Transformer-based Deep Learning Approach to Real-Time Violence Detection

by

Md Ahsan Hasib
21101119
Md Hasibul Hasan
21101314
Md Ragib Asif
21101083
Sadril Amin Shrestho
21101235
Nazmul Haque Omi
21101272

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

Department of Computer Science and Engineering
Brac University
May 2024

Declaration

It is hereby declared that

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

Student's Full Name & Signature:

Md Ahsan Hasib
21101119

Md Ragib Asif
21101083

Md Hasibul Hasan
21101314

Sadril Amin Shrestho
21101235

Nazmul Haque Omi
21101272

Approval

The thesis/project titled "Transformer-based Deep Learning Approach to Real-Time Violence Detection" submitted by

1. Md Ahsan hasib (21101119)
2. Md Hasibul Hasan (21101314)
3. Md Ragib Asif (21101083)
4. Sadril Amin Shrestho (21101235)
5. Nazmul Haque Omi (21101272)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 03, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

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

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Violence detection has always been a challenging task in the field of computer vision and machine learning due to the complexity of real-world environments, imbalanced data, and the need for real-time performance. Furthermore, automated violence detection in surveillance systems is essential for enhancing public safety and enabling advanced security applications. Over these years several models such as CNN+LSTM, MSBT, SlowFast and many machine learning techniques have been adopted to classify violence. While existing models have achieved strong results in binary classification, their performance often falters when applied to large, imbalanced multiclass datasets like UCF-Crime. In this work, we propose a transformer based lightweight model, the Dynamic Memory Bank Fused Attention Network (DMFA-Net), designed to overcome these limitations. Our model leverages a Cross Attention mechanism to selectively retrieve relevant information from a Memory Bank, allowing it to achieve significantly higher accuracy in both binary and multiclass violence detection tasks. Experimental results demonstrate that DMFA-Net outperforms existing state-of-the-art models in the field. We also discuss the practical integration of our approach into real-time, autonomous surveillance systems for reliable violence detection.

Keywords: Real-time violence, Transformers

Table of Contents

List of Figures

List of Tables

Nomenclature

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

$BEiT$ BERT Pre-Training of Image Transformers

$BERT$ Bidirectional Encoder Representations from Transformers

$BiLSTM$ Bidirectional Long Short-Term Memory

$CCTV$ Closed Circuit Television

CNN Convolutional neural network

DNN Deep neural network

$LSTM$ Long Short-Term Memory

$SMAST$ Spatio-Temporal Action Detection

$YOLO$ You only look once

Chapter 1

Introduction

1.1 Introduction

Violence in public places is increasing worldwide, it is becoming a very serious problem in most of the populated countries like Bangladesh. Some places are always risky to visit because of the previous experience of violence which is a huge problem for society and the whole country. To detect violence in public places, a manual human monitoring system is a very labor-intensive task and also it needs more and more manpower. In most of the crowded places, CCTV cameras are available, it will be very efficient to monitor violence autonomously based on computer vision techniques instead of manual human monitoring. We are developing a technique to detect violence from CCTV footage using Artificial Intelligence and modern Computer Vision techniques. For that, we intend to develop a Transformer architecture-based deep learning model to train on a large-scale dataset capable of processing complex video data to detect violence with maximum accuracy. The main target of our research is to develop an efficient and accurate system for real-time violence detection. Our model will address the limitations of traditional methods and will be able to handle large-scale datasets effectively.

Traditional CNN and LSTM-based approach has some problems with Violence Detection due to its sequential processing which is slow and inefficient for training large-scale datasets. These approaches face difficulties in detecting complex video data in crowded public places like shopping malls, and educational institutions. Additionally, it also confused normal human activities like hugging and handshaking with violent activity, as a result, the detection accuracy drops for these traditional CNN and LSTM-based approaches. To solve this problem, we are introducing a Transformer-based deep learning model to detect complex human interaction in crowded places.

In this paper, we propose a model that leverages DINOv2 [21] as a feature extractor, incorporating a positional encoder and a memory block to capture temporal dependencies. The extracted features are then processed through cross-attention layers, followed by a pooling layer, and finally passed through a dense Fully Connected Network (FCN) layer for class prediction.

1.2 Problem Statement

Violence recognition in real-time video streams is an extremely important issue with a wide range of applications in public security, surveillance, and law enforcement. After all these significant advancements, existing models face various challenges and limitations. Take the combined model of CNN and LSTM [11] as an example, which does not have adaptive Conv3D [14] and has limited effectiveness in handling large-scale datasets. Not only that, models that use OpenPose for multi-person 2D pose estimation and YoloV3 for person detection [9] — or any other models similarly configured confuses non-violent actions like hugging and handshaking with violent ones like punching.

In addition, deep learning architectures (like ResNet50 [6], ConvLSTM [14], CUE-NET[24]), although powerful, have high risk of over-fitting, great computational cost and applicable real-time infeasible but this brings issues regarding duplication, modality unfairness and asynchronous in a multi-modal manner, and therefore the detection process becomes more complicated. For example, Multi-scale Bottleneck Transformer (MSBT) [25]. ViTPose [19] is another model that does a fine job in human pose and key-point estimation, but needs a high-quality dataset and faces a generalization problem among various angles & scenes.

After all, we can say there are some common problems like high computational cost, dataset dependency, limited classification, miss-classifications in violent detection. That's why we intend to develop a lightweight Transformer-based model that will address these limitations.

- Operates with low computation requirements for real-time detection of violence, making it useful on average machines.
- Categorizing varied actions of violence (punch, kick, slapping, kidnapping etc) - Threat level assessment as well.
- Minimize the errors in classifying adding in methods to track difficult human hands (a punch vs. a high five).

We aim to bridge the gap between existing limitations and the need for an efficient, real-time violence detection system. We also intend to build a dataset collecting videos from different datasets and YouTube with occlusions, in-crowd etc kinds of videos for better training purposes.

1.3 Research Objective

Machine learning have come a long way since the introduction of transformers back in 2017; by using Transformer's self-attention mechanism this research aims to develop and implement a model that will be able to identify, classify and determine the severity of the environment. Moreover, it will be a lightweight model that does not require more computational power and can be used in a moderate setup. In terms of classification, the model will be able to classify different types of violence punching, kicking, slapping, kidnapping, snatching, etc. Furthermore, this model

will be able to generate threat levels to determine the severity; for example, fist fighting and fighting with a lethal weapon are not the same kind of violence they have different threat levels. Another thing is that in most types of violence, there needs to be human interaction among themselves; sometimes this interaction can be misleading "fist bump" may look like a fist fight and "hi-five" may look like slapping each other so the model will have to take series of action to determine if it is real violence or not. Then there will be a recording system that will record the video and save it in a storage system which will help later in terms of proving there was violence. Last but not least we have to ensure the ethical use and privacy concerns of deploying a real-time violence detection system.

The potential objectives of the research are:

1. Understanding of the transformers and video transformers model and how to implement them in a real-life environment.
2. Develop a lightweight transformers-based model that will detect and identify violence in real-time.
3. Evaluate the performance and compare them with available models.
4. Find out the usefulness, way of improvement and drawbacks of the model.

Chapter 2

Related Work

2.1 Existing violence detection models

Experiments with several models for violence detection in videos demonstrate significant gains in performance accuracy and efficiency on different datasets. While the CUE-Net architecture [24] leads the way with 94.00% accuracy on the RWF-2000 dataset [8] and 99.50% on the RLVS dataset [3]. The early fusion model in the audio-visual fusion study [13] achieves an 88% accuracy rate, surpassing late fusion and AVE models. The ViTPose model [19] and CNN-BiLSTM network achieves state-of-the-art performances of 81.3% and 79.8% in terms of training and testing in human pose, together with real-time running times. Efficiency over accuracy: Models like ConvLSTM and LRCN [14] shows 80% and 83.33% test accuracy respectively, whereas MobileNet V2 [6] shows 92.3% action recognition accuracy. Combining Conv3D with LSTM [11] also yields a 98.6% accuracy on the brawl dataset, and the ViViT model [15] achieves 97.14% on the Hockey Fight dataset and 98.46% on the Violent Crowd dataset. Finally, the CNN-BiLSTM architecture [16] demonstrates solid accuracy rates across multiple datasets, including 94.9% on Hockey.

These models have far outperformed traditional methods. However, introduces novel ways to do better in terms of detecting violence efficiently and accurately in video analysis. Such as the integration of the CUE-Net with spatial cropping and MEAA, effective multi-modal fusion capabilities of the MSBT, and data augmentation strategies of the ViViT, which offer more robust and efficient solutions for use in real-world violence detection scenarios.

2.2 Literature Review

In paper [6], a real-time action recognition system that uses CNN and DNN instead of LSTM is described and they also used an optical flow image instead of RGB to reduce the size of the data. They have achieved a lightweight model to identify three sorts of violence punching, kicking and slapping. They collected 5 seconds of 600 videos (200 for each type) and hand-labelled them. Afterward, they used some pre-trained models VGG-16 and Mobile-Net-V2 to predict the output. They identified which model works better in terms of precision, recall, and F-1 score and found that Mobile-Net-V2 showed the most promising results. By using this system it will be possible to detect violent activity in videos. However, this system restricts itself to

only three types of violent activity other types of violent robbery, kidnapping etc.

This research paper [12] has developed a system that has a graphics user interface for interacting with the system. The system takes in real-time data and processes the data using various models. Firstly, the video goes through YOLOv3 and classifies the objects in the image then sends the classification and bounding box to DeepSORT which is an object detection model that works well with real-time operations. Furthermore, they used LSTM for the output of DeepSORT and then made predictions. This system is quite useful for detecting violent activity with the graphical user interface anyone will be able to use this. This system was trained and tested in only one dataset and that may lead to limitations of the system's ability. This model can be trained with more datasets then it might perform better in real-world situations.

This paper[24] has an architecture for automated violence detection. Here they have improved the Uniformer-V2 by cropping the image where humans are using YOLOv8 and then passing it into the model because they are passing cropped images in the Uniformer-V2 the model needs to deal with areas of interest where violence can be done. Furthermore, this model uses transformers which means it has to use self-attention which has quadratic time complexity, to reduce this time complexity they used self-developed MEAA(Modified Efficient Additive Attention) which uses element-wise multiplication instead of matrix multiplication that changes the complexity from quadratic to linear complexity. This architecture is quite impressive in terms of accuracy and can outperform many models like Video Swin Transformer and ACTIONVST in RWF-2000 and RLVS. This accuracy comes with some tradeoffs that require a huge amount of computational power and can be outperformed by lightweight models in terms of speed. A lightweight version of this model will be great for detecting violence in real-time in a real-world scenario as most of the surveillance stations do not have that much computational power.

To enhance public safety, the paper's [23] model recognizes crowd behavior as a secure smart surveillance system. This model efficiently handles computer vision tasks by utilizing deep learning techniques, specifically the Swim Transformer architecture. Using PyTorch's "onnx" module the DL model is converted into ONNX format for standardized representation. The system integrates geographically dispersed CCTV cameras and routers in the Source Spoke Layer, processes DL models in real-time at the Central Hub Layer, and ensures data security through a VPN and firewall in the Secure Transportation Layer. This model offers a proactive approach to crowd surveillance, providing smarter insights into city operations. However, it relies on a single dataset for training and testing, suggesting the need for broader dataset inclusion to enhance generalization. Future improvements could optimize computational resources for real-time analysis and explore additional DL architectures or ensemble learning techniques to improve classification accuracy.

A methodology for creating a video surveillance technique has been put out where LSTM was used to discover time-related correlations and for spatial feature extraction CNN was utilized in the research [11]. Their research reveals that transfer learning shows promising and quick results in detecting violence while combining

CNN, Conv3D and LSTM. For example, YouTube videos were tested at random and showed 100% accuracy in identifying. While the model performed well in identifying two people in a violent scene, it struggles with detecting in the crowd. Separating suspicious individuals and their actions in the Violent-flows dataset could enhance performance, with the choice of violence detection threshold affecting false positives and negatives. Future research might investigate adaptive thresholds based on scene context. The model could be improved by separately identifying suspicious individuals and their actions. Additionally, while LSTM models detect sequential patterns, transformer models with attention mechanisms excel at capturing both sequential and complex non-sequential patterns for sophisticated feature extraction.

In this paper [9], the researchers have proposed a complete system of violence detection that can work with surveillance systems that allows real-time video. In addition, OpenPose was used for pose detection, YoloV3 was used for person detection. They used Li-Net CNN as the classifier of their model. They were able to classify kicking and punching with a good frame in real time. This solution's drawback was that the classifier would flag an embrace as punching when two people approached to hug. They suggested a potential workaround that might be avoided by employing an RNN based model, and that utilizing LSTM rather than CNN would boost the model's reliability. However, transformer-based models can tackle this type of situation in terms of parallelization, long-range dependencies and transfer learning. Again, though their model had the accuracy of 94.0032 and 96.1199 for punching and kicking classifiers, other categories of violence were not included.

In order to accurately classify violent behavior in real-time and customize it to the specific needs of smart airports, this paper [19] presents a promising framework that utilizes ViTPose for human pose estimation and implements a CNN-BiLSTM based model for analyzing spatial and temporal information within key points sequences. Their selected final model had an accuracy of 81.3% and integrating with SAFE client and testing on DARTeC building at Cranfield University showed an effective result to use it in real-world scenarios. They mentioned that by adding attention mechanisms, the classifier's performance can potentially be enhanced by better capturing the temporal relationship within key points. Along with this, transformer models allow for parallel computing and the management of long-range relationships, which makes them perfect for jobs involving greater numbers of input.

This paper's [1] model also automates the recognition of violent actions in surveillance videos which makes manual monitoring inefficient. By utilizing the pipeline of BoW with HOG and HOF descriptors for feature extraction, enhancing classification accuracy. Early methods relied on specific cues like blood, flame detection, and sounds. Those often lead to misclassifications in crowded scenes because those were limited and audio-dependent. The introduction of Outlier-Resistant VLAD (OR-VLAD), a novel feature encoding technique based on higher-order statistics, significantly improves traditional VLAD performance in violence detection. Still, this model has some notable drawbacks. Rely on a single dataset for training and testing. It needs broader datasets to enhance model generalization. Future work could focus on optimizing computational resources for real-time analysis, exploring additional deep-learning architectures, or implementing ensemble learning tech-

niques to improve classification accuracy.

This paper [13] proposed a multi-modal violence detection that integrates audio and video data using pre-trained feature extractors to identify violent actions in transport environments. To process audio and video features separately, this model utilizes LSTM networks and fully connected layers before merging them for classification. The OpenL3 model extracts audio features, which are normalized and processed by an LSTM and fully connected layers. For video, the I3D feature extractor uses LSTM and fully connected layers to process various cropping zones. Leveraging multi-modal data and automating the recognition of violent actions, this model enhances transport safety and improves accuracy. However, the main change is differentiating violent actions from non-violent interactions in crowded or occluded environments. This leads to false predictions and reduced accuracy. Future developments could concentrate on improving the model's fusion approaches to better manage occlusions and distinguish between violent and nonviolent activities, ultimately increasing overall accuracy and lowering false positives.

A model Convolutional Neural Network Transformer (CNN-Transformer) where a 2D CNN network for extracting spatial information was wrapped by a temporal feature extractor time- distributed layer and then it was fed into a transformer was proposed in this paper [14]. The attention mechanism of the transformer was used to find both sequential and non- sequential patterns from the output of the time-distributed layer. The novel structure of the encoder-decoder has been proven several times to find linguistic features. It is also very effective in finding patterns in video frames allowing the layering of modules several times on each other. Relative position is added to each video's embedded vector. The mechanism of attention module processes different video sequences for values (V) and queries (Q), while Multi-Head Attention performs the same in a parallel manner to provide more focus on multiple frame sequences. The attention mechanism calculates weights using Softmax, influencing Q and K. During training, weight matrices adjust to learn patterns in V, Q, and K. The feed-forward layer applies identical linear transformations across each video in the sequence. However, the paper didn't specify any particular preprocessing pipeline and full model architecture and it was benchmarked on DAViD "A dataset for automatic violence detection in videos" dataset which consists of just 350 video clips which makes it questionable in real-world scenarios.

This paper [20] suggested preprocessing the video feed into two separate streams I^{RGB} and I^{FLOW} Then these two streams were passed to the semantic detection module and motion enhancement and segmentation module respectively. At the same time bounding box B^G was added to the motion semantics. Then the motion semantic vector S^V was fed into the motion memory module and motion semantics for both persons and objects SM was passed to the multi-feature selective semantic attention (MFSSA) module. Positional encoding from the motion semantic vector S^V was also fed into the (MMSSA) module. The output of the (MFSSA) module is AH, which represents the most informative patterns of the spatial and motion semantics which was extracted by the multi head attention layers. AH along with the motion memory features (F_M) is the input of the Sequence Based Temporal Attention (SB temporal attention) module which deals with both (A^H and F_M in

its own attention mechanism.The final stage of the proposed pipeline is the moduled called classification and regression consists of CONV and Softmax which gives the final logits. The multi-label classification layer includes two frame-level and sequence-level predictions. From that, we get our final predictions. However, the proposed pipeline uses a pre-trained network for motion enhancement & segmentation and semantic detection. A seamless architecture may deliver better encoding of the action semantics.Moreover, a server PC equipped with a dual Nvidia RTX 3090 GPUs (24GB VRAM), an AMD Ryzen Threadripper 3990X 64-Core Processor, and 256GB RAM was used for this experiment. The computation of 30 frames took 976 ms on this system. It calls into question how cost efficient it will be on a cloud server and real time usability on local deployment.

A multimodal approach for weakly supervised violence detection which Includes three components was discovered in this paper [25]. An unimodal encoder to encode RGB, flow and Audio streams (Z^R, Z^F, Z^A), a Multimodal Fusion Module which consists of MSBT layers which take (Z^R, Z^F, Z^A) as input and pass the output to the next MSBT layer and a Global Encoder Module which consists of a transformer and a regressor for final classification. There are a total of three input streams for this multimodal model. They are RGB stream, Flow stream and Audio stream. Each input stream's feature is extracted by its respective feature extractor backbone. I3D model for RGB or optical flow and the VGGish model for audio followed by a linear projection layer that outputs an embedded feature vector z which is then passed to an unimodal transformer and the output is Z^R, Z^F, Z^A for RGB, flow and audio stream respectively. Then Za_l, Zbt_l and Zb_l is the input of the (MSBT) layer of the multimodal Fusion Module. Here a ϵ M = R, F, A denotes one modality and l denotes to number of MSBT layer. After fusing all pairs of modalities, we obtain six fused features, including $Z^{RF}, Z^{FR}, Z^{RA}, Z^{AR}, Z^{FA}$, and Z^{AF}. And Z = $[Z^{RF}||Z^{FR}||Z^{RA}||Z^{AR}||Z^{FA}||Z^{AF}]$. Also, the output of the MSBT layers are fed into a transformer and regressor for weight computation which is then multiplied with Z and outputted as Z'. Z' is then fed into the Global Encoder Module which consists of a multimodal transformer followed by a regressor which gives the final prediction. This paper also proposes a new loss function Temporal Consistency Contrast (TCC) loss. Which detects any synchronization between audio, RGB, and Flow channel and allows the model to automatically minimize it. However, the effectiveness of this model with two modalities is not satisfactory in critical surveillance scenarios. Moreover, this experiment doesn't show any Computational Analysis which raises a question about real-time performance on a particular device in a surveillance scenario.

In this paper [15] an end-to-end deep learning-based method to detect violence using a video vision transformer known as ViViT. The authors show that the deep learning based approach acts better than the Handcrafted features-based approach in terms of accuracy. The authors introduced efficient techniques for preprocessing large amounts of data, video frame augmentation, and training the ViViT model efficiently. However, the model classifies violence based on two classes like violent and nonviolent. The model is computationally more efficient than the previous CNN, LSTM-based models. The ViViT-based violence detection model results from a promising performance of more than 97% accuracy. Despite the advantages in de-

tection accuracy of the model, it takes a large amount of data and longer training time to provide better results. In the future to make the model more powerful they can work on minimizing the training time without compromising the detection accuracy.

Patak et all. [22] proposed the Slow Fast Network which is a deep learning method for detecting anomalies in surveillance videos. The model is designed for capturing both spatial and temporal information. This model has two parts, the slow pathway and the fast pathway. The slow pathway deals with the spatial information which includes a dependable and slow CNN-based feature extractor and then passes the output of the extractor to a Bidirectional Long Short-Term Memory(BiLSTM). However, the fast pathway deals with temporal features and uses a shallower CNN-based feature extractor which is also passed to BiLSTM. Then both pathway combined gives the output. They achieved an overall 99 % accuracy on the UCF Crime dataset in terms of binary classification among Noramal and Anomaly.

A methodology has been proposed by Ullah at all. [10], in which they used a CNN network for feature extractor backbone and multiple LSTM blocks each corresponding to a particular input frame for capturing temporal information from frame to frame. This method shows 78.43% accuracy for multiclass classification on UCF-Crime dataset. It shows 98.20% and 98.80% accuracy scores for binary classification on UMN and Avenue datasets respectively.

Another method proposed by Qasim and Verdu[18] has shown comprehensive study on CNN+LSTM architecture. They also compared different pretrained CNN networks such as ResNet18, ResNet34, ResNet50 with their SRU (Simple Recurrent Unit) module and got the highest accuracy score with their ResNet50+SRU model. They have achieved 91.64% binary accuracy on UCF-Crime dataset with ResNet50+SRU model.

Chapter 3

Methodology

3.1 Datasets

Some key datasets for video violence detection include RWF2000, XD-Violence, RLVS, DAViD, Violent-flows, and UCF-Crime. These contain various footage types from surveillance cameras, movies, and online videos to help develop AI violence recognition systems.

Out of all these, we will specifically describe the dataset given below.

RWF-2000: This dataset [7] is one of the large-scale datasets made from raw surveillance videos from YouTube. Each clip is 5s with 30 fps and labeled as: Fight or Non-Fight behavior.

XD-Violence: Here [5] the dataset is made from 91 movies and also collected "in-the-wild" scenes from YouTube. Then to extract features from video clips pre-trained deep learning models were used: I3D model (for visual features) and VGGish model (for audio). It is a great dataset to use for building and testing machine learning models that aim at violence detection in videos, especially using visual and audio information combined. A total of six violence categories: Abuse, Car Accident, Explosion, Fighting, Riot, and Shooting. Video labels for the training set and frames are labels with temporal annotations for the test set.

RLVS: The RLVS [3] is a compilation of video clips sourced from public CCTV footage, social media platforms, and news reports. Collection of 1000 violent and 1000 non-violent videos containing street fight scenes, death footage, backgrounds, etc. Also include images of people doing other things like eating, walking in the park playing sports, etc. Each video clip is categorized by violence type, location, time of day, number of individuals, and presence of intervention (yes/no) means whether someone tries to stop or calm down a violent situation, like stepping in to break up a fight or talking to prevent further harm.

UCF Crime: UCF Crime dataset, a comprehensive collection of surveillance footage published by Boston University (BU) [2] focusing on crime and anomaly detection. This dataset offers significant contributions to the field of computer vision and surveillance technology, providing one of the most extensive collections of real-world crime-related video data. Because the dataset contains video samples representing 13 different crime categories and normal scenarios. The images were captured using various surveillance cameras to ensure the prevention of over exposure and distortion while representing authentic, unmodified footage. The aspect ratio and video qualities vary to reflect real-world recording conditions.

Table 3.1: Comparison among available datasets

Feature	XD-Violence [5]	RLVS [3]	RWF-2000 [8]	UCF-Crime [2]
Num. of Class	6	2	2	14
Num. of Videos	4754	2000	2000	1900
Temporal Annotation	No	Yes	Yes	Yes
Runtime (Hours)	217	20	3	128

3.1.1 Reason why we used UCF Crime

Out of all these datasets, we got the best performance from the UCF Crime. The UCF Crime dataset offers several unique features:

Real-world surveillance footage capturing diverse scenarios: Data collection is made up of Real CCTV and security camera footage. The security cameras are placed in different places and record actual events. This diversity is crucial for our model to run in the real-world.

Comprehensive representation of 14 crime and normal categories: This broad classification of this dataset helps in the development of our automated violence detection models to differentiate between many forms of criminal activity and normal behavior.

Unedited, authentic video recordings: All videos are presented in their raw form, not like data collected from movie scenes, or produced scenes, or edited scenes. So the dataset is filled with all sorts of challenging aspects of CCTV. It can be varying weather and day/night lighting conditions, different camera angles, occlusions, and different video quality. All of these make the dataset more typical to the monitoring real-world.

Temporal annotations for advanced analysis: The dataset tells us exactly when things happen - the start and end time of each event. This timing information helps our model better understand and track what's happening in the videos.

Coverage of indoor and outdoor environments: Footage from both indoor and outdoor spaces, including parking lots, stores, streets, and buildings is included in the dataset. This is one of the important diversity that helps our morel grow faster to operate well in both geographic and environmental environments.

3.1.2 Dataset Composition

The dataset consists of 1750 videos with a total duration of approximately 121 hours all are 30 fps. The videos in the dataset have a minimum frame count of 104 and a maximum frame count of 976,503. It's distinct 14 classes are namely Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Normal Videos, Road Accidents, Robbery, Shooting, Shoplifting, Stealing, and Vandalism shown below on Figure 3.1.

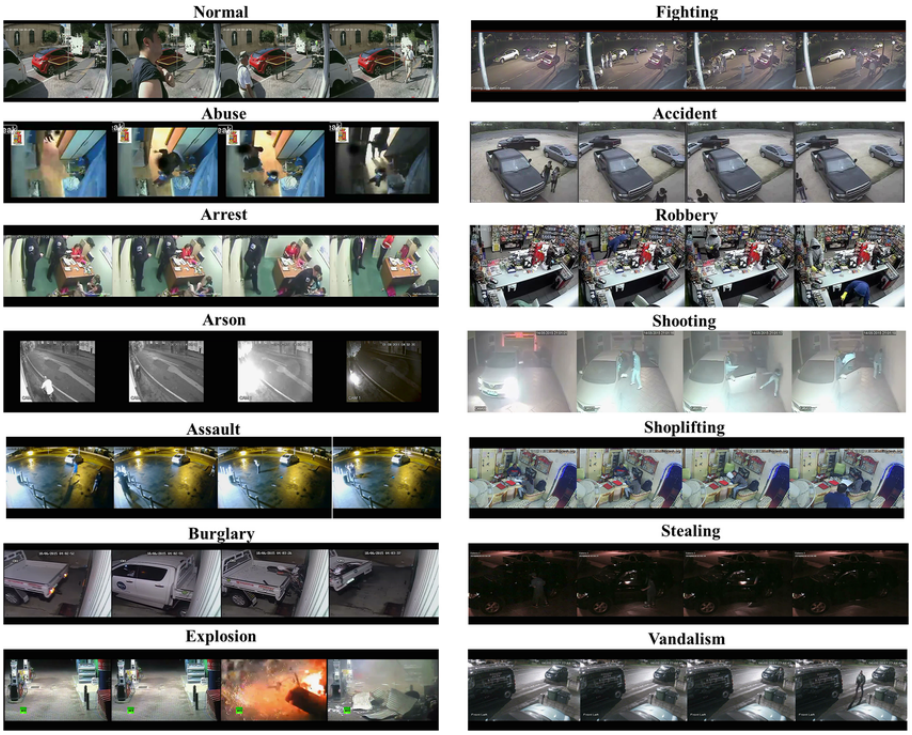


Figure 3.1: Representative Shots from different Categories [17]

This diversity maintains a distinct collection of data to assess the model's performance on unknown cases while guaranteeing that it has enough data to learn from, to work well with new, untested information. The table (3.2) provides detailed analysis of the dataset, outlining the distribution of videos, duration, and frame count for each category in the overall sets.

Category	No. of Videos	Duration (s)	Frame Count
Abuse	50	6449.9	193,497
Arrest	50	9913.4	297,395
Arson	50	9063.4	271,903
Assault	50	4331.5	129,946
Burglary	100	15706.9	471,206
Explosion	50	8413.5	252,404
Fighting	50	8629.8	258,895
Normal	800	315891.8	9,474,387
Road Accidents	150	8695.8	260,873
Robbery	150	14086.5	422,594
Shooting	50	4915.8	147,473
Shoplifting	50	10812.2	324,367
Stealing	100	15580.5	467,416
Vandalism	50	4905.4	147,162
Total	**1750**	**432490.97**	**13,119,518**

Table 3.2: Full dataset's Statistics

Figure 3.2: Normal Vs Anomaly

3.2 Workflow

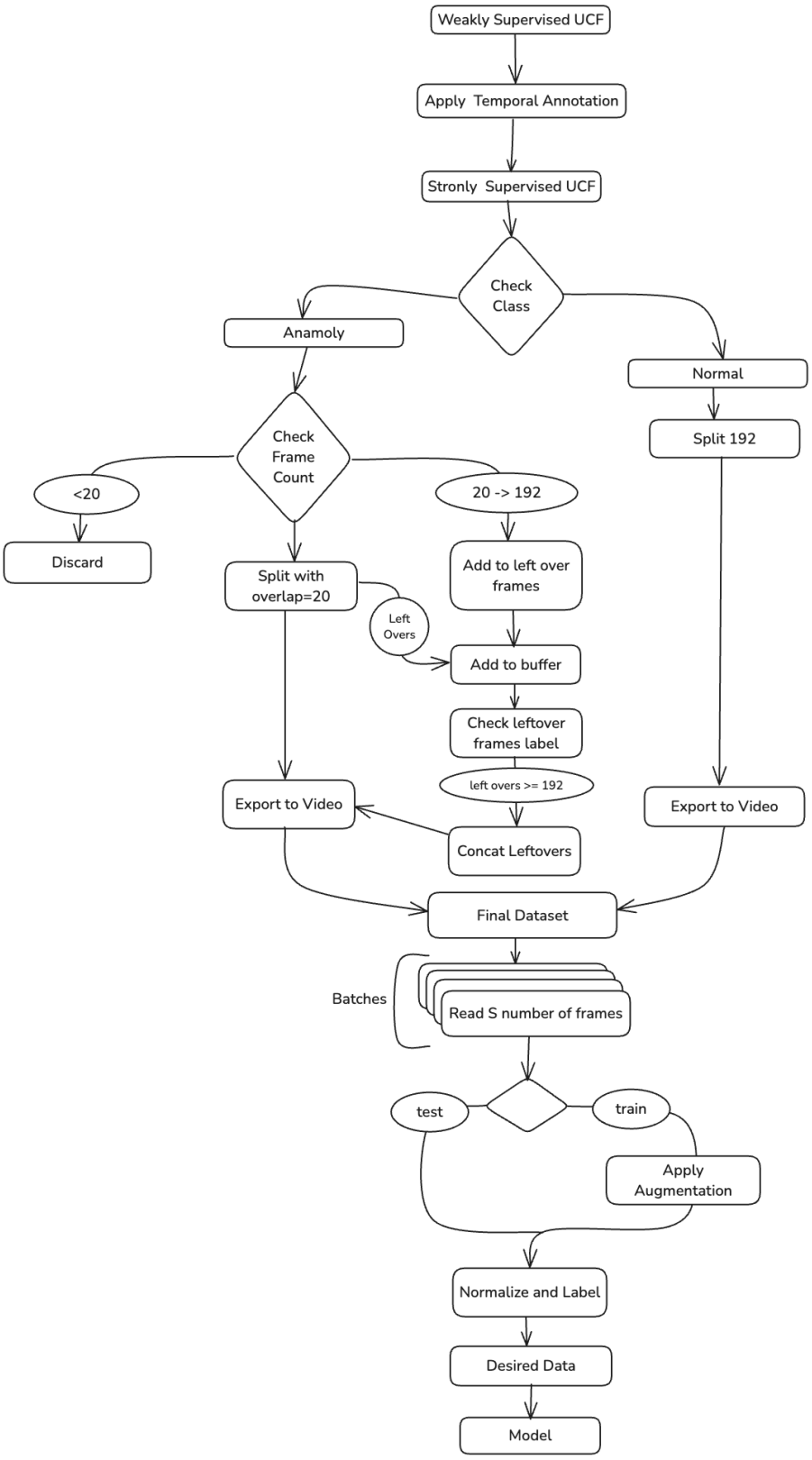


Figure 3.3: Methodology Flow Chart

3.2.1 Temporal Annotation and Video Trimming

As mentioned previously we are going to conduct our research on the UCF-Crime dataset which is quite a big and versatile dataset. It has videos of 13 abnormal classes and also includes a large number of normal class videos. Without any doubt, UCF-Crime is a strong dataset for training our kind of model.

In the UCF-Crime dataset, the labels indicate whether a video contains an anomaly or is normal; there is a frame-level annotation or temporal boundary for anomalies in the videos in their official site. For example, the **Robbery006_x264** video has a total runtime of 1 minute and 31 seconds or 2730 frames, but the actual robbery occurred between frames 300 and 870, and the rest does not contain any sort of anomalies. As our model takes a finite number of frames as input, there might be a sequence where there are anomalies, but it is still identified as an anomaly. Therefore, this makes the model inefficient in terms of identification.

By using the official temporal anomaly annotations we aimed to trim the video in a way that we could extract the anomaly or violent part from the actual video; we wrote a Python script in order to trim the video and save them into separate folders. By doing this we have reduced the size of the dataset which gave us faster training speed and now the data is more accurate with the labels.

3.2.2 Video Chunking and Preprocessing

The videos in the UCF-crime dataset do not have a uniform length, which means we can not process the data as it is. Because the dimensions or shape of the data would not match one another and we will not be able to fit the data in the model. So we must find a way, there are a few ways this can be done. The first one is to convert frames of the videos to images and save them in a folder with the label of the video, then take a constant number of images at a time. The second is storing the values in the form of Numpy (.np) or Torch (.pt) then reading the saved file and when required, the desired number of frames of data can be read from the saved file. Another option is to split the video in such a way that it is easier to fit in the model, we have divided the videos into chunks of 192 and for those videos whose frame count is not divisible 192, we have added padding if there are more than or equal to the sixty per cent of frames taken then padding was added at the end. By using the last mentioned way we have achieved a parallel, faster, and memory-efficient data loader which was sufficient for our model to run pre-process data using the CPU. At the same time, the GPU was busy doing forward pass and backward pass.

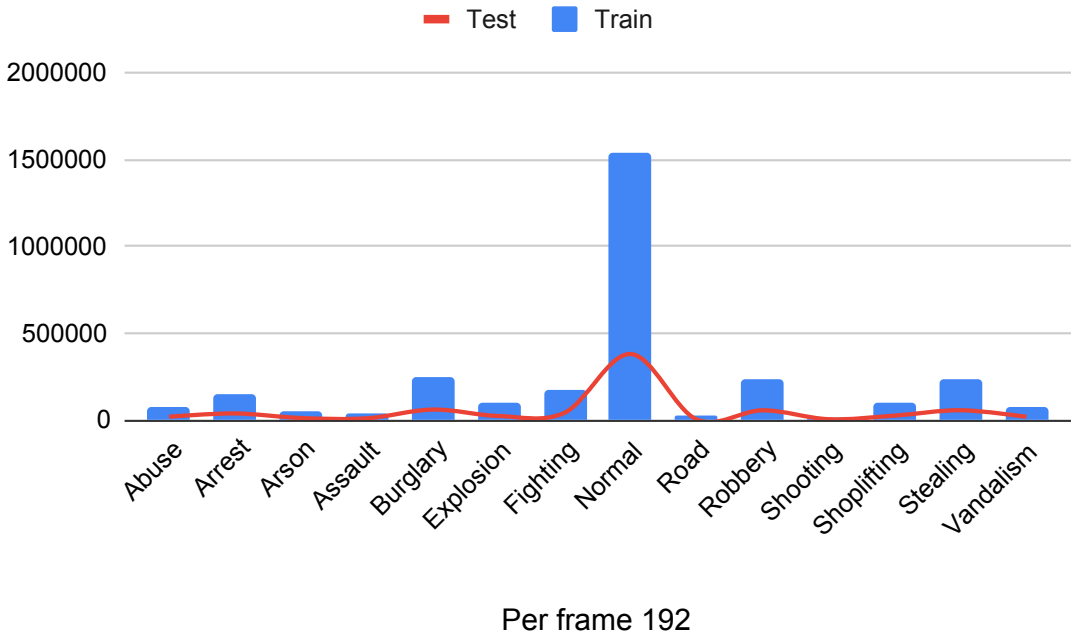


Figure 3.4: After pre-processing

3.2.3 Batching and Pulling

In the dataloader module, we took a sequence of frames and set the sequence size to a divisor of 192. Lets assume the sequence length is 24 and took 6 batches at a time, for our model to work properly we need to pass all 192 frames sequentially so we stored the batch ID and video path. Later, we calculated the starting and ending frame numbers using batch ID and then pulled the frames from the saved path(1).

Algorithm 1 Sequential chunk input for our model

$numChunks \leftarrow total\_frame/S$
for idx in $PathIndexes$ **do**
  $Chunk\_Idx \leftarrow idx \ \% \ numChunks$
  $start\_video\_idx \leftarrow Batch\_Idx \times Batch\_size$
  $Batch\_videos \leftarrow Videos[start\_video\_idx : start\_video\_idx + Batch\_size]$
  $Batch\_tensor \leftarrow readAndStack(Batch\_videos)$
  $StartFrame \leftarrow Chunk\_Idx \times chunkSize$
  $EndFrame \leftarrow StartFrame + chunkSize$
  $Batch\_chunk \leftarrow Batch\_tensor[StartFrame : EndFrame]$
  $Batch\_chunk \leftarrow Augmix(Batch\_chunk)$
end for

3.2.4 Data Augmentation

Additionally, for training, we used Augmix[4] for augmentation purposes, which allows us to increase the robustness of our model. Augmentation is a process of creating new data out of existing data. Here augmentation is done by using a randomly selected sequence of image alteration methods like rotation, contrast adjustment, translation etc. In this process, the essence of the data remains the same but the representation gets modified so that the model does not memorize the training data and becomes more efficient in a dynamic environment. Lastly, for each step we get the data with the shape of (6,24,3,224,244) then we pass this to our model.

Figure 3.5: Augmix Example

Chapter 4

Model Architecture

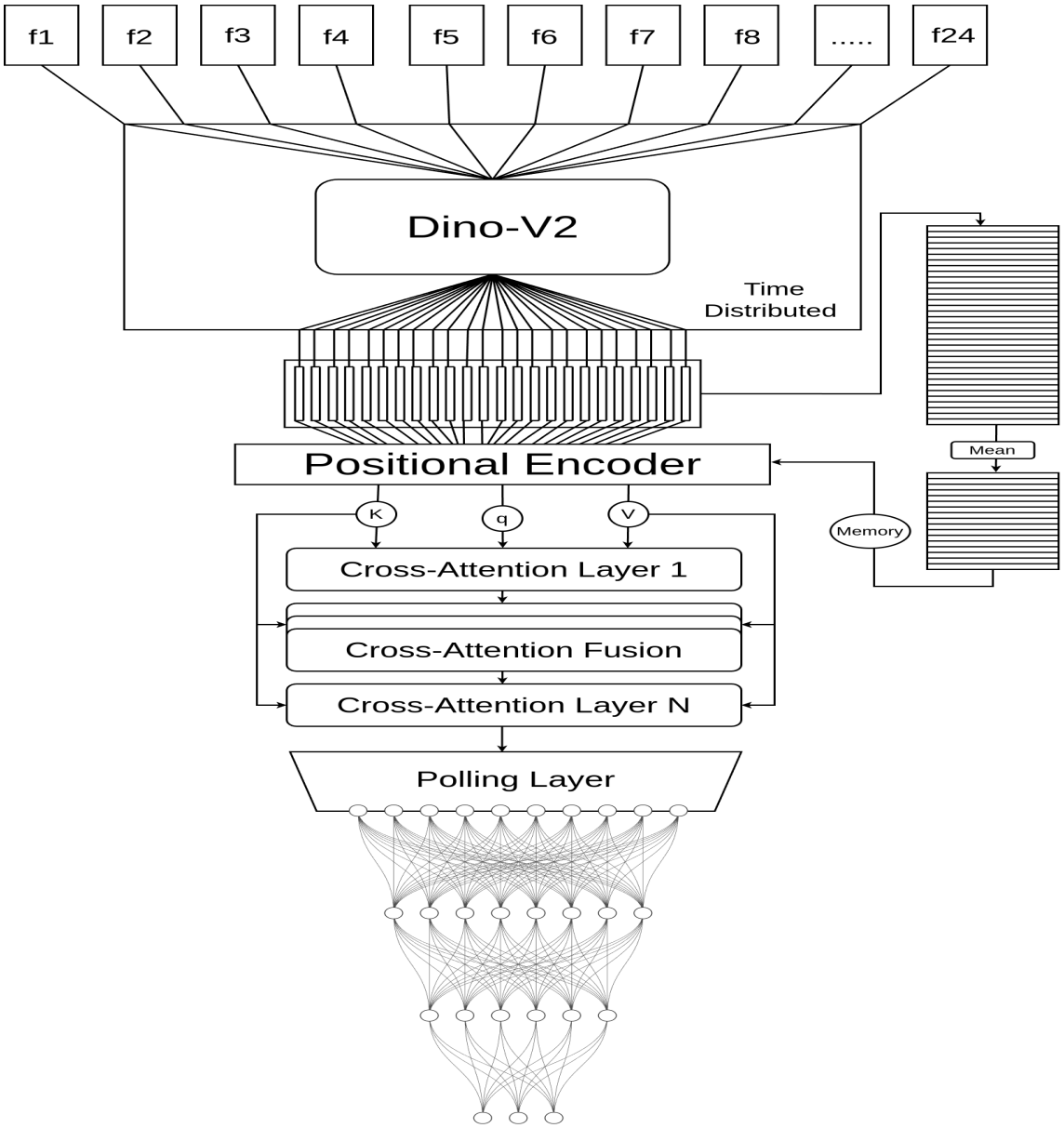


Figure 4.1: Model Architecture DMFA Net v1.1

4.1 DINOv2

We have selected the base variant of **DINOv2** [21] as a backbone for our model shown in figure (4.1), which is a self-supervised model based on **Vision Transformer** [15]. It produces robust performance in vision-related tasks, thanks to the **(Teacher/ Student)** architecture. We are choosing the base variant because it has the best balance between speed and feature quality. Unlike the small variant, which compromises accuracy, or the large variant which would be too slow for real-time applications. This makes it perfect for violence detection, where accurate detection and quick reaction are essential. More details on table 4.1

Feature	Arch	Food	C10	C100	SUN	Cars	Aircr	VOC	DTD	Pets	Cal101	Flowers	CUB	Avg
OpenCLIP	ViT-G/14	94.5	98.7	91.0	**84.0**	**96.1**	80.2	**89.3**	**86.0**	95.7	**98.1**	99.5	89.9	91.9
MAE	ViT-H/14	78.4	96.1	83.9	63.9	56.1	63.4	84.3	75.4	89.4	95.9	92.3	57.2	78.0
DINO	ViT-B/8	85.1	97.2	86.9	70.3	76.6	70.6	86.7	79.6	93.2	95.4	97.6	81.7	85.1
iBOT	ViT-L/16	91.0	99.0	92.8	75.6	71.8	72.4	89.0	80.7	87.7	97.5	99.6	82.1	86.6
DinoV2	ViT-S/14	89.1	97.7	87.5	74.4	81.6	74.0	87.8	80.6	95.1	97.0	99.6	88.1	87.7
	ViT-B/14	92.8	98.7	91.3	77.3	88.2	79.4	88.2	83.3	96.2	96.1	99.6	89.6	90.1
	ViT-L/14	94.3	99.3	93.4	78.7	90.1	81.5	88.3	84.0	96.6	97.5	99.7	90.5	91.2
	ViT-g/14	**94.7**	**99.5**	**94.4**	78.7	91.4	**87.2**	89.0	84.5	**96.7**	97.6	**99.7**	**91.6**	**92.1**

Table 4.1: Linear evaluation of frozen features on fine-grained benchmarks. [21]

Generate feature vector from last hidden state dimension: DINOv2 has **(14 × 14)** patches, **1 [CLS]** token, and **60** additional register tokens. Together, they form the resulting last hidden state shape of **(1, 257, 768)**. Here, the 196 patch embeddings hold the local information, the [CLS] token aggregates information from all tokens, and the 60 register tokens capture global information.

We used Custom Linear Pooling to cast all the 257 token of information as a large vector representation of shape **(1, 1024)**.

4.2 Custom Linear Pooling

Linear Pooling: This formula (4.1) can be thought of as a means to score the relative importance of each piece of information. In this case, ($\mathbf{h_i}$) represents a single token's information (image patch), and ($\mathbf{w_i}$) is the "importance score" that we determine for it. To learn complex relationships, we use two linear layers with a tanh function in between. The tanh function helps by squishing values between -1 and 1, making the scores more stable. It's like having a sophisticated scoring system that learns what features are worth paying attention to.

$$w_i = \text{Linear}_2\Big(\tanh\big(\text{Linear}_1(h_i)\big)\Big) \tag{4.1}$$

$$h_i = \text{hidden state of the } i\text{-th token} \tag{4.2}$$

h $\in \mathbb{R}^{1\times257\times768}$
$w \in \mathbb{R}^{1\times257}$

Softmax: Using the formula (4.3), we are converting our raw importance scores ($\mathbf{w_i}$) into probabilites ($\mathbf{a_i}$). The exponential function **(exp)** makes large scores even larger, resulting in greater contrasts. Then dividing by the sum of all ($\mathbf{w_i}$) ensures all scores add up to 100%. For example, if we have three tokens with scores of w_1 = 2, w_2 = 1, and w_3 = 0, after softmax they might became α_1 = 0.7, α_2 = 0.2, α_3 = 0.1, meaning we'll pay 70% attention to the first token, 20% to the second, and 10% to the third.

$$\alpha_i = \frac{\exp(w_i)}{\sum_{j=1}^{257} \exp(w_j)} \tag{4.3}$$

w $\in \mathbb{R}^{1\times257}$
$\alpha \in \mathbb{R}^{1\times257}$

Weighted sum: By learning the relative value of each token, the weighted sum (4.4) functions as a clever averaging method. Instead of treading all tokens equally, we determine how much each token's information ($\mathbf{h_i}$) should contribute to the field output **Z** using the attention probabilities ($\mathbf{a_i}$) as weights.

$$Z = \sum_{i=1}^{257} \alpha_i h_i \tag{4.4}$$

h $\in \mathbb{R}^{1\times257\times768}$
$z \in \mathbb{R}^{1\times768}$

Output projection: Output projection is a linear transformation layer. In order to store more contextual information within the Cross Attention Fusion layers, this output projection adjusts the dimensionality of each vector coming out of the weighted sum.

$$O = W_z Z + b_z \tag{4.5}$$

where W_z and $\mathbf{b}_z$ are learnable parameters.
$z \in \mathbb{R}^{1\times768}$
$O \in \mathbb{R}^{1\times1024}$

4.3 Time distributed Layer

Our model simultaneously takes the **"sequence length"** number of frames of 3 channel inputs. But DINOv2 takes 1 frame input (Excluding batch dimension). So we reshaped/ projected/ cast **(B, S, C, H, W)** to **(B × S, C, H, W)**. here **B × S** is the batch size for DINOv2. Then we reshaped the output **(B × S, hidden_dim)** to **(B, S, hidden_dim)** to get the "sequence length" dimension back again.

where **B** = Batch size
S = Sequence length
C = Channels
H = Height
W = Width

4.4 Positional encoding

To infuse temporal information into the vector sequence we use sinusoidal positional encoding. This encoding applies the sine function (4.6) to the odd positions and cosine functions (4.7) to the even positions. By doing this, absolute and relative temporal information between frames is guaranteed to be preserved.

$$PE_{p,2k} = \sin\left(\frac{\mathrm{P_{odd}}}{10000^{\frac{2k}{d}}}\right) \tag{4.6}$$

$$PE_{p,2k+1} = \cos\left(\frac{\mathrm{P_{even}}}{10000^{\frac{2k}{d}}}\right) \tag{4.7}$$

$\mathrm{p_{even}}$ = position of even frames→ (0, 2, 4,....., 22)
$\mathrm{p_{odd}}$ = position of odd frames→ (1, 3, 5,....., 23)
k = dimension indexes of the hidden_dim
d = dimension of hidden_dim

4.5 Memory Bank

In our Memory Bank module, we are using a **FIFO queue** to store the output from Time Distributed Layer. In the dataset each video has constant number of n frames.

$$\text{Max memory length} = \frac{n}{\text{seq\_len}} \tag{4.8}$$

Suppose, video has 96 frame. Then with seq_len of 24, it's memory length will be max 4. When current memory length < Max memory length, we apply zero padding. Before doing that we multiply temporal weights (0.5 → 1.0) to each memory elements (4.9). So that most recent ones gets more priority.

$$Memory_i = \alpha_i \times Memory_i \tag{4.9}$$

Finally, we apply **"mean"** function to reduce from shape **(B, n, hidden_dim)** to **(B, seq_len, hidden_dim)** to make it supported for the positional encoding and query for the cross attention fusion.

$$\mathrm{Memory_{out}} = \mathrm{LayerNorm}(Mean(Memory)) \tag{4.10}$$

Now, after applying positional encoding, $\mathrm{Memory_{out}}$ will serve as a summary of the previous frames.

4.6 Cross Attention Fusion

In the cross-attention fusion, we have taken a positionally encoded Memory Matrix as query and positionally Encoded Time Distributed output as **k** and **v**.

$$k = W_k(PE(X)) + b_k \tag{4.11}$$

$$v = W_k(PE(X) + b_k \tag{4.12}$$

Then to produce Numerical stability, we need to scale. We prescaled query with a fixed scaler $\sqrt{\mathbf{d_k}}$.

$$d_k = \frac{\text{hidden\_dim}}{\text{num\_heads}} \tag{4.13}$$

Then again scaled $\mathbf{qk^T}$ with a learnable scaler τ. This has produced numerical stability in the $\mathbf{qk^T}$ matrix.

$$q = \frac{W_q(PE(\text{Memory}_{\text{out}}) + b_q}{\sqrt{d_k}} \quad \text{[Prescaled query]} \tag{4.14}$$

$$A = \text{softmax}\left(\frac{qk^T}{\tau}\right) \quad \text{[Scaling } qk^T \text{ with a learnable scalar } \tau\text{]} \tag{4.15}$$

We applied output projection to the Cross attention output followed by a residual connection with X encoded and a feed forward network. We then repeated the whole cross attention calculation "num_layer" times.

$$\begin{aligned} O &= AV \\ O &= W_0 O + b_0 \\ O_{\text{res}} &= PE(x) + \text{Dropout}(O) \quad \rightarrow \text{residual connection} \\ O_{\text{ffn}} &= \text{FFN}(\text{LayerNorm}(O_{\text{res}})) \\ Z &= O_{\text{res}} + O_{\text{ffn}} \end{aligned}$$

4.7 Prediction Layer

We are pooling the output (4.16) coming from the Cross Attention Fusion Layer to reduce its dimensionality before passing it to the FCN layer (4.17) for final prediction.

$$Z = W_p Z + b_p \tag{4.16}$$

$$\hat{y} = \text{FFN}(z) \tag{4.17}$$

4.8 DMFA NetV1

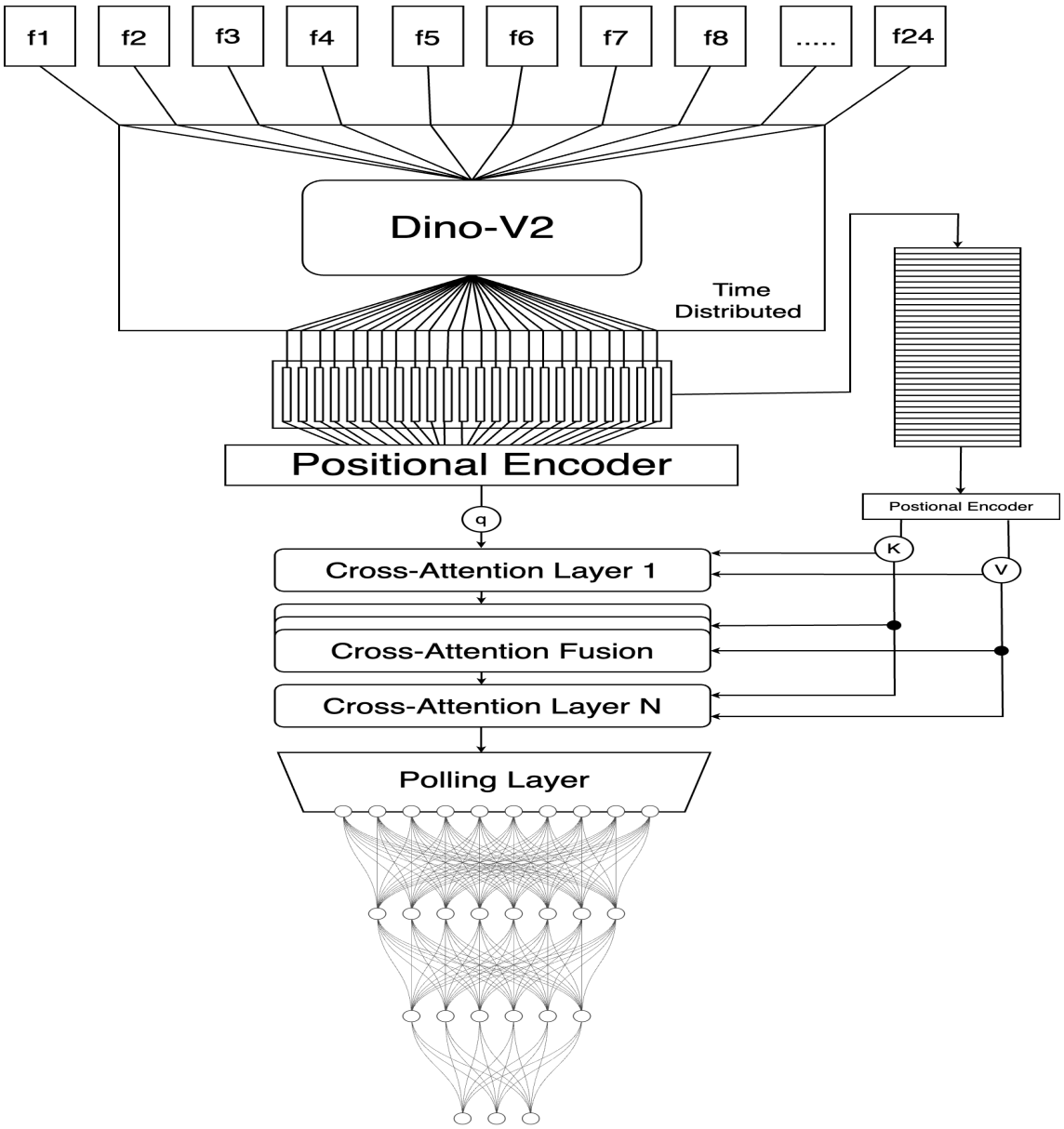


Figure 4.2: Model Architecture DMFA Net v1.0

In our 2nd model shown in figure (4.2), we've changed a couple of things to have further study on the characteristics of key, value & query in attention mechanism in video sequence classification task.

In this setup, we have discarded the Custom Linear Pooling of DINOv2 output. Rather we are taking the [CLS] token and Mean of all 257 tokens and concatenating those to form a (1, 1536) vector representation (4.18).

In this setup, we are using the entire memory as key & value. Since unlike query, key & value are not constrained by shapes in the Attention mechanism. We are not applying any Mean function to reduce the shapes. So, (4.10) will be replaced by (4.19).

Also, instead of pre-scaling the query by $(\sqrt{d_k})$ then scaling (qk^T) is again with learnable scaler (τ), this time we are scaling the whole (qk^T) by $(\sqrt{d_k}.\tau)$ which gave us similar numerical stability in attention scores. Hence, (4.14), (4.15) will be replaced by (4.20), (4.21) respectively in our new Model.

$$PooL = X_i[CLS] + Mean(X_i) \tag{4.18}$$

$$\text{Memory}_{\text{out}} = \text{LayerNorm}(Memory) \tag{4.19}$$

$$q = W_q(PE(\text{Memory}_{\text{out}}) \tag{4.20}$$

$$A = \text{softmax}\left(\frac{qk^T}{\sqrt{d_k}.\tau}\right) \tag{4.21}$$

Chapter 5

Result

Evaluation plays a crucial role in terms of understanding the performance, accuracy and effectiveness of a model. It helps to determine if the model is meeting its requirements or how far away it is from the goal and also helps to identify the scopes of improvement. Moreover, it allows us to compare different models and reliability.

5.1 Commonly used evaluation methods

There exists quite a few approaches to evaluate a model, it can be qualitative like feedback, opinions or subjective measures or it can be quantitative using different metrics or numeric representation for accuracy and error rates. For evaluating our model we will mostly focus on a quantitative approach, there are a number of metrics which can be used for a violence detection model like ours. The most effective and widely used are:

1. Accuracy
2. Precision
3. Recall
4. F1-Score (F-Measure)
5. Confusion matrix
6. mAP
7. K - Fold Cross Validation ()

Accuracy: Accuracy is a metric which determines how often the model provides the right predictions or it can be described as the ratio of correctly predicted outcomes to the total outcomes. It works great when the dataset is balanced but when the dataset is not balanced this is less effective.

It is calculated by:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5.1}$$

Where,

TP - True Positive

TN - True Negative

FP - False Positive

FN - False Negative

Precision: Precision is another metric that determines the accuracy of positive instances out of all the positive instances including True Positives(TP) and False Positives(FP).

Precision for each class i is:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{5.2}$$

Weighted-averaged precision: This is required for imbalanced datasets.

$$\text{Precision}_{weighted} = \sum_{i=1}^{n} w_i \times \text{Precision}_i \tag{5.3}$$

where w_i is the proportion of instances in class i.

Recall: Recall is the frequency with which a machine learning model correctly identifies true positives, among all of the actual positive samples in the dataset.

Recall for each class i is:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{5.4}$$

Weighted-averaged recall: This is required for imbalanced datasets.

$$\text{Recall}_{weighted} = \sum_{i=1}^{n} w_i \times \text{Recall}_i \tag{5.5}$$

F1-score: The F1 score, often called the F-measure, is the harmonic mean of the accuracy and recall of a classification model. Given that both measures contribute equally to the score, the **F1 measure** appropriately captures the reliability of a model. In cases where the dataset is unbalanced, it is far more effective.

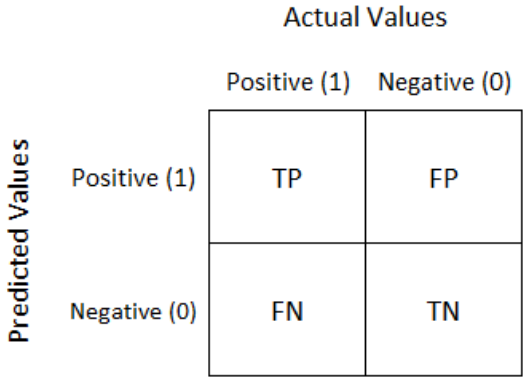
F1-score for each class i is:

$$F1_i = \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{5.6}$$

Weighted-averaged F1-score: This is required for imbalanced datasets.

$$F1_{weighted} = \sum_{i=1}^{n} w_i \times F1_i \tag{5.7}$$

Confusion Matrix: Confusion Matrix is a matrix that illustrates the performance of a machine learning model on a set of test data. Based on the model's predictions, it shows the proportion of accurate and wrong occurrences.



Mean Average Precision: Mean Average Precision() is most commonly used in detection or multi-labels classification models. It works by calculating the average of precision at different recall levels. It provides a robust measurement across different numbers of thresholds.

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_{\text{interp}}(R_n) \tag{5.8}$$

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^{n} \text{AP}_i \tag{5.9}$$

K-Fold Cross Validation: Through continually dividing the dataset into distinct training and testing sets, this strategy lowers variance by making sure that every data point is used for both training and validation. Additionally, by repeatedly training and verifying the model on various subsets of the data, it helps reduce over-fitting. Moreover, hyper-parameter tuning and model selection both frequently employ this strategy.

$$cv_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i \tag{5.10}$$

5.2 Best suited evaluation methods

According to our dataset description, We are using accuracy, precision, recall, and F1-Score to evaluate the results of our model. Also, we are using a confusion matrix to evaluate the model visually so that we can understand the total counts of False Positive, False Negative and True Positive, True Negative by using this we can find the area of improvement of our model.
Here, **Accuracy** represents the overall measure of our model's performance; it shows that from all of the predictions, how many of our predictions are correct. We have used accuracy to evaluate and easily understand our model's performance. Also, **Precision** represents the correctness of the positive prediction of our model. We

are using that method in our research so that we can ensure that our model minimizes the false positives and we can easily understand how reliable the positive predictions of our model are. Also, this method works fine for multiclass classification also **Recall** represents the completeness of the positive predictions of our model. We are using this method in our research to ensure that the model minimizes false negatives and captures as many true positive instances as possible. Recall helps us understand how well our model is at identifying all the actual positive cases in the dataset. In the context of multiclass classification, this metric is crucial to measure the model's ability to detect and correctly classify each class without missing important instances. The **F1 Score** represents the harmonic mean of Precision and Recall. We are using this method in our research so that we can balance the trade-offs of the previously used methods (**Precision** and **Recall**). This is a great method to assess our model's performance for both false positives and negatives. This method works fine for multiclass classification. Finally, the **Confusion Matrix** represents the overall performance of our model. We are using this method in our research so that we can easily understand the correct predictions and the wrong predictions for both cases, like false positive and true negative. It is a great tool to understand the overall model's accuracy and which class is causing more problems to the accuracy of our model.

5.3 Training Evaluation

Training evaluation has a significant amount of importance, it helps to address most of the common problems during the training phase including overfitting, underfitting, poor convergence and others. First of all it provides insight into areas that need further improvements and figuring out problems for which the model is not converging. Additionally, helps guide hyperparameter tuning to improve the performance of the model. We have used accuracy, loss and f1-score as training metrics which we monitored via Tensorbaord which gives a web app where you can monitor the states of the model and makes it easier to decide if the model is converging or not and if it is overfitting or underfitting. Figure **??** and Figure **??** shows that the accuracy and F1-Score is increasing with each epoch suggesting the model is getting better at generalizing. Figure **??** shows that loss is decreasing as the model is converging. Again, During the 7th epoch we found the best training and validation result with accuracy of 96.21%.

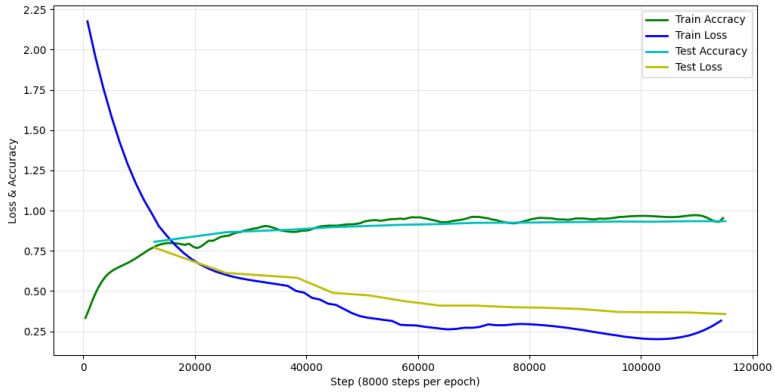


Figure 5.1: Training and testing evaluation of DMFA-Net v1.0

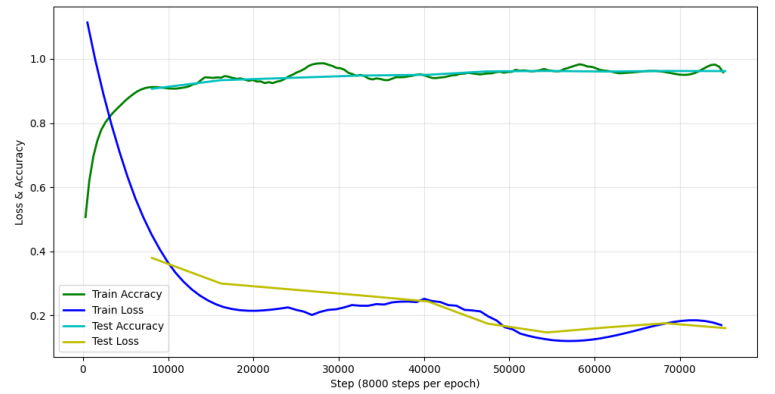


Figure 5.2: Training and testing evaluation of DMFA-Net v1.1

5.4 Test Evaluation

Assessing model performance in Machine Learning research is crucial for determining its ability to generalize to new data, which helps avoid overfitting. This process supports fair comparison with established methods, ensuring reproducibility and showcasing robustness. Employing a test dataset provides dependable benchmarks that render the results credible and significant.

Metric	DMFA-NetV1	DMFA-NetV1.1
Accuracy	92.65	96.21
Precision(weighted)	92.23	96.14
Recall(weighted)	92.65	96.21
F1-score(weighted)	92.01	96.08

Table 5.1: Test results multiclass classification in percentage

Overall metrics in table (5.1) show a promising result for the model making it reliable for violence detection. The results demonstrate a highly effective violence

detection model, achieving 96.21% accuracy with strong precision and recall. The F1-score of 96.08% indicates a well-balanced performance.

DMFA-Net v1.0 has an accuracy score of 92.65% on the other hand DMFA-Net v1.1 has higher accuracy score of 96.21% where DMFA-Net v1.0 uses a simple Avg Pool and concatenation (4.18) to form an embedding vector from 257 token vectors, DMFA-Net v1.1 uses a sophisticated approach leveraging the mechanism of Linear Pooling (4.1).
Also, DMFA-Net v1.0 uses the full key & value memory to generate attention output, which uses more GPU memory and computational power when the memory is full. On the other hand DMFA-Net v1.1 summarizes the whole query memory using average pooling on consequent group of frames and converts the query as a static summary of the entire scene. Which reduces the memory usage in the Cross-Attention-Fusion layers while improving the accuracy score.

We have trained both model for 10 epochs with 10e−7 learning rate for DMFA-Net v1.0 and 1e-6 learning rate for DMFA-Net v1.1. We have got the best result for DMFA-Net v1.1 at the 7th epoch and for DMFA-Net v1.0 at the 10th epoch. So we can hope for better scores for DMFA-Net v1.0 also if we give it training time and optimize the pooling layer.

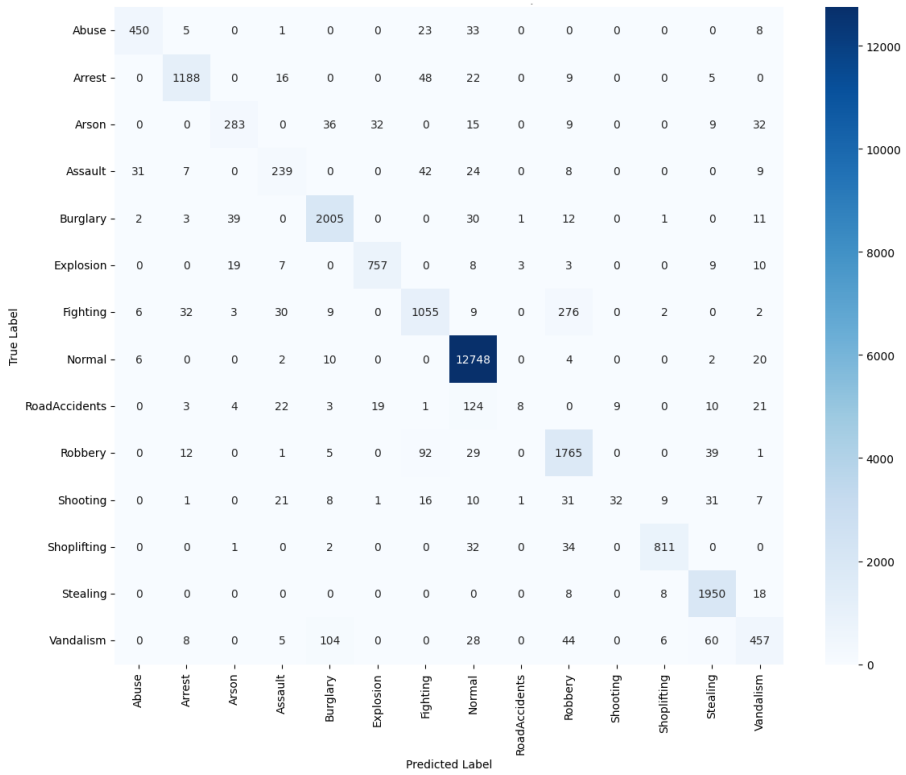


Figure 5.3: Confusion matrix of testing data for DMFA-NetV1.

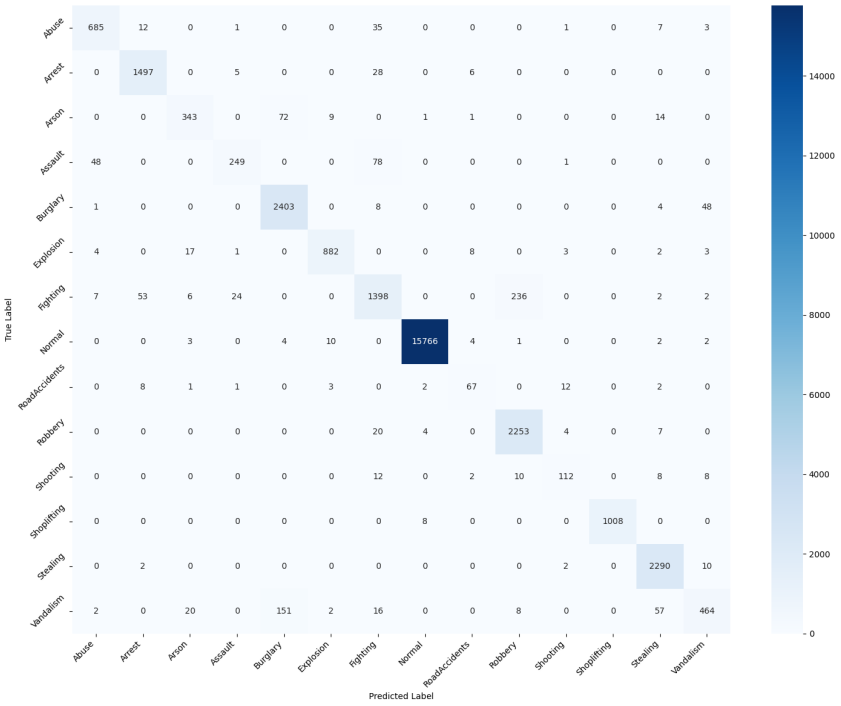


Figure 5.4: Confusion matrix of testing data for DMFA-NetV1.1

Class	DMFA-NetV1	DMFA-NetV1.1
Normal	99.66	99.84
Stealing	98.29	99.39
Shoplifting	92.16	99.21
Robbery	90.79	98.47
Burglary	95.29	97.52
Arrest	92.24	97.46
Explosion	92.77	95.87
Abuse	86.54	92.07
Fighting	74.09	80.9
Arson	68.03	77.95
Shooting	19.05	73.68
RoadAccident	03.57	69.79
Assault	66.39	66.22
Vandalism	64.19	64.44

Table 5.2: Models Accuracy for Different Classes in percentage

These results in table 5.2 highlight the strong performance of the model in distinguishing major anomaly classes with high accuracy. 99.84% accuracy achieved for normal vs. abnormal classes. However, certain classes, such as Arson, Shooting, Assault, and Vandalism, exhibit lower accuracy, indicating areas where misclassification is more common.

Our proposed models have performed well on the UCF-Crime dataset with higher accuracy, precision, recall and f-1 scores. However, confusion matrices in figure

5.3 and figure 5.4 for v1.0 and v1.1, shows that classes such as Arson, Burglary, Assult, Shooting has bean misclassified more than other classes such as Normal, Stealing, Shoplifting. These errors suggest that there is feature overlap among anomaly classes. Furthermore, class imbalance is a notable challenge in this context. The UCF-Crime dataset, the largest available collection of CCTV videos, is not well balanced. For instance, the dataset includes 1311 Burglary and 1239 Robbery videos vs 93 Shooting and 242 Arson videos. Along with, each video consists of 192 frames, with frames processed in segments of 24. The first two segments are often misclassified due to visual similarities with other classes. For instance, an arson video may at first look like a burglary, while a shooting could initially appear as a robbery or a fight due to similar confrontational beginnings. Similarly, assault and fighting exhibit similar physical actions, complicating their differentiation. Such early similarities lead the model to possibly mislabel these events until the complete context is revealed.

5.5 Real-World Implementation

Today we can see that different types of violence and abnormal activity are happening all around public places. The rate of violent activity is increasing day by day, which is not controllable manually by using human laboured process, especially when multiple acts of violence occur at the same places. To solve this problem, we have developed a Transformer-based violence detection model that is trained on the UCF crime dataset. Our computer vision model is trained on almost 3 million frames of multiple classes that can detect different types of violent activity based on 13 different classes.

We can implement our model with CCTV cameras in different places based on the location's needs. Our both models are trained on 13 different violent classes, like explosion, shooting, arson, stealing, and others. After testing and validating our model's prediction, we can see that our model **DMFA Net v1.1** performed better in some specific classes compared to others. The classes are Stealing (99.39%), Shoplifting (99.21%), Robbery (98.47%), Burglary (97.52%), Arrest (97.46%), Explosion (95.87%), and Abuse (92.07%). It can be highly effective in different places where this type of violent activity is common. There are some areas where we are hoping that our model will be highly beneficial to ensure the safety and security of the people.

Shopping Mall: Our model has some violent activity classes that will be best suited to places like shopping malls, where a lot of people are available and violent activity can occur based on the type of location As we can see, in our model's results, some violent activity classes like Stealing (99.39%), Shoplifting (99.21%), Robbery (98.47%), Burglary (98.47%) got better results. We think it will be more effective to use our model in places like shopping malls. Our model can detect this violent activity and ensure the safety and security of places like shopping malls.

University Campus: It is very common to see some violent activity in places like universities or college campuses, with ragging or abusing becoming a serious problem that should be controlled urgently. To solve this types of problem we have

trained our model with some violent activity classes like Abuse, Arrest and others. Also, we got good accuracy in this violent activity classes: they are Arrest (97.46%) and Abuse (92.07%). As our model is good at detecting these types of problems, we hope, by using our model, places like universities and colleges will be safer compared to the previous times.

Political Assembly and Concerts: As we know, places like political assemblies and concerts are the places where a lot of people stay at the same place; if any violent activity happens, it will be very difficult to detect and solve it by manual process, so solve this problem, we have trained our model on Arrest, Explosion and Abuse classes, and we got accurate results in these classes like Arrest (97.46%), Explosion (95.87%) and Abuse (92.07%) so we think that it is more appropriate to use our model at this type of place.

As we have seen previously, this model can be used to detect different violent activities in real-time with multiple streams at a time, with an impressive response rate, which will ensure the safety and security of public places. The variance of the different violent classes made our model more appropriate for multiple places. We are hoping that the earth will become safer with our work.

5.6 Model Inference

How fast is our model or how many streams can we check in real-time? For that, the performance of our violence detection model for multi-stream inference has been checked through an experimental setup. The experimental setup we used for our transformer-based model is **NVIDIA RTX 3070 8GB vRAM, 12core, 32GB RAM**. The input data consisted of videos with a frame rate of 30 FPS. To evaluate the system, we conducted tests with varying numbers of simultaneous streams: 1, 2, 4, and 8. We recorded metrics such as average FPS, total system throughput, and the average processing time for each test shown in the table 5.3.

Below are the results of the experiment for various stream counts:

Number of Streams	Average Stream FPS	Total Throughput (FPS)	Average Latency (ms)
1	179.13	179.13	131.54
2	100.205	196.97	236.35
4	50.865	198.27	468.83
8	25.46	197.66	939.18

Table 5.3: Performance metrics for varying numbers of streams.

Our model can handle up to 8 streams while maintaining 25 FPS in real-time. Though our multi-stream inference is yet to be validated through practical demonstrations, these results showcase the potential of our violence detection model.

As mentioned earlier our model can take up to 8 streams on a decent setup with a mid-range GPU we surely can say that it will be more cost-effective than manual detection. As we need to monitor this 24/7 we will need multiple people monitoring this simultaneously which is quite expensive but when we automate the detection process we do not need people to monitor this thoroughly. Moreover, one man can only focus on one feed only but this way we can monitor multiple feeds simultaneously thus outperforming humans. Though the first-time installation cost can be quite high for wide-area coverage but it does not require multiple people to monitor it proving it to be more cost-effective in the long run.

5.7 Comparison

Author/(s) Year, Ref	Deep Learning model used	Dataset	Number of classes	Accuracy	Precision	Recall	F1-Score
Ullah et all.,2021[10]	MobileNetV2 residual attention based LSTM	UCF Crime	5	78	87	78	81
Qasim and Verdu, 2023, [18]	ResNet50 + SRU	UCF Crime	2	91.63	91.71	-	91.33
			14	91.25	91.54	-	91.94
Pathak et all.,2024[22]	Slow Fast	UCF Crime	2	99	96.5	99	97.5
Proposed Models	**DMFA Net v1.1**	**UCF Crime**	**14**	**96.21**	**96.14**	**96.21**	**96.08**
	DMFA Net v1.1		**2**	**99.87**	**99.90**	**99.84**	**99.87**
	DMFA Net v1		**14**	**92.65**	**92.23**	**92.65**	**92.01**
	DMFA Net v1		**2**	**98.29**	**96.80**	**99.60**	**98.18**

Table 5.4: Comparison between models in terms of Accuracy, Precision, Recall and F1 in percentage

We have compared our proposed model with several other well-defined and state of the art models shown in table 5.4. It is important to compare the models with other models because it allows us to determine the best-fit model and find out the potential problem and requirements. Here we can see several models with different values for the metrics, some models identified two classes only and some did with all the 14 classes. The models with two classes tend to give better results than the model with 14 classes because in terms of identifying two classes the dataset is more balanced. Moreover, the Normal and Anomaly classes are mutually exclusive means that the normal class does not include any part where it identifies one anomaly class as another unlike 14 class classification we see that some classes overlap with one another.

Chapter 6

Conclusion

6.1 Future Improvement

In the future, we can improve our model across several dimensions, each of which focuses on a distinct performance improvement component. We can do cross-dataset validation to unlock wider applicability. It will be necessary to test the model's performance on a variety of datasets, potentially covering various domains such as violence detection, anomaly detection, or action recognition. Understanding how well the model generalizes across different video qualities, lengths, and contexts will be key to refining its robustness.

Alternative pooling mechanisms will be one area of focus, where different strategies such as attention-based pooling, hierarchical pooling, or adaptive learnable pooling may be used to improve feature representation. These methods may make it easier to capture global and fine-grained visual patterns.

Another potential direction is optimizing model fusion strategies. Future studies may look into ways to merge our two architectures. To improve accuracy and resilience, combining different pooling mechanisms, experimenting with hybrid attention mechanisms, or even developing ensemble approaches could leverage both models.

In the future, we could also change the backbone of our architecture, DINOv2. Even though DINOv2 has proven to be successful, future research may involve evaluating newer vision transformers, experimenting with CNN-based architectures, or testing hybrid models that integrate both transformers and CNNs. This research will assist in identifying the most accurate and effective foundation for our work.

Our goal is to improve the model's effectiveness and flexibility in a variety of settings by consistently improving our methodology.

6.2 Conclusion

In conclusion, this study endeavors to harness the potential of transformer-based deep learning models to develop a sophisticated system capable of classifying anomalies in CCTV videos. The necessity of violence detection in public places poses a

critical challenge world-wide. To tackle the challenge, researchers have developed various models leveraging CNN, LSTM, and other techniques, highlighting the crucial role of machine learning and computer vision methods in achieving effective solutions. But, existing models primarily focus on binary classification (violence vs non-violence), limiting their applicability in complex real-world scenarios. In contrast, our proposed models demonstrates better accuracy, achieving 96.21% overall accuracy. Our evaluation metrics further outperform existing methods with 96.14& precision, 96.21% recall, 96.08% F-1 score, ensuring robust and reliable performance. Additionally, our model is designed to handle multiple video streams simultaneously, making it significantly more practical for real-world surveillance applications. This capability enhances real-time monitoring and improves response times in critical situations, making it a more effective solution for large-scale deployments.

Bibliography

[1] T. Deb, A. Arman, and A. Firoze, "Machine cognition of violence in videos using novel outlier-resistant vlad," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, IEEE, 2018, pp. 989–994.

[2] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[3] M. H. Mohamed Elesawy and M. A. E. Massih, *Real-life violence situations dataset*, Accessed: 2025-02-06, 2019. [Online]. Available: https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset.

[4] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, *Augmix: A simple data processing method to improve robustness and uncertainty*, 2020. arXiv: 1912.02781 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1912.02781.

[5] P. Wu, J. Liu, Y. Shi, *et al.*, "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, Springer, 2020, pp. 322–339.

[6] T. S. Apon, M. I. Chowdhury, M. Z. Reza, A. Datta, S. T. Hasan, and M. G. R. Alam, "Real time action recognition from video footage," in *2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, IEEE, 2021. DOI: 10.1109/STI53101.2021.9732601.

[7] 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)*, 2021, pp. 4183–4190. DOI: 10.1109/ICPR48806.2021.9412502.

[8] 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, pp. 4183–4190.

[9] D. S. S, S. Govindraj, and S. Omkar, "Real-time violence activity detection using deep neural networks in a cctv camera," in *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 2021, pp. 1–6. DOI: 10.1109/CONECCT52877.2021.9622739.

[10] W. Ullah, A. Ullah, T. Hussain, Z. A. Khan, and S. W. Baik, "An efficient anomaly recognition framework using an attention residual lstm in surveillance videos," *Sensors*, vol. 21, no. 8, p. 2811, 2021.

[11] S. D. J, M. T. Ahammed, U. M. Boppana, *et al.*, "Real-time based violence detection from cctv camera using machine learning method," in *2022 International Conference on Industry 4.0 Technology (I4Tech)*, 2022, pp. 1–6. DOI: 10.1109/I4Tech55392.2022.9952805.

[12] T. Kayani *et al.*, "Real-time violence detection using deep learning techniques," in *2022 3rd International Conference on Innovations in Computer Science & Software Engineering (ICONICS)*, IEEE, 2022, pp. 1–8.

[13] T. Marteau, D. Sodoyer, S. Ambellouis, and S. Afanou, "Level fusion analysis of recurrent audio and video neural network for violence detection in railway," in *2022 30th European Signal Processing Conference (EUSIPCO)*, IEEE, 2022, pp. 563–567.

[14] L. A. Siddique, R. Junhai, T. Reza, S. S. Khan, and T. Rahman, "Analysis of real-time hostile activitiy detection from spatiotemporal features using time distributed deep cnns, rnns and attention-based mechanisms," in *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, IEEE, 2022, pp. 1–6.

[15] S. Singh, S. Dewangan, G. S. Krishna, V. Tyagi, S. Reddy, and P. R. Medi, "Video vision transformers for violence detection," *arXiv preprint arXiv:2209.03561*, 2022.

[16] K. R. Talha, K. Bandapadya, and M. M. Khan, "Violence detection using computer vision approaches," in *2022 IEEE World AI IoT Congress (AIIoT)*, IEEE, 2022, pp. 544–550.

[17] X. Zheng, Y. Zhang, Y. Zheng, F. Luo, and X. Lu, "Abnormal event detection by a weakly supervised temporal attention network," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 3, pp. 419–431, 2022.

[18] M. Qasim and E. Verdu, "Video anomaly detection system using deep convolutional and recurrent models," *Results in Engineering*, vol. 18, p. 101 026, 2023.

[19] İ. Üstek, J. Desai, I. L. Torrecillas, *et al.*, "Two-stage violence detection using vitpose and classification models at smart airports," in *2023 IEEE Smart World Congress (SWC)*, 2023, pp. 797–802. DOI: 10.1109/SWC57546.2023.10448548.

[20] M. Korban, P. Youngs, and S. T. Acton, "A semantic and motion-aware spatiotemporal transformer network for action detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[21] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, *Dinov2: Learning robust visual features without supervision*, 2024. arXiv: 2304.07193 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2304.07193.

[22] A. Pathak, R. Jayaswal, and S. Mahajan, "Exploring deep learning techniques for abnormality detection: A comparative analysis on ucf crime dataset.," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 3, pp. 392–401, Mar. 2024. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/5263.

[23] M. Qaraqe, A. Elzein, E. Basaran, *et al.*, "Publicvision: A secure smart surveillance system for crowd behavior recognition," *IEEE Access*, vol. 12, pp. 26 474–26 491, 2024.

[24] D. C. Senadeera, X. Yang, D. Kollias, and G. Slabaugh, "Cue-net: Violence detection video analytics with spatial cropping, enhanced uniformerv2 and modified efficient additive attention," *arXiv preprint arXiv:2404.18952v1*, 2024.

[25] S. Sun and X. Gong, "Multi-scale bottleneck transformer for weakly supervised multimodal violence detection," *arXiv preprint arXiv:2405.05130*, 2024.