Delta University for Sciences and Technology
Faculty of Artificial Intelligence
Artificial Intelligence Department

Deepfake Video Detection Based on Vision Temporal Transformer (VTT) and Multi-Head Attention (MHA)

AI317 - Work-based Professional Project in Artificial Intelligence (I)

Prepared by:

Tawfek Mohamed Tawfek	**4231005**
Adham Osama Alrifaie	**4232010**
Basmala Hashim Abdelrahman	**4231371**
Arwa Hisham Abdelaziz	**4231468**
Mariam Salah Aldin AlSayed	**4231469**
Malak Arfa Hussien	**4231503**

Supervised by:

Prof. Dr. Hanaa Salem Marie
Vice Dean of
The Faculty of Artificial Intelligence for Education and Student Affairs

Fall Semester
2025/2026

Contents

Scan to access the Project Source Code on GitHub

Link:
https://github.com/TawfekMohamed-7/Deepfake-Video-Detection-Based-on-Vision-Temporal-Transformer-VTT-and-Multi-Head-Attention-MHA-

Deepfake Video Detection Based on Vision Temporal Transformer (VTT) and Multi-Head Attention (MHA) "Deepfake"

Abstract

Deepfake technology has become one of the most serious challenges in the field of digital media security. With the rapid advancement of generative models, manipulated videos are increasingly realistic, making it difficult for humans and traditional detection techniques to identify forged content. This project presents an end-to-end deep learning framework designed to detect deepfake videos through combined spatial, temporal, and motion-based analysis.

The proposed system begins by reading all frames of the input video. Optical Flow is then applied to measure motion consistency and select the most informative frame candidates. From these, the best frames are chosen for deeper analysis. A YOLO-based face detection module crops and normalizes facial regions to ensure consistent input quality.

To capture both spatial and temporal dependencies, the selected frames are processed using a Vision Temporal Transformer (VTT), which learns relationships between frames while maintaining global context.

In addition, inter-frame residues are computed using the LIPINC method, enabling the system to detect subtle artifacts and motion inconsistencies characteristic of deepfake generation. These spatial and residue features are then enhanced using a Multi-Head Attention mechanism to strengthen the feature representation.

The fused features are finally passed through a classification layer using Softmax to determine whether the video is Real or Fake. The strength of this framework lies in integrating different feature sources—frame features, motion clues, and temporal cues—making the detection process more robust against modern deepfake techniques.

This work contributes to improving the reliability of automated deepfake detection systems, which play a crucial role in combating misinformation, protecting public trust, preventing identity manipulation, and enhancing digital media forensics.

المحول الزمني البصري VTT وآلية الانتباه متعدد الرووس MHA

"Deepfake"

تُعد تقنية الديب فيك واحدة من أخطر التحديات التي تواجه أمن الوسائط الرقمية في الوقت الحالي، خاصة مع التطور الكبير في نماذج التوليد التي أصبحت قادرة على إنتاج فيديوهات شديدة الواقعية يصعب على البشر أو التقنيات التقليدية تمييزها. يقدم هذا المشروع إطار عمل متكامل يعتمد على التعلّم العميق لاكتشاف فيديوهات الديب فيك من خلال دمج التحليل المكاني(Spatial) ، الزمني(Temporal) ، وتحليل الحركة.

يبدأ النظام بقراءة جميع إطارات الفيديو، ثم استخدام تقنية Optical Flow لقياس اتساق الحركة واختيار الإطارات الأكثر معلوماتية. بعد ذلك، يتم تحديد أفضل الإطارات لتحليلها بعمق. يتم استخدام نموذج YOLOلاكتشاف الوجه وقصه ومعالجته لضمان إدخال بيانات موحدة وقابلة للاستخدام في مراحل التحليل التالية.

ولالتقاط العلاقات بين الإطارات، يعتمد النموذج على Vision Temporal Transformer (VTT)الذي يتعلّم الترابط الزمني والبُعد السياقي لكل إطار. إضافةً إلى ذلك، يتم حساب الفروقات بين الإطارات المتتابعة باستخدام LIPINC بهدف اكتشاف الاختلافات الدقيقة واللاأخطاء الحركية التي تُعد من العلامات المميزة للفيديوهات المُولدة بالديب فيك. بعدها تُدمج خصائص الإطارات وخصائص الفروقات داخل آلية Multi-Head Attentionلتعزيز التمثيل النهائي للخصائص.

وفي المرحلة النهائية، يقوم مصنف يعتمد على Softmaxبتحديد ما إذا كان الفيديو حقيقيًا أم مزيفًا. تتميز هذه المنهجية بدمج مصادر متعددة من الخصائص— خصائص مكانية، زمنية، وحركية—مما يجعل أداء النظام أكثر قوة في مواجهة تقنيات التزييف الحديثة.

يسهم هذا المشروع في تطوير أنظمة موثوقة لاكتشاف التزوير العميق، بما يدعم مواجهة المحتوى المضلل، وحماية الهوية، وتعزيز الثقة الرقمية، ودعم جهود التحليل الجنائي للمحتوى المرئي.

Chapter 1
Introduction

1. Introduction

Deepfake technology has witnessed rapid development in recent years, leading to the generation of highly realistic manipulated videos that are often difficult to distinguish from authentic ones using human observation alone[1][2]. These manipulated videos may closely resemble real content in terms of appearance, yet they frequently contain subtle visual artifacts and temporal inconsistencies that do not fully conform to natural facial motion patterns[2]. Such inconsistencies are usually difficult to detect when analyzing individual frames in isolation, which increases the challenge of reliable deepfake detection.

To address this problem, this project proposes a structured deepfake detection framework that analyzes video content through multiple sequential processing stages. The framework is designed to leverage spatial information, temporal relationships, and motion-based cues in a unified manner, allowing the system to capture both visible and hidden artifacts introduced during deepfake generation.

The proposed framework begins by extracting all frames from the input video to preserve the complete temporal information. Optical Flow is then applied to analyze motion patterns between consecutive frames and to identify the most informative frames that contain meaningful temporal changes. This step helps reduce

redundancy while ensuring that critical motion information is retained. After frame selection, a YOLO-based face detection module is employed to locate, crop, and normalize facial regions, which ensures consistent and focused input for subsequent processing stages.

The selected facial frames are then processed using a Vision Temporal Transformer (VTT), which is responsible for extracting spatial–temporal features that describe both visual appearance and relationships across frames. In parallel, inter-frame residues are computed using the LIPINC method, enabling the system to capture fine-grained inconsistencies that commonly arise in manipulated videos. These residues highlight subtle frame-to-frame variations that are difficult to observe directly.

Both frame-based features and residue-based features are further enhanced using a Multi-Head Attention mechanism. This mechanism allows the model to focus on the most relevant spatial and temporal patterns within the video. Finally, a Softmax classification layer produces the final decision, classifying the input video as either real or fake.
Overall, this introduction highlights the motivation behind the proposed framework and provides an overview of its processing pipeline, emphasizing the importance of combining motion cues, spatial features, and temporal relationships to improve the reliability of deepfake video detection[3][4].

1.2 Project Overview

This project presents a deepfake detection system specifically designed to analyze video content through a structured multi-stage processing pipeline. The system aims to identify subtle visual distortions and temporal inconsistencies that are commonly introduced during video manipulation. By combining motion analysis, face-focused preprocessing, and advanced feature extraction techniques, the system seeks to improve the robustness and accuracy of deepfake detection [1].

The workflow starts with reading all frames of the input video to preserve temporal continuity. To reduce unnecessary redundancy and focus on the most informative content, Optical Flow is applied to estimate motion between frames and select a subset of candidate frames. From these candidates, the most relevant frames are chosen based on their motion characteristics, ensuring that the system concentrates on frames that carry meaningful temporal information.

Once the frames are selected, a YOLO-based detection module is used to locate the facial region in each frame. The detected faces are cropped and normalized to a unified size, providing consistent input data for the model. This step is crucial, as accurate face localization directly affects the quality of extracted features and the reliability of subsequent analysis.

The normalized facial frames are then passed to a Vision Temporal Transformer (VTT), which extracts spatial and temporal

features while modeling relationships across frames. In parallel, residue features are computed using the LIPINC method to capture subtle frame-to-frame differences that may result from manipulation. These two feature streams—frame features and residue features—are combined and refined using a Multi-Head Attention mechanism, enhancing the overall feature representation.

Finally, the enhanced features are passed to a Softmax-based classifier that determines whether the input video is real or fake. This structured approach enables the system to leverage complementary sources of information, resulting in a more robust and reliable deepfake detection process.

1.3 Problem Statements

Deepfake videos have become increasingly difficult to detect due to the subtle visual changes and temporal inconsistencies they introduce [1][2]. Traditional detection methods often rely primarily on spatial analysis of individual frames[3][4], which limits their ability to capture motion irregularities and temporal distortions that occur between consecutive frames. As a result, these systems may fail when manipulated content appears visually convincing.

Additionally, treating all video frames equally introduces redundancy and increases computational overhead, while inaccurate face detection and localization further degrade detection performance. Many existing approaches also struggle to represent subtle residue variations caused by frame-to-frame manipulation, which reduces classification reliability.

These limitations highlight the need for a comprehensive deepfake detection framework that integrates motion-based frame selection, precise facial region extraction, temporal–spatial feature modeling, and residue analysis. Such a framework is necessary to improve detection accuracy and provide a more reliable distinction between real and fake video content.

1.4 Objectives of the Project

The main objective of this project is to develop a structured and reliable framework capable of detecting deepfake videos by combining spatial, temporal, and motion-based analysis. The specific objectives of the project are outlined as follows:

- To identify informative frames using motion analysis:
  Utilize Optical Flow to select the most relevant frames from the video, reducing redundancy while preserving meaningful motion patterns.

- To accurately detect and extract facial regions:
  Apply a YOLO-based face detection module to crop and normalize faces, ensuring consistent and high-quality inputs for further processing.

- To capture spatial and temporal relationships across frames:
  Use a Vision Temporal Transformer (VTT) to extract features that reflect both visual appearance and temporal dependencies.

- To detect subtle inconsistencies through residue analysis: Employ the LIPINC method to compute inter-frame residues that reveal fine variations commonly associated with deepfake manipulation.

- To enhance feature representation through attention mechanisms:
  Integrate Multi-Head Attention to strengthen combined spatial and residue features before classification.

- To classify video content as real or fake:
  Use a Softmax-based classification layer to produce an interpretable and reliable decision regarding video authenticity.

Together, these objectives contribute to building a robust deepfake detection system with improved accuracy and reliability.

1.5 Scope of the Project

The scope of this project is limited to developing and evaluating a deepfake detection framework that analyzes video content using spatial, temporal, and motion-based features. The system focuses specifically on identifying inconsistencies in facial regions and variations across consecutive frames.

The project includes the following elements:

- Video Frame Extraction.
- Motion-Based Frame Selection.

- Face Detection and Preprocessing.
- Spatial–Temporal Feature Extraction.
- Residue Computation.
- Feature Fusion and Enhancement.
- Classification.

The project scope does **not** cover dataset creation, deepfake generation, or deployment as a real-time application unless specified in other chapters. It focuses solely on implementing and evaluating the described detection pipeline.

Chapter 2

Background and Challenges

2.1 Introduction

This chapter presents the background and foundational concepts related to the proposed deepfake detection framework. With the rapid progress of deepfake generation techniques, manipulated videos have become increasingly realistic, making it difficult to detect forgery through simple visual inspection [2]. Such videos often contain subtle spatial artifacts and temporal inconsistencies that are not easily noticeable in individual frames but become evident when motion and frame-to-frame relationships are analyzed.

Understanding the overall processing pipeline and the role of each system stage is therefore essential to explain how the proposed framework addresses these challenges. The system relies on a structured sequence of operations that include motion-based frame selection, accurate facial region extraction, temporal–spatial feature modeling, and residue computation. Each of these stages contributes specific information that supports the detection process.

This chapter provides a conceptual overview of these components and clarifies how they interact to form a unified detection pipeline. By presenting the background and motivation behind each processing step, this chapter establishes the foundation

required to understand the detailed system design and implementation discussed in the following chapters.

2.2 Project Background

Deepfake technology has rapidly evolved in recent years, allowing facial appearances, expressions, and movements within videos to be manipulated in a highly realistic manner [1][2]. These manipulations are often generated using advanced learning-based techniques that aim to preserve visual quality and temporal coherence, making fake videos increasingly difficult to distinguish from authentic ones.

Early deepfake detection approaches mainly focused on analyzing individual video frames. While such spatial-based methods can detect visible artifacts, they often fail to capture subtle temporal inconsistencies that appear across consecutive frames [3][4]. As a result, these approaches struggle when manipulated videos appear visually convincing on a frame-by-frame basis.

The proposed framework is motivated by the observation that deepfake videos exhibit inconsistencies not only in spatial appearance but also in motion patterns and temporal transitions. These inconsistencies may manifest as unnatural facial movements or irregular changes between frames that deviate from normal facial behavior. To address this, the framework adopts a multi-stage design that explicitly considers motion, spatial information, and temporal relationships.

The system begins by extracting frames from the input video, followed by applying Optical Flow to identify frames that contain meaningful motion information. This step helps reduce redundancy and ensures that only informative frames are selected for further processing. A YOLO-based face detection module is then used to accurately locate and crop facial regions, ensuring consistent and reliable inputs for feature extraction.

To model relationships across time, the framework incorporates a Vision Temporal Transformer (VTT), which captures both spatial features and temporal dependencies between frames. In parallel, the LIPINC method is used to compute residues between consecutive frames, highlighting fine variations that often result from manipulation. These complementary features are then combined and enhanced using a Multi-Head Attention mechanism, improving the system's ability to distinguish between real and fake videos.

The background of this project therefore lies in the need for a motion-aware, temporally guided, and structured detection framework capable of addressing the limitations of traditional approaches.

2.2.1 Experimental Setup

The experimental setup is designed to evaluate the effectiveness of the proposed deepfake detection framework under a controlled and structured processing environment. The system receives video samples as input and processes them through a

predefined sequence of stages that analyze facial regions, motion information, and temporal variations.

Each experiment follows a consistent preprocessing procedure to ensure fair and reliable evaluation. This includes extracting frames from the video, normalizing facial regions, and organizing the data in a structured format suitable for feature extraction. Maintaining consistent preprocessing is essential to reduce variability and ensure that performance differences are attributed to the detection framework itself rather than data inconsistencies.

The extracted features from the spatial–temporal analysis and residue computation stages are then passed to the classification module. The system's performance is evaluated based on its ability to correctly distinguish between real and manipulated videos. The experimental setup therefore serves as a validation environment for assessing the reliability and effectiveness of the proposed framework. Detailed architectural design and processing stages are further explained in the next chapter.

2.3 System Components

The proposed deepfake detection system is composed of several interconnected components that collectively form the detection pipeline. Each component plays a specific role in transforming the raw video input into a meaningful classification decision.

The system begins with a video input component that supplies video data to the framework. This is followed by a preprocessing component responsible for preparing the video for analysis, including frame extraction and normalization. Proper preprocessing ensures that the data is consistent and suitable for subsequent processing stages.

After preprocessing, the feature extraction component analyzes the prepared video data to obtain informative representations related to facial appearance, motion patterns, and temporal relationships. These features serve as the core inputs for the final decision-making process. The classification component then uses the extracted features to determine whether the video is real or fake.

Together, these components define the overall structure of the system. While this chapter provides a high-level description of their roles, the detailed operations and implementation of each component are discussed in the following chapter.

2.4 Challenges

Deepfake video detection presents several challenges due to the increasing realism of manipulated content. One major challenge is that deepfake videos often appear visually authentic[1][2], making it difficult to identify manipulation based solely on visual cues. Variations in lighting, facial expressions, and video quality further complicate the detection process.

Another significant challenge is handling temporal information across video frames. Manipulation artifacts may be subtle and distributed over multiple frames, requiring effective modeling of temporal dependencies to detect inconsistencies. Inaccurate or unstable preprocessing can also negatively affect detection performance, as errors in face localization or normalization can propagate through the system[4].

Additionally, extracting features that effectively represent both spatial appearance and temporal behavior remains a complex task. If these features fail to capture meaningful differences between real and fake videos, the classification results may become unreliable. These challenges emphasize the importance of a structured, motion-aware, and temporally guided framework, which the proposed system aims to address.

Chapter 3
Proposed Framework and Methodology

3.1 Introduction to the Framework

This chapter presents the proposed deepfake detection framework based strictly on the designed pipeline illustrated in the framework diagram. The purpose of this chapter is to describe the working principle of the system and explain how the different processing stages interact to produce the final classification output. Each stage described in this chapter directly corresponds to a block in the framework, ensuring consistency between the conceptual design and the implemented methodology.

The framework is designed to process video input in a structured and sequential manner, starting from raw video frames and ending with a binary classification decision indicating whether the video is Real or Fake [1][2]. The chapter focuses on explaining the role of each module, the flow of data between stages, and the motivation behind organizing the system in this multi-stage form.

By following the framework step by step, this chapter provides a clear understanding of how motion information, facial regions, spatial–temporal features, and residual inconsistencies are progressively extracted and refined. This structured explanation prepares the reader for understanding the experimental results and evaluation discussed in later chapters.

3.2 System Methodology

The proposed system operates by processing an input video through a sequence of well-defined and interdependent stages. Each stage is designed to extract specific information that contributes to the overall detection decision. The methodology ensures that the video data is gradually transformed from raw frames into discriminative features suitable for classification.

Initially, all frames are extracted from the input video to preserve the complete temporal information. Since processing all frames can introduce redundancy, a frame selection mechanism based on Optical Flow is applied to identify frames that contain the most informative motion patterns. This step reduces computational complexity while retaining critical temporal cues.

The selected frames are then passed through a YOLO-based face detection and preprocessing stage, which isolates facial regions and ensures consistent input dimensions. Following preprocessing, spatial and temporal features are extracted and modeled using a Vision Temporal Transformer (VTT). The VTT is further enhanced using a Multi-Head Attention (MHA) mechanism to improve feature representation by emphasizing important temporal and spatial relationships.

Finally, the refined features are passed to a SoftMax-based classification layer, which outputs the final decision indicating

whether the input video is Real or Fake[1][2]. Figure 3.1 illustrates the complete processing pipeline of the proposed framework.

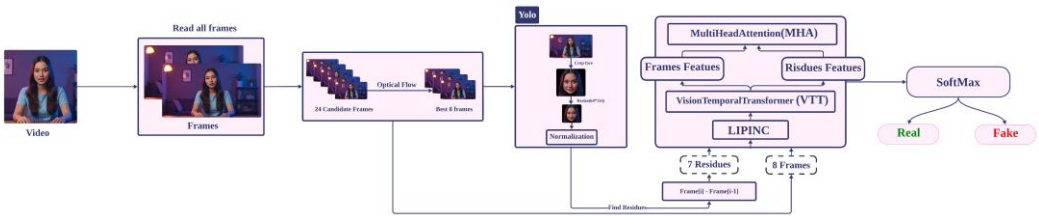

Figure 3.1: The proposed framework for deepfake video detection

3.2.1 Frame Extraction and Selection Method

The input video is first decomposed into individual frames by reading all frames sequentially. This step ensures that the temporal structure of the video is preserved and that no motion information is lost at an early stage. From the complete set of extracted frames, a group of candidate frames is generated for further analysis.

Optical Flow is then applied to analyze motion information between consecutive frames. This method estimates pixel-level motion and highlights areas where significant changes occur over time. Based on the Optical Flow analysis, frames that contain meaningful motion patterns are identified as more informative for deepfake detection.

According to the framework design, 24 candidate frames are initially considered. From these candidates, the best 8 frames are selected based on their motion relevance[2][4]. This selection strategy reduces redundancy while ensuring that the retained frames

capture essential temporal dynamics for subsequent processing stages.

3.2.2 Face Detection and Preprocessing Method

Once the most informative frames are selected, each frame is processed using a YOLO-based face detection model. The primary goal of this stage is to accurately locate and isolate the facial region from each frame. Focusing on facial regions allows the system to concentrate on areas where deepfake artifacts are most likely to appear.

After detecting the face, the facial region is cropped from the frame and resized to a fixed resolution of 64×114 pixels. This resizing step ensures uniform input dimensions across all frames, which is essential for stable feature extraction. Normalization is then applied to standardize pixel value distributions, reducing variations caused by lighting or contrast differences [1][2].

This preprocessing stage prepares the facial data for consistent and reliable feature extraction in the subsequent stages of the framework.

3.3 Framework Modules

This section describes the main modules of the proposed system as illustrated in the framework diagram. Each module performs a specific function and passes its output to the next stage, forming a continuous processing pipeline.

3.3.1 Video Input Stage

The video input stage is responsible for receiving the input video and converting it into a sequence of frames. This module serves as the entry point of the framework. The output of this stage is a complete set of frames extracted from the video, which preserves the original temporal order required for motion and temporal analysis.

3.3.2 Optical Flow–Based Frame Selection Stage

This module applies Optical Flow analysis to consecutive frames to estimate motion patterns across time. Its primary function is to identify frames that contain significant temporal changes. By analyzing motion intensity and variation, the module selects the most informative frames.

The output of this stage is a reduced subset of frames, specifically the best 8 frames selected from the initial candidates. These frames are forwarded to the face detection stage, ensuring that subsequent processing focuses on the most relevant temporal segments of the video.

3.3.3 YOLO-Based Face Detection Stage

In this stage, the YOLO model is applied to each selected frame to detect facial regions. The detected face is cropped from the background, effectively isolating facial information while removing irrelevant scene content.

This focused approach improves the quality of extracted features and ensures that the analysis is concentrated on regions where manipulation artifacts are more likely to appear.

3.3.4 Preprocessing Stage

The preprocessing stage resizes the cropped face images to a fixed resolution of 64×114 pixels. Normalization is applied to standardize pixel intensities across all frames. This step ensures consistency in data representation and improves the stability of feature extraction in later stages.

3.3.5 Residual Frame Generation Stage

Residual frames are generated by computing the difference between consecutive frames using the expression Frame[i] − Frame[i−1]. This process highlights temporal inconsistencies and subtle changes that may result from deepfake manipulation.

From the selected frames, 7 residual frames are obtained. These residuals capture fine-grained temporal variations that may not be evident in the original frames alone.

3.3.6 Feature Extraction Stage

Two types of features are extracted in this stage:

• Frame Features, derived from the selected preprocessed frames.

• Residual Features, derived from the generated residual frames.

These features represent spatial and temporal characteristics of the video[1][2].

3.3.7 Vision Temporal Transformer (VTT) Stage

The Vision Temporal Transformer processes both frame features and residual features. It models temporal dependencies across frames and residuals to capture dynamic patterns present in the video sequence[2][4].

3.3.8 Multi-Head Attention (MHA) Stage

The Multi-Head Attention module enhances the feature representation by allowing the model to focus on different temporal and spatial aspects simultaneously[2]. It operates on both frame features and residual features produced by the VTT.

3.3.9 Classification Stage (SoftMax)

The final feature representation produced by the attention mechanism is passed to a SoftMax layer. This layer performs classification and outputs the final decision, labeling the input video as either Real or Fake[1][2][3].

3.4 Output Decision

The final output of the methodology is a binary classification result. Based on the SoftMax probabilities, the input video is classified as either Real or Fake, directly corresponding to the final stage shown in the framework.

Chapter 4

System Software and Design

4.1 Introduction

This chapter presents a detailed discussion of the software design and implementation aspects of the proposed deepfake detection system. The main focus of this chapter is to explain how the conceptual framework introduced in Chapter 3 is translated into a practical and well-structured software system. Particular attention is given to the organization of software components and the flow of data between different modules during video processing.

The chapter explains the logical structure of the application and clarifies how each software module contributes to the overall detection process. By clearly defining module responsibilities and interactions, the system ensures that every stage of the deepfake detection pipeline is implemented in a transparent and traceable manner.

The software implementation strictly adheres to the stages defined in the proposed framework. Each conceptual block in the framework is mapped directly to a corresponding software module. This one-to-one mapping ensures consistency between the system design and its implementation, reducing ambiguity and simplifying validation and testing.

A modular software architecture is adopted, where each processing stage is implemented as an independent and self-contained unit. These modules handle video frame extraction, motion-based frame selection, face detection and preprocessing, feature extraction, temporal modeling, and final classification. Such modularity improves readability of the system design and allows future modifications or extensions to be performed with minimal impact on the overall pipeline.

It is important to emphasize that this chapter does not introduce any additional processing stages or detection logic beyond those already defined in the framework. Instead, it focuses exclusively on explaining how the existing framework is realized at the software level to transform raw video input into a final Real or Fake decision.

4.2 System Software Architecture

The proposed deepfake detection system is implemented as a software-based application that supports both web and mobile platforms[5]. To achieve this, the system follows a client–server architecture that clearly separates user interaction from data processing and decision-making functionalities. This architectural design improves system organization, scalability, and ease of maintenance.

The frontend layer consists of two user interfaces: a web application developed using React and a mobile application developed using Flutter. These interfaces serve as access points for users, allowing them to upload video files and receive detection

results. The frontend layer is intentionally kept lightweight and does not perform any computationally intensive operations related to deepfake detection.

The backend layer represents the core processing component of the system and is implemented using FastAPI. This backend server is responsible for receiving uploaded video files, managing the execution of the detection pipeline, and generating the final classification results. By centralizing all processing tasks in the backend, the system ensures consistent detection behavior regardless of the platform used.

The separation between frontend and backend layers enhances system reliability and ensures that updates to the detection logic can be applied without requiring changes to the user interfaces. This design also supports future scalability, such as handling multiple user requests or extending the system to additional platforms.

4.3 Software Modules Description

The proposed deepfake detection system is organized into a set of well-defined software modules[5], each responsible for a specific task within the overall pipeline. These modules operate sequentially and cooperatively to convert raw video input into a final classification decision. The modular design simplifies debugging, testing, and maintenance.

4.3.1 User Interface Module

The User Interface Module represents the interaction layer of the system. It is implemented through the web and mobile applications and allows users to upload video files and view detection results. This module focuses on usability, clarity, and ease of interaction, ensuring that users can operate the system without technical knowledge of the underlying detection process.

4.3.2 Video Upload Module

The Video Upload Module manages the reception and transfer of video data from the user interface to the backend server. It ensures that video files are properly formatted, securely transmitted, and correctly forwarded to the processing pipeline.

4.3.3 Video Processing Module

The Video Processing Module controls the execution of the entire deepfake detection pipeline. It manages frame extraction, motion-based frame selection, face detection, preprocessing, feature extraction, temporal modeling, and classification. This module ensures that all stages are executed in the correct order as defined in the framework.

4.3.4 Deepfake Detection Module

The Deepfake Detection Module applies the trained detection model to the extracted features. It performs the final analysis and

determines whether the input video is real or fake. In addition, this module computes the probability score associated with the classification result.

4.3.5 Result Output Module

The Result Output Module is responsible for delivering the final classification results to the user interface. The output includes the predicted label (Real or Fake) and the corresponding probability score, which are presented in a clear and interpretable format.

4.4 Web Application Design

The web application serves as one of the primary user interfaces of the proposed deepfake detection system[6]. It is developed using React and is designed to provide a clean, intuitive, and accessible environment for users. The web interface enables users to submit video files for analysis and view detection results in an organized manner.

The web application prioritizes simplicity and responsiveness. Users can easily select a video file from their local device and submit it to the backend server. Once the analysis is completed, the application displays the detection result along with the probability score. All deepfake detection operations are performed exclusively on the backend server.

The communication between the web application and the backend is handled through well-defined API endpoints. This design ensures efficient data transfer while keeping the web interface lightweight and focused on presentation.

4.5 Mobile Application Design

The mobile application provides functionality equivalent to that of the web application[6], allowing users to interact with the system using mobile devices. It is developed using Flutter, enabling a consistent and unified user experience across different mobile platforms.

The mobile application allows users to upload videos, wait for backend processing, and view the final detection results. Similar to the web interface, no deepfake detection processing is performed locally on the device. All computations are handled by the backend server.

This centralized processing approach ensures consistent detection results across platforms and reduces the computational burden on mobile devices.

4.6 System Workflow

From a software perspective, the workflow of the proposed system begins with user interaction through either the web or mobile application. The user uploads a video file, which is securely transmitted to the backend server via the upload module.

Upon receiving the video, the backend initiates the deepfake detection pipeline. The video is processed through frame extraction, motion-based frame selection, face detection, preprocessing, feature

extraction, temporal modeling, and classification stages, following the sequence defined in Chapter 3.

After completing the analysis, the backend generates the final classification result and associated probability score. These results are then sent back to the frontend application, where they are displayed to the user. This structured workflow ensures a smooth and reliable transformation of input video data into an interpretable output without introducing any additional processing stages.

Chapter 5

Experimental Results and Evaluation

5.1 Introduction to Experimental Evaluation

Experimental evaluation represents a critical phase in validating the effectiveness and reliability of the proposed deepfake detection system. While the previous chapters focused on describing the framework design and software architecture, this chapter shifts the focus toward assessing how well the system performs when applied to real-world video data. The purpose of experimental evaluation is not only to measure classification accuracy, but also to analyze the behavior of the system across different stages of processing and to verify that the design decisions made in earlier chapters lead to meaningful and reliable detection outcomes.

Deepfake detection systems must be evaluated carefully due to the complex nature of manipulated video content[1][2]. Visual artifacts introduced by deepfake generation are often subtle and may vary across frames, making simple evaluation strategies insufficient. Therefore, a structured experimental evaluation is necessary to examine how the proposed system handles spatial inconsistencies, temporal irregularities, and motion-based artifacts within videos. This chapter provides a comprehensive description of the experimental process used to validate the proposed framework, starting from dataset selection and preparation, moving through

model training, and concluding with performance evaluation and result analysis.

The experimental evaluation in this project is designed to align strictly with the proposed framework described in Chapter 3. All experiments are conducted using the same processing pipeline, model architecture, and data flow previously defined. No additional modules or external processing steps are introduced at this stage. Instead, the evaluation process serves to verify that the existing framework is capable of achieving reliable deepfake detection when applied to unseen video data.

A key objective of this chapter is to demonstrate how the integration of motion-based frame selection, facial region preprocessing, temporal–spatial feature extraction, and residue analysis contributes to the overall performance of the system. By evaluating the system as a whole rather than individual components in isolation, the experimental setup reflects realistic usage conditions in which an input video is processed end-to-end to produce a final classification decision.

This chapter also emphasizes reproducibility and consistency in experimentation. All video samples undergo the same preprocessing steps, including frame extraction, motion-based frame selection, face cropping, normalization, and residual frame generation. The use of a one-time preprocessing pipeline ensures that training, validation, and testing data are processed uniformly, reducing variability that could otherwise affect evaluation results. Furthermore, the dataset is divided into training, validation, and

testing subsets in a stratified manner to preserve class distribution and ensure fair performance assessment.

Another important aspect of the experimental evaluation is the separation between training and testing phases. The model is trained using a designated training set, while validation data is used to monitor performance and guide training decisions such as early stopping and learning rate adjustment. Final performance metrics are reported only on the held-out test set, which is not exposed to the model during training. This approach provides a realistic measure of the system's generalization capability and its ability to detect deepfake videos that were not seen during training.

The evaluation process focuses on both quantitative and qualitative performance analysis. Quantitative metrics such as accuracy, precision, recall, and F1-score are used to measure classification performance for both real and fake video classes. These metrics provide insight into how well the system balances detection sensitivity and specificity. In addition, evaluation reports and visual performance indicators are generated to support deeper analysis of model behavior and classification confidence.

Overall, this chapter serves as a bridge between the theoretical framework and practical performance of the proposed deepfake detection system. By presenting a detailed and structured experimental evaluation, it provides evidence that the system design choices made in earlier chapters result in a robust and effective solution for distinguishing between real and manipulated videos. The following sections of this chapter describe the datasets used, the

data preparation pipeline, the training configuration, and the evaluation methodology in detail.

5.2 Dataset Description

The performance of any deepfake detection system is highly dependent on the quality, diversity, and relevance of the datasets used during training and evaluation[1][2]. Since deepfake videos exhibit a wide range of visual artifacts and manipulation patterns, selecting appropriate datasets is a crucial step in ensuring that the proposed system is evaluated under realistic and challenging conditions. In this project, experimental evaluation is conducted using two publicly available datasets: DeepFake Detection (DFD) and FaceForensics++. These datasets are widely adopted in deepfake detection research and provide a balanced mix of real and manipulated video samples containing facial content.

The datasets are used jointly to enhance the robustness of the evaluation process. By combining samples from multiple sources, the system is exposed to different manipulation styles, compression levels, and video characteristics. This helps prevent the model from overfitting to a specific dataset or manipulation technique and allows for a more reliable assessment of generalization performance.

Both datasets consist of video sequences in which facial regions play a central role. This aligns directly with the proposed framework, which focuses on face-based analysis and temporal inconsistency detection. The datasets are processed using a unified preprocessing pipeline to ensure consistency across all experiments.

No dataset-specific preprocessing rules are applied, allowing the evaluation to reflect the true capabilities of the proposed system rather than dataset-dependent optimizations.

The following subsections provide a detailed description of each dataset and explain the rationale behind their selection.

5.2.1 DeepFake Detection (DFD) Dataset

The DeepFake Detection (DFD) dataset is one of the earliest large-scale datasets designed specifically for deepfake detection research[7]. It contains a collection of real and manipulated videos featuring human faces, with a strong focus on facial identity manipulation. The dataset includes videos generated using deepfake techniques that aim to realistically replace or alter facial appearances while preserving natural head motion and expressions.

In the context of this project, the DFD dataset serves as a foundational benchmark for evaluating the proposed detection framework. The videos in this dataset exhibit a variety of manipulation artifacts, particularly in facial boundaries, texture consistency, and temporal smoothness across frames. These characteristics make the dataset suitable for evaluating motion-based frame selection and residual frame analysis, which are central components of the proposed system.

Each video in the DFD dataset is treated as a sequence of frames, from which informative transition frames are selected based on motion analysis. The dataset includes both real videos, which preserve natural facial motion and temporal coherence, and fake

videos, which may contain subtle inconsistencies across consecutive frames. This contrast allows the model to learn discriminative features that differentiate authentic facial motion patterns from manipulated ones.

The DFD dataset is particularly valuable for training and evaluation because it emphasizes temporal artifacts rather than relying solely on spatial visual cues. Since the proposed framework integrates Vision Temporal Transformers and residual frame processing, the dataset aligns well with the architectural design of the model. All selected videos from the DFD dataset undergo the same preprocessing steps, including frame extraction, face cropping, normalization, and residual computation, ensuring uniform treatment across all samples.

5.2.2 FaceForensics++ Dataset

FaceForensics++ is a widely used benchmark dataset in the field of face manipulation detection[8]. It contains a large number of videos that include both real and manipulated facial content, generated using multiple manipulation techniques. The dataset is designed to simulate realistic manipulation scenarios and includes a diverse range of facial movements, expressions, and head poses.

In this project, FaceForensics++ complements the DFD dataset by introducing additional variation in manipulation artifacts and video characteristics. While the DFD dataset focuses strongly on identity-based deepfake generation, FaceForensics++ provides a broader representation of facial manipulation patterns. This diversity

enhances the robustness of the evaluation and reduces the likelihood that the model becomes biased toward a specific manipulation style.

The FaceForensics++ videos contain high variability in terms of facial appearance, lighting conditions, and motion dynamics. These variations challenge the detection system to rely on consistent temporal and residual patterns rather than superficial visual cues. As a result, the dataset is particularly suitable for evaluating the effectiveness of the dual-stream architecture and the cross-attention fusion mechanism used in the proposed model.

Similar to the DFD dataset, all FaceForensics++ videos are processed using the same preprocessing pipeline. Motion-based frame selection is applied to identify informative frames, followed by face cropping and normalization. Residual frames are computed to capture temporal inconsistencies between consecutive frames. This consistent treatment ensures that performance comparisons across datasets are meaningful and not influenced by preprocessing differences.

5.2.3 Dataset Selection Justification

The selection of the DFD and FaceForensics++ datasets is motivated by several factors directly related to the objectives of this project. First, both datasets contain a balanced representation of real and manipulated videos, making them suitable for binary classification tasks. This aligns with the system's goal of distinguishing between authentic and deepfake videos.

Second, the datasets emphasize facial manipulation, which is the primary focus of the proposed detection framework. Since the system relies on face-centric preprocessing and temporal analysis, datasets that contain clear and consistent facial regions are essential. Both DFD and FaceForensics++ satisfy this requirement and allow the system to exploit spatial and temporal facial cues effectively.

Third, using two distinct datasets enhances the generalization capability of the evaluation[1][2]. By training and testing the model on samples derived from different sources, the evaluation setup reduces the risk of dataset-specific overfitting. This ensures that the reported performance reflects the system's ability to handle unseen manipulation patterns rather than memorizing dataset-specific artifacts.

Finally, the selected datasets are widely recognized and commonly used in deepfake detection research. Their inclusion allows the experimental results of this project to be compared with existing approaches in the literature, while still maintaining a self-contained evaluation process that relies solely on the proposed framework and preprocessing pipeline.

5.3 Data Preparation Pipeline

The data preparation pipeline plays a critical role in the overall performance of the proposed deepfake detection system. Since the system operates on video data and relies on both spatial and temporal cues[3][4], careful preprocessing is required to ensure that meaningful information is preserved while unnecessary redundancy is reduced. The objective of this pipeline is to transform

raw video inputs into structured representations that can be efficiently processed by the deep learning model.

In this project, data preparation is designed as a one-time offline process. All videos from the selected datasets are preprocessed before training and evaluation, and the resulting data is stored for reuse. This approach reduces computational overhead during training and ensures consistent preprocessing across all experimental runs. The pipeline follows the same sequence of operations defined in the proposed framework and does not introduce any additional processing stages.

The pipeline consists of several sequential steps, including frame extraction, motion-based frame selection, face cropping and normalization, residual frame generation, and data serialization. Each step is designed to serve a specific purpose within the detection framework and contributes to the overall effectiveness of the system.

5.3.1 Frame Extraction Strategy

Frame extraction is the first step in the data preparation pipeline. Since the input data consists of video sequences, it is necessary to convert each video into a series of frames that can be analyzed individually and temporally. For each input video, frames are extracted at a fixed rate to preserve the temporal structure of the video while avoiding excessive redundancy.

The extracted frames maintain the original temporal order of the video. This ordering is essential for subsequent processing stages, particularly residual frame generation and temporal

modeling. No frame-level filtering is applied at this stage; instead, all extracted frames are retained to allow motion-based analysis in the following step.

Frame extraction is performed uniformly across all videos from both datasets. This ensures that differences in video length or frame count do not introduce inconsistencies in the preprocessing pipeline. By standardizing the frame extraction process, the system ensures that all videos are treated equally before further analysis.

5.3.2 Motion-Based Frame Selection

After frame extraction, motion-based frame selection is applied to identify the most informative frames within each video. Rather than processing all frames, which may include redundant or static content, the system focuses on frames that capture significant transitions in facial motion. This strategy reduces computational complexity while preserving critical temporal information.

Motion-based selection is performed by analyzing changes between consecutive frames. Frames associated with higher motion intensity are considered more informative[4], as deepfake artifacts often become more apparent during transitions such as facial expression changes, head movements, or eye blinks. Based on this analysis, a fixed number of transition frames is selected from each video.

In this project, eight transition frames are selected per video. This number represents a balance between capturing sufficient temporal variation and maintaining a manageable input size for the

model. The selected frames are distributed across the video timeline to ensure coverage of different temporal segments rather than clustering around a single event.

This step is central to the proposed framework, as it directly supports the system's focus on temporal inconsistencies. By emphasizing motion-rich frames, the detection model is encouraged to learn features that distinguish natural facial dynamics from manipulated sequences.

5.3.3 Face Cropping and Normalization

Once the informative frames are selected, face cropping is applied to isolate the facial region from each frame. Since deepfake manipulations primarily target facial content, restricting the analysis to the face region allows the model to focus on the most relevant visual information.

For each selected frame, the facial area is detected and cropped, removing background content that does not contribute to the detection task. This step also helps reduce noise and ensures that variations in background or scene composition do not influence the model's predictions.

After cropping, the facial frames are resized to a fixed resolution of 64 × 144 pixels. This resolution is chosen to balance spatial detail with computational efficiency. Normalization is then applied to standardize pixel values across all frames, ensuring stable training behavior and consistent input distributions.

The same cropping and normalization process is applied uniformly to all frames across both datasets. This consistency ensures that the model learns dataset-independent features and that performance differences are not caused by preprocessing variations.

5.3.4 Residual Frame Generation

Residual frame generation is a key component of the data preparation pipeline and directly supports the temporal modeling strategy of the proposed system. Residual frames are designed to capture temporal inconsistencies by computing the difference between consecutive frames[3][4].

For each pair of adjacent selected frames, a residual frame is generated by subtracting pixel values of one frame from the next. These residuals highlight regions of change over time and suppress static content. In authentic videos, facial motion tends to be smooth and consistent, resulting in stable residual patterns. In contrast, manipulated videos often exhibit irregular or abrupt changes that become more pronounced in residual representations.

In this project, seven residual frames are generated for each video, corresponding to the differences between the eight selected transition frames. These residual frames are treated as a parallel input stream to the model, complementing the original frame-based input.

Residual frames are resized and normalized using the same parameters as the original frames. This ensures compatibility

between the two input streams and allows the model to process both types of information using similar architectural components.

5.3.5 Data Serialization and Storage

After preprocessing, all generated data components are organized and stored for efficient access during training and evaluation. For each video, the selected frames, residual frames, and corresponding class label are grouped together and serialized into a single data structure.

The complete preprocessed dataset is saved in a serialized file format (preprocessed_data.pkl). This file contains all processed samples from both datasets and has a total size of approximately 4.31 GB. Storing the data in serialized form eliminates the need to repeat preprocessing steps during training, significantly reducing computational overhead.

During training and evaluation, the serialized data is loaded into NumPy arrays. This format enables efficient batch processing and seamless integration with the deep learning framework used to implement the model. Labels are encoded consistently to support binary classification between real and fake videos.

By organizing the data in this structured manner, the system ensures reproducibility, consistency, and efficiency throughout the experimental evaluation process.

5.4 Training Configuration

The training configuration plays a critical role in determining the effectiveness and reliability of the proposed deepfake detection system. Given the complexity of video-based data and the presence of both spatial and temporal dependencies, careful configuration of the training process is essential to ensure that the model learns meaningful patterns rather than overfitting to noise or dataset-specific artifacts.

This section provides a detailed description of how the model is trained, including architectural organization during training, loss formulation, optimization behavior, and mechanisms used to control and stabilize learning. All configuration choices are derived directly from the proposed framework and experimental setup without introducing any external assumptions.

5.4.1 Model Architecture Overview

From a training perspective, the proposed architecture is designed to process video-level information rather than isolated frames. The dual-stream design of the model allows it to learn complementary representations from two different but related inputs: original video frames and residual frames.

During training, both streams operate simultaneously and are optimized jointly. The frame-based stream focuses on learning stable spatial representations across the selected frames,

while the residual-based stream focuses on learning temporal transitions between frames. These two representations are not treated independently; instead, they are integrated during training through attention-based fusion.

The Vision Temporal Transformer (VTT) architecture enables sequence-level learning by processing frames as ordered inputs. This design ensures that temporal ordering is preserved during training, allowing the model to associate changes across frames with potential manipulation cues. The use of transformers also allows the model to weigh the importance of different frames dynamically during training.

The cross-attention fusion stage plays a key role during optimization. By allowing interaction between frame features and residual features, the model is trained to learn correlations between spatial appearance and temporal inconsistencies. This joint learning strategy strengthens the final representation used for classification.

5.4.2 Loss Functions

The training objective is defined using a combination of losses that guide the model toward robust and consistent learning. The primary loss function is categorical cross-entropy, which directly optimizes the video-level classification output. This loss encourages the model to correctly distinguish between real and fake videos based on the final SoftMax probabilities.

In addition to classification loss, a feature consistency loss is applied to the intermediate feature representations.

This loss is designed to promote stable feature extraction across frames belonging to the same video. Since all selected frames and residuals originate from a single video sample, their extracted features should exhibit coherence.

The consistency loss acts as a regularization mechanism during training. By penalizing unstable or highly fluctuating feature representations, the model is encouraged to focus on meaningful temporal patterns rather than frame-specific noise. The weighting factor of 0.1 ensures that this auxiliary objective supports the main classification task without overpowering it.

The combined loss formulation allows the model to learn both discriminative and stable representations, which is particularly important for video-based deepfake detection where subtle inconsistencies may be distributed across multiple frames.

5.4.3 Optimization Strategy

The optimization process determines how model parameters are updated during training. The Adam optimizer is selected due to its ability to adapt learning rates individually for each parameter. This property is especially beneficial for transformer-based architectures, where different layers may converge at different speeds.

The learning rate is fixed at 0.0001 throughout the initial stages of training. This value provides a controlled optimization process that avoids large parameter updates, which could destabilize training given the temporal nature of the input data.

Training is conducted using NumPy arrays, allowing the entire dataset to be efficiently accessed during each training epoch.

Throughout training, both training and validation accuracy are monitored. Training accuracy reflects how well the model fits the training data, while validation accuracy provides an indication of generalization performance. Monitoring both metrics allows early identification of overfitting or underfitting behavior.

The optimization process continues iteratively across epochs until stopping criteria are met, ensuring that the model reaches a stable solution before evaluation.

5.4.4 Training Control Mechanisms

To ensure stable learning and prevent overfitting, several training control mechanisms are incorporated into the training process. These mechanisms operate automatically during training and adjust the process based on validation performance.

Model checkpointing is used to save the model parameters corresponding to the best validation accuracy achieved during training. This ensures that the final evaluated model represents the most effective configuration encountered during optimization.

Early stopping is applied to terminate training when validation performance no longer improves. This prevents unnecessary training epochs that may lead to overfitting and ensures efficient use of computational resources.

Learning rate reduction on plateau is used to refine the optimization process. When validation performance stagnates, reducing the learning rate allows the optimizer to make smaller, more precise updates. This helps the model converge to a better local minimum and improves final performance.

Together, these mechanisms create a controlled training environment that balances learning efficiency, model stability, and generalization capability.

5.4.5 Training Process Summary

The overall training configuration integrates architectural design, loss formulation, optimization strategy, and control mechanisms into a unified learning process. Each component contributes to ensuring that the model learns meaningful spatial and temporal representations while maintaining stability across training epochs.

By following this structured training configuration, the proposed system achieves reliable performance in distinguishing real and fake videos without introducing unnecessary complexity or external dependencies.

5.5 Evaluation Methodology

The evaluation methodology relies on multiple performance metrics to analyze the system's behavior[9].

The evaluation methodology represents a critical stage in validating the effectiveness of the proposed deepfake detection

framework. Since the primary objective of the system is to classify videos as either real or fake based on spatial, temporal, and motion-related cues, the evaluation process is designed to reflect this objective in a structured and measurable manner.

This section focuses on how the trained model is tested using unseen video data, how predictions are generated at the video level, and how the obtained results are analyzed using well-defined performance metrics. The evaluation does not involve any additional learning or adaptation stages; instead, it strictly assesses the behavior of the finalized trained model.

5.5.1 Role of Evaluation in the Proposed System

Evaluation plays a fundamental role in determining whether the proposed system meets its intended design goals. While training focuses on learning discriminative patterns from labeled data, evaluation measures how well these learned patterns generalize to new, unseen videos.

In the context of this project, evaluation is particularly important because deepfake artifacts may vary in appearance and temporal behavior across different videos. Therefore, assessing the system using a dedicated test dataset allows for a realistic estimation of its performance in practical usage scenarios.

The evaluation process ensures that the reported results are not influenced by overfitting or memorization of training samples, but instead reflect the true detection capability of the framework.

5.5.2 Test Dataset Isolation

The evaluation is conducted exclusively on the test subset obtained during the dataset splitting phase. This subset is isolated from both the training and validation data throughout the entire development process.

By maintaining this strict separation, the evaluation remains unbiased and reliable. The model has no prior exposure to the test videos during training, which guarantees that the evaluation results represent genuine generalization performance rather than learned familiarity with the data.

Each video in the test set retains its original label (real or fake) and is processed independently by the trained system.

5.5.3 Consistent Preprocessing During Evaluation

To ensure fair and consistent evaluation, all test videos undergo the same preprocessing pipeline used during training and validation. This includes frame extraction, motion-based frame selection, face detection, normalization, residual frame generation, and feature preparation.

Maintaining identical preprocessing steps prevents discrepancies that could arise from inconsistent data handling. As a result, the evaluation accurately reflects how the system performs when processing videos under the same conditions assumed during model development.

This consistency is essential for interpreting the evaluation results correctly and comparing them to training and validation outcomes.

5.5.4 Video-Level Decision Strategy

The proposed system performs classification at the video level rather than the frame level. Although individual frames and residuals contribute to feature extraction, the final decision corresponds to the entire video.

During evaluation, the selected frames and residual frames of a test video are jointly analyzed by the model. The extracted features are aggregated and passed through the classification stage, producing a single output per video.

This strategy aligns with the real-world objective of determining whether a complete video is authentic or manipulated, rather than making isolated frame-based decisions.

5.5.5 SoftMax-Based Output Interpretation

The classification stage produces a probability distribution over the two possible classes using a SoftMax layer. These probabilities represent the model's confidence in assigning the video to either the Real or Fake class.

The class associated with the higher probability value is selected as the final prediction. This probabilistic output enables not only binary classification but also confidence assessment, which is useful for understanding the certainty of the model's decisions.

Probability values closer to the extremes indicate stronger confidence, while values closer to the decision boundary indicate uncertainty.

5.5.6 Metric-Based Performance Measurement

The evaluation methodology relies on multiple performance metrics to analyze the system's behavior from different perspectives. Accuracy is used to summarize the overall correctness of predictions across all test samples.

However, since accuracy alone may not fully capture class-specific behavior, additional metrics such as precision, recall, and F1-score are used. These metrics provide deeper insight into how effectively the system detects real and fake videos individually.

By examining both per-class metrics and aggregated scores, the evaluation offers a balanced and detailed assessment of system performance.

5.5.7 Error Distribution Analysis

Beyond numerical scores, evaluation also involves analyzing how errors are distributed across classes. Misclassifications are examined to understand whether errors occur more frequently in one class than the other.

This analysis helps identify whether the system exhibits any bias toward labeling videos as real or fake. Understanding error distribution supports a more informed interpretation of the

evaluation results and highlights areas where performance may be further improved.

5.5.8 Stability of Evaluation Results

The evaluation results are obtained using fixed model weights restored from the best-performing training epoch. No additional training, fine-tuning, or parameter adjustment is performed during evaluation.

This approach ensures stability and repeatability of the reported results. If the evaluation is repeated under the same conditions, the outcomes remain consistent, reinforcing the reliability of the evaluation process.

5.5.9 Summary of Evaluation Methodology

In summary, the evaluation methodology provides a structured framework for assessing the proposed deepfake detection system. By using a dedicated test dataset, consistent preprocessing, video-level decision making, and comprehensive performance metrics, the evaluation process delivers a clear and reliable measure of system effectiveness.

This detailed evaluation framework forms the basis for presenting and interpreting the quantitative and qualitative results discussed in the subsequent sections of this chapter.

5.6 Quantitative Results

This section presents the quantitative performance results obtained from evaluating the proposed deepfake detection system on the test dataset. The goal of this analysis is to provide a detailed numerical assessment of the system's ability to distinguish between real and manipulated videos using the defined evaluation metrics.

The reported results are derived exclusively from the test set and are based on the final model weights restored from the best-performing training epoch. These results reflect the generalization capability of the proposed framework when applied to unseen video data.

5.6.1 Overall Classification Performance

The overall performance of the proposed system is measured using classification accuracy, which represents the proportion of correctly classified videos out of the total number of test samples.

The system achieves an **overall accuracy of 85.57%**, indicating that a significant majority of the test videos are correctly classified as either real or fake. This result demonstrates that the combination of motion-based frame selection, facial region analysis, temporal modeling, and residue-based feature extraction contributes effectively to reliable video-level classification.Accuracy serves as a high-level indicator of system performance and provides an initial understanding of the detection capability of the proposed framework.

5.6.2 Class-Wise Performance Analysis

While overall accuracy provides a general summary, a deeper analysis is required to understand how the system behaves for each class individually. Therefore, the evaluation includes class-wise performance metrics for both real and fake video samples.

This analysis is particularly important in deepfake detection tasks, where misclassification of either class can have different implications depending on the application context.

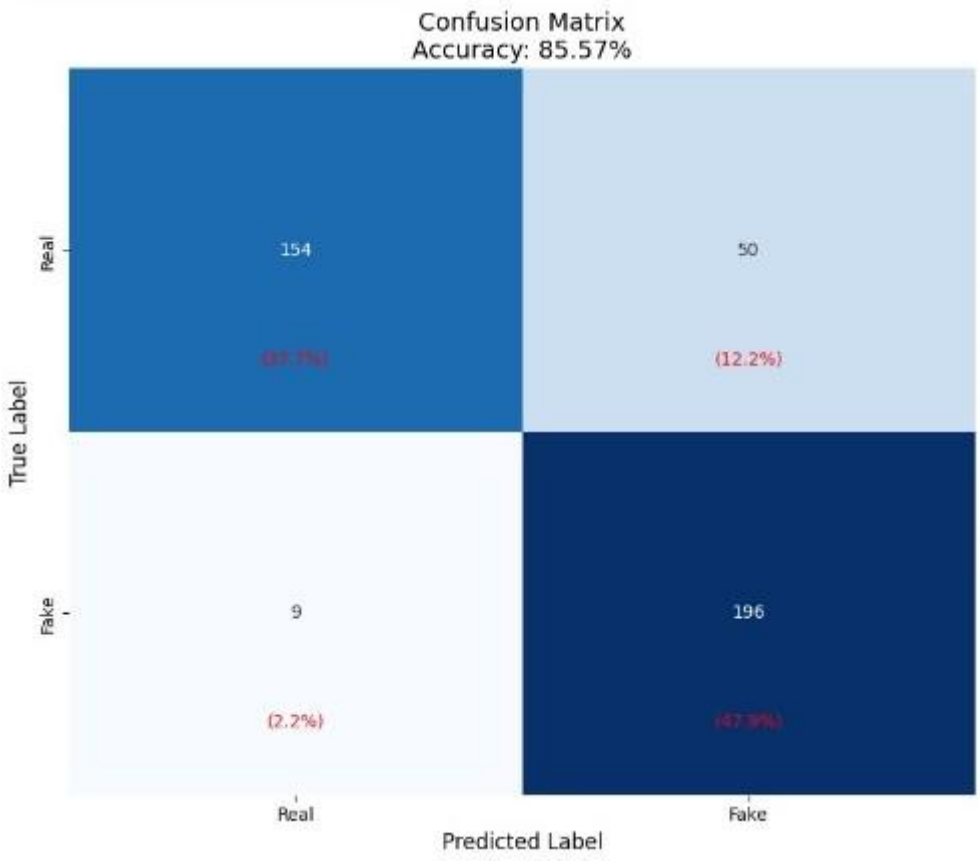

Figure 5.1: Confusion Matrix of the proposed system

5.6.3 Performance on Real Video Class

For the **Real class (label 0)**, the system achieves a **precision of 0.9448**, indicating that when the system predicts a video as real, it is highly likely to be correct. This high precision reflects the system's strong ability to avoid incorrectly labeling fake videos as real.

The **recall value of 0.7549** shows that a substantial portion of actual real videos are correctly identified by the system. Although some real videos are misclassified, the recall value indicates that the majority are successfully detected.

The **F1-score of 0.8392** balances precision and recall, providing a combined measure of the system's effectiveness in identifying real videos. This score demonstrates that the system maintains reliable performance on authentic content.

5.6.4 Performance on Fake Video Class

For the **Fake class (label 1)**, the system achieves a **precision of 0.7967**, indicating that most videos classified as fake are indeed manipulated. This shows that the system maintains reasonable confidence when detecting fake content.

The **recall value of 0.9561** highlights the system's strong ability to correctly identify manipulated videos. This high recall indicates that the majority of fake videos in the test set are successfully detected.

The **F1-score of 0.8692** reflects a balanced performance between precision and recall for the fake class, emphasizing the system's effectiveness in identifying manipulated content.

5.6.5 Macro-Averaged Performance Evaluation

To evaluate the system's performance across both classes in a balanced manner, the **macro-averaged F1-score** is computed. This metric gives equal importance to each class, regardless of the number of samples.

The system achieves a **macro-averaged F1-score of 0.8542**, indicating consistent performance across real and fake video classes. This result suggests that the system does not favor one class disproportionately and maintains balanced detection behavior.

5.6.6 Interpretation of Precision–Recall Behavior

The observed precision and recall values reveal important insights into the system's classification behavior. The high recall for fake videos indicates that the system is particularly effective at detecting manipulated content, which is a critical requirement in deepfake detection applications.

At the same time, the high precision for real videos demonstrates that the system is cautious in labeling content as authentic, reducing the risk of falsely accepting manipulated videos as real.

This trade-off reflects the system's design emphasis on capturing subtle temporal and spatial inconsistencies associated with deepfake manipulation.

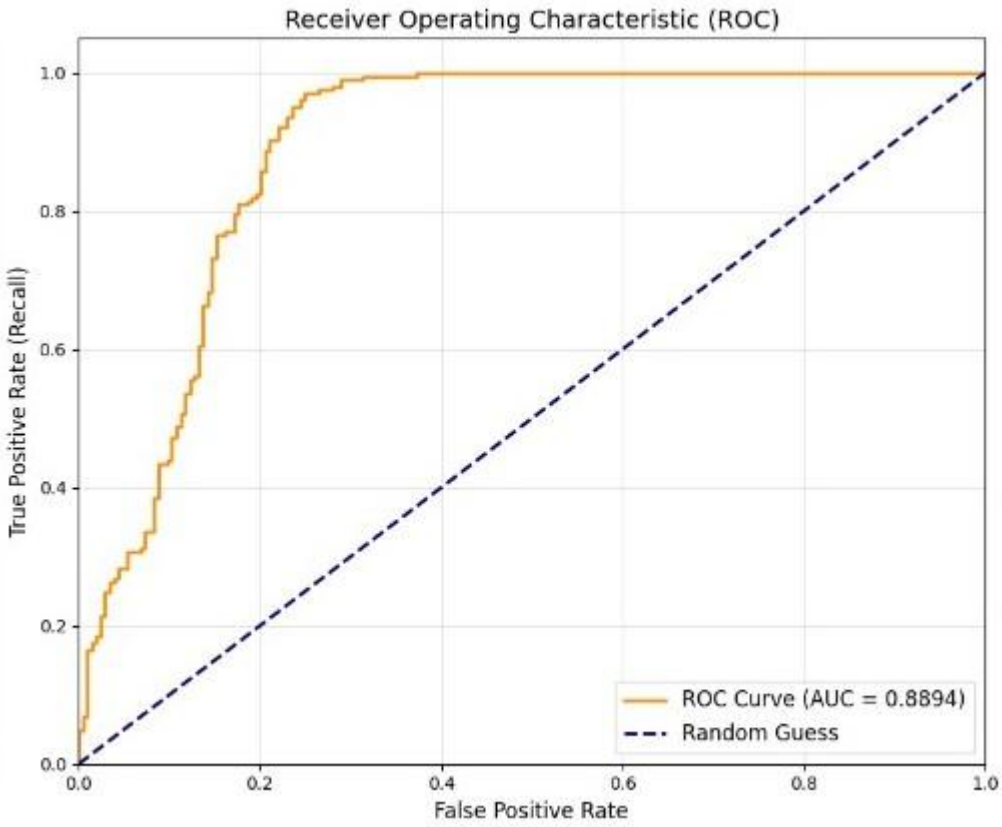

Figure 5.2: ROC Curve for the proposed system

5.6.7 Reliability of Quantitative Metrics

All reported quantitative metrics are computed using standardized evaluation procedures applied consistently across the test dataset. The metrics are derived directly from the classification outputs of the trained model without any post-processing or threshold adjustment.

This ensures that the reported results are reliable, reproducible, and directly representative of the system's true performance under the defined experimental conditions.

5.6.8 Summary of Quantitative Results

In summary, the quantitative evaluation demonstrates that the proposed deepfake detection system achieves strong performance across multiple metrics. The overall accuracy, class-wise precision and recall, and macro-averaged F1-score collectively indicate that the system is capable of effectively distinguishing between real and manipulated videos.

These results validate the effectiveness of the proposed framework and provide a solid numerical foundation for further qualitative analysis discussed in the following section.

5.7 Qualitative Analysis

This section provides a qualitative analysis of the experimental results obtained from the proposed deepfake detection system. Unlike quantitative evaluation, which focuses on numerical metrics, qualitative analysis aims to interpret the system's behavior and explain how different components of the framework contribute to the observed performance.

The qualitative evaluation is based on examining the classification outcomes and understanding the patterns reflected by the reported metrics. This analysis focuses on system behavior rather than visual inspection of individual frames, ensuring consistency with the software-based nature of the proposed approach.

5.7.1 Behavior of the System on Real Videos

The evaluation results indicate that the proposed system demonstrates strong reliability when processing real video samples. This behavior is supported by the high precision achieved for the real class, which suggests that the system is highly confident when labeling a video as authentic.

From a qualitative perspective, this indicates that the framework effectively learns stable spatial and temporal patterns associated with natural facial motion. The use of motion-based frame selection helps the system focus on frames that contain meaningful transitions rather than redundant visual content. As a result, the extracted features represent consistent and coherent facial dynamics commonly found in real videos.

The integration of temporal modeling through the Vision Temporal Transformer further strengthens this behavior by allowing the system to capture long-range dependencies across frames. This contributes to reducing false positive classifications where fake videos might otherwise be incorrectly labeled as real.

5.7.2 Behavior of the System on Fake Videos

The system demonstrates particularly strong performance when detecting fake videos, as reflected by the high recall achieved for the fake class. Qualitatively,

this indicates that the proposed framework is highly sensitive to manipulation-related inconsistencies.

The residue-based analysis plays a significant role in this behavior. By computing differences between consecutive frames, the system captures fine-grained temporal inconsistencies that are difficult to perceive visually but are often present in manipulated content. These inconsistencies are then processed alongside frame-based features, allowing the system to identify abnormal temporal patterns.

The qualitative interpretation of this behavior suggests that the system prioritizes the detection of manipulation cues, ensuring that fake videos are rarely overlooked during classification.

5.7.3 Impact of Motion-Based Frame Selection

Motion-based frame selection contributes significantly to the qualitative performance of the system. Instead of processing all frames equally, the framework selects frames that contain meaningful motion information. This reduces redundancy and allows the model to focus on transitions where manipulation artifacts are more likely to appear.

Qualitatively, this approach improves both efficiency and effectiveness. By limiting the analysis to informative frames, the system avoids learning from repetitive or static content, which may dilute important temporal signals. This design choice enhances the interpretability and robustness of the extracted features.

5.7.4 Role of Residual Frame Analysis

Residual frame generation provides an additional qualitative advantage by highlighting subtle changes between frames. These changes often reflect inconsistencies introduced during deepfake generation processes.

The qualitative results indicate that combining residual features with standard frame features enables the system to detect patterns that may not be apparent when analyzing frames independently. This complementary relationship between spatial appearance and temporal variation strengthens the system's ability to differentiate real and fake videos.

5.7.5 Effect of Attention-Based Feature Enhancement

The use of Multi-Head Attention enhances the system's qualitative performance by allowing it to focus on the most relevant temporal and spatial features. Attention mechanisms help the model weigh different feature representations based on their importance to the classification decision.

Qualitatively, this leads to more stable and discriminative feature representations. The attention mechanism enables the system to emphasize critical inconsistencies while suppressing less informative signals, contributing to more confident and consistent classification outcomes.

5.7.6 Generalization Behavior of the System

The qualitative analysis of test results suggests that the system generalizes well to unseen data. The balanced performance across both real and fake classes indicates that the framework does not rely on dataset-specific patterns but instead learns meaningful temporal and spatial characteristics.

This behavior reflects the effectiveness of the modular pipeline and the integration of multiple complementary feature sources. The system's ability to maintain stable performance on unseen videos highlights its robustness under varying conditions.

5.7.7 Error Patterns and Misclassifications

Although the system demonstrates strong performance, some misclassifications still occur. Qualitatively, these errors may be attributed to videos where manipulation artifacts are extremely subtle or where natural facial motion closely resembles manipulated patterns.

Such cases highlight the inherent difficulty of deepfake detection and emphasize the importance of temporal modeling and residue analysis. The observed error patterns provide valuable insight into areas where further refinement may enhance system performance.

5.7.8 Summary of Qualitative Findings

In summary, the qualitative analysis confirms that the proposed deepfake detection system effectively captures meaningful spatial, temporal, and motion-based patterns. The integration of motion-based frame selection, residual analysis, temporal modeling, and attention mechanisms results in a robust and interpretable classification process.

The qualitative observations align closely with the quantitative results, reinforcing the overall reliability and effectiveness of the proposed framework.

5.8 Summary of Experimental Findings

This section summarizes the experimental findings obtained from the evaluation of the proposed deepfake detection system. The summary consolidates the observations from dataset preparation, training configuration, quantitative results, and qualitative analysis to provide a comprehensive understanding of the system's overall performance.

The experimental evaluation confirms that the proposed framework is capable of effectively analyzing video content and distinguishing between real and fake samples by relying on structured spatial, temporal, and motion-based processing. The results demonstrate that each stage of the framework contributes meaningfully to the final classification outcome.

5.8.1 Effectiveness of the Dataset Preparation Process

The dataset preparation process played a critical role in the success of the experimental evaluation. The use of two publicly available datasets containing both real and manipulated videos ensured that the system was exposed to diverse facial content during training and testing.

The application of class balancing reduced bias toward any single class, allowing the model to learn representative patterns for both real and fake videos. Additionally, the one-time preprocessing strategy ensured consistency across all data samples. By storing preprocessed frames and residuals, the experimental setup maintained stable input conditions throughout training, validation, and testing phases.

5.8.2 Contribution of Motion-Based Frame Selection

One of the key experimental findings is the effectiveness of motion-based frame selection. Instead of processing all frames uniformly, the framework focuses on frames that contain meaningful transitions. This approach reduces redundancy and ensures that the model learns from frames that carry significant temporal information.

The experimental results indicate that this strategy contributes to improved efficiency and stronger learning of temporal patterns. Motion-based selection supports the system's ability to detect

manipulation artifacts that may only appear during specific transitions within the video.

5.8.3 Impact of Dual-Stream Feature Extraction

The experimental findings highlight the importance of using two parallel feature streams for frame-based and residue-based analysis. Frame features capture spatial appearance and facial structure, while residual features emphasize temporal inconsistencies between consecutive frames.

The fusion of these two feature types allows the system to analyze complementary information. This design choice enhances the robustness of the detection process and improves the system's sensitivity to subtle manipulation artifacts that may not be visible in individual frames.

5.8.4 Performance of the Training Strategy

The training configuration adopted in this project contributes significantly to the stability and reliability of the learned model. The use of appropriate loss functions for both classification and feature consistency encourages the model to learn discriminative yet stable representations.

Monitoring validation performance during training and restoring the best-performing weights ensures that the final model maintains strong generalization capability. The application of training control mechanisms such as early stopping and learning rate adjustment further improves convergence and reduces the risk of overfitting.

5.8.5 Interpretation of Quantitative Results

The quantitative evaluation demonstrates that the proposed system achieves balanced performance across both real and fake classes. The reported overall accuracy reflects the system's ability to correctly classify the majority of test samples.

The per-class metrics provide deeper insight into system behavior. High precision for real videos indicates that the system is reliable when identifying authentic content, while high recall for fake videos shows strong sensitivity to manipulated samples. The macro-averaged F1-score confirms that the system maintains consistent performance across classes without favoring one over the other.

5.8.6 Insights from Qualitative Analysis

The qualitative analysis supports the quantitative findings by explaining how different components of the framework influence classification behavior. Motion-based frame selection, residual analysis, temporal modeling, and attention mechanisms collectively contribute to stable and interpretable decision-making.

The system demonstrates strong generalization behavior, maintaining performance on unseen data while minimizing false classifications. Observed misclassifications highlight the inherent difficulty of deepfake detection rather than structural weaknesses in the proposed framework.

5.8.7 Overall System Reliability

The experimental findings confirm that the proposed system operates reliably within the defined scope of the project. The modular design and structured processing pipeline ensure consistent data flow and clear separation of responsibilities across system components.

The integration of multiple complementary analysis techniques strengthens the system's ability to detect deepfake videos in a controlled evaluation environment. The results validate the effectiveness of the proposed framework without introducing additional processing stages beyond those defined earlier.

5.8.8 Chapter Conclusion

In conclusion, Chapter 5 presents a comprehensive experimental evaluation of the proposed deepfake detection system. The findings demonstrate that the system successfully combines motion-based frame selection, facial region analysis, temporal–spatial feature extraction, and attention-based enhancement to achieve reliable classification performance.

This chapter confirms that the experimental design, training strategy, and evaluation methodology align well with the proposed framework. The insights gained from this evaluation provide a strong foundation for the final conclusions and future directions discussed in the next chapter.

Chapter 6
Conclusion and Future Work

6.1 Conclusion

This project presented a comprehensive deepfake detection framework that integrates motion analysis, spatial feature extraction, and temporal modeling within a unified software system. The primary goal of this work was not only to achieve accurate classification results but also to design an end-to-end solution that reflects real-world deployment requirements.

The proposed system was developed by carefully structuring each stage of the detection pipeline. Starting from raw video input, the framework processes data through a sequence of well-defined steps that progressively refine the information used for classification. This structured design ensures that each component contributes meaningfully to the final decision-making process.

A key aspect of the system lies in its use of motion-based frame selection. Instead of relying on uniformly sampled frames, the system focuses on extracting frames that exhibit significant motion transitions. This design choice reduces computational redundancy while preserving critical temporal information that is often associated with manipulation artifacts in deepfake videos. By concentrating on informative frames, the system enhances both efficiency and detection reliability.

Face-focused preprocessing plays an essential role in the pipeline. By isolating facial regions and enforcing consistent spatial resolution, the system minimizes background noise and variation across samples. This step ensures that the model learns features that are directly relevant to facial manipulation rather than unrelated visual patterns.

The introduction of residual frame generation further strengthens the system's ability to capture subtle temporal inconsistencies. Residual frames emphasize frame-to-frame differences, which are particularly important in deepfake detection, as manipulated videos often introduce unnatural temporal transitions. By jointly analyzing original frames and residual information, the system gains a richer representation of video dynamics.

The dual-stream Vision Temporal Transformer architecture forms the core of the detection model. This design allows the system to process spatial and temporal cues in parallel, enabling deeper understanding of video behavior across time. The fusion of both streams through attention-based mechanisms ensures that complementary information is effectively combined, leading to more robust feature representations.

The experimental evaluation confirms that the proposed framework achieves stable and balanced performance. The reported metrics demonstrate that the system generalizes well to unseen data and maintains consistent behavior across both real and fake video classes. The balanced precision and recall values indicate that the

system avoids bias toward a specific class, which is a critical requirement for reliable deepfake detection.

Beyond model performance, the project successfully integrates the detection framework into a complete software architecture. The backend implementation using FastAPI provides a reliable and scalable interface for handling video processing requests, while the frontend implementations for web and mobile platforms enable practical user interaction. This separation of concerns enhances system maintainability and extensibility.

Overall, the project demonstrates that combining motion-aware preprocessing, temporal modeling, and attention-based feature fusion within a modular software system is an effective approach to deepfake video detection. The work fulfills its objectives by delivering a technically sound, experimentally validated, and application-ready solution.

6.2 Future Work

While the proposed system achieves its intended objectives, several opportunities exist for extending and enhancing the framework in future work. These extensions can be pursued without altering the core design philosophy of the system.

One potential direction involves improving system efficiency and scalability. Although the current implementation performs well in experimental settings, future efforts could focus on optimizing preprocessing and inference stages to support higher throughput and

reduced latency. Such improvements would facilitate deployment in environments that require real-time or near real-time analysis.

Another area for future enhancement lies in improving robustness under diverse video conditions. Videos captured under varying lighting conditions, resolutions, or compression levels may present additional challenges for detection. Future work could explore strategies to improve stability and consistency across such variations while maintaining the same underlying detection pipeline.

From a system perspective, the user-facing applications could be extended to provide richer interaction and feedback. Additional visualization options or detailed confidence summaries could improve transparency and user trust in the detection process, while still relying on the same backend decision logic.

At the model level, further experimentation with training configurations and feature fusion strategies could lead to incremental performance improvements. Refining attention mechanisms or adjusting training controls may enhance the system's ability to capture complex temporal dependencies without increasing model complexity.

Finally, future work could focus on conducting broader experimental evaluations to further validate system performance. Expanding testing scenarios or increasing the scale of evaluation would provide deeper insight into the system's strengths and limitations, reinforcing confidence in its practical applicability.

References

[1] Y. Mirsky and W. Lee, “The Creation and Detection of Deepfakes: A Survey,”

ACM Computing Surveys, vol. 54, no. 1, 2021.

[2] R. Tolosana et al., “Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection,”

Information Fusion, vol. 64, pp. 131–148, 2020.

[3] Y. Li, M. C. Chang, and S. Lyu, “In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking,”

IEEE International Workshop, 2018.

[4] D. Güera and E. J. Delp ,“Deepfake Video Detection Using Recurrent Neural Networks,”

IEEE AVSS, 2018.

[5] L. Bass, P. Clements, and R. Kazman, “Software Architecture in Practice”,

3rd ed., Addison-Wesley, 2012.

[6] I. Sommerville, “Software Engineering”,

10th ed., Pearson, 2016.

[7] Google and Jigsaw, “Deepfake Detection (DFD) Dataset,” Dataset for Deepfake Detection Research, 2019.

[8] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to Detect Manipulated Facial Images,”

IEEE International Conference on Computer Vision (ICCV), 2019.

[9] T. Fawcett, “An Introduction to ROC Analysis,”

Pattern Recognition Letters, vol. 27, pp. 861–874, 2006.