**FORGERY IMAGE DETECTION SYSTEM**

**ABSTRACT**

This project is about detecting whether an image is real or edited. We use three methods together to make the detection more accurate. First, a deep learning model checks if the image is tampered. Then, Grad-CAM highlights the areas where the model thinks editing has happened. Next, SIFT finds important keypoints in the image to understand its structure. Finally, a Copy-Move detection method checks if any part of the image has been copied and pasted somewhere else. By combining these techniques, our system can detect different types of forgery such as splicing, copy-move, and region editing. The project provides clear visual results like heatmaps, keypoints, and matched regions. It works through a simple Streamlit interface for easy use. This helps users quickly identify manipulated images in a reliable and understandable way.

**TEAM MEMBERS**

|  |  |
| --- | --- |
| Name | Register Number |
| ABRAHAM JUSTIN | 23MIA1027 |
| VENKATARAMAN R | 23MIA1025 |
| SUDARSHAN MANIKANDAN | 23MIA1078 |

**BASE PAPER REFERENCE**

**Title:** *An Effective Image Copy-Move Forgery Detection Using Entropy Information*  
**Authors:** *Li Jiang, Zhaowei Lu*  
**Publisher:** IEEE  
**Year:** *2024*



**TOOLS AND LIBRARIES USED**

* Python 3.8+
* PyTorch
* OpenCV
* NumPy
* Matplotlib
* Streamlit
* scikit-image
* torchvision
* SIFT / ORB keypoint extractors
* Machine Learning Model (EfficientNet-B0)
* Grad-CAM visualization
* glob, os, uuid libraries

**STEPS TO EXECUTE THE CODE**

**A) Streamlit App**

1. Open terminal
2. Navigate to project folder
3. cd project/src
4. Run Streamlit
5. streamlit run app.py
6. Upload an image
7. The system displays:
   * Original image
   * Grad-CAM heatmap
   * SIFT keypoint visualization
   * Copy-Move detection result
   * Summary panel

**B) Console Version**

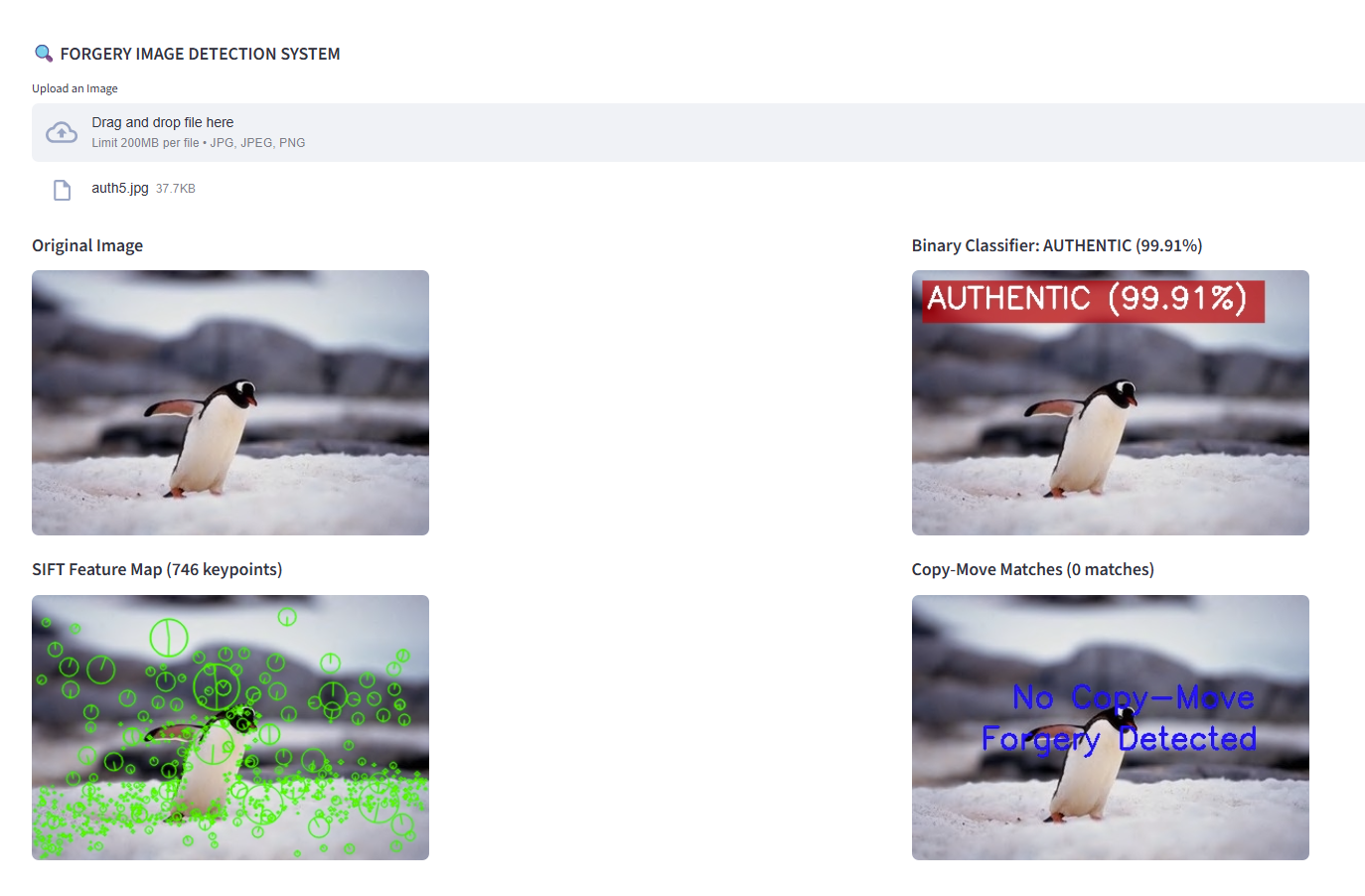
1. Navigate to src/
2. Run:
3. python main\_integrated.py
4. Enter image path
5. All results are saved inside results/ folder.

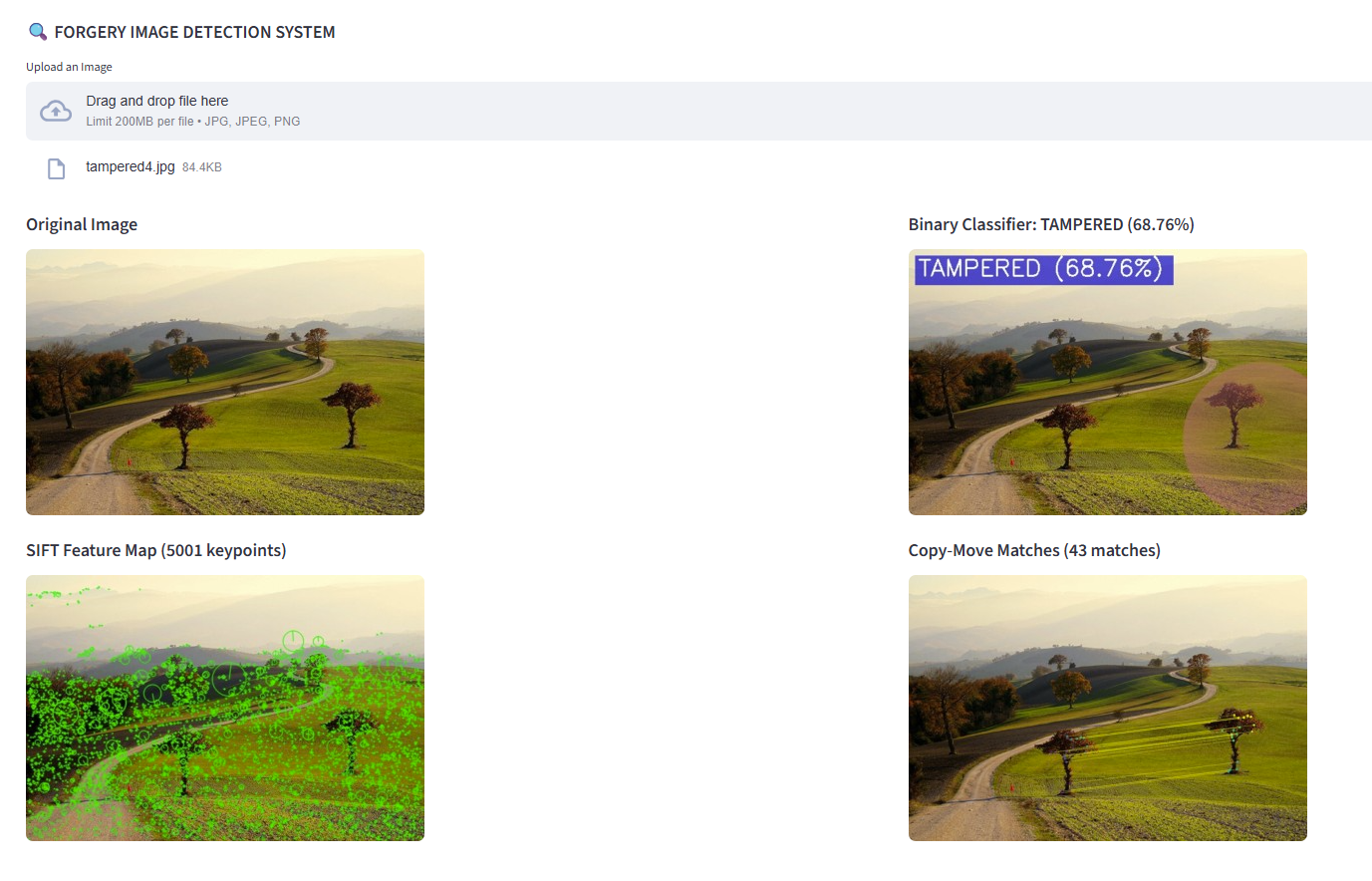
**DESCRIPTION OF DATASET**

The dataset used for training includes **tampered and authentic images** collected from publicly available sources such as CASIA v2.0 Dataset. These datasets contain images manipulated through splicing, copy-move, object removal, and region editing. Each image is labeled as either *tampered* or *authentic*, enabling supervised training of the binary classifier. The dataset includes a variety of image categories (indoor, outdoor, objects, people) ensuring generalization across diverse manipulation types.

**OUTPUT SCREENSHOTS**

**The system provides the following outputs:**





**YouTube Demo Link**

**YouTube Demo:** [*https://youtu.be/H1L5Xe34UgU*](https://youtu.be/H1L5Xe34UgU)