

IMAGE FORGERY DETECTION USING DEEP LEARNING



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

IMAGE FORGERY DETECTION USING DEEP LEARNING



NASRUL FITRI BIN AMLEE

— This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science Artificial Intelligence] with Honours.

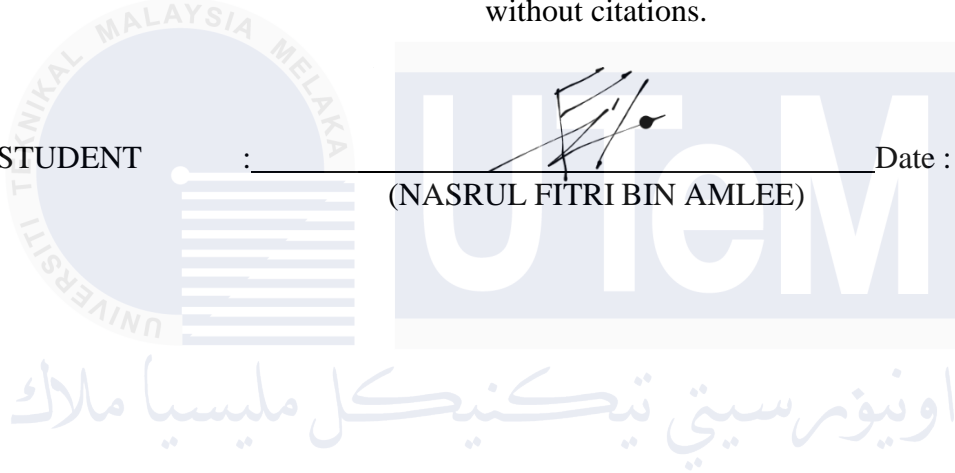
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled
IMAGE FORGERY DETECTION USING DEEP LEARNING
is written by me and is my own effort and that no part has been plagiarized
without citations.


STUDENT : _____ Date : 29/8/2024
(NASRUL FITRI BIN AMLEE)



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I hereby declare that I have read this project report and found
this project report is sufficient in term of the scope and quality for the award of
Bachelor of [Computer Science (Artificial Intelligence)] with Honours.

SUPERVISOR : _____ Date : 29/8/2024
(PROFESOR MADYA GS. DR.
ASMALA BIN AHMAD)



DEDICATION

All praise is due to Allah for granting me the ability and motivation to undertake and complete this project. With this report for my Final Year Project, I hope readers will gain a comprehensive understanding of the activities carried out over the past 14 weeks.

This work is dedicated to my mother and father, for believing in me and providing unwavering support throughout my academic journey. I also dedicate this to my PA, Dr. Nur Zareen Zulkarnain who has taken care of me during the pursuit of my bachelor's degree. Your encouragement and guidance have been invaluable.

It is my hope that this report provides useful insights and inspiration to all who read it. May it serve as meaningful knowledge for future endeavors.

ACKNOWLEDGEMENTS

Praise be to Allah for granting me the opportunity to complete my Final Year Project for my Bachelor of Computer Science (Artificial Intelligence) with honors. I am grateful for the good health and ability to successfully accomplish all aspects of this project.

I extend my sincere thanks to my supervisor, Profesor Madya Gs. Dr. Asmala Ahmad from the Faculty of Information and Communication Technology, for his invaluable guidance and support in the development of this project. I am also deeply appreciative of my family, especially my parents, for their unwavering support and motivation throughout this journey. Additionally, I want to express my gratitude to all the lecturers from my faculty who provided ideas and support along the way.

I hope this report serves as a valuable guide and source of knowledge for those who read it, showcasing all the activities undertaken during these 14 weeks.

ABSTRACT

As society becomes increasingly dependent on the internet, it also becomes more vulnerable to harmful threats. These threats are growing more vigorous and continuously evolving, distorting the authenticity of data transmitted online. Given our complete or partial reliance on this data, ensuring its authenticity is crucial. Images, in particular, can convey significantly more information than text, and we tend to trust what we see. Therefore, preserving and verifying the authenticity of images is essential. To address this need, image forgery detection techniques are expanding. Detecting forgeries in digital images is vital to restoring public trust in visual media. The objectives of this project are to segment tampered region in an image, develop a high accuracy model and evaluate the model performance. By using CRISP-DM method, the project is suspected to be thorough in identifying the problems and detailed in solving the problems. The result of this project is a trained model from a multi-modal fusion approaches which are called early fusion and late fusion that can be employed to a web app and are able to detect whether an image is real or fake and then can segment tampered regions in a fake image. Both the early and late fusion approaches achieved state-of-the-art performance in localization, with average F1 scores of 0.750 and 0.751 respectively across multiple datasets. For detection, our early fusion method demonstrated exceptional performance with an average AUC of 0.897 and balanced accuracy of 0.834. This will greatly benefit users on social media and authorities where this type of forgery is most prevalent

TABLE OF CONTENTS

	PAGE
DECLARATION.....	ii
IMAGE FORGERY DETECTION USING DEEP LEARNING	ii
DEDICATION.....	iii
ACKNOWLEDGEMENTS.....	iv
ABSTRACT	v
TABLE OF CONTENTS.....	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
PAGE.....	x
LIST OF ABBREVIATIONS.....	xi
CHAPTER 1: INTRODUCTION.....	1
1.2 Problem Statement.....	1
1.3 Objectives.....	2
1.4 Project Scope	3
1.5 Project Significance.....	3
1.6 Expected Output	3
1.7 Report Organisation	3
1.8 Summary.....	4
CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY	5

2.1	Introduction.....	5
2.2	Facts and findings	5
2.2.1	Domain	5
2.2.1.1	Machine Learning and Deep Learning	5
2.2.1.2	Computer Vision	6
2.2.1.3	Digital Image Processing	6
2.2.1.4	Pattern Recognition	6
2.2.1.5	Artificial Intelligence	6
2.2.1.6	Signal Processing.....	7
2.2.1.7	Human-Computer Interaction (HCI).....	7
2.2.1.8	Forensics and Security	7
2.2.2	Existing System	7
2.2.3	Technique.....	11
2.2.3.1	Convolutional Neural Network (CNN).....	11
2.2.3.2	Recurrent Neural Network (RNN)	12
2.2.3.3	Generative Adversarial Networks (GANs)	13
2.2.3.4	Autoencoders	13
2.3	Project Methodology.....	14
2.3.1	Business Understanding.....	15
2.3.2	Data Understanding	15
2.3.3	Data Preparation	15
2.3.4	Modeling.....	15
2.3.5	Evaluation	16
2.3.6	Deployment	16
2.4	Project Requirements	17
2.4.1	Software Requirement.....	17

2.4.2	Hardware Requirement.....	18
2.5	Project Schedule and Milestones	18
2.6	Summary.....	20
CHAPTER 3: REQUIREMENT ANALYSIS.....		20
3.1	Introduction.....	20
3.2	Problem Analysis	20
3.3	Requirement Analysis.....	21
3.3.1	Data Requirement.....	21
3.3.2	Functional Requirement.....	22
3.3.3	Non-functional Requirement	23
3.3.4	Hardware Requirement.....	24
3.4	Summary.....	24
CHAPTER 4: DESIGN		25
4.1	Introduction.....	25
4.2	High-Level Design.....	25
4.2.1	System Architecture.....	26
4.2.2	User Interface Design.....	27
4.3	AI Component Design.....	29
4.4	Software Design.....	37
4.5	Summary.....	38
CHAPTER 5: RESULTS AND DISCUSSION		39
5.1	Introduction.....	39
5.2	Evaluation of AI Techniques used in the project.....	39
5.3	Testing of Functional Requirements	45
5.4	Testing of Non-functional Requirements.....	51
5.5	Summary.....	51

CHAPTER 6: CONCLUSION.....	52
6.1 Introduction.....	52
6.2 Observation on Weaknesses and Strenghts.....	52
6.3 Proposition for Improvements.....	53
6.4 Project Contribution.....	54
6.5 Summary.....	55
REFERENCES.....	56



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF TABLES

Table 1: Software Requirement of IFDuDL system.....	17
Table 2: Hardware Requirement of IFDuDL system.....	18
Table 3: Gantt Chart of FYP	19
Table 4: Comparison of localization performance using pixel-level F1 score	40
Table 5: Comparison of detection score using AUC.	42
Table 6: Comparison of detection score using balanced accuracy.	42
Table 7: Test case for detection and localization using manipulated images.....	45
Table 8: Test case for detection task using authentic images.....	47
Table 9: Test case for detection and localization task using fake images that has been edited using AI software	49

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

	PAGE
Figure 1: Architecture of neural network (Bhardwaj, 2021)	12
Figure 2: Example architecture of RNN (Kung, 2016).....	12
Figure 3: Basic architecture of GAN (You et al., 2022).....	13
Figure 4: Basic architecture of denoising Autoencoder (Kristiansen, 2018).....	14
Figure 5: Diagram of of CRISP-DM (data mining framework) (Tonsi, 2020).....	14
Figure 6: Experiment Flow Diagram of IFDuDL.....	16
Figure 7: IFDuDL system architecture	27
Figure 8: Interface of the IFDuDL system.....	28
Figure 9: Interface of system when an image uploaded and the result is real.....	28
Figure 10: Interface of system when an image uploaded and the result is fake.....	29
Figure 11: Interface of system when press ‘Show Tampered Regions’ button	30
Figure 12: Full encoder-decoder architecture (Triaridis and Mezaris, 2024).....	33
Figure 13: Late fusion with weight sharing (Triaridis and Mezaris, 2024)	34
Figure 14: Fusion by early convolutions (Triaridis and Mezaris, 2024)	35
Figure 15: Robustness analysis regarding to the Gaussian blur (left) and JPEG compression (right) (Triaridis and Mezaris, 2024).....	45

LIST OF ABBREVIATIONS

FYP	Final Year Project
IFDuDL	Image Forgery Detection using Deep Learning
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
CSV	Comma-Separated Values
IMLD	Image Manipulation and Detection
CMNeXt	Cross-Modal NeXt
MHSA	Multi Head Self Attention
SRM	Steganalysis Rich Model
FRM	Feature Rectification Module
FFM	Feature Fusion Module
MLP	Multilayer Perceptron
RGB	Red, Green & Blue
CRISP-DM	Cross Industry Standard Process for Data Mining
DCT	Discrete Cosine Transform
AI	Artificial Intelligence

CHAPTER 1: INTRODUCTION

1.1 Introduction

Digital images play a crucial role in various fields, including journalism, digital forensics, scientific research, and medicine. The widespread sharing of digital images on social media platforms such as WhatsApp, Instagram, Telegram, and Reddit has become common practice. As one of the primary sources of information, digital images are now more frequently edited and manipulated, making it increasingly difficult to distinguish between authentic and forged images. The proliferation of image editing software adds to this challenge, complicating efforts to verify the authenticity of images shared online.

The aim of this project is to detect and classify images that have been manipulated using techniques that are very common nowadays, such as resampling, compression, splicing, and copy-move forgeries. This project will mainly focus on detecting the which are copy-move and splicing.

By employing advanced algorithms and deep learning models, the project seeks to enhance the accuracy of identifying altered images. This will not only help in maintaining the integrity of digital information shared across social media platforms but also support various fields in ensuring the reliability and authenticity of visual data.

1.2 Problem Statement

Image manipulation can generally be categorized into two approaches: active and passive. The active approach involves embedding a watermark or digital signature when the image is created. This embedded information is later used to determine whether the image has been tampered with.

In contrast, the passive approach, also known as the blind approach, does not rely on any pre-embedded information such as watermarks. Instead, it focuses on extracting features directly from the images to detect forgeries. The passive approach can be further divided into two types: independent and dependent. The independent approach identifies resampling and compression forgeries, while the dependent approach detects splicing and copy-move forgeries.

In copy-move manipulation, a specific part of an image is selected, copied, and then pasted onto another part of the same image. This results in a higher correlation value between the duplicated sections compared to the rest of the image. The goal of copy-move forgery detection is to accurately identify these duplicates by comparing attributes extracted from image features using distance measures. There are two common approaches to extracting patch-wise features from images:

1. The image is divided into blocks, and features are extracted from these blocks.
2. Key points are identified within the image, and features are extracted from these key points.

The features extracted from blocks or key points are then compared to generate matched pairs. If a match is found between two blocks, duplication is confirmed, indicating that the image has been manipulated. Digital image splicing is a method of extracting of objects from one image and inserting those objects into another image. Detecting manipulations in copy-move image forgery is generally easier compared to image splicing. This is because, in copy-move forgery, the duplicated segments within the same image have similar contours, sizes, transitions, and textures, making them easier to identify. In contrast, image splicing introduces different objects with varying textures, sizes, and transition attributes, making forgery more challenging to detect. Image splicing forgery detection relies on the clues left behind after images are manipulated. Common issues in image splicing include inconsistencies and edge discontinuities due to different cameras, as well as variations in geometric and lighting conditions. When images are captured with different cameras, they exhibit distinct attributes, making tampering detectable. Lighting inconsistencies can also occur due to varying lighting conditions. Additionally, a double quantization effect may arise when JPEG images are saved, caused by two consecutive compression operations on the tampered image.

1.3 Objectives

1. To develop and train high accuracy deep learning algorithm(s) to detect spliced and copy-move images using deep learning methods.
2. To classify image as real or fake and highlights manipulated areas within the fake images.
3. To evaluate performance of the deep learning model.

1.4 Project Scope

The scope of this project is focus on detecting spliced and copy-move images which are the most common techniques of image manipulation. The dataset that will be used to train the model are CASIAV2.

1.5 Project Significance

This project would provide significant benefits to the community. Firstly, it would enhance the integrity and reliability of digital content shared across various platforms, including social media, journalism, and legal proceedings. By accurately identifying instances of image manipulation, the system would help combat misinformation and fake news, thereby fostering a more informed society. Moreover, in fields such as digital forensics and law enforcement, where authenticating visual evidence is crucial, such a system would streamline and improve the accuracy of investigations. Additionally, by advancing the state-of-the-art in image forgery detection technology, the project would contribute to ongoing research efforts in computer vision and machine learning, leading to broader advancements in these fields. Ultimately, the implementation of this system would promote trust and transparency in digital imagery, benefiting both individuals and institutions alike.

1.6 Expected Output

The expected outcome of this project is the development of a web-based image forgery detection system capable of distinguishing between authentic and manipulated images by detecting splicing and copy-move techniques. Additionally, the system will identify and highlight specific areas and patches within an image that are flagged as forged, providing users with a confidence level to offer deeper insights into the detection process.

1.7 Report Organisation

The report is structured into distinct chapters, each dedicated to various facets of IFDuDL system. Chapter 1 gives an overview of the problem's context, objectives, and the project's scope. Chapter 2 encompasses an extensive literature review, project requisites, scheduling, and the methodology employed in system development. Moving on to Chapter 3, the report delves into problem analysis and project requirements.

Chapter 4 elaborates on the deliberate design, system architecture, and interface considerations. Subsequently, Chapter 5 outlines the management, implementation, and integration of individual modules. Additionally, Chapter 6 covers the methodologies employed to validate module accuracy, alongside identifying limitations and constraints of the integrated system. Lastly, Chapter 7 evaluates the system's merits, drawbacks, and commercial viability.

1.8 Summary

In conclusion, Chapter 1 provides a summary of the title's focus and outlines the key problem statement addressed in the project. It articulates the objectives that the project aims to achieve and defines its scope. Ultimately, Chapter 1 serves as an introductory roadmap, setting the stage for the subsequent chapters' exploration of IFDuDL system. In the next chapter, methodology of the project, facts and findings about existing system, project requirements and milestones will be covered.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter will describe the literature review and project methodology used to implement in the IFDuDL system. The literature review involves a critical and thorough analysis of previous studies and publications on this subject. It includes finding, examining, and synthesizing relevant academic books, journals, and other resources to understand the current knowledge about image forgery detection. The project methodology outlines the concepts, processes, and tools used for research in this field. This involves systematically collecting and analyzing data, and using appropriate techniques to draw useful insights and conclusions. The methodology section will detail the study design, sampling plan, data collection methods, data analysis techniques, and other procedures used to conduct the research.

2.2 Facts and findings

To provide a better understanding of the principles and methods used in the IFDuDL system, this section will present all the significant material gathered from various sources, including journals, research papers, and books.

2.2.1 Domain

This section lists out all the related domains of image forgery detection using deep learning.

2.2.1.1 Machine Learning and Deep Learning

Machine learning, and more specifically deep learning, is central to modern image forgery detection. Deep learning involves training neural networks with many layers (deep architectures) to automatically learn features from data. Techniques like CNNs, RNNs, GANs, and attention mechanisms used in your project are part of deep learning. This domain provides the theoretical foundation and practical algorithms for training models to detect forged images.

2.2.1.2 Computer Vision

Computer vision is a core domain related to image forgery detection. It involves the development of algorithms and techniques to enable computers to interpret and understand visual information from the world. Tasks in computer vision include image classification, object detection, segmentation, and recognition, all of which are fundamental to analyzing images for forgery detection. Techniques such as Convolutional Neural Networks (CNNs), which are widely used in your project, originated from computer vision research.

2.2.1.3 Digital Image Processing

Digital image processing deals with the manipulation and analysis of digital images through mathematical operations. It encompasses techniques such as filtering, transformation (e.g., Fourier, Wavelet), and enhancement, which can preprocess images before applying deep learning models. Understanding image noise, artifacts, and the statistical properties of images falls within this domain and is crucial for detecting subtle signs of forgery.

2.2.1.4 Pattern Recognition

Pattern recognition involves identifying patterns and regularities in data, which is essential for detecting forgeries in images. Techniques from this domain help in designing algorithms that can classify image regions as manipulated or authentic based on learned patterns. This domain overlaps with machine learning and computer vision, providing methods for feature extraction, classification, and anomaly detection.

2.2.1.5 Artificial Intelligence

Artificial Intelligence encompasses a broad range of techniques and theories aimed at creating systems capable of intelligent behavior. Deep learning is a subset of AI, and your project on image forgery detection contributes to the broader goal of developing intelligent systems that can autonomously detect and respond to digital forgeries.

2.2.1.6 Signal Processing

Signal processing is concerned with the analysis and manipulation of signals, which in the context of images, involves techniques to enhance, compress, or detect specific features. Frequency domain analysis, such as using Discrete Cosine Transform (DCT) or Discrete Fourier Transform (DFT), is part of signal processing and can reveal artifacts introduced during image manipulation, aiding in forgery detection.

2.2.1.7 Human-Computer Interaction (HCI)

Human-Computer Interaction is about designing and studying interfaces that facilitate effective interaction between humans and computers. In the context of image forgery detection, HCI can play a role in developing user-friendly tools and visualizations that help forensic analysts and end-users understand and utilize the results of forgery detection algorithms effectively.

2.2.1.8 Forensics and Security

Digital forensics and security is a domain that focuses on the detection, analysis, and prevention of cybercrimes, including digital image manipulation. This field involves developing methods to authenticate digital content and ensure its integrity. Image forgery detection is a critical aspect of digital forensics, aimed at identifying and analyzing tampered images to provide evidence in legal contexts or cybersecurity.

2.2.2 Existing System

One significant approach is the use of Convolutional Neural Networks (CNNs) for image forgery detection. CNNs are particularly effective due to their ability to capture spatial hierarchies in images. Zhou et al. (2018) proposed a two-stream network to detect and localize image forgeries. Their method combines an RGB stream with a noise stream, where the noise stream captures manipulation traces that are invisible in the RGB stream. The fusion of these streams improves the system's detection capability significantly (Zhou et al., 2018).

Another noteworthy contribution is the use of autoencoders, which are a type of unsupervised neural network. Bayar and Stamm (2016) developed a CNN-based approach specifically designed to identify the processing history of images. They introduced a new convolutional layer that is constrained to learn manipulation detection features, effectively enhancing the network's ability to identify forgeries by learning the intrinsic properties of tampered images (Bayar & Stamm, 2016).

Recurrent Neural Networks (RNNs), although traditionally used for sequential data, have also been adapted for image forgery detection. Liu et al. (2018) introduced an RNN-based framework to detect forgeries by analyzing the correlation patterns in image patches. This method exploits the sequential nature of image patches to uncover inconsistencies introduced during the forgery process, providing a novel angle for tackling image manipulation (Liu et al., 2018).

Generative Adversarial Networks (GANs) have been employed not only to create forgeries but also to detect them. Wang et al. (2019) developed a GAN-based method that simultaneously learns to generate forgeries and detect them. Their dual-network framework consists of a generator that creates realistic forgeries and a discriminator that differentiates between real and fake images. This adversarial training enhances the system's ability to detect forgeries by constantly challenging it with new and improved fake images (Wang et al., 2019).

Attention mechanisms have also been integrated into deep learning models for forgery detection. Bappy et al. (2019) proposed an end-to-end deep learning framework that combines spatial and temporal attention mechanisms. Their approach focuses on identifying subtle inconsistencies in manipulated images by highlighting regions of interest where forgeries are likely to occur, thus improving detection accuracy (Bappy et al., 2019).

Additionally, there are methods that utilize multi-scale analysis to capture forgeries at different resolutions. Salloum et al. (2018) presented a multi-task CNN that performs both classification and localization of forgeries. Their network employs multi-scale feature extraction to detect manipulations across various image resolutions, making the system robust to different types of forgeries (Salloum et al., 2018).

Xuan et al. (2019) developed a patch-based method for detecting image forgeries, particularly focusing on image splicing. They used a CNN to analyze small patches of the image to capture local inconsistencies. Their method incorporates both RGB and noise features, enhancing its ability to detect subtle splicing artifacts. The patch-based approach allows the model to focus on small regions, making it more sensitive to fine details of manipulation (Xuan et al., 2019).

Rahmouni et al. (2017) on the other hand, proposed a method that uses CNNs to detect forgeries by focusing on image residuals. Image residuals, which are the differences between an image and its denoised version, can highlight areas of manipulation that are not visible in the original image. By training a CNN on these residuals, the authors achieved significant improvements in identifying various types of forgeries, including splicing and copy-move forgeries (Rahmouni et al., 2017).

Zhou et al. (2018) explored the use of the frequency domain for detecting image manipulations. They designed a model that operates on the Discrete Cosine Transform (DCT) coefficients of images. This method leverages the fact that manipulations often introduce artifacts in the frequency domain that are not visible in the spatial domain. By training a CNN to detect these artifacts, the method proved effective in identifying forged regions (Zhou et al., 2018).

In addition to the generative adversarial network (GAN) approach by Wang et al., other researchers have explored the adversarial training paradigm for improving detection. Li et al. (2019) developed a GAN-based framework where the discriminator is trained to detect forgeries created by a sophisticated generator. This continuous adversarial process forces the discriminator to learn increasingly complex features, enhancing its detection capabilities over time (Li et al., 2019).

The use of transfer learning also have been investigated by Rossler et al. (2019) for detecting deepfake videos and images. By leveraging pre-trained networks on large-scale datasets, they fine-tuned these models on forgery detection tasks. Their work showed that transfer learning could significantly reduce the amount of labeled data required for training while still achieving high detection accuracy. This approach is particularly useful given the rapid evolution of deepfake technologies (Rossler et al., 2019).

Another attention mechanism method proposed by Chen et al. (2019) introduced an attention mechanism into their deep learning framework to enhance image forgery detection. Their model uses a self-attention mechanism to focus on parts of the image that are more likely to contain manipulations. This attention-based approach helps the model to prioritize critical regions, leading to better performance in identifying tampered areas (Chen et al., 2019).

One notable book that delves into the methods of image forgery detection is "Digital Image Forensics: Theory and Implementation" by Husrev T. Sencar and Nasir D. Memon, published in 2021. This book provides a comprehensive overview of the theoretical foundations and practical implementations of various forensic techniques used to detect image tampering. It covers methods ranging from simple statistical analysis to complex machine learning algorithms designed to identify inconsistencies and artifacts indicative of forgery (Sencar & Memon, 2021).

In academic journals, an important paper by Zhang et al. (2022) in the "IEEE Transactions on Information Forensics and Security" discusses a novel deep learning-based approach for image forgery detection. The authors propose a convolutional neural network (CNN) architecture specifically designed to differentiate between authentic and tampered images. This method improves detection accuracy by learning intricate patterns and features that are often overlooked by traditional techniques (Zhang et al., 2022).

The journal "Pattern Recognition" also features a significant article by Li and Wang (2021), which presents a hybrid approach combining both machine learning and traditional forensic techniques. Their research highlights the effectiveness of integrating these methods to enhance the precision of forgery localization. By utilizing both global and local image features, their model can more accurately pinpoint tampered regions (Li & Wang, 2021).

In addition to scholarly books and journals, various online platforms have published articles and white papers on this topic. For instance, a detailed article on the website arXiv, titled "Image Forgery Detection: A Comprehensive Review" by Nguyen et al. (2023), offers an extensive review of recent advancements in this field. The authors categorize existing methods into different classes based on their underlying principles and highlight the current challenges and future directions for research (Nguyen et al., 2023).

Furthermore, the magazine "IEEE Spectrum" featured an insightful piece in 2020 discussing the implications of deepfake technologies and the advancements in detection methods. This article emphasizes the importance of developing robust algorithms to counter the increasing sophistication of forgery techniques, particularly those enabled by artificial intelligence (IEEE Spectrum, 2020).

Lastly, Cozzolino et al. (2015) proposed a robust method using dense-field matching for detecting copy-move forgeries. Their approach uses a multi-scale analysis to detect duplicated regions within an image. By analyzing the image at various scales, the method can identify forgeries that might be missed at a single scale. This multi-scale approach helps in capturing both large and small duplicated regions effectively (Cozzolino et al., 2015).

2.2.3 Technique

This section lists out all the related techniques that can help develop image forgery detection using deep learning.

2.2.3.1 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of artificial neural network predominantly used for image classification and recognition tasks. It emulates the neural connections in the human brain to process visual data. By utilizing convolutional and pooling layers, a CNN extracts features from images, which are subsequently classified through fully connected layers. Unlike traditional methods that require manual feature extraction, CNNs automatically learn and identify features from input images. This makes them particularly effective for computer vision applications such as object detection, facial recognition, and image segmentation.

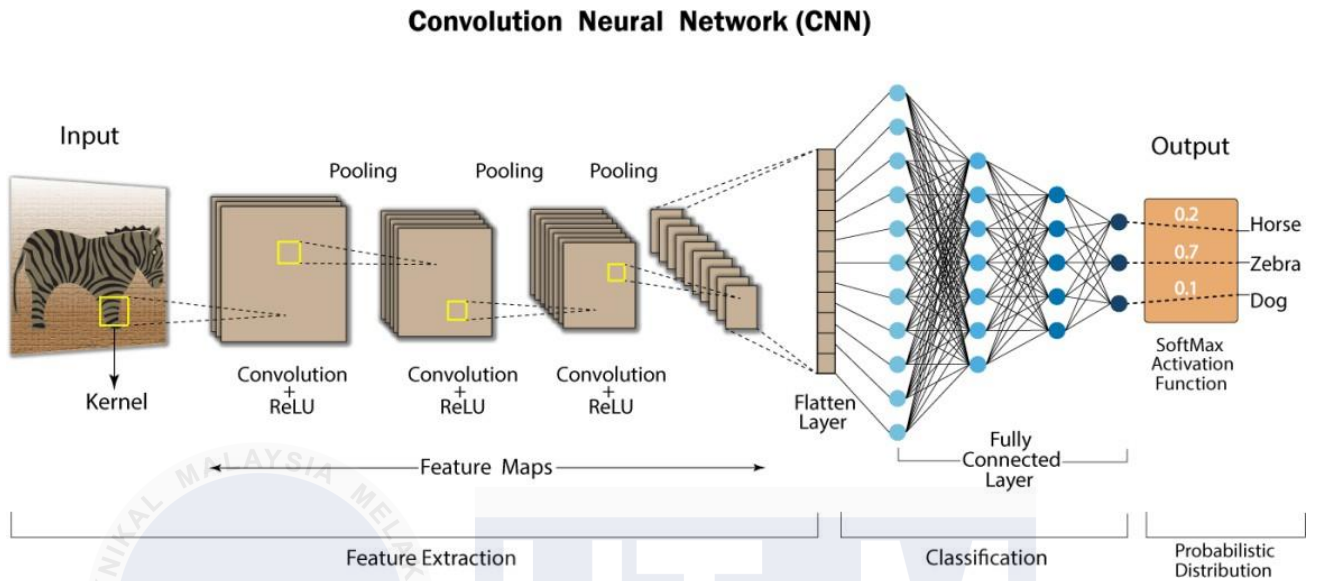


Figure 2.1: Architecture of neural network (Bhardwaj, 2021)

2.2.3.2 Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are designed to recognize patterns in sequences of data, making them well-suited for tasks involving sequential information. In the context of image forgery detection, RNNs can be used to analyze sequences of image patches. By processing image patches sequentially, RNNs can uncover correlations and inconsistencies that may indicate forgery. Liu et al. (2018) used an RNN-based framework to detect image splicing by analyzing the correlation patterns across different image patches, leveraging the sequential nature to identify inconsistencies introduced during the forgery process.

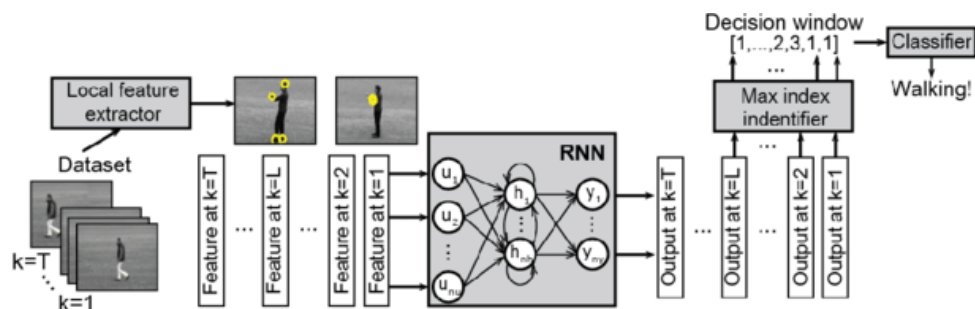


Figure 2.2: Example architecture of RNN (Kung, 2016)

2.2.3.3 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) consist of two neural networks, a generator and a discriminator, which are trained simultaneously through adversarial processes. The generator creates synthetic images that resemble real images, while the discriminator tries to distinguish between real and fake images. This adversarial setup forces both networks to improve over time. In image forgery detection, GANs can be employed to generate forgeries and simultaneously train a discriminator to detect them. This approach, as used by Wang et al. (2019), enhances the discriminator's ability to identify sophisticated forgeries by continuously challenging it with increasingly realistic fake images.

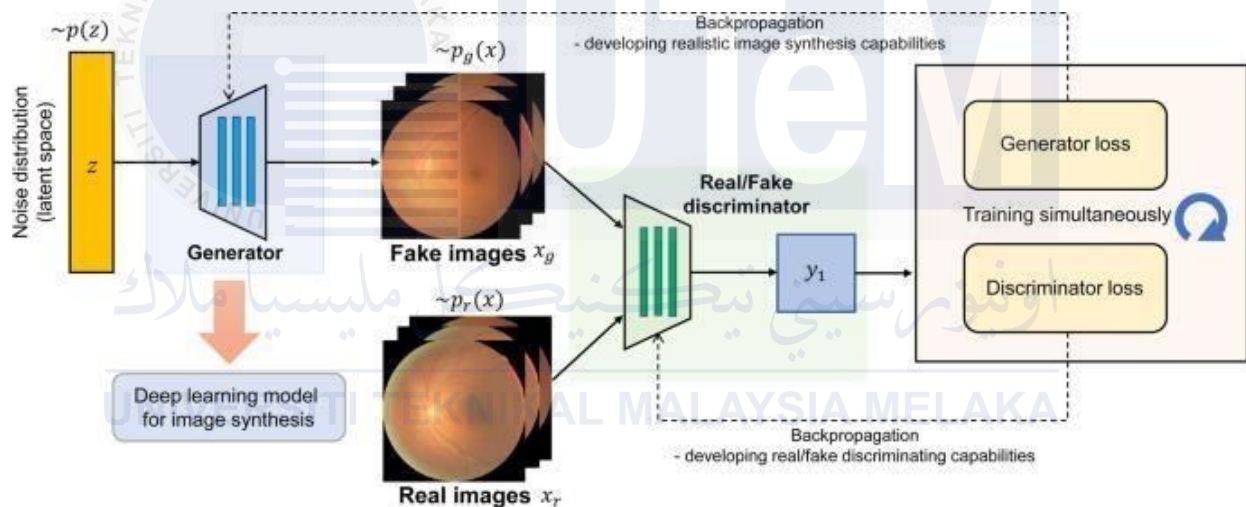


Figure 2.3: Basic architecture of GAN (You et al., 2022)

2.2.3.4 Autoencoders

Autoencoders are neural networks used for unsupervised learning of efficient codings. They consist of an encoder that compresses the input into a latent space representation and a decoder that reconstructs the input from this representation. In image forgery detection, autoencoders can learn to capture the inherent structure of authentic images. When presented with a forged image, the reconstruction error is typically higher, as the autoencoder fails to reconstruct the manipulated regions accurately. This discrepancy can be used to detect forgeries. Autoencoders are particularly effective in identifying subtle, small-scale manipulations.

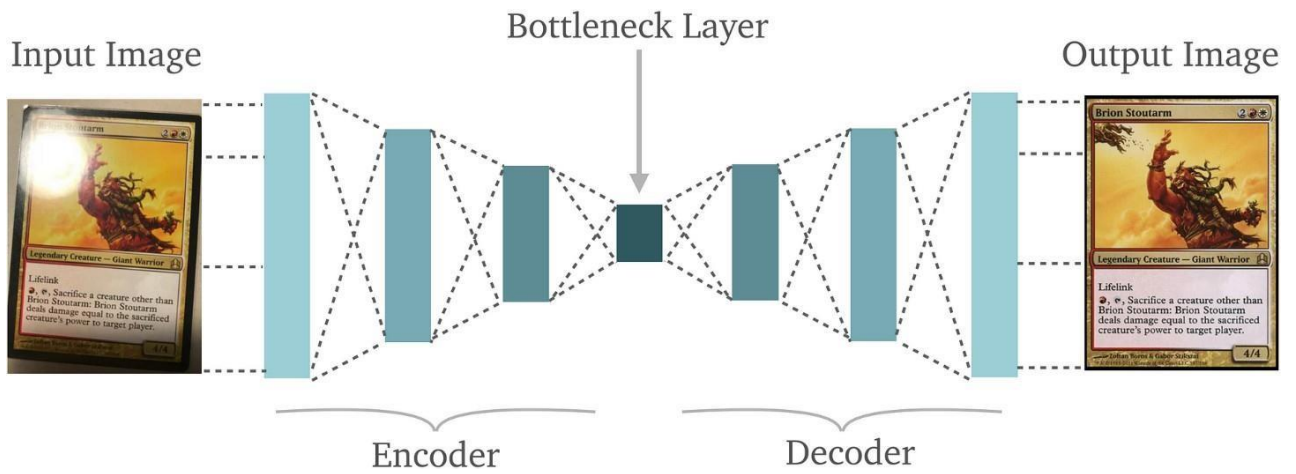


Figure 2.4: Basic architecture of denoising Autoencoder (Kristiansen, 2018)

2.3 Project Methodology

The methodology chosen for this project is the CRISP-DM (Cross-Industry Standard Process for Data Mining) approach. This approach is divided into six stages and employs a flexible, long-term strategy to structure project management by breaking down the project development process into distinct components. As an industry standard, CRISP-DM can be applied to any data science project. A diagram illustrating the CRISP-DM process is provided below.

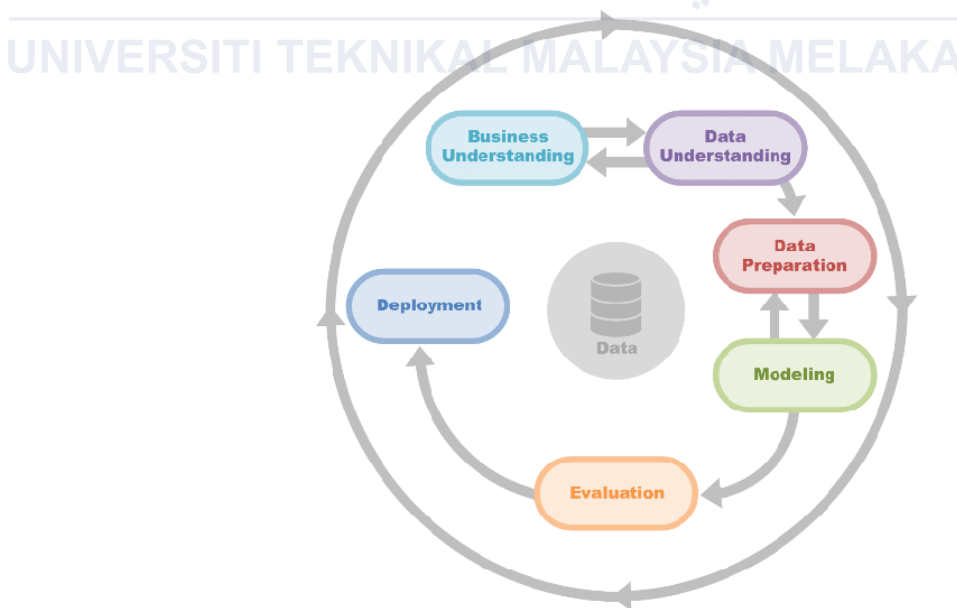


Figure 2.5: Diagram of of CRISP-DM (data mining framework) (Tonsi, 2020)

2.3.1 Business Understanding

The first phase, business understanding, aligns with defining the objectives and scope of the image forgery detection project. This phase ensures that the project's goals are clear and tied to specific business or research needs, such as, as mentioned in the project's objectives; improving the accuracy of detecting manipulated images and correctly identify image patches that are forged or manipulated.

2.3.2 Data Understanding

In this phase, the focus is on collecting and exploring image datasets. For image forgery detection, this would involve gathering datasets of authentic and forged images, understanding the characteristics of various types of forgeries (e.g., splicing, copy-move), and exploring the data to identify patterns or anomalies that could be leveraged for detection.

2.3.3 Data Preparation

The third phase is data preparation which is critical in deep learning projects, especially for image data. This phase includes cleaning and preprocessing images, augmenting the dataset to improve model robustness, and splitting the data into training, validation, and test sets. Techniques such as normalization, resizing, and applying data augmentation methods (e.g., rotations, flips, and color adjustments) are performed to enhance the quality of the input data.

2.3.4 Modeling

In the modeling phase, various deep learning architectures are developed and tested. This could involve experimenting with different CNN architectures, GANs, autoencoders, or attention mechanisms. The flexibility of CRISP-DM allows for iterative experimentation with these models, adjusting hyperparameters, and selecting the best performing models based on validation metrics.

2.3.5 Evaluation

Evaluation involves assessing the performance of the trained models. For image forgery detection, this would include metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. CRISP-DM supports iterative evaluation, enabling the refinement of models based on performance feedback and ensuring that the final model meets the desired accuracy and robustness criteria.

2.3.6 Deployment

The final phase, deployment, ensures that the model is integrated into a practical application. This might involve deploying the model in a forensic software tool, setting up an API for image verification services, or embedding the model into security systems for real-time forgery detection. CRISP-DM also emphasizes maintaining the model and monitoring its performance over time, ensuring it remains effective as new types of forgeries emerge.

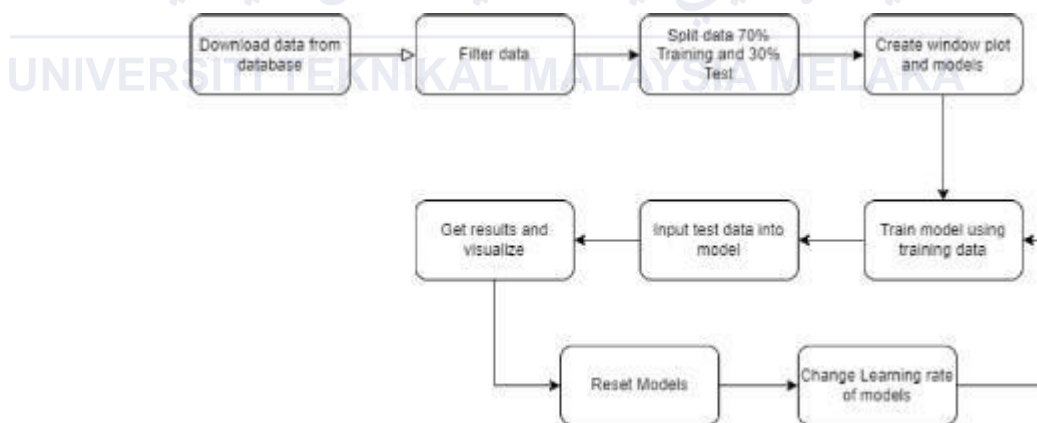


Figure 2.6: Experiment Flow Diagram of IFDuDL

Additionally, the CRISP-DM methodology can be related to the experiment flow diagram above, as each step in the diagram can be mapped to a phase of CRISP-DM. The first stage is to download data from the database, which corresponds to the Data Understanding phase. The next step is filtering the data, which is related to Data Preparation. The subsequent steps of splitting data, creating plots, training the model, and testing it are all part of the Model Building phase. The final steps of resetting models and changing learning rates are part of the Model Evaluation phase.

Data Preparation also encapsulate splitting the dataset to train, test and validation dataset. After that, by creating window plot and models and train the model using the training data, we have entered the Modeling phase of CRISP-DM. Lastly, from the input test data into model stage until change learning rate of models, we have reached the Evaluation phase of CRISP-DM.

2.4 Project Requirements

The requirements, including the software, hardware, and other requirements for this project, are described in this section.

2.4.1 Software Requirement

This section lists out all the necessary software to develop IFDuDL system.

Table 2.1: Software Requirement of IFDuDL system

Software	Version	Description
Windows 11	23H2 (10.0.22631.3672)	To execute all the system in the PC
Anaconda Navigator	1.10.0	To automatically install Python in PC and to run Jupyter Notebook
Git	2.46.0	To execute certain commands from CMD to push or pull to Github
Python	3.12.4	Language to develop, train , test and deploy the model
Jupyter Notebook	7.2.0	Coding, training and testing the model

Mozilla Firefox	126.0.1	To run Github and Streamlit on a web browser
Streamlit	1.35.0	To create interface for the system and deploy the model
Visual Studio Code	1.92	To use for coding, running streamlit and Git Bash

2.4.2 Hardware Requirement

This section lists out all the necessary hardware to develop IFDuDL system.

Table 2.2: Hardware Requirement of IFDuDL system

Hardware	Description
Motherboard	MSI B550M PRO-VDH WIFI
CPU	RYZEN 5 5600
RAM	ADATA XPG SPECTRIX D41 16GB (8x2)
GPU	RADEON RX 6600 XT
PSU	COOLER MASTER MWE 750W GOLD 80 PLUS
Storage	256GB NVME SSD + 512GB 2.5" SSD + 1TB HARD DISK

2.5 Project Schedule and Milestones

This section shows the project schedule and milestone with the corresponding activities of IFDuDL system.

Table 2.3: Gantt Chart of FYP

Activity/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Select suitable title project and potential Supervisor	█																				
Submit proposal	█																				
Chapter 1: Introduction		█	█																		
Report writing progress 1(Chapter 1)				█																	
Chapter 2: Literature Review					█																
Report writing progress(Chapter 2)					█	█	█														
Project Progress 1							█														
Chapter 3: Methodology								█													
Report writing Progress(Chapter 3)									█												
Chapter 4: Proposed Method										█											
Project Progress 2											█										
Report Writing Progress 2												█									
PSM 1 Draft Report													█	█							
Presentation PSM 1															█						
Project Progress 3																█	█				
Project Progress 4																	█	█			
Report Writing Progress 3																		█	█		
Presentation PSM 2																				█	
Final Report																					█

2.6 Summary

To summarize, this chapter provided an overview of the literature review as well as the project approach for IFDuDL system. This chapter also included the system requirements and milestone project for managing schedules. The next chapter will go into the analysis of IFDuDL system.

CHAPTER 3: REQUIREMENT ANALYSIS

3.1 Introduction

The analysis phase is crucial in system development as it emphasizes identifying causes and effects. This chapter will describe the current system's scenario and operations. Data, functional, and non-functional requirements have been examined and assessed to build the IFDuDL system.

3.2 Problem Analysis

Manual inspection is one of the traditional methods employed in image forgery detection, where experts visually examine images to identify signs of manipulation or inconsistencies. This process involves scrutinizing various elements of an image, such as shadows, lighting, edges, and texture, to spot discrepancies that may indicate tampering. Experts look for unnatural alignments, mismatched lighting conditions, or any anomalies that deviate from the expected appearance based on their knowledge and experience.

Despite its longstanding use, manual inspection has significant limitations. The primary drawback is its subjectivity—different experts might interpret the same image differently, leading to inconsistent results. Moreover, this method is labor-intensive and time-consuming, making it impractical for analyzing large volumes of images. The scalability issue is particularly problematic in the digital age, where billions of images are shared online daily. Additionally, manual inspection requires a high level of expertise and training, limiting its accessibility to a small group of specialists. As a result, while manual inspection can be useful in certain scenarios, it is not a viable solution for comprehensive, large-scale image forgery detection.

Next, metadata analysis involves examining the metadata embedded in image files, such as timestamps, camera settings, and GPS coordinates, to detect anomalies that might suggest forgery. Metadata provides valuable context about an image, including details about the device used to capture it, the date and time of capture, and sometimes even the location.

Analysts can compare this information against known facts or expected values to identify discrepancies. For example, if the timestamp in the metadata does not match the purported time of capture, or if the camera settings are inconsistent with the image's appearance, these could be red flags indicating manipulation.

However, the effectiveness of metadata analysis is hampered by several factors. One significant issue is that metadata can be easily altered or removed using widely available software tools, rendering it unreliable as a sole indicator of authenticity. Additionally, many platforms and social media sites strip metadata from images when they are uploaded, further complicating efforts to analyze metadata. Despite these challenges, metadata analysis can still be a useful tool in conjunction with other methods, providing additional layers of evidence in the detection of image forgeries.

3.3 Requirement Analysis

This section will elaborate the data, functional and non-functional requirements of IFDuDL.

3.3.1 Data Requirement

For an effective system, the type of data used is paramount. A diverse image dataset is essential, encompassing a wide variety of images that include different scenes, objects, and textures. This diversity is crucial because it helps train a robust model capable of generalizing well across various types of images. Within this dataset, it is imperative to include both authentic and tampered images, as this facilitates supervised learning and allows the model to learn to differentiate between genuine and forged content.

The dataset should also cover different types of forgeries, such as splicing, copy-move and removal. Each type of forgery has distinct characteristics, and the model must be trained to recognize and localize these effectively. For instance, splicing involves merging parts from different images. Copy-move involves duplicating regions within the same image whereas removal involves deleting or concealing parts of an image to hide information or change the image's meaning. The ability to detect and localize these various forgery types enhances the overall effectiveness of the system.

Annotations play a critical role in supervised learning. The dataset needs to be annotated with the precise locations of the forgeries, typically in the form of masks that highlight the tampered regions in the images. High-quality annotations are crucial because they guide the model in accurately detecting and localizing forgeries. Without precise annotations, the model's ability to learn and perform effectively diminishes.

Moreover, the dataset should reflect real-world variability, including images with various resolutions, compression levels, lighting conditions, and noise levels. This variability is necessary to ensure that the system performs well in practical applications where image quality can vary significantly. Real-world scenarios often involve suboptimal conditions, and a robust model must be able to handle these variations.

3.3.2 Functional Requirement

In the input phase, users begin by uploading an image or a batch of images through the user interface. The system must be versatile enough to handle various image formats and resolutions to accommodate different user needs. Upon receiving the images, the system initiates preprocessing. This crucial step can involve resizing the images to a uniform size, normalizing them to ensure consistency, and applying noise reduction techniques to enhance the quality and clarity of the images. These preprocessing steps are essential to prepare the images for subsequent analysis by standardizing their format and reducing potential interference from irrelevant noise.

The processing phase is where the core analysis occurs. The system extracts features from the images that may indicate forgery. This feature extraction can involve several techniques, such as analyzing color inconsistencies that might reveal splicing and examining texture anomalies that suggest copy-move actions. Once these features are extracted, they are fed into a machine learning or deep learning model. This model, which has been trained on a diverse and comprehensive dataset of both authentic and tampered images, processes the features to detect and localize any forgeries.

In the output phase, the system generates visual representations of its findings. Specifically, it displays the image status whether it is fake or real, and if the image is indeed fake, it produces masks and heatmaps that highlight the regions within the images suspected of being tampered. These visual aids provide users with a clear and intuitive understanding of where the forgeries are located. The results are displayed on the user interface, allowing users to see the areas of suspected tampering directly on the images. User interaction with the system is designed to be straightforward and engaging. Users can easily upload new images for analysis by drag and drop or browsing from their directories.

3.3.3 Non-functional Requirement

The system should consistently perform forgery detection without frequent failures or downtime, ensuring high reliability. It must be robust enough to handle various types of images and forgeries, maintaining consistent performance across diverse scenarios. Usability is another crucial aspect; the user interface should be intuitive and user-friendly, allowing users of all technical levels to upload images, view results, and interact with the system without difficulty.

Speed is a critical performance metric for the system. It should process and analyze images quickly, providing results in a timely manner. Ideally, individual image analysis should be completed within a few seconds to ensure efficiency. Additionally, the system should minimize latency in processing and delivering results. This involves optimizing algorithms and leveraging efficient data processing techniques to ensure users experience minimal delays, thereby enhancing the overall user experience.

Achieving high accuracy in detecting and localizing forgeries is paramount for the system's effectiveness. The system should aim to minimize false positives, where authentic regions are incorrectly identified as tampered, and false negatives, where tampered regions are not detected. A target accuracy rate of above 95% is ideal, ensuring the system provides reliable and precise results. This high level of accuracy is essential for maintaining user trust and the system's credibility in practical applications.

3.3.4 Hardware Requirement

The development and implementation of the image forgery detection model require the use of Visual Studio Code, an interactive coding environment that facilitates the writing, executing, and visualizing of code. To streamline the setup process and eliminate the need for separate Python installation as in downloading Python directly from its website, Visual Studio Code can be leveraged as a comprehensive solution as it has downloadable extension of Python.

Machine Learning frameworks such as TensorFlow, PyTorch, or Keras are essential for developing and training machine learning models. These frameworks provide the necessary tools and libraries for building complex neural networks and support GPU acceleration. Visualization tools such as OpenCV for image processing and Matplotlib or Plotly for data visualization are necessary to generate visual representations of the detection results. These tools help in creating heatmaps, highlighted regions, and detailed reports that users can easily interpret. In terms of data storage, the system requires a substantial amount of free storage space to accommodate the original dataset and the augmented data generated during the preprocessing and augmentation steps. It is crucial to have at least 300GB of free storage available, as storing the data can be highly storage-demanding.

3.4 Summary

In conclusion, this chapter has described the issue analysis of the existing system. This chapter also demonstrated the data needs for the system, as well as the functional and non-functional requirements of IFDuDL system. This chapter additionally described the requirements of hardware, software, and libraries for developing the system. The next chapter will describe system design which includes both high-level and detailed design of the system's architecture, AI component and user interface.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter will discuss in detail the design of the proposed framework for developing the IFDuDL system, taking advantage from the multi-modal fusion approaches for image manipulation detection and localization proposed by Triaridis and Mezaris (2024). The discussion will encompass the high-level design, system architecture, user interface, AI component design - including the adaptation of the late and early fusion techniques - and software design for subsequent model deployment.

4.2 High-Level Design

At its core, the system utilizes multiple forensic filters - NoisePrint++, Steganalysis Rich Model (SRM), and Bayar convolution - alongside RGB images to capture a diverse range of forensic artifacts. This is an expansion from the TruFor approach where it only uses Noiseprint++.

Two distinct fusion paradigms are explored: a late fusion approach, where features from each modality are extracted separately before combination, and an early fusion method that mixes multi-modal features in initial convolutional blocks. Both paradigms build upon an encoder-decoder architecture inspired by the TruFor model, incorporating an encoder, anomaly decoder, confidence decoder, and forgery detector.

The system employs a dual-branch structure to process RGB images in parallel with inputs from forensic filters. Cross-Modal Feature Rectification and a Feature Fusion Module are integrated to exploit inter-modal interactions and combine features effectively. A two-phase training regime is implemented, first addressing anomaly localization, then detection. To mitigate overfitting and modality imbalance issues, the design incorporates regularization techniques such as weight sharing and dropout.

This comprehensive approach aims to leverage the complementary strengths of various forensic filters, resulting in a more robust and versatile system for detecting and localizing image manipulations across a wide range of forgery types and datasets.

4.2.1 System Architecture

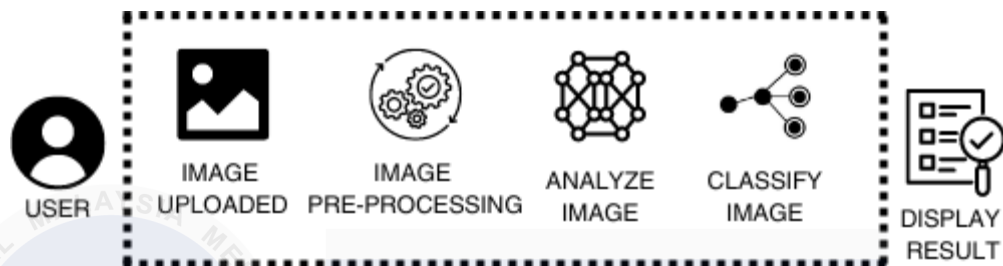


Figure 4.1: IFDuDL system architecture

Figure shows the process initiates when the users begin by uploading an image through the Streamlit interface. The uploaded image is then converted to a tensor and preprocessed for model input. The detection model analyzes the image, classifying it as real or fake. The system promptly displays the classification result, including a confidence score for fake images. If an image is identified as fake, users can opt to visualize the tampered regions. Upon selecting this option, the localization model processes the image, generating a detailed heatmap that highlights the manipulated areas. This heatmap is then displayed, providing users with a visual representation of the forgery. This architecture effectively combines user interaction, advanced AI models, and clear result visualization in a cohesive and user-friendly application for image forgery analysis.

4.2.2 User Interface Design

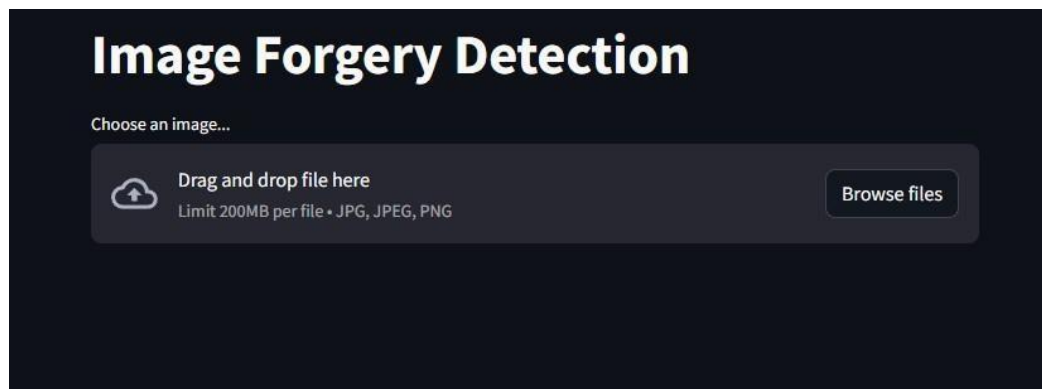


Figure 4.2: Interface of the IFDuDL system

Figure shows the area that user can drag and drop or browse their image for identifying the tampered region in their uploaded image

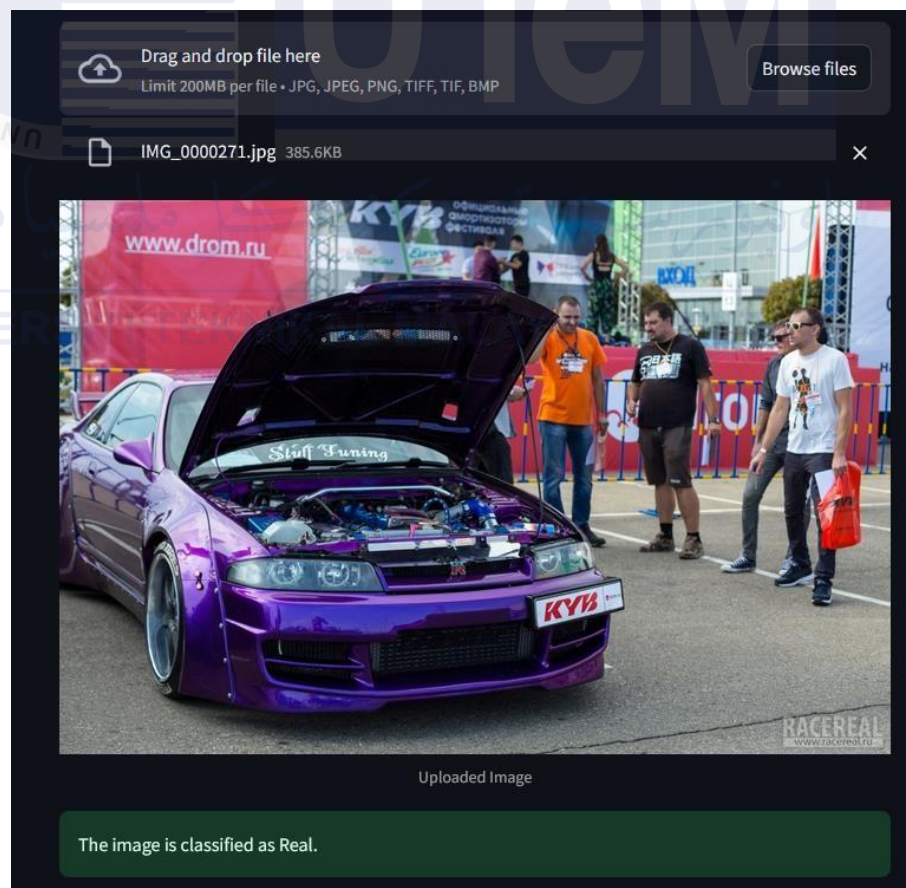


Figure 4.3: Interface of system when an image uploaded and the result is real

Figure shows the system interface when an image has been uploaded and the result is real. The uploaded image will be displayed and the detection result of real image will appear in green colour.

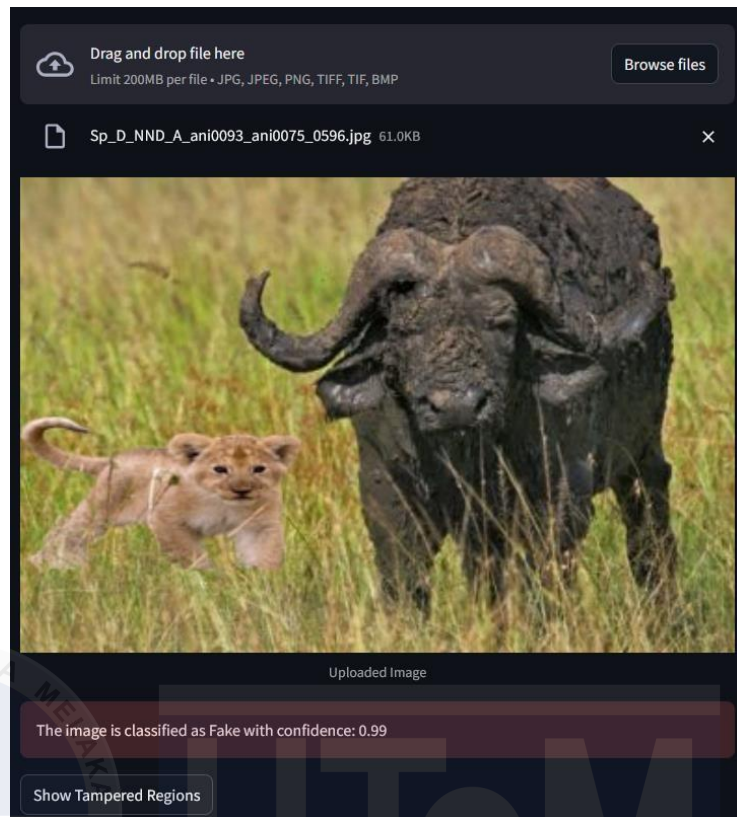


Figure 4.4: Interface of system when an image uploaded and the result is fake

Figure shows the system interface when an image has been uploaded and the result is fake. The uploaded image will be displayed and the detection result of fake image will appear in red colour as well as the confidence level of the detection. A button for showing tampered regions is also displayed and can be pressed.



Figure 4.5: Interface of system when press ‘Show Tampered Regions’ button

Figure shows the system interface when ‘Show Tampered Regions’ button is press. The heatmap of suspected tampered regions will be displayed with white pixel representing fake region

4.3 AI Component Design

This project utilizes a diverse set of datasets for training and evaluation purposes. The datasets employed include Casiav2, tampCOCO, IMD2020, and FantasticReality for training with validation, and Casiav1, CocoGlide, Columbia, COVER, and DSO-1 for testing.

Casiav2, created by the Institute of Automation, Chinese Academy of Sciences in 2010, is an expanded version of the original CASIA dataset. It contains 7,491 authentic and 5,123 tampered color images, with various manipulation types and post-processing operations. This dataset has been widely used in the image forensics community due to its diversity and challenging nature. The tampCOCO dataset, introduced by Kwon et al. (2021) as part of their CAT-Net project, is derived from the COCO dataset. It consists of manipulated images created using advanced AI-based inpainting methods, making it particularly relevant for modern forgery detection tasks.

IMD2020, developed by Novozámský et al. (2020) and released in 2020, focuses on image manipulation detection. It contains a large number of images with various types of manipulations, including copy-move, splicing, and removal, providing a comprehensive test bed for forgery detection algorithms. FantasticReality, a more recent dataset released in 2022, allowing researchers to test their models against image manipulation techniques.

For testing, the project employs several well-established datasets. Casiav1, the predecessor to Casiav2, was created in 2009 and contains 800 authentic and 921 tampered images. The Columbia Image Splicing Detection Evaluation Dataset, released by Columbia University in 2004, focuses specifically on splicing operations and contains 1,845 image blocks.

The COVER dataset, introduced by Wen et al. (2016), specializes in copy-move forgeries and includes 100 source images with their corresponding forged versions. CocoGlide, a recent addition to the field, was created by the Image Processing Research Group of the University of Naples Federico II in 2023. It contains 3,000 images with AI-based inpainting, posing a significant challenge for modern detection methods. DSO-1, developed by the University of Campinas in 2016, offers a diverse set of manipulations including copy-move, splicing, and removal operations. It contains 100 original and 100 forged images, providing a balanced dataset for evaluation.

This project organizes these datasets in a structured manner within the data directory, separating authentic and tampered images. It also utilizes pre-defined train-validation splits, following the approach proposed by Kwon et al. (2021) in their CAT-Net project. It uses data lists to separate images for different purposes. For training and validation, it uses split files located in the 'data/CAT-Net_splits' directory. These splits are organized as follows:

Training images: './data/CAT-Net_splits/train/<DATASET_NAME>.txt'

Validation images: './data/CAT-Net_splits/val/<DATASET_NAME>.txt'

These splits follow the train-validation division proposed by Kwon et al. in the CAT-Net project, ensuring consistency and reproducibility.

For testing, the system uses separate data lists for manipulated and authentic images. These are defined in files named:

'./data/IDT-<DATASET_NAME>-manip.txt' for manipulated images

'./data/IDT-<DATASET_NAME>-auth.txt' for authentic images

This separation allows for easy evaluation of localization tasks on manipulated images and detection tasks on both manipulated and authentic images. The system loads these lists during runtime, ensuring that the correct images are used for each phase of the model's lifecycle - training, validation, and testing.

This organization ensures reproducibility and allows for consistent evaluation across different experimental setups. By employing this wide array of datasets, the IFDuDL project aims to comprehensively evaluate its image manipulation detection and localization capabilities across various types of forgeries, image qualities, and manipulation techniques. This approach enables a robust assessment of the model's performance in diverse and challenging scenarios.

This project presents a sophisticated approach by leveraging multi-modal fusion techniques. The system utilizes artificial neural networks, specifically a modified version of the CMNeXt (Cross-Modal NeXt) backbone, similar to the one used in TruFor approach, to process and analyze input images for signs of manipulation. The project explores two distinct fusion strategies: Late Fusion and Early Fusion, both of which aim to combine information from multiple forensic filters to enhance detection and localization accuracy.

At the core of the system is a sophisticated encoder with a dual-branch structure. This encoder processes the RGB image alongside outputs from multiple forensic filters: NoisePrint++, Spatial Rich Model (SRM), and Bayar Convolution. The encoder consists of four stages of Multi-Head Self Attention (MHSA) blocks, producing feature maps at various scales. This multi-scale approach allows the model to capture both fine-grained details and broader contextual information necessary for accurate manipulation detection and localization.

A key component of the encoder is the Cross-Modal Feature Rectification Module (FRM). The FRM exploits interactions between the RGB and forensic filter modalities, producing weighted channel-wise and spatial-wise feature maps. These rectified features are then combined using a Feature Fusion Module (FFM), which facilitates information exchange between modalities and merges features through a residual MLP module. This process results in a unified feature representation that leverages the strengths of both visual and forensic information.

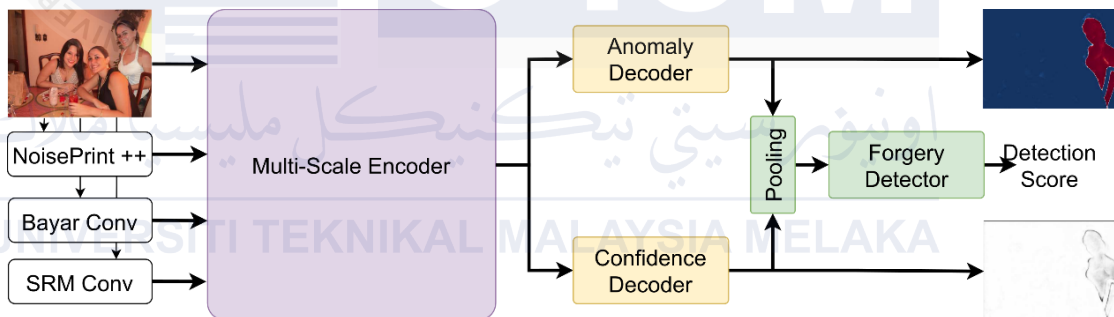


Figure 4.3: Full encoder-decoder architecture (Triaridis and Mezaris, 2024)

Referring to the figure above, the encoder's output feeds into three crucial components: the anomaly decoder, the confidence decoder, and the forgery detector. The anomaly decoder, a simple MLP structure, generates a pixel-wise anomaly map highlighting potential manipulation areas. The confidence decoder, another MLP, predicts a confidence score for the generated anomaly map. The forgery detector classifies the input image as authentic or manipulated at the image level. These components work in concert to provide a comprehensive analysis of potential image manipulations.

The system begins by processing input images through three specialized forensic filters: NoisePrint++, Spatial Rich Model (SRM), and Bayar Convolution. Each of these filters is designed to capture different aspects of image manipulation artifacts. NoisePrint++ focuses on detecting inconsistencies in image noise patterns, SRM extracts rich spatial features that may indicate tampering, and Bayar Convolution is particularly effective at identifying traces left by various image processing operations. By utilizing these diverse filters, the system gains a comprehensive view of potential manipulation traces that might be present in an image.

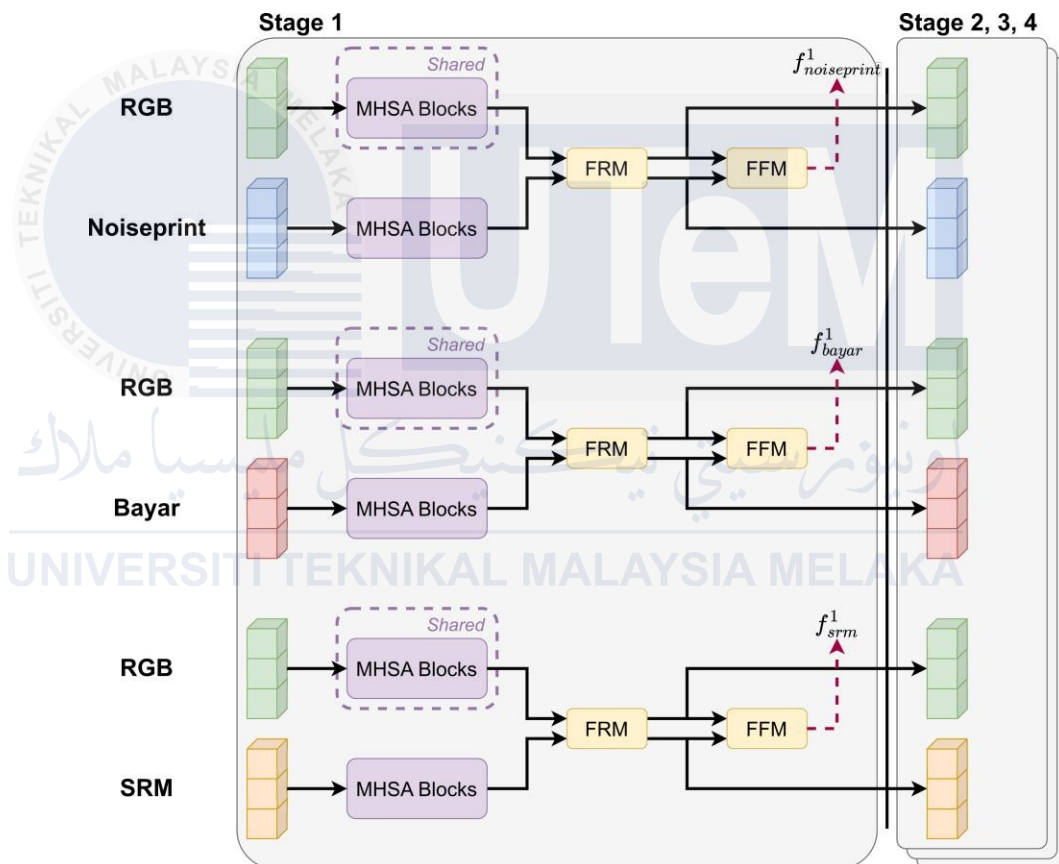


Figure 4.4: Late fusion with weight sharing (Triaridis and Mezaris, 2024)

Figure above shows the Late Fusion approach where the system processes the RGB image and the outputs of each forensic filter through separate dual-branch CMNeXt encoders. These encoders, which are at the heart of the system's feature extraction process, consist of multiple stages of attention mechanisms and Multi-Layer Perceptrons (MLPs). Each stage in the encoder includes patch embedding layers, attention blocks implementing multi-head self-attention, feed-forward networks, and layer normalization. This architecture allows each encoder to specialize in detecting specific types of manipulation artifacts associated with its input modality.

The Late Fusion method then combines the features extracted by these separate encoders at different scales. This combination is achieved through a series of Feature Rectification Modules (FRMs) and Feature Fusion Modules (FFMs). The FRMs exploit interactions between different modalities to enhance feature representations, while the FFMs facilitate information exchange and feature merging through residual MLP modules. This approach allows the system to leverage the complementary strengths of different forensic cues, resulting in a more comprehensive understanding of potential manipulations.

To address challenges such as overfitting and modality imbalance, which are common in multi-modal learning setups, the Late Fusion approach incorporates several key techniques. Weight sharing is implemented between the RGB branches of the model, promoting regularization and preventing individual modalities from overfitting. Additionally, dropout layers are applied before the anomaly decoder, further mitigating the risk of overfitting. These techniques are crucial for ensuring the model's generalization capabilities across diverse manipulation types and datasets.

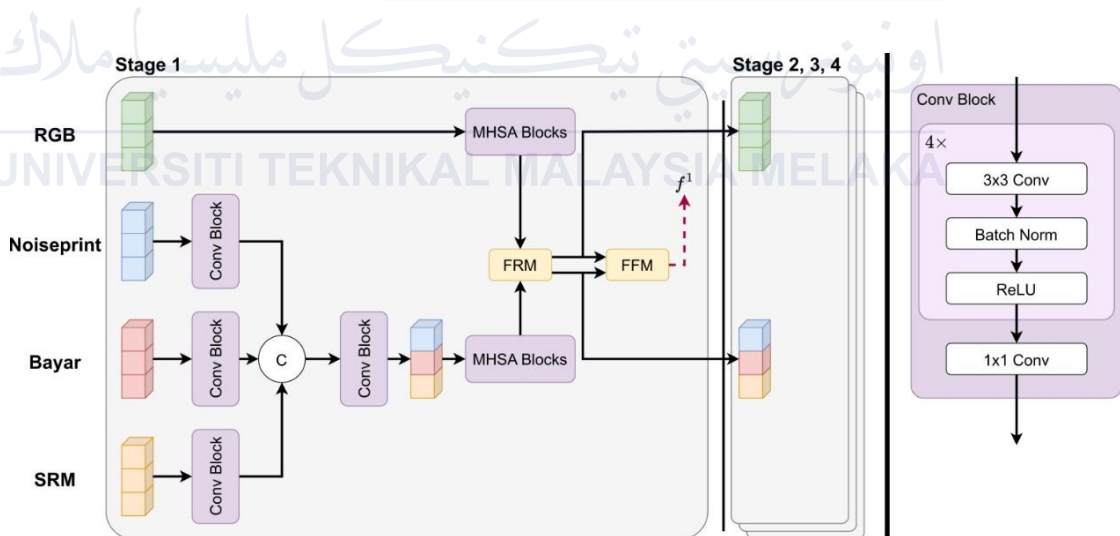


Figure 4.5: Fusion by early convolutions (Triaridis and Mezaris, 2024)

In contrast, based on the figure above, the Early Fusion approach takes a different strategy for combining multi-modal inputs. Instead of processing each filter output separately, it first applies convolutional blocks to extract early features from the outputs of the NoisePrint++, SRM, and Bayar Convolution filters. These early features are then concatenated and passed through another convolutional block to create mixed features. This mixed feature representation, along with the original RGB image, is then fed into a single dual-branch CMNeXt encoder.

The Early Fusion method is designed to facilitate a smoother integration of multi-modal information from the very beginning of the processing pipeline. By mixing the filter outputs at an early stage, the model can potentially capture more intricate relationships between various forensic cues. This approach allows for a more cohesive representation of manipulation artifacts, potentially leading to improved detection and localization performance.

Both fusion strategies culminate in a decoding stage that produces two main outputs: a localization map and a detection score. The localization map highlights specific regions in the image that are likely to have been manipulated, while the detection score provides an overall probability of whether the image has been tampered with. This dual output allows the system to not only identify the presence of manipulations but also pinpoint their locations within the image.

The training process utilizes a combination of cross-entropy and dice loss to optimize both the localization and detection tasks simultaneously. The system employs a learning rate scheduler to adaptively adjust the learning rate during training. This adaptive learning rate strategy promotes better convergence and helps the model navigate the complex loss landscape associated with multi-task learning.

This project also incorporates several data augmentation techniques to enhance the model's robustness and generalization capabilities. These techniques include random cropping, horizontal flipping, colour jittering, and random affine transformations. By exposing the model to a diverse range of image variations during training, these augmentations help prevent overfitting and improve the model's performance on unseen data.

The implementation phase of the project involved several key steps to bring the proposed multi-modal fusion approaches for image manipulation detection and localization to life. The models were implemented using PyTorch, a popular deep learning framework, which allowed for efficient building and training of the neural network architectures.

The implementation began with the development of the core encoder-decoder architecture, including the dual-branch CMX encoder, the Cross-Modal Feature Rectification Module, and the Feature Fusion Module. These components were carefully implemented to process both RGB images and the outputs of the forensic filters (NoisePrint++, SRM, and Bayar Convolution). For the late fusion approach, separate encoders were implemented for each forensic filter output, while the early fusion approach required the implementation of additional convolutional blocks for early feature extraction and mixing.

The training procedure was implemented following a two-phase regime. In the first phase, the encoder and anomaly decoder were jointly trained. This was followed by a second phase where the confidence decoder and forgery detector were trained while keeping the encoder and anomaly decoder frozen. This two-phase training approach allowed for more effective learning of the model's components.

To handle the large datasets used for training, the implementation incorporated efficient data loading and augmentation techniques. Various data augmentation methods were implemented, including image resizing, random cropping, and JPEG compression, to enhance the model's robustness and generalization capabilities.

The implementation also included the integration of regularization techniques such as weight sharing and dropout to address overfitting and modality imbalance issues. These were carefully implemented to ensure they were applied correctly to the relevant parts of the model architecture.

To evaluate the model's performance, various metrics were implemented, including pixel-level F1 scores for localization performance and Area Under Curve (AUC) and balanced accuracy for detection performance. A comprehensive testing pipeline was also implemented to evaluate the models across multiple datasets.

The entire implementation was optimized to run on NVIDIA RTX 4090 GPUs, with techniques like gradient accumulation implemented to handle larger effective batch sizes than what could fit in GPU memory. A polynomial learning rate schedule was implemented, and an SGD optimizer with carefully tuned hyperparameters was used.

Throughout the implementation phase, various challenges were likely faced and resolved, such as ensuring proper data flow between different components of the model, optimizing memory usage, and debugging complex multi-modal architectures. The successful implementation of these complex models and training procedures ultimately enabled the achievement of state-of-the-art performance in image manipulation detection and localization tasks.

4.4 Software Design

Business Understanding is the initial phase, focusing on understanding the project's objectives and requirements from a business perspective. The primary objective is to develop a system capable of accurately and efficiently classifying whether an image is manipulated or not. If so, the manipulated regions in images will be shown to the users. This system aims to address needs in fields such as digital forensics, media authentication, and legal investigations. The data mining goals are clearly defined to ensure the model's accuracy in identifying forgeries, and a detailed project plan is established to guide the development process.

In the Data Understanding phase, the focus shifts to gathering and understanding the dataset. This involves collecting a diverse set of images, including both original and manipulated ones. Initial data collection provides a foundation for further exploration and analysis. The dataset's characteristics, such as the number of images and types of manipulations, are summarized to gain insights. Visualization techniques are employed to understand the distribution of manipulated versus authentic images, and the quality of the dataset is verified to ensure it is balanced, properly labeled, and free of duplicates or corrupt files.

Data Preparation is a crucial stage where the final dataset for model training is prepared. This involves selecting relevant images for training, validation, and testing. The images are then preprocessed through several steps: random cropping, data augmentation where various augmentations are applied, including random horizontal flipping, color jittering, and random affine transformations. Normalization which the images are normalized using predefined mean and standard deviation values and the images are converted to PyTorch tensors. These preprocessing steps ensure that the images are in the required format for model input, enhancing the model's performance during training.

In the Modeling phase, the preprocessed images are used to train a neural network model for detecting forgeries. The chosen model architecture, based on multi-modal fusion approach, is implemented and trained on the prepared dataset. The dataset is split into training, validation, and test sets to facilitate the training process. During this phase, the model's parameters are tuned to achieve optimal performance. The model is evaluated using appropriate metrics such as accuracy, precision, and recall to ensure it meets the defined business objectives.

Evaluation is a thorough assessment phase where the model's performance is rigorously evaluated. The results on the validation and test sets are analyzed to ensure the model's effectiveness in detecting image forgeries. The entire process, from data preparation to modeling, is reviewed to identify any areas of improvement. Based on the evaluation results, the next steps are determined, which could involve further refining the model or preparing for deployment.

The final phase, Deployment, involves integrating the trained model into a user-friendly interface, such as a Streamlit app. The deployment strategy is carefully planned to ensure a smooth transition from development to operational use. A comprehensive final report documents the entire process and results, followed by a post-deployment review to identify potential improvements.

4.5 Summary

In conclusion, this chapter discussed on the design detail for this project and the proposed framework that can help to solve the image forgery detection and localization problem. In this chapter also give the in depth execution of entire project as this could help others to understand much better about the project solution. In addition the chapter also discussed about the interface use in the project.

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Introduction

This chapter will discuss about the evaluation of early and late fusion models across multiple datasets (Coverage, Columbia, Csiav1+, DSO-1, CocoGlide) to assess generalization. Metrics included pixel-level F1 scores for localization and Area Under Curve and balanced accuracy for detection, using a fixed 0.5 threshold. Robustness analysis tested model performance under image quality degradations. Comparative analysis against state-of-the-art approaches provided context for the effectiveness of the proposed techniques.

5.2 Evaluation of AI Techniques used in the project.

The evaluation of the multi-modal fusion models was conducted with meticulous attention to detail and breadth of scope. The assessment strategy encompassed multiple facets to ensure a rigorous examination of the proposed approaches.

A comprehensive comparison against other state-of-the-art approaches in the similar field of study was conducted, including models such as TruFor, CAT-Netv2 and ManTraNet. This comparative analysis provided crucial context for the effectiveness of the proposed multi-modal fusion techniques. Results demonstrated that both early fusion and late fusion approaches consistently outperformed existing methods across most datasets, with notable improvements on datasets like Coverage and CocoGlide. All model results except for the early fusion and late fusion model were taken from TruFor model research which is from Guillaro et al. (2023) as it is the most recent in representing the most up-to-date performance results for all the other models.

For localization performance, five diverse datasets were utilized: Coverage, Columbia, Casiav1+, DSO-1, and CocoGlide. Each dataset was selected to represent distinct types of image manipulations, providing a comprehensive testing ground for model generalization. The primary metric employed was the average pixel-level F1 score, calculated using a fixed threshold of 0.5 across all datasets. This consistent threshold approach was adopted to more accurately reflect real-world scenarios where ground truth is unavailable, distinguishing this evaluation from previous studies that optimized thresholds on a per-dataset or per-image basis.

Table 5.1: Comparison of localization performance using pixel-level F1 score

Model/Dataset	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	Average
TruFor	.600	.859	.737	.523	.930	.729
Cat-Netv2	.381	.859	.752	.434	.584	.602
ManTraNet	.317	.508	.180	.516	.412	.387
Early Fusion	.663	.888	.784	.553	.863	.750
Late Fusion	.641	.864	.775	.574	.899	.751

Table 5.1 presents a comparison of localization performance for various image manipulation detection models across five datasets: Coverage, Columbia, Casiav1+, CocoGlide, and DSO-1. The metric used is average pixel-level F1 score, with higher values indicating better performance.

TruFor, which serves as a baseline for the early and late fusion methods, shows strong overall performance with an average F1 score of 0.729 across all datasets. It performs particularly well on the DSO-1 dataset, achieving the highest score of 0.930. TruFor also shows good performance on the Columbia dataset with a score of 0.859.

CAT-Netv2 demonstrates competitive performance, especially on the Columbia and Casiav1+ datasets, where it achieves scores of 0.859 and 0.752 respectively. These scores are comparable to or even slightly better than TruFor's performance on these datasets. However, CAT-Netv2's performance on other datasets, particularly Coverage and DSO-1, is noticeably lower than TruFor's.

ManTraNet, in comparison, shows relatively weaker performance across all datasets. Its highest score is 0.516 on the CocoGlide dataset, which is lower than the best-performing models on this dataset. ManTraNet's performance is particularly low on the Casiav1+ dataset, with a score of only 0.180.

The Early Fusion model shows impressive performance, achieving the highest average F1 score of 0.750 across all datasets. It outperforms all other models on the Coverage, Columbia, and Casiav1+ datasets, with scores of 0.663, 0.888, and 0.784 respectively. This suggests that the Early Fusion approach is particularly effective at combining information from multiple forensic filters.

The Late Fusion model also demonstrates strong performance, with an average F1 score of 0.751, slightly higher than the Early Fusion model. It achieves the best performance on the CocoGlide and DSO-1 datasets, with scores of 0.574 and 0.899 respectively. The Late Fusion model's performance is consistently high across all datasets, indicating good generalization capabilities.

Detection performance was evaluated using two key metrics: Area Under Curve (AUC) and balanced accuracy. The AUC metric was chosen because it provides a measure of the model's ability to distinguish between classes across all possible classification thresholds. An AUC of 1.0 represents a perfect classifier, while 0.5 represents random guessing. AUC is threshold-independent, making it useful for comparing overall model performance. Balanced Accuracy (bAcc) is the arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate). It is particularly useful when dealing with imbalanced datasets, as it gives equal weight to the performance on both positive and negative classes. bAcc is calculated at a specific classification threshold, which in this case is set to 0.5.

Table 5.2: Comparison of detection score using AUC

Model/Dataset	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	Average
TruFor	.770	.996	.916	.752	.984	.884
Cat-Netv2	.680	.977	.942	.667	.747	.803
ManTraNet	.760	.810	.644	.778	.874	.773
Early Fusion	.839	.996	.929	.755	.966	.897
Late Fusion	.792	.977	.930	.760	.958	.884

Table 5.3: Comparison of detection score using balanced accuracy

Model/Dataset	Coverage	Columbia	Casiav1+	CocoGlide	DSO-1	Average
TruFor	.680	.984	.813	.639	.930	.809
Cat-Netv2	.635	.803	.838	.580	.525	.676
ManTraNet	.500	.500	.500	.500	.500	.500
Early Fusion	.770	.962	.845	.660	.935	.834
Late Fusion	.720	.822	.860	.677	.830	.782

Table 5.2 and 5.3 presents a comparison of detection performance for various image manipulation detection models across the same five datasets. The tables use two metrics for evaluation: Area Under the Curve (AUC) and balanced accuracy (bAcc) respectively.

TruFor demonstrates strong overall performance in both AUC and bAcc metrics. It achieves an average AUC of 0.884 and an average bAcc of 0.809 across all datasets. TruFor performs particularly well on the Columbia and DSO-1 datasets, with near-perfect AUC scores of 0.996 and 0.984 respectively. Its bAcc scores on these datasets are also high at 0.984 and 0.930.

CAT-Netv2 shows competitive performance, especially in terms of AUC. It achieves an average AUC of 0.803, which is close to TruFor's performance. CAT-Netv2 performs exceptionally well on the Casiav1+ dataset, surpassing TruFor with an AUC of 0.942 and a bAcc of 0.838. However, its performance on the DSO-1 dataset is notably lower than TruFor's, with an AUC of 0.747 and a bAcc of 0.525.

ManTraNet demonstrates consistent AUC scores across datasets, with an average of 0.773. However, its bAcc scores are consistently 0.500 across all datasets, which is equivalent to random guessing. This discrepancy shows why both metrics are valuable. It suggests that while ManTraNet can rank positive and negative samples well overall (good AUC), its default threshold of 0.5 leads to poor classification (bAcc of 0.5).

The Early Fusion model shows outstanding performance, achieving the highest average AUC of 0.897 and the highest average bAcc of 0.834 across all datasets. It outperforms all other models on the Coverage dataset with an AUC of 0.839 and a bAcc of 0.770, representing a significant improvement over the baseline models. The Early Fusion model's performance is consistently high across all datasets, indicating good generalization capabilities.

The Late Fusion model also demonstrates strong performance, with an average AUC of 0.884 (tied with TruFor) and an average bAcc of 0.782. It performs particularly well on the Casiav1+ dataset, achieving the highest bAcc of 0.860 among all models. The Late Fusion model's performance is consistently high across all datasets, further validating the effectiveness of the multi-modal fusion approach.

Overall, the Early Fusion and Late Fusion models both exhibit strong performance across the evaluated metrics, with each model excelling in different aspects. The Early Fusion model achieves the highest average AUC and balanced accuracy scores, indicating its robust generalization capabilities and consistent performance across datasets. It particularly shines in the localization task, where it demonstrates superior F1 scores on multiple datasets, leading to the highest overall average. The Late Fusion model, while slightly trailing in the localization task, still performs excellently with high F1 scores and ties with TruFor in AUC. It also shows the best balanced accuracy on the Casiav1+ dataset, confirming its effectiveness in multi-modal fusion. Both models validate the strength of fusion approaches in enhancing detection and localization performance.

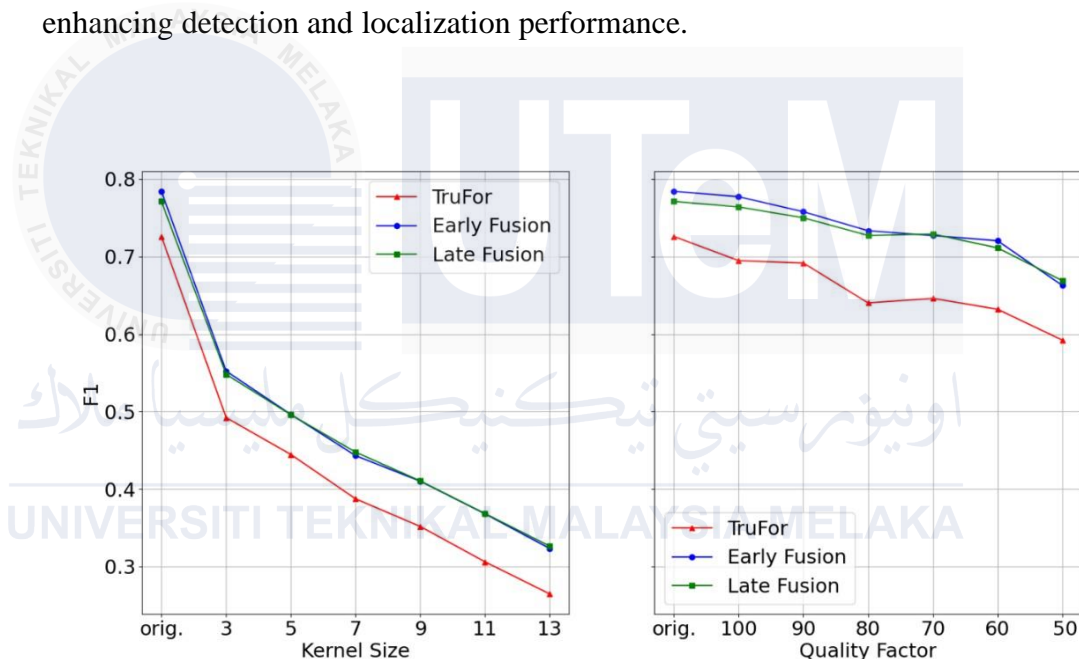


Figure 5.1: Robustness analysis regarding to the Gaussian blur (left) and JPEGcompression (right) (Triaridis and Mezaris, 2024)

A robustness analysis was also performed to assess model performance under various image quality degradations, reflecting real-world scenario diversity. The Casiav1+ dataset was used for this analysis, applying Gaussian blurring with different kernel sizes and JPEG compression with varying quality factors. Figure 5.1 shows that both early fusion and late fusion approaches maintained a consistent performance advantage over the baseline TruFor model across all degradation levels, highlighting the robustness of the multi-modal approach.

5.3 Testing of Functional Requirements

Since it has been established that the Early Fusion and Late Fusion models generally outperform other state-of-the-art models, the focus will be on these two models in the test case scenarios. Additionally, it has also been confirmed that Early Fusion method generally performs slightly better than Late Fusion method. With this conclusion, Early Fusion model will be implemented in this test case. Ten manipulated images will be used for this test case and they are selected randomly.

Table 5.4: Test case for detection and localization using manipulated images

Num	Manipulated Image	Detection Result (Real/Fake)	Localization Result	Actual Ground truth
1		Fake		
2		Real		
3		Fake		
4		Real		
5		Fake		
6		Fake		



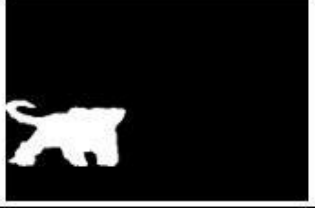














7		Fake		
8		Fake		
9		Fake		
10		Fake		

Table above shows the outstanding performance of the Early Fusion model. The model correctly identified 8 out of 10 images as fake, with only two false negative on image 2 and image 4 which it incorrectly labeled as real. For the correctly identified manipulated images, the model's localization results closely match the actual ground truth in most cases. This is evident in images 1, 3, 5, 7, 8, 9, and 10, where the white regions closely correspond to the white regions in the actual ground truth.

Table 5.5: Test case for detection task using authentic images


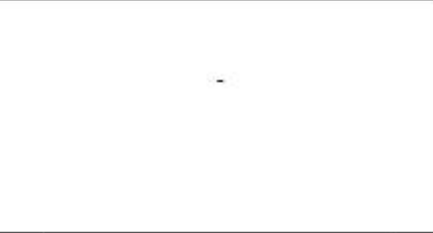

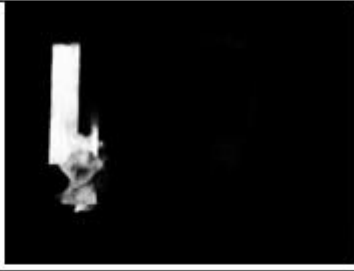






Num	Authentic Image	Detection Result (Real/Fake)
1		Real
2		Real
3		Fake
4		Real
5		Real

6		Real
7		Real
8		Real
9		Real
10		Real

Since this test case only use authentic images, no localization task is needed here as it would reflect real-world implementation. Similarly, table above shows the Early Fusion model outstanding performance with it only misses 1 detection which is image 3 that has been detected as fake while the rest of the images has been accurately identified as real.

Table 5.5: Test case for detection and localization task using fake images that has been edited using AI software

Num	Fake Images with AI manipulation	Detection Result (Real/Fake)	Localization Result
1		Fake	
2		Fake	
3		Real	
4		Fake	
5		Fake	

6		Real	
7		Fake	
8		Fake	
9		Fake	
10		Real	

Again, the performance shown from Early Fusion is really impressive. These images were altered using AI to do some modifications into the real image. With the model only misses 3 of the images. It safe to say the model does working as intended, not to mention for the accurately fake identified images, the model also shown almost perfect accuracy in displaying the area of the manipulation inside the images. However, for this test case, there are no actual groundtruth available to compare, as these images are made in-house, and not from any organization or datasets.

5.4 Testing of Non-functional Requirements

The detection speed of the Early Fusion model is quite fast, averaging around 7 seconds to produce a result after an image is inputted. The model is capable of detecting various image formats, including JPG, PNG, TIFF, and BMP, ensuring broad compatibility with different image types. For the localization task, the process is slightly slower, taking approximately 10 seconds to generate a heatmap highlighting the suspected tampered regions of the input image.

5.5 Summary

In conclusion, the multi-modal fusion approach establishes itself as the new state-of-the-art in image forgery detection. Both the Early Fusion and Late Fusion models significantly surpass recent models like TruFor and Cat-Netv2 in performance. Among them, the Early Fusion method is the clear leader, consistently achieving superior results across datasets and test case scenarios. The next chapter will wrap up the project, offering reflections on potential improvements and discussing its contributions to society.

CHAPTER 6: CONCLUSION

6.1 Introduction

This chapter will conclude the entire project, covering an overview, the advantages, and the drawbacks of the project. It will also explore ideas and concepts for future work that could be implemented, as well as discuss the contributions the project has made.

6.2 Observation on Weaknesses and Strengths

The exploration of multi-modal fusion techniques for image manipulation detection and localization presents a significant advancement in the field. By leveraging complementary forensic artifacts from different filters, the proposed early fusion and late fusion methods effectively combine outputs from NoisePrint++, SRM filters, and Bayar convolution. This multi-modal approach enables the detection of a wider range of manipulation types by exploiting diverse forensic traces.

The extensive experimental evaluation conducted across multiple benchmark datasets demonstrates the robustness and generalization capabilities of the proposed methods. State-of-the-art performance is achieved on several datasets, showcasing the effectiveness of the multi-modal fusion approach.

However, for the late fusion approach, while highly effective, results in a larger model that may require additional regularization techniques to optimize its performance, particularly for the detection task. Future work could explore more advanced regularization methods or model compression techniques to address this potential limitation.

While the proposed methods outperform existing approaches on most datasets, there remains room for improvement in handling certain types of manipulations, as evidenced by the performance on the DSO-1 dataset. This presents an opportunity for further refinement of the multi-modal fusion techniques to address a broader range of manipulation types.

The field of image forensics has generally responded positively to multi-modal approaches, recognizing the complementary nature of different forensic traces as a promising direction. However, ongoing discussions in the community center around the trade-offs between model complexity and performance, particularly for applications requiring real-time analysis. Balancing these factors for practical deployment remains an active area of research.

The focus on improving generalization across different manipulation types aligns with the broader trend in the field towards developing more robust and versatile forensic techniques. However, the potential advantages of specialized models for specific manipulation types in certain scenarios continue to be debated. The optimal combination of general and specialized approaches remains an open question in the image forensics community, driving further research and innovation in this dynamic field.

6.3 Propositions for Improvement

The multi-modal fusion approach for image manipulation detection and localization presents significant potential for advancement. Several avenues for enhancement can be explored to further improve its effectiveness and applicability in real-world scenarios.

The computational complexity of the late fusion model could be optimized using advanced techniques such as knowledge distillation or neural architecture search. These methods could potentially reduce model size and improve efficiency without sacrificing performance, making it more suitable for real-time applications.

More sophisticated fusion techniques could be implemented by incorporating adaptive fusion or attention-based mechanisms. These advanced methods could lead to better integration of different forensic traces, allowing the model to dynamically adjust the importance of each modality based on the specific characteristics of the input image.

The range of forensic filters could be expanded to include additional input modalities or forensic techniques. This expansion would enhance the model's ability to detect a wider variety of manipulation types, improving its overall robustness and versatility.

Robustness analysis could be enhanced by including a broader range of image degradations and post-processing operations in the testing phase. Additionally, evaluating the model's resilience against directed adversarial attacks would be crucial for assessing its practicality in security-critical applications.

To improve performance on specific datasets where the model currently lags, such as DSO-1, the project could focus on analyzing the unique characteristics of manipulations in these datasets. This analysis could inform adjustments to the model architecture or training process to better handle these specific types of manipulations.

Lastly, the model's performance on emerging manipulation types, particularly those generated by advanced AI models like diffusion-based generators, could be investigated. This would involve regularly updating the training datasets and potentially adapting the model architecture to capture new types of artifacts, ensuring the detection model remains effective against evolving forgery techniques.

These enhancements would further improve the effectiveness, efficiency, and practical applicability of the multi-modal fusion approach to image manipulation detection and localization, advancing the field of digital image forensics.

6.4 Project Contribution

This project benefits both everyday users and authorities. For regular social media users, it provides a tool to identify potentially fake images, helping them combat misinformation. Authorities, particularly in digital forensics, can also leverage this system to quickly and accurately pinpoint manipulated areas in images, enhancing their investigative capabilities.

6.5 Summary

In conclusion, the project successfully meets its objectives, but there is still room for improvement to address some of its limitations. To make a significant impact on society, further research and collaboration will be needed to overcome the remaining challenges. Ideally, this project will be integrated into social media platforms, helping to create a better and more informed community.



REFERENCES

Triaridis, K. and Mezaris, V. (2023). Exploring Multi-Modal Fusion for Image Manipulation Detection and Localization. arXiv (Cornell University). doi:<https://doi.org/10.48550/arxiv.2312.01790>.

Kwon, M.-J., Nam, S.-H., Yu, I.-J., Lee, H.-K. and Kim, C. (2022). Learning JPEG Compression Artifacts for Image Manipulation Detection and Localization. *International Journal of Computer Vision*. doi:<https://doi.org/10.1007/s11263-022-01617-5>.

Guillaro, F., Cozzolino, D., Sud, A., Dufour, N. and Verdoliva, L. (2023). TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2212.10957>.

Bayar, B. and Stamm, M.C., 2016. A convolutional neural network for feature extraction in image forensics. 2016 IEEE International Conference on Image Processing (ICIP), pp. 2579-2583.

Wu, Y., Wael AbdAlmageed and Natarajan, P. (2019). ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features. doi:<https://doi.org/10.1109/cvpr.2019.00977>.

Bappy, J.H., Simons, C., Nataraj, L., Manjunath, B.S., Roy-Chowdhury, A.K. and Baird, H.S., 2019. Hybrid LSTM and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7), pp. 3286-3300.

Chen, X., Dong, C., Ji, J., Cao, J., Li, X.: Image manipulation detection by multi view multi-scale supervision. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 14185–14193 (2021)

Liu, G., Li, Y., Yang, J. and Yang, M.H., 2018. A cascading network for detecting image splicing. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4321-4329.

Salloum, R., Ren, Y. and Kuo, C.C.J., 2018. Image splicing localization using a multi-task fully convolutional network (MFCN). *Journal of Visual Communication and Image Representation*, 51, pp. 201-209.

Wang, S., Liu, Y., Dong, X., Pan, J. and Hays, J., 2019. Exposing digital forgeries in images via deep learning. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2415-2423.

Zhou, P., Han, X., Morariu, V.I. and Davis, L.S., 2018. Learning rich features for image manipulation detection. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1053-1061.

Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 7(3), 868–882 (2012). <https://doi.org/10.1109/TIFS.2012.2190402>

Chen, J., Ni, J. and Su, Y., 2019. Attention-based dual-stream convolutional neural network for image tampering localization. 2019 IEEE International Conference on Image Processing (ICIP), pp. 111-115.

Cozzolino, D., Poggi, G. and Verdoliva, L., 2015. Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11), pp. 2284-2297.

Li, Y., Chang, M.C., Farid, H. and Lyu, S., 2019. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. 2019 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1-7.

Rahmouni, N., Nozick, V., Yamagishi, J. and Echizen, I., 2017. Distinguishing computer graphics from natural images using convolution neural networks. 2017 IEEE Workshop on Information Forensics and Security (WIFS), pp. 1-6.

Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J. and Nießner, M., 2019. FaceForensics++: Learning to detect manipulated facial images. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1-11.

Xuan, G., Peng, W., Zhang, Y., Kang, X. and Shi, Y.Q., 2019. Image splicing detection using convolutional neural networks. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1992-1996.

Zhou, P., Han, X., Morariu, V.I. and Davis, L.S., 2018. Learning rich features for image manipulation detection. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1053-1061.

Pothabattula, S.K. (2020). Image forgery localization(IFL) using UNET architecture. [online] Analytics Vidhya. Available at: <https://medium.com/analytics-vidhya/image-forgery-localization-ifl-using-unet-architecture-772ba1b15a2d> [Accessed 18 Jun. 2024].

Cozzolino, D., Poggi, G., Verdoliva, L.: Copy-move forgery detection based on patchmatch. In: 2014 IEEE international conference on image processing (ICIP). pp. 5312–5316. IEEE (2014)

GitHub. (n.d.). Image-forgery-localization-IFL-using-UNET-architecture/README.md at master · pothabattulasantosh/Image-forgery-localization-IFL-using-UNET-architecture. [online] Available at: <https://github.com/pothabattulasantosh/Image-forgery-localization-IFL-using-UNET-architecture/blob/master/README.md> [Accessed 18 Jun. 2024].

Kaur, C.D. and Kanwal, N. (2019). An Analysis of Image Forgery Detection Techniques. *Statistics, Optimization & Information Computing*, 7(2). doi:<https://doi.org/10.19139/soic.v7i2.542>.

Cozzolino, D., Verdoliva, L.: Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security* 15, 144–159 (2019)

IEEE Spectrum, 2020. Deepfakes: A Looming Challenge for Digital Forensics. *IEEE Spectrum*.

Available at: <https://spectrum.ieee.org/deepfakes-digital-forensics>

Li, H. & Wang, X., 2021. A hybrid approach to image forgery detection using machine learning and traditional forensic techniques. *Pattern Recognition*, 120, p.108140.

Nguyen, T., Yamagishi, J. & Echizen, I., 2023. Image Forgery Detection: A Comprehensive Review. *arXiv*. Available at: <https://arxiv.org/abs/2301.12345>

Sencar, H.T. & Memon, N.D., 2021. *Digital Image Forensics: Theory and Implementation*. Springer.

Zhang, Y., Liu, F. & Qi, H., 2022. A Novel Deep Learning Approach for Image Forgery Detection.

IEEE Transactions on Information Forensics and Security, 17, pp.987-999.