UNIVERSITÉ MOULAY ISMAÏL

FACULTÉ DES SCIENCES

Master en Intelligence artificielle et Analyse des données

End-of-studies project for Master's degree

Major: Artificial Intelligence and Data Analysis

Presented By	Supervised by
MHASNI Khalid	**Pr. BEKRI Ali**

On the topic:

"AI vs. AI": Using Advanced Algorithms in ML and DL to Identify AI-Generated Faces

Defended on Wednesday 11th of September 2024 before the jury:

Pr. BEKRI Ali	(Supervisor)	Professor at the Faculty of Sciences, Meknès
Pr. OUBELKACEM Ali	(President)	Professor at the Faculty of Sciences, Meknès
Pr. BOURRAY Hamid		Professor at the Faculty of Sciences, Meknès
Pr. SABBANE Mohammed		Professor at the Faculty of Sciences, Meknès

Faculté des Sciences, B.P. 11201 Zitoune Meknès, Maroc.

Tel: +212 5 35 53 73 21, Fax: +212 5 35 53 68 08, http://www.fs-umi.ac.ma.

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1 INTRODUCTION AND PROJECT OVERVIEW

1.1 Background and Context

The rapid advancement of artificial intelligence (AI) has transformed several industries, including healthcare, finance, entertainment, and security. A particular area of AI that has drawn significant attention is the ability to generate synthetic media, especially human faces, using techniques such as Generative Adversarial Networks (GANs). According to Karras et al., the rise of powerful architectures such as StyleGAN has enabled the creation of photorealistic human faces that are virtually indistinguishable from real ones [1]. This growing capability presents not only creative opportunities but also challenges, such as the proliferation of deepfakes, which are often used for malicious purposes, including identity theft and disinformation.

AI-generated faces have already been integrated into various applications, from creating synthetic characters in movies to assisting in generating datasets for machine learning tasks. However, the growing use of these fake faces in deceptive practices highlights the pressing need for robust detection mechanisms.

1.2 Motivation for the Project

The misuse of AI-generated content, particularly deepfakes, has emerged as a significant concern in the realms of digital security and privacy. In recent years, deepfakes have gained widespread attention, with a substantial number of these manipulated videos being created and shared online. These synthetic videos have sparked serious public alarm due to their potential to incite political unrest, extort individuals, or even simulate acts of terrorism [2].

Research by Korshunov and Marcel highlights that even highly skilled human reviewers often struggle to differentiate between real and AI-generated faces, emphasizing the urgent need for automated detection methods. This underscores the necessity for robust detection algorithms capable of keeping pace with the rapidly advancing technology behind AI-generated faces [3].

1.3 Research Problem

GANs have achieved remarkable success in producing photorealistic images, particularly human faces, that are often indistinguishable from real ones. However, this growing realism poses significant challenges for detection. In earlier generations of GAN-produced images, synthetic faces often exhibited clear artifacts—visual imperfections like unnatural textures, asymmetrical features, or pixel-level inconsistencies—that made them relatively easy to detect using traditional image analysis methods.

As GANs have evolved, particularly with advanced architectures like StyleGAN and StyleGAN2, these obvious artifacts have largely disappeared. The high levels of realism in GAN-generated faces today are a result of more sophisticated training processes that mimic the intricacies of human facial structures, textures, and expressions. As a result, the subtle differences between real and AI-generated images can now go unnoticed, even by human reviewers. This has rendered conventional detection techniques, which relied on identifying such artifacts, largely ineffective in modern contexts.

Detecting AI-synthesized images has thus become a much more complex task. According to Wang et al., modern GANs generate images with minimal visible discrepancies, making it difficult to differentiate between real and synthetic faces using traditional methods like pixel analysis or simple feature extraction [4].

This research seeks to tackle these challenges by developing innovative methods that can detect subtle differences, ensuring they remain effective and adaptable to the rapidly advancing GAN technologies.

1.4 Objectives of the Project

The primary objectives of this project are outlined as follows:

1. **To collect and preprocess a dataset of AI-generated and real human faces:** The initial phase of this project involves building a robust dataset that includes both real and AI-generated human faces. Data collection will involve obtaining images from publicly available datasets and generating synthetic faces using GANs. It is critical to ensure the dataset is balanced and diverse to minimize biases and enhance the model’s capacity to

accurately distinguish between real and AI-generated faces under various conditions, such as different lighting environments, facial expressions, and features.

2. **To develop a deep learning model for detection:** Given deep learning's proven success in image classification and object detection tasks, this project aims to design a model specifically optimized for detecting AI-generated faces. The model will be built on state-of-the-art deep learning techniques to maximize detection accuracy and robustness.
3. **To evaluate model performance using a range of metrics:** Ensuring the model’s effectiveness across different conditions is key, and this will be assessed through several performance metrics, such as accuracy, precision, recall, and F1-score. In addition, confusion matrices will provide further insights into misclassifications and evaluate how well the model generalizes to new, unseen data.

1.5 Structure of the Report

The rest of this report is organized as follows:

- **Chapter 2: AI Foundations for Detecting Generated Faces**: This chapter delves into the key AI technologies relevant to detecting AI-generated faces, including computer vision, deep learning, and GANs. These foundational concepts establish the context for understanding the challenges and advancements in detecting AI-generated images.
- **Chapter 3: Overview of Advanced AI Image Detection Techniques**: This chapter offers a comprehensive review of cutting-edge methods for detecting AI-generated images, focusing on techniques such as spatial, frequency, semantic, and neuron activation-based approaches. It evaluates the strengths and weaknesses of these methods and highlights areas where further development is needed.
- **Chapter 4: Proposed Enhanced Detection Framework**: This chapter presents the proposed detection framework, detailing its architecture, feature extraction processes, and classifier design. It also covers the experimental setup, including dataset preparation, model training, and evaluation metrics. The performance of the proposed model is compared against other state-of-the-art methods, with an analysis based on metrics like accuracy, precision, recall, F1-score, and AUC.

- **Chapter 5: Conclusion and Future Work**: The final chapter summarizes the study's key findings, particularly the effectiveness of the enhanced detection framework in identifying AI-generated faces. It also suggests potential future research directions and improvements to the detection model.

CHAPTER 2 AI FOUNDATIONS FOR DETECTING GENERATED FACES

2.1 Introduction

In this chapter, we explore the foundational concepts in artificial intelligence that are critical to understanding the challenge of detecting AI-generated faces. We begin with **Computer Vision**, a field that empowers machines to interpret and act on visual data, much like humans do. This is followed by an examination of **Deep Learning**, a powerful subset of machine learning that has driven advancements in numerous AI applications, including the generation and recognition of synthetic images. Finally, we delve into **Generative Adversarial Networks (GANs)**, a groundbreaking technology that has transformed image synthesis, enabling the creation of highly realistic human faces. These concepts set the stage for understanding the complexities and challenges inherent in distinguishing between real and AI-generated faces.

2.2 Computer Vision

Computer Vision is a branch of artificial intelligence that enables computers to interpret and understand the visual world in a manner similar to the human visual system. It equips machines with the ability to see, understand, and make observations based on visual inputs, such as digital images, videos, and other visual data. The fundamental goal of computer vision is to automate tasks that the human visual system can do, such as recognizing objects, analyzing visual content, and making decisions based on visual information.

In computer vision, digital images and videos serve as the primary sources of data. These images are processed by computer vision algorithms, which are designed to identify patterns, detect objects, and recognize features within the visual data. This capability is crucial for various applications, including autonomous vehicles, where computer vision is used to detect and classify objects, such as pedestrians, vehicles, and traffic signs, ensuring safe and efficient navigation. Deep learning has revolutionized computer vision by providing the computational power to automatically learn features from raw image data, which has led to breakthroughs in tasks such as object detection, image segmentation, and facial recognition. With the advent of more sophisticated models like Convolutional Neural Networks (CNNs) and Vision

Transformers, computer vision continues to evolve, enabling the creation and detection of highly realistic synthetic images [5].

2.3 Deep Learning

Deep Learning is a subset of machine learning that mimics the human brain's ability to recognize patterns and make decisions. It is an essential component of artificial intelligence, designed to simulate the way humans learn and process information through hierarchical neural networks. These networks consist of multiple layers that progressively extract higher-level features from raw input data, enabling machines to perform complex tasks such as image recognition, natural language processing, video analysis and even generating new content [6]. Recent advancements in deep learning have fueled significant progress in various AI applications, including the generation and recognition of synthetic images.

2.4 Generative Adversarial Networks

GANs have become a cornerstone in the field of artificial intelligence, particularly for generating synthetic data such as images, videos, and even text. Introduced by Ian Goodfellow and colleagues in 2014, GANs are composed of two neural networks: the **generator** and the **discriminator**. The generator creates synthetic data, while the discriminator evaluates the authenticity of this data by comparing it with real data. The two networks are trained in tandem, with the generator improving its output to deceive the discriminator, which, in turn, becomes more adept at distinguishing real from synthetic data. This adversarial process drives the generation of increasingly realistic synthetic content, making GANs one of the most powerful tools for data generation in AI [7].

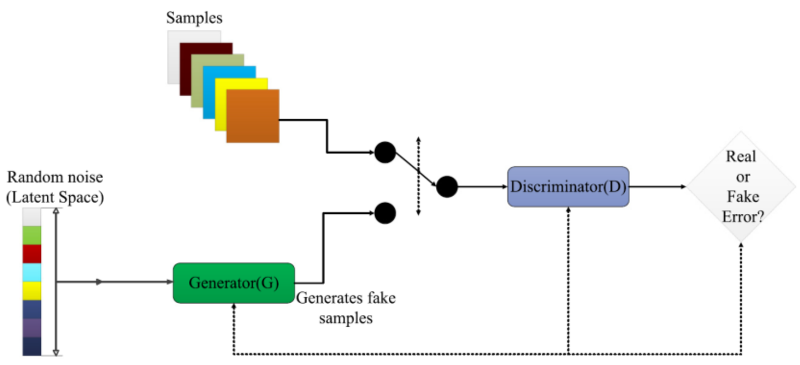


Figure 2-1 How Generative Adversarial Networks works [8]

Within the GANs framework, there are two primary types:

- **Conditional GANs (cGANs),** such as CycleGAN, StarGAN, and GauGAN, are designed to produce data that is conditioned on additional information, such as labels or specific attributes [9]. This conditioning allows the model to generate outputs that meet particular criteria or belong to a specific category. For example, in a cGAN trained on a labeled dataset of images, the model can generate images of a particular class (e.g., cats, dogs) based on the provided label as a condition. The conditional input guides both the generator, which creates the images, and the discriminator, which distinguishes between real and generated images. This makes cGANs particularly powerful in tasks requiring targeted generation.
- **Unconditional GANs,** such as ProGAN, StyleGAN, and BigGAN, generate data purely from random noise without any external conditions [9]. This type of GAN learns to approximate the overall data distribution from the training dataset, producing realistic outputs that broadly reflect the diversity of the training data. The simplicity of unconditional GANs makes them effective for generating a wide range of outputs, but they lack control over specific attributes in the generated content, which limits their applicability in tasks requiring precision.

GANs have found applications far beyond just image generation. They are extensively used in video processing, where they assist in tasks such as video synthesis, style transfer, and future frame prediction. In the field of text generation, GANs are employed to create realistic text sequences that can mimic human writing styles. Additionally, in the medical field, GANs are applied in various areas such as DNA generation, drug development and medical imaging for dental restorations. These applications demonstrate the versatility and impact of GANs across different domains [7].

2.5 Problem Statement

The rapid development of GANs has enabled the creation of highly realistic human faces that are often indistinguishable from genuine ones. While this technology offers transformative potential in areas such as content creation, virtual reality, gaming, and medical imaging, it also presents significant challenges in differentiating AI-generated faces from real ones. GANs empower artists and creators to produce lifelike visuals and characters, democratizing access to high-quality content, and accelerating advancements in medical diagnostics through synthetic training data.

However, alongside these benefits, there is a growing concern over the misuse of GANs, particularly in the production of deepfakes and fraudulent digital identities. This misuse has resulted in serious consequences, including identity theft, online fraud, and a decline in trust in digital media.

The primary challenge addressed in this research is the development of reliable methods for detecting and classifying human faces as either AI-generated or real. The subtle differences between these images, often imperceptible to the human eye, make this task particularly challenging. To explore this issue, we conducted a survey on AI-generated face detection, where 250 participants were presented with 20 images, equally comprising real and AI-generated faces from our dataset. The participants were asked to identify the authenticity of each image, with an average score of only 7 out of 20 correct responses.

This outcome highlights the inherent difficulty humans face in accurately distinguishing between real and AI-generated faces, further emphasizing the urgent need for reliable automated detection solutions.

2.6 Conclusion

This chapter has laid the groundwork for understanding the key AI technologies—Computer Vision, Deep Learning, and GANs—that are central to both generating and detecting synthetic faces. The exploration of these fields highlights not only the remarkable advancements that have been achieved but also the complex challenges they present, particularly in maintaining the integrity and security of digital media. As we move forward, the focus will shift to examining the current state of detection techniques for AI-generated images and assessing their effectiveness.

CHAPTER 3 OVERVIEW OF ADVANCED AI IMAGE DETECTION TECHNIQUES

3.1 Introduction

The proliferation of AI-generated images has brought significant advancements in creative content generation, but it has also introduced substantial challenges, particularly in distinguishing real images from synthetic ones. This chapter reviews various state-of-the-art techniques developed to detect AI-generated images.

We selected a set of representative methods in AI-generated image detection to serve as baselines for state-of-the-art evaluations. These include *spatial artifact-based* methods like **CNNDetect** [4], which identify anomalies in spatial data through a data-driven approach. *Frequency-based* methods, such as **FreqDetect** [10], analyze discrepancies in the frequency domain to reveal artifacts introduced by generative models. *Semantic-based* approaches like **UnivFD** [11] leverage pretrained vision-language models to detect inconsistencies in feature space. *Neuron activation-based* methods, such as **FakeSpotter** [12], monitor neuron coverage behaviors in deep face recognition systems to detect subtle differences between real and AI-generated faces. *Global-local fusion* techniques, exemplified by **Fusing** [13], integrate global spatial information with localized features for improved detection accuracy. Finally, *texture-based methods* like **PatchCraft** [14] focus on identifying the challenges generative models face in replicating complex textures, using inter-pixel correlation as a key indicator of synthetic content.

By exploring these methods, we aim to understand the current capabilities and limitations in detecting AI-synthesized content, especially AI-generated faces, and to identify areas where further research is needed to enhance detection accuracy and robustness.

3.2 CNNDetect

In their 2020 study **CNN-Generated Images are Surprisingly Easy to Spot... for Now,** researchers Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros investigate the potential for a universal detector capable of distinguishing real images from those generated by convolutional neural networks (CNNs). This exploration focuses on identifying whether current CNN-generated images share common detectable features or artifacts that a classifier, trained on images from one CNN, could leverage to generalize detection across other CNN architectures.

The methodology involves training a binary classifier using images generated by a specific model, ProGAN, and subsequently testing its ability to recognize fake images produced by various other models, including StyleGAN, BigGAN, and CycleGAN. The classifier, built on a ResNet-50 architecture pre-trained on ImageNet, is trained with a large dataset composed of real and ProGAN-generated images. The authors place significant emphasis on the role of data augmentation techniques, such as Gaussian blurring and JPEG compression, in enhancing the model's ability to generalize across different CNN-generated images.

To validate their approach, the authors created the ForenSynths dataset, which includes images from 11 distinct CNN models, each with varying architectures, datasets, and tasks. The models tested encompass both unconditional GANs like ProGAN and StyleGAN, as well as conditional GANs like CycleGAN and StarGAN.

The findings are noteworthy: the classifier, trained on ProGAN images, exhibited a remarkable capacity to generalize across various CNN models, achieving high average precision scores, especially when data augmentation was applied. For instance, in a practical scenario using real and fake faces scraped from the website whichfaceisreal.com, the model demonstrated a robust performance, achieving an accuracy of 83.6% and an average precision (AP) of 93.2% when images were directly center-cropped to 224 pixels. Although resizing images before testing led to a slight drop in accuracy to 74.9%, the results still surpassed random chance [4]. However, when evaluated on StyleGAN2-generated images, the model achieved only 68.0% accuracy, as noted in the comparison by Manos Schinas and Symeon Papadopoulos in their work *SIDBench: A Python Framework for Reliably Assessing Synthetic Image Detection Methods*, highlighting challenges in detecting more sophisticated GAN-generated images [9].

3.3 FreqDetect

In the 2020 paper titled **Leveraging Frequency Analysis for Deep Fake Image Recognition**, Joel Frank, Thorsten Eisenhofer, and their colleagues present a novel approach that utilizes frequency domain analysis to detect deep fake images. They argue that while deep fake images, particularly those created by GANs, are challenging to distinguish from real images in the spatial domain, they often exhibit discernible artifacts in the frequency domain. These artifacts, associated with the upsampling processes inherent in GAN architectures, introduce consistent patterns that can be leveraged for detection.

The method centers on transforming images from the spatial domain to the frequency domain using the Discrete Cosine Transform (DCT). By analyzing the frequency spectra of real versus GAN-generated images, the researchers identified significant anomalies in the high-frequency components of GAN-generated images—anomalies absent in real images. These frequency-based inconsistencies are prominent across different GAN architectures, including BigGAN, ProGAN, and StyleGAN. The authors demonstrate that even simple classifiers trained on these frequency domain features can achieve high accuracy in distinguishing real from fake images. As depicted in Figure 3.1, a detailed side-by-side comparison of real and generated faces in both the image and frequency domains reveals that generated images, compared to real ones, diverge most in the higher frequencies, where real images typically exhibit minimal energy.

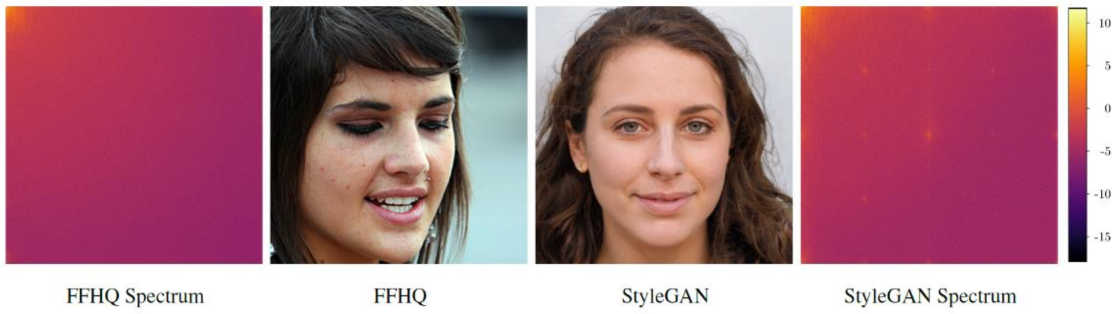


Figure 3-1 A comparison of real and generated faces in image and frequency domain.

The experiments conducted utilized datasets comprising images generated by several GANs like BigGAN, ProGAN, StyleGAN, and SN-DCGAN, with training conducted on the Stanford dogs and Large-scale Scene UNderstanding Challenge (LSUN) bedrooms datasets. The results were compelling: classifiers trained on frequency domain data significantly outperformed those trained on pixel data. For example, when applied to data samples generated by StyleGAN using ridge regression, the classifier achieved a classification accuracy of 98.24% with frequency domain data, compared to 74.77 % with pixel data [10]. Yet, when evaluated on StyleGAN2-generated images, this frequency-based technique achieved only 72.3% accuracy, according to a comparison by Manos Schinas and Symeon Papadopoulos in *SIDBench: A Python Framework for Reliably Assessing Synthetic Image Detection Methods* [9]. This outcome underscores the limitations of frequency analysis when confronted with more advanced GAN models like StyleGAN2, emphasizing the need for more sophisticated detection strategies.

3.4 UnivFD

In their 2023 publication **Towards Universal Fake Image Detectors That Generalize Across Generative Models,** researchers Utkarsh Ojha, Yuheng Li, and Yong Jae Lee introduce an innovative method for detecting AI-generated images across a broad spectrum of generative models. They underscore the shortcomings of traditional deep learning models, which are typically trained to detect specific types of fake images and often struggle to generalize to images produced by unseen models, particularly those from diverse generative families, such as diffusion models.

The distinguishing feature of their approach is the use of a feature space not explicitly trained for the real-vs-fake classification task. Instead, they utilize the feature space from a large, pretrained vision-language model, specifically a variant of the Vision Transformer (ViT) known as CLIP-ViT. By leveraging this pretrained model, which was trained on a vast dataset of 400 million image-text pairs, they avoid the bias that can occur when training solely on a particular set of fake images. The researchers explore two methodologies within this framework: nearest neighbor classification and linear probing. Both approaches demonstrated exceptional generalization capabilities, particularly when detecting fake images generated by models unseen during training.

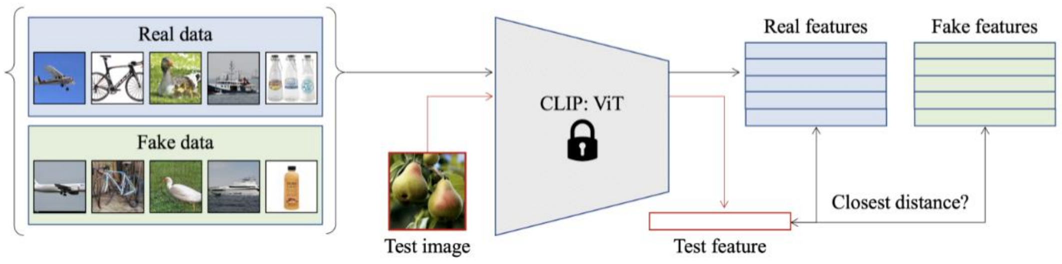


Figure 3-2 Nearest neighbors for real-vs-fake classification

The authors assessed their method using ProGAN as the primary training data source. For testing, they included a range of generative models such as BigGAN, StyleGAN, CycleGAN, StarGAN, and newer models like diffusion-based (LDM, Guided, Glide) and autoregressive models like DALL-E. The results were striking: the nearest neighbor method improved classification accuracy by over 25% and mean average precision (mAP) by more than 15 points when compared to state-of-the-art methods on unseen diffusion and autoregressive models. Specifically, the nearest neighbor approach achieved an average classification accuracy of approximately 84% on these unseen models, significantly outperforming traditional deep learning classifiers [11].

However, the performance on StyleGAN2-generated images revealed a challenge: the method achieved only 75.65% accuracy, as reported by Manos Schinas and Symeon Papadopoulos in *SIDBench: A Python Framework for Reliably Assessing Synthetic Image Detection Methods* [9]. This suggests difficulties in maintaining high detection accuracy across more sophisticated generative models like StyleGAN2, pointing to the need for further advancements in universal detection methods.

3.5 FakeSpotter

In the paper titled **FakeSpotter: A Simple Baseline for Spotting AI-Synthesized Fake Faces**, Run Wang, Lei Ma, Felix Juefei-Xu, Xiaofei Xie, Jian Wang, and Yang Liu introduce an innovative method designed to detect AI-synthesized fake faces by leveraging neuron coverage behaviors in deep face recognition (FR) systems. The authors argue that, while existing deep learning models excel in face recognition, they often fall short in distinguishing between real and AI-generated faces. To address this gap, the authors propose FakeSpotter, a technique that monitors neuron activation patterns across different layers in deep FR systems to identify subtle differences between real and fake faces.

The core idea behind FakeSpotter is that neuron activation patterns differ significantly between real and AI-synthesized faces. The approach involves capturing these patterns in deep FR systems such as VGG-Face, and then using these patterns to train a simple linear binary classifier. This classifier differentiates real from fake faces based on the number of activated neurons in each layer. The method relies on a novel neuron coverage criterion, called Mean Neuron Coverage (MNC), which calculates a fixed threshold for each layer based on the mean value of neuron outputs across large datasets of real and fake faces.

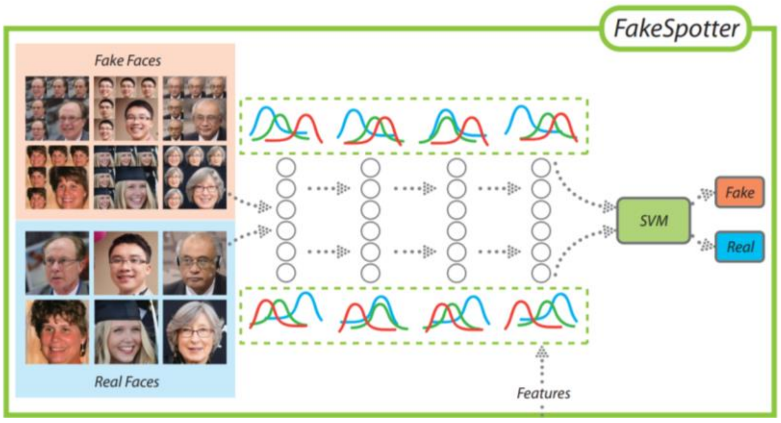


Figure 3-3 An overview of the FakeSpotter fake face detection method

The authors tested FakeSpotter using two high-resolution public face datasets, CelebA-HQ and FFHQ, as the source of real faces. For fake faces, they generated images using two GAN variants, PGGAN and StyleGAN, focusing on high-quality, 1024x1024 resolution images. The training dataset consisted of 6,000 real faces and 5,500 fake faces, while the testing dataset included 1,000 real faces and a combination of 500 fake faces generated by the two GANs.

FakeSpotter demonstrated impressive results, outperforming traditional deep learning classifiers in fake face detection. Specifically, the method achieved accuracy rates of 78.23% on VGG-Face [12].

These findings underscore the potential of neuron coverage-based approaches in enhancing the detection of AI-synthesized faces, offering a promising alternative to traditional deep learning methods that often struggle with generalization and robustness in adversarial settings.

3.6 Fusing

In their 2022 paper titled **Fusing Global and Local Features for Generalized AI-Synthesized Image Detection**, Yan Ju, Shan Jia, Lipeng Ke, Hongfei Xue, Koki Nagano, and Siwei Lyu propose a novel approach aimed at enhancing the generalization ability of AI-synthesized image detectors. The authors identify a key limitation in existing methods: while effective on images generated by familiar models, these methods often falter when dealing with images from unseen models, especially those featuring subtle local manipulations.

To tackle this issue, the authors developed a two-branch model that integrates both global and local features for improved detection accuracy. The global branch focuses on capturing the structural information of the entire image, while the local branch zeroes in on fine-grained artifacts within specific image regions. These critical regions are selected by a novel Patch Selection Module (PSM), which autonomously identifies the most informative patches based on global feature maps. These patches are then processed through the local branch to extract detailed local features. The final step involves an Attention-based Feature Fusion Module (AFFM), which merges the global and local features, allowing the model to leverage both types of information for more precise classification.

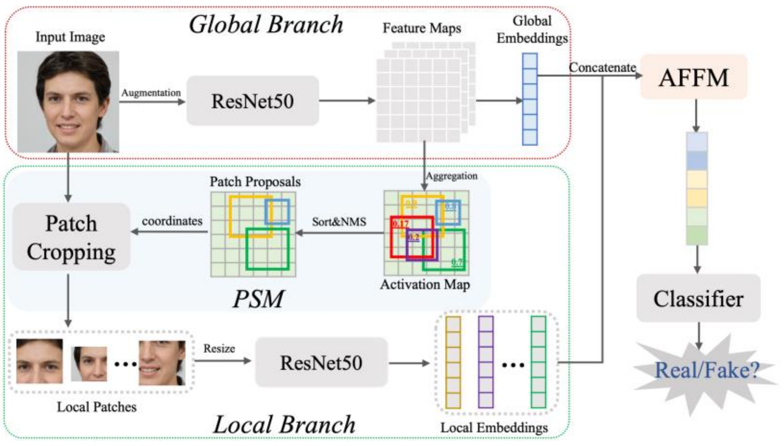


Figure 3-4 The architecture of the Fusing model

The model was trained on a substantial dataset containing 362,000 real images from the LSUN dataset and an equal number of images generated by ProGAN. To assess its generalization ability, the researchers compiled a diverse testing dataset of 128,424 images generated by 19 different models, including both conditional and unconditional GANs, as well as models utilizing perceptual loss and low-level vision techniques.

The outcomes were noteworthy. The proposed model, enhanced with the PSM for patch selection, achieved the highest mean average precision (mAP) across most model families tested, outperforming baseline methods. Specifically, the model reached a total mAP of 91.732 and a global average precision (AP) of 96.906, indicating robust generalization capabilities even on unseen models and varied datasets. Moreover, the model demonstrated resilience to common post-processing techniques like Gaussian blur and JPEG compression, further proving its effectiveness in practical scenarios [13].

Nevertheless, when the model was specifically evaluated on StyleGAN2-generated images, it attained an accuracy of 80.08%, as noted by Manos Schinas and Symeon Papadopoulos in their work *SIDBench: A Python Framework for Reliably Assessing Synthetic Image Detection Methods* [9]. This outcome highlights the ongoing challenges in detecting images from more advanced models like StyleGAN2, suggesting that further refinement in detection techniques is necessary. The research underscores the significance of combining global and local features to improve AI-synthesized image detection.

3.7 PatchCraft

In the research paper titled **PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection**, authors Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang present an innovative approach aimed at enhancing the detection of AI-generated images by focusing on the inter-pixel correlation of texture patches within images. The authors argue that AI-generated images, despite their photorealistic appearance, often fail to replicate the intricate inter-pixel relationships found in real images, especially in regions with rich textures.

The PatchCraft technique operates by dividing an image into multiple patches and analyzing the texture diversity within each patch. These patches are categorized into rich and poor texture regions, and the inter-pixel correlations within these regions are extracted to serve as a universal fingerprint for identifying AI-generated images. The technique further employs a novel Smash&Reconstruction process, illustrated in Figure 3.5, which breaks down global semantic information and emphasizes the texture patches, thereby enhancing the model's ability to detect subtle artifacts left by generative models. High-pass filters are also applied to magnify the discrepancies in inter-pixel correlations, making the technique more robust against post-processing operations such as JPEG compression, Gaussian blur, and downsampling.

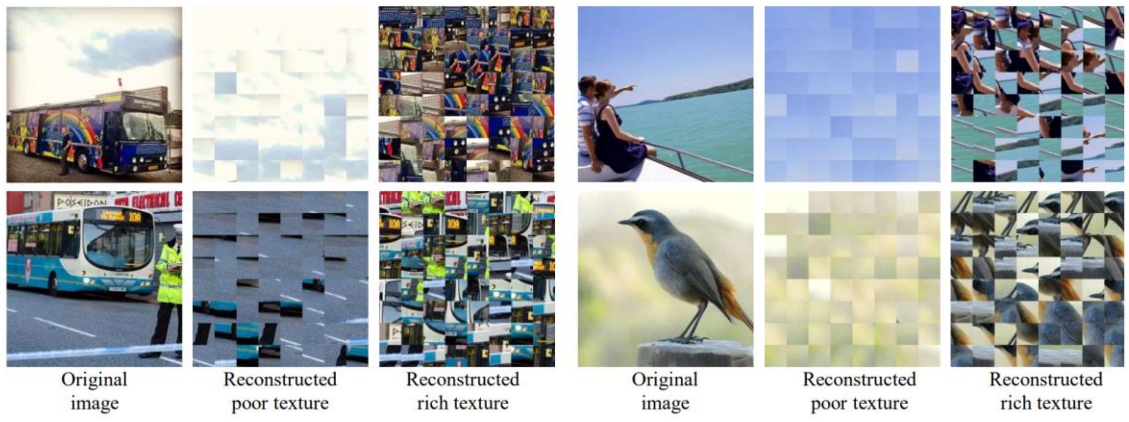

Original image
Reconstructed poor texture
Reconstructed rich texture
Original image
Reconstructed poor texture
Reconstructed rich texture

Figure 3-5 The Illustration of Smash&Reconstruction

To validate the effectiveness of PatchCraft, the authors conducted experiments using a comprehensive benchmark that includes images generated by 17 different generative models, such as ProGAN, StyleGAN, BigGAN, CycleGAN, GauGAN, and advanced models like Diffusion and DALL-E 2. The training dataset consisted of 720,000 images, equally divided between real images from the LSUN dataset and fake images generated by ProGAN.

The results indicated that PatchCraft significantly outperformed existing state-of-the-art detection methods. The technique achieved an average detection accuracy of 89.85% across all generative models tested, marking a substantial improvement over other leading methods. Notably, when applied to images generated by StyleGAN2, PatchCraft achieved an accuracy of 89.55%, demonstrating its robustness in detecting sophisticated AI-generated content. The method also showed strong resilience to common image distortions, further validating its effectiveness in real-world scenarios [14].

These findings highlight the potential of using texture analysis as a reliable feature for detecting AI-generated images. By focusing on the inherent weaknesses of generative models in replicating complex textures, PatchCraft provides a robust and generalized approach to addressing the challenges posed by the rapid advancement of AI-generated content.

3.8 Summary of Techniques, Datasets, and Evaluation Results

Technique	GANs Used for Fake Data in Training	GANs Used for Fake Data in Testing	Evaluation Results	Accuracy on StyleGAN2
CNNDetect [4]	ProGAN	StyleGAN, BigGAN, CycleGAN, etc	83.6% (Acc.), 74.9% (Acc. after resizing)	68.0% [9]
FreqDetect [10]	SN-DCGAN, ProGAN, Cramer-GAN, etc	BigGAN, ProGAN, StyleGAN, SN-DCGAN	98.24% (Acc. on StyleGAN, using frequency domain)	72.3% [9]
UnivFD [11]	ProGAN	BigGAN, StyleGAN, CycleGAN, StarGAN, Diffusion Models	84%(Acc.)	75.65% [9]
FakeSpotter [12]	PGGAN StyleGAN	PGGAN StyleGAN	78.23% (Acc. using VGG-Face)	N/A
Fusing [13]	ProGAN	19 different GANs	96.906% (AP), 91.732% (mAP)	80.08% [9]
PatchCraft [14]	ProGAN	17 different GANs	89.85% (average Acc. across all models)	89.55%

Table 3-1 Summary of Techniques, Datasets, and Evaluation Results

This table offers a clear comparison of the GANs used for generating fake data in both training and testing, alongside the overall performance on general testing datasets and the specific accuracy achieved on StyleGAN2-generated images.

3.9 Conclusion

The techniques explored in this chapter demonstrate the diverse strategies researchers have employed to tackle the detection of AI-generated images, particularly faces. From leveraging frequency analysis to focusing on neuron activation patterns and texture analysis, each method brings its strengths and challenges. While some techniques show promise in detecting a wide range of generative models, others still face difficulties, especially with more advanced models like StyleGAN2. The ongoing evolution of generative models necessitates continuous innovation in detection methodologies.

Comparing the results of these previous techniques, we have decided to enhance the *texture-based method* in our work, aiming to further improve its effectiveness and robustness in detecting AI-generated faces across one of the most advanced generative models StyleGan2.

CHAPTER 4 PROPOSED ENHANCED DETECTION FRAMEWORK

4.1 Introduction

In recent years, the proliferation of AI-generated images has presented a significant challenge in maintaining the authenticity of visual content across various platforms. Accurately distinguishing between real and AI-generated images has become increasingly important, especially as generative models continue to evolve and improve. This chapter presents the original PatchCraft [14] framework, which leverages texture-based detection techniques to address this issue. We then introduce our enhanced detection framework, developed to address the limitations of PatchCraft [14]. The improvements in our proposed framework focus on texture patterns that differentiate real images from synthetic ones, providing a more robust solution for detecting AI-generated content.

4.2 Original PatchCraft Framework

The PatchCraft [14] framework, as discussed in the state-of-the-art section of this report, is specifically designed to detect AI-generated images by leveraging the unique texture patterns that distinguish real images from synthesized ones. As illustrated in Figure 4-1, the architecture consists of three primary components: **Smash & Reconstruction**, **Feature Extraction**, and **Classification**, each playing a critical role in the overall detection process.

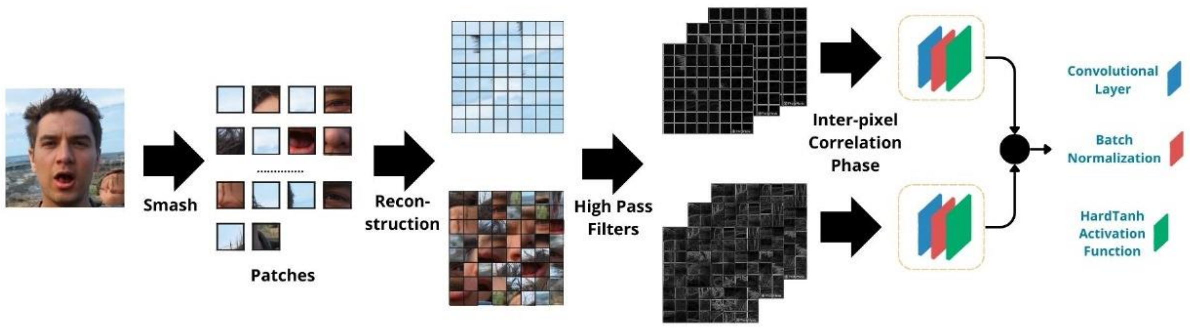


Figure 4-1 PatchCraft Architecture

4.2.1 Preprocessing in PatchCraft

The preprocessing phase (see Figure 4-1) is a crucial part of the PatchCraft [14] framework. It is designed to transform the input images in a way that accentuates the critical texture features that AI-generated images often fail to replicate accurately. This process involves several steps, each tailored to prepare the data for effective feature extraction and classification.

- **Image Division into Patches:**

The input images, which are of size **512x512 pixels,** are first divided into smaller patches. Typically, each image is split into patches of **64x64 pixels**, resulting in a total of **64 patches per image (8x8 grid)**. This division is essential for isolating different regions of the image, allowing the model to focus on localized texture patterns rather than being influenced by the global structure of the image.

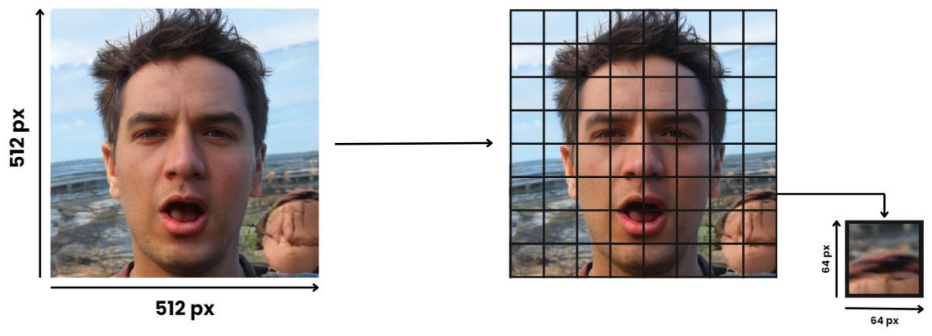


Figure 4-2 Division of images in patches

The choice of **64x64 pixel patches** strikes a balance between capturing enough detail within each patch and maintaining a manageable number of patches for the model to process. This granularity ensures that even subtle texture variations are preserved for analysis.

- **Categorization of Patches into Rich and Poor Texture Regions:**

Once the image is divided into 64 patches, each patch is analyzed to determine whether it falls into the "**rich texture**" or "**poor texture**" category. Rich texture patches are those that contain a high level of detail—such as areas with complex patterns, intricate lines, or varied color gradients. In contrast, poor texture patches have simpler, more uniform features, like smooth skin or a plain background.

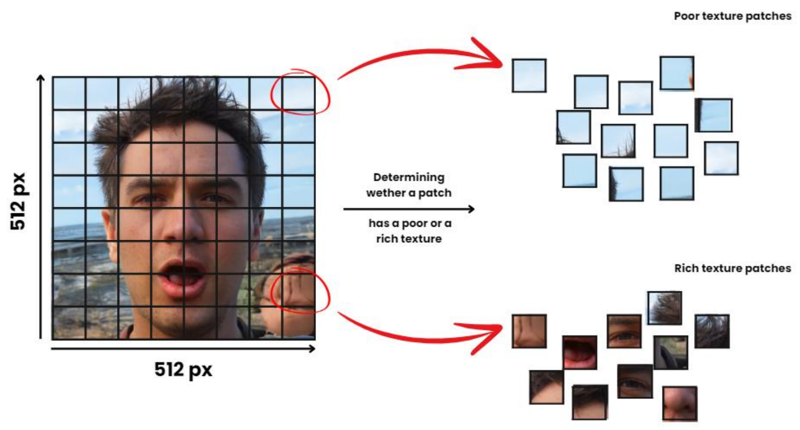


Figure 4-3 Poor and Rich Texture patches for an AI-generated Face

In a typical image, about **30-40% of the patches** may be categorized as rich texture, while the remaining **60-70%** fall into the poor texture category. This distribution is essential because it reflects the natural variability found in real images, where detailed textures are intermixed with smoother regions.

- **Smash & Reconstruction Process**

The **Smash & Reconstruction** process further refines the analysis of these patches. "Smash" refers to reducing the influence of global semantic information, such as the overall facial structure, which AI-generated models can replicate fairly well. This reduction is achieved by processing each patch independently, thus "smashing" the global coherence of the image.

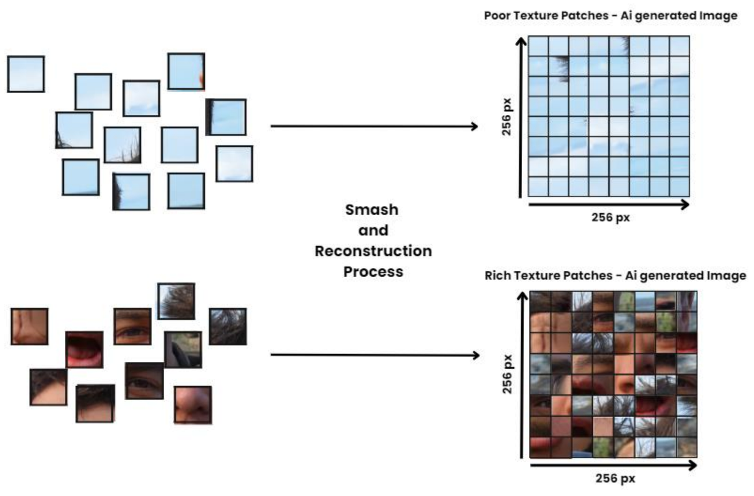

Poor Texture Patches - Ai generated Image
256 px
256 px
Smash and Reconstruction Process
Rich Texture Patches - Ai generated Image
256 px
256 px

Figure 4-4 Smash and Reconstruction Process for an AI-generated Face

After smashing, the patches are then "reconstructed" into a new image format, emphasizing the contrast between the rich and poor texture regions. This reconstruction often involves applying high-pass filters, which highlight the fine-grained details within the patches, making any inconsistencies in texture more pronounced. This process ensures that the reconstructed image is a potent representation of the original, but with a focus on texture discrepancies that are key to distinguishing real from AI-generated content.

- **Original Feature Extraction Layer using Inter-Pixel Correlations**

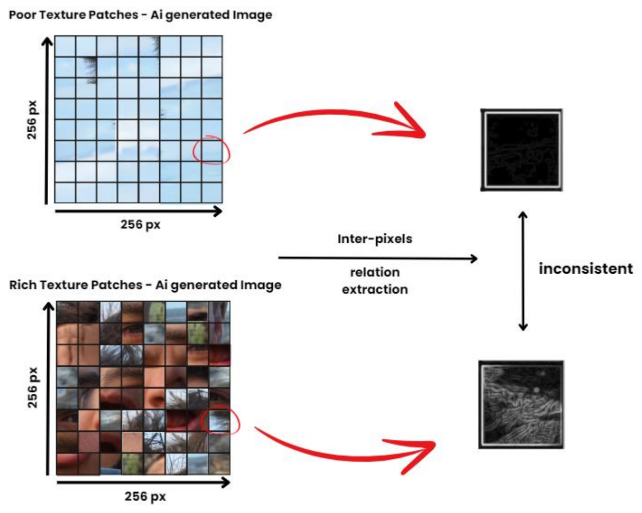

Poor Texture Patches - Ai generated Image
256 px
256 px
Inter-pixels
relation extraction
inconsistent
Rich Texture Patches - Ai generated Image
256 px
256 px

Figure 4-5 Inter-Pixels relation extraction for an AI-generated Face

The reassembled image, now focused on texture, is analyzed for inter-pixel correlations, which are critical for distinguishing real images from AI-generated ones. **Real images** exhibit natural and **consistent** correlations between neighboring pixels, while **AI-generated images** often display **unnatural** patterns, especially in areas of high texture complexity.

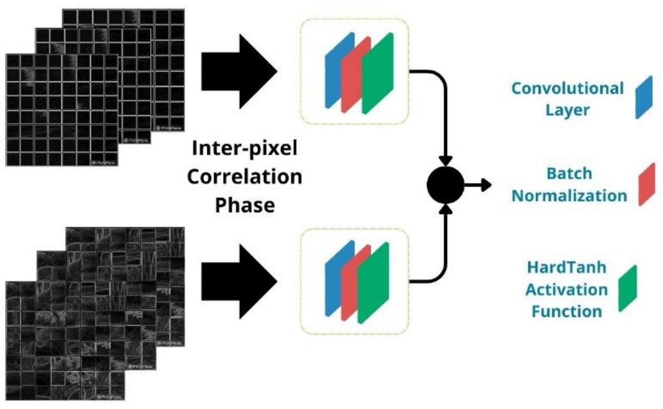


Figure 4-6 Original Feature Extraction Layer

4.2.2 PatchCraft Classifier Architecture

The classifier phase in this model architecture is the final step, responsible for converting the rich-poor texture contrast information into a binary decision—classifying the input image as either Real or AI-generated. As illustrated in Figure 4-7, this phase integrates several key components to ensure that the model accurately interprets the texture features extracted from the input images.

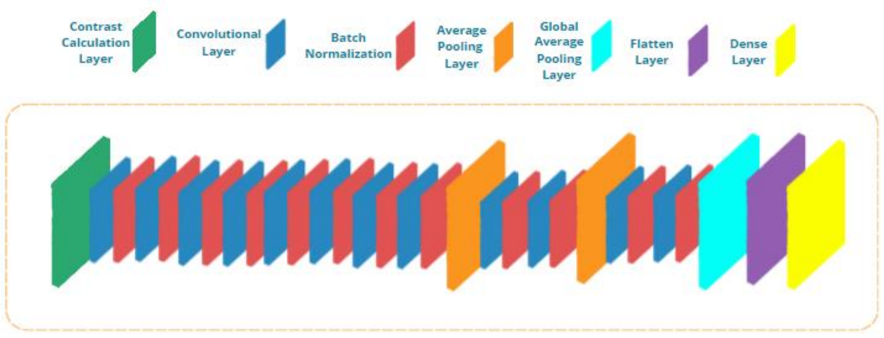


Figure 4-7 Classifier architecture

In the original approach, the classifier phase is designed to transform the **contrast information** between rich and poor texture features into a definitive classification. The process begins with the contrast calculation, where the model **subtracts** the features extracted from poor texture regions from those of rich texture regions. This subtraction is crucial as it highlights texture discrepancies, which are often indicative of synthetic images.

Following this, the contrast map undergoes several layers of **convolution** with **ReLU** activations and **batch normalization**. These **convolutional** layers refine and amplify significant texture patterns within the contrast map. Batch normalization is applied after each convolution to stabilize the learning process, ensuring consistent and efficient training.

To further streamline the information, the model applies **average pooling layers**, reducing the spatial dimensions of the feature maps and focusing on the most relevant aspects of the texture contrast. This step is followed by additional convolutional and pooling layers, which deepen the model's understanding of the extracted features, ensuring that even subtle differences are captured.

The refined feature maps are then passed through a **global average pooling layer**, which condenses each feature map into a single value. This operation effectively reduces the dimensionality of the data while preserving the most important information. The output is **flattened** into a single-dimensional vector, preparing it for the final classification. The final step involves a **dense layer** with a **sigmoid** activation function, which converts the flattened vector into a probability score. This score represents the model's confidence in whether the image is AI-generated or real, with values closer to 1 indicating a higher likelihood of the image being synthetic. This entire process ensures that the model can accurately and efficiently classify images based on the nuanced texture contrasts identified in the preprocessing phase.

4.3 Enhanced Detection Framework (Our Proposed Approach)

4.3.1 Enhanced Feature Extraction Layer

To overcome the limitations of the PatchCraft [14] architecture, which relies on simpler, shallow models that primarily focus on extracting basic low-level features like edges and textures, we **introduced** an **enhanced feature extraction layer** (illustrated in Figure 4-8) in our new architecture, bringing several key improvements. The feature extraction layer starts with **convolutional layers (Conv2D)** that detect local pixel correlations within the 64x64 patches, capturing essential texture features. These layers are vital for moving beyond simple edges and textures to more complex patterns that represent detailed aspects of the image. **Batch normalization** layers then stabilize these features, ensuring the model consistently emphasizes critical inter-pixel correlations, which are crucial for accurately distinguishing between real and AI-generated images.

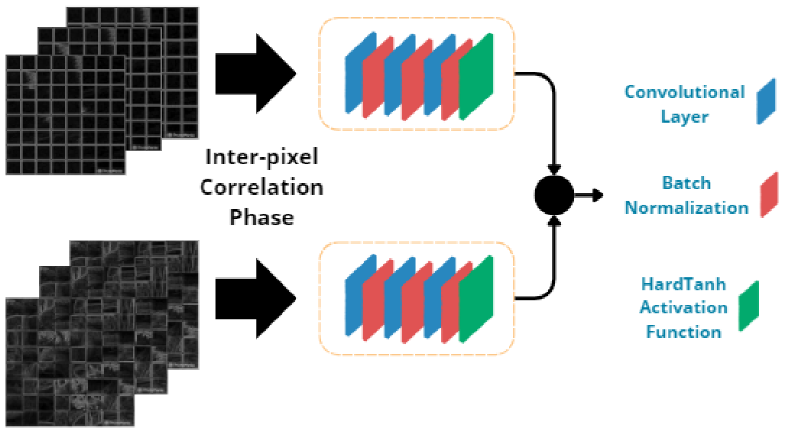


Figure 4-8 Enhanced Feature Extraction Layer

The introduction of a **residual block**, comprising a convolutional layer and batch normalization, plays a crucial role in enhancing the model's performance by incorporating a

skip connection. This skip connection adds the input back to the processed output, improving feature preservation and learning efficiency. In our study, this residual block reinforces the extracted correlations by retaining essential low-level features while simultaneously enabling the model to learn more complex representations. This approach ensures that no valuable information is lost, especially in deeper networks, and that both simple and complex features are effectively captured. Lastly, the **hard_tanh** activation refines the extracted features by emphasizing significant variations, further improving the model's sensitivity to subtle differences.

This deeper architecture, supported by residual learning, shifts the model's capability from capturing basic patterns to understanding more complex textures and subtle differences. These improvements enable the model to more accurately detect nuanced texture information required for precise classification.

4.3.2 Enhanced Classifier Architecture

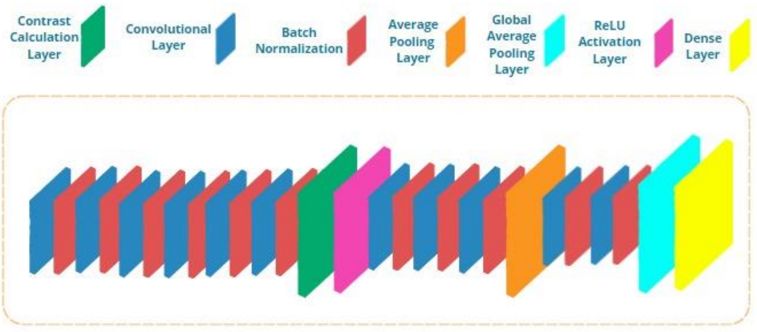


Figure 4-9 New approach's classifier architecture

The updated classifier, as illustrated in Figure 4-9, incorporates several significant enhancements to improve performance. The primary improvements include the introduction of deeper convolutional layers with increased filter sizes, enabling the model to capture more detailed texture information. In contrast to the original architecture, which used a **subtraction** of rich and poor texture features at the beginning of the classification process, the new approach employs an **addition** operation combined with ReLU activation. This change is designed to better emphasize texture discrepancies by preserving and amplifying both the presence and intensity of texture features.

By using an addition operation with ReLU activation, the updated model captures a more nuanced interplay between different texture regions, leading to more accurate contrast maps. This allows the model to detect even subtle differences between real and AI-generated images.

Additionally, the enhanced architecture incorporates convolutional layers with larger filter sizes (128 and 256 filters), which facilitate the learning of higher-level texture patterns and complex relationships crucial for distinguishing between real and synthetic images. The updated model also employs global average pooling which help capture subtle details while reducing spatial dimensions without losing critical information. This ensures that high-resolution features are retained throughout the classification process.

These advancements result in a more robust classifier that is better equipped to differentiate between real and AI-generated images. By leveraging detailed texture information and advanced contrast representations, the updated model improves its ability to detect subtle artifacts and discrepancies present in synthetic images. This leads to higher classification accuracy and reliability in distinguishing between authentic and artificially generated content.

4.4 Summary of Key Improvements in the Proposed Approach

The enhancements introduced in our proposed architecture, as compared to the original PatchCraft [14] framework, provide significant improvements in detecting AI-generated images. The key advancements include:

- **Deeper Feature Extraction Layer**: The enhanced feature extraction layer incorporates deeper convolutional layers, allowing the model to capture more complex and detailed texture patterns. The original PatchCraft [14] relied on shallow models that primarily extracted basic low-level features, but our approach moves beyond simple edge detection to capture more intricate, high-level features crucial for distinguishing between real and AI-generated images.

- **Residual Block Introduction**: By integrating a residual block with a skip connection, the model preserves essential low-level features while enabling learning of more complex patterns. This prevents information loss and ensures that both simple and intricate texture details are retained, enhancing the model’s ability to capture subtle variations in AI-generated content.

- **Enhanced Classifier Architecture**: The classifier architecture was significantly upgraded by introducing deeper convolutional layers with increased filter sizes (up to 512 filters). These layers allow the model to better differentiate between real and synthetic textures by learning more sophisticated texture representations and relationships.
- **Addition Operation with ReLU Activation**: Replacing the original subtraction-based operation with an addition operation, combined with ReLU activation, enables the model to emphasize and amplify texture discrepancies. This approach enhances the model's ability to detect subtle artifacts in synthetic images by preserving both positive and negative texture contrasts.

4.5 Experimental Results

4.5.1 Dataset Description

In this study, we utilized the FFHQ (Flickr-Faces-HQ) dataset for real images, an openly accessible resource on Kaggle, known for its high-quality images of human faces [15]. FFHQ is particularly valuable due to its diversity in age, ethnicity, and background, making it an excellent choice for training robust face detection models. The images in this dataset are of high resolution (512x512), ensuring that a broad range of facial features and details are captured, which is crucial for developing accurate and reliable models.

For the generated images, we used StyleGAN2, a state-of-the-art generative adversarial network recognized for its ability to produce highly realistic synthetic faces. The choice of StyleGAN2 was driven by its complexity; as discussed in the previous chapter, many existing detection techniques struggle with images generated by this network, often resulting in lower accuracy rates compared to other GANs. By focusing our detector training specifically on StyleGAN2-generated images, we aim to enhance detection accuracy and robustness, particularly in scenarios where existing methods have shown limitations.

To generate the fake images, we employed a custom Python-based tool optimized for downloading large quantities of images from thispersondoesnotexist.com. This website leverages StyleGAN2 to generate highly realistic images of human faces, each of which is entirely fictional and does not correspond to any real individual. Every time the page is refreshed, a new face is created on-the-fly by the GAN, which has been extensively trained on a large dataset of real human faces to accurately replicate the subtleties of facial features, expressions, and textures. The site serves as a prime example of the advanced capabilities of modern AI in generating photorealistic images.

Given the requirement to generate thousands of images for our training dataset, we employed a Command-Line Interface (CLI) tool specifically optimized for this task. This tool utilized the *aiohttp* library to manage asynchronous HTTP requests, which significantly accelerated the image download process. By fine-tuning parameters such as concurrency limits and backoff strategies, we successfully downloaded 10,000 unique images in under an hour, ensuring that all images were consistently sized at 512x512 pixels.

Our dataset comprises 42,000 real images from the FFHQ dataset and 42,000 fake images generated by StyleGAN2 for training. For validation and testing, we used an additional set of 14,000 real and 14,000 fake images each. This balanced dataset allows for a thorough evaluation of our model's effectiveness in distinguishing between real and AI-generated images.

Dataset	Real faces (FFHQ)	Fake faces (StyleGan2)
Training	42000	42000
Validation	14000	14000
Testing	14000	14000

Table 4-1 Dataset Composition for Training, Validation, and Testing

4.5.2 Training Process

To prepare the data for training, the images were first shuffled to introduce randomness, preventing the model from learning any unintended patterns based on the sequence of the data. Next, the data was processed through a series of mapping functions that applied the necessary preprocessing steps, including the extraction of rich and poor texture features from each image. The processed images were then grouped into batches of 16, ensuring that each batch provided a diverse set of training examples.

The model was trained on the training dataset using the Adam optimizer, a widely used optimization algorithm in deep learning. The Adam optimizer is an extension of the stochastic gradient descent (SGD) algorithm, combining the advantages of both Adaptive Gradient

Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). The update rule for the Adam optimizer is given by:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t$$

Where:
$\theta_t$are the model parameters at time step$t$,
$\eta$is the learning rate,
$\hat{m}_t$is the biased-corrected first moment estimate(related to the gradient),
$\hat{v}_t$is the biased-corrected second moment estimate(related to the squared gradient),
$\epsilon$is a small constant to prevent division by zero

The training process spanned 30 epochs with a batch size of 16. During each epoch, the model's parameters were updated through backpropagation, minimizing the loss function to improve accuracy. The Binary Cross-Entropy loss function, commonly used for binary classification tasks, was employed. The Binary Cross-Entropy loss is defined as:

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right]$$

Where:
$N$is the number of samples,
$y_i$is the true label(0 or 1),
$\hat{y}_i$is the predicted probability.

The training process was conducted on an Nvidia RTX 4090 GPU with 16GB of memory, taking approximately 4 days and 13 hours to complete. During this time, the model's performance was continuously monitored by tracking both accuracy and loss on the training and validation datasets. The figure below presents the results, showing the trends in training and validation accuracy and loss over the course of the training. These metrics were crucial for evaluating the model's learning progress and ensuring it did not overfit to the training data.

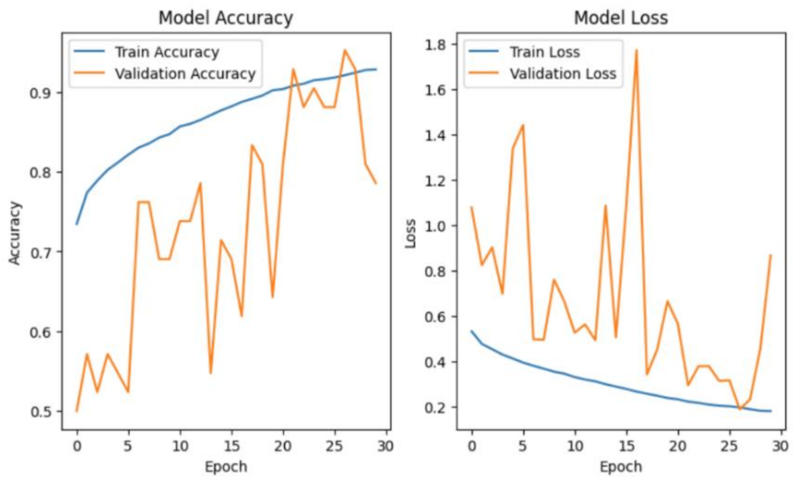


Figure 4-10 Training and validation accuracy and loss curves over the course of the training process

To optimize the training process, the model was trained using TensorFlow's mixed precision policy, which accelerates training by utilizing 16-bit floating-point numbers (half-precision) instead of the standard 32-bit floating-point numbers (single-precision) whenever possible. This approach not only speeds up computation but also reduces memory usage, enabling the model to handle larger batches or more complex architectures.

Throughout the training, the model's weights were periodically saved to checkpoint files. This checkpointing strategy ensured that the best-performing model, as determined by the lowest validation loss, was preserved.

Upon completing the training process, we analyzed the trends in the model's accuracy and loss curves. It was observed that validation accuracy began to decline and validation loss started to increase at epoch 28. Based on this analysis, we selected the model from checkpoint at epoch 27, which exhibited the lowest validation loss, for further evaluation and testing on the separate test dataset. This saved model encapsulates the learned patterns and correlations between rich and poor textures, making it well-optimized for accurately classifying real and AI-generated images.

4.5.3 Model Evaluation and Performance Metrics

After completing the training process and selecting the best-performing model based on validation loss, the model was evaluated on a separate test dataset to assess its effectiveness. The test dataset consisted of an equal number of real and AI-generated images, totaling 28,000 images, offering a solid foundation for assessing the model's generalization capabilities. The evaluation metrics used to assess the model's performance are as follows:

- **Confusion Matrix:** provides a detailed breakdown of the model's classification results, including:
    - **True Positives (TP):** The number of correctly identified AI-generated images.
    - **True Negatives (TN):** The number of correctly identified real images.
    - **False Positives (FP):** The number of real images incorrectly classified as AI-generated.
    - **False Negatives (FN):** The number of AI-generated images incorrectly classified as real.

The confusion matrix helps identify the types of errors the model makes and the balance between correct and incorrect classifications.

- **Accuracy:** is the proportion of correctly classified images (both real and fake) out of the total number of images in the test dataset. It represents the model's overall effectiveness in accurately classifying both real and AI-generated faces. It is calculated as:

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

- **Precision:** is the proportion of correctly identified positive instances (AI-generated images) out of all instances classified as positive. It focuses on the model's ability to correctly identify AI-generated faces when it makes a positive prediction. It is calculated as:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Precision is particularly important when the cost of false positives is high—such as incorrectly flagging real images as AI-generated, which could result in unnecessary actions or misjudgments.

- **Recall (Sensitivity or True Positive Rate):** is the proportion of correctly identified positive instances out of all actual positive instances. It measures the model's ability to detect AI-generated faces among all the AI-generated faces present. It is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN}$$

A high recall indicates that the model is effective at identifying AI-generated images, minimizing the number of false negatives. This is particularly crucial in scenarios where missing an AI-generated image (false negative) could lead to significant consequences, such as allowing a false identity to go undetected.

- **F1-Score:** The F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-Score is particularly useful in situations where precision and recall need to be balanced, offering a comprehensive measure of the model's performance.

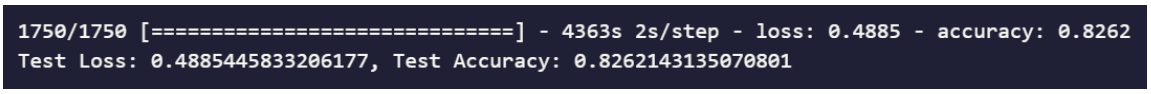
- **AUC (Area Under the Curve)**

The AUC metric evaluates the model's ability to distinguish between classes by measuring the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve plots the true positive rate (recall) against the false positive rate (which is the proportion of real images incorrectly classified as AI-generated out of all actual real images) at various threshold settings. The AUC is calculated as the integral of the ROC curve, and its value ranges from 0 to 1, with 1 indicating perfect classification and 0.5 representing random guessing. A higher AUC value signifies that the model is effective at ranking positive instances higher than negative ones.

4.6 Experimental Evaluation

4.6.1 Results of Model Evaluation and Performance Metrics

The evaluation of the trained model on the test dataset provided valuable insights into its ability to differentiate between real and AI-generated images.



```
1750/1750 [==============================] - 4363s 2s/step - loss: 0.4885 - accuracy: 0.8262
Test Loss: 0.4885445833206177, Test Accuracy: 0.8262143135070801
```

Figure 4-11 Test Accuracy and Loss of the Model Saved at Epoch 27

The model, saved at epoch 27, achieved a test accuracy of 82.62% and a test loss of 0.4885. These results demonstrate the model's effectiveness in accurately classifying over 82% of the test dataset, distinguishing between real and AI-generated content. The relatively low loss further underscores the model's robustness, making it well-suited for practical applications that require reliable detection of AI-generated images.

The results of the evaluation and performance metrics discussed in the previous section are summarized as follows:

- **Confusion Matrix,** as shown in the figure bellow, provides a detailed breakdown of the model’s performance:

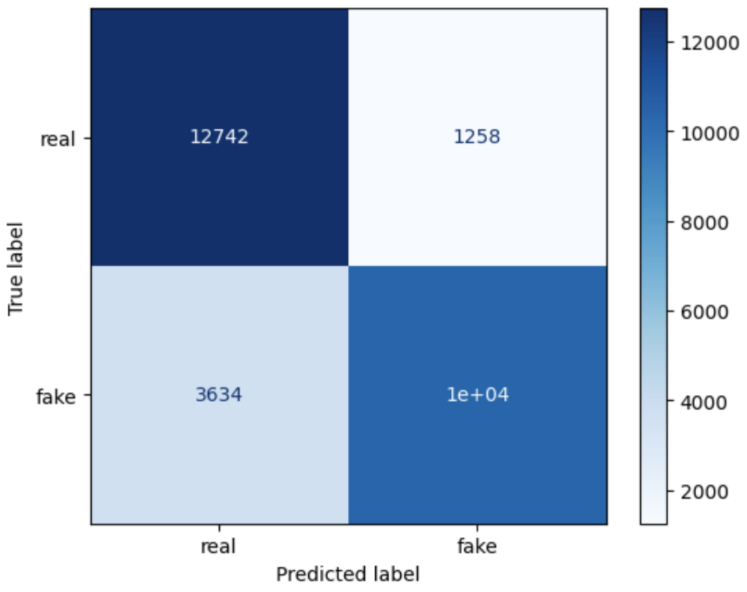
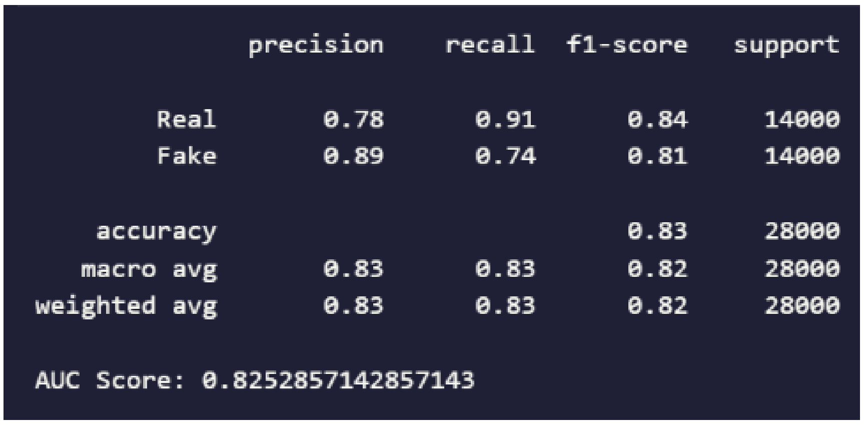


Figure 4-12 Confusion Matrix illustrating the model’s classification performance

  - **True Positives (TP): 10,000** AI-generated images were correctly classified, reflecting the model’s ability to accurately detect synthetic images.
  - **True Negatives (TN): 12,742** real images were correctly identified as real, demonstrating the model’s proficiency in recognizing authentic content.

- **False Positives (FP): 1,258** real images were incorrectly classified as AI-generated, indicating areas where the model could be improved to reduce false alarms.
- **False Negatives (FN): 3,634** AI-generated images were misclassified as real, suggesting potential enhancements in the model's sensitivity to subtle synthetic artifacts.

The confusion matrix reveals a generally balanced performance across both classes, though there is a noticeable number of false negatives, which could be an area for further improvement.



```
              precision    recall  f1-score   support

        Real       0.78      0.91      0.84     14000
        Fake       0.89      0.74      0.81     14000

    accuracy                           0.83     28000
   macro avg       0.83      0.83      0.82     28000
weighted avg       0.83      0.83      0.82     28000

AUC Score: 0.8252857142857143
```

Figure 4-13 Model Performance Metrics on the Test Dataset

- **Accuracy:** As depicted in Figure 4-13, the model achieved an overall accuracy of 83%. This means that the model correctly classified 83% of all images in the test dataset, regardless of whether they were real or AI-generated.
- **Precision:** As shown in Figure 4-13, precision is calculated separately for each class—real images and AI-generated images. This detailed breakdown provides valuable insights into the model's performance across different categories. However, since the primary task of our project is to detect AI-generated images, the focus is on how accurately the model identifies AI-generated faces out of all the images it predicted as AI-generated.

  Precision is **0.89**, meaning that when the model predicts an image as AI-generated, 89% of those predictions are correct. This reflects the model's strong ability to correctly identify AI-generated faces when it makes a positive prediction.

- **Recall:** As illustrated in Figure 4-13, recall is calculated separately for each class—real images and AI-generated images. This detailed breakdown offers valuable insights into the model's performance within each category. Given that the primary objective of our project is to detect AI-generated images, the focus is on how effectively the model identifies AI-generated images out of all the actual AI-generated images in the dataset.

  The recall for AI-generated faces is 0.74, meaning that the model correctly identified 74% of all the AI-generated images in the dataset. This reflects the model's effectiveness in detecting AI-generated faces.

- **F1-Score:** In Figure 4-13, the F1 score is calculated for both classes—AI-generated and real faces. However, since our primary task is to detect AI-generated images, the emphasis is on the balance between precision and recall specifically for AI-generated images. The F1 score for AI-generated images is 0.81, offering a balanced measure of both precision and recall in this context.
- **AUC Score:** The model achieved an AUC score of 0.8253, which indicates that it has a strong ability to distinguish between real and AI-generated images. A higher AUC score signifies better performance in ranking positive instances higher than negative ones.

4.6.2 Comparison with PatchCraft Method

To further evaluate the performance of our proposed method, we conducted a direct comparison with the PatchCraft [14] framework. We utilized the same original architecture for PatchCraft [14] to ensure it was tested in its intended configuration, as designed by its authors. This approach guarantees a reliable and accurate comparison.

Both models were evaluated on the same dataset of real and AI-generated images, ensuring a balanced and fair assessment. By using identical evaluation metrics—such as accuracy, precision, recall, F1-score, and average precision—and maintaining a consistent testing environment, we minimized any potential variability that could skew the results. Both models were trained from scratch in the same environment, using identical settings, such as the number of epochs and training parameters.

We chose PatchCraft [14] for comparison because our method is built upon and enhances its framework. Given that PatchCraft's [14] texture-based approach aligns closely with ours, and it has demonstrated high accuracy in AI detection, it serves as the most relevant and reliable baseline for our study. This made it the ideal benchmark for a meaningful comparison between the two techniques.

Technique	**PatchCraft** [14]	**Our Approach**
Accuracy	72.35%	82.62%
Precision (Real)	68.92%	78%
Precision (AI-generated)	79%	89%
Average Precision	89%	91.22%
Recall (Real)	79%	91%
Recall (AI-generated)	62%	74%
F1-score (Real)	80%	84%
F1-score (AI-generated)	68%	81%

Table 4-2 Comparison of Performance Metrics Between PatchCraft [14] and Our Enhanced Approach

The comparison reveals notable improvements in performance with our approach across all key metrics. Our model achieves an accuracy of 82.62%, significantly outperforming PatchCraft's 72%, demonstrating greater reliability in overall classification. When it comes to precision, particularly for AI-generated images, our model excels with a score of 89% compared to PatchCraft's 79%, highlighting its stronger ability to accurately identify synthetic content. Similarly, our model shows a marked improvement in recall for AI-generated images, scoring 74% versus PatchCraft's 62%, indicating its enhanced ability to detect a higher proportion of AI-generated images and reduce false negatives.

Moreover, the F1-score, which balances both precision and recall, is significantly higher in our approach, reaching 81% for AI-generated faces compared to PatchCraft's 68%.

The comparison clearly demonstrates that the enhancements we made to the PatchCraft [14] framework significantly improved performance, resulting in higher accuracy, precision, recall, and F1-scores across all key metrics. These improvements affirm the effectiveness of our approach in better detecting and classifying AI-generated images.

4.7 Deployment

As part of our project **Recognizing Faces Generated by Artificial Intelligence**, our goal was to develop a user-friendly platform that leverages advanced deep learning techniques to detect AI-generated faces. This initiative transforms our AI face detection model from a research tool into an accessible web application that anyone can use. Named **TrueFaces**, the platform allows users to upload facial images, which are then analyzed by our AI model to determine whether the image is real or AI-generated.

The significance of **TrueFaces** lies in its ability to address pressing concerns around deepfakes, which can be exploited for malicious purposes such as misinformation, fraud, or identity theft. By providing this solution through an intuitive web interface, we enable both individuals and organizations to verify the authenticity of facial images without needing specialized knowledge of machine learning or AI.

The following subsections describe the system architecture, detailing how each component contributes to delivering seamless and accurate predictions.

4.7.1 System Architecture

The system architecture of **TrueFaces** combines an intuitive user interface with our deep learning model specifically designed for AI face detection. The web application's frontend allows users to easily upload facial images, which are then processed via the Flask API. Flask acts as the bridge between the frontend and backend, managing the image upload, invoking the AI model, and delivering the prediction results.

As shown in Figure 4-14, this architecture ensures smooth interaction between the user interface, backend services, and the AI model, enabling efficient and accurate detection of whether a face is real or AI-generated.

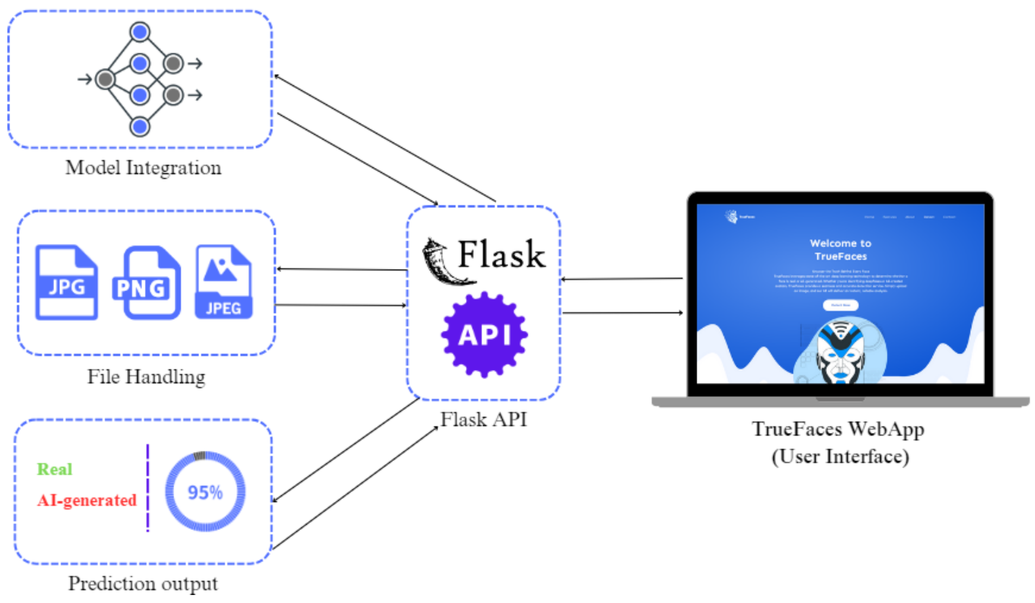


Figure 4-14 System Architecture Diagram

4.7.2 Frontend Development

The frontend of the **TrueFaces** web application is crafted to provide a smooth and user-friendly experience. Upon arrival, users are greeted by a simple interface: the first section welcomes them and invites them to upload their facial images, as shown in Figure 4-15(a). Following this, there is a features section highlighting the unique benefits and reasons for choosing our platform (see Figure 4-15(b)). The primary section is the upload area, where users can easily submit images via a drag-and-drop feature or by browsing their files. Once uploaded, the results are displayed clearly, indicating whether the face is real or AI-generated, accompanied by a confidence score, as illustrated in Figure 4-15(c). Lastly, the footer section contains a contact form for user inquiries and additional footer information (see Figure 4-15(d)).

By leveraging modern web technologies such as HTML, CSS, and JavaScript, the application is built to be responsive, fast, and easy to navigate. The frontend is fully adaptable to various devices and screen sizes, while JavaScript is used to dynamically handle the image upload process, enhancing user engagement and interactivity.

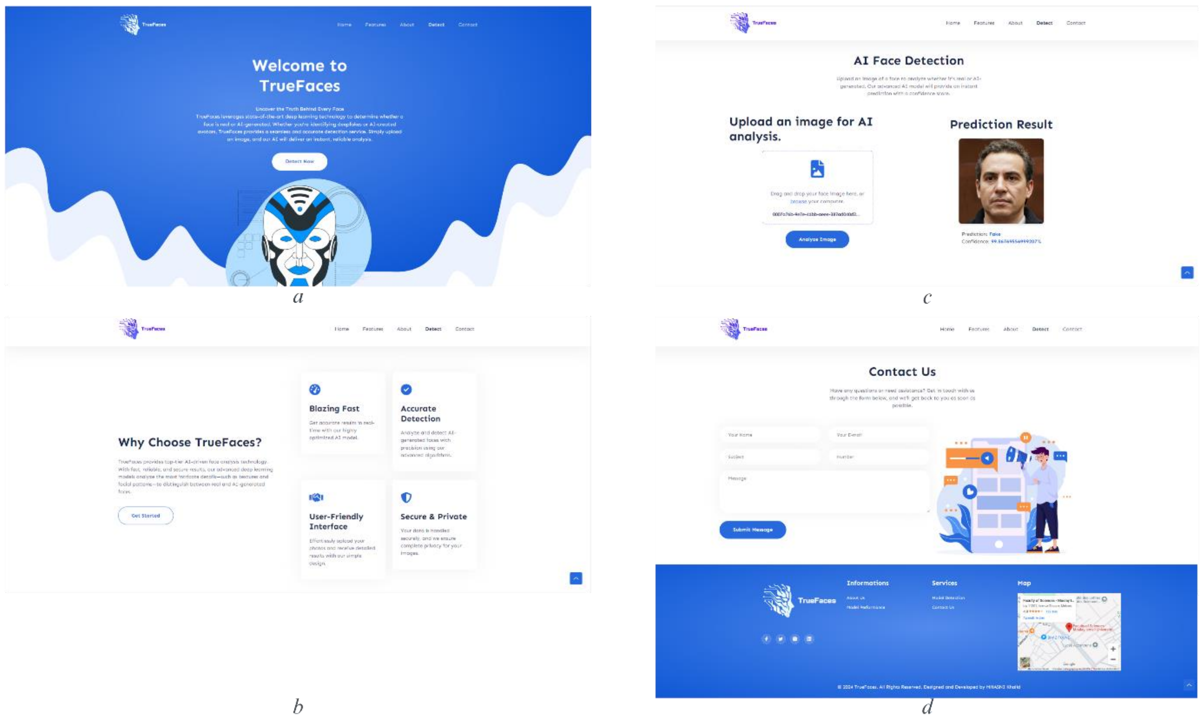


Figure 4-15 TrueFaces web interface overview. (a) Welcome section. (b) Features section. (c) Image upload and result display. (d) Footer with contact form and website details

4.7.3 Backend Integration

Flask is a lightweight and flexible Python web framework that enables rapid development of web applications. It handles essential tasks like routing user requests (e.g., file uploads) and sending responses, while seamlessly integrating with machine learning models. As the core of the application, Flask connects the front-end user interface to the back-end AI model, ensuring smooth request processing.

When a user uploads an image, Flask processes it and passes it to the AI model for analysis. After the model predicts whether the face is real or AI-generated, Flask formats the result and returns it to the frontend, delivering a smooth, real-time experience. Flask's integration with Python-based deep learning models makes it an ideal choice for combining AI power with a user-friendly web interface.

4.8 Conclusion

In this chapter, we presented a comprehensive exploration of our enhanced detection framework, building upon the foundational PatchCraft [14] architecture. By addressing the limitations inherent in the original PatchCraft [14] method, particularly with respect to shallow feature extraction and basic texture analysis, our proposed approach offers a deeper and more refined model capable of capturing complex patterns.

Key enhancements, such as the introduction of residual blocks, deeper convolutional layers, and a more sophisticated classifier, significantly improve the model's ability to differentiate between real and AI-generated images. These advancements ensure that our framework not only captures intricate textures but also maintains high-resolution features throughout the classification process.

Our experimental results clearly demonstrate the effectiveness of these improvements, showcasing superior performance across key metrics, including accuracy, precision, recall, and F1-score, compared to the original PatchCraft [14] method. These findings affirm that the enhancements introduced in our framework provide a more robust and reliable solution for detecting AI-generated content, marking a significant step forward in the field of AI-based image detection.

CHAPTER 5 CONCLUSION AND FUTURE WORK

5.1 Overview and Recap of the Project

The primary objective of this project was to develop a robust detection framework for distinguishing between real and AI-generated human faces. The growing sophistication of GANs has led to the creation of highly realistic synthetic faces, making it increasingly difficult to differentiate them from real ones. This study set out to address these challenges by improving upon the PatchCraft [14] framework, focusing on texture-based detection techniques and incorporating enhancements such as deeper feature extraction layers and advanced classifier architectures.

Throughout the course of the project, we successfully trained and evaluated our enhanced detection model that outperformed baseline methods in several key areas. Our model was rigorously tested on a balanced dataset of real and AI-generated faces and demonstrated significant improvements in accuracy, precision, recall, and overall robustness. The findings have important implications for fields such as digital media authentication, identity verification, and deepfake detection.

5.2 Key Findings and Contributions

The project introduced several key improvements over existing methods, particularly when compared to the PatchCraft [14] framework. Notable findings and contributions include:

- **Performance of the enhanced detection framework**: The proposed framework significantly outperformed PatchCraft [14], achieving higher accuracy and robustness. Specifically on AI-generated faces produced by more advanced GANs like StyleGAN2.
- **Deeper feature extraction**: One of the major enhancements was the introduction of a deeper feature extraction layer, which allowed the model to capture more intricate texture patterns. This deeper architecture enabled the model to more effectively differentiate between real and synthetic textures.

- **Residual block integration**: The addition of residual blocks with skip connections improved the model's ability to retain low-level features while learning complex patterns, resulting in more accurate texture analysis.

- **Enhanced classifier architecture**: By introducing deeper convolutional layers and a more refined classifier architecture, the model was able to detect subtle differences in rich and poor texture regions, leading to more precise classifications.

- **Application of the TrueFaces platform**: The deployment of the detection model through the TrueFaces web platform provides a practical and user-friendly solution for detecting AI-generated faces in real-time. This is an important contribution to the growing demand for tools that combat malicious uses of deepfake technology.

5.3 Limitations of the Project

While the project achieved significant advancements, there were several limitations that affected the outcomes:

- **Computational constraints**: Training the model on a large dataset demanded significant computational resources. The process, which took several days, required advanced hardware like the Nvidia RTX 4090 GPU. This dependency on high-end infrastructure limits the accessibility of the approach for researchers with less powerful resources.
- **Generalization concerns**: The model’s ability to generalize to unseen generative models was not fully explored. Although it showed strong performance on StyleGAN2-generated images, it was not tested on more recent models such as diffusion-based models or autoregressive architectures like DALL-E. Future research will need to address this limitation.

5.4 Future Research Directions

There are several potential avenues for future research that could build on the findings of this project:

- **Refining texture analysis techniques**: Future research could explore advanced texture analysis methods that capture even more subtle differences between real and AI-generated images. Techniques such as multi-scale texture feature extraction or incorporating spectral domain analysis could improve detection accuracy.

- **Exploring new AI architectures**: As the field of generative models evolves, incorporating hybrid detection approaches that leverage both spatial and temporal features could be an interesting research direction. Combining traditional texture-based analysis with more advanced neural architectures like transformers could further enhance detection capabilities.
- **Future of TrueFaces**: As the TrueFaces platform evolves, there are numerous possibilities for improving its capabilities. Future versions of the platform could include support for video analysis, allowing users to upload and analyze video content for deepfake detection. Additionally, expanding the platform's API offerings could make it a valuable tool for businesses looking to integrate AI-detection capabilities into their workflows.

5.5 Closing Remarks

The ongoing advancements in AI-generated content, particularly in the creation of synthetic human faces, present both opportunities and challenges. While the ability to generate photorealistic faces holds great potential for creative industries, it also poses significant ethical and security risks, particularly in the realm of deepfakes. This research has demonstrated the effectiveness of texture-based detection methods in distinguishing between real and AI-generated faces, but it is clear that as GANs continue to evolve, detection methods must keep pace.

The findings of this project highlight the dual role of AI: as both a tool for creative expression and a potential threat when misused. Moving forward, the development of more robust and adaptive detection frameworks will be crucial in maintaining the integrity of digital media and protecting individuals from the misuse of AI-generated content.

BIBLIOGRAPHY

[1] W. &. K. T. &. G. A. &. D. S. &. P. A. &. P. A. &. A. A. Nie, "Semi-Supervised StyleGAN for Disentanglement Learning," in *International Conference on Machine Learning*, 2020.

[2] B. a. C. M. a. C. J. a. M. X. a. J. Y.-G. Zi, "WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[3] P. &. M. S. Korshunov, "Deepfake detection: humans vs. machines.," 2020.

[4] S.-Y. &. W. O. &. Z. R. &. O. A. &. E. A. Wang, "CNN-Generated Images Are Surprisingly Easy to Spot… for Now," in *Conference: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] S. a. N. P. a. B. P. a. N. D. a. M. A. Lahange, "Computer Vision Techniques in Autonomous Vehicles: A Survey," in *4th International Conference on Communication & Information Processing (ICCIP)* , 2022.

[6] X. Zhao, "Research and application of deep learning in image recognition," in *Journal of Physics: Conference Series*, 2023.

[7] D. O. &. P. O. &. T. C. Esan, "Generative Adversarial Networks: Applications, Challenges, and Open Issues," in *Deep Learning - Recent Findings and Researches*, vol. Multimedia Tools and Applications, 2023.

[8] P. W. K. A. Shi Dong, "A survey on deep learning and its applications," in *Computer Science Review*, 2021.

[9] M. a. P. S. Schinas, "SIDBench: A Python framework for reliably assessing synthetic image detection methods," in *ICMR '24: International Conference on Multimedia Retrieval*, 2024.

[10] J. &. E. T. &. S. L. &. F. A. &. K. D. &. H. T. Frank, "Leveraging Frequency Analysis for Deep Fake Image Recognition.," in *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, 2020.

[11] U. &. L. Y. &. L. Y. J. Ojha, "Towards Universal Fake Image Detectors that Generalize Across Generative Models," in *Conference: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[12] R. &. M. L. &. J.-X. F. &. X. X. &. W. J. &. L. Y. Wang, "FakeSpotter: A Simple Baseline for Spotting AI-Synthesized Fake Faces," 2019.

[13] Y. &. J. S. &. K. L. &. X. H. &. N. K. &. L. S. Ju, "Fusing Global and Local Features for Generalized AI-Synthesized Image Detection," in *2022 IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, 2022.

[14] N. Z. a. Y. X. a. S. L. a. Z. Q. a. X. Zhang, "PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection," 2024.

[15] A. ROUGETET, "Flickr-Faces-HQ Dataset (FFHQ)," [Online]. Available: https://www.kaggle.com/datasets/arnaud58/flickrfaceshq-dataset-ffhq.