HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
AI&Computer Science Department

[AI VS AI]

A graduation project dissertation by:

[Thanaa Khairy Sayed (20210247)]
[Jawaher Ibrahim Ali (20210258)]
[Rodina Yahia Mohamed (20210344)]
[Mona Mohamed Abdel Aziz (20210967)]
[Omar Mohamed Omar (20210624)]
[Abdel Rahman Khaled Hassan (20210499)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Computer Science Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

[Dr. Mohammed El-Said]

June 2025

ACKNOWLEDGEMENT

It is our pleasure to express our heartfelt thanks to **Dr. El-Said**, Supervisor for our graduation project, for his supervision and guidance which enabled us to understand and develop this project.

Abstract

Our project presents a dual-function system that integrates state-of-the-art artificial intelligence for both **image generation** and **image authenticity detection**. The generation module fine-tunes **Stable Diffusion v1.5** using **Low-Rank Adaptation (LoRA)** on the **FFHQ dataset** to produce high-quality, realistic human face images from user-provided descriptions. In parallel, the detection module employs deep learning techniques to identify whether an image has been artificially generated by models such as **GANs or diffusion models**, or is a real, human-captured photograph. Together, these components serve the growing need for creative AI applications alongside tools for verifying visual media authenticity in an era of increasing synthetic content proliferation

AI VS A: Website based on AI & image generation and detection services, consisting of

Two main parts:

- **Generation section: Abstract Steps**

- **First Phase – Text Prompt Input**

  The user enters a text description (prompt) describing the desired human face (e.g., “a smiling young woman with curly hair”).

- **Second Phase – Fine-tuned Model Activation**

  The system passes the prompt to a **Stable Diffusion v1.5 model**, which has been **fine-tuned using LoRA (Low-Rank Adaptation)** for improved efficiency.

- **Third Phase – Specialized Training Foundation**

  This model has been trained on the **FFHQ (Flickr-Faces-HQ) dataset**, which provides high-resolution human face images, enabling the system to learn **facial structures, features, and details**.

- **Fourth Phase – Image Synthesis**

  The diffusion model transforms the text into a detailed image, generating **photorealistic human faces** with natural-looking skin texture, lighting, and facial symmetry.

- **Final Step – Output Delivery**

  The generated image is saved under the user's profile and displayed as a **PNG output**, ready for preview, download, or email sharing.

- **Detection section: Abstract Steps**

- **First Phase – Input Image Upload**

  The user uploads an image either from their local device or through a URL.

- **Second Phase – Feature Extraction**

  The system uses a high-capacity CNN (e.g., **EfficientNet** or **ResNet**) to extract visual features and statistical patterns from the image.

- **Third Phase – Multi-domain Analysis**

  The extracted features are compared against patterns learned from **various AI generation models** like StyleGAN, StyleGAN2, EF3D, and Stable Diffusion.

- **Fourth Phase – Fine-grained Classification**

  A trained deep learning classifier examines **tiny artifacts** such as texture irregularities, lighting mismatches, unnatural edges, and noise inconsistencies that indicate whether the image is AI-generated or real.

- **Fifth Phase – Robust Evaluation**

  The classifier makes a final prediction, even if the image has been **compressed, resized, or filtered**, and returns the result with a **confidence score**.

- **Final Step – Output**

  The system displays whether the image is **real or AI-generated**, along with the detection confidence percentage.

Chapter 1: Introduction

Our idea is about helping content creators generate realistic human faces for media without using real models,It supports journalists in verifying the authenticity of images to prevent the spread of fake news,Marketing teams can create diverse avatars and visuals for personalized campaigns and Social media platforms can detect AI-generated images to ensure transparency and trust.

1.1 Introduction

With the rapid evolution of generative AI, particularly models like GANs and diffusion models, synthetic image creation has reached unprecedented levels of realism. While these advancements unlock creative potential and automation, they also raise concerns around misinformation, digital authenticity, and visual media manipulation. This project addresses these dual aspects by offering both **AI image generation** and **detection** capabilities within a unified system

1.2 Motivation

- **Rising realism of AI-generated images** makes it increasingly difficult to differentiate between synthetic and real visuals, especially human faces.
- The spread of **deepfakes and manipulated content** poses societal risks in media, politics, and identity verification.
- There is a growing demand for tools that can **generate high-quality AI images** responsibly and **detect fake visuals** for security and ethical

oversight.

1.3 Problem Definitions

Two core challenges are addressed:

1) **Generation Task**: How to generate photorealistic human faces with fine detail using user-provided text descriptions.
2) **Detection Task**: How to accurately distinguish AI-generated images (e.g., from GANs or diffusion models) from authentic, human-taken images.

- Purpose

- Enable **controlled generation of realistic human faces** using natural language prompts.
- Build a **robust detection pipeline** to classify and verify whether an image is AI-generated or real.

- Provide tools that support **content creation, digital forensics, and media integrity** in a single, integrated platform

1.4 Scope

- **Image Generation**: Fine-tuning **Stable Diffusion v1.5** using **LoRA** on the **FFHQ dataset**, with a focus on facial features and visual detail.
- **Image Detection**: Developing a classifier using **deep neural networks** trained on real vs. synthetic image datasets, targeting outputs from GANs and diffusion models.
- **User System**: Web-based interface to input prompts, upload images, and view predictions. Includes user management, admin roles, and session-based access

1.5 Overview

- **Text-to-Image Generation**:
- Prompt → Stable Diffusion + LoRA → Face Image Output.
- Fine-tuned on FFHQ for realistic human features.
- **Image Authenticity Detection**:
- Uploaded image → CNN-based classifier → Real or AI-generated label.
- Trained on datasets of both real and synthetic (e.g., StyleGAN, DDPM) images.
- **User Interface & APIs**:
- Register/login, image upload, generation via prompt, detection result, and email-based services.
- Admin dashboard to monitor user activity and manage system content.

1.6 Similar Systems

- **DeepFake Detection (Facebook, Microsoft)**: Focus on detecting manipulated videos or synthetic faces using AI.
- **Generated Media Platforms (e.g., Artbreeder, Runway ML)**: Provide AI-based image creation but lack built-in detection.
- **AI Or Not**: A simple online classifier for distinguishing AI vs. human-made images but limited in scope and dataset coverage

Similar papers

Paper Title	Link	Tools & Libraries	Dataset	Results
AI vs. AI: Can AI Detect AI-Generated Images?	link	CNNs, Transfer Learning, EfficientNetB4, Adam Optimizer, Class Activation Maps	COCO-Stuff Dataset	Accuracy: 100%
Identifying AI-Generated Art with Deep Learning	link	VGG-19, ResNet-50, Vision Transformers	ArtGraph (Zenodo), Artifact (Kaggle)	Accuracy: ViT: 97.5%, ResNet-50: 96.5%, VGG-19: 95.8%
Detection of AI-Created Images Using Pixel-Wise Feature Extraction and CNNs	link	MATLAB, CNNs, SGDM, Wavelet Transform-based Wiener filter	459 AI images (DALL-E, Stable Diffusion, OpenArt); 459 real images (Dresden, VISION, personal collections)	PRNU: 95%, ELA: 98%, F1 Scores: 95% & 98%
Generation of Images from Text Using AI	link	TensorFlow, PyTorch, GloVe, Tkinter (GUI), Flask (Web)	Oxford-102 Flowers dataset + captions from Amazon Mechanical Turk	Discriminator accuracy: 43.4%, showing realistic generation
Text to Image Generation Using AI	link	DALL-E, GPT-3, BigGAN, Kotlin & Java, Transformer models, GANs, Diffusion Models	Internet-crawled dataset of text-image pairs (2D & 3D models)	High-quality images at 256×256, 512×512, 1024×1024. Outperformed others in visual & semantic alignment

1.7 Related work

DeepFake Detection, HuggingFace, LoRA and Do GANs

Background:

As the capabilities of generative AI rapidly evolve, the ability to create hyper-realistic images using models such as GANs and diffusion networks has raised both creative opportunities and concerns around misinformation, manipulation, and digital trust. At the same time, the need to detect such synthetic content has become critical in maintaining authenticity across digital platforms. Key research initiatives, including the **DeepFake Detection Challenge by Facebook AI**, have emphasized the importance of benchmarking detection models. In parallel, open-source libraries like **Diffusers by HuggingFace and Stability AI** have made cutting-edge diffusion models more accessible for developers and researchers. Foundational datasets like **FFHQ (Flickr-Faces-HQ)** have been instrumental in training high-quality generative models, while studies like **"Do GANs Leave Artifacts?" by Yu et al.** have investigated the statistical anomalies in AI-generated images. Moreover, **LoRA (Low-Rank Adaptation)** has provided an efficient method for fine-tuning large models with minimal computational overhead, further enabling the refinement of generation capabilities.

Objective:

To develop a system capable of both generating realistic human face images from natural language prompts and detecting whether a given face image is real or AI-generated, using deep learning techniques. The goal is to support responsible AI use by offering tools for creative generation and image verification.

Methods:

The image generation component of the system is built on **Stable Diffusion v1.5**, fine-tuned using **LoRA** on the **FFHQ dataset** to improve facial realism and detail. The detection component utilizes deep convolutional neural networks (e.g., **EfficientNet**, **ResNet**) trained on a diverse set of real and synthetic face images from sources like **StyleGAN**, **StyleGAN2**, **EF3D**, and **Stable Diffusion**. These networks identify subtle visual artifacts in textures, lighting, edges, and noise patterns to classify images. The system is evaluated across benchmarks to ensure robustness, even under conditions of resizing, compression, and post-processing. Unlike systems that operate independently of user context, this platform is designed to assist its users—such as students, researchers, and professionals—through guided, supervised usage, blending automated AI functions with informed human oversight.

Enhancing Facial Realism: Fine-tuning Stable Diffusion with LoRA on FFHQ Dataset

Project Overview

This project involves fine-tuning Stable Diffusion v1.5 using Low-Rank Adaptation (LoRA) on the Flickr-

Faces-HQ (FFHQ) dataset. The primary goal was to enhance the model's ability to generate high-quality,

realistic human faces with improved detail rendering and natural features.

Motivation

The project was specifically motivated by a significant limitation in Stable Diffusion 1.5's capability to

generate realistic human faces:

Poor Face Quality: The base SD1.5 model often produces faces with noticeable artifacts, unnatural

proportions, and inconsistent features

Easy Detection: Images generated by SD1.5 frequently contain telltale signs of AI generation,

particularly in facial areas (asymmetrical features, unnatural eye placement, distorted teeth, etc.)

Uncanny Valley Effect: Many generated faces fall into the "uncanny valley," appearing almost-but-
not-quite human, creating discomfort for viewers

Limited Photorealism: The base model struggles with realistic skin textures, pore details, and natural
lighting interactions on faces.

By fine-tuning specifically on the high-quality FFHQ dataset with detailed facial descriptions, this project
aimed to address these limitations and create a model capable of generating faces that are significantly
more difficult to distinguish from real photographs.

Dataset

FFHQ Dataset

Description: The Flickr-Faces-HQ dataset consists of 70,000 high-quality PNG images at 1024×1024
resolution

Challenge: Unlike many other datasets used for training image generation models, FFHQ does not
come with associated text descriptions or captions

Advantage: The dataset contains diverse, high-quality facial images with excellent detail, making it
ideal for improving facial generation

Text Description Generation

Approach: To overcome the lack of descriptions, LLaVA (Large Language and Vision Assistant) was
utilized to generate descriptive captions for each face image

Process:

1. Images from the FFHQ dataset were processed through LLaVA
2. The AI model analyzed facial features, expressions, and visible attributes
3. Detailed descriptions were generated for each image, focusing on specific facial characteristics
4. These descriptions were saved as text files corresponding to their respective images

Dataset Preparation

Images were resized to 512×512 resolution for training efficiency
A subset of 52,000 image-caption pairs was selected for training.
Image-caption pairs were created by matching image files with their corresponding descriptions generated by the LLaVA model.

Model Architecture

Base Model

Stable Diffusion v1.5 (runwayml/stable-diffusion-v1-5) was used as the foundation model

LoRA Configuration

Rank (r): 16

Alpha: 32

Target Modules: Attention layers (to_q, to_k, to_v, to_out.0), projection layers (proj_in, proj_out), and convolution layers (conv1, conv2, conv_shortcut)

Dropout Rate: 0.1

Training Strategy

Only the UNet component was fine-tuned, with VAE and text encoder kept frozen

Gradient checkpointing was enabled to reduce memory usage

Trainable parameters were approximately 12.37 million (about 1.42% of total model parameters)

Training Process

Training Configuration

Batch Size: 4

Gradient Accumulation Steps: 4 (effective batch size of 16)

Learning Rate: 5e-5

Learning Rate Scheduler: Cosine with warmup

Mixed Precision: FP16

Number of Epochs: 30

Training Approach

Training Approach:

1. The training was divided into phases corresponding to the dataset size.
2. For every 10,000 images, the model was trained for 5 epochs.

3. In total, with 52,000 images, the model was trained for 30 epochs (i.e., 5 epochs × ~6 phases).
4. LoRA weights were saved and used to resume training across phases to ensure improved convergence.

3. **Monitoring**: Loss values and preview images were generated after each epoch to track progress

Training Infrastructure

GPU acceleration was used for efficient training

The training process was executed in a Kaggle notebook environment

Results and Evaluation

Performance Metrics

The loss consistently decreased over training epochs, indicating effective learning

Average loss values stabilized around 0.137-0.140 in later epochs

Qualitative Assessment

Preview images generated after each epoch showed progressive improvement in:

Facial detail rendering

Texture quality

Natural lighting effects

Overall realism of generated faces

Comparison Testing

Test prompts were created to evaluate the model's capabilities

Comparisons between base model outputs and LoRA-enhanced outputs showed improvements in:

Hair strand detail

Skin texture rendering (pores, subtle imperfections)

Facial feature definition (more natural eye placement, better proportions)

Lighting and shadow subtlety

Reduction in common SD1.5 face artifacts

Improved Photorealism

The LoRA-enhanced model produced faces with:

More natural skin textures and tones

Better handling of lighting on facial features

More realistic eye details (iris definition, catchlights, natural eyelashes)

Improved rendering of fine details like hair strands and subtle skin features

Significantly reduced telltale AI artifacts that make faces easily detectable as fake

Challenges and Solutions

Challenge 1: Lack of Text Descriptions

Problem: FFHQ dataset doesn't include descriptive captions

Solution: Used LLaVA to generate detailed descriptions for each image

Result: Created a comprehensive set of image-caption pairs suitable for fine-tuning

Challenge 2: **Resource Constraints**

Problem: Limited GPU memory for high-resolution training

Solution:

Reduced image resolution to 512×512

Implemented gradient checkpointing

Used LoRA to minimize trainable parameters

Result: Successfully trained on consumer-grade hardware

Challenge 3: **Training Continuation**

Problem: Need to extend training beyond initial epochs

Solution: Implemented checkpoint saving and loading functionality

Result: Successfully continued training from epoch 5 to 30

Challenge 4: Overcoming SD1.5's Face Generation Limitations

Problem: Base model's inherent weakness in realistic face generation

Solution: Targeted fine-tuning on high-quality facial dataset with detailed descriptions

Result: Significant improvement in facial realism and reduction in detectable AI artifacts

Challenge 4: **Overcoming SD1.5's Face Generation Limitations**

Problem: Base model's inherent weakness in realistic face generation

Solution: Targeted fine-tuning on high-quality facial dataset with detailed descriptions

Result: Significant improvement in facial realism and reduction in detectable AI artifacts

Future Improvements

Potential Enhancements

1. **Description Quality:** Refine the LLaVA-generated descriptions with human review for higher quality
2. **Dataset Expansion:** Include more diverse faces from different demographics
3. **Hyperparameter Optimization:** Experiment with different LoRA ranks and learning rates
4. **Extended Training:** Train for more epochs to further improve quality
5. **Resolution Scaling:** Implement progressive resolution scaling during training
6. **Anti-Detection Focus:** Further refine the model to address specific detection algorithms that identify

AI-generated faces

Conclusion

The project successfully demonstrated how LoRA fine-tuning can enhance Stable Diffusion's facial

generation capabilities. By targeting one of SD1.5's most notable weaknesses—poor quality face

generation that is easily detectable as artificial—this approach has created a specialized model capable of

producing significantly more realistic human faces.

The use of AI-generated captions for an unlabeled dataset like FFHQ provides a viable method for

working with similar unlabeled image collections. The resulting model shows markedly improved detail rendering and more realistic facial features compared to the base model, making the generated images less easily identifiable as AI-created content.

EfficientNet-B6 Model

EfficientNet-B6 is a high-performance convolutional neural network architecture that is part of the EfficientNet family, which was introduced by Google AI in 2019. The EfficientNet models use a novel compound scaling method to systematically scale up the
network's depth, width, and resolution in a balanced way, leading to improved accuracy and efficiency compared to traditional CNN architectures.
EfficientNet-B6 specifically offers state-of-the-art accuracy while maintaining a relatively low computational cost compared to other large models like ResNet or Inception. It is designed for high-resolution image inputs (528x528) and has approximately 43 million parameters. The model is well-suited for tasks requiring fine-grained image classification and is widely used in industrial and academic applications

C.7 Efficient Net Model Architecture

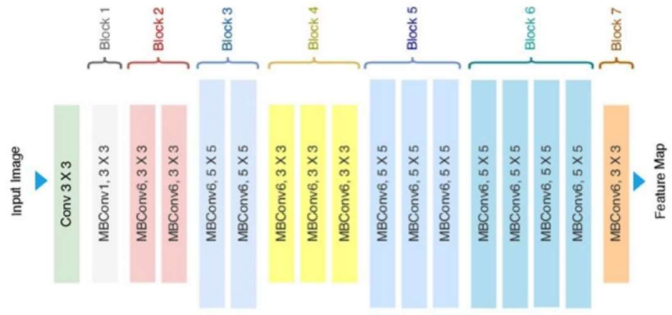


Figure 0C.7 Efficient Net Model Architecture

Task Objective:

AI Image Generation Detection:

Our system leverages advanced deep learning techniques to accurately detect AI-generated images (such as those produced by GANs, diffusion models, or other generative techniques) and distinguish them from authentic, human-created images. This involves:

• Feature Extraction: Utilizing high-capacity convolutional neural networks (e.g., EfficientNet, ResNet) to extract subtle statistical patterns and artifacts commonly found in AI-generated images.

• Multi-domain Training: Training on diverse datasets that include synthetic images from popular generative models like StyleGAN, StyleGAN2, EF3D and Stable Diffusion to ensure generalization across different sources.

• Fine-grained Classification: Implementing classifiers that can detect minute inconsistencies in textures, lighting, edges, and noise patterns that are often imperceptible to the human eye.

• Robust Performance: Evaluated across multiple benchmarks to ensure high precision and recall, even when images are post-processed, resized, or compressed.

Used Datasets:

• 140k Real and Fake Faces: Combines 70,000 real faces from the Flickr dataset with
70,000 fake faces generated by StyleGAN, resized to 256px.

• CelebA-HQ resized (256x256): Contains 30,000 high-quality celebrity faces, suitable for training and evaluating generative models.

• Synthetic Faces High Quality (SFHQ) Part 2: Includes 91,361 high-quality 1024x1024 curated face images, enhanced using StyleGAN2 techniques.

• Face Dataset Using Stable Diffusion v1.4: Comprises real faces from the Flickr dataset and fake faces generated by Stable Diffusion models, resized to 256px.

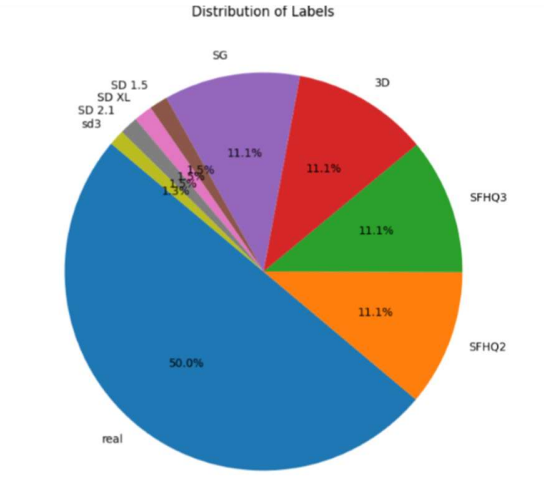
• Stable Diffusion Face Dataset: AI-generated human faces using Stable Diffusion 1.5, 2.1, and SDXL 1.0 checkpoints, covering resolutions of 512x512, 768x768, and
1024x1024.

• Synthetic Faces High Quality (SFHQ) Part 3: Contains 118,358 high-quality 1024x1024 face images generated by StyleGAN2, utilizing advanced truncation tricks.

• Synthetic Human Faces for 3D Reconstruction: Generated by drawing samples

from the EG3D model, resulting in high-quality 512x512 synthetic face images.

200K images					
100 k Real 100 k Fake					
Train		Validate		Test	
Real: 70 k	Fake: 70 k	Real: 15 k	Fake: 15 k	Real: 15 k	Fake: 15 k
70k Flickr	3k SD V2.1	15 k CelebA HQ	3k SDXL 1.0	15 k CelebA HQ	2536 SD 1.4
	3k SD V1.5		3k SG1		3116 SG1
	16k SG1		3k EG3D		3116 EG3D
	16k EG3d		16k S SG2		3116 S SG2
	16k S SG2		16k S SD1.4		3116 S SD1.4
	16k S SD1.4				
70% Data		15% Data		15% Data	



Dataset Overview:

SD 1.5

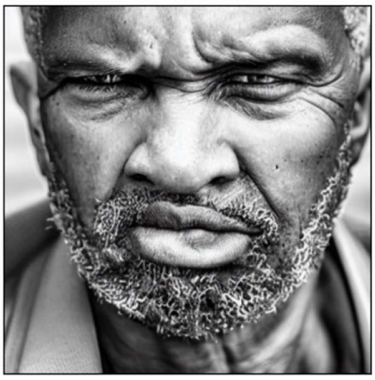
SD 1.5

SD XL

sd3

sd3

sd3



SFHQ2

SFHQ2

SFHQ2

SFHQ3

SFHQ3

SFHQ3

Model Selection:

Overview of Initial Model Candidates:

We initially evaluated several models for image detection, and we chose one based on the

reported performance for our dataset characteristics.

• **EfficientNet:** Chosen for its balance between accuracy and computational efficiency.

• **ResNet:** Evaluated for its strong performance in image classification.

• **Xception:** Considered for its capability to detect fine-grained image features.

Evaluation Criteria:

Models were evaluated based on the following criteria

• **Accuracy:** Overall classification performance.

• **Precision and Recall:** Balance between false positives and false negatives.

• **F1 Score:** Combined measure of precision and recall.

• **ROC-AUC:** Trade-off between true positive and false positive rates.

Experimental Setup:

Each model was trained using a standardized dataset split of 70% training, 15% validation,

and 15% testing, and models were trained on high-performance GPUs.

Results of Initial Models:

• **EfficientNet:** Balanced accuracy and computational efficiency, with an accuracy of
99%.

• **ResNet:** Achieved an accuracy of 99% but required more computational resources.

• **Xception:** Offered detailed feature extraction but was less efficient compared to EfficientNet.

Metrics	Accuracy	Precision	Recall	F1-Score
EfficientNetB2	94.3%	94.3%	94.2%	94.2%
EfficientNetB6	99.5%	99.5%	99.5%	99.5%
Xception	97%	98%	97%	98%
ResNet-50	98.2%	98.2%	98.2%	98.2%
VIT-224	96.7%	96.6%	96.7%	96.6%

Final Model Selection:

Based on the initial evaluations, the following refinements were made:

EfficientNet was selected for its optimal balance between accuracy and computational
efficiency, with additional data augmentation applied to improve generalization, Preferred
for its efficiency and high accuracy in distinguishing Real from AI-generated images.

Training Setup:

EfficientNet-B6 Model was trained with:

• Batch Size:16

• Learning rate: 5e-5

• Epochs: 2.5

Performance Metrics:

Classification Report:

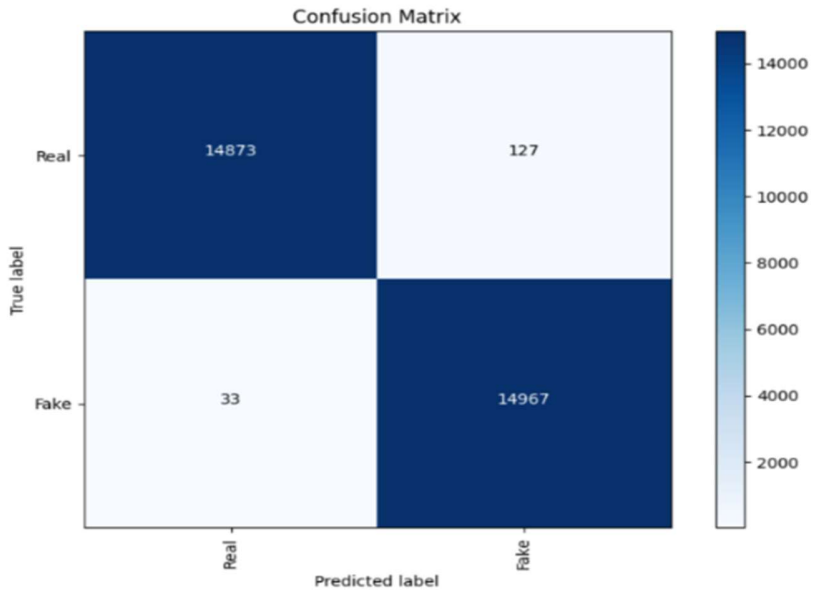
```
Classification report:

              precision    recall  f1-score   support

        Real     0.9978    0.9915    0.9946     15000
        Fake     0.9916    0.9978    0.9947     15000

    accuracy                         0.9947     30000
   macro avg     0.9947    0.9947    0.9947     30000
weighted avg     0.9947    0.9947    0.9947     30000
```

Confusion Matrix:



Chapter 2: Background

- This section summarizes some of the most relevant prior works and technologies referred to for this project, which focuses on generating high-quality synthetic face images using AI and detecting whether an image is AI-generated or real.
- Modern deep learning has enabled significant advancements in computer vision, particularly in generative models like GANs and diffusion models. These models are capable of synthesizing highly realistic images that are often indistinguishable from real photographs. In parallel, detection systems have emerged to help verify the authenticity of such content. Our project builds upon the combination of both directions—AI image generation and AI-generated image detection—leveraging cutting-edge libraries, frameworks, and datasets.

2.1 AI Image Generation

2.1.1 Stable Diffusion v1.5

- Stable Diffusion is a latent diffusion model that generates images from natural language prompts. Version 1.5 of this model is known for its ability to produce highly detailed and realistic visuals.

- **Latent Diffusion Models (LDMs):**
- Unlike pixel-space generation, LDMs operate in a compressed latent space, making image generation more efficient and controllable.
- **Text-to-Image Mapping:**
- The model takes in a prompt (e.g., "a smiling woman with curly hair") and uses cross-attention mechanisms to map textual features to image components.
- **Image Resolution:**
- Stable Diffusion v1.5 can generate high-resolution images, typically at **512×512 pixels** or higher.

2.1.2 LoRA (Low-Rank Adaptation)

- LoRA is a technique used for efficient fine-tuning of large models without retraining the entire architecture.
- **Efficiency:**
- LoRA introduces trainable rank-decomposed matrices into existing layers, significantly reducing GPU memory usage and training time.
- **Use in This Project:**
- We used LoRA to fine-tune Stable Diffusion v1.5 on the **Flickr-Faces-HQ (FFHQ)** dataset, enhancing facial realism and detail rendering in generated images.

2.1.3 FFHQ Dataset

- Developed by NVIDIA, the **Flickr-Faces-HQ (FFHQ)** dataset contains high-quality images of human faces across various ages, ethnicities, and conditions.
- **Image Quality:**
- 1024×1024 resolution with consistent lighting and background.
- **Importance:**
- Serves as a foundation for training face generation models due to its diversity and clarity.

2.2 AI Image Detection

2.2.1 DeepFake Detection Challenge (Facebook AI)

- A large-scale competition launched to evaluate the robustness of deepfake detection systems.
- **Goal:**
- Develop models that can distinguish manipulated media (especially facial content) from real footage.
- **Benchmark Impact:**

- Set the standard for datasets, evaluation metrics, and performance in AI-based detection systems.

2.2.2 "Do GANs Leave Artifacts?" (Yu et al.)

- This research explores whether AI-generated images leave detectable traces.
- **Key Insight:**
- GAN-generated images often exhibit statistical artifacts, particularly in frequency domains or texture consistency.
- **Use in Detection:**
- Such artifacts are leveraged by CNN-based classifiers to identify synthetic images.

2.2.3 Detection Models (CNNs, EfficientNet, ResNet)

- Our system uses deep convolutional neural networks like **EfficientNet** and **ResNet** for feature extraction and classification.
- **Feature Learning:**
- The models are trained on both real and AI-generated images (from StyleGAN, Stable Diffusion, etc.) to capture fine inconsistencies in:
    - Textures
    - Edges

- Lighting
- Noise patterns

- **Performance:**
- The model is designed to be robust, maintaining high accuracy even when images are resized, compressed, or lightly edited.

- ### *2.2.4 Multi-Domain Training*

- The detection model is trained on a variety of image sources to ensure it generalizes across different types of AI-generated faces.
- **Sources Used:**
  - StyleGAN
  - StyleGAN2
  - EF3D
  - Stable Diffusion
  - Real-world datasets (e.g., VISION, Dresden Image Database)

Chapter 3: System Design

AI VS AI

System Features

Software Requirements Specification (SRS)

1 Functional Requirements

- User registration, login, and profile management.
- Image upload, generation, and detection modules.
- Admin panel with stats, user list, and access control.
- Export results to PDF.
- Google OAuth integration.

2 Non-Functional Requirements

- Performance: Real-time processing.
- Scalability: Batch processing and multithreaded architecture.
- Security: OAuth, password reset, admin roles.
- Compliance: GDPR adherence.

3 External Interfaces

- Web API (RESTful) for interaction between frontend and backend.
- Model server for detection/generation tasks.
- File storage for images and results.

2. API Documentation

Login Register APIs

Register - User Registration Description

This API is used to register a new user into the system.

Method:

POST URL: http://127.0.0.1:5000/register

Body (JSON):

{"name": "John Doe", "email": "johndoe@gmail.com", "password": "password123"}

Verify-code - Code Verification Method

POST URL: http://127.0.0.1:5000/verify-code

Body (JSON):

{"email": "johndoe@gmail.com", "code": "123456"}

Resend-verification-code - Resend Verification Code Method

POST URL: http://127.0.0.1:5000/resend-verification-code

Body (JSON):

{"email": "johndoe@gmail.com"}

Login - User Login Method

POST URL: http://127.0.0.1:5000/login

Body (JSON):

{"email": "johndoe@gmail.com", "password": "password123"}

Forgot-password - Forgot Password Request Method

POST URL: http://127.0.0.1:5000/forgot-password

Body (JSON):

{"email": "johndoe@gmail.com"}

Verify-reset-code - Verify Password Reset Code Method

POST URL: http://127.0.0.1:5000/verify-reset-code

Body (JSON):

{"email": "johndoe@gmail.com", "reset_code": "123456"}

Reset-password - Reset Password Method

POST URL: http://127.0.0.1:5000/reset-password

Body (JSON):

{"email": "johndoe@gmail.com", "new_password": "newpassword123"}

Auth/google - Google Login Method

GET URL: http://127.0.0.1:5000/auth/google

Auth/google/callback - Google OAuth Callback Method

GET URL: http://127.0.0.1:5000/auth/google/callback

Logout - User Logout Method

POST URL: http://127.0.0.1:5000/logout

Admin APIs

Users - Get All Users Method

GET URL: http://127.0.0.1:5000/users

Admins - Get All Admins Method

GET URL: http://127.0.0.1:5000/Admins

Admin/promote-to-admin - Promote a User to Admin Method

POST URL: http://127.0.0.1:5000/admin/promote-to-admin

Body (JSON):

{"email": "user@example.com"}

Admin/demote-to-user - Demote an Admin to Regular User Method

POST URL: http://127.0.0.1:5000/admin/demote-to-user

Body (JSON):

{"email": "user@example.com"}

System-stats - Get System Stats Method

GET URL: http://127.0.0.1:5000/system-stats

Delete-user-by-email - Delete a User by Email Method

DELETE URL: http://127.0.0.1:5000/delete-user-by-email

Body (JSON):

{"email": "user@example.com"}

Detection APIs

Upload-image-local - Upload a Local Image and Detect Content Method

POST URL: http://127.0.0.1:5000/upload-image-local

User-detection-images - Get User's Detected Images (Base64) Method

GET URL: http://127.0.0.1:5000/user-detection-images

Delete-detection-image - Delete Detection Image by ID Method

DELETE URL: http://127.0.0.1:5000/delete-detection-image

Body (JSON):

{"image_id": 5}

Delete-all-detection-images - Delete All Detection Images Method

DELETE URL: http://127.0.0.1:5000/delete-all-detection-images

Generation APIs

Generate-image - Generate AI Image from Description Method

POST URL: http://127.0.0.1:5000/generate-image

Body (JSON):

{"description": "A futuristic city at night"}

User-generated-images - Get All User Generation Images Method

GET URL: http://127.0.0.1:5000/user-generated-images

Delete-generation-image - Delete Generation Image by ID Method

DELETE URL: http://127.0.0.1:5000/delete-generation-image

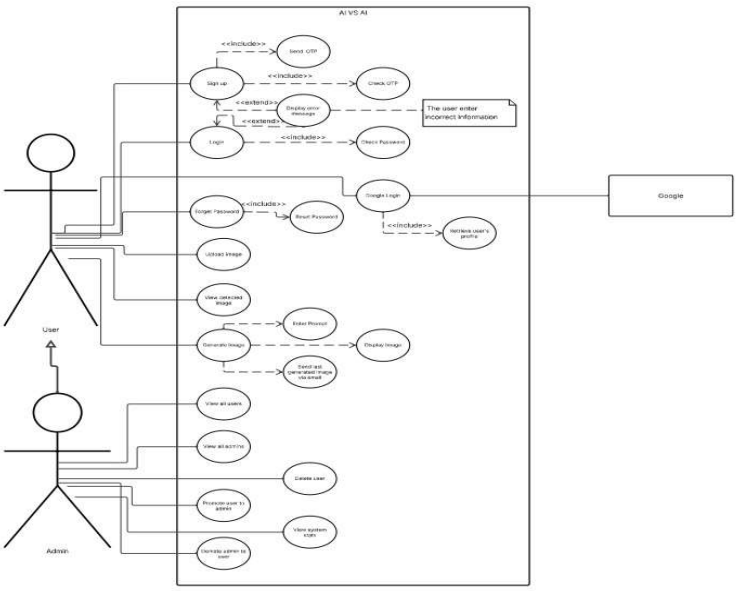
Body (JSON):

{"image_id": 7}

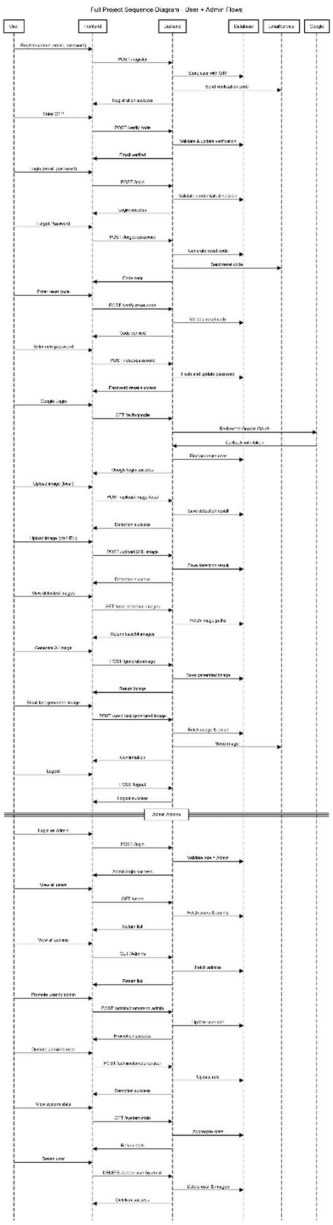
Delete-all-generation-images - Delete All Generation Images Method

DELETE URL: http://127.0.0.1:5000/delete-all-generation-images

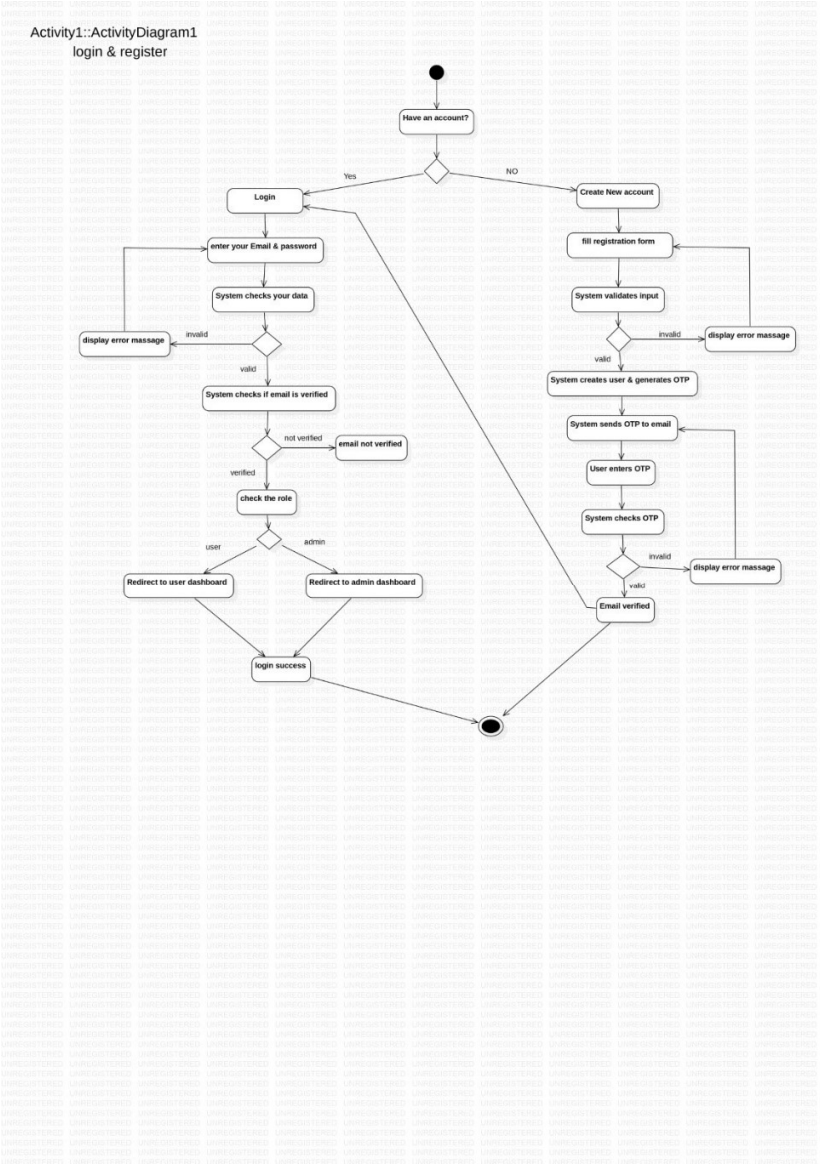
Usecase Diagram



Sequence Diagram

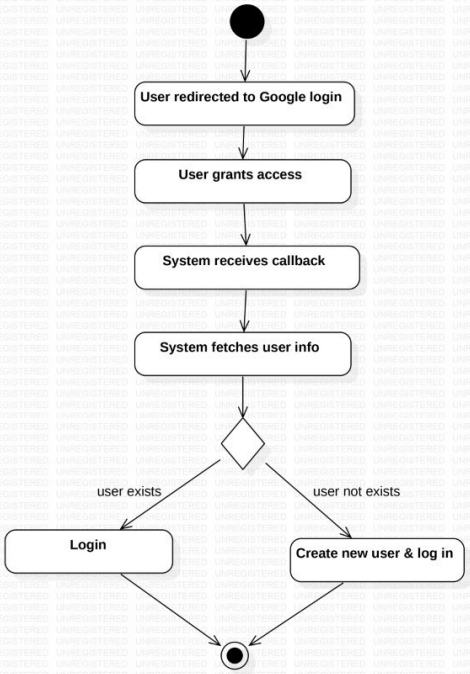


Activity diagram



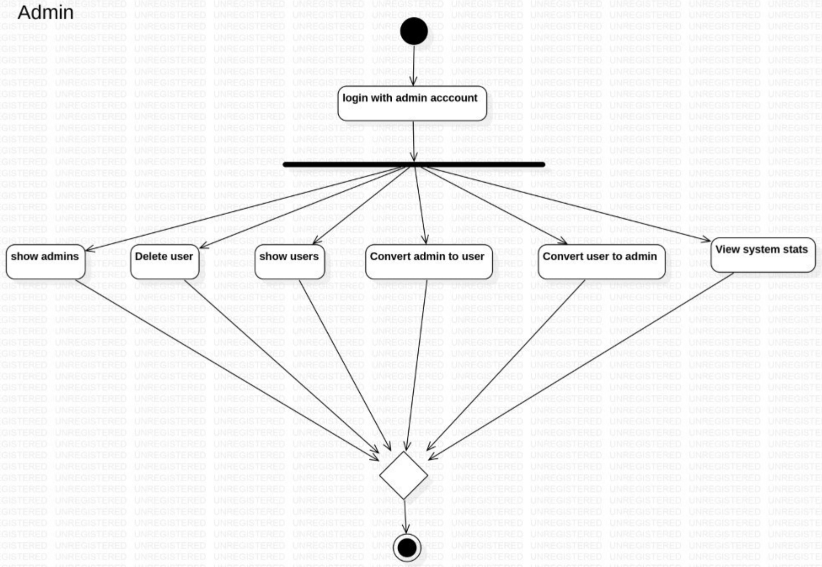
Activity1::ActivityDiagram1

Sign in with Google



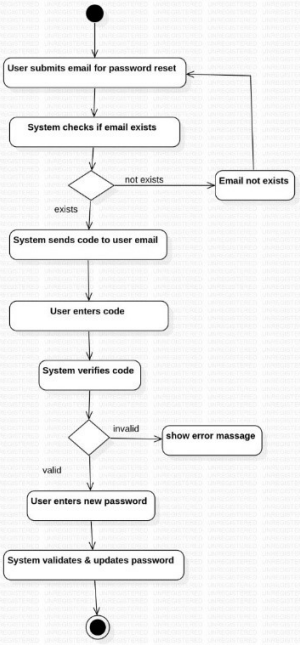
Activity1::ActivityDiagram1

Admin



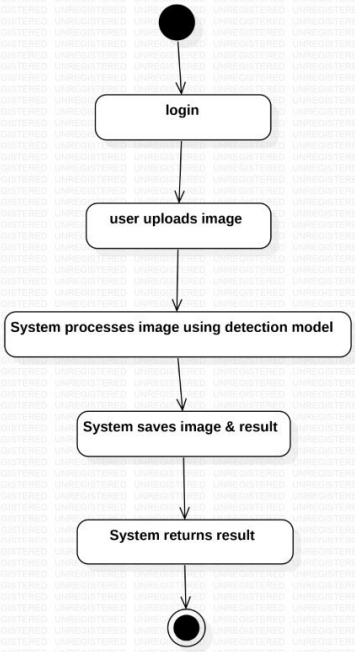
Activity1::ActivityDiagram1

Forgot Password



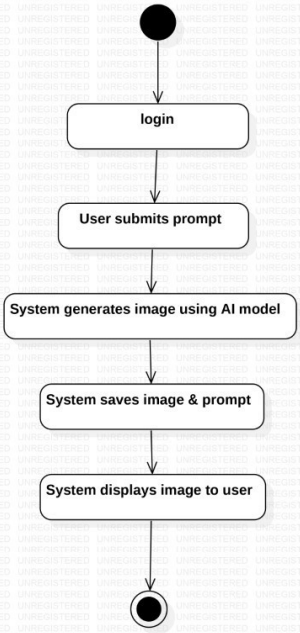
Activity1::ActivityDiagram1

Image Detection

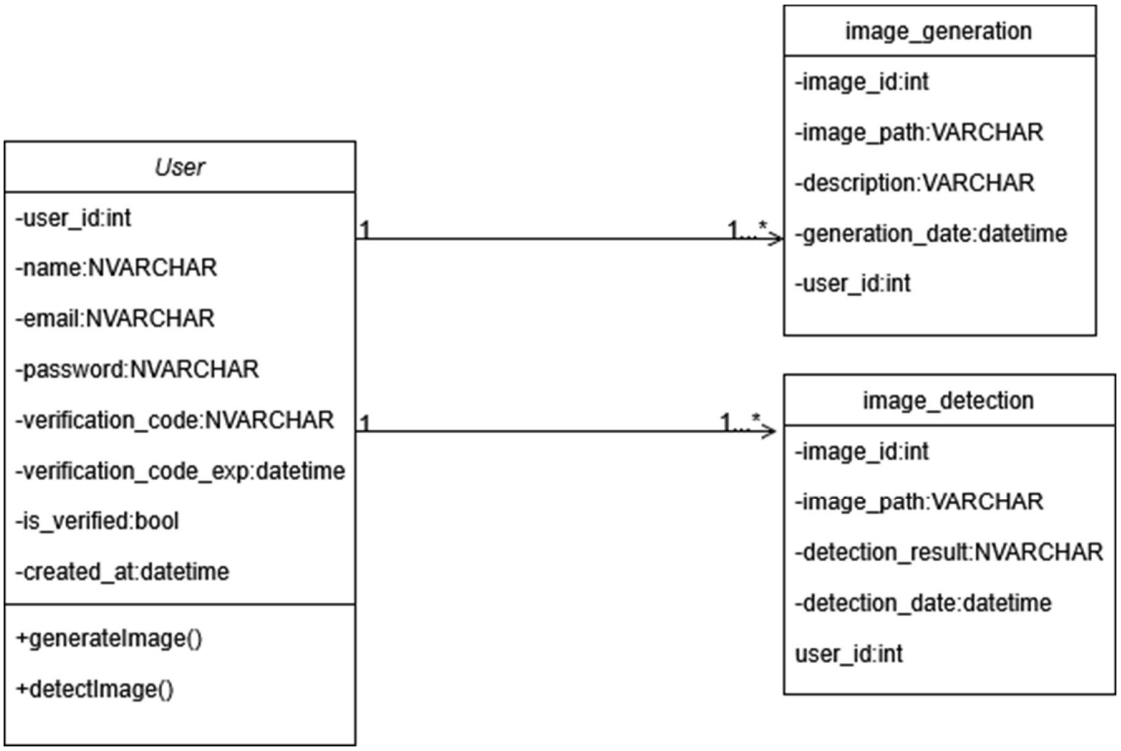


Activity1::ActivityDiagram1

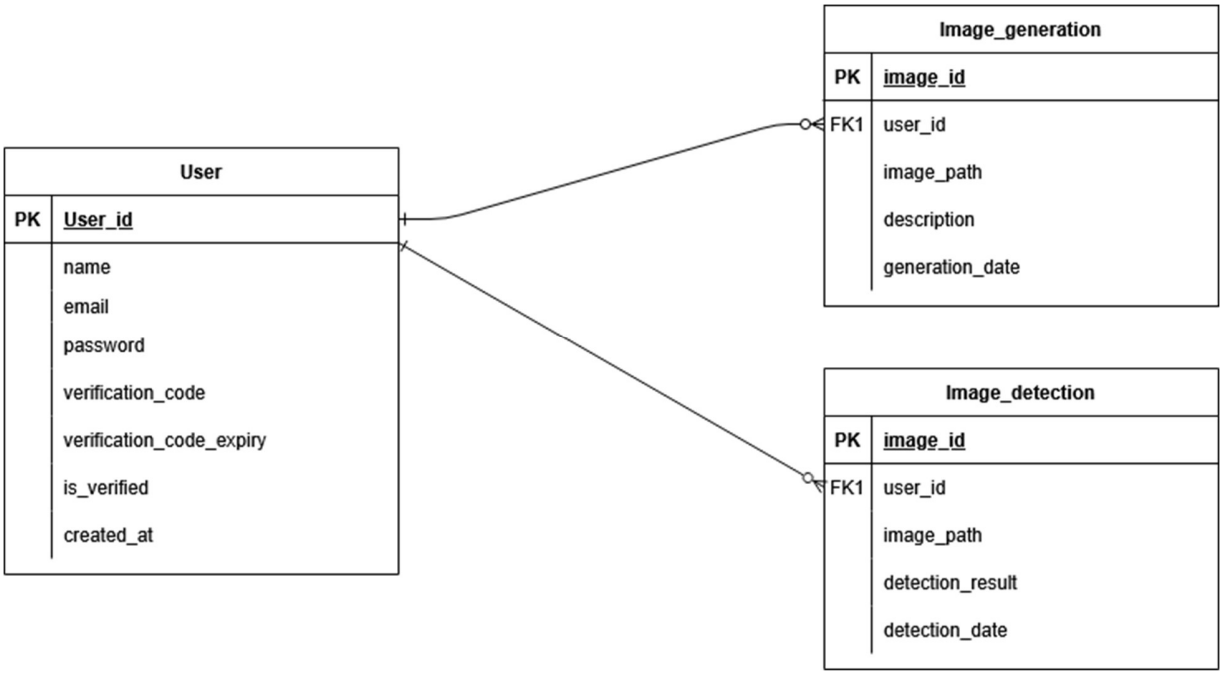
Image Generation



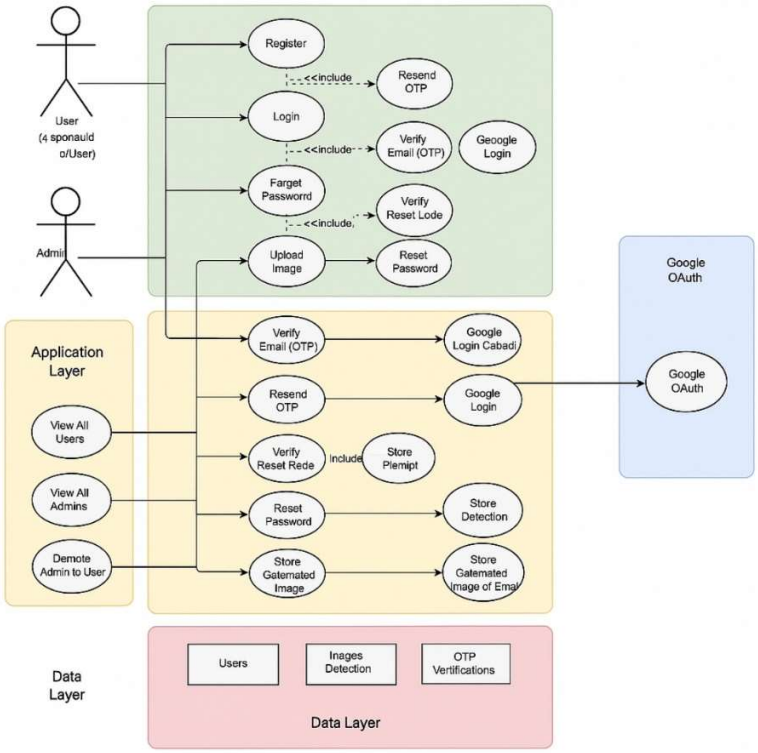
Class Diagram



ERD Diagram


User
PK
User_id
name
email
password
verification_code
verification_code_expiry
is_verified
created_at
Image_generation
PK
image_id
FK1
user_id
image_path
description
generation_date
Image_detection
PK
image_id
FK1
user_id
image_path
detection_result
detection_date

System Architecture



Chapter 4: System Implementation and results

Overview on Implementation steps

This section outlines the major steps taken during the development of the system, starting from data preparation and model fine-tuning, to image generation and final AI image detection evaluation.

4.1 Implementation Steps:

1. **Dataset Preparation**
    a. Collected and resized FFHQ images
    b. Generated text descriptions using LLaVA
2. **Model Fine-Tuning**
    a. Applied LoRA to Stable Diffusion v1.5
    b. Configured attention and projection layers
    c. Trained model over 30 epochs
3. **Image Generation**
    a. Developed inference pipeline
    b. Generated realistic face images from user prompts
4. **Detection System**
    a. Trained CNN classifier to distinguish AI-generated images
    b. Evaluated detection accuracy on test set
5. **Frontend Integration**
    a. Created input form for text prompts
    b. Show results with real-time feedback

Figure 10: Task Management Board for AI Image System

4.2 Future Work

While the current system successfully allows users to generate realistic human faces using fine-tuned Stable Diffusion and detect whether face images are AI-generated or real, there are several directions in which this project can be extended and improved.

1 Improving Detection Accuracy

Future versions of the system can explore more advanced architectures such as **Vision Transformers (ViT)** or **Swin Transformers** to enhance the precision and recall of detecting AI-generated images. These models have shown high performance in image classification tasks and may help capture more complex features and patterns left by synthetic content.

2 Cross-Domain Detection

To ensure the system can generalize to newer models, the detection module can be trained on an even wider variety of AI-generated images from emerging tools like **Midjourney**, **DALL·E 3**, and **RunwayML**. This would improve its ability to recognize images generated by unseen architectures.

3 Explainable Output

Adding explainability features such as **heatmaps**, **highlighted regions**, or **confidence score visualizations** can help users understand why an image was classified as real or AI-generated. This would increase trust and transparency, especially in sensitive use cases.

4 Real-Time and Bulk Processing

In future versions, real-time detection and **batch uploading** can be introduced to allow faster processing of multiple images simultaneously. This would benefit users who need to verify large numbers of images quickly, such as media platforms or research teams.

5 Active Learning from User Feedback

Integrating user feedback (e.g., marking false positives or false negatives) could enable **active learning**, where the model improves continuously based on real-world use and corrections.

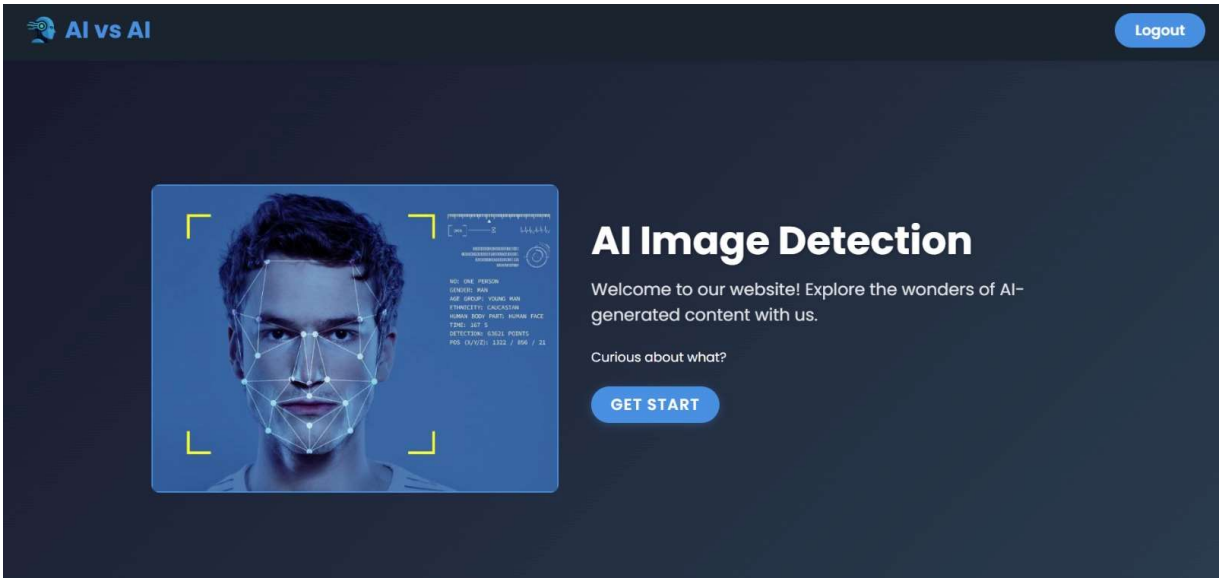
6 Mobile and Cross-Platform Support

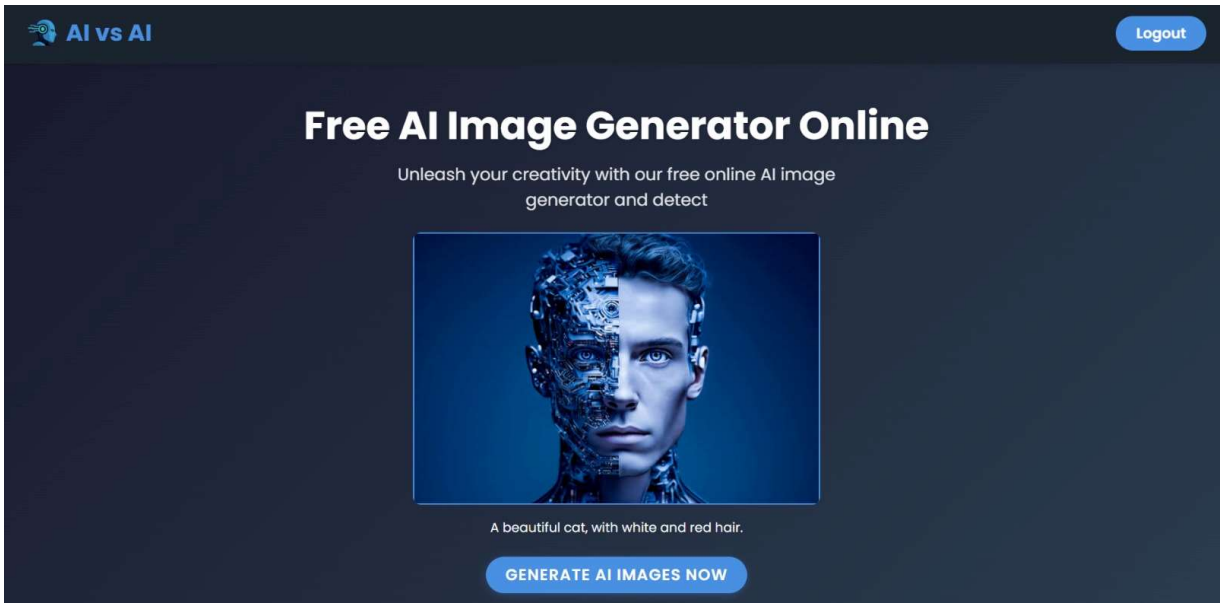
Deploying the system as a **mobile app** or **cross-platform web application** would increase its accessibility and usability in real-world settings, including field verification or content moderation workflows.

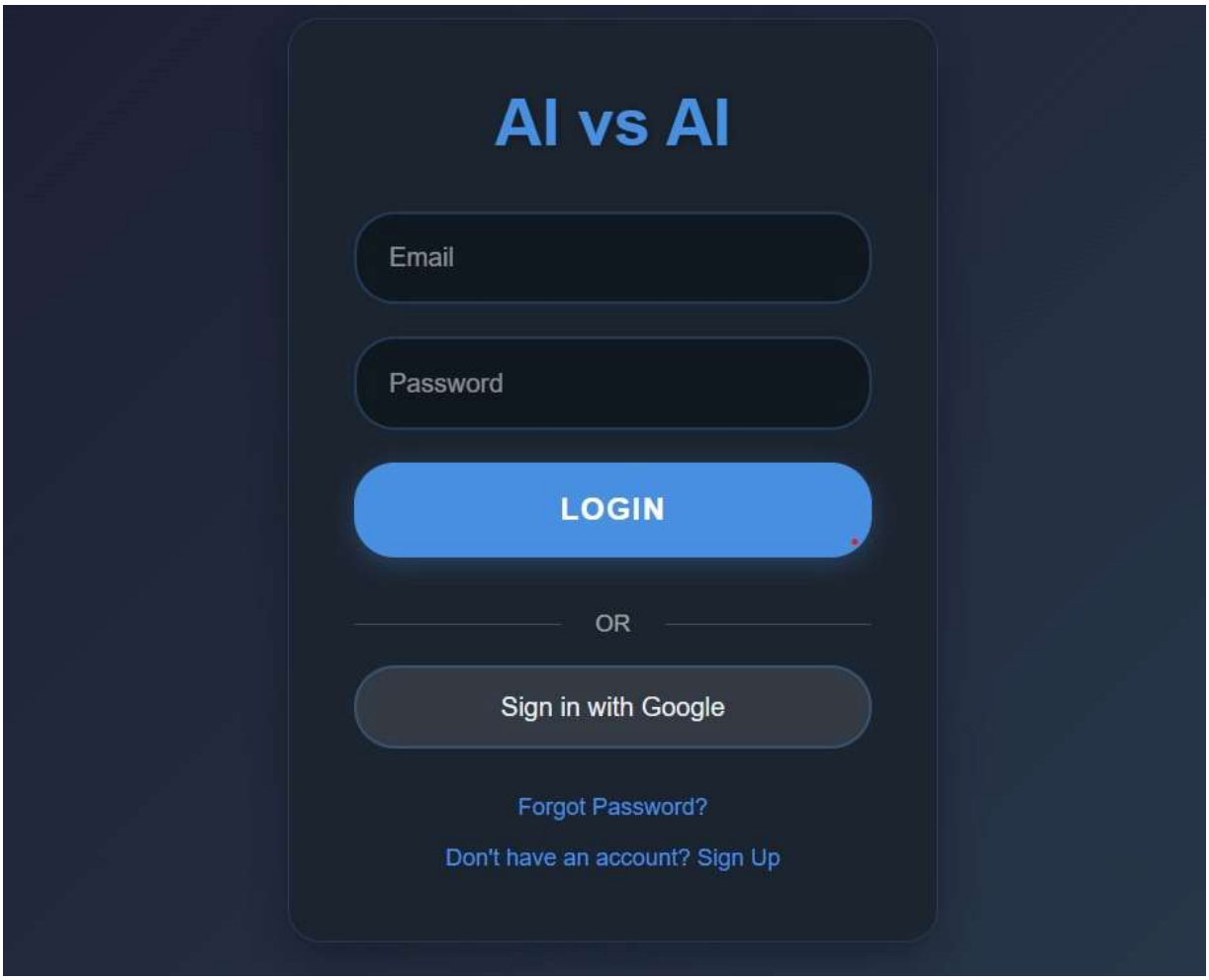
7 Deployment as an API

The system could also be turned into a **public RESTful API**, allowing third-party developers to integrate face generation and detection into their own apps or websites.

Screenshots:






AI vs AI
Email
Password
LOGIN
OR
Sign in with Google
Forgot Password?
Don't have an account? Sign Up

Email Verification

We've sent a verification code to your email address. Please enter the code below to verify your account.

Email Address

Enter your email

Verification Code

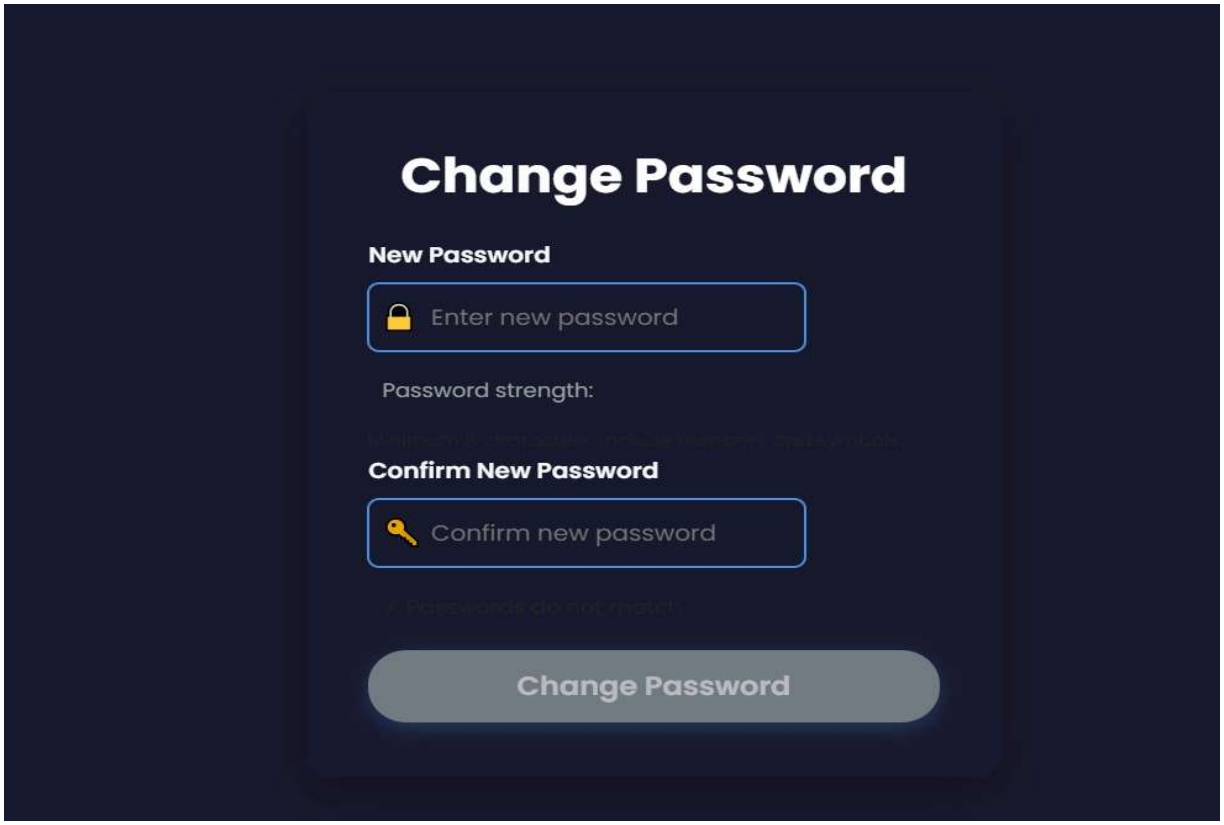
ENTER 6-DIGIT CODE

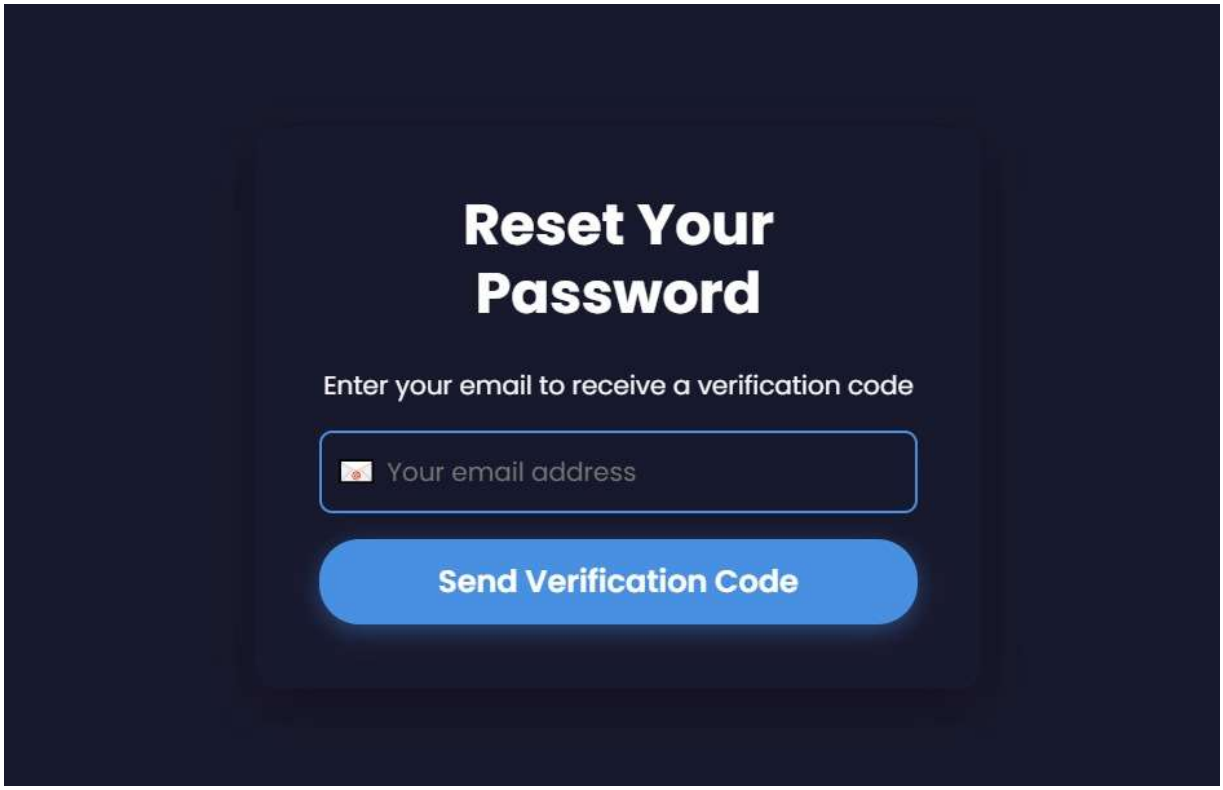
VERIFY EMAIL

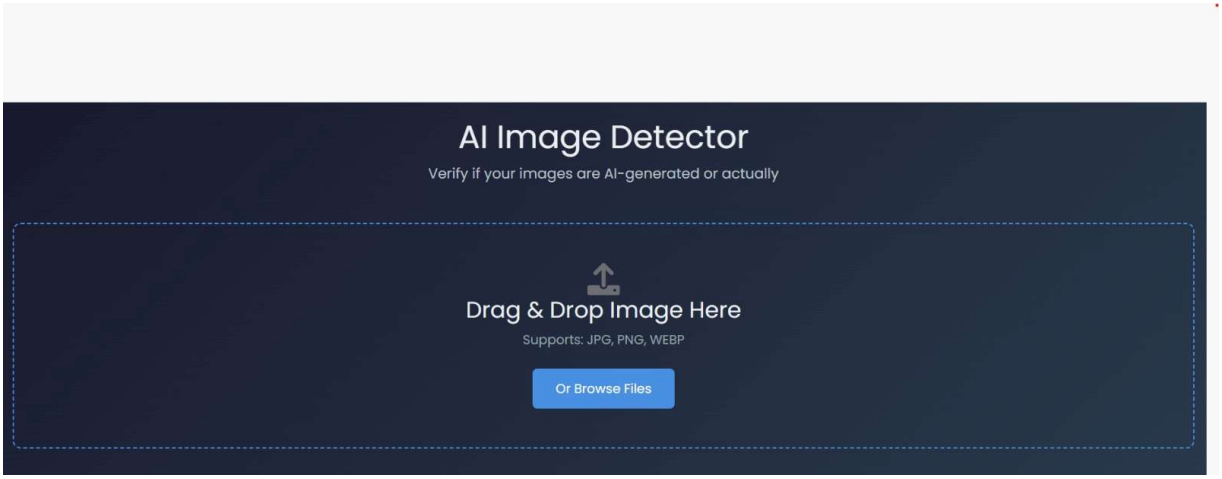
Code Sent

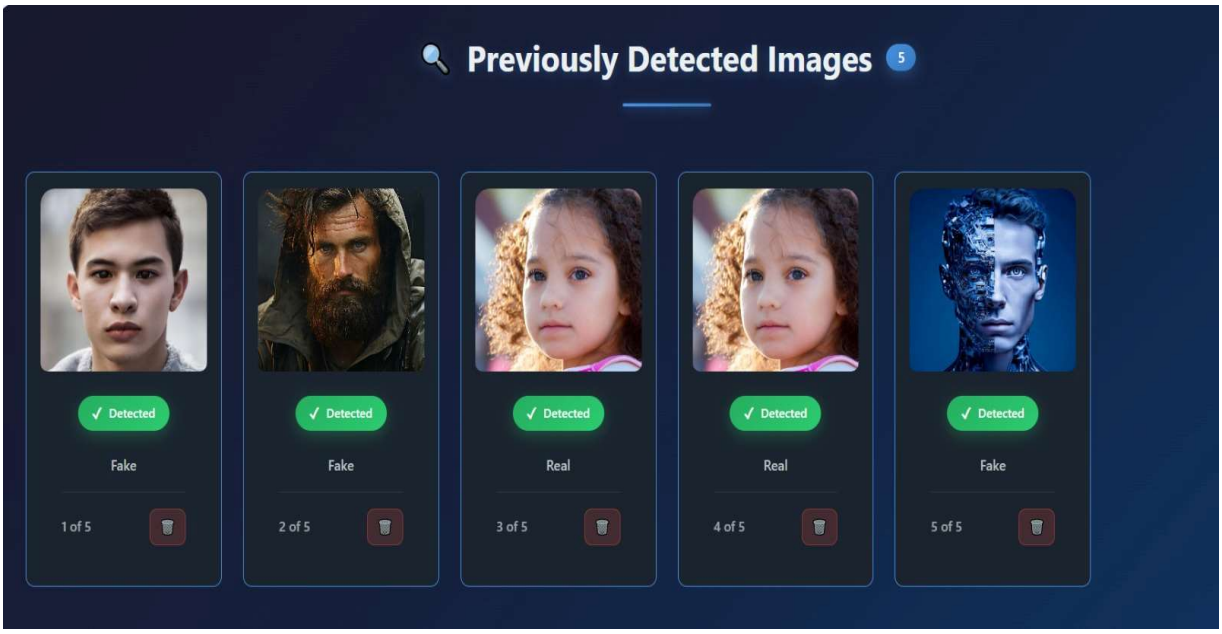
Resend available in 170 seconds

← Back to Register


Change Password
New Password
Enter new password
Password strength:
Confirm New Password
Confirm new password
Change Password

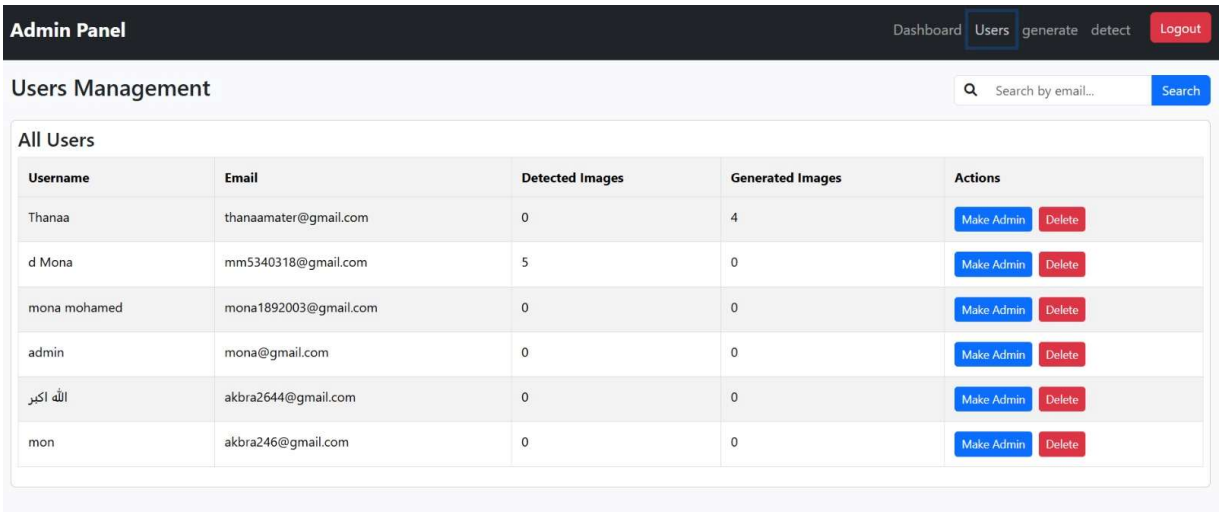

Reset Your Password
Enter your email to receive a verification code
Your email address
Send Verification Code

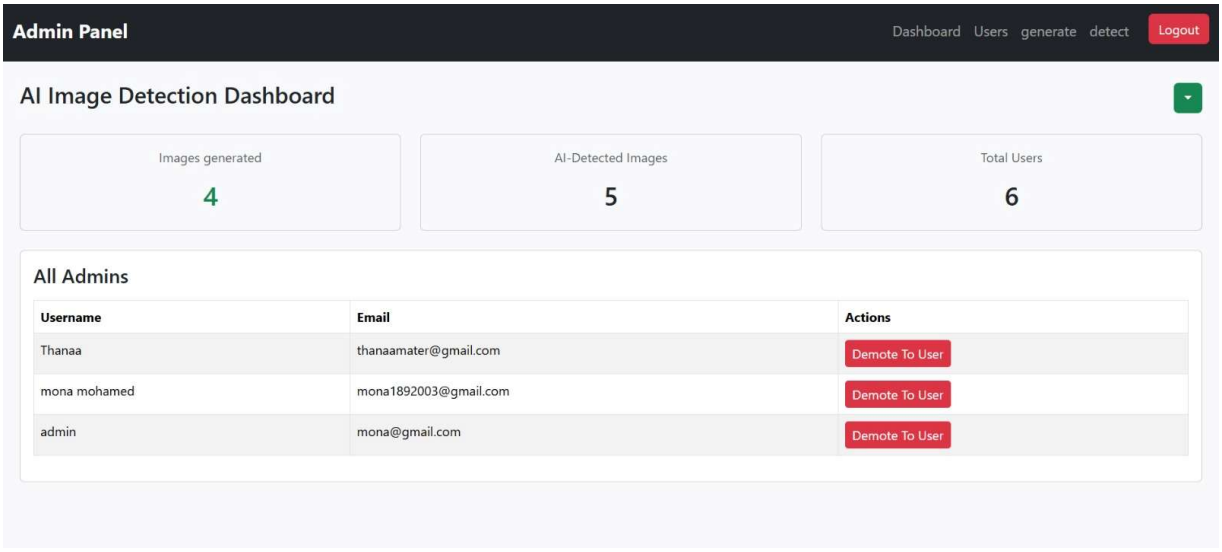

AI Image Detector
Verify if your images are AI-generated or actually
Drag & Drop Image Here
Supports: JPG, PNG, WEBP
Or Browse Files


Previously Detected Images
5
Detected
Fake
1 of 5
Detected
Fake
2 of 5
Detected
Real
3 of 5
Detected
Real
4 of 5
Detected
Fake
5 of 5


Free AI Image Generator Online
Unleash your creativity with our free online AI image generator
Describe the image you want to generate...
Generate AI Image


Your Generated Masterpieces
4
Clear All
PROMPT:
PROMPT:
PROMPT:


Admin Panel
Dashboard
Users
generate
detect
Logout
Users Management
Search by email...
Search
All Users
Username
Email
Detected Images
Generated Images
Actions
Thanaa
thanaamater@gmail.com
0
4
Make Admin
Delete
d Mona
mm5340318@gmail.com
5
0
Make Admin
Delete
mona mohamed
mona1892003@gmail.com
0
0
Make Admin
Delete
admin
mona@gmail.com
0
0
Make Admin
Delete
الله اكبر
akbra2644@gmail.com
0
0
Make Admin
Delete
mon
akbra246@gmail.com
0
0
Make Admin
Delete


Admin Panel
Dashboard
Users
generate
detect
Logout
AI Image Detection Dashboard
Images generated
4
AI-Detected Images
5
Total Users
6
All Admins
Username
Email
Actions
Thanaa
thanaamater@gmail.com
Demote To User
mona mohamed
mona1892003@gmail.com
Demote To User
admin
mona@gmail.com
Demote To User

Generation Results (A & B):

Before & After LoRA

Without LoRA

With LoRA

Without LoRA

With LoRA

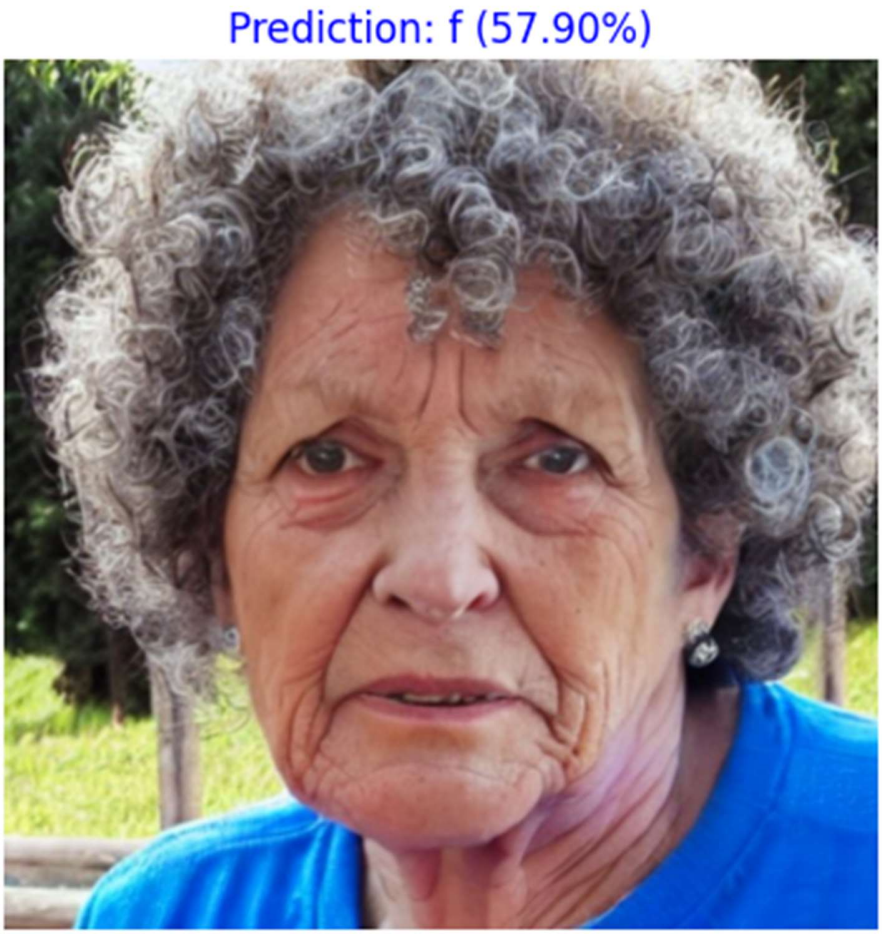
Without LoRA

With LoRA

Without LoRA

With LoRA

Detection Results (F & R):



F→ 57.90 % generated by AI so it's Fake

Prediction: r (66.76%)

R→ 66.76% real image so it's Real

Bibliography

[1] https://angular.io/docs
[2] https://flask.palletsprojects.com/en/latest/
[3] https://werkzeug.palletsprojects.com/en/latest/
[4] https://testdriven.io/blog/angular-flask/
[5] https://angular.io/guide/http
[6] https://flask.palletsprojects.com/en/latest/patterns/fileuploads/
[7] https://flask.palletsprojects.com/en/latest/quickstart/#sessions
[8] https://angular.io/guide/forms-overview
[9] https://medium.com/swlh/file-upload-in-angular-4a0880d0be23
[10] https://docs.microsoft.com/en-us/dotnet/api/system.speech.recognition?view=netframework- 4.8
[11] https://huggingface.co/docs/diffusers/index
[12] https://huggingface.co/docs/peft/index
[13] https://pytorch.org/docs/stable/index.html
[14] https://flask.palletsprojects.com/en/latest/
[15] https://docs.python.org/3/library/smtplib.html
[16] https://docs.python.org/3/library/email.examples.html
[17] https://keras.io/api/applications/efficientnet/
[18] https://keras.io/api/applications/xception/
[19] https://github.com/mkleehammer/pyodbc/wiki