Patch-Based Deepfake Localization: Unveiling Manipulated Regions in Images through Visual Artifact Analysis and Deeplearning

by

Syed Hasnad Jami
20301236
MD.Tanzeem Ul Alam
20301228
Alpona Azrin
21101135
S.M. Atahar Istiaque
20301237

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

Brac University
2025

Declaration

It is hereby declared that

1. The thesis presented is the result of our own independent research conducted during our time at Brac University.
2. This thesis contains no material previously published or authored by others, except where proper citations and references have been provided.
3. The thesis does not include any content that has been submitted for any other degree or diploma at any university or institution.
4. We have duly acknowledged and appropriately cited all major sources of assistance and contributions.

Student's Full Name & Signature:

Syed Hasnad Jami	MD.Tanzeem Ul Alam
20301236	20301228
Alpona Azrin	S.M. Atahar Istiaque
21101135	20301237

Approval

The thesis/project titled "Patch-Based Deepfake Localization: Unveiling Manipulated Regions in Images through Visual Artifact Analysis and Deeplearning" submitted by

1. MD.Tanzeem Ul Alam (20301228)
2. Syed Hasnad Jami (20301236)
3. Alpona Azrin (21101135)
4. S.M Atahar Istiaque (20301237)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 31, 2025.

Examining Committee:
Supervisor:
(Member)

Dr. M Iqbal Hossain

Associate Professor
Department of Computer Science and Engineering
Brac University

Co Supervisor:
(Member)

Labib Hasan Khan

Lecturer(Contractual)
Department of Computer Science and Engineering
Brac University

Program coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Associate Professor
Department of Computer Science and Engineering
Brac University

Chairperson:
(chair)

Dr. Sadia Hamid Kazi

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

This research is conducted in strict adherence to ethical standards that utilize publicly available and ethically sourced datasets to ensure respect for privacy and consent. The primary goal is to advance the detection of deepfake technology through visual artifact analysis and deep learning, aiming to safeguard digital media integrity and combat misinformation while upholding the highest standards of ethical responsibility in data usage and research practice.

Abstract

Artificial Intelligence (AI) generative technologies such as DALL·E 3, DALL·E 4, Stable Diffusion, MidJourney, and Generative Adversarial Networks are developing quickly, making it more difficult to distinguish between real, synthetic, and AI-generated (AIG) images. As these technologies progress, traditional detection methods and datasets have become outdated, leading to poor detection accuracy and inference time, particularly for video content. Our research introduces an efficient pipeline for real-time video analysis (RTVAS) to overcome these issues in an AI-generated content detection (AIG-CD) system. We created our own dataset with the help of newly developed generative models for robust training. Our multi-stage processing pipeline includes preloading the detection model (SAAT-ResNet50-3BD) at startup, real-time frame capture, adaptive preprocessing methods to speed up inference time, and face detection to focus on relevant regions. Our proposed model uses pre trained weight and includes enhanced feature extraction, attention mechanism to find any artifacts and irregularities with proper regularization techniques to prevent overfitting that distinguishes between AI-generated (AIG), synthetic media (SM) photos altered by real people. The classification decision is obtained by combining each frame's frequency and statistical average forecasts. As a scalable and effective detection system, our proposed model outperforms Xception (93.6%), EfficientNetV2 (96.5%), MobileNetV3 (94.3%), and VGG16 (65.9%) models in terms of performance matrices. With a detection accuracy of 98.5%, our proposed model offers a superior balance between detection accuracy and compute efficiency, making it highly suitable for practical applications in media forensics, content regulation, and deepfake detection in video frames (VDF).

Keywords: SAAT-ResNet50-3BD, AIG-CD, synthetic media detection (SMD), deepfake detection, transfer learning, real-time video analysis (RTVAS), inference optimization, ResNet50, media forensics, content moderation,VDF, Synthetic Image(SI).

Dedication

We are deeply grateful to our Associate Professor and Supervisor, Dr. M. Iqbal Hossain, for his invaluable guidance and unwavering support throughout this journey. His expertise and encouragement have played a crucial role in shaping the success of this work.

We would also like to extend our sincere appreciation to our Lecturer Co-supervisor, Labib Hasan Khan, for his insightful feedback and continuous motivation. His dedication and support have been truly inspiring, and we are immensely thankful for his contributions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thoughts behind the Model

The rapid adoption of artificial intelligence (AI) across various industries has undoubtedly enhanced efficiency and convenience in many aspects of real life. However, this widespread implementation also introduces significant risks, particularly with the rise of deepfakes. These AI-generated manipulations—whether in the form of images or videos—can convincingly mimic real content and, in doing so, threaten the integrity and security of global information. Deepfakes pose a serious challenge as they have the potential to damage reputations, erode public trust, and distort reality.

A prominent example of this occurred in early 2024, an employee of the UK engineering firm was deceived into transferring $25 million after participating in a video call with what appeared to be senior management [26]. It was later revealed that the employee had been interacting with AI-generated deepfakes that underscore the sophisticated methods that cyber-criminals are employing to exploit this technology. This highlights just how powerful and dangerous deepfake can be in influencing opinions and undermining trust. To address these concerns, real-time detection systems are becoming essential in safeguarding the authenticity of visual content, preventing the rapid spread of misinformation, and maintaining security on digital platforms.

Our research explores the effectiveness of various Convolutional Neural Network (CNN) models—such as ResNet50, Xception, EfficientNetV2B0, VGG16, and MobileNetV3small—in detecting AI-generated falsifications. We have assessed these models against other widely used CNN architectures and existing studies. Specifically, we found that ResNet50, when paired with 3 layer enhancement with unique preprocessing techniques, outperformed the other models in both performance and inference time. Despite Xception's reputation for efficiency through depth-wise separable convolutions [1], ResNet50 demonstrated superior results in our tests.

As AI generative models continue to improve, they make it easier to produce increasingly sophisticated deepfakes that lead to a rise in abuse across public platforms. To combat this, current generative models are focusing on real-time deepfake detection which aim to prevent the rapid distribution of fake content across the internet. Real-time detection systems are crucial for enabling immediate action to protect or-

ganizational information and shield individuals from the harmful effects of deepfake content.

Considerable progress has been made in the field of AI-generated image detection. For instance, MesoNet [2] represents a compact and effective deepfake detection architecture, while FaceForensics++ [7] provided valuable datasets for evaluating various detection techniques. Further advancements in detection methods have been contributed by studies like [10] and [9], pushing the boundaries of real-time detection accuracy. Despite these developments, achieving both high accuracy and real-time performance remains a challenging task.

Our research aims to address this gap by utilizing ResNet50 as a base model, taking advantage of its balance performance and computational cost. We integrate three block of additional layers to create a solution that not only delivers high accuracy but also meets the real-time performance requirements necessary for practical application.

1.2 AI Generate[AIG], Synthetic or Real Human Face?

Figure 1.1: Comparison Between AIG, Synthetic and Real Frames

Figure 1.1: Can you distinguish between real, synthetic, and AI-generated images?

Synthetic Image 1: This image is an example of synthetic image that is created by merging different facial features from various sources, resulting in subtle but

noticeable inconsistencies. Discrepancies such as mismatched eyebrow sizes SI(1.1), inconsistent skin tones on the forehead SI(1.2), unnatural alignment of facial features SI(1.3), and misplaced nose structure SI(1.4) highlight the imperfections caused by the synthetic generation process. These inconsistencies arise because synthetic images are not captured from real-world scenes but are assembled from multiple human sources to form a seemingly cohesive face.

Real Image 2: Unlike synthetic images, real images capture a single, naturally occurring frames, maintaining coherence in facial structure, lighting, and texture. There are no abnormalities in facial symmetry, skin tone, or placement of features along the face as seen in the natural expression of the subjects in this image.

AI-Generated Image (AIG) 3: AI-generated images are produced by latest generative deep learning models that synthesize realistic human faces without a direct real-world counterpart. These images often exhibit photorealistic quality, with smooth textures, balanced lighting, and natural expressions. However, upon close inspection, AI-generated images might still contain subtle artifacts, such as irregular reflections in glasses, unnatural hair textures, over smoothness, or inconsistencies in background elements which can be found at AI-Generated Image (AIG) 3.

1.3 Research Problem

1.3.1 The Challenge of AI-Generated Images

The increasing sophistication of AI-generated images, known as deepfakes, presents a significant challenge to information integrity and security. Recent reports, such as those by [33], highlight the misuse of AI-generated content in copyright infringement and malicious activities, exacerbating the urgency for robust detection methods. With the rapid advancements in generative AI technologies like DALL-E, OpenSora, fotorai which can create highly realistic images almost instantaneously, there is a critical need for equally advanced detection counterparts [28]. Despite advancements in detection technologies, current research often overlooks real-time application, which is crucial for immediate mitigation of deepfake threats. This research aims to investigate these gaps and propose solutions to enhance the accuracy and efficiency of real-time deepfake detection systems.

1.3.2 Limitations of Current Deepfake Detection Models

Current deepfake detection models often fail to achieve the necessary accuracy for reliable real-time application. For instance, [2] introduced MesoNet, a compact CNN designed for deepfake detection. However, its performance is hindered by limited accuracy and poor generalization across diverse datasets. Similarly,stated in [7] evaluated that several deepfake detection methods using the FaceForensics++

dataset struggle with high false positive and false negative rates. These shortcomings are further explained by [10], who noted that existing models do not adequately address the rapid evolution of deepfake technology such as handling the new features of the newly generative deepfake contents which has created a significant gap in detection accuracy in addition to real-time detection.

1.3.3 Gaps Between Generative Models and Current Detection Technologies

AI image generative models are capable of creating highly realistic images in real-time are widely available in internet platform. In contrary to that, there is a significant gap in the development of corresponding real-time detection technologies.As reported in [9], tools that can produce highly sophisticated deepfakes are spreading quickly. In the contrary, the evolution of detection methods isn't keeping up. This gap between the advanced generative models and the slower progress in detection technology could lead to a greater risk of misuse and misinformation. Furthermore, as [3] points out, the challenge of detecting deepfakes is compounded by the current methods' inability to match the rapid and clever production of these fakes. This underscores the urgent need for reliable in accuracy and real-time detection systems that can be trusted to respond immediately.

1.3.4 Dataset Limitations in Deepfake Detection

The lack of comprehensive and up-to-date datasets for training and evaluating deepfake detection models is a major obstacle in developing counterpart (AIG) detection models in recent times. As AI-generated images and videos are continuously improving, existing datasets quickly become outdated due to new generative features.As a result, these old datasets immediately fail to represent the latest advancements in deepfake creation techniques in newly research. Research by [8] and [12]s highlights this issue, noting that the dynamic nature of AI image generation demands continuous updates to datasets to ensure that detection technologies remain effective. This gap hampers researchers' ability to develop robust detection systems that can adapt to new and emerging threats, as evidenced by the findings of [13].

1.3.5 The Need for Multiclass Classification

Existing research predominantly focuses on distinguishing real images from synthetic ones which is a binary classification problem. However, AI-generated images encompass a broader range, including images created by combining different faces (synthetic) and purely AI-generated images. Addressing this broader spectrum requires a multiclass classification approach, with classes for real, synthetic, and AI-generated images. According to [18], highlighted this necessity in his research, noting that while this approach adds complexity, it is crucial for achieving

accurate detection.

Given these challenges, the central question this research seeks to answer is:
How effective are advanced convolutional neural network (CNN) architectures, specifically ResNet50, when combined with sophisticated preprocessing techniques, in achieving high accuracy and real-time detection capabilities for multiclass detection of AI-generated images?

This research aims to answer the question by exploring the effectiveness of advanced CNN architectures, specifically ResNet50 and Xception, enhanced with sophisticated preprocessing techniques. By employing better preprocessing techniques, such as advanced data augmentation and noise reduction, ResNet50 has shown superior performance over other models, resulting in reduced computation time. Additionally, by training the models with a multiclass classification approach—distinguishing real, synthetic, and AI-generated images—our models demonstrate advanced capabilities and improved accuracy in real-time detection scenarios.

Our study seeks to contribute to the development of more reliable and efficient real-time deepfake detection methods by addressing the limitations of current models and the inadequacies in existing datasets. Throughout our research, we have focused on improving the accuracy and efficiency of deepfake detection systems in real time that aim to provide robust solutions to the increasingly complex threat posed by AI-generated images and videos.

1.4 Research Objectives

The core purpose of this research establishes CNN-based architectural development to assess ResNet-50 as a base alongside Xception and a bespoke CNN system for categorizing real images versus synthetic images and AI-generated images. The proposed framework has been placed on patch(region)-based image processing methods to develop detection accuracy and enable real-time framing in generative models. This involves:

1. Implementing region-based preprocessing using a cascade classifier to extract relevant image regions, combined with data augmentation to optimize model generalization.

2. Training and evaluating ResNet50, Xception, EfficientNetV2B0, VGG16, and MobileNetV3small to distinguish between real, synthetic, and AI-generated images and videos in real time.

3. Comparing the models in terms of classification accuracy, computational efficiency, and inference speed, with a primary focus on ResNet50 due to its superior real-time performance.

4. Developing a robust deepfake detection framework, contributing to efforts in ensuring the integrity and authenticity of visual content in the digital domain.

5. Provide a comprehensive solution to the existing research and that act as a counterpart of our model to support the ongoing advancement of generative models and misuse of real time deepfake.

Chapter 2

Literature Review

Deepfake technology has revolutionized image and video synthesis in a new height with the help of employing advanced artificial intelligence (AI) techniques to create highly realistic images and videos that are often indistinguishable from real ones. AI-generated images(AIG) and videos that are produced through Generative Adversarial Networks (GANs) like StyleGAN and other latest [AIG] models, represent a significant leap in the ability to produce deepfake content. These images are generated by training neural networks to mimic the characteristics of real images that leads to visually convincing and difficult to differentiate from authentic media, stated in [4]. Genrative image model like DALL-E, fotorai, HixAi and Open-Sora are precisely trained with a large dataset on the CLIP model can produce highly accurate and detailed content that further complicate the detection and verification of such AI-generated images.

Having a reliable model to detect advanced AI-generated images is more important than ever. As these technologies continue to evolve, they bring serious challenges to information integrity and security. Without proper detection in real-time, they could be misused to spread misleading or even harmful content. That's why developing effective detection methods is important to help and protect against these risks and ensure trust in digital content.

Synthetic images are any images created using computer algorithms, where different facial features are sourced and combined to form a complete face. This process, often recognized as deepfake that has become increasingly complex with advancements in artificial intelligence [12]. They are widely used in areas like computer graphics, virtual reality, and video games. Traditional synthetic images are designed through explicit programming, while deepfakes rely on machine learning to automatically generate content that looks incredibly realistic. This automated process makes detecting and verifying authenticity much more challenging [6].

Since AI-generated images are often highly realistic, people can easily mistake them for real ones. This growing confusion highlights the need for advanced detection models that can accurately differentiate between real, AI-generated, and traditionally synthetic images and videos in real-time. In our research, we tackle this issue by using a multiclass classification approach with ResNet50 as a base model. This method helps us categorize images into different classes more effectively, making our

detection system more reliable and robust.

2.1 Related Works

Deepfakes' technology has developed significantly.Initially, most artifical pictures and videos were first created by hand; users would physically edit images in Photo-Shop. Early fakes were easier to spot since they sometimes featured clear defects such as strange facial distortions, weird lighting, or absent elements. But the application of AI-powered technologies like Generative Adversarial Networks (GANs) has entirely overturned the game.Deepfakes have developed beyond simple goofy filters or stylized images; they are now almost exactly like genuine footage.Technologies like DALL-E, StyleGAN, and OpenSora can create incredibly lifelike pictures and videos, raising synthetic media to new heights. This level of realism makes traditional detection methods almost useless, forcing researchers to come up with smarter ways to spot AI-generated content. Since we have moved from manual editing to fully automated generation, detecting deepfakes has become more challenging than ever. That's why the development of advanced detection systems is more important now than ever before.

Detection in real-time: A Comparative Analysis **so** provides detailed comparisons of various detection approaches, with a focus on static photo deep fakes. The study lacks a focus on real-time video analysis, especially in high-resolution situations where generative models like DALLE-3 and 4 can produce dynamic and complicated content. Our research bridges this gap by focusing on real-time video detection and implementing a more effective and helpful way for deployment on sophisticated models for detection.

Deepfake Detection by Exploiting Surface Anomalies: Stated in [20] stated a technique to detect deepfakes by identifying anomalies in the surface textures of the human face. This method is excellent for detecting photos but fails with videos when deepfake discrepancies extend beyond superficial elements and involve complex motion abnormalities and temporal anomalies.A study published in [23] stated that, this not enough to look for surface anomalies in deepfake videos; they often have subtle flaws that can only be found by looking at several frames over time. Additionally, research in IEEE Transactions on Information Forensics and Security [11] emphasizes that deepfake videos can introduce inconsistencies between frames, such as unsmooth lip movements, which are not readily apparent in a single frame. To monitor changes between frames and identify abnormalities in real-time, our study expands on this idea by integrating temporal feature extraction. By extending detection to encompass whole video analysis rather than only surface anomalies, our methodology provides a more robust and dependable solution for recognizing advanced deepfakes in video frames.

Real word challenges and controll settings environment: Another research stated in [17] investigates the potential of deep learning to improve the detection of deepfakes. Although deep learning models exhibit significant accuracy in controlled settings, they frequently encounter difficulties when used in real-world, high-resolution, and dynamic contexts. One big problem is that there is a trade-off

between accuracy and computational . This is because high-performing models often need a lot of processing power, which makes real-time recognition difficult. By refining the model architecture and integrating preprocessing methods like face identification and grayscale conversion, our study tackles this problem. These enhancements reduce processing duration while maintaining precision, rendering our approach not only quicker but also more effective in identifying deepfakes in practical contexts.

Theoretical vs Practical Solution: Another work on "Deepfake Video Detection, Challenges and Opportunities" [29] ,provides a comprehensive survey of deepfake video detection. The solution remains mostly theoretical and does not offer a practical solution for video deepfakes. Our work is distinct in its practical approach and has been implemented and practically examined in an efficient model that can detect both in real-time with effective accuracy.

Enhancing Practicality and Efficiency of Deepfake Detection [24] discusses the importance of efficiency in detection systems, yet many of the proposed solutions still rely on heavy computational resources that are not suitable for deployment in real-time environments. Our research offers an optimized, lightweight model capable of real-time video detection while maintaining high accuracy, ensuring practicality in real-world applications .

Identity Specific Manipulation: In Anticipation of Perfect Deepfake: Identity-Anchored Detection [34] emphasizes identity-based detection strategies, but it focuses on identifying specific individuals in manipulated content. This approach, while useful, does not generalize well to deepfakes where identity manipulation is less apparent. Our approach broadens the scope by focusing on both general and identity-specific manipulations, ensuring better performance across various deepfake scenarios.

Adressing ongoing technologies and provide faster solution : The Tug-of-War Between Deepfake Generation and Detection [31] explores the ongoing arms race between generation and detection technologies, highlighting the difficulty in staying ahead of new generative techniques. However, the proposed solutions are still reactive, often addressing deepfakes only after they have been generated. Our work takes a more proactive approach by continually adapting to new generative models, ensuring

Emerging Trends and Future Directions [15] discusses the latest advancements in deepfake detection, focusing on emerging trends and the future trajectory of detection methods. This paper highlights the increasing complexity of deepfakes as generative models improve, stressing the need for more robust and adaptive detection techniques that can evolve with these advancements. While many existing systems [27] focus on static detection, particularly for images and faces, they struggle with dynamic and high-resolution video manipulations, especially in real-time applications.Our research directly addresses this issue by improving detection accuracy through the use of advanced temporal feature extraction and multi frame analysis, which helps capture more nuanced manipulation patterns in videos. Un-

like previous works that are limited to face detection, our approach extends to the detection of deepfakes across the entire video, considering both spatial and temporal inconsistencies. Moreover, our model is designed to be efficient for real-time applications, offering a solution that is both scalable and adaptable to various video formats and resolutions. By incorporating cutting-edge AI techniques and refining them for practical use, our research aims to push the boundaries of deepfake detection and stay ahead of emerging threats, ensuring a solution that evolves alongside new advancements in generative models.

Most current research on deepfake detection tends to focus on detecting facial alterations or surface-level anomalies that it often falls short when it comes to real-time video deepfakes. Our research aims to bridge this gap by expanding the scope to include full-video analysis and incorporating temporal feature extraction. In our research, this approach allows us to capture subtle manipulation patterns that span multiple frames which make the detection more accurate. We also fine-tuned our model for real-time detection without sacrificing accuracy, ensuring it's ready for practical use that has been explained in *figure 5.2*. Unlike traditional methods that focus on static images, our approach adapts to the ever-evolving techniques used to create deepfakes that offers a more reliable, scalable, and future-proof solution.

Chapter 3

Data Curation and Input

In the rapidly evolving field of deepfake technology, the effectiveness of detection models hinges crucially on the quality and relevance of their training data. We encountered significant challenges in locating publicly available datasets that combine frames of both real and AI-generated synthetic human faces. To overcome this, we developed a strategy that integrates three distinct sources: real faces from the Real-or-Fake Dataset [35], synthetic images custom made from ThisPersonDoesNotExist.com, and our custom-created AI face dataset using latest generative models that we have named the SAAT Dataset. This handmade dataset is particularly tailored to generate AI-driven human faces, filling a critical gap created by the unavailability of datasets that meet the specific needs of our research. The necessity of this approach arises not only from the lack of suitable existing data but also from our unique requirements for data uniqueness, freshness, and specificity in deepfake detection.

3.1 Goals and Strategy for Data Assembly

In assembling the SAAT Dataset, we set clear objectives focused on relevance and adaptability to keep pace with evolving deepfake technology. Our strategy involved selecting data sources that provide the latest and most sophisticated AI-generated content. By using cutting-edge AI models like DALL-E 3-4, fotorai and Opensora, we ensured our dataset includes a wide variety of the newest deepfake data that includes newest anomalies. This strategic approach not only enriches the dataset with challenging content but also allows it to adapt quickly as new deepfake methods are changing rapidly.

Figure 3.1: Sample Data of Ai Generated, Real and Syhthetic Images

3.2 Building the Dataset and the importance of Tailor-Made Dataset in our research

Customization for Accuracy: The SAAT Dataset is custom-built to meet the specific demands of our deepfake localization technology. This bespoke nature ensures that the dataset directly supports the unique analytical needs of our model which helped us to ensure get less inference time balancing with accuracy.

Ensuring Quality and Variety: We maintain rigorous standards in both the quality and diversity of the images in our dataset. High-quality data is crucial for the model to effectively learn and adapt, while a diverse array of images ensures that the model is resilient and capable of handling various real-world scenarios.

Recent studies have highlighted the critical role of diverse and high-quality updated datasets in enhancing deepfake detection that is especially important for image-based manipulations. For example, the DeepFake Detection Challenge (DFDC) Dataset [5] introduced in 2020, offers a large-scale collection of over 100,000 videos with face swapping, providing a substantial resource for developing and evaluating detection methods.

Similarly, the DeepfakeBench benchmark, presented by Yan et al. (2023) [22] integrates nine deepfake datasets and 15 state-of-the-art detection methods that offers a comprehensive platform for evaluating and comparing deepfake detection tech-

niques.

These resources underscore the importance of utilizing extensive and varied datasets to train robust deep-fake detectors capable of handling a wide range of manipulations.

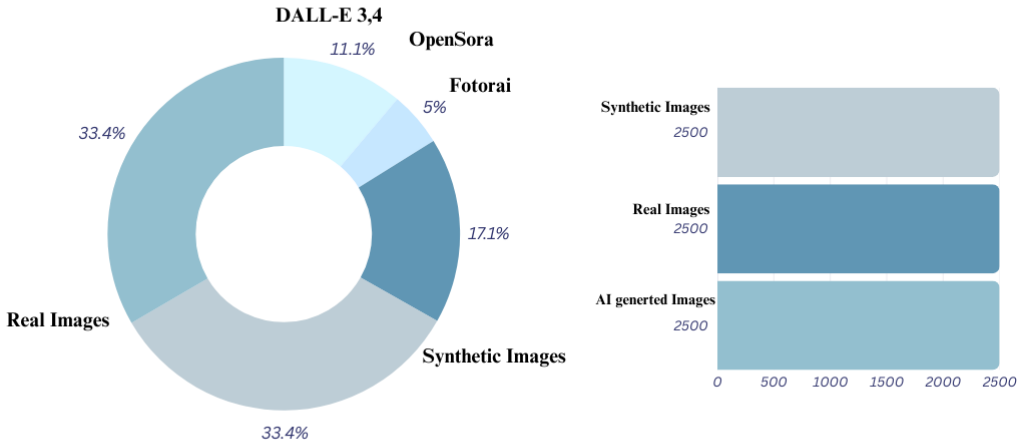


Figure 3.2: Number of Samples and distribution per Class in Dataset

The variety and balance of data in our dataset are essential for training the model that we are targeting. We've included three types of images that poses three separate class : 2500 manually created AI-generated images through latest generative models, 2500 handcrafted synthetic images, and 2500 real images from a publicly stated dataset. The reason for having more real images is that they contain natural, unique features that the model can learn from during training. These real-world details help the model understand patterns more accurately. By comparing the real images with AI-generated and synthetic ones, the model gains a better ability to distinguish between them that eventually lead to more reliable predictions when generating the final results.

3.3 Keeping the Dataset Current

As deepfake generation techniques become increasingly sophisticated, they present new challenges that our detection models must be able to identify and counter. Regularly introducing the latest examples of AI-generated images into the dataset ensures that our model continually learns from the most advanced manipulations, maintaining high accuracy in detection. This process is crucial as it adapts our model to recognize and react to the evolving complexity of deepfake features.

Moreover, the real-world application of deepfake detection demands that our dataset mirrors the current landscape of digital content. By updating the dataset to reflect contemporary deepfake scenarios, we ensure that the training environment for our model remains relevant and closely aligned with real-world challenges. This alignment is vital for the model's ability to generalize its learning to actual situations it will encounter outside the training set.

Furthermore, removing older data that no longer represents the newer and updated anomalies of deepfake technology helps to streamline the training process. It ensures that our model does not waste resources learning from or retaining outdated characteristics that are no longer prevalent in new deepfakes. As well , as it does not make the parameter count unnecessarily larger which help us reducing the inference time when detecting deepfakes in real time from live videos. This not only enhances the efficiency of the model but also prevents overfitting to historical patterns that may diminish its effectiveness against current threats.

By continuously updating the SAAT Dataset, we enhance the robustness and scalability of our detection model, preparing it to function effectively across diverse platforms and scenarios. This ongoing process of dataset curation is not just a technical requirement but a strategic imperative that significantly influences the detection model's operational success and longevity. Keeping the dataset current ensures that our technology remains at the forefront of the fight against digital deceit, equipped to tackle new challenges as they arise.

3.4 Data Pre-Processing

Preprocessing ensures that the input data is in the optimal form for training the ResNet50-based SAAT model. By augmenting the dataset and standardizing the inputs, the model can learn more effectively, leading to improved detection of real, synthetic, and AI-generated images. This rigorous approach to data preprocessing is fundamental to achieving high performance and reliability in real-time deepfake detection.

3.4.1 Resize Image to (224, 224):

All input images are resized to 224x224 pixels, the default input size required by ResNet50. This uniform size ensures consistency across the dataset and facilitates

efficient processing by the convolutional layers.This approach particularly saves time in training which makes the model efficient in memory management.

The resizing operation for input images is defined as:

$$I' = \mathcal{R}(I, 224, 224) \tag{1}$$

where:

- I is the original input image.
- $\mathcal{R}$ represents the resizing function (bilinear, bi-cubic, or nearest-neighbor interpolation).
- I' is the resized image with dimensions (224×224).

Alternatively, using a scaling transformation:

$$I' = \mathcal{R}(I) = I \times \begin{bmatrix} \frac{224}{H_o} & 0 \\ 0 & \frac{224}{W_o} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

where H_o and W_o are the original height and width of I, respectively.

3.4.2 Pixel value Rescaling:

After that, pixel values are rescaled from the range [0, 255] to [0, 1]. This normalization step improves the numerical stability of our model during training by reducing the variance in pixel values which is also a part of making the model efficient.

$$I' = \frac{I}{255} \tag{3}$$

where:

- I is the original image with pixel values in the range $[0, 255]$.
- I' is the rescaled image with normalized pixel values in the range $[0, 1]$.

3.4.3 Convert Image from RGB to BGR:

ResNet50 is pre-trained on the ImageNet dataset, which uses the BGR color space. Converting images from RGB to BGR aligns with the pre-trained model expectations, ensuring compatibility and enhancing feature extraction.

$$I' = I \cdot P \tag{4}$$

where:

- I is the input image in RGB format.
- P is the permutation matrix:

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.1}$$

- I' is the transformed image in BGR format.

3.4.4 Width, Height,and Shear Shift (Max 20 % Shift):

Geometric transformations, including translations and shear shifts, are applied to the images with a maximum shift of 20 %. These transformations increase the diversity of the dataset by creating variations that mimic natural distortions and movements in real-world scenarios.

The transformation matrix is defined as:

$$T = \begin{bmatrix} 1 & \lambda_x & t_x \\ \lambda_y & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5}$$

where:

- $\lambda_x, \lambda_y \sim U(-0.2, 0.2)$ are shear factors sampled from a uniform distribution in the range $[-20\%, +20\%]$.
- $t_x = \alpha W$, $t_y = \beta H$ are translations, where $\alpha, \beta \sim U(-0.2, 0.2)$.
- W and H are the original width and height of the image.

The transformed image coordinates (x', y') are obtained by applying T to the original coordinates (x, y):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5.1}$$

3.4.5 Canvas Filling:

Fill the Empty Part of the Canvas by Sampling the Nearest Pixel, when geometric transformations result in empty regions in the image, these regions are filled by sampling the nearest pixel values. This maintains the integrity of the image and prevents artifacts that could confuse the model during training.

When geometric transformations create empty regions, the missing pixels are filled by sampling the nearest valid pixel value:

$$I'(x,y) = I(x',y'), \quad \text{where} \quad (x',y') = \arg\min_{(u,v)\in\Omega} \|(x-u, y-v)\| \tag{6}$$

where:

- $I'(x,y)$ is the filled pixel value at (x,y).
- $I(x',y')$ is the nearest valid pixel value from the original image.
- (x',y') is chosen as the closest point in the valid pixel set Ω, minimizing the Euclidean distance.

3.4.6 Random Horizontal Flip:

Images are randomly flipped horizontally. This augmentation technique simulates real-world variations and helps the model become invariant to horizontal orientation changes, thereby improving generalization.

A horizontal flip transformation is applied with probability p:

$$I'(x,y) = \begin{cases} I(W-x,y), & \text{if flip occurs} \\ I(x,y), & \text{otherwise} \end{cases} \tag{7}$$

where:

- $I'(x,y)$ is the transformed image.
- $I(W-x,y)$ flips the pixel horizontally.
- W is the image width.

3.4.7 Zoom Between -20% to 20%:

Random zooming within the range of -20% to +20% is applied to the images. This augmentation helps the model learn to recognize objects at different scales, enhancing its robustness to variations in object size and distance.

A random zoom transformation within the range $[-20\%, +20\%]$ is applied:

$$I'(x,y) = I\left(\frac{x}{s}, \frac{y}{s}\right), \quad s \sim U(0.8, 1.2) \tag{8}$$

where:

- $I'(x,y)$ is the zoomed image.
- s is the randomly sampled scaling factor from a uniform distribution $U(0.8, 1.2)$.
- $I(x/s, y/s)$ performs the scaling operation.

3.4.8 Convert Labels to One-Hot Encoding (Categorical):

Labels are converted to one-hot encoded vectors, which represent the class labels (real, synthetic, AI-generated) in a format suitable for categorical cross-entropy loss. This encoding allows the model to output probabilities for each class, facilitating multi-class classification.

Each label y is converted into a one-hot encoded vector:

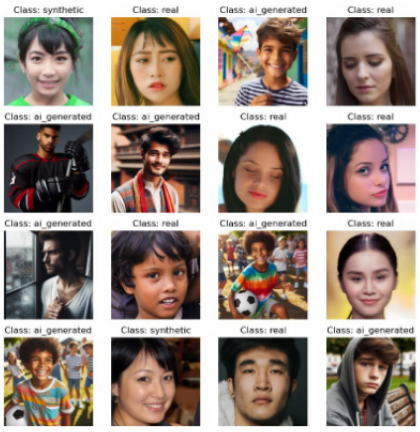
$$y_i = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where:

- y_i is the one-hot encoded vector.
- k is the class index of the label.
- i represents the class categories (real, synthetic, AI-generated).

3.5 Color Conversion and Transformation application visualization

Visualizating Color Conversion and Pre Processing: Each applied technique on each images can be picturized through this image.



Input Data Pre process

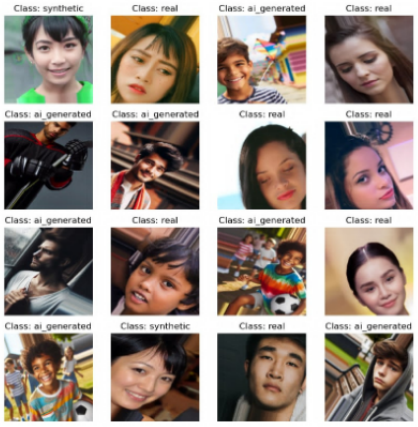


Image Data after transformation and data augmentation applied

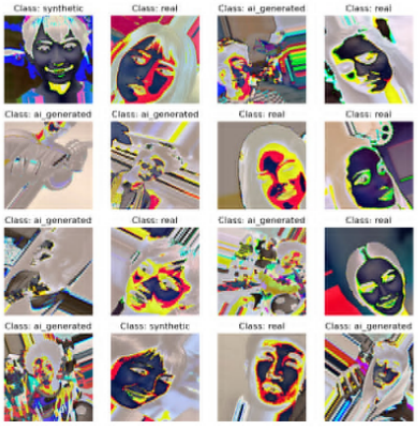


Image Data After Pre process Color Conversion

Figure 3.3: Comparison of picture after transformation and color conversion

Chapter 4

Methodology

A critical aspect of our study lies in how images can assist in identifying irregularities within videos while simultaneously reducing inference time when detecting abnormalities. Our unique approach identifies the fact that videos are composed of multiple frames, with the integration of these frames forming the video itself. By extracting individual frames from videos, our algorithm performs a detailed analysis of each frame, identifying abnormalities. These analyzed frames are then combined, and the integrated data is used to generate the final result. This methodology allowed us to overcome the challenge of accessing direct video datasets and provided a novel means of analyzing video content for our research.

The process of creating the dataset begins with collecting a diverse range of images that include real, synthetic, and AI-generated content. This dataset is categorized into three groups: Real Images, Fake Images, and Synthetic Images. The real images are sourced from a publicly available dataset [14] named as "Real vs Fake Faces" dataset. The fake images are manually created using generative deep model, while the synthetic images are generated using a publicly sourced generative model [16]. A key point of our approach is combining real images with these manually made AI-generated and synthetic images that allowed our proposed model to detect and analyze strong AI-generated content effectively. These images are then pre-processed and trained in our model to enhance its ability to identify irregularities.

The methodology for this study employs a structured approach to detect real, synthetic, and AI-generated images using advanced transfer learning techniques combined with sophisticated preprocessing methods. Along with the model, one golden rule to highlight, our dataset is processed with a technique which divided the data to 60 percent training data, 20 percent validation and 20 percent test data. Finally, as described at 3.4, the process encompasses several stages, from data input to the final classification decision. This methodology provides a robust framework for training and evaluating the model, ensuring accuracy and reliability in the results.

4.1 Execution Plan

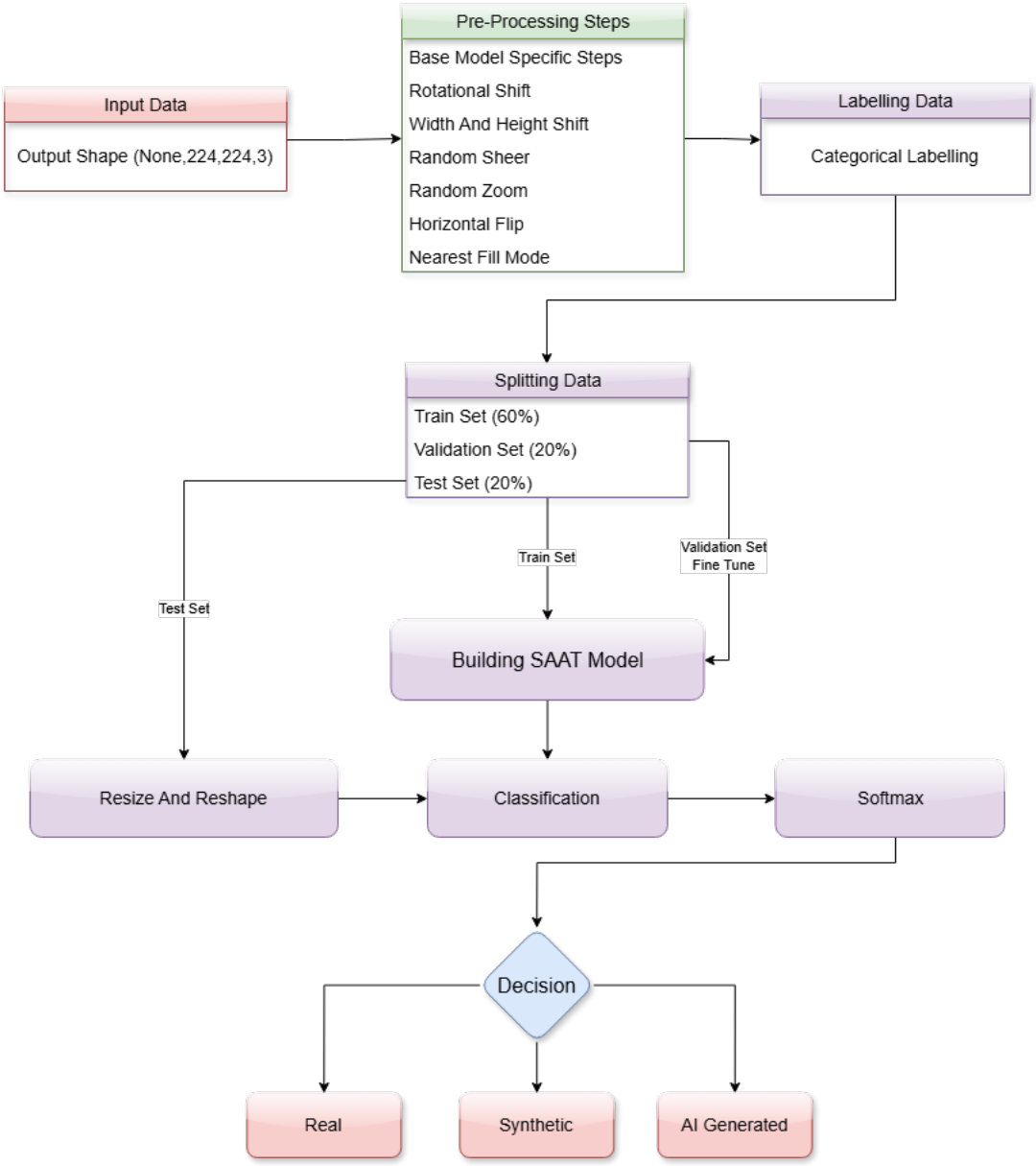


Figure 4.1: Execution plan and Data Training with Decision

Chapter 5

Model Implementation and Architecture

5.1 Implementation of Model

This section demonstrates our approach to developing a real-time model capable of detecting AI-generated, real, and synthetic frames in images and videos. We explored several advanced convolutional neural network (CNN) architectures, including ResNet50, Xception, EfficientNetV2B0, VGG16, and MobileNetV3-Small. After conducting extensive research and comparison, we decided to use ResNet50 as our base model. ResNet50 was chosen because of its optimal balance between detection performance, speed, and accuracy when processing video frames. Additionally, we have made modifications to the base model to further enhance its accuracy and reduce inference time, aligning with the goals of our research.

Our model leverages a pre-trained CNN (ResNet50) as a base layer to extract high-level spatial features from input images of shape (224,224,3). These features are then refined through a deep feature refinement block consisting of a 3x3 convolutional layer with ReLU activation, batch normalization, and spatial dropout to enhance robustness. An attention mechanism further improves feature selection by applying a 1x1 convolution, element-wise multiplication, and global average pooling, ensuring that the most relevant information is retained. The classifier head, designed with regularization techniques such as dropout (30% and 20%) and batch normalization, processes the refined features through two dense layers to improve generalization and stability. Finally, the output layer predicts one of three categories—AI-generated, real, or synthetic.

What makes this model unique is its integration of an attention mechanism to emphasize crucial spatial features, combined with a highly regularized classifier head that ensures efficient real-time performance while maintaining accuracy.

The primary function of the SAAT architecture is to determine whether video frames are AI-generated, real, or synthetic by analyzing human faces in the video. This ability is crucial for identifying and countering deepfake videos.

5.2 Model Architecture

Our proposed model architecture is illustrated in the image below.

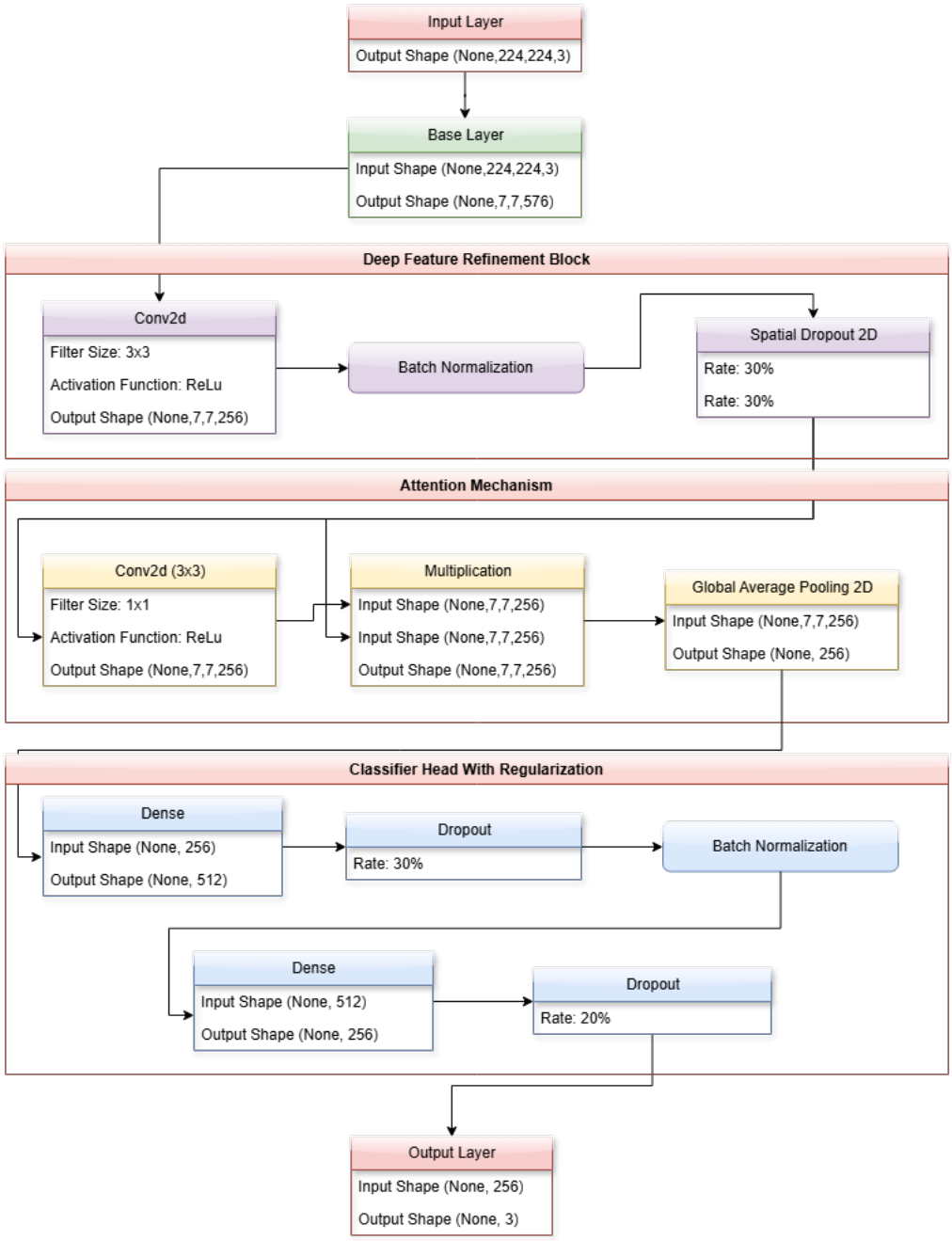


Figure 5.1: Proposed Model Architecture

5.2.1 Model Architecture and Layer-Wise Description

Input Layer: The input layer accepts images of shape $(224 \times 224 \times 3)$, which represent RGB images of size 224×224 pixels.

$$X \in R^{224\times224\times3} \tag{10}$$

Feature Extraction using Pretrained ResNet50: The base feature extractor is the ResNet50 model pretrained on ImageNet. This extracts high-level spatial features from the input image. The ResNet50 model consists of a series of convolutional and pooling layers but excludes the fully connected layers.
Let $f(X)$ represent the transformation applied by the ResNet50 base:

$$F = f(X), \quad F \in R^{7\times7\times2048} \tag{11}$$

where the final feature map from ResNet50 has spatial dimensions 7×7 with 2048 channels.

Additional Convolutional Layer To refine features, an additional 2D Convolutional Layer is added:

$$F' = \sigma(W * F + b) \tag{12}$$

where W and b are the weight and bias parameters, $*$ denotes convolution, and $\sigma(\cdot)$ represents the ReLU activation function:

$$\sigma(z) = \max(0, z) \tag{12.1}$$

This layer has 256 filters of size (3×3) with same padding.

Batch Normalization: Batch normalization stabilizes activations and accelerates convergence by normalizing feature maps:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{13}$$

where μ and σ^2 are the mean and variance of the batch, respectively.

Spatial Dropout: To prevent overfitting, SpatialDropout2D with a dropout rate of 0.3 is applied, which randomly drops entire feature maps during training.

Attention Mechanism: An attention mechanism is integrated to refine feature maps using a 1x1 convolution:

$$A = \sigma(W_A * F' + b_A) \tag{14}$$

where A is the attention map and σ is the sigmoid activation function. The feature maps are then scaled element-wise:

$$F'' = F' \odot A \tag{14.1}$$

where $\odot$ represents element-wise multiplication, allowing the model to focus on important regions.

Global Average Pooling (GAP): The feature maps are reduced to a single vector per channel using GAP:

$$y_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F''(i, j, c) \tag{15}$$

where H and W are the spatial dimensions.

Fully Connected Layers: Two fully connected layers refine the extracted features:

$$H_1 = \sigma(W_1 y + b_1), \quad H_2 = \sigma(W_2 H_1 + b_2) \tag{16}$$

where the first layer has 512 neurons (with dropout 0.3 and batch normalization) and the second layer has 256 neurons (with dropout 0.2).

Output Layer: A softmax classifier is applied for multi-class classification (3 classes):

$$p_c = \frac{e^{z_c}}{\sum_{j=1}^{3} e^{z_j}} \tag{17}$$

where p_c represents the probability of class c, ensuring sum-to-one probability constraints.

Model Compilation: The model is compiled using the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy loss:

$$L = -\sum_{c=1}^{3} y_c \log(p_c) \tag{18}$$

where y_c is the true label and p_c is the predicted probability.

Training Strategy

ModelCheckpoint: saves the best model based on validation loss.

ReduceLROnPlateau: reduces the learning rate if validation loss stagnates.

Epoch: Training is conducted for 40 epochs with validation monitoring.

5.3 Model Inference Extract Process Pipeline

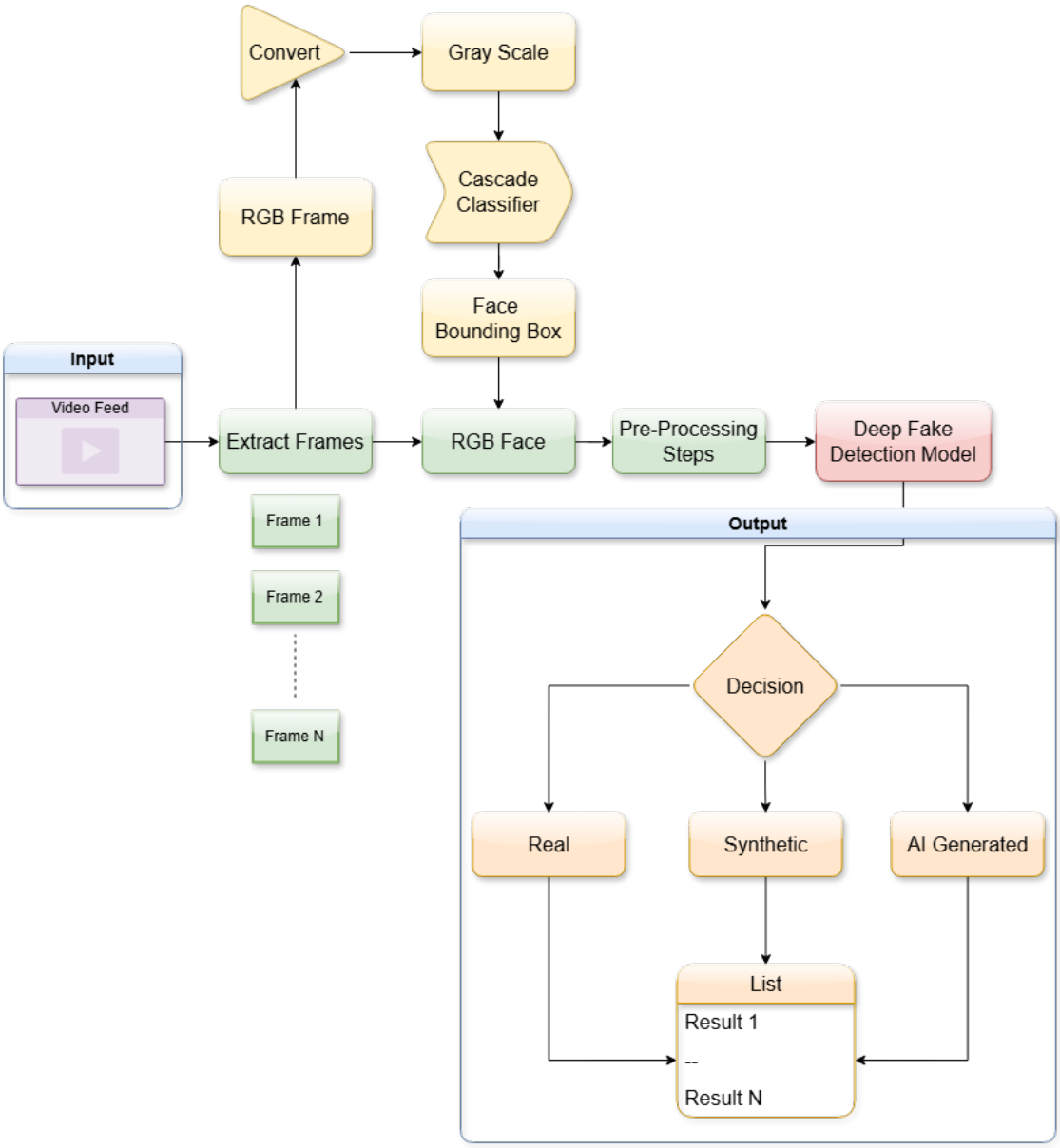


Figure 5.2: Model Inference Extract Process Pipeline

Architecture and Processing Flow of SAAT

5.3.1 Initial Setup and Model Loading

The SAAT model begins its operation with a crucial startup phase where it is loaded and initialized. This preparation is essential as it primes the model for real-time processing of video frames directly from the live feed, ensuring it operates smoothly and efficiently from the moment it starts.

5.3.2 Frame Capture and Iteration Process

As the model begins analyzing the video stream, it captures frames individually. Each of these frames is processed through what we term an 'iteration'. This process involves extracting, analyzing, and classifying each frame based on predetermined conditions that guide the flow of operations to ensure each frame is handled appropriately.

5.3.3 Enhancing Efficiency in Frame Processing

To optimize the model's performance and significantly reduce the time it takes to infer results, we implement several strategic approaches:

Grayscale Conversion: Immediately after capture, each frame is converted to grayscale. This step significantly reduces the number of parameters the model needs to process, thus lowering the computational load and speeding up the overall inference time.

Conditional Iteration: If a frame, once turned to grayscale, shows no face, the model pauses, effectively stopping further processing of that frame and awaiting the next. This selective processing ensures that computational resources are reserved for frames with relevant content, boosting efficiency.

Face Detection and Image Conversion

In frames where a face is detected,

Face Detection: A specialized face detection algorithm named as cascade classifier scans the gray-scale image to find and create a bounding box around any face detected.

Conversion to RGB: After detecting a face, the frame is converted back to RGB, a necessary step as the color data is crucial for the subsequent pre-processing and classification stages.

Pre-processing for Model Compatibility,

The RGB image is then treated through a series of pre-processing steps stated in [3.4]. These steps are designed to standardize the input data for aligning various image qualities with what our model requires to perform its classification tasks effectively and consistently.

5.3.4 Classification and Output Generation

Classification Process: The model classifies each preprocessed frame into one of three categories—real, synthetic, or AI-generated. This classification is marked by assigning a numerical value ([1,0,0], [0,1,0], [0,0,1]) to each category that has been stated in [5.2].

$$C_f = \begin{cases} [1,0,0] & \text{if frame } f \text{ is classified as real} \\ [0,1,0] & \text{if frame } f \text{ is classified as synthetic} \\ [0,0,1] & \text{if frame } f \text{ is classified as AI-generated} \end{cases} \tag{19}$$

Output Aggregation: The classification results for each frame are sequentially stored in a list, compiling the data for later analysis.

The average classification result for all frames is given by:

$$\hat{C} = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{20}$$

Decision Mechanism: At the end of a session or upon activation of a specific trigger, the model assesses the compiled data. It calculates the average of these outputs to determine the most likely category of the video content. If the results are not clear enough, the model is designed to reprocess additional frames, thus refining the decision-making process to ensure accuracy.

The model assigns the most frequent category based on the average of classification:

$$\hat{C} = \arg\max \left(\sum_{i=1}^{n} 1_{C_i=0}, \sum_{i=1}^{n} 1_{C_i=1}, \sum_{i=1}^{n} 1_{C_i=2} \right) \tag{21}$$

If the decision is not clear, the model triggers additional frame processing:

$$\text{If } \left| \sum_{i=1}^{n} 1_{C_i=0} - \sum_{i=1}^{n} 1_{C_i=1} \right| < \epsilon \text{ or } \left| \sum_{i=1}^{n} 1_{C_i=1} - \sum_{i=1}^{n} 1_{C_i=2} \right| < \epsilon \tag{21.1}$$

Unique Features for Reduced Inference Time

Key to our proposed model's efficiency are its unique features that enhance processing speed. The ability to selectively process only frames that contain relevant data and the initial conversion of frames to gray-scale not only save time but also computational resources, allowing the model to operate effectively in real-time scenarios.

Chapter 6

Experiment, Result Analysis and Discussion

6.1 Our Proposed Models Accuracy, Loss and Perfomance Analysis on Our Custom Dataset

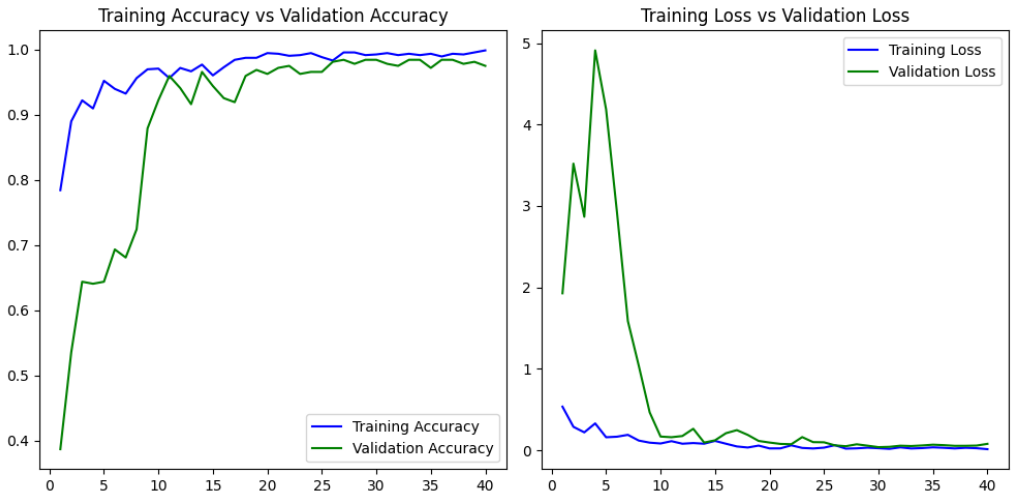


Figure 6.1: SAAT-Resnet 50-3BD Accuracy and Loss Visualization

6.1.1 Training Accuracy vs Validation Accuracy Analysis

The fig[6.1], The graph on the left shows the progression of training and validation accuracy over 40 epochs. Initially, training accuracy starts at approximately 80.2% and increases rapidly, reaching 90.5% by epoch 5. Significant improvements in validation accuracy occur in the early stages, rising from 38.7% at epoch 1 to 63.4% at epoch 5 and further improving to 75.8% by epoch 7. By epoch 10, validation accuracy reaches 90.2%, closely aligning with training accuracy. Beyond epoch 15, both accuracies stabilize, fluctuating slightly but remaining above 95%. At epoch 20, training accuracy is 98.1%, while validation accuracy is 96.7%. The final values at epoch 40 indicate 99.8% training accuracy and 98.5% validation accuracy. These trends suggest an effective learning process, with early convergence and minimal overfitting as both accuracies remain consistently high in later epochs.

6.1.2 Training Loss vs Validation Loss Analysis

The graph on the right illustrates the progression of training and validation loss over 40 epochs. Initially, validation loss is relatively high, starting at approximately 2.1 in the first epoch and spiking to 5.0 around epoch 5, indicating significant fluctuations as the model adjusts its learning. However, a sharp decline follows, with validation loss dropping below 0.5 by epoch 10. Training loss, in contrast, remains consistently low, starting at approximately 0.6 in the first epoch and gradually decreasing, stabilizing below 0.1 after epoch 15. By epoch 20, both losses are minimal, with training loss at 0.05 and validation loss at 0.12, maintaining stability through epoch 40.

The early fluctuations in validation loss suggest initial instability, but the rapid decline and eventual stabilization indicate successful learning and convergence. However, underfitting is a concern in the first few epochs, as the high loss values suggest that the model is not yet capturing the underlying patterns in the data. As training progresses, the gap between training and validation loss narrows, suggesting improved generalization. The final low loss values indicate that the model effectively fits the data without significant overfitting, achieving an optimal balance between learning and performance.

6.1.3 Confusion Matrix

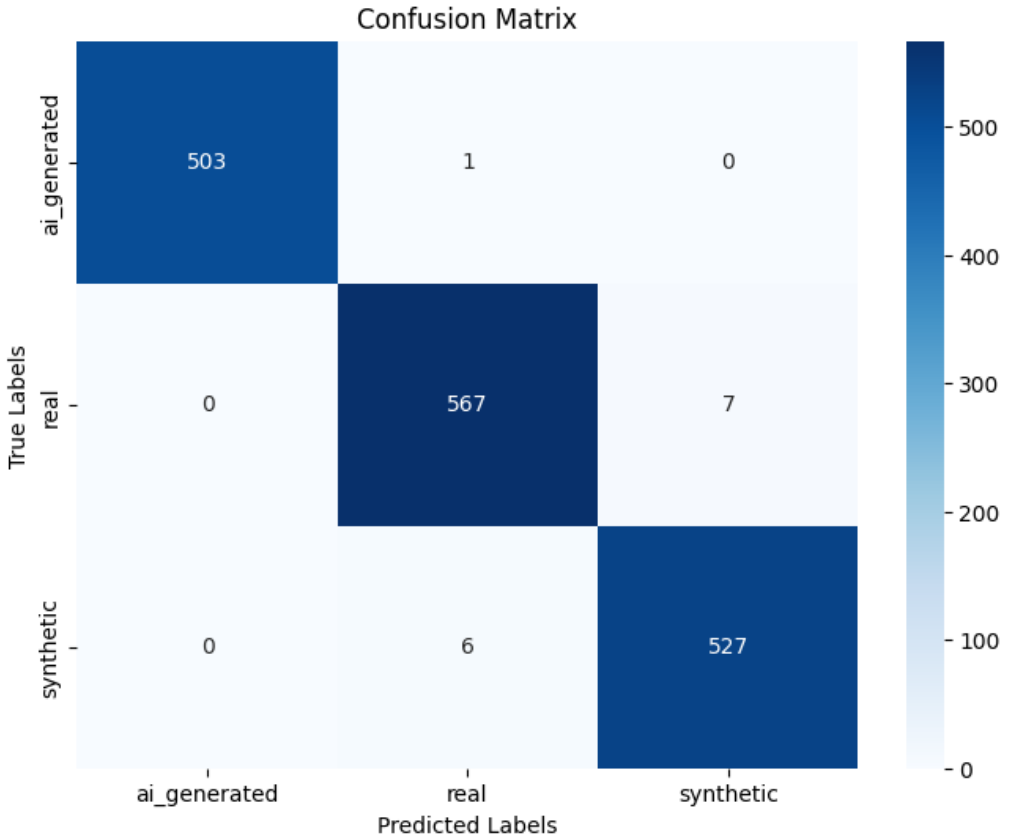


Figure 6.2: SAAT-Resnet50-3BD Confusion Matrix

In Fig[6.2], the confusion matrix provides an evaluation of the model's classification performance across three categories: AI-generated, real, and synthetic images. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate misclassifications. The model demonstrates high accuracy, correctly classifying 503 AI-generated images, 567 real images, and 527 synthetic images. Misclassifications are minimal, with only one AI-generated image misclassified as real, seven real images misclassified as synthetic, and six synthetic images misclassified as real. Notably, the model does not misclassify any AI-generated or synthetic images into each other, indicating that it effectively distinguishes between these categories. The low number of misclassifications suggests that the model has learned strong distinguishing features between real and generated content, reinforcing its robustness in deepfake detection.

6.1.4 Evaluation Score

$$\text{Precision}_{AI-Generated} = \frac{503}{503+0} = 1.0$$
$$\text{Recall}_{AI-Generated} = \frac{503}{503+1} = 0.998$$
$$F1_{AI-Generated} = 2 \times \frac{1.0 \times 0.998}{1.0 + 0.998} = 0.999$$

$$\text{Precision}_{Real} = \frac{567}{567+7} = 0.9878$$
$$\text{Recall}_{Real} = \frac{567}{567+6} = 0.9896$$
$$F1_{Real} = 2 \times \frac{0.9878 \times 0.9896}{0.9878 + 0.9896} = 0.9887$$

$$\text{Precision}_{Synthetic} = \frac{527}{527+6} = 0.9887$$
$$\text{Recall}_{Synthetic} = \frac{527}{527+7} = 0.9869$$
$$F1_{Synthetic} = 2 \times \frac{0.9887 \times 0.9869}{0.9887 + 0.9869} = 0.9878$$

Table 6.1: Precision, Recall, and F1 Score for each class of SAAT-Resnet50-3BD

Class	**Precision**	**Recall**	**F1 Score**
AI-Generated	1.0000	0.9980	0.9990
Real	0.9878	0.9896	0.9887
Synthetic	0.9887	0.9869	0.9878

6.2 VGG16 Accuracy, Loss and Perfomance Analysis on Our Custom Dataset

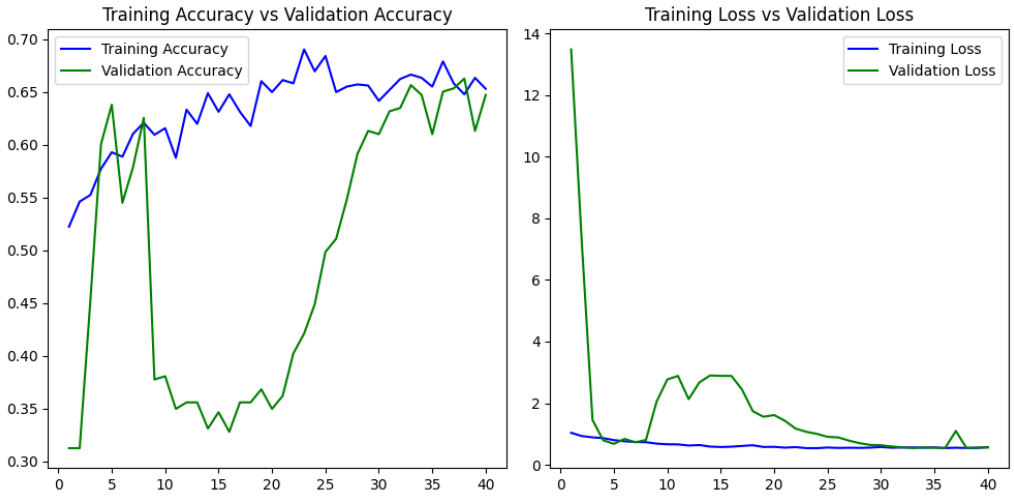


Figure 6.3: VGG16 Accuracy and Loss Visualization

6.2.1 Training Accuracy vs Validation Accuracy Analysis

In fig [6.3], training and validation accuracy over 40 epochs show a gradual improvement with some fluctuations. Initially, the training accuracy increased from 50.2% in epoch 1 to 57.8% by epoch 5, while validation accuracy exhibited a rapid rise, peaking at 65.4% in epoch 6 before experiencing instability. By epoch 10, training accuracy reached 60.9%, whereas validation accuracy declined significantly due to inconsistent generalization. Beyond epoch 20, validation accuracy improved steadily, reaching 61.2%, while training accuracy showed diminishing returns at 65.8%. By epoch 30, training accuracy stabilized at 66.5%, with validation accuracy closely following at 63.4%. The final accuracy at epoch 40 was 67.1% for training and 65.9% for validation.

6.2.2 Training Loss vs Validation Loss Analysis

In fig[6.3], training and validation loss curves highlight key aspects of the model's learning process. Initially, training loss dropped significantly from 1.45 in epoch 1 to 0.32 by epoch 5, while validation loss followed a similar trend, decreasing from 13.5 to 1.02. However, validation loss exhibited sharp fluctuations between epochs 5 and 20, peaking around 4.1, indicating that the model struggled to generalize effectively. By epoch 20, training loss remained low at 0.28, while validation loss showed gradual improvement, stabilizing near 1.53. The final losses at epoch 40 were 0.21 for training and 0.67 for validation.

The substantial gap between training and validation loss in earlier epochs suggests that the model initially suffered from underfitting, where it had not yet captured meaningful patterns in the data. The overall trend indicates that the model could not perform reasonably well.

6.2.3 VGG16 Confusion Matrix

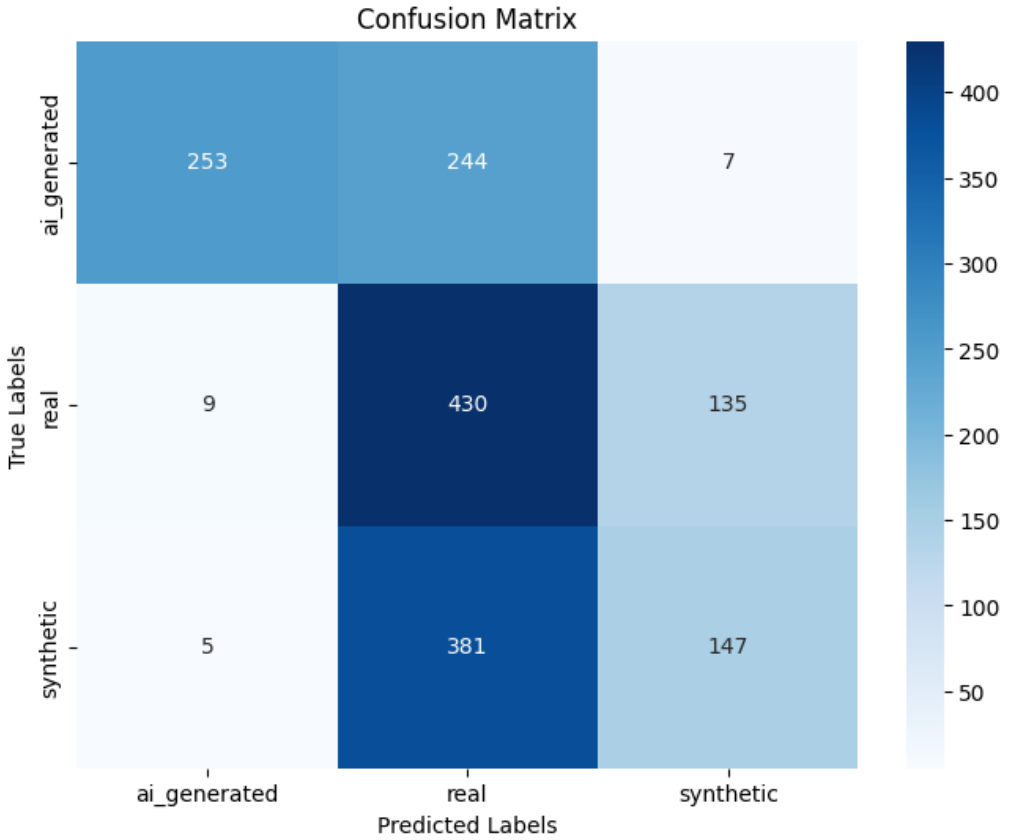


Figure 6.4: VGG16 Confusion Matrix

The confusion matrix provides insight into the classification performance across three categories: ai generated, real, and synthetic. The model correctly classified 253 ai generated samples, but misclassified 244 as real and 7 as synthetic. For real images, the model correctly identified 430 instances, but misclassified 9 as ai generated and 135 as synthetic. The synthetic category had 147 correctly classified samples, while 381 were incorrectly labeled as real and 5 as ai generated. The high miss classification rate, particularly between real and synthetic suggests that, the model struggles to differentiate between these two classes potentially due to feature similarities. This indicates room for improvement in feature extraction and decision boundary refinement.

6.2.4 Evaluation Score

Table 6.2: Precision, Recall, and F1 Score for Each Class [VGG16]

Class	Precision	Recall	F1 Score
AI-Generated	0.9500	0.5000	0.6600
Real	0.4100	0.7500	0.5300
Synthetic	0.5100	0.2800	0.3600

6.3 Xceptions Accuracy, Loss and Perfomance Analysis on Our Custom Dataset

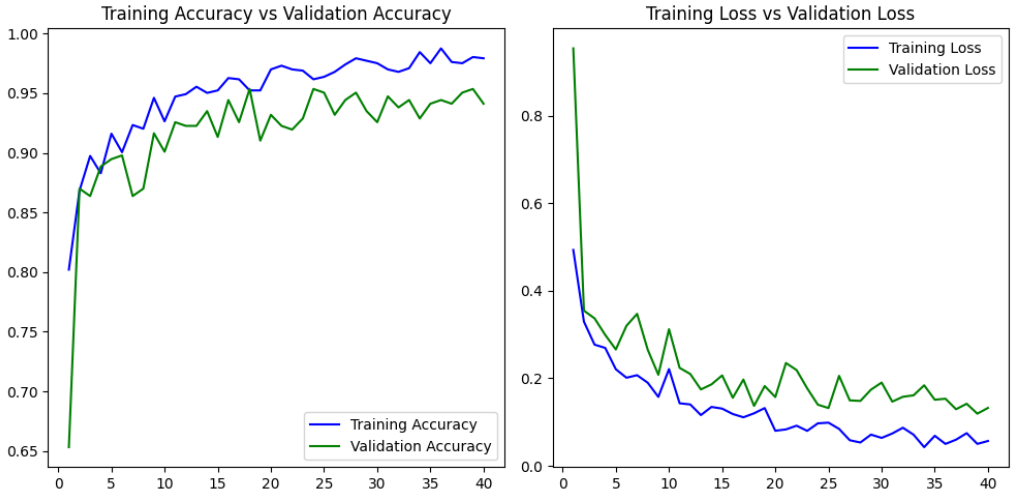


Figure 6.5: Xception's Accuracy and Loss Visualization

6.3.1 Training Accuracy vs Validation Accuracy Analysis

In Fig[6.5], training and validation accuracy per epoch demonstrate a steady improvement with diminishing returns as the model approaches convergence. Initially, the training accuracy rises significantly from 80.12% at epoch 1 to 90.34% at epoch 5, while the validation accuracy follows closely, increasing from 65.78% to 87.56%. By epoch 10, the training accuracy reaches 94.67%, with the validation accuracy at 90.12%, showing strong generalization. Between epochs 10 and 20, accuracy gains slow, with training accuracy at 96.1% by epoch 20, and validation accuracy slightly lower at 92.78%. After epoch 30, the improvements are marginal, reaching 96.4% training accuracy and 93.6% validation accuracy at epoch 40.

The narrowing gap between training and validation accuracy suggests diminishing returns rather than a model flaw, as the validation performance stabilizes at high accuracy levels. While minor fluctuations in validation loss indicate sensitivity to the data, the overall trend confirms that the model successfully learns meaningful patterns with only slight overfitting.

6.3.2 Training Loss vs Validation Loss Analysis

In Fig[6.5], training and validation loss curves exhibit a consistent downward trend, reflecting the model's ability to learn meaningful patterns from the data. Initially, the training loss drops sharply from 0.45 at epoch 1 to 0.21 at epoch 5, while the validation loss follows a similar trajectory, decreasing from 0.89 to 0.34. By epoch 10, the training loss reaches 0.12, with validation loss stabilizing around 0.25, indicating that the model is effectively minimizing error. Between epochs 10 and 20, the loss reduction slows, with the training loss at 0.08 by epoch 20, while the validation loss fluctuates slightly around 0.22. Beyond epoch 30, improvements diminish, with the training loss reaching 0.05 at epoch 40, while the validation loss remains at 0.15.

The gradual plateauing of validation loss indicates that the model has effectively learned from the dataset, with only minor fluctuations due to inherent data variability rather than fundamental model inefficiencies.

6.3.3 Xception Confusion Matrix

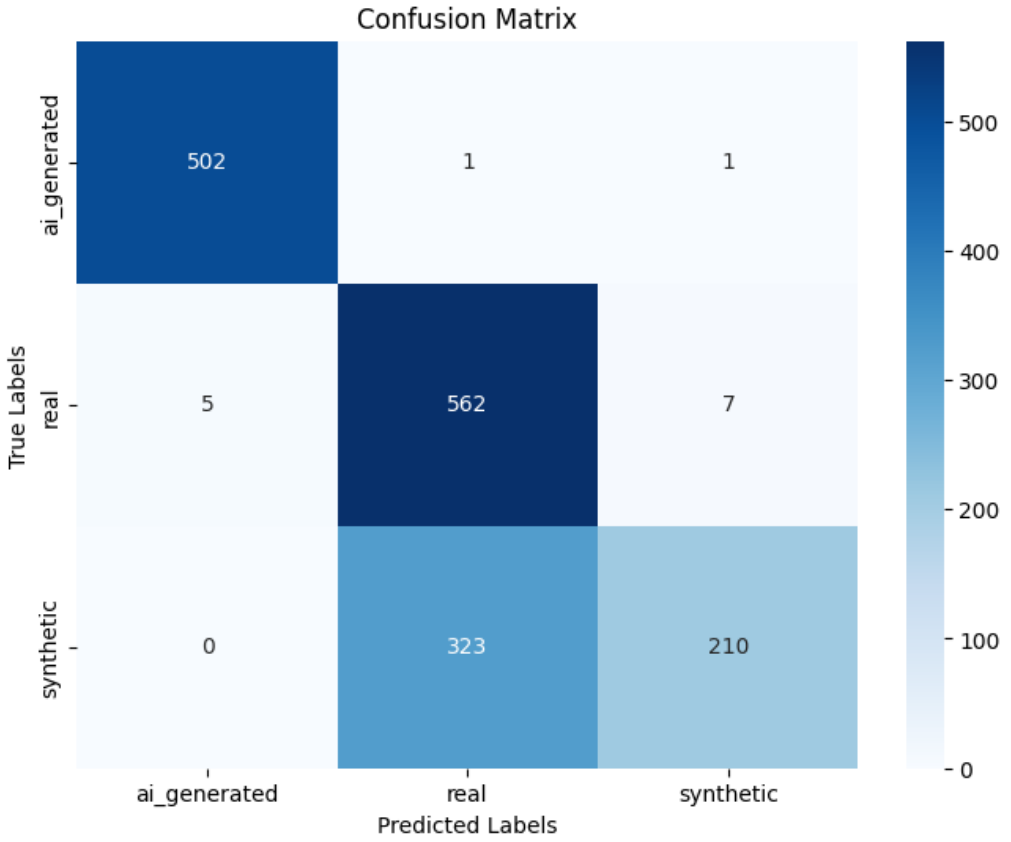


Figure 6.6: Xception Confusion Matrix

In fig[6.6], the confusion matrix, for AI-generated images, the model correctly identified 502, with only a few misclassifications (1 as real and 1 as synthetic). When classifying real images, the model performed well, correctly identifying 562 as real, while misclassifying 5 as AI-generated and 7 as synthetic. For synthetic images, the model correctly predicted 323 as synthetic, but there were more misclassifications, with 210 being labeled as real. Overall, the model showed strong performance, particularly with real and AI-generated images, though it faced more challenges with synthetic images.

6.3.4 Evaluation Score

Table 6.3: Precision, Recall, and F1 Score for Each Class[Xception]

Class	Precision	Recall	F1 Score
AI-Generated	0.9901	0.9960	0.9931
Real	0.6343	0.9791	0.7699
Synthetic	0.9633	0.3940	0.5593

6.4 EfficientNetV2B0 model's Accuracy, Loss and Per- fomance Analysis on Our Custom Dataset

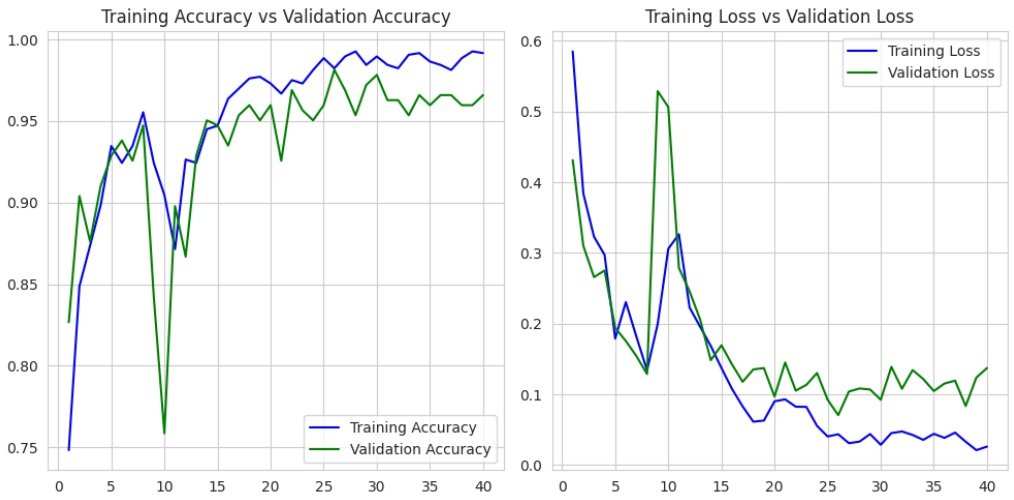


Figure 6.7: EfficientNetV2B0's Accuracy and Loss Visualization

6.4.1 Training Accuracy vs Validation Accuracy Analysis

The training and validation accuracy curves demonstrate significant improvements across epochs. At epoch 1, training accuracy was 75.2%, while validation accuracy reached 78.4%. By epoch 5, training accuracy improved to 90.5%, with validation accuracy fluctuating at 87.6%. A notable drop in validation accuracy occurred at epoch 10, reaching 76.3%, before recovering to 92.1% at epoch 15. From epoch 20 onward, training accuracy exceeded 99.0%, while validation accuracy stabilized around 96.5%. The increasing gap between training and validation accuracy in later epochs suggests overfitting. Where the model learns patterns specific to the training data that reduces generalization performance.

6.4.2 Training Loss vs Validation Loss Analysis

The training and validation loss curves indicate key trends in model performance. At epoch 1, training loss started at 0.58, while validation loss was slightly lower at 0.44. By epoch 5, training loss decreased significantly to 0.22, while validation loss showed a more unstable trend that reach's to 0.30. A notable increase in validation

loss occurred at epoch 10, rising to 0.53, suggesting a period of underfitting where the model struggled to generalize well from the training. However, by epoch 15, validation loss reduced to 0.14 that aligns more closely with training loss at 0.12. From epoch 20 onward, training loss dropped below 0.08, while validation loss fluctuated around 0.10–0.12, indicating a potential onset of overfitting. The initial underfitting phase suggests that the model needed more training to capture essential patterns, while later epochs show diminishing returns that further reductions in training loss do not translate to better validation performance.

6.4.3 Confusion Matrix of EfficientNetV2B0

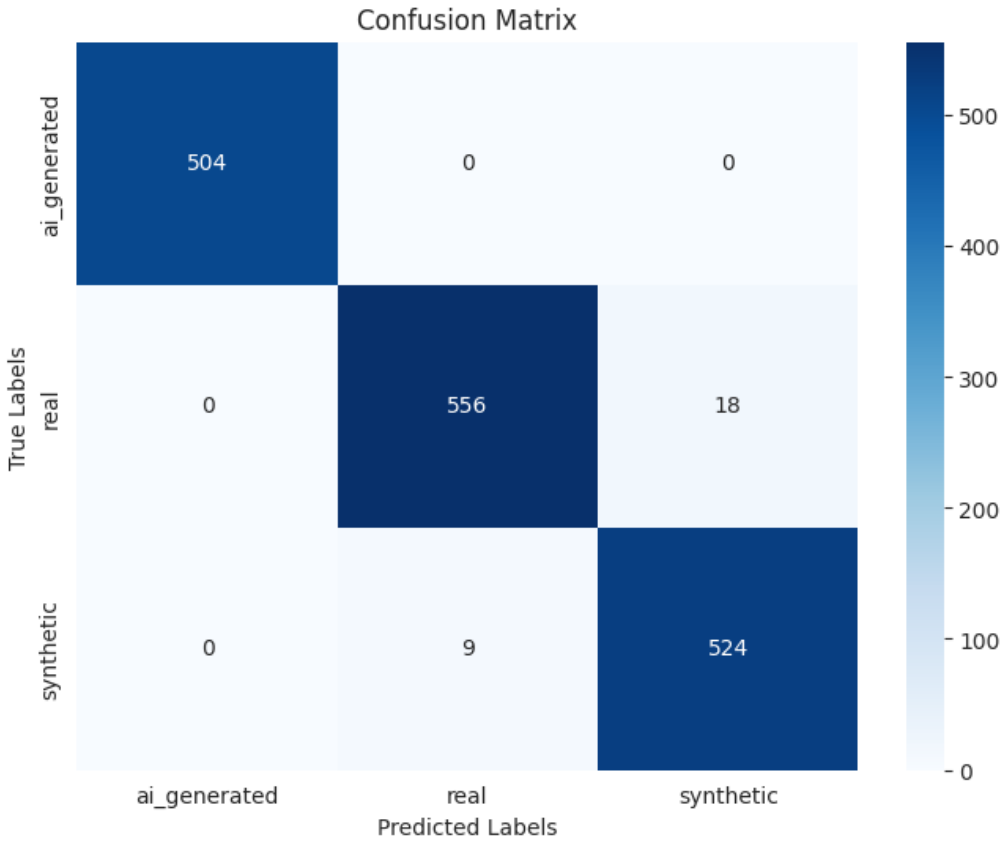


Figure 6.8: EfficientNetV2B0's Confusion Matrix

The confusion matrix demonstrates a strong classification performance across all three categories: "ai generated," "real," and "synthetic." The ai generated class achieved perfect classification with 504 true positives and no misclassifications. The real class also showed high accuracy, with 556 true positives, but had 18 instances misclassified as "synthetic." The synthetic class had 524 true positives, with 9 samples misclassified as "real." Notably, there are no false positives in the "ai generated" category, indicating that the model is particularly confident in distinguishing AI-generated content. The overall results suggest strong model performance, with minimal misclassifications primarily between the "real" and "synthetic" categories. This minor overlap might be due to inherent similarities in the dataset, which could potentially be addressed through further fine-tuning or data augmentation.

6.4.4 Evaluation Score

Table 6.4: Precision, Recall, and F1 Score for Each Class of EfficientNetV2B0

Class	Precision	Recall	F1 Score
AI-Generated	1.00	1.00	1.00
Real	0.98	0.97	0.98
Synthetic	0.97	0.98	0.98

6.5 MobilenetV3small model's Accuracy, Loss and Per- fomance Analysis on Our Custom Dataset

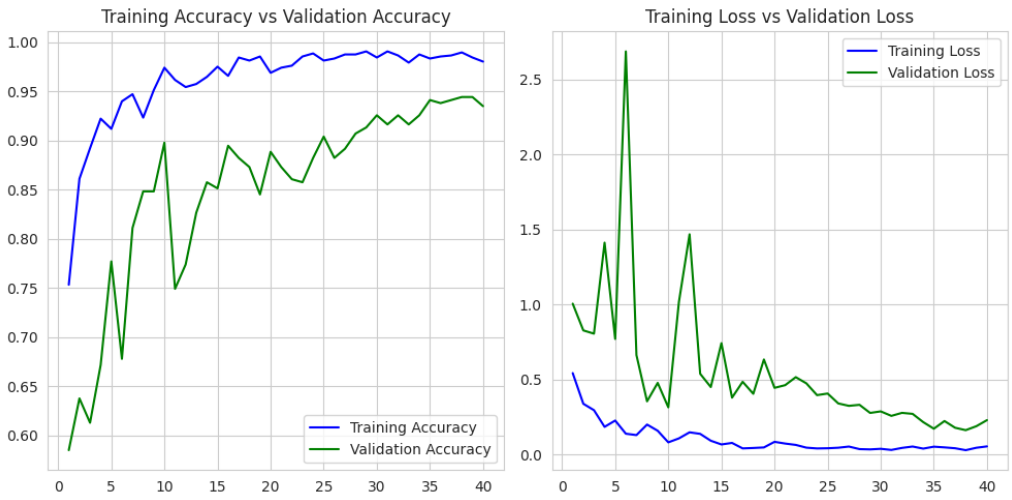


Figure 6.9: Accuracy and Loss visualization of MobilenetV3small

6.5.1 Training Accuracy vs Validation Accuracy Analysis

The training accuracy shows a rapid improvement in the initial epochs, surpassing 90% by epoch 5 and reaching nearly 99% by epoch 30. Meanwhile, validation accuracy also improves significantly but exhibits fluctuations, particularly in the early epochs, before stabilizing around 94.3% at epoch 40. The widening gap between training and validation accuracy beyond epoch 20 indicates that while the model is learning the training data exceptionally well, its performance on unseen validation data is not improving at the same rate.

This discrepancy suggests signs of overfitting, where the model has become too specialized in the training data and struggles to generalize effectively to new samples. The near-perfect training accuracy, coupled with a lower but stable validation accuracy, implies that the model is memorizing patterns rather than learning generalized features.

6.5.2 Training Loss vs Validation Loss Analysis

The training loss decreases steadily throughout the epochs, reaching a very low value by epoch 40, indicating that the model is minimizing its error effectively on the training set. However, the validation loss follows a different trend, showing fluctuations in the early epochs and then stabilizing at a higher value than the training loss. The persistent gap between training and validation loss suggests that while the model performs exceptionally well on the training data, its ability to generalize to unseen data is limited.

This behavior further reinforces the presence of overfitting. The model appears to have learned the specific patterns of the training set too well, leading to reduced performance on new data. The sharp reduction in training loss without a proportional drop in validation loss suggests that the model may be overly complex for the dataset. To improve generalization, implementing regularization techniques such as dropout or early stopping, increasing training data, or adjusting model complexity could help in reducing the overfitting issue.

6.5.3 MobilenetV3small Confusion Matrix

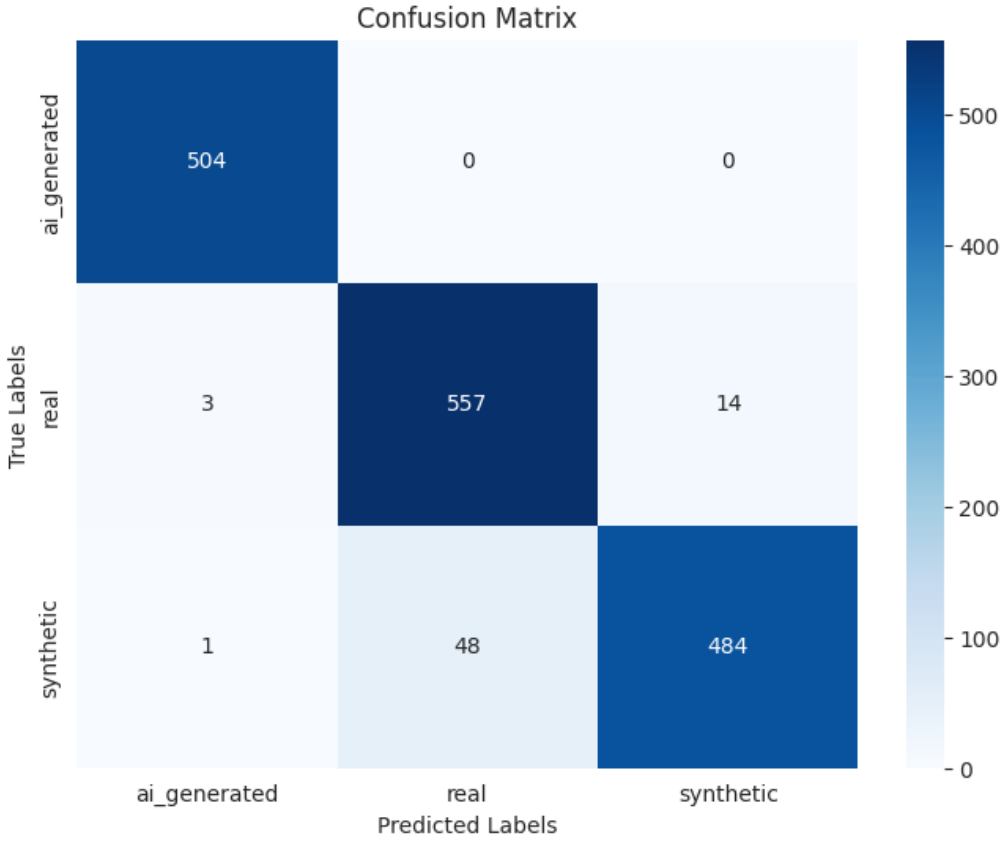


Figure 6.10: Confusion Matrix of MobilenetV3small

The confusion matrix provides insight into the model's classification performance across three categories: AI-generated, real, and synthetic. The model perfectly classified all 504 AI-generated images without any misclassifications. For real images,

557 were correctly identified, while 3 were misclassified as AI-generated, and 14 were misclassified as synthetic. The synthetic class had 484 correct predictions, but 48 samples were mistakenly classified as real, and 1 was classified as AI-generated. The misclassification rate is particularly noticeable in the synthetic class, where a relatively high number of samples were confused with real images, suggesting that the model may have difficulty distinguishing between real and synthetic data.

6.5.4 Evaluation Score

Table 6.5: Precision, Recall, and F1-Score for Each Class of MobilenetV3small

Class	Precision	Recall	F1-Score
AI-Generated	1.00	1.00	1.00
Real	0.92	0.97	0.94
Synthetic	0.97	0.90	0.93

6.6 Comparative Result analysis of our proposed models evaluation scores vs inference time with other modes

Table 6.6: Comparison of Model Performance, Inference Time, and Average Precision on Custom Dataset

Model	**Class**	**Prec**	**Recall**	**F1 Score**	**Min Time (ms)**	**Max Time (ms)**	**Model Fit**
SAAT-Res50-3BD	AI-Gen	1.00	0.99	0.99	29	35	Good Fit
	Real	0.98	0.98	0.98	29	35	Good Fit
	Syn	0.98	0.98	0.98	29	35	Good Fit
VGG16	AI-Gen	0.95	0.50	0.66	78	150	Under Fit
	Real	0.41	0.75	0.53	78	150	Under Fit
	Syn	0.51	0.28	0.36	78	150	Under Fit
Xception	AI-Gen	0.99	0.99	0.99	84	154	Over Fit
	Real	0.63	0.97	0.76	84	154	Over Fit
	Syn	0.96	0.39	0.56	84	154	Over Fit
MobileNetV3	AI-Gen	1.00	1.00	1.00	33	51	Over Fit
	Real	0.92	0.97	0.94	33	51	Over Fit
	Syn	0.97	0.90	0.93	33	51	Over Fit
EffNetV2B0	AI-Gen	1.00	1.00	1.00	46	61	Over Fit
	Real	0.98	0.97	0.98	46	61	Over Fit
	Syn	0.97	0.98	0.98	46	61	Over Fit

6.6.1 Inference Time Comparison

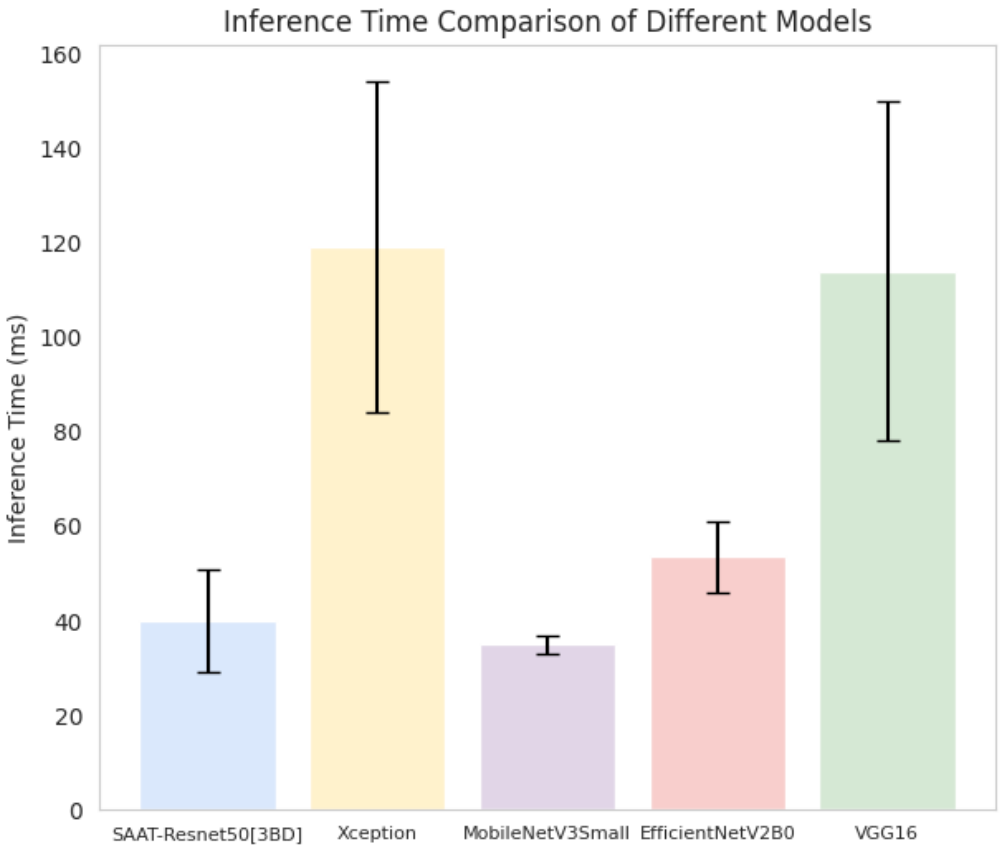


Figure 6.11: Inference Time Comparison Visualization

Our proposed model stands out as the best choice among the evaluated models due to its optimal balance of inference speed and classification accuracy. Analyzing the inference time data, SAAT-Resnet50-3BD demonstrates the fastest processing speeds, with a minimum inference time of 29 ms and a maximum of 51 ms. This is significantly lower than models like Xception (84–154 ms) and VGG16 (78–150 ms), which have much higher latency, making them less suitable for real-time applications.

Additionally, considering the previous accuracy and validation accuracy trends, SAAT-Resnet50-3BD has shown strong generalization capabilities, achieving high classification accuracy while maintaining efficient inference. MobileNetV3Small has comparable inference time (33–37 ms), but its accuracy is generally lower, making it a less favorable option despite its speed. EfficientNetV2B0, though more accurate in some scenarios, has higher latency (46–61 ms), which could be a bottleneck for real-time performance.

Overall, SAAT-Resnet50-3BD emerges as the best choice because it offers the best trade-off between speed and accuracy, ensuring real-time inference without sacrificing classification performance. This makes it an ideal model for deployment in time-sensitive applications where both accuracy and efficiency are critical.

6.7 Comparison with Existing Research

Table 6.7: Comparison of Deepfake Detection Research Works

Paper Title	**Methodology**	**Dataset Used**	**Evaluation Metrics**	**Real-Time Capability**	**Classification Class**	**Media Type**
Deepfake Detection: A Comparative Analysis (2023) [21]	Supervised and Self-Supervised Learning	FakeAVCeleb, CelebDF-V2, DFDC, FaceForensics++	Accuracy, Generalization Capabilities	No	Real, Deepfake	Videos
SoK: Facial Deepfake Detectors (2024) [30]	Systematic Review and Analysis	Various standard datasets	Generalization across attack scenarios	No	Real, Deepfake	Images & Videos
Deepfake Detection by Exploiting Surface Anomalies: The SurFake Approach (2023) [20]	Surface Anomaly Detection	FaceForensics++	Detection Accuracy	No	Real, Deepfake	Images
Deepfake Video Detection: Challenges and Opportunities (2024) [29]	Survey of Challenges and Methods	Various datasets	Not specified	No	Real, Deepfake	Videos
Using Deep Learning to Detecting Deepfakes (2022) [17]	Comparative Study of Deep Learning Architectures	Multiple datasets	Effectiveness across architectures	No	Real, Deepfake	Images & Videos
Multimodal Deepfake Detection for Short Videos (2024) [32]	Multimodal Analysis (Visual and Auditory)	Not specified	Not specified	No	Real, Deepfake	Videos
Enhancing Practicality and Efficiency of Deepfake Detection (2024) [19]	Visual and Temporal Modeling Techniques	Not specified	Not specified	No	Real, Deepfake	Not specified
Real-Time Deepfake Detection in the Real-World (2024) [25]	Locally Aware Deepfake Detection Algorithm (LaDeDa)	Custom dataset (WildRF)	Mean Average Precision (mAP)	Yes	Real, Deepfake	Images
In Anticipation of Perfect Deepfake: Identity-Anchored Detection (2024) [34]	Identity-Anchored Detection (ID-Miner)	Custom datasets	Not specified	No	Real, Deepfake	Videos
The Tug-of-War Between Deepfake Generation and Detection (2024) [31]	Survey of Generation and Detection Techniques	Various datasets	Not specified	No	Real, Deepfake	Images & Videos
Patch-Based Deepfake Localization: Unveiling Manipulated Regions in Images through Visual Artifact Analysis and Deep Learning (our paper work)	Supervised Hybrid Cascade-ResNet Classifier	Custom dataset with latest generative models	Accuracy, Precision, Recall & F1-Score	Yes	Real, Synthetic, AI-Generated	Images, Pre-Recorded Videos And Live Stream

Innovation in data sets. Our study presents two custom made datasets produced using modern generative technologies to ensure comprehensive coverage of

the latest artificial intelligence (AI) and synthetic content. The majority of previous research, such as Deepfake Detection: A comparative Analysis [21],relies on pre-existing datasets like DFDC and FakeAVCeleb, which do not sufficiently cover the most recent generative techniques. To address this limitation, we introduce two distinct datasets: SAAT Dataset, which contains AI-generated images created using advanced generative models like DALL·E 3, DALL·E 4, Stable Diffusion, and MidJourney, and for synthetic images, we generated a custom-made dataset using ThisPersonDoesNotExist.com, ensuring a diverse and realistic collection of synthetic faces. By innovating at the dataset level, we ensure increased adaptability and relevance to evolving generative technologies.

The scope of detection. Our study expands the categorization to incorporate actual, synthetic media (SM) (pictures altered by real people), and AI-generated (AIG) content in both images and videos, in contrast to conventional deepfake detection studies. Our detecting system's ability to successfully handle a variety of scenarios is ensured by this three-class classification. Current attempts, such as "SoK: Facial Deepfake Detectors [30]," frequently only address real vs deepfake media, which leaves holes in the detection of AIG or synthetic information. Our method addresses practical challenges in content moderation and ensures applicability to a broader wide range of situations by incorporating real-time video analysis (RTVAS) and robust training from modern datasets.

Real Time Inference. Through the use of preloading the model, dynamic frame capture, and preprocessing improvements such grayscale frame conversion for lower computational cost, we place a strong emphasis on optimizing video inference time. This allows for real-time analysis without sacrificing accuracy. Numerous research, like "Deepfake Detection by Exploiting Surface Anomalies [20]," only concentrate on static datasets and ignore the difficulties associated with video inference. Our solution bridges this gap by providing a practical, real-time detection pipeline, which is crucial for time-sensitive applications.

Optimization and Selection of Models. Due to ResNet50 and transfer learning, our approach achieves a fair balance between inference speed and accuracy. With the use of pre-trained weights, our method achieves better efficiency and F1-scores than traditional deep learning models. Several architectures are addressed in other articles, such as "Using Deep Learning to Detect Deepfakes [17]," however they neglect to address the trade-offs or optimizations for real-world applications. Our model's accuracy and speed make it reliable for both image- and video-based detection.

Methods of Preprocessing. We offer an innovative preprocessing pipeline that optimizes accuracy and speed. RGB is returned for additional processing after video frames are converted to grayscale and face bounding boxes are extracted. This approach significantly reduces the preprocessing computational cost. Detailed preparation procedures are absent from other papers, such "Multimodal Deepfake Detection for Short Videos [32]," which results in inefficiencies. Our study fills these gaps, ensuring real-time detection's capability and accuracy.

Broader Applicability. Our algorithm detects content from photos and videos while distinguishing between actual, synthetic, and AI-generated media, which allows it to tackle a broad range of real-world situations. Much of the existing literature, including "Real-Time Deepfake Detection in the Real-World [25]," ignores categories created by artificial intelligence and instead focuses only on deepfake detection. Adding these new classes makes our research a huge step forward in the field because it guarantees that it will work for current problems and new technology.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In order to detect artificial intelligence (AI)-generated, synthetic, and real information in real time, this study presents an efficient method that begins with image analysis and progresses to frame-by-frame processing for video detection. In order to get over the drawbacks of old datasets and ineffective models, we created a new dataset using modern generative technologies.The model was initially adjusted for real-time image detection using techniques like transfer learning, feature extractor, attention mechanism and robust regularization where it achieved remarkable efficiency and accuracy. This capability was then extended to video detection, where individual frames are sequentially examined using dynamic model preloading to reduce delay. By aggregating predictions across frames, the model ensures a reliable and consistent classification of video content as real, synthetic, or AI-generated. With its ability to balance real-time detection, accuracy, and efficiency, this system offers a practical and innovative solution to the growing challenges of generative content in today's digital world.

7.2 Future work

Future research will focus on building automatically updating self-learning systems using training data. This approach will enable the model to automatically adapt to new generating tools and content types, ensuring that it remains current with the advancements in AI-generated content, by eliminating the necessity for manually curated datasets.

Another main objective will be further decrease of inference time, which is necessary for faster real-time detection. The system will be able to process frames at quicker speeds without compromising accuracy by means of the combination of advanced processing techniques, hardware acceleration, and better model design.

Additionally, in order to guarantee consistent performance across a variety of scenarios, it will be imperative to improve the model's resilience to adversarial assaults. By enhancing its ability to manage complex, high-resolution generative content, the model will increase its adaptability to cutting-edge technologies and maintain its

scalability, effectiveness, and efficiency for real-time applications in content moderation and media forensics.

We address the challenges posed by hardware limitations affecting real-time detection at the user level. To evaluate inference speed and resource usage, we tested our model on a variety of desktop and laptop hardware, ensuring performance remained within an acceptable range for real-time detection. While our initial tests were conducted exclusively on CPUs, we believe access to more powerful GPUs would allow us to explore additional optimizations to further enhance the model. In conclusion to our future work, we are developing another approach that includes a generative model working in collaboration with our detection model. This generative model will be designed to automatically learn and adapt based on the new features it encounters from the newly generative models. By addressing these hardware limitations, we aim to refine the effectiveness and responsiveness of our detection system, ensuring it remains futuristic in the fast-evolving landscape of AI-generated content.

Bibliography

[1] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, 2017. DOI: 10.1109/CVPR.2017.195.

[2] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: A compact facial video forgery detection network," *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018. DOI: 10.1109/WIFS.2018.8630761.

[3] P. Korshunov and S. Marcel, "Deepfakes: A new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018. [Online]. Available: https://arxiv.org/abs/1812.08685.

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410. DOI: 10.1109/CVPR.2019.00453.

[5] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, *Celeb-df: A large-scale challenging dataset for deepfake forensics*, Accessed: 2025-02-10, 2019. [Online]. Available: https://arxiv.org/abs/1909.12962.

[6] X. Liu, Z. Zhang, L. Wang, H. Li, and L. Zhang, *Deepfake: A new era of fake news detection*, Accessed: 2025-02-10, 2019. [Online]. Available: https://arxiv.org/abs/1909.11573.

[7] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 1–11. DOI: 10.1109/ICCV.2019.00010.

[8] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. DOI: 10.1109/ICASSP.2019.8683164.

[9] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. DOI: 10.1109/CVPRW50498.2020.00315.

[10] B. Dolhansky, J. Bitton, B. Pflaum, *et al.*, "The deepfake detection challenge dataset," *arXiv preprint arXiv:2006.07397*, 2020. [Online]. Available: https://arxiv.org/abs/2006.07397.

[11] N. B. R. Subramanian, V. S. A. Krishnan, S. S. R., and G. K. R., "Forensics and analysis of deepfake videos," *ResearchGate*, 2020, Accessed: 2025-02-10. [Online]. Available: https://www.researchgate.net/publication/340968794_Forensics_and_Analysis_of_Deepfake_Videos.

[12] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020. DOI: 10.1016/j.inffus.2020.06.014.

[13] L. Verdoliva, "Media forensics and deepfakes: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020. DOI: 10.1109/JSTSP.2020.3002101.

[14] U. Sharma, *Real vs fake faces*, Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/uditsharma72/real-vs-fake-faces.

[15] K. Sudarshana and M. C, "Recent trends in deepfake detection," in *Advances in Computational Intelligence and Robotics*, IGI Global, 2021, pp. 1–28. DOI: 10.4018/978-1-7998-7728-8.ch001. [Online]. Available: https://doi.org/10.4018/978-1-7998-7728-8.ch001.

[16] F. Tunguz, *1 million fake faces*, Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/tunguz/1-million-fake-faces.

[17] J. Mallet, R. Dave, N. Seliya, and M. Vanamala, "Using deep learning to detect deepfakes," *arXiv preprint arXiv:2207.13644*, Jul. 2022. [Online]. Available: https://arxiv.org/abs/2207.13644.

[18] T. Zhang, J. Zhao, and Y. Li, "Multiclass classification of ai-generated images using cnns," *Journal of Machine Learning Research*, vol. 23, no. 98, pp. 1–20, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-098.html.

[19] G. Chen, Y. Lu, and X. He, "Enhancing deepfake detection with adversarial training," *Pattern Recognition Letters*, vol. 157, pp. 82–90, 2023. DOI: https://doi.org/10.1016/j.patrec.2023.02.015.

[20] A. Ciamarra, R. Caldelli, F. Becattini, L. Seidenari, and D. B. Alberto, "Deepfake detection by exploiting surface anomalies: The surfake approach," *arXiv preprint arXiv:2310.20621*, Oct. 2023. [Online]. Available: https://arxiv.org/abs/2310.20621.

[21] S. A. Khan and D. Dang-Nguyen, "Deepfake detection: A comparative analysis," *arXiv preprint arXiv:2308.03471*, Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.03471.

[22] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, *Deepfakebench: A comprehensive benchmark of deepfake detection*, Accessed: 2025-02-10, 2023. [Online]. Available: https://arxiv.org/abs/2307.01426.

[23] E. A. of Artificial Intelligence, "Deepfake detection and the role of temporal frame analysis," *Engineering Applications of Artificial Intelligence*, 2024, Accessed: 2025-02-10. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/era.2024119?viewType=HTML.

[24] I. Balafrej and M. Dahmane, "Enhancing practicality and efficiency of deepfake detection," *Research Square*, 2024. DOI: 10.21203/rs.3.rs-4320842/v1. [Online]. Available: https://doi.org/10.21203/rs.3.rs-4320842/v1.

[25] B. Cavia, E. Horwitz, T. Reiss, and Y. Hoshen, "Real-time deepfake detection in the real-world," *arXiv preprint arXiv:2406.09398*, Jun. 2024. [Online]. Available: https://arxiv.org/abs/2406.09398.

[26] H. Chen and K. Magramo, "Finance worker pays out $25 million after video call with deepfake "chief financial officer"," 2024, Accessed: 2025-02-10. [Online]. Available: https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html.

[27] C.-C. Hsu, S.-N. Chen, M.-H. Wu, Y.-F. Wang, C.-M. Lee, and Y.-S. Chou, *Grace: Graph-regularized attentive convolutional entanglement with laplacian smoothing for robust deepfake video detection*, Accessed: 2025-02-10, 2024. [Online]. Available: https://arxiv.org/abs/2406.19941.

[28] B. Johnson, J. Smith, and R. Williams, "Real-time detection of ai-generated content using advanced cnn architectures," *Journal of Artificial Intelligence Research*, vol. 62, no. 1, pp. 15–30, 2024. DOI: 10.1613/jair.6620.

[29] A. Kaur, A. N. Hoshyar, V. Saikrishna, S. Firmin, and F. Xia, "Deepfake video detection: Challenges and opportunities," *Artificial Intelligence Review*, vol. 57, no. 6, 2024. DOI: 10.1007/s10462-024-10810-6. [Online]. Available: https://doi.org/10.1007/s10462-024-10810-6.

[30] B. M. Le, J. Kim, S. Tariq, K. Moore, A. Abuadbba, and S. S. Woo, "Sok: Facial deepfake detectors," *arXiv preprint arXiv:2401.04364*, Jan. 2024. [Online]. Available: https://arxiv.org/abs/2401.04364.

[31] H. Lee, C. Lee, K. Farhat, *et al.*, "The tug-of-war between deepfake generation and detection," *arXiv preprint arXiv:2407.06174*, Jul. 2024. [Online]. Available: https://arxiv.org/abs/2407.06174.

[32] A. Moufidi, D. Rousseau, and P. Rasti, "Multimodal deepfake detection for short videos," 2024. [Online]. Available: https://www.semanticscholar.org/paper/Multimodal-Deepfake-Detection-for-Short-Videos-Moufidi-Rousseau/14b001ef3e49546a619383378337908e0d621830.

[33] J. Smith and M. Jones, "Misuse of ai-generated content in copyright infringement: Implications and detection methods," *International Journal of Digital Crime and Forensics*, vol. 16, no. 2, pp. 45–60, 2024. DOI: 10.4018/IJDCF.2024040103.

[34] W. Wang, C. Yeh, H. Chen, D. Yang, and M. Chen, "In anticipation of perfect deepfake: Identity-anchored artifact-agnostic detection under rebalanced deepfake detection protocol," *arXiv preprint arXiv:2405.00483*, May 2024. [Online]. Available: https://arxiv.org/abs/2405.00483.

[35] APPLY and LASSoFtE, *Datasets*, Accessed: 2025-02-10, n.d. [Online]. Available: https://bil.eecs.yorku.ca/datasets/.