DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY - IST

Project Report on

AI Image Identifier

in

Society Oriented Project– IT23505

Submitted by

Tharanesh –2023115053

Mohammed Nabeel Ahmed– 2023115008

Dasari Chinna Tirumanideep – 2023115134

AI IMAGE IDENTIFIER

1. Introduction

With the rapid advancement of artificial intelligence, the creation of AI-generated images has become increasingly realistic and difficult to distinguish from real photographs. This has led to challenges in maintaining authenticity, digital trust, and ethical transparency across online platforms. This system aims to address these issues through the development of an **AI Image Identifier**, which automatically classifies whether an image is AI-generated or real, while also providing visual explanations for its predictions.

The system uses a client-server architecture built on the MVC pattern. The frontend (View) is implemented using **Streamlit**, offering a clean and interactive user interface where users can upload images, view classification results, Grad-CAM heatmaps, metadata details, and interact with an integrated chatbot. The backend (Controller) is developed using **Python-based APIs** that handle image processing, manage model inference, and coordinate communication between the frontend and backend components, while the Model layer encapsulates the **ResNet-based deep learning classifier**, Grad-CAM visualization logic, and database operations for secure user authentication and data storage.

Additionally, an **AI-powered chatbot (TinyLlama)** is integrated into the system to enhance user interaction and understanding. It helps users interpret Grad-CAM visualizations, understand model predictions, and form independent conclusions — ensuring responsible and transparent use of AI.

2. Problem Statement

In recent times, AI-generated images have become increasingly realistic and widespread, causing significant challenges such as:

- Difficulty in distinguishing real from fake images. Difficulty in managing and tracking adoption records.
- Increased risk of misinformation, scams, and misuse in various fields, such as job applications and news reporting.
- Loss of trust in digital content due to unverified sources.

- Growing demand for reliable systems to detect AI-generated images.

- A need for mechanisms to preserve the authenticity and integrity of online media.

Hence, there is an urgent need for reliable systems that can accurately distinguish real images from AI-generated ones, ensuring the trustworthiness of digital content.

3. Objectives

1. To develop a system for identifying whether an image is real or AI-generated.

2. To extract and analyze EXIF metadata for clues regarding the image's authenticity.

3. To use heatmap visualization to highlight areas potentially manipulated by AI.

4. To provide an easy-to-use web interface for users to upload and analyze images.

5. To ensure accurate detection methods to maintain trust in digital content.

6. To create a reliable platform that empowers users to distinguish between real and AI-generated images effectively.

4. Methodology

The development follows the **MVC (Model-View-Controller)** pattern:

- **Model:** Represents the machine learning and data processing components. It includes the ResNet18/ResNet50-based binary classifier, the Grad-CAM visualization module, and the PostgreSQL database for storing user credentials and uploaded images

- **View:** Implemented using Streamlit for an interactive and user-friendly interface. It manages image uploads, displays AI/Real predictions, Grad-CAM heatmaps, metadata details, and provides access to the integrated chatbot for assistance.

- **Controller:** Python-based scripts handle user requests, process uploaded images through the model, manage database operations, and communicate results to the frontend. The chatbot controller connects to the TinyLlama model for generating intelligent, context-aware responses.

Development Process (SDLC - Agile/Iterative Hybrid):

1. Requirement Analysis

2. System Design (Architecture, Data Flow)

3. Implementation (Frontend + Backend)

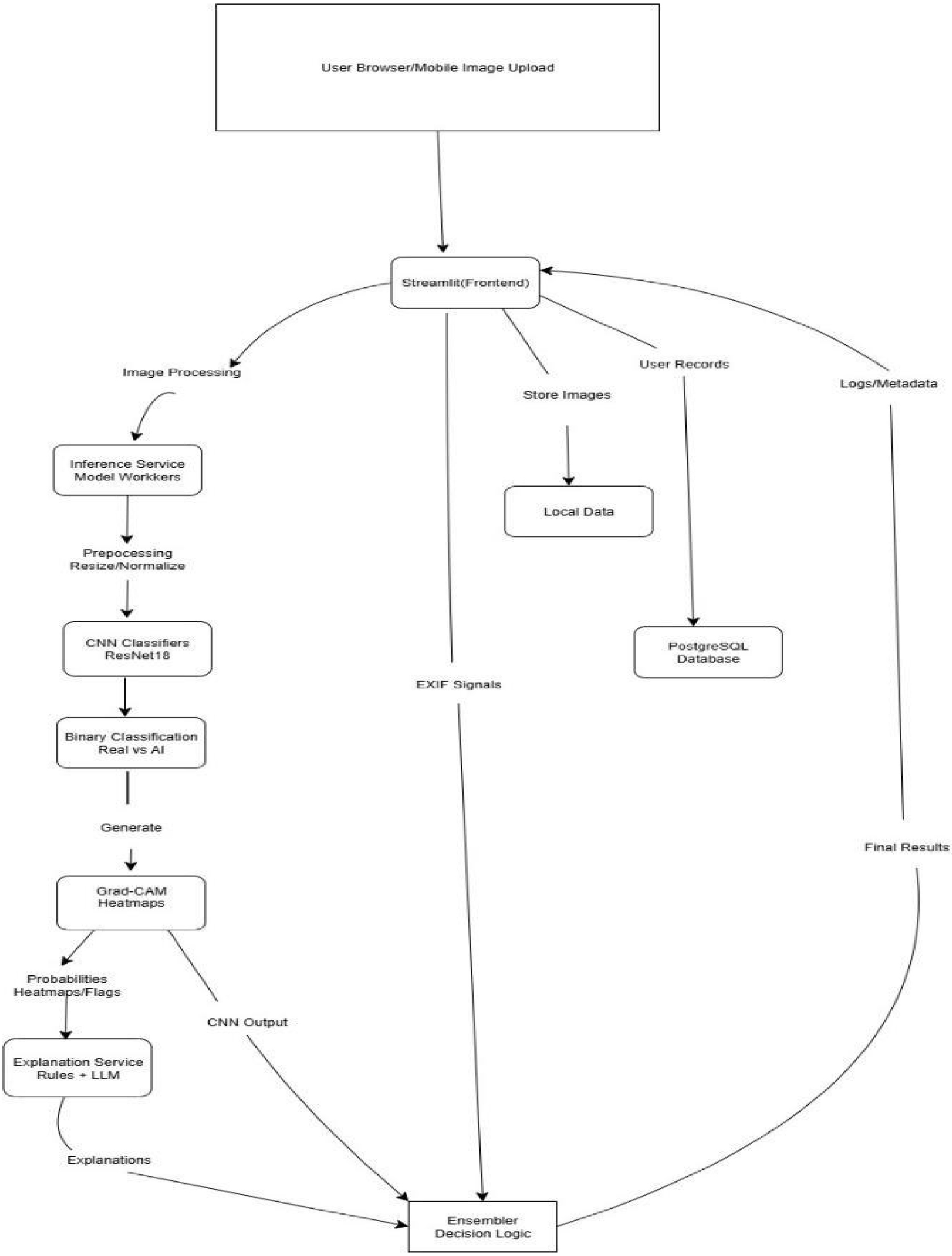
4. Integration (Model+Database+Chatbot)

5. Testing (Unit & Functional)

6. Deployment

5. SYSTEM ARCHITECTURE

Architecture Diagram:



6. Tools and Technologies Used

This project follows the Model–View–Controller (MVC) architecture using:

- **Model:** Implemented using **PyTorch** for deep learning model training and inference (ResNet architecture), and **PostgreSQL** for user authentication and data storage.
- **View**: Developed using **Streamlit**, providing an interactive web interface for users to upload images, visualize Grad-CAM results, and interact with the integrated chatbot.
- **Controller**: Built using **Python APIs** to connect the frontend (Streamlit) with the model backend, handle data flow, and manage prediction logic and database interactions

Layer	**Technology Used**
Frontend (View)	Streamlit,HTML/CSS, Python
Backend (Controller)	Python, REST-like function calls,Torch-based inference
Model & Database	PyTorch(ResNet18), Grad-CAM, PostgreSQL
AI Chatbot	TinyLlama(via Hugging Face Transformers)
Tools & Libraries	PyTorch, OpenCV, Pillow, Pandas,NumPy,Transformers, psycopg2
Version Control	Git & GitHub
Development Environment	VS Code, Streamlit CLI

7. CONCLUSION

The **AI Image Identifier** System provides a reliable and transparent platform for distinguishing between AI-generated and real images. Using the MVC architecture, the application separates model inference, user interaction, and control logic, ensuring modularity, scalability, and ease of maintenance.

With its integration of Streamlit, Python-based controllers, and a ResNet-powered classifier, the system offers an intelligent, interactive, and explainable environment for users. It not only enhances digital authenticity and transparency but also helps promote trust in online visual content through metadata analysis and visual pattern recognition..