

Riphah International University, Islamabad

Faculty of Computing

Subject: Data Structures



Contributor	Name	SAP ID
1	Muhammad Sadeem Choudhary	66385
2	Muhammad Yousaf	66160

Project Proposal:

AI-Based Fake Image Detection: Identifying AI-Generated and Manipulated Images

1. Introduction / Background:

In today's digital world, fake and AI-generated images are spreading rapidly across online platforms, often misleading users and spreading misinformation. With the rise of deepfakes, GAN-generated content, and

image editing tools (e.g., Photoshop), it has become increasingly difficult to differentiate between real and manipulated images by human inspection alone. This project proposes to build an AI-based fake image detection system that integrates metadata analysis, forensic methods, reverse image search, and a deep learning model (CNN) to classify images as real or fake.

2. Problem Statement:

The rise of fake and AI-generated images poses a serious challenge to digital trust and security. Without an automated detection system, it becomes increasingly difficult to verify the authenticity of images shared on social media, news platforms, and communication channels. of images in fields like healthcare, surveillance, and digital content creation.

3. Objectives:

Define and classify “fake images” (AI-generated or digitally manipulated).

Develop detection modules using:

- Metadata analysis (EXIF data, editing traces).
- Reverse image search (checking originality & online presence).
- Forensic methods (Error Level Analysis, pixel noise).
- Deep learning models (CNNs trained on real vs fake datasets).

Integrate these modules into a single decision-making pipeline.

Provide a simple Google Colab/Gradio interface for testing images.

4. **Proposed Solution:**

To solve the problem of fake and AI-generated images, we propose a hybrid detection system that integrates three complementary approaches:

1. **Metadata Analysis**

- Extract EXIF data (camera model, date, editing software, GPS, etc.).
- Suspicious cases: missing camera info, editing traces (e.g., Photoshop tags), or incomplete metadata.

2. **Reverse Image Search**

- Connect with TinEye or Bing Visual Search API.
- Check if the uploaded image exists online and compare earliest sources to verify originality.

3. **Image Forensics**

- Apply Error Level Analysis (ELA) to highlight regions of possible tampering.
- Use noise/resampling analysis to detect compression artifacts caused by manipulation.

4. **Deep Learning Detection (CNN-based)**

- Train a Convolutional Neural Network (e.g., EfficientNet or XceptionNet) on Kaggle datasets (*Real and Fake Face Detection, CIFAKE*).
- The model outputs the probability of the image being **real** or **fake**.

5. **Decision Fusion**

- Combine all modules (metadata, reverse search, forensics, CNN output).
- Generate a final Fake Score with supporting evidence.

6. User Interface (Colab/Gradio)

- Provide a simple interface where users can upload an image.
- The system will analyze it and return:
 - Prediction (Real/Fake)
 - Probability score
 - Explanation (metadata flags, forensic visual, CNN prediction).

5. Methodology:

- **Data Collection** – Acquire real vs fake image datasets from Kaggle (e.g., Real and Fake Face Detection, CIFAKE).
- **Metadata Analysis** – Extract EXIF data (camera info, editing software, GPS tags) and flag anomalies.
- **Reverse Image Search** – Use TinEye/Bing API to check image provenance online.
- **Forensic Analysis** – Apply ELA and pixel-noise analysis to detect manipulation traces.
- **Model Building** – Train a CNN (EfficientNet/XceptionNet) on real vs fake datasets.
- **Model Evaluation** – Assess model performance using accuracy, precision, recall, and F1-score.

- **Decision Fusion & Visualization** – Combine module results, display Fake Score, and visualize analysis outputs in Colab/Gradio.

6. Tools and Technologies

- Programming Language: Python
- Libraries: PyTorch/TensorFlow, OpenCV, ExifTool, scikit-learn, Gradio
- Environment: Google Colab (with GPU support)
- External Services: TinEye/Bing Visual Search API (optional for reverse image search)

7. Timeline:

Week 1: Literature review, dataset collection, setup in Colab

Week 2: Metadata extraction and reverse image search integration

Week 3: Forensic analysis implementation (ELA, noise detection)

Week 4: CNN model training and evaluation

Week 5: Decision fusion and interface development (Gradio in Colab)

Week 6: Testing, optimization, documentation, and presentation preparation.

8. References

- Kaggle – deepfake-vs-real Dataset:
<https://www.kaggle.com/datasets/prithivsakthiur/deepfake-vs-real-60k/data>
- Kaggle – CIFAKE (Real and Synthetic Images):
<https://www.kaggle.com/datasets/joeychan2000/cifake-real-and-ai-generated-synthetic-images>

- FaceForensics++ Dataset:
<https://github.com/ondyari/FaceForensics>