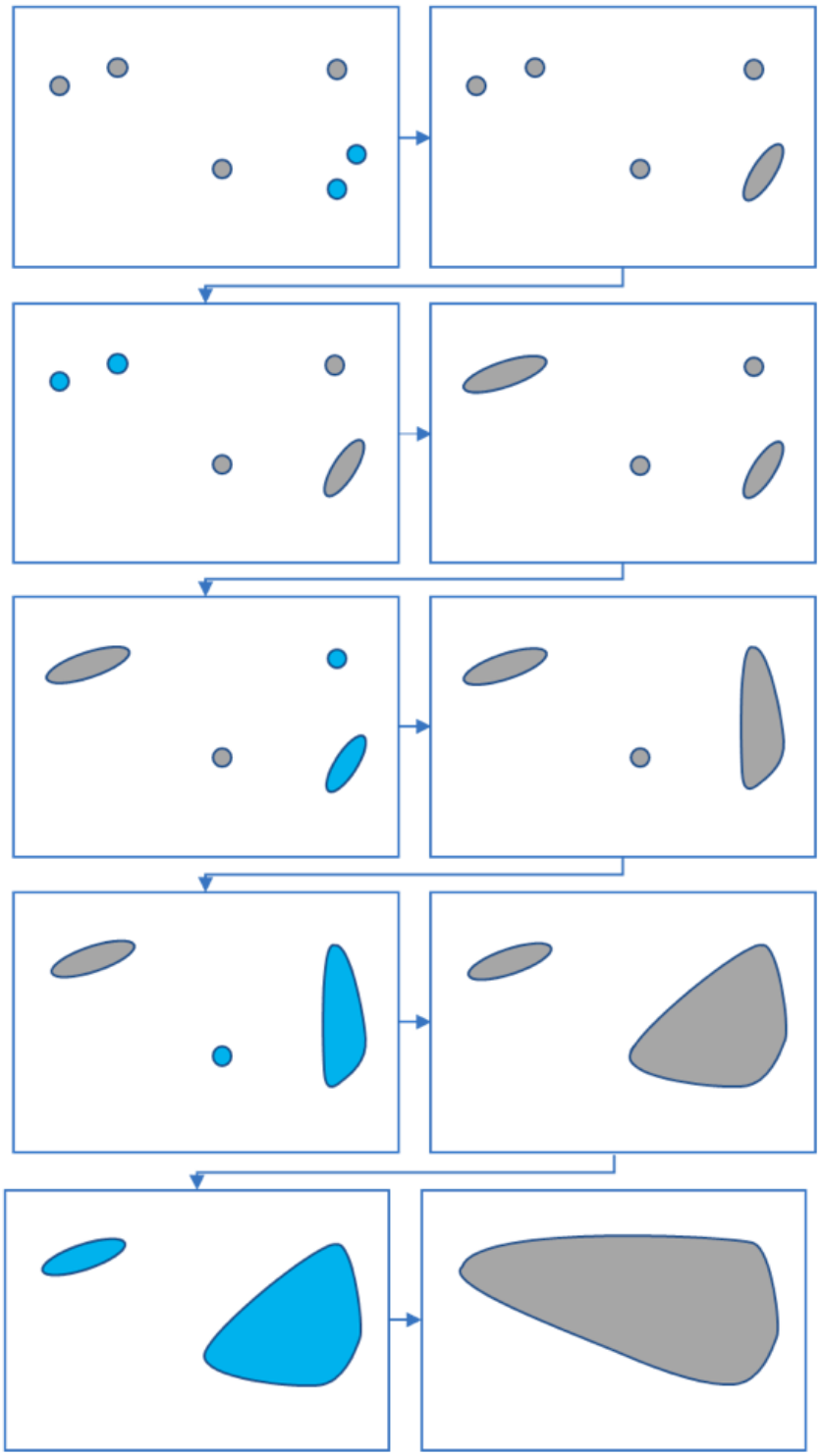


**Figure 2.26:** Hierarchical Clustering [72]

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

1. Identify the two closest clusters
2. Merge those two clusters

This iterative process continues until all the clusters are merged together.

### Association

Association Rule Learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. In any given transaction with a variety of items, association rules are meant to discover the rules that determine how or why certain items are connected based on a series of metrics such as, support and confidence.

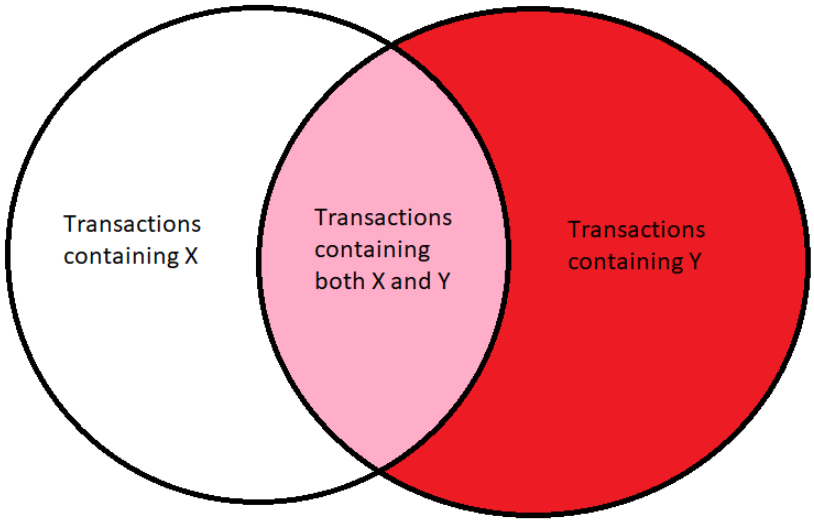



**Figure 2.27:** A Venn Diagram to show the associations between item sets X and Y of a dataset [73]

For example, the rule {onions, potatoes} =¿ {burger} found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, promotional pricing, product placements, etc [73].

### Dimensionality reduction

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables.

There are two components of dimensionality reduction: feature selection and feature extraction.

- Feature selection:

  Feature selection, also known as variable selection, is the process of selecting a subset of relevant features for use in model construction. Feature selection techniques are used for several reasons: simplification of models to make them easier to interpret by users, shorter training times, to avoid the curse of dimensionality, improve data's compatibility with a learning model class, etc. As a stand-alone task, feature selection can be unsupervised (e.g., Variance Thresholds) or supervised (e.g., Genetic Algorithms).

- Feature extraction:

  Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

  Note: feature selection keeps a subset of the original features while feature extraction creates new ones.

### c, Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Reinforcement learning differs from supervised learning in not needing labelled input/output pairs be presented, and in not needing sub-optimal actions to be explicitly corrected. Instead, the focus is on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).

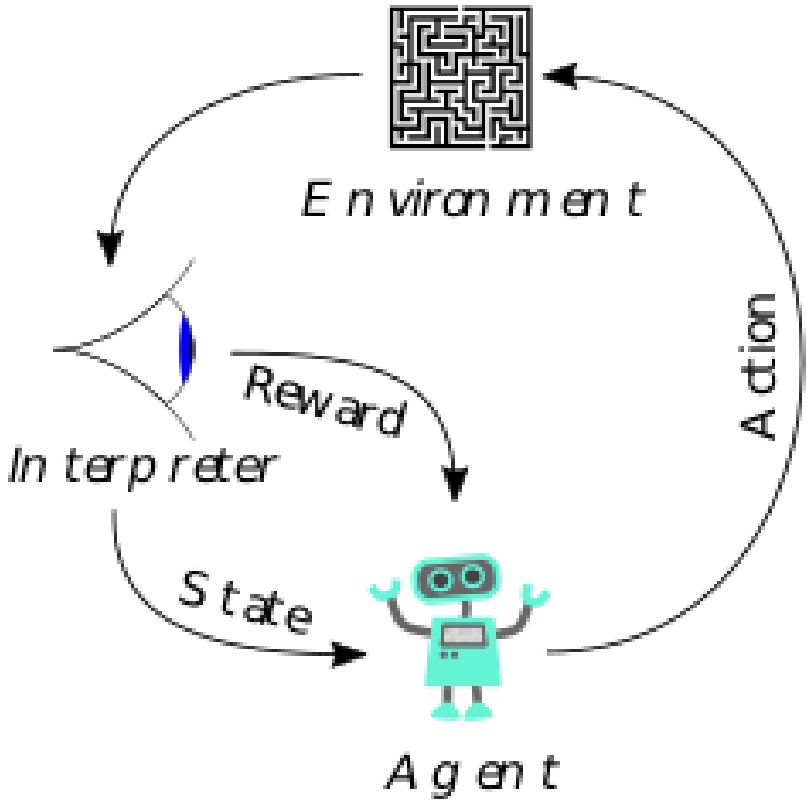



**Figure 2.28:** The typical framing of a Reinforcement Learning scenario [72]

Simply put, the purpose of reinforcement learning is for the agent to learn an optimal, or nearly-optimal, policy that maximizes the ”reward function” or other user-provided reinforcement signal that accumulates from the immediate rewards. This is similar to processes that appear to occur in animal psychology. For example, biological brains are hardwired to interpret signals such as pain and hunger as negative reinforcements, and interpret pleasure and food intake as positive reinforcements. In some circumstances, animals can learn to engage in behaviours that optimize these rewards. Due to its generality, reinforcement learning is studied in many disciplines, such as game theory, control theory, operations research, information theory, statistics, etc.

### 2.3.2 Deep Learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human

intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Note: Machine learning algorithms leverage structured, labelled data to make predictions—meaning that specific features are defined from the input data for the model and organized into tables. Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. In machine learning, this hierarchy of features is established manually by a human expert. Then, through the processes of gradient descent and back propagation, the deep learning algorithm adjusts and fits itself for accuracy, allowing it to make predictions with increased precision.

### a, Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs) are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

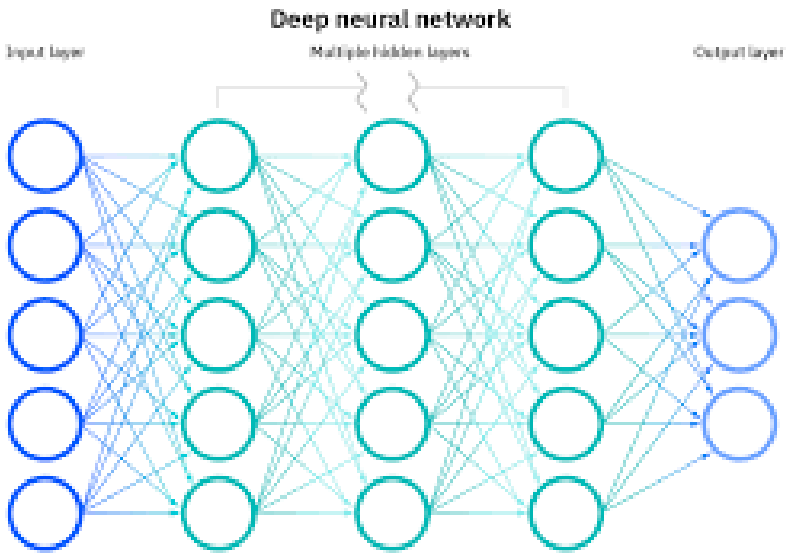



**Figure 2.29:** Deep Neural Network [74]

Artificial neural networks (ANNs) are comprised of a node layer, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is

activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google's search algorithm [9].

- Feed-forward Neural Networks:

  Feedforward Neural Networks, or Multi-layer Perceptron (MLPs) are comprised of an input layer, a hidden layer or layers, and an output layer. While these neural networks are also commonly referred to as MLPs, it's important to note that they are actually comprised of sigmoid neurons, not perceptrons, as most real-world problems are nonlinear. Data usually is fed into these models to train them, and they are the foundation for computer vision, natural language processing, and other neural networks.

- Convolutional neural networks (CNNs):

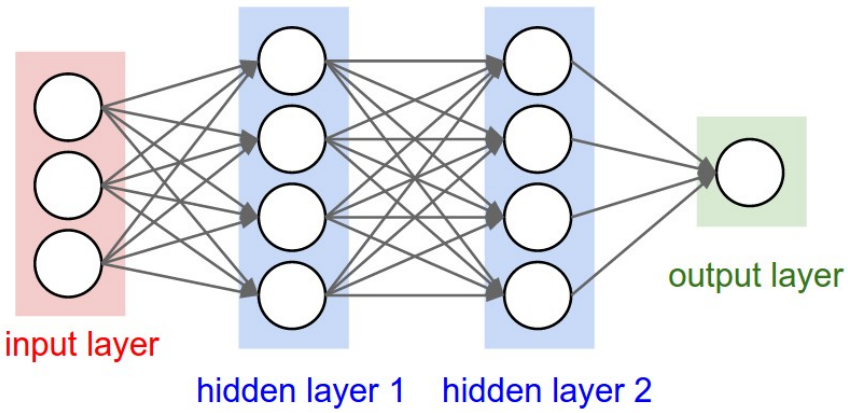



**Figure 2.30:** Regular 3-layer Neural Network [74]

  Convolutional neural networks (CNNs) are similar to feedforward networks, but they're usually utilized for image recognition, pattern recognition, and/or computer vision. In other words, ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the number of parameters in the network. [3] These networks harness principles from linear algebra, particularly

matrix multiplication, to identify patterns within an image.

Convolutional Neural Networks take advantage of the fact that the input consists of images and they constrain the architecture in a more sensible way. In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height and depth. (Note that the word depth here refers to the third dimension of an activation volume, not to the depth of a full Neural Network, which can refer to the total number of layers in a network.)

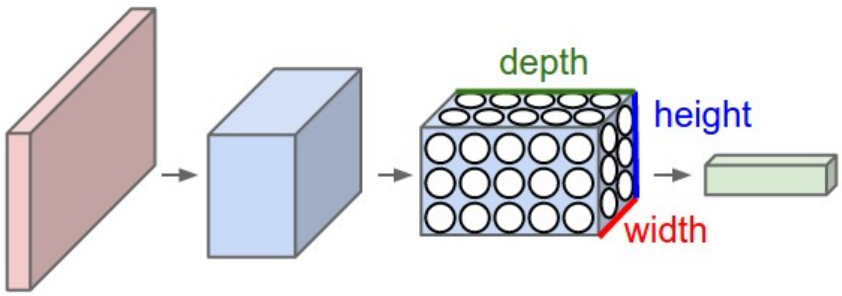



**Figure 2.31:** A ConvNet arranges its neurons in three dimensions: width, height and depth [74]

A simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. Three main types of layers are used to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer (exactly as seen in regular Neural Networks). We will stack these layers to form a full ConvNet architecture.

- Recurrent neural networks (RNNs):

  Recurrent neural networks (RNNs) are identified by their feedback loops. These learning algorithms are primarily leveraged when using time-series data to make predictions about future outcomes, such as stock market predictions or sales forecasting.

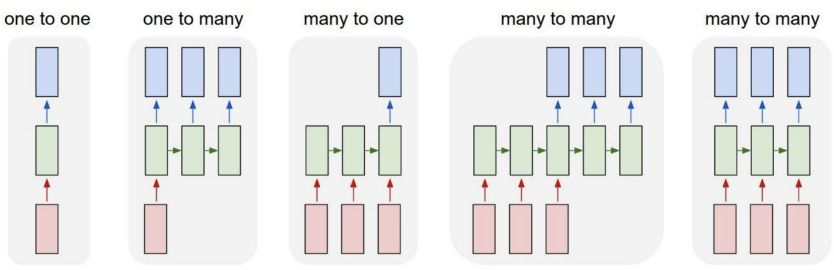



**Figure 2.32:** Different (non-exhaustive) types of Recurrent Neural Network architectures [74]

The main difference between a CNN and an RNN is the ability to process temporal information — data that comes in sequences, such as a sentence. Recurrent neural networks are designed for this very purpose, while convolutional neural networks are incapable of effectively interpreting temporal information. As a result, CNNs and RNNs are used for completely distinct purposes, and there are differences in the structures of the neural networks themselves to fit those different use cases. In general, RNNs allow us to wire up an architecture, where the prediction at every single timestep is a function of all the timesteps that have come before.

### b, Optimizing Neural Networks

- Cost function:

  The cost function is the technique of evaluating "the performance of our algorithm/model". It takes both predicted outputs by the model and actual outputs and calculates how much wrong the model was in its prediction. It outputs a higher number if our predictions differ a lot from the actual values. As we tune our model to improve the predictions, the cost function acts as an indicator of how the model has improved. This is essentially an optimization problem. The optimization strategies always aim at "minimizing the cost function". There are many cost functions in machine learning and each has its use cases depending on whether it is a regression problem (e.g.: Mean Squared Error, Mean Absolute Error, etc) or classification problem (e.g.: Cross-Entropy).

- Gradient Descent:

  Gradient descent (GD) is an iterative first-order optimisation algorithm used to find a local minimum or a local maximum of a given function. This method is commonly used in machine learning (ML) and deep learning (DL) to minimise a cost function (e.g., in a linear regression). However, its use is not limited to Machine Learning or Deep Learning only, it's being widely used also in areas like, control engineering (robotics, chemical, etc.), computer games and mechanical engineering.[4]

  For a Gradient descent algorithm to work properly, a function should be differentiable (i.e.: has a derivative for each point in it's domain) and convex (i.e.: the line segment between any two points on the graph of the function does not lie below the graph between the two points).

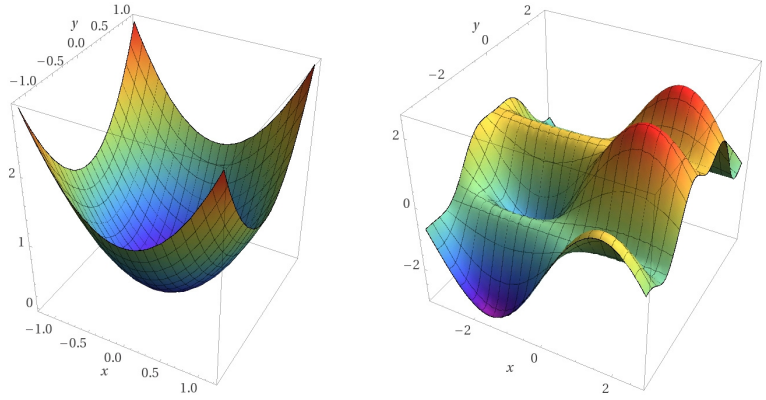


**Figure 2.33:** Convex Vs. non-convex function [74]

Gradient Descent Algorithm repetitively calculates the next point using gradient at the current position, then scales it by a user-defined learning rate () and subtracts obtained value from the current position. Simply put, the algorithm 'takes a step' towards the lowest point of the graph, which in this case is the smallest cost function, for each and every iteration until convergence. This process can be expressed as:

$$X_i = X_i - \alpha \frac{d}{dx} f(x) \tag{2.1}$$

It should be noted that the learning rate () has a strong influence on performance of the algorithm. If the learning rate is too small, it takes longer for Gradient Descent to converge, hence could be computationally expensive. Further, the algorithm may reach maximum iteration before reaching the optimum point. On the other hand, if the learning rate is too large, the algorithm may not converge to the optimal point (jump around) or even to diverge completely.

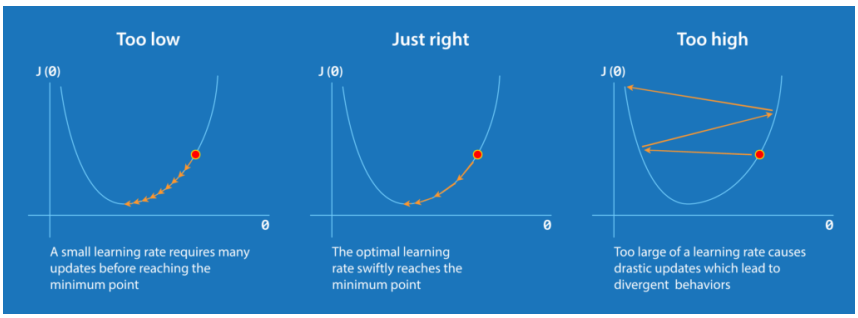



**Figure 2.34:** The effect of the learning rate [74]

### c, Forward and Back Propagation

- Forward propagation:

Forward propagation (or forward pass), which is the first step of training a neural network, refers to the calculation and storage of intermediate variables (including outputs) for a neural network in order from the input layer to the output layer. [5] To further understand the concept of 'Forward Propagation', a thorough understanding of the concept of 'Activation Function' is required.

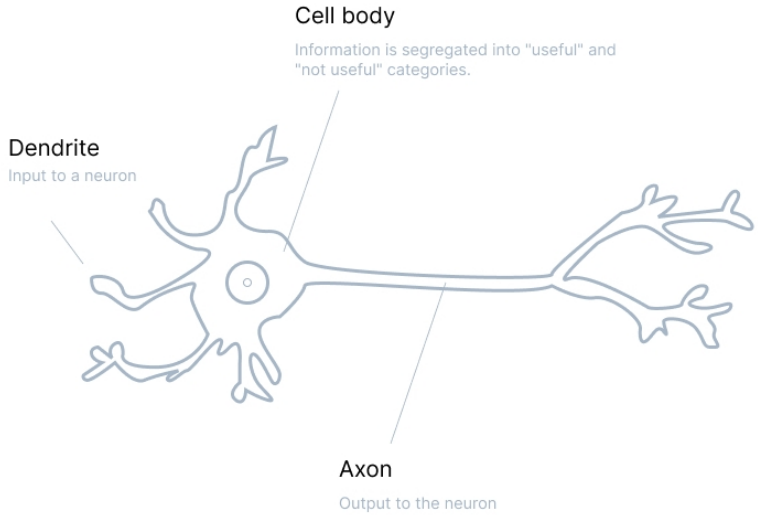



**Figure 2.35:** A Neuron [74]

In a Neural Network, the node is the replica of an actual Neuron. A Neuron receives a set of input signals, which are called 'External Stimuli'. Depending on the nature and intensity of these input signals, the brain processes them and decides whether the neuron should be activated ("fired") or not. Similarly, in Artificial Intelligence, an Activation Function will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations. In other words, Activation Function helps the neural network to use important information while suppressing irrelevant data points.

In the Forward Propagation, the Activation Function is a mathematical "gate" in between the input feeding the current neuron and its output going to the next layer [10].In simpler terms, Forward Propagation follows 3 main steps:

1. Calculating the weighted input to the hidden layer.
2. Applying the activation function and pass the result to the final layer.
3. Repeating step 2, except this time the input wight is replaced by the hidden layer's output.

- Back propagation:

Back propagation is the method of fine-tuning the weights of a neural network based on the 'error rate' obtained, as a result of the forward pass, in the previous epoch. Simply put, at the output layer, the forward pass will make a prediction, which might not be the correct prediction, hence giving rise to an 'error' (i.e.: the difference between the prediction and the ground-truth). Then, the backward pass will travel back from the output layer to the hidden layers to adjust the weights in a way that the 'error' is reduced. This process, a forward pass followed by a backward pass, will be repeated until the neural network is optimized or stuck in a local optimum. In summary, proper tuning of the weights reduces error rates and make the model reliable by increasing its generalization.

### 2.3.3 Motion estimation methods

Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may relate to the whole image (global motion estimation) or specific parts, such as rectangular blocks, arbitrary shaped patches or even per pixel.

In order to perform motion estimation, the 'correspondence' between the considered frames must be taken in to account. In the domain of motion estimation, the 'correspondence problem' refers to the problem of ascertaining which parts of one image correspond to which parts of another image, where differences are due to movement of the camera, the elapse of time, and/or movement of objects in the photos.

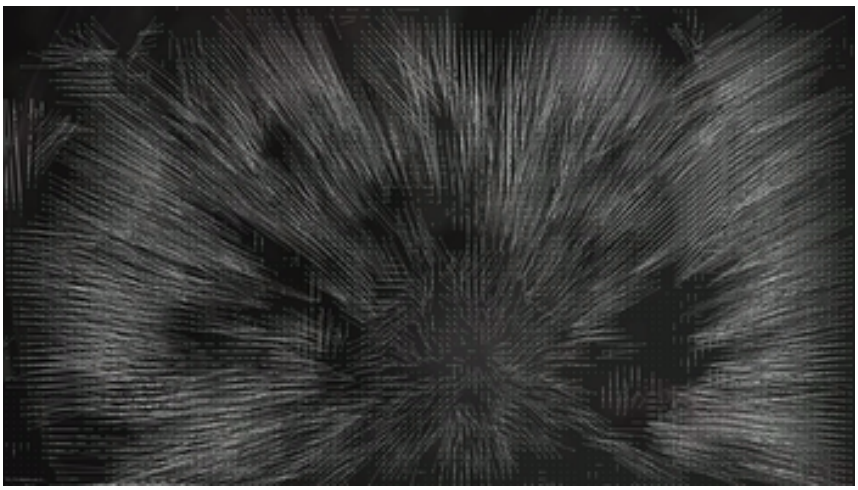


**Figure 2.36:** A visualization of the motion estimation performed in order to compress an MPEG movie [75]

Through the ‘correspondence problem’ the ‘motion vector’, which is the key element of motion estimation, can be defined. The methods for finding motion vectors can be categorised into pixel-based methods (”direct”) and feature based methods (”indirect”).

### a, Direct methods

Block-matching algorithm, Pixel recursive algorithms and Optical flow are a few examples for direct methods of motion estimation.

- Optical flow:

  Optical flow is the motion of objects between consecutive frames of sequence, caused by the relative movement between the object and camera. Based on the pixels that’s being processed, there are two main methods: Sparse and Dense Optical flow.

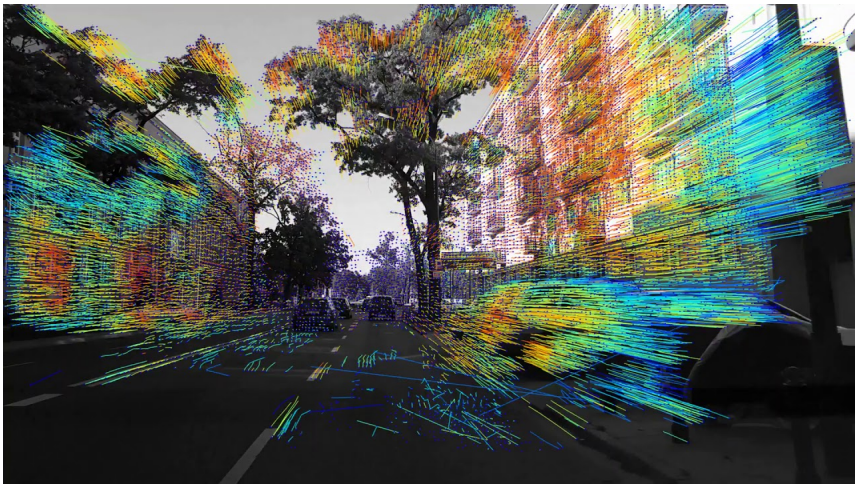


**Figure 2.37:** Sparse Optical flow: considers the flow vectors of some ”interesting features” (e.g.: few pixels depicting the edges or corners of an object) within the frame [76]

**Figure 2.38:** Dense Optical flow: considers the flow vectors of the entire frame (all pixels) - up to one flow vector per pixel [76]

### b, Indirect methods

Indirect methods use features, such as corner detection, and match corresponding features between frames, usually with a statistical function applied over a local or global area. The purpose of the statistical function is to remove matches that do not correspond to the actual motion.

# CHAPTER 3. PROPOSED METHOD

In the previous chapter, state-of-the-art approaches for violence detection are described. In this chapter, the selected benchmark dataset (called AICS - violence dataset [11]) and a novel method for violence detection will be discussed.

## 3.1 AICS - violence dataset

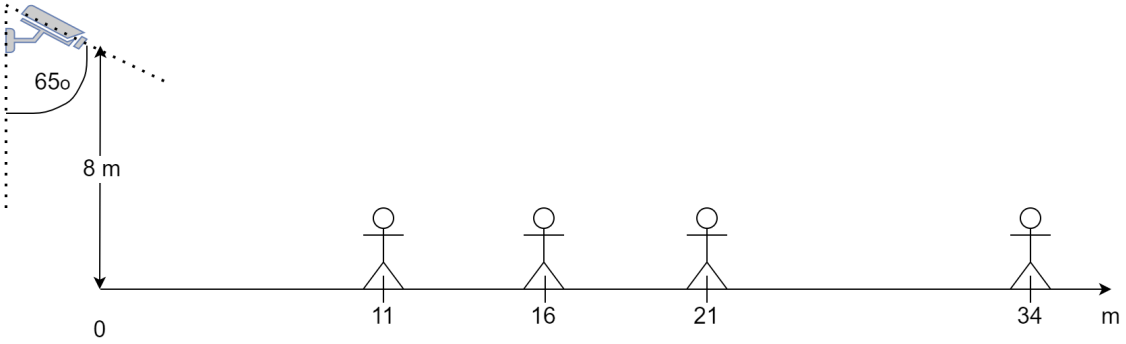



**Figure 3.1:** Experimental setup of the first camera [11]

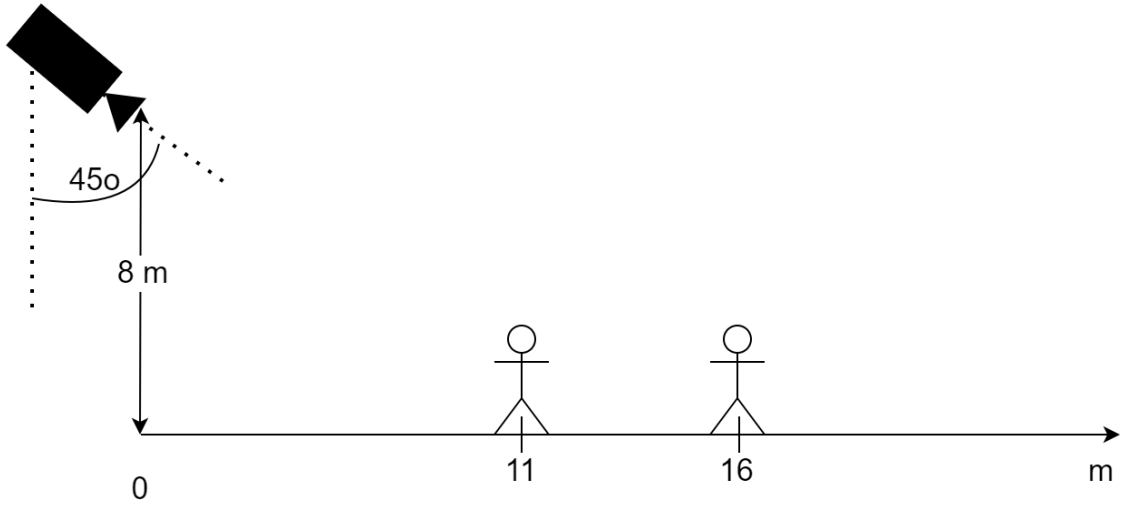



**Figure 3.2:** Experimental setup of the second camera [11]

Figure 3.1 and 3.2 illustrates the camera setup for collecting the AICS - violence dataset. Both cameras are placed 8 meters in height from the ground. The angle between the camera and vertical axis of the first camera (Cam1) is 65 degrees, meanwhile, 45 degrees is selected for the second camera (Cam2). In terms of resolution, Cam1 and Cam2 are 1920x1080 and 2048x1536 respectively. Additionally, each camera is used for capturing different distances human groups as depicted in the horizontal axis, for example, Cam1 recorded a group of people at distances of 11, 16, 21, and 34 meters consecutively.

In the previous version of the AICS - violence dataset [11], the size of each test set is only 6.25% of the training set. Therefore, we collected more samples to ensure the ratio between each test set and training set is 1/9. The latest version of the AICS-Violence dataset contains 7576 video clips collected by two input cameras. The first camera (Cam1) is used to build the training and test data sets while the second camera (Cam2) is only used to evaluate the algorithm. The dataset is labeled into two classes namely violent and non-violent, and divided into 3 subsets:

1. Training set consists of 3100 samples of each class collected by Cam1.
2. Cam1 test set consists of 344 samples of each class collected by Cam1.
3. Cam2 test set consists of 344 samples of each class collected by Cam2.

## 3.2 Proposed Violence Detection Method

The abstract architecture of our proposed violence detection method is illustrated in Figure 3.4. It consists of three steps: human detection, candidate box extraction, and feature extraction and classification.

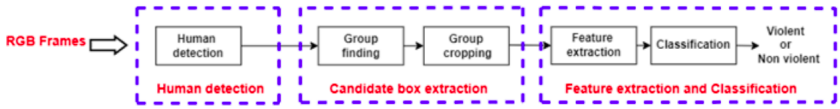



**Figure 3.3:** Abstract architecture of our proposed violence detection method

Our method receives an RGB frame sequence as the input, which is captured from our surveillance cameras. Each time, the human detection module processes a single frame. Frame without detected human would be ignored. Otherwise, the candidate box extraction component will determine the location and distances between the human bounding boxes on each frame. Subsequently, the regions that include multiple close people (called candidate boxes) are identified, cropped, and resized. The candidate boxes are applied to the next 15 consecutive frames. Each cropped frame sequence (including the cropped areas of the 16 consecutive frames) will be fed into the next block for feature extraction using our proposed method. Finally, the extracted features are classified using a fully connected layer to decide whether the group acts are violent or not.

### a, Human detection

In this thesis's topic, the surveillance cameras capture a wide outdoor area as input to our method. Because the camera is placed at a high place (8 meters) compared to the ground level, the region of interest (human group) in a frame

would be considerred small despite having full HD resolution. For example, at a distance of 16 meters, the area that includes a violent group of three people occupies only approximately 0.8% of the entire input frame. Consequently, using the original frames serves directly as input for the feature extraction and classification stage could lead to background noise and cause poor performance. Additionally, in order to be fed into a 3D CNN, each frame has to be resized resulting in a very small number of pixels for human groups which would cause low accuracy. Therefore, human detection and region of interest selection in the frames are crucial. YOLOv4 [77] model is chosen as a human detector since it is capable of balance between speed and accuracy as well as being suitable for detecting small objects at long distances. It receives the frame sequences as input and returns information about the position and size of the human bounding boxes on those frames.

### b, Candidate box extraction

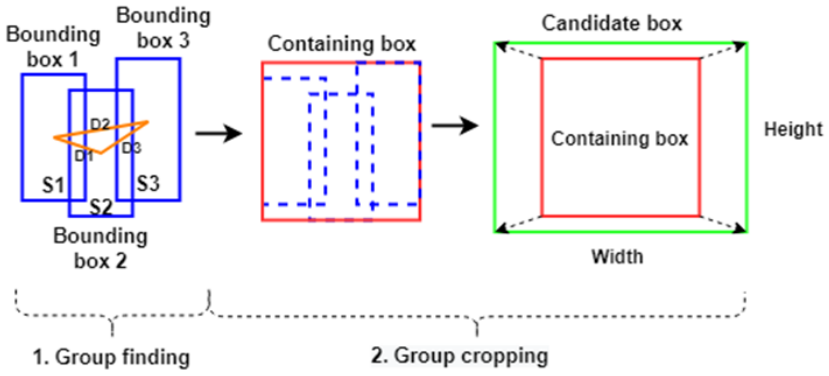



**Figure 3.4:** Steps of candidate box estraction

We developed this step based on Human Area Detection algorithm [11]. After detecting bounding boxes of humans, the goal is to determine and crop areas (called candidate boxes) containing groups of humans. There are two sub-steps in this stage namely group finding and group cropping as depicted in Figure **??**.

The target of the first sub-step is to determine groups of close people because these are the only areas where violent acts couple potentially happen. Firstly, the Euclidean distance between each pair of human bounding box centroids is calculated. Afterward, people are considered to be in a candidate group if the distance is smaller than a threshold (dynamically changed based on the size of human bounding boxes).

The main goal of the consecutive step is to crop input frames into groups of humans which are called candidate boxes (each frame could contain none, one, or multiple groups). A containing box is determined based on the outermost edges

of bounding boxes of each candidate group. After that, the containing box is expanded to generate the corresponding candidate box using (3.1) and (3.2) with S - average human bounding box area - is calculated using (3.3) and constant k (ratio of average human bounding box area in a group) is 20% based on our survey. However, if the width and height of candidate boxes do not ensure longer than one of containing boxes, the width and height of containing boxes will be selected to not ignore important information about humans.

$$Height = \sqrt{\frac{3}{4} \times \frac{S}{k}} \tag{3.1}$$

$$Width = \sqrt{\frac{4}{3} \times Height} \tag{3.2}$$

$$S = \frac{\Sigma_{i=1}^{m} S_i}{m} \tag{3.3}$$

Similarly, candidate boxes are generated and resized to 320x240 for the next 15 consecutive frames. The feature-extraction-and-classification stage uses every 16 consecutive frames to determine whether a group is violent or not.

**c, Feature extraction and Classification**

In general, among non-fusion deep learning algorithms experimented on the AICS-Violence dataset, 3D DenseNet Lean [78] method achieved the highest accuracy on the Cam1 test set, on the other hand, its performance on the Cam2 test set witnessed a significant decrease as depicted in details at section **??**. Furthermore, our preliminary experiments illustrated that using visualized optical flow as input for 3D DenseNet Lean [78] yielded remarkable improvement (from 82.03 to 93.7% of accuracy) for the Cam2 test set meanwhile only facing a slight decline in performance on the Cam1 test set. Therefore, we proposed a novel method to combine the strengths of visualized optical flow and 3D DenseNet Lean [78] in order to perform better on both test sets. Figure 3.5 shows the architecture of our proposed method.

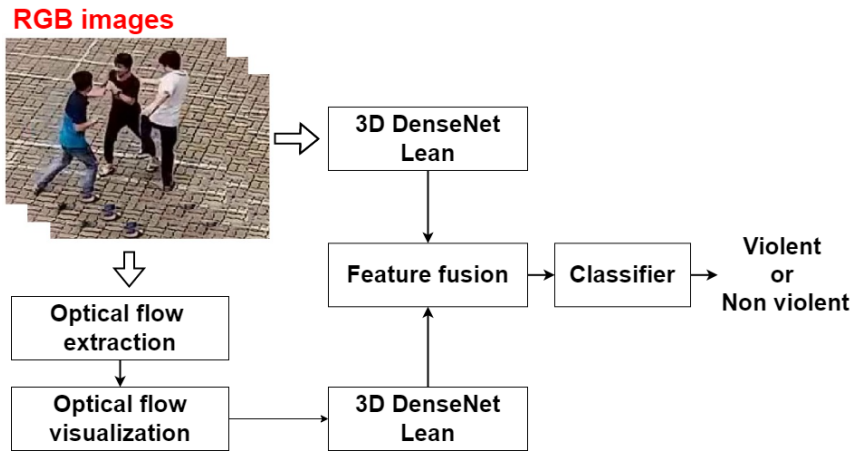



**Figure 3.5:** Architecture of our proposed method

In our proposed method, two different 3D DenseNet Lean [78]'s feature extractor streams are utilized to generate features, $R = [r_1 r_2 ... r_n]^T$ and $O = [o_1 o_2 ... o_n]^T$ from RGB and visualized optical flow frame sequences respectively. Consecutively, both feature vectors are fused to create global features $F_b = [f_{b1} f_{b2} ... f_{bn}]^T$. Finally, a fully-connected layer will be used to classify the fused features.

Particularly, at the fusion step, the feature vector $O = [o_1 o_2 ... o_n]^T$, extracted by 3D DenseNet Lean [78] on the visualized optical flow, is then normalized to $L = [l_1 l_2 ... l_n]^T$ using (3.4). After that, element-wise multiplication is applied to $L$ and $R$ as in (3.5) to augment features extracted from moving areas.

$$L = \frac{O}{max(O)} = [\frac{O_1}{max(O)} \frac{O_2}{max(O)} ... \frac{O_n}{max(O)}]^T \tag{3.4}$$

$$F_b = [f_{b1} f_{b2} ... f_{bn}]^T = O.R = [o_1.r_1 o_2.r_2 ... o_n.r_n]^T \tag{3.5}$$

# CHAPTER 4. EXPERIMENTAL RESULTS

## 4.1 Evaluation metrics

| | | *True class* | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| *Predicted class* | **Positive** | TP | FP |
| | **Negative** | FN | TN |

**Table 4.1:** Confusion matrix

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.1}$$

Confusion matrix and accuracy (ACC), described at Table 4.1 and (4.1) respectively, are used to evaluate performances of SOTA and our proposed method.

Confusion matrix provides information of predicted classes of a classification model and corresponding actual labels including: true positive (TP) - the correct prediction number that the violent video is predicted as violence, false positive (FP) - the incorrect prediction number that the non-violent video is predicted as violence, true negative (TN) - the correct prediction number that the non-violent video is predicted as non-violence, and false negative (FN) - the incorrect prediction number that the violent video is predicted as non-violence.

Accuracy, computed as ratio between the number of correct predictions (TP and TN) and all predictions (TP, TN, FP, and FN), describes generally the model performance across all classes.

## 4.2 Experimental method

### 4.2.1 Evaluation on AICS - violence dataset

To create a baseline for the AICS - violence dataset, we have evaluated the accuracies of some SOTA methods in action recognition in general and violence detection in particular namely Convolutional 3D (C3D) [79], Convolutional Long Short Term Memory (ConvLSTM) [80], 3D DenseNet [78], and Two-Stream Network for Violence Detection Using Separable Convolutional LSTM (TSSCL) [81].

As described in section 3.1, the AICS - violence dataset contains a training set and 2 test sets. The ratio of each test set to the training set is 1:9. Furthermore, during training stage, 20% of the training set is split as a validation set which is used for tuning hyper-parameters (such as epoch, batch size, learning rate,...)

We selected fine-tuning through the full network as our transfer learning strategy

for all baseline methods. All baseline SOTA are pre-trained on standard benchmark datasets including Kinetics [82], ImageNet [83], and RWF-2000 [84]. The three mentioned well-known datasets consist of about 400, 500, and 1000 samples per class respectively. However, only Kinetics [82] and RWF-2000 [84]'s contents, which are human action and violence on surveillance cameras, are similar to the AICS - violence dataset. Therefore, compared to those datasets, the AICS - violence dataset, which consists of more than 3000 images per class, could be considered large. In conclusion, we could fine-tune the entire network with significant concern about overfitting.

For each method, we trained and evaluated once on the AICS - violence dataset. Besides, as depicted in Table 4.2, hyper-parameters (such as batch size, epoch, and learning rate) are selected to balance the capacity of our GPU (depicted in 4.2.3 and the convergence of models.

| Method | Learning rate | Batch size | Epoch |
|---|---|---|---|
| C3D [79] | $10^{-4}$ | 8 | 100 |
| ConvLSTM [80] | $10^{-3}$ | 32 | 100 |
| 3D DenseNet [78] | $10^{-3}$ | 32 | 100 |
| 3D DenseNet Lean [78] | $10^{-3}$ | 32 | 100 |
| TSSCL-A [81] | $5^{-5}$ | 4 | 50 |
| TSSCL-C [81] | $5^{-5}$ | 4 | 50 |
| TSSCL-M [81] | $5^{-5}$ | 4 | 50 |

**Table 4.2:** Selected hyper-parameters for baseline methods

### 4.2.2 Evaluation on standard benchmark dataset

We selected Hockey Fights [28], a well-known benchmark dataset, to further evaluate our proposed model. The dataset consists of fighting clips from Hockey matches that are not what our proposed method is meant to be trained on. However, we want to test our model's generalizability in different scenarios.

As mentioned in section **??**, our proposed method is developed from 3D DenseNet Lean [78] Therefore, in this experiment, we compare these 2 approaches on the Hockey Fights dataset [28].

Hockey Fights Dataset [28] consists of 1000 clips including fighting and normal plays captured in hockey matches. We split the dataset into training and test sets containing 800 and 200 samples respectively.

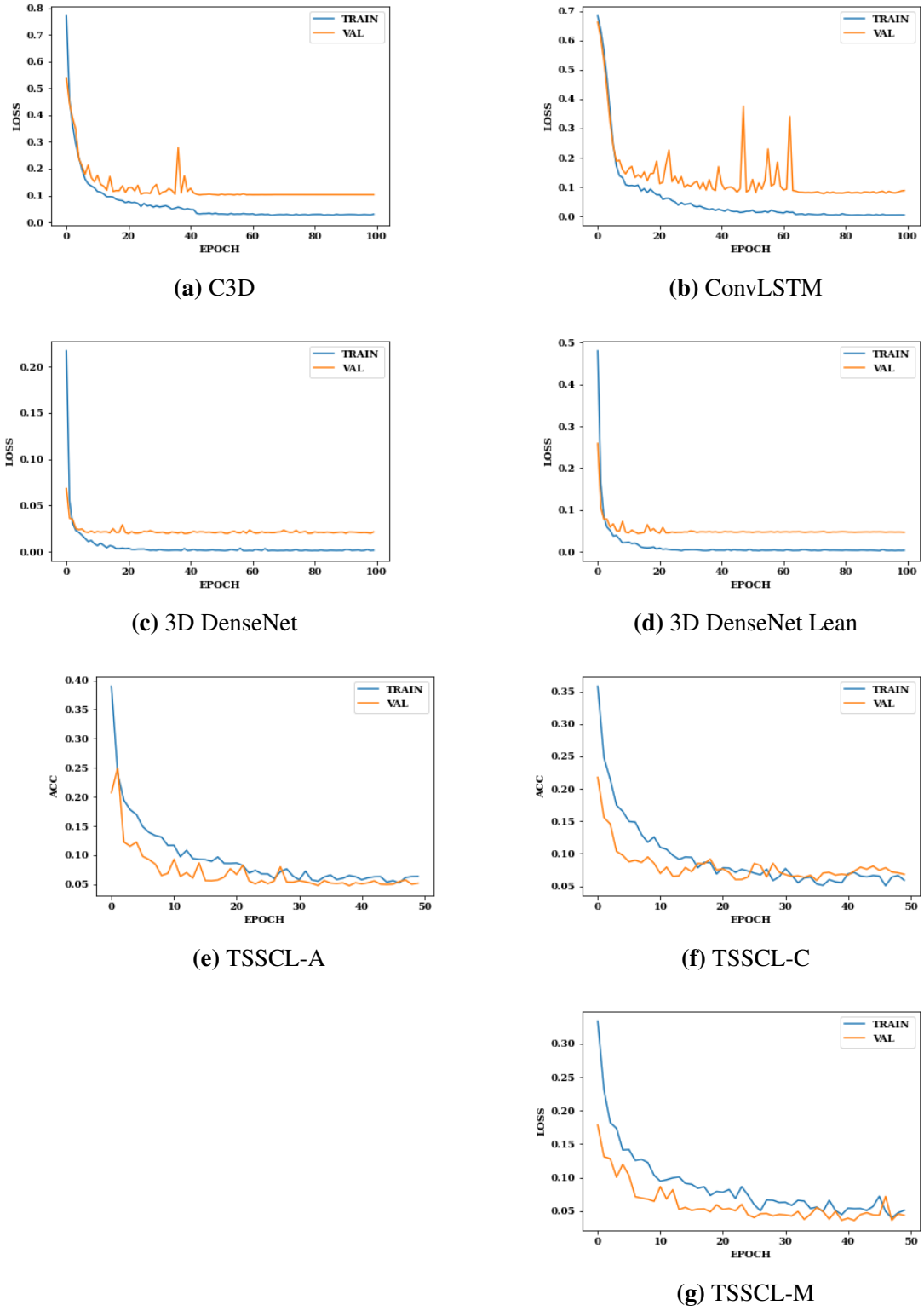
Finally, epoch, batch size, and learning rate are 150, 32, and $10^{-3}$ respectively for all models.

### 4.2.3 Infrastructures and frameworks

To conduct all mentioned experiments, we used HP Workstation Z640 having GeForce GTX 1080 Ti GPU with 12GB VRAM. Additionally, famous deep learning frameworks such as PyTorch [85], and TensorFlow [86] are utilized for implementing algorithms.

## 4.3 Results on AICS - violence dataset

### 4.3.1 Results of baseline methods on AICS - violence dataset




**(a)** C3D

**(b)** ConvLSTM

**(c)** 3D DenseNet

**(d)** 3D DenseNet Lean

**(e)** TSSCL-A

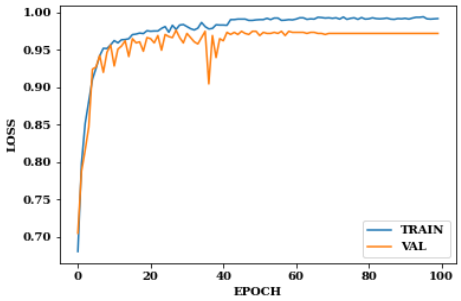
**(f)** TSSCL-C

**(g)** TSSCL-M

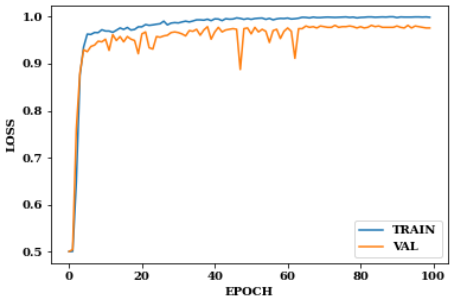
**Figure 4.1:** Training and validation losses of baseline methods on AICS - violence dataset

Figure 4.1 illustrated changes in training and validation losses over epochs for each selected baseline method. In general, with the selected hyper-parameters

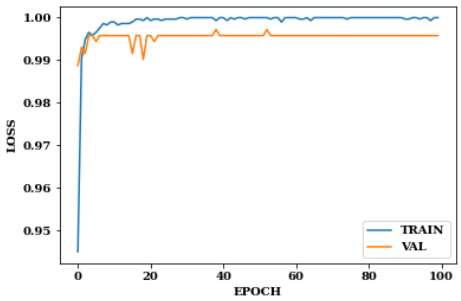
depicted in 4.2, all baseline methods converged. However, 3D DenseNet [78] and 3D DenseNet Lean [78] optimized losses at a significantly faster speed (converged after around 30 epochs) compared to the remaining algorithms (achieved best results after about 40 to 70 epochs).



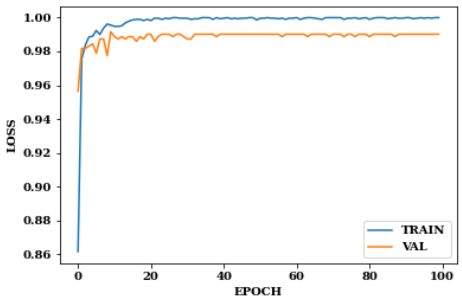

**(a)** C3D



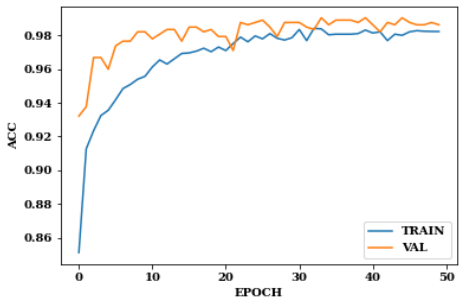

**(b)** ConvLSTM



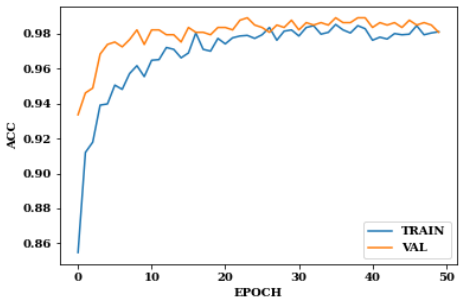

**(c)** 3D DenseNet



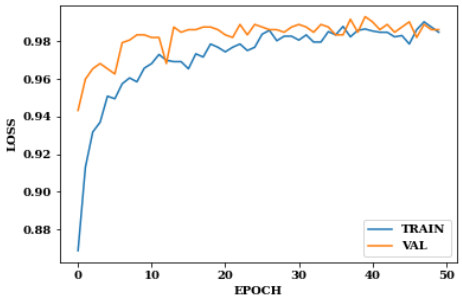

**(d)** 3D DenseNet Lean




**(e)** TSSCL-A




**(f)** TSSCL-C




**(g)** TSSCL-M

**Figure 4.2:** Training and validation accuracies of baseline methods on AICS - violence dataset

Figure 4.2 depicts changes in training and validation accuracy over epochs for baseline methods. Generally, among all experimented methods, only three TSSCL variants [81] do not have overfitting. However, the 4 remaining methods only face an insignificant of that problem. Particularly, the difference between the best validation and corresponding training accuracies of C3D [79], ConvLSTM [80], 3D DenseNet [78], and 3D DenseNet Lean [78] are 1.8, 1.82, 0.3, and 0.5

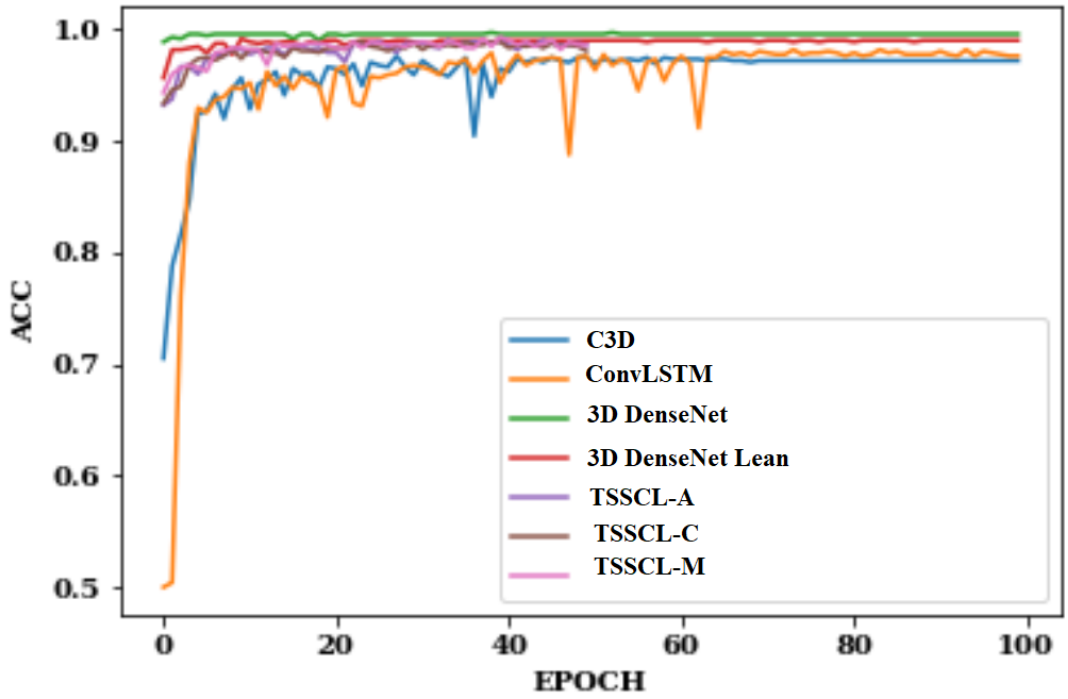



**Figure 4.3:** Comparison of validation accuracy over epoch of baseline methods on AICS - violence dataset

Figure 4.3 illustrates the changes in validation accuracies over epochs for all selected baseline methods. Generally, all baseline methods have robust performance (more than 95%) on the validation set of the AICS - violence dataset. Among them, 3D DenseNet [78] achieves the highest performance followed by its lightweight version - 3D DenseNet Lean [78], and 3 variants of TSSCL [81]. Finally, the worst performances belong to ConvLSTM [80] and C3D [79] consecutively.

### 4.3.2 Comparison of our proposed and baseline methods on AICS - violence testsets

| Method | Accuracy on test sets (%) | |
|---|---|---|
| | **Cam1** | **Cam2** |
| C3D [79] | 76.25 | 67.25 |
| ConvLSTM [80] | 91.75 | 90.25 |
| 3D DenseNet [78] | 95.2 | 81.98 |
| 3D DenseNet Lean [78] | 96.55 | 82.03 |
| TSSCL-A [81] | 95 | **95** |
| TSSCL-C [81] | 92.5 | 92.75 |
| TSSCL-M [81] | 96.25 | 94.25 |
| 3D DenseNet Fusion OF RGB [11] | 97.33 | 92.55 |
| **Our proposed method** | **97.675** | 93.55 |

**Table 4.3:** Comparison of baseline and our proposed methods on AICS - violence test sets.

Table 4.3 depicts the results of baseline and our proposed methods on the AICS - violence dataset. Generally, all accuracies on the Cam1 test set are better than one on the more challenging test set Cam2 which has a different angle than the training set. C3D [79], 3D DenseNet [78], and 3D DenseNet Lean [78] witnessed significant declines in accuracies (9%, 13.22%, and 14.52% respectively) between the two test sets. Meanwhile, ConvLSTM [80] and TSSCL [81]'s performances are more stable. Except for C3D [79], the remaining methods all predict correctly more than 90% of Cam1 test sets. Furthermore, TSSCL-A [81] and TSSCL-M [81] achieve high accuracies on both cameras (More than 94.25%). Among all experimented methods, our proposed method achieves the best results on the Cam1 test set (97.675%), meanwhile, maintaining performance on Cam2 at the top 3 (Only worse than versions A and M of TSSCL [81]). On the other hand, the Cam2 test set's best model is TSSCL-A [81] (95%).

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 256 | 88 |
| | **Negative** | 76 | 268 |

**(a)** C3D

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 344 | 0 |
| | **Negative** | 57 | 287 |

**(b)** ConvLSTM

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 344 | 0 |
| | **Negative** | 33 | 311 |

**(c)** 3D DenseNet

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 342 | 2 |
| | **Negative** | 22 | 322 |

**(d)** 3D DenseNet Lean

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 318 | 26 |
| | **Negative** | 9 | 335 |

**(e)** TSSCL-A

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 314 | 30 |
| | **Negative** | 22 | 322 |

**(f)** TSSCL-C

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 326 | 18 |
| | **Negative** | 8 | 336 |

**(g)** TSSCL-M

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 344 | 0 |
| | **Negative** | 19 | 325 |

**(h)** 3D DenseNet Fusion OF RGB

*True class*

| *Predicted class* | | **Positive** | **Negative** |
|---|---|---|---|
| | **Positive** | 344 | 0 |
| | **Negative** | 16 | 328 |

**(i)** Our proposed method

**Table 4.4:** Confusion matrix of selected baseline methods on AICS - violence Cam1 testset

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 192 | 152 |
| | **Negative** | 74 | 270 |

**(a)** C3D

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 330 | 14 |
| | **Negative** | 54 | 290 |

**(b)** ConvLSTM

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 342 | 2 |
| | **Negative** | 122 | 222 |

**(c)** 3D DenseNet

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 344 | 0 |
| | **Negative** | 124 | 220 |

**(d)** 3D DenseNet Lean

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 328 | 16 |
| | **Negative** | 19 | 325 |

**(e)** TSSCL-A

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 319 | 25 |
| | **Negative** | 25 | 319 |

**(f)** TSSCL-C

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 326 | 18 |
| | **Negative** | 22 | 322 |

**(g)** TSSCL-M

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 340 | 4 |
| | **Negative** | 48 | 296 |

**(h)** 3D DenseNet Fusion OF RGB

| | | True class: Positive | True class: Negative |
|---|---|---|---|
| Predicted class | **Positive** | 340 | 4 |
| | **Negative** | 41 | 303 |

**(i)** Our proposed method

**Table 4.5:** Confusion matrix of selected baseline methods on AICS - violence Cam2 testset

Tables 4.4 and 4.5 depict confusion matrix of baseline and our proposed methods on AICS - violence Cam1 and Cam2 test sets consecutively. At Cam1, our proposed method achieves a TP rate of 100% but still predict incorrectly 16 samples from non-violent class. This pattern is reasonable because the non-violent class consists of some challenging scripts such as running and shuttle kicking which could be confused with violent behaviors. On the other hand, TSSCL-A [81], which has the highest accuracy on the second camera, maintains a quiet balance of results between TN and TP on the Cam2 test set.

### 4.3.3 False cases of proposed method on AICS - violence dataset

**Figure 4.4:** Two frames from a clip with a non-intense fight.

**Figure 4.5:** Two frames from running and shuttlecock kicking respectively

In this section, we analyze in detail the false cases of our proposed method on the AICS - violence dataset. In violent cases, those with non-intense fights usually confuse our proposed method as non-violent. Particularly, the scenario illustrates in figure 4.4, in which 2 actors are going to finish the fight, causes the model to predict non-violent behavior. On the other hand, all false positive cases are running and shuttlecock kicking which contains multiple actions similar to violence such as waving hands, and kicking as shown in figure 4.5

## 4.4 Accuracy on well-known benchmark dataset

| **Methods** | **Accuracy (%)** |
|---|---|
| 3D DenseNet Lean [78] | **98.5** |
| **Our proposed method** | 97 |

**Table 4.6:** Comparison of our proposed method and 3D DenseNet Lean [78] on Hockey Fights test set [28]

Table 4.6 describes accuracies of 3D DenseNet Lean [78] and our proposed methods. Generally, both methods achieved high accuracies on the Hockey Fight dataset [28] (At least 97%). However, there is a significant decline in the generalization of our fusion method compared to 3D DenseNet Lean [78] (From 98.5 to 97%). Our model extracted normalized features from visualized optical flow, then use it to augment features from RGB images. Therefore, Hockey fight scenes are captured from moving cameras which will cause noise. In conclusion, our method performs better on fixed cameras (such as surveillance) rather than on moving ones.

# CHAPTER 5. CONCLUSION

In this thesis, we have extended test sets of the AICS-Violence dataset which is designed specifically for outdoor surveillance cameras. Furthermore, a preprocessing method (based on a previous Human Area Detection algorithm) was proposed to focus on each group of people called candidate boxes. Finally, a novel method was introduced for violence detection by fusing features extracted from RGB and visualized optical flow frames using a 3D CNN. We then evaluated them on the extended test sets. Results illustrate that our proposed approach achieves high accuracies compared to other SOTA methods. In the future, we will enhance our models to achieve higher performance on more different camera views and angles.

## REFERENCES


[1] I. for Economics **and** P. (IEP), "Measuring the global economic impact of violence and conflict," **in***The Economic Value of Peace - 2016* December 21, 2016.

[2] L. L. Jiangfan Feng Yukun Liang, "Anomaly detection in videos using two-stream autoencoder with post hoc interpretability," **in***Computational Intelligence and Neuroscience: 7367870* 2021.

[3] A. S. for Photogrammetry **and** R. Sensing, **in***Photogrammetric Engineering Remote Sensing Vol. 84* April, 2018.

[4] T. B. Mehran Yazdi, **in***New trends on moving object detection in video images captured by a moving camera: A survey, Computer Science Review, Volume 28, 2018*.

[5] E. Z. Amira Ben Mabrouk, **in***Abnormal behavior recognition for intelligent video surveillance systems: A review, Expert Systems with Applications, Volume 91, 2018: ISSN 0957-4174*.

[6] J. C. Z. Shao **and** Z. Wang, "Smart monitoring cameras driven intelligent processing to big surveillance video data," **in***n IEEE Transactions on Big Data, vol. 4, no. 1* March 01, 2018.

[7] M. D. Genemo, "Suspicious activity recognition for monitoring cheating in exams," *Proceedings of the Indian National Science Academy. Part A, Physical Sciences*, **jourvol** 88, **number** 1, 1—10, 2022, ISSN: 0370-0046. **url**: https://europepmc.org/articles/PMC8866922.

[8] W. H. Organization, **in***World report on violence and health* 2002.

[9] I. C. Education, "Neural networks," **in***https://www.ibm.com/cloud/learn/neural-networks* August 17, 2020.

[10] P. Baheti, "12 types of neural network activation functions: How to choose?" **in***https://www.v7labs.com/blog/neural-networks-activation-functions* May 26, 2022.

[11] V. K. D. Nguyen Hong Son, "Violence detection in video using optical flow and deep learning features," School of Electronics, Telecommunications - Hanoi University of Science **and** Technology, VN), year = 2021.

[12] Z. Z. G. A. K. M. Omarov B Narynov S, "State-of-the-art violence detection techniques in video surveillance security systems: A systematic review," 2022, ISSN: 10.7717/peerj-cs.920.

[13] *Edge detection*, https://www.researchgate.net/figure/Top-Lena-Image-Bottom-Detected-edges-using-the-Local-

Threshold-and-Boolean-Function_fig3_262276701, Accessed: 2022-06-30.

[14] *Feature extraction using deep learning*, https://www.slideshare.net/TerryTaewoongUm/introduction-to-machine-learning-and-deep-learning, Accessed: 2022-06-30.

[15] N. Dalal **and** B. Triggs, "Histograms of oriented gradients for human detection," **in***2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* Ieee, **volume** 1, 2005, **pages** 886–893.

[16] N. Dalal, B. Triggs **and** C. Schmid, "Human detection using oriented histograms of flow and appearance," **in***European conference on computer vision* Springer, 2006, **pages** 428–441.

[17] I. Laptev **and** T. Lindeberg, "Local descriptors for spatio-temporal recognition," **in***International Workshop on Spatial Coherence for Visual Motion Analysis* Springer, 2004, **pages** 91–103.

[18] Laptev **and** Lindeberg, "Space-time interest points," **in***Proceedings Ninth IEEE International Conference on Computer Vision* 2003, 432–439 vol.1. DOI: 10.1109/ICCV.2003.1238378.

[19] M.-Y. Chen **and** A. Hauptmann, "MoSIFT: Recognizing Human Actions in Surveillance Videos," **august** 1995. DOI: 10.1184/R1/6607523.v1. **url**: https://kilthub.cmu.edu/articles/journal_contribution/MoSIFT_Recognizing_Human_Actions_in_Surveillance_Videos/6607523.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, **jourvol** 60, **number** 2, **pages** 91–110, 2004.

[21] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma **and** B. Yu, "Recent advances in convolutional neural network acceleration," **july** 2018.

[22] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," **in***ECML* 1998.

[23] G. Csurka, C. Dance, L. Fan, J. Willamowski **and** C. Bray, "Visual categorization with bags of keypoints," **in***Workshop on statistical learning in computer vision, ECCV* Prague, **volume** 1, 2004, **pages** 1–2.

[24] A. Lopes, E. Jr **and** A. Araújo, "Action recognition in videos: From motion capture labs to the web," *ArXiv CoRR*, **june** 2010.

[25] K. Soomro, A. R. Zamir **and** M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, **jourvol** abs/1212.0402, 2012. arXiv: 1212.0402. **url**: http://arxiv.org/abs/1212.0402.

[26] .

[27] O. Deniz, I. Serrano Gracia, G. Bueno **and** T.-T. Kim, "Fast violence detection in video," **volume** 2, **december** 2014.

[28] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García **and** R. Sukthankar, "Violence detection in video using computer vision techniques," **in***Computer Analysis of Images and Patterns* P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano **and** W. Kropatsch, **editors**, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, **pages** 332–339, ISBN: 978-3-642-23678-5.

[29] P. Wang, P. Wang **and** E. Fan, "Violence detection and face recognition based on deep learning," *Pattern Recognition Letters*, **jourvol** 142, **pages** 20–24, **february** 2021. DOI: 10.1016/j.patrec.2020.11.018.

[30] M. yu Chen **and** A. Hauptmann, "Mosift : Recognizing human actions in surveillance videos cmu-cs-09-161," 2009.

[31] U. P **and** L. G G, "Skeleton-based stip feature and discriminant sparse coding for human action recognition," *International Journal of Intelligent Unmanned Systems*, **jourvol** 9, **pages** 20–24, **february** 2021. DOI: 10.1016/j.patrec.2020.11.018.

[32] F. Souza **and** H. Pedrini, "Detection of violent events in video sequences based on census transform histogram," **october** 2017, **pages** 323–329. DOI: 10.1109/SIBGRAPI.2017.49.

[33] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma **and** B. Yu, "Recent advances in convolutional neural network acceleration," *CoRR*, **jourvol** abs/1807.08596, 2018. arXiv: 1807.08596. **url**: http://arxiv.org/abs/1807.08596.

[34] P. C. Ribeiro, R. Audigier **and** Q. C. Pham, "Rimoc, a feature to discriminate unstructured motions: Application to violence detection for video-surveillance," *Computer Vision and Image Understanding*, **jourvol** 144, **pages** 121–143, 2016, Individual and Group Activities in Video Event Analysis, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2015.11.001. **url**: https://www.sciencedirect.com/science/article/pii/S1077314215002374.

[35] C. Yao, X. Su, X. Wang, X. Kang, J. Zhang **and** J. Ren, "Motion direction inconsistency-based fight detection for multiview surveillance videos," *Wireless Communications and Mobile Computing*, **jourvol** 2021, **pages** 1–11, **may** 2021. DOI: 10.1155/2021/9965781.

[36] Y. Wang, K. Huang **and** T. Tan, "Human activity recognition based on r transform," **june** 2007. DOI: 10.1109/CVPR.2007.383505.

[37] T. Zhang, W. Jia, B. Yang, J. Yang, X. He **and** Z. Zheng, "Mowld: A robust motion image descriptor for violence detection," *Multimedia Tools and Applications*, **jourvol** 76, **january** 2017. DOI: 10.1007/s11042-015-3133-0.

[38] V. Machaca Arceda, K. Fernández Fabián, P. Laguna Laura, J. Rivera Tito **and** J. Gutiérrez Cáceres, "Fast face detection in violent video scenes," *Electronic Notes in Theoretical Computer Science*, **jourvol** 329, **pages** 5–26, 2016, CLEI 2016 - The Latin American Computing Conference, ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2016.12.002. **url**: https://www.sciencedirect.com/science/article/pii/S1571066116301050.

[39] M. Rahman, R. Rahman, K. A. Supty **andothers**, "A real time abysmal activity detection system towards the enhancement of road safety," **in***2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)* 2022, **pages** 1–5. DOI: 10.1109/IRASET52964.2022.9738165.

[40] J. Xie, W. Yan, C. Mu, T. Liu, P. Li **and** S. Yan, "Recognizing violent activity without decoding video streams," *Optik*, **jourvol** 127, **number** 2, **pages** 795–801, 2016, ISSN: 0030-4026. DOI: https://doi.org/10.1016/j.ijleo.2015.10.165. **url**: https://www.sciencedirect.com/science/article/pii/S0030402615015338.

[41] K. Reddy **and** M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, **jourvol** 24, **july** 2013. DOI: 10.1007/s00138-012-0450-4.

[42] I. P. Febin, K. Jayasree **and** P. T. Joy, "Violence detection in videos for an intelligent surveillance system using mobsift and movement filtering algorithm," *Pattern Anal. Appl.*, **jourvol** 23, **number** 2, 611–623, 2020, ISSN: 1433-7541. DOI: 10.1007/s10044-019-00821-3. **url**: https://doi.org/10.1007/s10044-019-00821-3.

[43] V. Kantorov **and** I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," **in***Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* **jourser** CVPR '14, USA: IEEE Computer Society, 2014, 2593–2600, ISBN: 9781479951185. DOI: 10.1109/CVPR.2014.332. **url**: https://doi.org/10.1109/CVPR.2014.332.

[44] T. Senst, V. Eiselein, A. Kuhn **and** T. Sikora, "Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation," *IEEE Transactions on Information Forensics and Security*, **jourvol** PP, **pages** 1–1, **july** 2017. DOI: 10.1109/TIFS.2017.2725820.

[45] T. Hassner, Y. Itcher **and** O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," **june** 2012, **pages** 1–6, ISBN: 978-1-4673-1611-8. DOI: 10.1109/CVPRW.2012.6239348.

[46] T. Cheng **and** D. Williams, "Space-time analysis of crime patterns in central london," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, **jourvol** 39, **pages** 47–52, **july** 2012. DOI: 10.5194/isprsarchives-XXXIX-B2-47-2012.

[47] L. Ye, L. Wang, H. Ferdinando, T. Seppänen **and** E. Alasaarela, "A video-based dt–svm school violence detecting algorithm," *Sensors (Basel, Switzerland)*, **jourvol** 20, 2020.

[48] T. Zhang, Z. Yang, W. Jia, B. Yang, J. Yang **and** X. He, "A new method for violence detection in surveillance scenes," *Multimedia Tools and Applications*, **jourvol** 75, **pages** 7327–7349, 2015.

[49] S. Blunsden **and** R. B. Fisher, "The behave video dataset: Ground truthed video for multi-person behavior classification," 2010.

[50] J. L. Crowley, P. Reignier **and** S. Pesnel, *Context aware vision using image-based active recognition*, 2004.

[51] Y. Gao, H. Liu, X. Sun, C. Wang **and** Y. Liu, "Violence detection using oriented violent flows," *Image and Vision Computing*, **jourvol** 48-49, **february** 2016. DOI: 10.1016/j.imavis.2016.01.006.

[52] k. xu, X. Jiang **and** T. Sun, "Anomaly detection based on stacked sparse coding with intraframe classification strategy," *IEEE Transactions on Multimedia*, **jourvol** PP, **pages** 1–1, **march** 2018. DOI: 10.1109/TMM.2018.2818942.

[53] D. K., V. L.K.P. **and** C. S., "Autocorrelation of gradients based violence detection in surveillance videos," *ICT Express*, **jourvol** 6, **number** 3, **pages** 155–159, 2020, ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2020.04.014. **url**: https://www.sciencedirect.com/science/article/pii/S2405959520300990.

[54] M. Al-N'awashi, O. Al-hazaimeh **and** M. Saraee, "A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments," *Neural Computing and Applications*, **jourvol** 28, **december** 2017. DOI: 10.1007/s00521-016-2363-z.

[55] D. Song, K. Chansu **and** S.-K. Park, "A multi-temporal framework for high-level activity analysis: Violent event detection in visual surveillance," *Information Sciences*, **jourvol** 447, **march** 2018. DOI: 10.1016/j.ins.2018.02.065.

[56] N. Zhuang, G.-J. Qi, T. Kieu **and** K. Hua, "Differential recurrent neural network and its application for human activity recognition," **may** 2019.

[57] P. Vashistha, C. Bhatnagar **and** M. A. Khan, "An architecture to identify violence in video surveillance system using vif and lbp," **in***2018 4th International Conference on Recent Advances in Information Technology (RAIT)* 2018, **pages** 1–6. DOI: 10.1109/RAIT.2018.8389027.

[58] C. Ding, S. Fan, M. Zhu, W. Feng **and** B. Jia, "Violence detection in video by using 3d convolutional neural networks," **in***ISVC* 2014.

[59] D. R, E. Fenil, G. Manogaran **andothers**, "Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional lstm," *Computer Networks*, **jourvol** 151, **march** 2019. DOI: 10.1016/j.comnet.2019.01.028.

[60] G. Mu, H. Cao **and** Q. Jin, "Violent scene detection using convolutional neural networks and deep audio features," **volume** 663, **november** 2016, **pages** 451–463, ISBN: 978-981-10-3004-8. DOI: 10.1007/978-981-10-3005-5_37.

[61] C.-H. Demarty, B. Ionescu, Y.-G. Jiang, V. L. Quang, M. Schedl **and** C. Penet, "Benchmarking violent scenes detection in movies," **in***2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)* 2014, **pages** 1–6. DOI: 10.1109/CBMI.2014.6849827.

[62] S. Sudhakaran **and** O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," *CoRR*, **jourvol** abs/1709.06531, 2017. arXiv: 1709.06531. **url**: http://arxiv.org/abs/1709.06531.

[63] A. Naik **and** M. Gopalakrishna, "Deep-violence: Individual person violent activity detection in video," *Multimedia Tools and Applications*, **jourvol** 80, **pages** 1–16, **may** 2021. DOI: 10.1007/s11042-021-10682-w.

[64] M. Blank, L. Gorelick, E. Shechtman, M. Irani **and** R. Basri, "Action as space-time shapes," **volume** 29, **november** 2005, 1395–1402 Vol. 2, ISBN: 0-7695-2334-X. DOI: 10.1109/ICCV.2005.28.

[65] C. Schüldt, I. Laptev **and** B. Caputo, "Recognizing human actions: A local svm approach," **volume** 3, **september** 2004, 32 –36 Vol.3, ISBN: 0-7695-2128-2. DOI: 10.1109/ICPR.2004.1334462.

[66] Z. Meng, J. Yuan **and** Z. Li, "Trajectory-pooled deep convolutional networks for violence detection in videos," **in***International Conference on Computer Vision Systems* Springer, 2017, **pages** 437–447.

[67] F. Ullah, A. Ullah, K. Muhammad, I. Haq **and** S. Baik, "Violence detection using spatiotemporal features with 3d convolutional neural network," English,

*Sensors (Switzerland)*, **jourvol** 19, **number** 11, **june** 2019, ISSN: 1424-8220. DOI: 10.3390/s19112472.

[68] R. Roy, "Ai, ml and dl: How not to get them mixed!" **in***Towards Data Scince* April 29, 2020.

[69] Wikipedia, "Machine learning," **in***https://en.wikipedia.org/wiki/Machine*$_{l}earning$.

[70] I. C. Education, "Supervised learning," **in***https://www.ibm.com/cloud/learn/supervised-learning* August 19, 2020.

[71] *Knn*, https://www.researchgate.net/figure/K-Nearest-Neighbor-KNN-classification-principle_fig4_343080916, Accessed: 2022-7-24.

[72] *Basic machine learning*, https://www.coursera.org/learn/machine-learning, Accessed: 2022-7-24.

[73] Wikipedia, "Association rule learning," **in***https://en.wikipedia.org/wiki/Association*$_{r}ule_{l}earning$

[74] *Cs231n*, https://cs231n.github.io/, Accessed: 2022-7-24.

[75] *Optical flow*, https://medium.com/swlh/what-is-optical-flow-and-why-does-it-matter-in-deep-learning-b3278bb205b5, Accessed: 2022-7-30.

[76] *Opencv optical flow*, https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html, Accessed: 2022-7-30.

[77] A. Bochkovskiy, C. Wang **and** H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, **jourvol** abs/2004.10934, 2020. arXiv: 2004.10934. **url**: https://arxiv.org/abs/2004.10934.

[78] .

[79] D. Tran, L. Bourdev, R. Fergus, L. Torresani **and** M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," **in***2015 IEEE International Conference on Computer Vision (ICCV)* Santiago, Chile: IEEE, **december** 2015.

[80] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. K. Wong **and** W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," **june** 2015.

[81] Z. Islam, M. Rukonuzzaman, R. Ahmed, M. Kabir **and** M. Farazi, "Efficient two-stream network for violence detection using separable convolutional lstm," **july** 2021, **pages** 1–8. DOI: 10.1109/IJCNN52387.2021.9534280.

[82] J. Carreira, E. Noland, C. Hillier **and** A. Zisserman, *A short note on the kinetics-700 human action dataset*, **july** 2019.

[83] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li **and** F.-F. Li, "Imagenet: A large-scale hierarchical image database," **june** 2009, **pages** 248–255. DOI: 10.1109/CVPR.2009.5206848.

[84] M. Cheng, K. Cai **and** M. Li, "Rwf-2000: An open large scale video database for violence detection," **in***2020 25th International Conference on Pattern Recognition (ICPR)* IEEE, 2021, **pages** 4183–4190.

[85] A. Paszke, S. Gross, F. Massa **andothers**, *Pytorch: An imperative style, high-performance deep learning library*, **december** 2019.

[86] M. Abadi, P. Barham, J. Chen **andothers**, "Tensorflow: A system for large-scale machine learning," **in***12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* 2016, **pages** 265–283. **url**: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.