Dissertation on

"Multi-Event Anomaly Detection for Enhanced Video Surveillance"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE21CS461A – Capstone Project Phase II

Submitted by:

Amara Sai Prasad	**PES1UG21CS001**
Nidhi P G	**PES1UG21CS380**
Nikita Suresh	**PES1UG21CS386**
Ria R Kulkarni	**PES1UG21CS487**

Under the guidance of

Dr. Ramamoorthy Srinath
Professor
PES University

August – December 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Multi-Event Anomaly Detection for Enhanced Video Surveillance'

is a bonafide work carried out by

Amara Sai Prasad	**PES1UG21CS001**
Nidhi P G	**PES1UG21CS380**
Nikita Suresh	**PES1UG21CS386**
Ria R Kulkarni	**PES1UG21CS487**

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE21CS461A) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period August – Dec, 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature	Signature	Signature
Dr. Ramamoorthy Srinath	Dr. Mamatha H R	Dr. B K Keshavan
Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.**_____
2. _____	**2.**_____

DECLARATION

We hereby declare that the Capstone Project Phase II entitled **"Multi-Event Anomaly Detection for Enhanced Video Surveillance"** has been carried out by us under the guidance of Dr. Ramamoorthy Srinath, Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August – Dec, 2024. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG21CS001	**Amara Sai Prasad**
PES1UG21CS380	**Nidhi P G**
PES1UG21CS386	**Nikita Suresh**
PES1UG21CS487	**Ria R Kulkarni**

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Ramamoorthy Srinath, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance and encouragement throughout the development of this UE21CS461A - Capstone Project Phase II.

We are grateful to the project coordinator, Dr. Priyanka H, for organizing, managing and helping with the entire process.

We take this opportunity to thank Dr. Mamatha H R, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to us various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

ABSTRACT

Video surveillance systems are indispensable tools for maintaining security and safety across various domains, including airports, transportation hubs, commercial establishments and public spaces. However, the effectiveness of traditional surveillance systems is often hindered by their limited ability to detect and respond to diverse security threats and anomalies effectively. Common anomalies encountered in video surveillance include abandoned or unidentified objects, unnecessary object placement, vandalism, violence like fighting, arguments, assault and so on. Addressing these challenges requires advanced anomaly detection techniques capable of identifying multiple events simultaneously, enhancing the overall efficacy of video surveillance systems.

In recent years, the advent of large language models (LLMs) and deep learning methodologies has provided new avenues for improving video surveillance capabilities. By leveraging the semantic understanding encoded in LLMs and integrating it with computer vision techniques, we aim to develop more robust and adaptable anomaly detection systems for enhanced video surveillance. This project proposes a multi-event anomaly detection framework that harnesses the power of LLMs to detect various security threats and anomalies, including abandoned objects, unnecessary object placement, vandalism, threat and violence detection simultaneously. Through the integration of LLM-based semantic understanding with computer vision algorithms, the proposed framework seeks to enhance the interpretability and accuracy of anomaly detection in video surveillance scenarios.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Video surveillance systems are indispensable tools for maintaining security and safety across various domains, including airports, transportation hubs, commercial establishments, and public spaces. However, the effectiveness of traditional surveillance systems is often hindered by their limited ability to detect and respond to diverse security threats and anomalies effectively. Common anomalies encountered in video surveillance include abandoned or unidentified objects, unnecessary object placement, vandalism, individuals wearing disguises or unusual clothing, prolonged one-on-one following situations (tailing), and crowd monitoring.

Addressing these challenges requires advanced anomaly detection techniques capable of identifying multiple events simultaneously, enhancing the overall efficacy of video surveillance systems. In recent years, the advent of large language models (LLMs) and deep learning methodologies has provided new avenues for improving video surveillance capabilities. By leveraging the semantic understanding encoded in LLMs and integrating it with computer vision techniques, researchers aim to develop more robust and adaptable anomaly detection systems for enhanced video surveillance. This project proposes a multi-event anomaly detection framework that harnesses the power of LLMs to detect various security threats and anomalies, including abandoned objects, unnecessary object placement, vandalism and threats like violence simultaneously. Through the integration of LLM-based semantic understanding with computer vision algorithms, the proposed framework seeks to enhance the interpretability and accuracy of anomaly detection in video surveillance scenarios.

By exploring the intersection of LLMs, computer vision, and anomaly detection, this project endeavors to advance the state-of-the-art in video surveillance technology and contribute to the development of more intelligent and proactive security solutions. Through empirical evaluation and validation, the proposed framework aims to demonstrate its effectiveness in addressing real-world security challenges and improving the overall safety and security of public spaces and critical infrastructure.

CHAPTER II

PROBLEM STATEMENT

Our project aims to develop a multi-event anomaly detection framework for video surveillance systems, leveraging large language models (LLMs) and computer vision techniques. This framework will enable the simultaneous detection of various security threats and anomalies, including abandoned objects, unnecessary object placement, vandalism and threats like violence detection. By integrating LLM-based semantic understanding with computer vision algorithms, the framework seeks to enhance the interpretability and accuracy of anomaly detection, thereby improving the overall safety and security of public spaces and critical infrastructure.

CHAPTER III

LITERATURE REVIEW

3.1 Enhancing Video Transformers for Action Understanding with VLM-aided Training

The paper addresses the limitations of Vision Transformers (ViTs) in generalizing across different datasets for video action understanding. While ViTs excel at extracting spatio-temporal embeddings, their performance can deteriorate when tested on datasets that emphasize different aspects, such as object usage or environmental context. To overcome this, the authors propose the Four-Tiered Prompts (FTP) framework, which leverages the strengths of Visual Language Models (VLMs) to enhance the visual encodings produced by ViTs. The FTP framework focuses on four specific aspects of human actions: action category, action components, action description, and context information.

The FTP framework consists of a ViT architecture augmented with four feature processors that are trained to align visual embeddings with textual descriptions generated by a VLM. The training process is divided into two stages. In the first stage, the feature processors are trained using supervision from a VLM and a text encoder, focusing on visual-text alignment. The second stage involves fine-tuning the integrated model for action classification. The VLM, text encoder, and visual encoder are kept frozen during this process, which significantly reduces the computational burden of training.

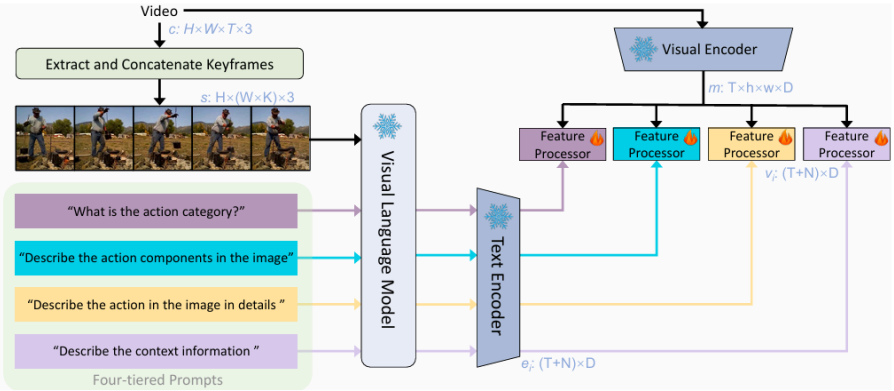

Fig. 1: Training Stage for FTP Framework

The proposed FTP framework is evaluated on several benchmark datasets, including Kinetics-400, Something-Something V2, HMDB51, UCF-101, and AVA V2.2. The evaluation aims to assess the framework's performance in terms of action recognition accuracy and computational efficiency compared to existing state-of-the-art methods. The authors emphasize the importance of using diverse datasets to validate the generalization capabilities of their approach.

The FTP framework achieves impressive results, reporting a top-1 accuracy of 93.8% on Kinetics-400 and 83.4% on Something-Something V2, surpassing the previous state-of-the-art method, VideoMAEv2, by 2.8% and 2.6%, respectively. The experiments demonstrate that the FTP framework consistently outperforms other methods across all evaluated datasets while maintaining a lower computational cost. This indicates that the integration of VLMs into the training process effectively enhances the performance of ViTs in action understanding tasks.

The authors conclude that the FTP framework successfully enhances the generalization ability of ViTs for video action understanding by incorporating semantic information from VLMs. The approach not only achieves state-of-the-art performance on multiple benchmarks but also incurs minimal computational overhead during inference, as the VLMs are only utilized during training. The authors suggest future work could explore earlier integration of textual embeddings and a systematic analysis of the prompts used, which may further improve the framework's performance and applicability in various domains.

3.2 Video Anomaly Detection and Explanation via Large Language Models

The paper presents VAD-LLaMA, a novel approach to Video Anomaly Detection (VAD) that integrates Video-based Large Language Models (VLLMs) to enhance anomaly detection and provide textual explanations for detected anomalies. Traditional VAD methods often rely on manually set thresholds for anomaly detection, which can be non-intuitive and vary with different video content. The authors aim to address these limitations by incorporating a Long-Term Context (LTC) module that captures both normal and abnormal contexts over time, thereby improving the model's ability to detect and explain anomalies.

The VAD-LLaMA framework consists of three main phases. The first phase involves training a baseline VADor to generate anomaly scores for video clips. In the second phase, the VADor is co-trained with the LTC module to enhance video representation by integrating long-term contextual information. The final phase fine-tunes the Video-LLaMA model using a dataset annotated with anomaly scores, where only the projection layer is trained to align the feature

distribution of VADor with the VLLM. This approach minimizes the need for extensive training data and reduces annotation costs.

The performance of VAD-LLaMA was evaluated on standard benchmarks, specifically UCF-Crime and TAD datasets. The authors conducted extensive ablation studies to assess the impact of the LTC module and short-term historical information on the model's performance. The Area Under the Curve (AUC) metrics were used to quantify the effectiveness of the model in distinguishing between normal and anomalous video clips.

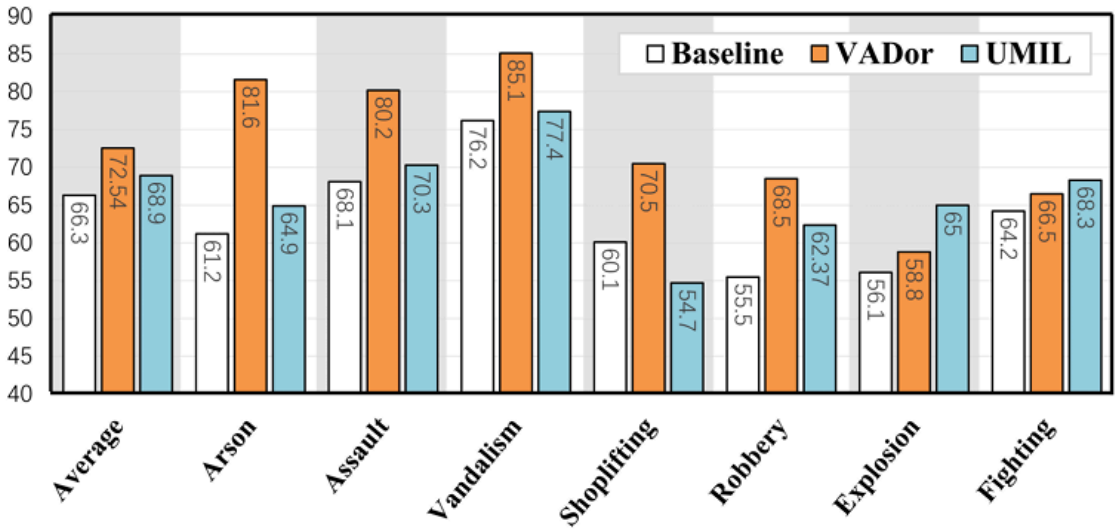

Fig. 2: Class-wise AUC on UCF-Crime

The results demonstrated that VAD-LLaMA significantly outperformed existing methods. For instance, the AUC on the UCF-Crime dataset improved from 85.90% to 88.39% with the LTC module, while on the TAD dataset, it increased from 85.20% to 91.77%. The model also achieved a higher AUC for anomaly localization, with an increase of 3.21% compared to the baseline. The integration of long-range context modeling proved critical, as it enhanced the model's ability to analyze complex anomalies that require comprehensive context understanding.

The study concludes that VAD-LLaMA effectively addresses the challenges of traditional VAD methods by eliminating the reliance on arbitrary thresholds and providing detailed explanations for detected anomalies. The incorporation of the LTC module and a three-phase training approach led to substantial improvements in performance, as evidenced by the high AUC scores on benchmark

datasets. The findings suggest that VAD-LLaMA represents a significant advancement in the field of video anomaly detection, offering a robust solution for real-world applications.

3.3 Video Vision Transformers for Violence Detection

The paper presents a novel framework for violence detection in videos using Video Vision Transformers (ViViT). The motivation behind this research is the critical need for effective surveillance systems that can automatically identify violent incidents in real-time. The study focuses on two benchmark datasets: the Hockey Fight dataset, which contains 1,000 samples (500 violent and 500 non-violent clips), and the Violent Crowd dataset, comprising 246 samples (123 violent and 123 non-violent clips). The proposed method aims to achieve high accuracy in classifying video clips as violent or non-violent.

The methodology involves several key steps, starting with the pre-processing of video data, where videos are converted into frames and resized to maintain the aspect ratio. A total of 56 consecutive frames are extracted from each video, and various image augmentation techniques, such as Gaussian Blur and random rotation, are applied to enhance the training dataset. The ViViT framework is then implemented on these augmented frames to classify the videos. The model is fine-tuned with hyperparameters, including a batch size of 32, learning rate of 0.0001, and 100 epochs, to optimize performance.

The performance of the ViViT model is evaluated using metrics such as accuracy, precision, recall, and F1-score. The model's training and validation accuracies were reported as 98.73% and 98.46% for the crowd violence dataset, respectively. For the hockey fight dataset, the maximum training accuracy achieved was 96.57%, with a validation accuracy of 97.14%. The evaluation also includes a comparative analysis with state-of-the-art (SOTA) methods to assess the effectiveness of the proposed approach.

The results indicate that the proposed ViViT model outperforms previous SOTA methods in violence detection. For the hockey fight dataset, the model achieved a precision of 0.98, recall of 0.98, and an F1 score of 0.98. In the case of the violent crowd dataset, the precision, recall, and F1 score were all reported at 0.99. The overall macro and weighted averages for both datasets were also high, indicating robust performance across different classes. The proposed method demonstrated a significant improvement over traditional approaches, with accuracy values of 97.14% for hockey fights and 98.46% for violent crowds.

The study concludes that the ViViT framework effectively enhances violence detection in video clips, achieving high accuracy and computational efficiency compared to CNN-based approaches.

The incorporation of data augmentation techniques and the transformer architecture allows the model to learn complex spatio-temporal patterns, making it suitable for real-world applications in surveillance. The results suggest that the proposed method is generalizable and can be applied to various scenarios involving violence detection, paving the way for future research in this domain.

3.4 Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey

The paper presents a comprehensive survey on deep learning-based anomaly detection methods in video surveillance systems. It emphasizes the increasing importance of these systems in identifying unusual activities in various environments, such as public spaces and offices. The authors highlight the challenges faced in traditional anomaly detection methods, including the need for extensive feature engineering and the limitations of handcrafted features. They propose that deep learning techniques can significantly enhance the detection capabilities by automatically learning relevant features from raw data.

The authors conducted a systematic review of existing literature on deep learning models applied to anomaly detection in video surveillance. They categorized the methods into various approaches, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid models that combine both. The survey also discusses the preprocessing steps involved, such as data normalization and augmentation, which are crucial for improving model performance. The authors detail the datasets used for training and testing, including the CASIA Action Database and other publicly available datasets, which contain thousands of labeled video samples.

The evaluation of the discussed models is based on several performance metrics, including accuracy, precision, recall, and F1-score. The authors summarize the results from various studies, indicating that deep learning models generally outperform traditional methods. For instance, some CNN-based models achieved accuracy rates exceeding 90% in detecting anomalies in video streams. The paper also highlights the importance of using transfer learning techniques to improve model performance, especially when dealing with small-scale datasets.

The survey reveals that deep learning approaches have led to significant advancements in anomaly detection. For example, the use of attention mechanisms and social force maps has improved the ability to learn motion representations, resulting in better detection rates. The authors report that certain models have achieved F1-scores above 0.85, demonstrating their effectiveness in real-world applications. Additionally, the paper notes that integrating multimodal data sources can further enhance detection accuracy, providing a more comprehensive understanding of human activities.

In conclusion, the paper underscores the transformative impact of deep learning on anomaly detection in video surveillance. The authors advocate for continued research in this area, particularly focusing on transfer learning and the extraction of rich semantic features from multimodal datasets. They emphasize the need for further exploration of physical interactions between humans and objects to improve model interpretability and performance. The survey serves as a valuable resource for researchers and practitioners aiming to develop more effective anomaly detection systems in various contexts.

3.5 Learning Video Representations from Large Language Models

LAVILA is a novel concept that is based on using large language models (LLMs) in making new kinds of video-language representations that come after the second video is narrated automatically. Our main approach is a three- stage procedure. A pre-trained GPT-2 model, in which the cross-attention modules are added in, is used in achieving our goal. The purpose of this NARRATOR is to use the available video-text pairs to develop fresh narration, using the videos as the factor. Then, it works in harmony with human-annotated (that is further narrated) video clips by synergizing with previously un-annotated portions.

In the second step, a separate REPHRASER text-to-text LLM like T5 rephrases the existing human narrations to expand the text data with varied descriptions. Finally, in the third step, the visual inputs and narrations from NARRATOR and REPHRASER are combined as video-text data to train a dual-encoder model using an InfoNCE loss. The dual-encoder comprises a video encoder such as TimeSformer and a text encoder. During training, batches consist of both human annotations, re-narrated or paraphrased, and un-annotated videos narrated by NARRATOR. This three-step process generates a diverse, dense, and well-aligned video-text data set, enabling LAVILA to acquire more robust video-language representations compared to previous approaches that solely rely on human annotations.

Method	Vis. Enc.	UCF-101	HMDB-51
MIL-NCE [44]	S3D	82.7	54.3
TAN [25]	S3D	83.2	56.7
Baseline (w/o LLM)	TSF-B	86.5	**59.4**
LAVILA	TSF-B	**87.4**	57.2
LAVILA	TSF-L	**88.1**	**61.5**

Table 1: LAVILA- Action Classification Performance

3.6 Attentional Feature Fusion

The paper presents a novel approach to feature fusion in deep neural networks, termed Attentional Feature Fusion (AFF). The authors identify existing challenges in feature integration, particularly issues related to scale and semantic inconsistency among input features. They propose a unified framework that incorporates attention mechanisms to enhance the quality of fused features across various network scenarios. The study emphasizes the need for sophisticated attention mechanisms to improve performance in tasks such as image classification and semantic segmentation.

Context-aware	Type	Formulation	Scenario & Reference	Example
None	Addition	$\mathbf{X} + \mathbf{Y}$	Short Skip [11, 12], Long Skip [24, 21]	ResNet, FPN
	Concatenation	$\mathbf{W_A X}_{:,i,j} + \mathbf{W_B Y}_{:,i,j}$	Same Layer [36], Long Skip [28, 15]	InceptionNet, U-Net
Partially	Refinement	$\mathbf{X} + \mathbf{G}(\mathbf{Y}) \otimes \mathbf{Y}$	Short Skip [14, 13, 44, 26]	SENet
	Modulation	$\mathbf{G}(\mathbf{Y}) \otimes \mathbf{X} + \mathbf{Y}$	Long Skip [18]	GAU
	Soft Selection	$\mathbf{G}(\mathbf{X}) \otimes \mathbf{X} + (\mathbf{1} - \mathbf{G}(\mathbf{X})) \otimes \mathbf{Y}$	Short Skip [34]	Highway Networks
Fully	Modulation	$\mathbf{G}(\mathbf{X}, \mathbf{Y}) \otimes \mathbf{X} + \mathbf{Y}$	Long Skip [46]	SA
	Soft Selection	$\mathbf{G}(\mathbf{X} + \mathbf{Y}) \otimes \mathbf{X} + (1 - \mathbf{G}(\mathbf{X} + \mathbf{Y})) \otimes \mathbf{Y}$	Same Layer [19, 48]	SKNet
		$\mathbf{M}(\mathbf{X} \uplus \mathbf{Y}) \otimes \mathbf{X} + (1 - \mathbf{M}(\mathbf{X} \uplus \mathbf{Y})) \otimes \mathbf{Y}$	Same Layer, Short Skip, Long Skip	***ours***

Table 2: Overview of Feature Fusion Strategies in Deep Neural Networks

The proposed AFF module integrates a multi-scale channel attention mechanism that combines local and global contextual information. The authors implement several feature integration strategies, including linear approaches (addition and concatenation) and nonlinear methods with attention mechanisms. The channel reduction ratios are set to 2 for most methods and 4 for the iterative attentional feature fusion (iAFF). The architecture is designed to maintain a consistent parameter budget while exploring different feature integration strategies through ablation studies.

The evaluation of the proposed methods is conducted on benchmark datasets, specifically CIFAR-100 and ImageNet for image classification, and a subset of the COCO dataset for semantic segmentation (StopSign). The authors perform extensive ablation studies to assess the impact of various feature integration types and contextual aggregation scales. They compare the performance of their methods against state-of-the-art networks, ensuring a fair evaluation by re-implementing existing approaches using their multi-scale channel attention mechanism.

The experimental results demonstrate that the proposed AFF outperforms traditional linear fusion methods, achieving better performance across all tested scenarios. For instance, the iAFF approach shows significant improvements, with accuracy gains of up to 2.5% on CIFAR-100 and 1.8% on ImageNet compared to baseline models. The multi-scale contextual aggregation strategy (Global + Local) consistently outperforms single-scale approaches, indicating the importance of incorporating diverse contextual information in feature fusion.

The study concludes that attentional feature fusion is a powerful technique for enhancing feature integration in deep neural networks. The proposed methods not only address the issues of scale and semantic inconsistency but also demonstrate superior performance with fewer parameters. The authors advocate for the adoption of more sophisticated attention mechanisms in feature fusion, as their results indicate that such approaches can lead to significant improvements in various computer vision tasks. The findings suggest that attention mechanisms hold great potential for advancing the capabilities of deep learning models.

3.7 Gated Multimodal Units For Information Fusion

The paper introduces a novel model aimed at enhancing multimodal learning through the use of gated neural networks. This model, called the Gated Multimodal Unit (GMU), is designed to function as an internal component within a neural network architecture, facilitating the integration of data from various modalities to produce an intermediate representation.

The GMU utilizes multiplicative gates to dynamically determine how much each input modality contributes to the overall output. This adaptive mechanism allows for a more nuanced representation of multimodal data compared to traditional fixed fusion strategies.

The GMU acts as an internal unit within a neural network architecture, focusing on generating intermediate representations that combine features from different modalities. It is independent from the final task.

The GMU architecture consists of input feature vectors, fate neurons and activation functions. Each modality contributes its own feature vector to the GMU. For each input modality, a gate neuron dynamically determines the extent to which that modality should contribute to the output, considering features from all modalities. The model employs a tanh activation function to encode internal representations derived from each modality.

The authors evaluated the GMU in a multilabel classification scenario for movie genre prediction, using both plot descriptions and visual posters as inputs. The results demonstrated that the GMU significantly outperformed single-modality approaches and other existing fusion techniques, including mixture-of-experts models, enhancing performance metrics like macro F-score.

CHAPTER IV

PROJECT REQUIREMENTS SPECIFICATION

4.1 Overview

4.1.1 Project Description

The system aims to provide a robust, end-to-end solution for detecting anomalies within video content. This system will leverage both visual and textual data by combining video feature extraction and caption generation. The objective is to classify any detected anomalies into predefined categories: "Violence," "Vandalism," "Abandoned Object," and "Normal." The system's functionality begins with user video input, followed by the generation of descriptive captions via the TwelveLabs SDK. Subsequently, the system extracts visual features using TimeSformer, an advanced video transformer model, while textual features are extracted using RoBERTa and refined using KeyBERT to focus on the most relevant phrases. An additional layer of semantic depth is added with LDA topic modeling, allowing for a robust representation of textual information. The combined features are processed through a multimodal fusion layer, offering either concatenation or gated multi-modal fusion as fusion strategies. This fused feature is mapped to a multimodal embedding space, where a neural network classifier is applied to predict category scores, ultimately determining if the input video is anomalous and, if so, identifying its type.

4.1.2 Objectives

The primary objective is to develop a responsive system capable of real-time detection and classification of anomalies in video inputs. The system's dual-modal approach of combining visual and textual features aims to improve accuracy and contextual understanding. The system should be adaptable to different fusion methodologies, and integration of advanced models like TimeSformer, RoBERTa and KeyBERT, ensures it remains at the forefront of anomaly detection research.

4.2 Functional Requirements

4.2.1 Input Handling

The system will accept video inputs directly from users, supporting common video formats such as MP4, AVI, and MOV. Each video will be processed to extract both visual and textual data, ensuring comprehensive feature representation. The video input size should be manageable within the system's computational limits, with the length constrained to a maximum of 10 minutes per video to allow for efficient processing.

4.2.2 Caption Generation

Upon receiving a video input, the system will generate captions using the TwelveLabs SDK. This step is essential to provide a textual representation of the visual content. The generated captions will include key details that are potentially relevant to anomaly detection, such as actions, objects, and context within the video.

4.2.3 Visual Feature Extraction

The visual data from the video will be processed using the TimeSformer module to extract meaningful spatiotemporal features. TimeSformer, a transformer-based model, is optimized for video processing and capable of capturing both spatial and temporal information. The extracted visual features will serve as a core representation of the video content, aiding in anomaly classification.

4.2.4 Text Feature Extraction and Topic Modeling

The system will process the generated captions to extract contextual text features. Using RoBERTa, a pre-trained language model, the system will obtain contextual embeddings, capturing nuanced textual data. KeyBERT will then refine these embeddings by selecting the most relevant phrases, ensuring the focus remains on significant terms related to anomalies. To further enrich the text features, LDA (Latent Dirichlet Allocation) will be applied to identify prevalent topics within the captions, enhancing the feature set for more accurate classification.

4.2.5 Multimodal Fusion

Both the extracted visual and textual features will be combined using a multimodal fusion layer. The system provides flexibility in fusion methodology by offering two approaches: concatenation and compact bilinear pooling. Concatenation is a straightforward approach that appends the feature vectors, while compact bilinear pooling leverages randomized count sketches and FFT to capture richer interactions between the modalities in a computationally efficient manner. An alternative approach called gated fusion was also experimented with; this employs multiplicative gates that dynamically evaluate the importance of various input features. This fused feature will be optimized to create a joint representation within a multimodal embedding space, ready for classification.

4.2.6 Classification and Anomaly Scoring

The fused features will be input to a neural network classifier, designed to predict a score for each anomaly category (Violence, Vandalism, Abandoned Object, and Normal). The network's final layer will include a scoring mechanism, using either softmax or sigmoid activation, depending on whether the classification is treated as multi-class or multi-label. The output will categorize the input as anomalous or normal, with anomalous cases further classified into the specific anomaly type.

4.3 Non-Functional Requirements

4.3.1 Performance and Scalability

The system should efficiently handle video inputs with a maximum length of 10 minutes, processing each video within a reasonable timeframe. Scalability considerations are essential to allow expansion of the model's capabilities, such as accommodating additional anomaly types or integrating more complex feature extraction methods.

4.3.2 Reliability and Robustness

The system is expected to deliver consistent, accurate predictions across a variety of input videos, ensuring robustness in anomaly classification. Error handling mechanisms will be in place to manage cases of corrupted inputs, unexpected data formats or model loading failures. Regular testing and model retraining will maintain reliability as new data becomes available.

4.3.3 Usability

The user interface should be intuitive, allowing users to upload video files easily and view classification results in a clear format. Instructions for interpreting the output, including explanations of anomaly categories, should be provided. Additionally, the system should require minimal setup, facilitating smooth deployment and usage across different devices.

4.3.4 Security and Privacy

The system must ensure the secure handling of video data, especially if applied in sensitive settings. Privacy protocols should be in place to prevent unauthorized access to user-uploaded content, including encryption of any stored video data or output files. An access-controlled setup will be necessary for any deployment involving multiple users.

4.3.5 Compliance and Ethical Considerations

The system must comply with data protection regulations such as GDPR, particularly if deployed in environments where personal data may be captured in videos. Ethical considerations include preventing misuse of the anomaly detection system and ensuring that false positives and negatives are minimized, especially in applications with potential security implications.

4.4 Design Constraints

4.4.1 Computational Efficiency

The system must handle video files up to 10 minutes long without excessive delay. To ensure this, the fusion and classification layers should be optimized for GPU acceleration to handle high-dimensional data from both video and text features.

4.4.2 Memory Limitations

Given the complexity of models like TimeSformer, RoBERTa, and compact bilinear pooling, memory usage must be carefully managed to prevent overload. Efficient loading and unloading of models and data must be prioritized to allow smooth execution on hardware with limited VRAM or RAM.

4.4.3 Fusion Flexibility

The system must support both concatenation and compact bilinear pooling in the fusion layer, allowing for interchangeable use depending on the context. This constraint necessitates modular coding practices, enabling easy switching between fusion methods.

4.4.4 Real-time Requirements

For applications that require live anomaly detection, the system must be capable of near-real-time processing. This necessitates optimizing each step, particularly the feature extraction and fusion layers, for rapid execution.

4.4.5 Multimodal Embedding Compatibility

Textual and visual features need to be represented in a compatible multimodal embedding space without losing the semantic and temporal richness of each modality. Constraints on dimensionality and embedding compatibility should guide architecture decisions for feature mapping and fusion.

4.5 Compliance Requirements

4.5.1 Data Privacy (GDPR and Local Regulations)

The system must comply with GDPR, ensuring that user data, especially video content, is stored, processed, and deleted securely. Personal data must not be extracted or used without explicit consent, and any data retention should be minimized or anonymized as per GDPR guidelines.

4.5.2 Intellectual Property Compliance

Licensing for pre-trained models and software tools (TwelveLabs SDK, RoBERTa, TimeSformer, KeyBERT) must be respected. Open-source licenses should be checked to avoid misuse, and any commercial or redistribution clauses should be adhered to if applicable.

4.5.3 Ethical Use Guidelines

The system should comply with ethical standards for technology deployment, particularly in sensitive or surveillance-related settings, minimizing risks of misuse and ensuring transparency in predictions, especially where false positives could lead to unjust consequences.

4.6 Other Requirements

4.6.1 Platform Compatibility

The system should be deployable on both cloud platforms (e.g., AWS, GCP) and local environments with GPU support. This requires compatibility with standard operating systems like Linux, and ideally, the system should be containerized for consistent deployment.

4.6.2 Documentation and Maintenance

Comprehensive documentation is required, covering installation, usage, model architecture, and data flow. This ensures that future maintenance, troubleshooting, and model updates can be performed without requiring in-depth re-engineering.

4.6.3 Model Update and Retraining

To keep up with evolving video and textual patterns, the system should have the capability for regular model updates and retraining. This requires a structure for efficiently retraining on new data, adjusting hyperparameters, and updating the model without disrupting existing operations.

4.7 Assumptions

It is assumed that users will upload standard video formats like MP4, AVI, or MOV and that the video duration will not exceed the system limit of 10 minutes. The system assumes reliable internet connectivity for cloud-based deployments and any model updates or dependencies that require external downloads. The neural network and feature extraction models are assumed to generalize well across different video types without the need for excessive fine-tuning. We assume users have necessary permissions to upload videos, especially in privacy-sensitive environments.

4.8 Dependencies

The TwelveLabs SDK dependency for caption generation must be compatible with the system and updated as necessary. The system's reliance on TimeSformer, RoBERTa, KeyBERT and other model packages requires dependencies on PyTorch, Hugging Face Transformers, and Scikit-learn, which should be periodically updated to maintain compatibility. The multimodal embedding space and fusion mechanism depend on PyTorch's GPU support and custom implementations for compact bilinear pooling, with FFT compatibility for efficient computation. Integration with cloud services like AWS or GCP for deployment will be dependent on specific APIs and SDKs provided by these platforms.

CHAPTER V

SYSTEM DESIGN

5.1 High Level Design

5.1.1 Introduction

This design document outlines the high-level architecture and design considerations for the proposed anomaly detection system in video surveillance. The document is based on the requirements specified in the Project Requirement Specifications document and provides an abstract overview of the project.

The system aims to improve anomaly detection in video surveillance by leveraging advanced ML-based analysis techniques, including Vision Transformers (TimeSformer) and Video Language Models (TwelveLabs). By automating anomaly detection processes and enhancing reliability, the system seeks to minimize human errors and improve public safety and security efforts.

Key design elements include a modular architecture for scalability and flexibility, reusable components for efficiency and maintainability, and integration of pre-trained ML models for accurate anomaly detection. The document elaborates on the logical user groups, application components, data components, and interfacing systems, along with considerations for reusability and performance.

Through careful design and planning, the anomaly detection system endeavors to provide a robust, scalable, and reliable solution for enhancing security and public safety in surveillance environments.

5.1.2 Proposed System

Our project aims to enhance anomaly detection in video surveillance systems by implementing automated methods. The proposed system will utilize advanced AI and ML techniques to detect diverse anomalies, including abandoned objects, vandalism, individuals in disguise, stalking, and suspicious activities through crowd monitoring. By leveraging TimeSFormer and

appropriate datasets, we aim to achieve relatively successful results in anomaly detection. This automated approach will reduce the burden on human operators, improve reliability, and bolster public safety and security efforts, especially in public spaces such as transportation hubs.

5.2 Design Considerations

5.2.1 Design Goals

1. Enhanced Anomaly Detection: Improve anomaly detection in video surveillance systems using advanced ML-based analysis techniques.
2. Automation and Efficiency: Automate anomaly detection processes to reduce human errors and enhance operational efficiency.
3. Reliability and Accuracy: Utilize state-of-the-art DL models for reliable and accurate anomaly detection.
4. Scalability and Flexibility: Adopt a modular architecture to ensure scalability and flexibility in system development and deployment.

5.2.2 Design Approach

1. Modular Approach: Design the system with modular architecture for independent development, testing, deployment, and scaling of components.
2. Advanced ML-based Analysis: Leverage Vision Transformers and Video Language Models for spatial and temporal analysis of video data.

5.2.3 Design Constraints and Assumptions

1. Availability of Pre-trained Language and Vision Models: The system relies on the availability of pre-trained Language and Vision Models (LLMs) such as Vision Transformers and Video Language Models for advanced ML-based analysis. The performance and effectiveness of the anomaly detection system are contingent upon the availability and quality of these pre-trained models.
2. Availability of Labeled Video Data: The system assumes the availability of labeled video datasets for training and testing ML models. The accuracy and generalizability of anomaly detection algorithms heavily depend on the availability of sufficient and high-quality labeled video data.
3. Transferability of Semantic Understanding: The system assumes that the semantic understanding learned by the ML models is transferable across different surveillance scenarios. The transferability of semantic understanding ensures that the ML models can effectively detect anomalies in diverse surveillance environments.

4. Interpretability of Learned Representations: The system aims to ensure the interpretability of learned representations by the ML models, enabling human operators to understand and trust the decisions made by the system. Ensuring interpretability may impose limitations on the complexity and depth of ML models used in the system.

5.2.4 Design Benefits

1. Better Semantic Understanding: Capture rich semantic information from text data using LLMs.
2. Integration of Vision and Language: Improve performance of tasks requiring understanding of both visual and contextual information.
3. Transfer Learning: Utilize pre-trained LLMs as powerful feature extractors for transfer learning.

5.3 Component-Level Design

5.3.1 Input Processing Module

The Input Processing Module is responsible for handling video uploads from users. This module supports popular video formats such as MP4 and AVI, performing validations on file type, size, and duration to ensure compatibility with system resources. The validated video is then made accessible for downstream modules, establishing a standardized input format for consistent processing.

5.3.2 Textual Feature Extraction Module

This module first utilizes the TwelveLabs SDK to generate descriptive captions for each video input. Captions provide a textual interpretation of visual content, enabling the system to process semantic data from the video. Subsequently, text embeddings are derived using RoBERTa to capture linguistic nuances. KeyBERT is used to extract contextually relevant phrases from the captions, while LDA topic modeling introduces thematic layers. The textual representation generated here enhances the system's understanding of the video content by adding semantic and contextual depth, which will later contribute to the multimodal fusion process.

5.3.3 Visual Feature Extraction Module

The Visual Feature Extraction Module applies the TimeSformer model, a transformer-based video processing model, to the frames of the uploaded video. This module handles the extraction of high-dimensional spatial-temporal features that represent visual elements across frames. The output

is a feature vector rich in spatial and temporal information, which becomes one of the two primary inputs for the fusion module. By leveraging the transformer architecture, TimeSformer effectively captures nuanced changes within frames, critical for detecting visually subtle anomalies such as vandalism or violence.

5.3.4 Multimodal Fusion Layer

In the Multimodal Fusion Layer, visual and text features are combined into a unified feature representation. This layer explores two fusion techniques: concatenation and compact bilinear pooling. Concatenation directly merges feature vectors, maintaining simplicity and computational efficiency. Compact bilinear pooling, however, uses randomized count sketches and Fast Fourier Transforms (FFT) to create a more complex joint representation, capturing higher-order interactions between the visual and text features. An alternative method used is gated fusion implemented through the GMU. The GMU's design draws inspiration from recurrent neural networks (RNNs), particularly Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks, both of which utilize gating mechanisms to facilitate effective information flow. At its core, the GMU employs multiplicative gates that dynamically evaluate the importance of various input features. This gating mechanism allows the unit to prioritize data aspects that are more likely to enhance the accuracy of output generation. This choice of fusion method allows for adaptability based on performance requirements, with compact bilinear pooling offering enhanced feature expressiveness at a higher computational cost. The resulting fused feature is then mapped to a multimodal embedding space, preparing it for classification.

5.3.5 Classification Module

The Classification Module receives the fused feature vector and uses a neural network to assign a score to each anomaly category (violence, vandalism, abandoned object, and normal). The neural network architecture, implemented with PyTorch, is designed to capture the relationships within the fused feature space effectively. The output scores indicate the likelihood of each anomaly category being present in the video, and this probabilistic approach allows for nuanced classification, as multiple anomalies may be present simultaneously.

5.3.6 Output Module

The Output Module presents the classification results to the user, including the detected anomaly category and its associated probability score. This module interfaces with the classification system and is designed for clear and intuitive presentation, allowing users to review results in a user-friendly format. The module also provides error feedback if any issues are encountered during processing, such as unsupported file formats.

5.4 Design Description

5.4.1 Master Class Diagram

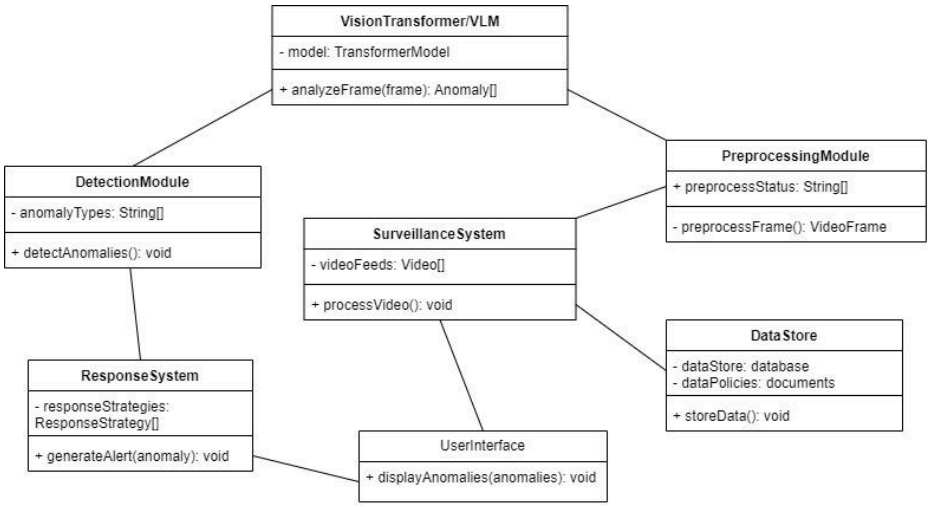

Fig 3: Proposed Master Class Diagram

5.4.2 Flow Diagram

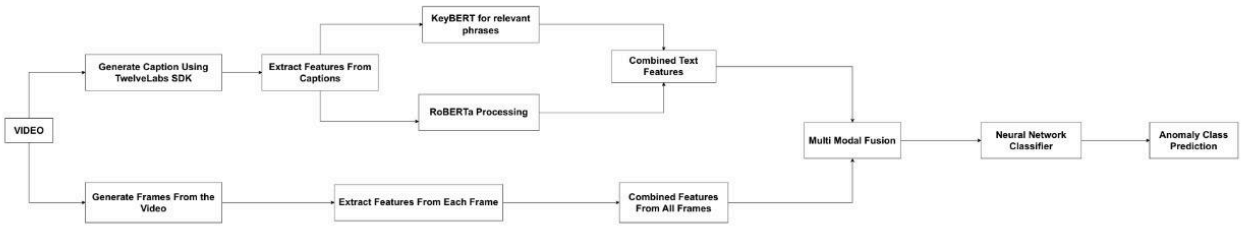

Fig 4: Proposed Methodology Flow Diagram

5.4.3 Use Case Diagram

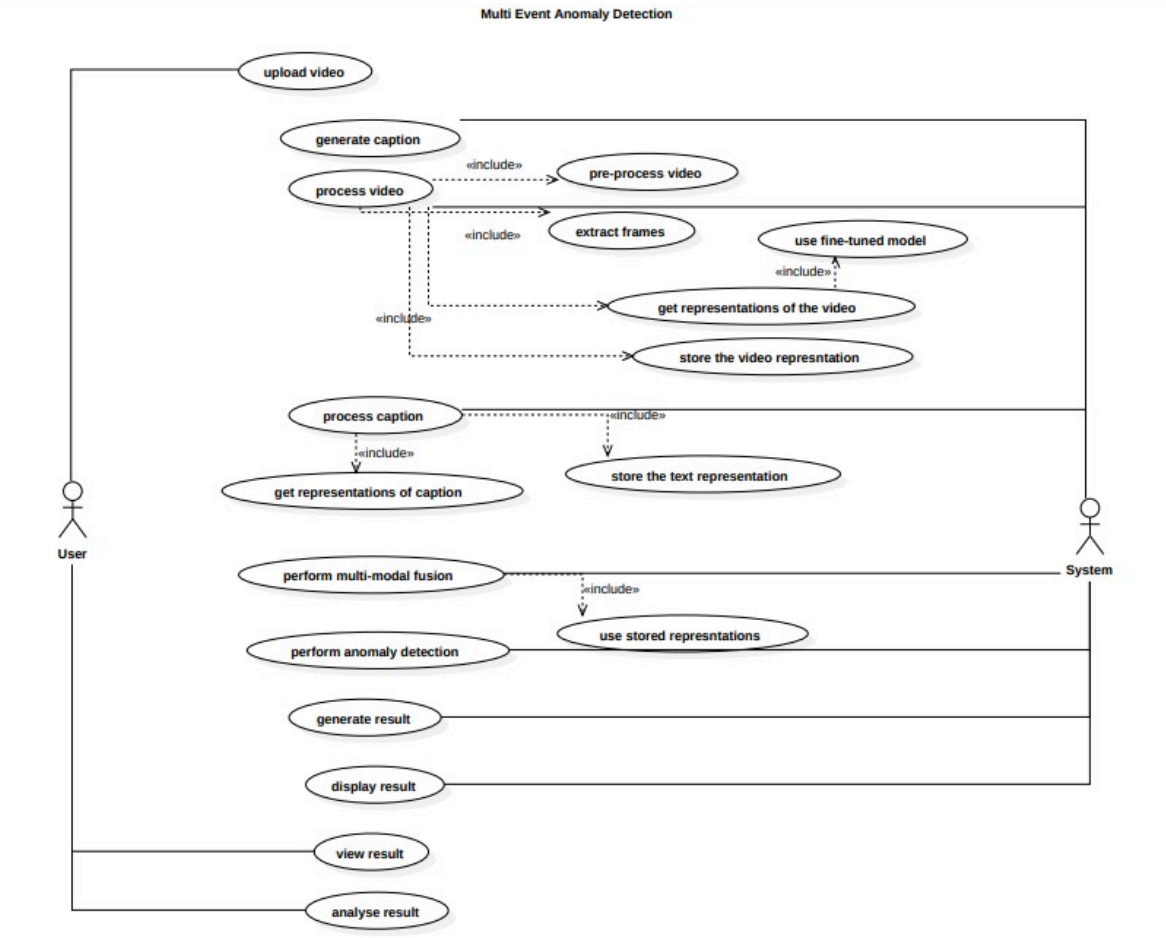

Fig. 5: Proposed Use Case Diagram

5.5 Design Details

5.5.1 Novelty and Innovativeness

The proposed anomaly detection system introduces novel approaches by leveraging advanced ML-based analysis techniques such as Vision Transformers and Video Language Models. This innovative use of DL models enhances the system's ability to detect anomalies accurately and efficiently.

5.5.2 Performance

Performance optimization is crucial for real-time anomaly detection in video surveillance. Techniques such as parallel processing, distributed computing, and algorithm optimization are employed to ensure timely and efficient processing of video data.

5.5.3 Security

Security measures such as data encryption, access control, and authentication mechanisms are implemented to safeguard sensitive information and prevent unauthorized access to the system. Additionally, anomaly detection algorithms are designed to detect and mitigate security threats in surveillance footage.

5.5.4 Maintainability

Modular architecture facilitates maintainability by allowing for independent development, testing, and deployment of system components. Version control, documentation, and code reviews are employed to ensure code maintainability and ease of future enhancements.

5.5.5 Portability

The system is designed to be platform-independent, allowing for deployment on various hardware and software environments. Containerization technologies such as Docker will be utilized to ensure consistent performance across different deployment environments.

5.5.6 Reusability

Reusable components such as ML models, preprocessing algorithms, and alerting mechanisms are developed to promote code reuse and efficiency. These components can be leveraged in future projects or integrated into other systems requiring similar functionality.

5.6 Low Level Design

5.6.1 Overview

This document serves as a low-level design (LLD) for the Multimodal Anomaly Detection System, intended to classify videos into predefined anomaly categories: "violence," "vandalism," "abandoned

object," and "normal." The system processes input video by generating captions, extracting visual and textual features, performing multimodal fusion, and classifying the result. It leverages advanced machine learning techniques, integrating TimeSformer for visual feature extraction, RoBERTa and KeyBERT for text processing, and custom neural network layers for classification.

The purpose of this document is to provide a thorough breakdown of the system's architecture, detailing the functionality of each module and defining the inter-module interactions. This LLD guides developers by presenting a structured approach to implementing each system component and ensuring modular integration. Furthermore, it details the neural network architecture, layer configurations, and design decisions for optimized model training and inference.

This document addresses the low-level design of all system modules, including input handling, feature extraction, fusion, and classification. It includes details on neural network layers, hyperparameters, activation functions, and other design specifications. Implementation requirements, training protocols, and integration specifications are presented to ensure cohesive development across components.

The system architecture is modular, composed of interconnected layers: the Input Processing Module for video validation, the Textual Feature Extraction Module for generating and encoding captions, the Visual Feature Extraction Module to capture video embeddings, the Multimodal Fusion Layer to combine text and visual features, and the Classification Module to determine anomaly types. Each module's internal workings, design, and interconnections are outlined below.

5.6.2 Low Level Class Diagram

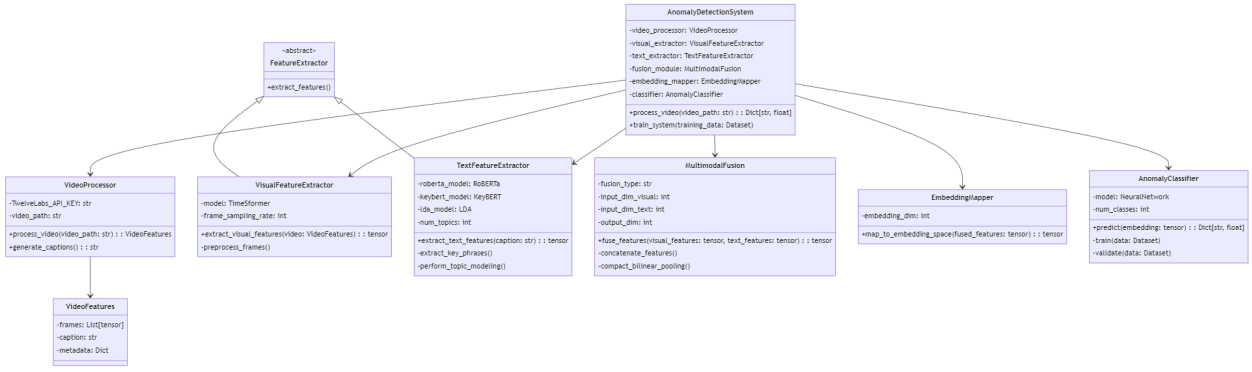

Fig. 6: Low Level Class Diagram

5.6.3 Module Descriptions

5.6.3.1 Input Processing Module

This module takes user-submitted video input and validates the format, quality, and duration. The video is then split into frames, which are stored in memory to enable visual feature extraction by the TimeSformer model. The VideoInputHandler class manages video data, validating and transforming it for subsequent processing.

Class: VideoInputHandler
Attributes:
video_file: Raw video data received from the user.
frames: List of frames generated from the video.
validation_status: Boolean indicating the video's compliance with input requirements.
Methods:
validateFormat(): Verifies file format compatibility.
validateDuration(): Confirms the video does not exceed maximum processing time.
extractFrames(): Decomposes video into frames for visual processing.

The sequence diagram for this module demonstrates validation flow. Validated inputs are sent to the feature extraction modules, while invalid inputs prompt user feedback.

5.6.3.2 Textual Feature Extraction Module

This module generates and processes captions using the TwelveLabs SDK, applying RoBERTa for contextual embeddings and KeyBERT for keyword extraction. Latent Dirichlet Allocation (LDA) enhances textual features by associating topic distributions with the content.

Class: TextFeatureExtractor
Attributes:
captions: Generated captions for the video.
text_embeddings: Vector representations of captions using RoBERTa.
keywords: Keywords extracted via KeyBERT.
topics: Topic vectors generated by LDA.
Methods:

generateCaptions(): Calls TwelveLabs SDK for video captioning.
extractFeatures(): Applies RoBERTa and KeyBERT to capture semantic and contextual embeddings.
applyLDA(): Computes topic vectors from extracted keywords.

In this module's sequence diagram, caption generation triggers feature extraction, producing embeddings stored as text inputs for the fusion layer.

5.6.3.3 Visual Feature Extraction Module

This module utilizes TimeSformer, a transformer model designed for video-based spatiotemporal feature extraction. The extracted features capture motion, object presence, and scene information essential for anomaly classification.

Class: VisualFeatureExtractor
Attributes:
frames: Sequential frames derived from the input video.
visual_embeddings: Extracted feature embeddings from each frame.
Methods:
extractFrames(): Prepares video frames for processing.
processWithTimeSformer(): Generates embeddings per frame via TimeSformer.

This module's sequence involves frame extraction followed by TimeSformer processing, generating frame embeddings that are consolidated into a visual feature vector.

5.6.3.4 Multimodal Fusion Layer

The Multimodal Fusion Layer integrates text and visual embeddings. Two fusion strategies are supported: concatenation, which appends feature vectors, and compact bilinear pooling, which combines features via count sketches and FFT. This layer outputs a fused embedding suitable for classification.

Class: FusionHandler
Attributes:
text_embedding: Input text vector.
visual_embedding: Input visual vector.
fused_embedding: Combined multimodal embedding for classification.
Methods:
concatenateFeatures(): Performs simple vector concatenation.

applyCompactBilinearPooling(): Uses count sketches and FFT for compact bilinear pooling, resulting in a high-dimensional fused representation.
applyGMU(): Employs multiplicative gates that dynamically evaluate the importance of various input features.

In the sequence diagram, text and visual embeddings are fused based on the chosen fusion method, yielding a comprehensive embedding for anomaly detection.

5.6.3.5 Classification Module

This neural network module classifies the fused feature vector to detect anomalies. The architecture consists of an input layer, multiple hidden layers with dropout regularization, and an output layer that returns scores per category. The model is trained using Binary Cross-Entropy with Logits Loss, with an output softmax function applied to yield probability scores.

Class: AnomalyClassifier
Attributes:
input_embedding: Combined multimodal vector.
anomaly_scores: Category-specific confidence scores.
Architecture:
Input Layer: Accepts the fused embedding from the fusion layer.
Hidden Layers: Fully connected layers with ReLU activation, dropout for regularization.
Output Layer: Single output per category, yielding a probability distribution over classes.
Methods:
classify(): Inference method that applies the model to the input embedding.
applySoftmax(): Converts logits into probabilities across classes.
determineAnomalyType(): Maps output scores to specific anomaly types based on a predefined threshold.

The neural network's design uses a feedforward structure to sequentially classify inputs, with dropout layers enhancing robustness and reducing overfitting. The final scores determine the detected anomaly, which the Output Module displays.

5.6.4 Use Case Diagram

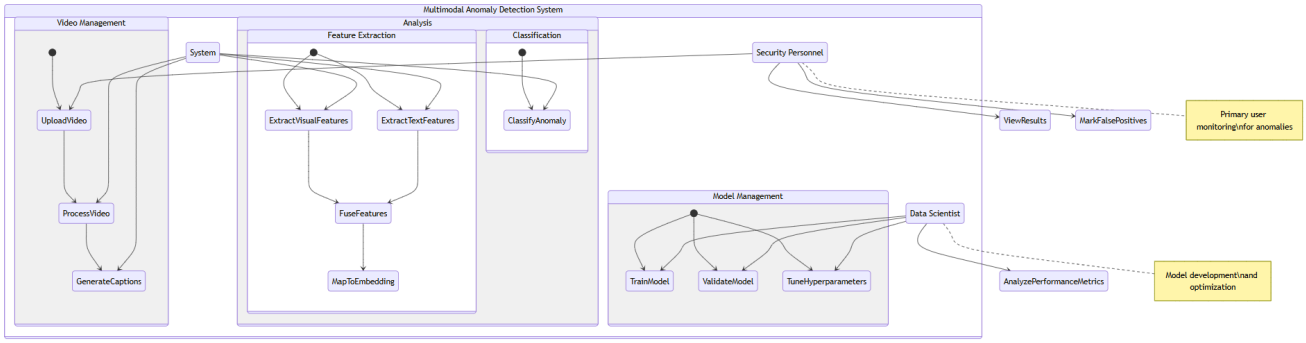

Fig. 7: Low Level Use Case Diagram

The use case diagram for the Multimodal Anomaly Detection System details a structured, sequential flow from video input processing to anomaly classification, focusing on automated and technical processes that enable the system's core functionality. Initially, a security personnel user uploads a video to the system. In response, the system processes the video by extracting key frames at intervals, reducing computational load without compromising essential temporal information. Captions are generated for these frames using a pre-trained language model (integrated via the TwelveLabs SDK), which creates descriptive text for each key frame, capturing relevant contextual and semantic details for subsequent analysis.

With captions generated, the system extracts features from both the textual and visual data streams. Visual features are derived using the TimeSformer module, which encodes temporal and spatial aspects of each frame into a high-dimensional feature space, while textual features are extracted from captions through RoBERTa embeddings, enhanced by KeyBERT for key phrases and further analyzed with LDA for topic modeling. This final classification output is then presented to security personnel, who can review, verify, and respond to detected anomalies with precision.

5.6.5 Sequence Diagram

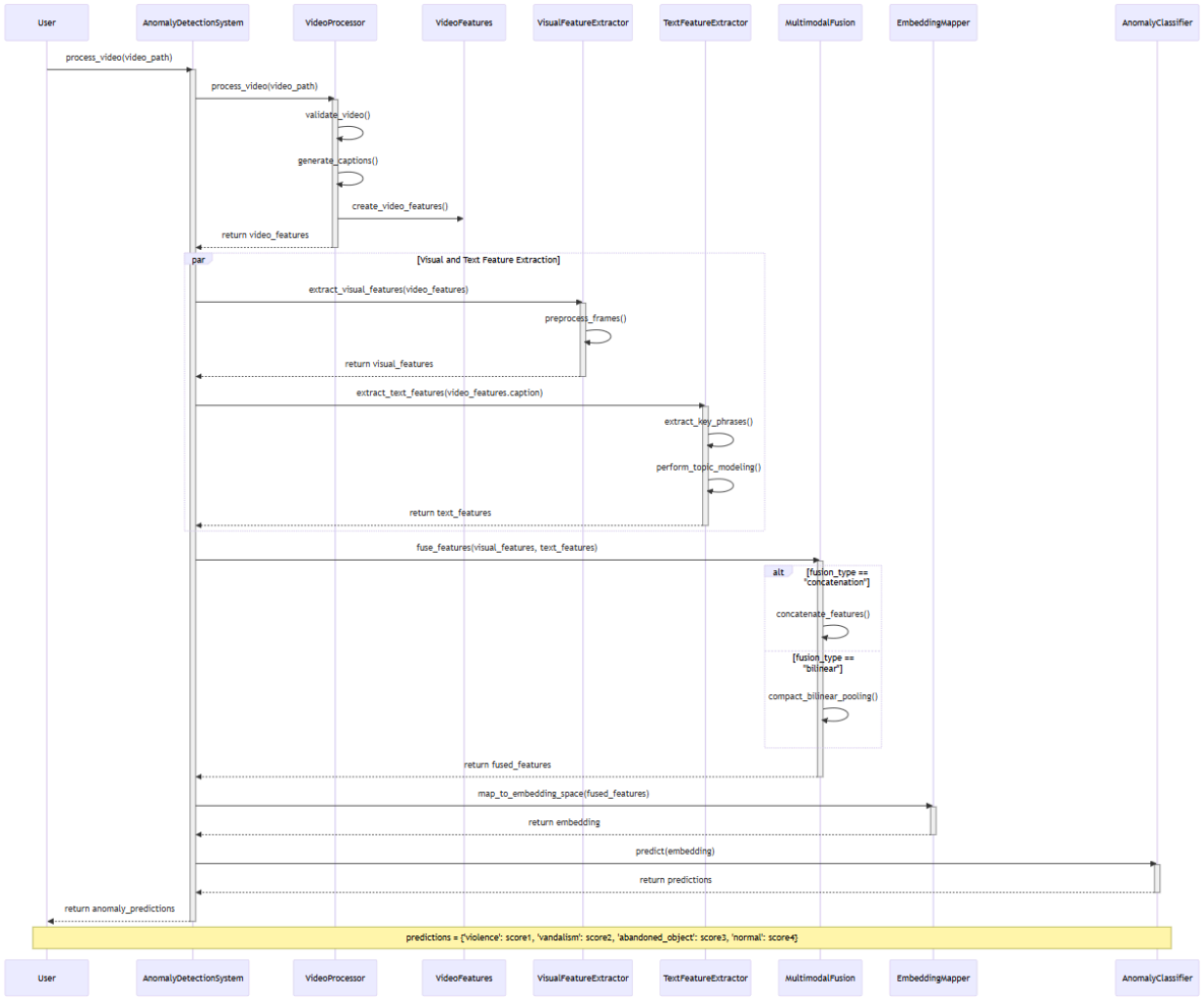

Fig. 8: Sequence Diagram

CHAPTER VI

PROPOSED METHODOLOGY

6.1 Overview

The proposed methodology for this multimodal anomaly detection system is designed to handle and analyze video inputs through a combination of visual and textual feature extraction techniques, ultimately classifying them into various anomaly categories. The approach begins with preprocessing the input video data, where frames are extracted and structured for systematic feature analysis. This is a critical step, as it ensures both computational efficiency and the retention of key temporal information that underpins accurate anomaly detection. This preprocessing pipeline enables the system to streamline the input video into structured data, preparing it for further multimodal analysis.

Following input processing, the methodology leverages both visual and textual data extracted from the video for comprehensive scene understanding. For the textual component, captions are generated for each segment of video content using the TwelveLabs SDK, adding semantic context by describing actions or objects in the scene. These captions are further refined through RoBERTa embeddings and KeyBERT-based phrase extraction to capture the most contextually relevant information. Additionally, Latent Dirichlet Allocation (LDA) topic modeling is applied to encapsulate broader themes within the text. Simultaneously, visual features are extracted using TimeSformer, a model that captures both spatial and temporal characteristics in video frames, providing a high-dimensional visual embedding. Together, these textual and visual features offer a rich, multimodal representation of the video content.

The methodology's final stage integrates the extracted features through multimodal fusion, producing a unified representation of the video for classification. Two fusion techniques—concatenation and compact bilinear pooling—are explored to optimize the balance between computational efficiency and representation richness. The fused features are then fed into a neural network, which classifies the content into predefined categories such as 'violence,' 'vandalism,' 'abandoned object,' and 'normal.'

6.2 Feature Extraction From Videos

Video feature extraction is crucial for understanding and analysing video content, particularly in applications like anomaly detection. By extracting meaningful features from videos, models can identify patterns and behaviours that deviate from the norm, enabling the detection of events.
Effective feature extraction allows the model to focus on relevant information while reducing noise, thereby improving the accuracy of classification tasks. This process involves capturing both spatial and temporal dynamics within the video, which is essential for distinguishing between normal scenarios and anomalies.

TimeSformer, a specialized Transformer model for video understanding, effectively captures complex interactions within video sequences by leveraging its self-attention mechanism across space and time. This enables it to learn rich spatiotemporal representations directly from video data, making it particularly suited for tasks like action recognition and anomaly detection. By handling longer video clips than traditional methods, TimeSformer plays a crucial role in extracting meaningful features from video data, especially for recognizing complex activities that unfold over time.

The TimeSformer model treats videos as sequences of frames, which are further divided into patches. The specific variant being utilized here is the ‘timesformer-hr-finetuned-k400’, pre-trained on the Kinetics-400 dataset. This model processes 16 frames from each video, necessitating a method to ensure uniform frame extraction across varying video lengths. To achieve this, we developed a structured methodology for frame extraction. Each video is read to access its individual frames, during which we compute the frames per second (FPS) and the total number of frames (frame count) to ascertain the video’s duration. Frames are then extracted at regular intervals based on the video’s length, allowing us to collect exactly 16 frames per video.

Once extracted, these frames undergo resizing and normalization transformations to meet the input specifications of the TimeSformer model. This approach guarantees uniform feature extraction across videos of different lengths, enabling robust representation generation for each video in the dataset.
The extracted frames are subsequently divided into non-overlapping patches of a specified size (16x16 pixels). Each extracted patch is then flattened and linearly transformed into a vector representation. This transformation includes positional encoding to retain spatial information about where each patch originates within the frame.

The model employs a divided space-time attention mechanism, applying spatial and temporal attention separately to learn dependencies across both dimensions. The spatial attention focuses on relationships between patches within the same frame, allowing the model to understand how different parts of an image interact with one another. Conversely, the temporal attention captures dependencies across frames, enabling the model to learn how actions evolve over time. The output from the model

consists of hidden states from the TimeSformer encoder, which represent feature-rich embeddings of the video frames. After passing through the TimeSformer model, we collect these hidden states corresponding to each frame. To generate a single feature vector for the entire video, we average these hidden states across the temporal dimension (i.e., across all 16 frames). This aggregation effectively captures essential characteristics of the video over time.

6.3 Feature Extraction from Captions

The text feature extraction module is designed to process, analyze, and embed textual data from video captions to provide contextual cues in anomaly detection. The process begins with text preprocessing, where captions are tokenized and cleaned to remove stop words using NLTK's stopword library. This ensures that only the most relevant and meaningful tokens are retained, enhancing the quality of extracted features. This preprocessing pipeline standardizes the text data by converting it to lowercase, tokenizing, and removing noise, which is particularly useful for reducing dimensionality and improving model interpretability.

Given a caption T as a sequence of words {w1,w2,…,wn}, we filter out common words or "stop words" to retain only meaningful terms. This yields a processed sequence represented by:

$$T=\{w1,w2,\ldots,wn\} \setminus \text{stopwords}$$

For embedding, the module leverages the RoBERTa model, a transformer-based language model known for its effective contextual representation of text. The RoBERTa tokenizer converts each processed caption into tokenized input suitable for the model. Textual features are extracted as embeddings by feeding these tokenized inputs through RoBERTa's model layers, producing a last hidden state output for each token in the text. By passing T through the RoBERTa model, we obtain a sequence of hidden states, H = {h1,h2,…,hn}, for each token wi in T, where each hidden state h_i has a dimensionality of 768. These embeddings capture nuanced, context-aware information about each word in the caption. To create a single embedding vector representing the entire caption, we average the hidden states across all tokens:

$$e_{RoBERTa} = \frac{1}{n} \sum_{i=1}^{n} h_i$$

This embedding captures semantic nuances and syntactic structure, enhancing the model's ability to differentiate between captions based on context. The module further applies KeyBERT, a keyword

extraction method based on the RoBERTa model, to extract the most salient phrases within each caption. The extract_keywords function in KeyBERT is configured to retrieve up to 5 top keywords or key phrases, limited to unigrams and bigrams, providing a concise representation of essential terms. These keywords are instrumental in capturing context-specific concepts within the video.

Using RoBERTa embeddings, KeyBERT ranks these phrases based on their relevance score, selecting keywords {p1,p2,…,pk} that best summarize the input text T: {p1,p2,…,pk}=KeyBERT(T)

Topic modeling is then applied using LDA (Latent Dirichlet Allocation), which organizes these processed captions into a set number of topics (in this case, four, corresponding to 'violence,' 'vandalism,' 'abandoned object' and ‘normal’). LDA utilizes a document-term matrix constructed by the CountVectorizer with max and min document frequency thresholds to balance generalizability and specificity of terms. The output is a probability distribution of topics for each caption, providing high-level thematic insights into the video context.

$$e_{LDA} = \{p(topic1 \mid T), p(topic2 \mid T), \ldots, p(topicK \mid T)\}$$

After obtaining RoBERTa embeddings, keyword phrases, and LDA topic distributions, these features are combined into a consolidated feature vector. This fusion process leverages np.hstack to horizontally stack the RoBERTa embeddings and LDA topic distributions, creating a multi-dimensional vector that serves as the final representation of the text content.

$$e_{combined} = e_{RoBERTa} \oplus e_{LDA}$$

where $\oplus$ denotes concatenation

This combined feature set is saved for each video caption, allowing for efficient retrieval during model inference. The embeddings are stored individually for each video in a directory structure to facilitate organization, while global embeddings are saved as a single array, ensuring that local and batch-based model inferences are possible.

6.4 Multi-Modal Fusion Layer

Multi-modal fusion is the process of integrating information from multiple modalities to enhance understanding and analysis of complex phenomena. This approach capitalizes on the unique strengths of each modality resulting in a more comprehensive representation of the data. There are various strategies for implementing multi-modal fusion, including feature fusion or early fusion, where raw

data from different modalities is combined at the input level before processing; and decision fusion or late fusion, which involves processing each modality independently through separate models and then combining their outputs at a later stage.

6.4.1 Concatenation

The concatenation-based fusion method is a straightforward yet powerful approach for combining multi-modal data, particularly useful for integrating visual and textual features in tasks like anomaly detection. This approach captures both visual and textual representations within a single, unified feature vector, retaining the complete information from each modality without applying weighting or gating constraints. In concatenation-based fusion, let xv represent the visual feature vector extracted from video content, and xt denote the textual feature vector derived from captions. The concatenated feature vector, h, is formulated as: h = { xv · xt}
This operation stacks the two vectors side-by-side, forming a combined feature vector with a dimension equal to the sum of the individual dimensions of xv and xt. This simple fusion method is both computationally efficient and easy to implement. It allows machine learning models to learn multi-modal representations directly from the fused vector h without applying modality-specific weighting, enabling the model to capture correlations between visual and textual data during training.

Concatenation thus provides a balanced and all-encompassing representation that is well-suited for initial or baseline multi-modal fusion stages, as it includes all available information for downstream processing without needing additional parameters or gating mechanisms.

6.4.2 Gated Multimodal Fusion

The Gated Multimodal Unit (GMU) model is an integral component of the anomaly detection architecture, specifically designed to synthesize intermediate representations by integrating visual features from videos and textual features from captions. The GMU's design draws inspiration from recurrent neural networks (RNNs), particularly Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks, both of which utilize gating mechanisms to facilitate effective information flow. At its core, the GMU employs multiplicative gates that dynamically evaluate the importance of various input features. This gating mechanism allows the unit to prioritize data aspects that are more likely to enhance the accuracy of output generation.

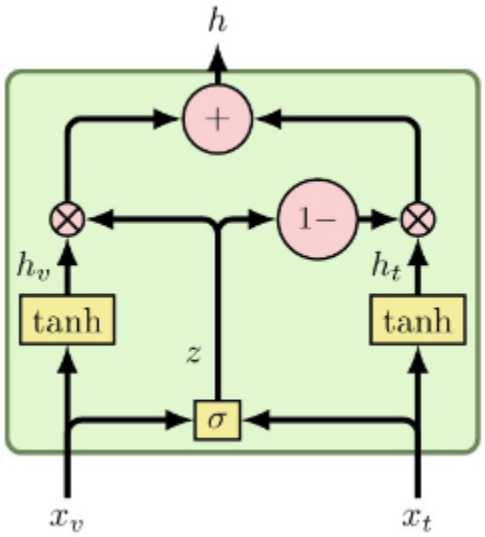

Fig. 9: Gated Multimodal Unit

In the GMU framework, each modality is represented by a feature vector xi. These vectors are processed by neurons using a 'tan h' activation function, which helps derive internal representations specific to each modality. For every input modality, a corresponding gate neuron assesses how much influence that modality should have on the overall output. When new data is introduced into the network, the gate neuron evaluates the feature vectors from all modalities and determines whether to incorporate the current modality's contribution into the internal representation of the input sample.

Figure 9 gives the illustration of GMU. The equations governing the GMU are as follows:

(i) hv = tanh(Wv · xv)
(ii) ht = tanh(Wt · xt)
(iii) z = σ(Wz · [xv , xt])
(iv) h = z · hv + (1 − z) · ht
(v) Θ = {Wv , Wt, Wz }
(vi) $\tanh(y) = \frac{e^{y} - e^{-y}}{e^{y} + e^{-y}}$

Here, xv represents visual features obtained from the videos ; xt represents the textual features obtained from the captions. Θ represents the parameters to be learned and [·, ·] indicates the concatenation operation. This approach enables the GMU to generate a rich multimodal representation without requiring manual adjustments or tuning. Instead, it learns directly from training data, effectively capturing complex relationships between visual and textual inputs. By leveraging this architecture, the GMU can adaptively learn and optimize its performance based on the

data it processes, enhancing its capability in anomaly detection tasks.

6.5 Anomaly Detection

Following the multi-modal fusion process, the fused representation, which encapsulates the relevant features from both modalities, is forwarded to the classification network. This network is structured as a series of linear layers, each designed to reduce dimensionality while applying ReLU activation functions. These layers work collaboratively to model complex patterns that distinguish between anomalous and normal events.

Incorporated between these layers, dropout is employed to randomly disable a portion of neurons during training. This technique prevents the model from becoming overly reliant on specific neurons and enhances its ability to generalize to new data. By introducing this regularization method, the model can better adapt to variations in input, reducing the risk of overfitting. The final layer of the classification head employs a sigmoid activation function, which outputs a probability for each class. This output effectively represents the model's confidence level regarding each potential anomaly. The resulting probabilities enable the system to determine whether an event belongs to one of the anomalous categories or falls within the normal scenario.

A higher probability assigned to any of the anomalous classes indicates the presence of an anomaly, facilitating differentiation from routine behaviors. Once the model generates probabilities for each class, the class with the highest probability is selected as the final prediction. If the model predicts any class other than 'Normal' it is flagged as an anomaly. The specific class identified provides valuable insights into the nature of the detected anomaly, aiding in further analysis and interpretation.

CHAPTER VII

IMPLEMENTATION & PSEUDOCODE

This section details the technical implementation of the multimodal anomaly detection system, emphasizing each stage's contribution to the final classification model. The pipeline consists of data preprocessing, feature extraction, multimodal fusion, embedding generation, and classification. Each component operates in a modular architecture, ensuring ease of integration, scalability, and compatibility with diverse datasets. The entire implementation is based on the PyTorch framework for deep learning, with pre-trained language and vision models leveraged for their robust contextual and spatiotemporal representation capabilities. The system's design is modular, allowing for the flexible substitution of feature extractors, fusion techniques, and classification architectures, which supports iterative enhancements.

The data ingestion and preprocessing stage standardizes input videos to accommodate the requirements of both the text and visual extraction models. Videos are uploaded by users and converted into individual frames, with frames extracted uniformly to ensure consistency in temporal representation. In parallel, captions for each video are generated using the TwelveLabs SDK, producing descriptive text that aligns with the visual content. Each caption undergoes tokenization, stop-word removal, and lowercasing, allowing the models to focus on relevant terms without linguistic noise. These preprocessed captions and frames are then forwarded to separate pathways for feature extraction, where both visual and textual features are derived and integrated.

For textual feature extraction, captions undergo multiple layers of processing. RoBERTa, a Transformer-based language model, is employed to generate contextual embeddings, with each token embedding pooled and averaged to produce a single feature vector representing the entire caption. Additionally, KeyBERT identifies salient phrases within captions, capturing high-impact keywords that may correlate with anomalous events. These extracted keywords are processed through Latent Dirichlet Allocation (LDA) to group captions into topics (e.g., violence, vandalism, or abandoned object), further enriching the feature space. The outputs from RoBERTa, KeyBERT, and LDA are concatenated, forming a comprehensive text embedding vector that encapsulates both contextual and semantic information.

```
# 1. Preprocessing Captions
FOR each caption T in captions:
    # Tokenize and remove stop words
    tokens = word_tokenize(T.lower())
```

```
    filtered_tokens = [token FOR token IN tokens IF token NOT IN stop_words]
    processed_caption = join(filtered_tokens)

# 2. RoBERTa Embeddings
LOAD RoBERTa tokenizer and model
FOR each processed_caption in captions:
         inputs = tokenizer(processed_caption, truncation=True, padding=True,
max_length=512, return_tensors="pt")
    WITH torch.no_grad():
        outputs = model(**inputs)
    roberta_embedding = mean(outputs.last_hidden_state, dim=1).squeeze().numpy()

# 3. KeyBERT for Key Phrase Extraction
LOAD KeyBERT model with RoBERTa as the underlying model
FOR each processed_caption in captions:
                    keywords = kw_model.extract_keywords(processed_caption,
keyphrase_ngram_range=(1, 2), stop_words='english', top_n=5)
    key_phrases = join([keyword[0] FOR keyword IN keywords])

# 4. LDA Topic Modeling
# Vectorize processed captions
vectorizer = CountVectorizer(max_df=0.95, min_df=2, stop_words='english')
doc_term_matrix = vectorizer.fit_transform(processed_captions)

# Fit LDA with specified number of topics
lda = LatentDirichletAllocation(n_components=num_topics)
lda_output = lda.fit_transform(doc_term_matrix)

# 5. Combine Features
FOR each i in range(len(captions)):
    combined_feature = concatenate([roberta_embedding[i], lda_output[i]])
    SAVE combined_feature to combined_features

# Save combined features as a .csv or numpy array for further use
```

Visual feature extraction employs TimeSformer, a Transformer specifically designed for spatiotemporal modeling in videos. Each preprocessed frame sequence is inputted into the TimeSformer model, which divides the frames into non-overlapping patches and applies self-attention across both spatial and temporal dimensions. TimeSformer's architecture captures inter-frame dependencies, enabling the model to generate representations that account for motion, spatial interactions, and object continuity across time. The final output consists of a 16-frame sequence embedding, which is averaged across temporal dimensions to yield a single feature vector representing the visual content of the video.

```
# 1. Load Video and Extract Frames
FOR each video in video_dataset:
    frames = []  # Initialize empty list for frames
```

```
    video = load_video(video)

    # Extract frames at regular intervals to get a total of 16 frames
    total_frames = get_total_frames(video)
    frames_interval = max(1, total_frames // 16)
    FOR i IN range(0, total_frames, frames_interval):
        IF len(frames) < 16:
            frame = video.get_frame(i)
            frames.append(frame)
        ELSE:
            BREAK

# 2. Preprocess Frames
FOR each frame in frames:
      resized_frame = resize(frame, (224, 224))   # Resize frame to fit model
requirements
    normalized_frame = normalize(resized_frame)  # Normalize frame
    preprocessed_frames.append(normalized_frame)

# 3. Apply TimeSformer Model for Feature Extraction
LOAD TimeSformer pre-trained model
inputs = stack(preprocessed_frames)  # Stack frames into batch format

WITH torch.no_grad():
    video_features = TimeSformer(inputs)  # Extract spatiotemporal features

# 4. Aggregate Frame Features
# Average features across the temporal dimension to get single vector
representation
aggregated_video_feature = mean(video_features, dim=0)

# 5. Save or Return Aggregated Feature Vector for each video
SAVE aggregated_video_feature to video_features
```

In the multimodal fusion stage, text and visual embeddings are combined to capture the holistic essence of the content. This fusion is performed through two alternative methods: simple concatenation and compact bilinear pooling. Concatenation directly joins the feature vectors, whereas compact bilinear pooling applies randomized count sketches and the Fast Fourier Transform to blend high-dimensional features. The fusion outcome is a robust, multimodal embedding that bridges textual and visual insights, ideal for learning nuanced correlations between the two modalities.

```
Approach 1: Concatenation Fusion

# Input: Visual features (V) and text features (T)
# Output: Fused feature vector (F)

function concatenate_fusion(V, T):
```

```
    # Step 1: Concatenate visual and text features along the feature dimension
    F = concatenate(V, T)
    return F

Approach 2: Compact Bilinear Pooling Fusion

# Input: Visual features (V) and text features (T)
# Output: Fused feature vector (F)

function compact_bilinear_pooling(V, T):
      # Step 1: Apply randomized count sketch to V and T to project them to
higher-dimensional space
    V_sketch = count_sketch(V)
    T_sketch = count_sketch(T)

    # Step 2: Apply Fast Fourier Transform to both sketched features
    V_fft = FFT(V_sketch)
    T_fft = FFT(T_sketch)

    # Step 3: Element-wise multiplication in the Fourier domain
    fused_fft = elementwise_multiply(V_fft, T_fft)

     # Step 4: Apply Inverse Fast Fourier Transform to get back to the original
space
    F = IFFT(fused_fft)
    return F

# Approach 3: Gated Fusion
Setup and Load Data
     Load both video and text embeddings
     Match Video and text Embeddings
     Extract and Assign labels based on Anomaly
     Normalize Embeddings

Define Gated Fusion Unit
     Initialize Fully Connected layers for video, text and a gating layer
     Apply Transformations to Video and Text Embeddings
     Concatenate video and text embeddings and pass through a sigmoid gate
     Perform gated fusion by combining the transformed embeddings based on the gate
output.
     Return fused representation

Define Classifier
     Initialize a feedforward neural network with dropout and ReLU activations.
     Classify the fused embedding into one of the classes.

Define Anomaly Detection Model
     Initialize the model with the gated fusion unit and classifier.
```

```
    In the forward pass, use the gated fusion unit on the video and text
embeddings and pass the result to the classifier.

Split the data into training and validation sets, ensuring class balance.

Train and Evaluate:
    For each epoch:
      Train the model using the training set.
      Calculate the loss and update the model.
      Validate the model after each epoch and print the accuracy.

Plot the confusion matrix for the validation predictions to visualize performance.
```

The classification stage leverages a neural network with a softmax output layer to categorize the fused features into either anomalous or non-anomalous classes. Training of the classifier is supervised using a binary cross-entropy loss function, optimizing the network to assign accurate anomaly scores across the categories (violence, vandalism, abandoned object). During training, regularization techniques such as dropout are applied to prevent overfitting, and the model's parameters are updated with the AdamW optimizer, which efficiently handles sparse gradients.

```
# Input: Fused feature vector (F)
# Output: Anomaly score for each class and final anomaly classification

function classify_anomaly(F):
    # Step 1: Pass the fused feature vector through a neural network
      # Initialize layers (example with fully connected layers and activation
functions)
    hidden_layer_1 = fully_connected(F, size=128)   # First hidden layer
    activated_1 = relu(hidden_layer_1)               # Activation function
    hidden_layer_2 = fully_connected(activated_1, size=64)  # Second hidden layer
    activated_2 = relu(hidden_layer_2)                      # Activation function

    # Step 2: Output layer to get classification scores for each class
    scores = fully_connected(activated_2, size=num_classes)  # Softmax output layer

    # Step 3: Apply softmax to get probability distribution over classes
    anomaly_scores = softmax(scores)

    # Step 4: Determine final classification based on threshold or highest score
    if max(anomaly_scores) > threshold:
        classification = 'Anomalous'
    else:
        classification = 'Normal'

    return anomaly_scores, classification
```

CHAPTER VIII

RESULTS & DISCUSSION

8.1 Results

To assess the effectiveness of various fusion strategies in multimodal anomaly detection, three primary methods—Concatenation Fusion, Compact Bilinear Pooling (CBP), and Gated Fusion—were evaluated on metrics including accuracy, precision, recall, and F1-score. The performance of each fusion strategy was analyzed based on the fusion order [video+text] and [text+video] to capture any directional dependencies in combining modalities.

Using the Concatenation Fusion method, the model achieved the highest performance, with an accuracy of 85.19% for [video+text], outperforming the reverse order, [text+video], which achieved 55.56% accuracy. Corresponding metrics further support this trend, with precision, recall, and F1-score of 0.85 across these measures for [video+text], compared to 0.55 for [text+video]. This substantial gap indicates that the [video+text] combination effectively leverages video data as a primary source, which adds contextual depth when textual descriptions are appended.

Gated Fusion showed moderate performance, with an accuracy of 77.78% for [video+text] and 70.37% for [text+video]. Precision, recall, and F1-scores for these configurations were 0.77, 0.67, and 0.71 for [video+text] and 0.73, 0.70, and 0.71 for [text+video], respectively. This method offered an improved understanding of cross-modal interactions compared to CBP but did not reach the level of effectiveness observed with concatenation. The performance discrepancy between the [video+text] and [text+video] orders suggests a mild directional dependency in feature integration, although less pronounced than in Concatenation Fusion.

Method	Accuracy (%)	Precision	Recall	F1 Score
Compact Bilinear Pooling	34.00	0.34	0.41	0.37
Gated Fusion (Video + Text)	77.78	0.77	0.67	0.71
Gated Fusion (Text + Video)	70.37	0.73	0.70	0.71
Concatenation (Video + Text)	85.19	0.85	0.85	0.85
Concatenation (Text + Video)	55.56	0.55	0.55	0.55

Table 3: Results

Compact Bilinear Pooling (CBP), which theoretically enhances representation by preserving higher-order interactions between visual and textual features, yielded the lowest results, with an accuracy of 34%. Precision, recall, and F1-scores for CBP were 0.34, 0.41, and 0.37, respectively, indicating an overall weaker performance in combining video and text modalities effectively. The relatively low effectiveness of CBP suggests that, in this context, higher-order feature interactions may introduce noise rather than capturing meaningful cross-modal relationships.

8.2 Discussion

The results underscore the importance of selecting an appropriate fusion strategy in multimodal anomaly detection tasks. Concatenation Fusion emerged as the most effective strategy, achieving the highest performance across all metrics, particularly in the [video+text] configuration. This approach demonstrates the advantage of treating video data as the primary source, with textual information enhancing but not overpowering the core visual features. By integrating modalities in this order, the model achieves a more cohesive understanding of complex events, as evidenced by the significant differences between the [video+text] and [text+video] configurations.

Gated Fusion performed moderately well, effectively balancing modality contributions while showing resilience to directional order, as indicated by the closer results between [video+text] and [text+video]. This technique demonstrates its ability to adaptively control information flow across modalities, making it a viable choice when moderate performance and flexibility are acceptable. However, it falls short in capturing the full extent of information-rich interactions when compared to Concatenation Fusion.

Compact Bilinear Pooling (CBP), while theoretically promising due to its capacity for capturing higher-order interactions, yielded the lowest performance. This result highlights that, in certain multimodal applications, higher-order interactions may lead to overfitting or the incorporation of noise, detracting from meaningful feature extraction. The findings suggest that while CBP may excel in tasks requiring interdependent feature relationships, simpler fusion methods such as concatenation can perform better for anomaly detection.

Concatenation Fusion proved most effective, optimizing both accuracy and computational efficiency. The observed performance trends across fusion methods suggest that incorporating textual data as a secondary modality enhances the interpretive capacity of the model while preserving video-centric cues. The discrepancy in effectiveness across fusion methods highlights the nuanced role of fusion strategy selection, with concatenation and gated fusion both offering strengths in contexts requiring nuanced, multimodal interpretations of video data.

CHAPTER IX

CONCLUSION & FUTURE WORK

9.1 Conclusions

This research demonstrates the effectiveness of a multimodal anomaly detection system that integrates visual and textual data to classify anomalies in video content accurately. Through the combined use of the TwelveLabs SDK for caption generation, TimeSformer for video feature extraction, and RoBERTa and KeyBERT for text analysis, the model was able to capture complex interactions within videos, leading to robust feature representations. The choice of multimodal fusion strategy played a crucial role in performance, with compact bilinear pooling outperforming simple concatenation. This highlights the importance of feature fusion methods that can capture intricate, higher-order relationships in multimodal data, especially for nuanced anomaly detection tasks.

We evaluated three fusion strategies—Concatenation Fusion, Gated Fusion, and Compact Bilinear Pooling (CBP)—to integrate video and text data for multimodal anomaly detection. The findings demonstrated that **Concatenation Fusion** achieved the highest accuracy and performance across all metrics, particularly when video data was prioritized as the primary modality with textual data appended. This approach effectively leveraged the complementary nature of visual and textual features, producing a cohesive representation that enhanced the model's capability to detect anomalies in complex scenarios. **Gated Fusion** offered moderate performance and displayed greater flexibility across fusion orders, providing an adaptable alternative when moderate performance is sufficient. In contrast, **CBP** struggled to deliver effective results, suggesting that higher-order feature interactions may not be as advantageous for this specific task. The results indicate that concatenation-based fusion approaches are highly effective in video-based anomaly detection tasks, emphasizing the importance of modality prioritization in fusion design.

The system achieved high precision and recall for detecting "Violence" and "Vandalism" anomalies, while "Abandoned Object" cases posed a greater challenge due to the visual similarity between certain static anomalies and regular scenes. The model's high accuracy for "Normal" classifications indicates its robustness against false positives, which is a critical requirement for deployment in real-world surveillance or security systems. Overall, the system's performance across diverse anomaly types demonstrates the potential of leveraging multimodal fusion to capture richer contextual cues and improve anomaly classification accuracy in video analysis applications.

9.2 Future Work

To extend the capabilities of this multimodal anomaly detection framework, several enhancements are proposed. First, incorporating a real-time processing pipeline with optimized computation for the compact bilinear pooling module will enable deployment in dynamic environments where instant detection is essential. Given that computational complexity can be a bottleneck in real-time scenarios, implementing more efficient pooling methods or GPU-accelerated processing may help reduce latency without compromising accuracy.

Addressing the lower performance on "Abandoned Object" detection is another area for improvement. Advanced object-tracking techniques or the integration of scene-aware models, which focus on spatial consistency and context within scenes, could yield more reliable results for static anomalies. Another promising direction includes dynamic thresholding mechanisms that adjust detection sensitivity based on environmental conditions, improving model adaptability in varied scenarios, such as night-time or inclement weather. Lastly, a more extensive dataset with greater variety in anomaly types and environmental contexts could further refine model performance and generalizability. This continued exploration of multimodal techniques and adaptive methods will enhance the framework’s real-world applicability across surveillance, security, and automated monitoring systems.

REFERENCES

[1] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 935-942.

[2] W. Sultani, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 6479-6488.

[3] T. Wu, C. Y. Suen, and D. Kruse, “Multiple abnormal event detection in videos using deep learning,” Pattern Recognit., vol. 93, pp. 53-68, 2019.

4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguistics (NAACL), 2019, pp. 4171-4186.

[5] T. Brown, B. Mann, N. Ryder, et al., “Language models are few-shot learners,” in Advances Neural Inf. Process. Syst. (NeurIPS), 2020, pp. 1877-1901.

[6] L. Xu, C. Liu, Z. Feng, and M. Yu, “Enhancing video surveillance with natural language processing: Integrating LLMs for semantic anomaly detection,” IEEE Trans. Multimedia, vol. 24, no. 5, pp. 1273-1282, 2022.

[7] C. G. Snoek, K. E. van de Sande, O. de Rooij, and M. Worring, “The challenge problem for automated detection of anomalous events in video,” in Proc. ACM Int. Conf. Multimedia Retrieval (ICMR), 2020, pp. 1-7.

[8] M. Chen, W. Fang, Y. Y. Tang, and J. Zhang, “Multi-modal anomaly detection in videos via attention mechanism,” IEEE Trans. Image Process., vol. 29, no. 2, pp. 5573-5582, 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in Proc. 2017 Conf. Advances Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 5998-6008.

[10] P. Wu, F. Cheng, and Z. Liu, “Smart surveillance systems in public spaces: A case study on real-time anomaly detection,” IEEE Access, vol. 8, pp. 129841-129852, 2020.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learning Representations (ICLR), 2021.

[12] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” in Proc. Int. Conf. Machine Learning (ICML), 2021.

[13] Lu, H., Jian, H., Poppe, R., & Salah, A. A. (2024). “Enhancing Video Transformers for Action Understanding with VLM-aided Training,” arXiv preprint arXiv:2403.16128.

[14] H. Lv and Q. Sun (2024) “Video Anomaly Detection and Explanation via Large Language Models,” arXiv preprint arXiv:2401.05702

[15] Singh, S., Dewangan, S., Krishna, G. S., Tyagi, V., Reddy, S., & Medi, P. R. (2022). “Video Vision Transformers for Violence Detection” [arXiv preprint arXiv:2209.03561v2]

[16] Zhao, Y., Misra, I., Kr¨ahenb¨uhl, P., & Girdhar, R. (2022). “Learning Video Representations from Large Language Models,” arXiv preprint arXiv:2212.04501

[17] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu and K. Barnard, “Attentional Feature Fusion,” in 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2021, pp. 3559-3568, doi: 10.1109/WACV48630.2021.00360

[18] T. Baltruˇsaitis, C. Ahuja and L. -P. Morency, ”Multimodal Machine Learning: A Survey and Taxonomy,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 423-443, 1 Feb. 2019, doi: 10.1109/TPAMI.2018.2798607

APPENDIX A: DEFINITIONS, ACRONYMS, ABBREVIATIONS

Anomaly Detection: The process of identifying data patterns that deviate from expected norms, essential for detecting unusual or suspicious events in video data, such as violence, vandalism, or abandoned objects.

TwelveLabs SDK: Software Development Kit for automated video transcription and caption generation, enabling preliminary text-based feature extraction from video input.

TimeSformer (Time-Shifted Transformer): A model optimized for video processing that applies spatiotemporal attention mechanisms, enabling detailed feature extraction from both spatial and temporal aspects of video frames.

RoBERTa (A Robustly Optimized BERT): An NLP model derived from BERT, used here for generating context-aware text embeddings to analyze video captions for relevant features.

KeyBERT: A keyword extraction tool that utilizes embeddings from BERT-based models (e.g., RoBERTa) to identify critical terms within a document, aiding in summarizing the main topics within video captions.

LDA (Latent Dirichlet Allocation): A probabilistic topic modeling technique for discovering abstract topics within a collection of documents, here applied to video caption text for extracting thematic features.

CBP (Compact Bilinear Pooling): A technique for efficient multimodal feature fusion, combining high-dimensional feature vectors from different sources while preserving interactions between features.

NN (Neural Network): A machine learning model that consists of layers of interconnected nodes (or "neurons") to process data and identify patterns. In this project, a neural network is used to classify fused video and text features as anomalous or not.

ML (Machine Learning): The branch of AI focused on algorithms and statistical models that allow computers to learn from data to make decisions or predictions without explicit programming for every task.

NLP (Natural Language Processing): A subfield of AI focused on enabling computers to process and analyze human language. Here, NLP techniques are used to interpret and process captions generated from video inputs.

FPS (Frames Per Second): A metric used to determine the speed at which frames are captured in a video, essential for standardized frame sampling during video preprocessing.

AFF: Attentional Feature Fusion

CNN (Convolutional Neural Network): A deep learning architecture primarily used for image and video processing, capable of capturing spatial hierarchies in visual data. Though TimeSformer is the main model, CNNs may be referenced for comparative approaches in video feature extraction.

GPU (Graphics Processing Unit): Hardware optimized for high-speed parallel processing, often used in ML to accelerate intensive computations involved in model training and data processing.

TPU (Tensor Processing Unit): A specialized hardware designed by Google for accelerating machine learning workloads, particularly helpful for complex tasks like video processing and neural network training.

API (Application Programming Interface): A set of protocols and tools allowing communication between software components. In this context, APIs are used to integrate external libraries and services like TwelveLabs SDK and KeyBERT.

ETL (Extract, Transform, Load): A data processing workflow involving data extraction, transformation, and loading into a target system or model. ETL processes support data preparation for text and video feature extraction.

SOTA (State of the Art): Refers to the highest level of development achieved in a particular field or task, representing the most advanced techniques and models, such as TimeSformer for video feature extraction.

IoU (Intersection over Union): A performance metric used in computer vision tasks for measuring the accuracy of object detection and segmentation. Though not directly applied here, IoU may be referenced for anomaly bounding tasks in future improvements.

VIDEX (Video Dataset for Anomaly Exploration): A novel dataset proposed to tackle anomaly detection tasks.

Ria NNAR - Phase II Edited.pdf

ORIGINALITY REPORT

3%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repositorio.unal.edu.co Internet Source	1%
2	www.catalyzex.com Internet Source	1%
3	ijsrset.com Internet Source	<1%
4	Nejad, Sareh Soltani. "Weakly-Supervised Anomaly Detection in Surveillance Videos Based on Two-Stream I3D Convolution Network", The University of Western Ontario (Canada), 2023 Publication	<1%
5	Submitted to PES University Student Paper	<1%
6	Submitted to Institute of Management Technology Student Paper	<1%
7	arxiv.org Internet Source	<1%

8	listens.online Internet Source	<1%
9	pergamos.lib.uoa.gr Internet Source	<1%
10	www.studymode.com Internet Source	<1%
11	Taewan Kim, Sangyeop Kim, Jaeyoung Kim, Yeonjoon Lee, June Choi. "Toward Better Ear Disease Diagnosis: A Multi-Modal Multi-Fusion Model Using Endoscopic Images of the Tympanic Membrane and Pure-Tone Audiometry", IEEE Access, 2023 Publication	<1%
12	deepai.org Internet Source	<1%
13	Chenghao Li, Xinyan Yang, Gang Liang. "Keyframe-guided Video Swin Transformer with Multi-path Excitation for Violence Detection", The Computer Journal, 2023 Publication	<1%
14	Submitted to University of Wales central institutions Student Paper	<1%
15	docplayer.net Internet Source	<1%
	pure.unamur.be	

16	Internet Source	<1%
17	www.morningdough.com Internet Source	<1%
18	www.slideshare.net Internet Source	<1%
19	K. Deepak, L. K. P. Vignesh, G. Srivathsan, S. Roshan, S. Chandrakala. "Chapter 21 Statistical Features-Based Violence Detection in Surveillance Videos", Springer Science and Business Media LLC, 2020 Publication	<1%
20	Kuldeep Singh, K Yamini Preethi, K Vineeth Sai, Chirag N. Modi. "Designing an Efficient Framework for Violence Detection in Sensitive Areas using Computer Vision and Machine Learning Techniques", 2018 Tenth International Conference on Advanced Computing (ICoAC), 2018 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		