PROJECT ABSTRACT

Deepfake Image Detection using Deep Learning

A forensic AI system that distinguishes real photos from AI-manipulated faces using a CNN with frequency-domain features. The web app gives a real/fake verdict with confidence and Grad-CAM heatmaps showing suspicious regions — a cutting-edge cybersecurity-AI project examiners love.

01 Title

Deepfake Image Detection using Deep Learning

02 Introduction / Background

Photorealistic face synthesis is now in everyone's hands: swapped identities, reenacted expressions and synthetic portraits flood social media within minutes. These deepfakes drive election misinformation, non-consensual imagery and financial fraud, eroding trust in visual evidence itself. Human analysts increasingly fail to spot them, so the defence must be computational: detectors that learn the statistical traces generative models leave behind.

03 Problem Statement

Most detectors inspect only pixels and explain nothing: they miss the frequency-domain upsampling artifacts generative models leave behind, and give examiners nothing to interrogate. A useful detector must flag manipulated faces across forgery methods and show which regions drove each decision, so every verdict stands up before examiners.

04 Objectives

- Build a dual-stream CNN: spatial features plus frequency artifacts.
- Train and evaluate on FaceForensics++ across four forgery methods.
- Output a calibrated real/fake verdict per face.
- Explain each verdict with Grad-CAM heatmaps.
- Ship the FakeSpot web app: image, video-frame and batch modes.
- Document accuracy, precision, recall, F1 and AUC.

05 Proposed System / Working

Input: an image, video frames, or a batch of files uploaded to the FakeSpot Flask app. Processing: OpenCV with MTCNN/Dlib detects, aligns and crops faces to 299×299. Two streams run in parallel — a fine-tuned Xception/EfficientNet-style CNN learns spatial traces (blending boundaries, texture inconsistencies, unnatural geometry) while an FFT/DCT stream captures frequency-domain upsampling artifacts invisible to pixel networks; both fuse before the final classifier. Training uses the FaceForensics++ c23/high-quality splits with logged loss/accuracy curves.

Decision and output: the model returns a calibrated real/fake verdict with a confidence score, and Grad-CAM back-projects the decision as a heatmap over the manipulated regions. Video mode scores extracted frames and aggregates a video-level verdict; batch mode exports results to CSV, and a REST-style endpoint serves programmatic inference. Weights ship as a .pth checkpoint — no GPU needed for demos.

06 Technologies / Components Used

Python · PyTorch · OpenCV · MTCNN/Dlib · NumPy · Flask · FaceForensics++ dataset · Grad-CAM

07 Key Features

- Real/fake classification with calibrated confidence score.
- Dual-stream detection: spatial CNN plus frequency (DCT/FFT) artifacts.
- Grad-CAM heatmaps of manipulated regions (eyes, mouth, blending boundaries).
- Automatic face detection, cropping and alignment.
- FakeSpot web app with batch mode, CSV export and REST endpoint.
- Frame-extraction mode for scoring deepfake videos.

08 Applications / Use Cases

- Social-media misinformation triage and newsroom verification desks.
- Forensic research on synthetic media.
- Cybercrime units flagging non-consensual imagery; identity-verification pipelines — always as a screening aid, never certified forensic evidence.

09 Future Scope

- Temporal video models (3D CNN / LSTM) for full deepfake-video scoring.
- Audio-deepfake detection fused with the visual verdict.
- Browser extension for on-the-fly image verification.
- Adversarial robustness testing against anti-forensic attacks.
- Real-time social-media monitoring pipeline for viral content.

10 Conclusion

This project delivers a reproducible dual-stream architecture, benchmark evaluation with explainability, and an end-to-end demonstrator. Fusing spatial and frequency features improves accuracy, and Grad-CAM turns every verdict into interrogable evidence.