PFDNET: A Deep Learning Approach for Robust Shared Photo Authentication and Tamper Recovery

JEEVA.D [1], MS.G.MAHALAKSHMI[2]
Student [1], Assistant Professor [2]
Master of Computer Applications [1, 2]
Dr.M.G.R. Educational and Research Institute, Chennai [1, 2]

***Abstract*—A deep learning-based framework called the Photo Forgery Detection Network (PFDNet) was created to combat image tampering by providing lossless recovery in addition to detection. To ensure content consistency, it incorporates a Cyber Vaccinator module that uses the original image and edge map to create an immunized image version. The Invertible Neural Network-based Forgery Detector module uses a forward pass to identify tampered areas and a backward pass to retrieve the original data. By contrasting the original and restored images, Run-Length Encoding (RLE) verifies the recovery. The shortcomings of conventional techniques are successfully addressed by PFDNet, which excels at processing low-resolution or compressed images while guaranteeing robustness, authenticity, and high fidelity in digital image integrity on online platforms.**

I. INTRODUCTION

In order to identify, safeguard, and retrieve tampered images from social media, this project creates PFDNet, a deep learning-based system. It consists of a Lossless Recovery module that uses Run-Length Encoding (RLE) for image restoration and validation, a Cyber Vaccinator for embedding tamper-resistant perturbations, and a Forgery Detection module for localizing altered regions. By alerting users of tampering and successful recovery, the system guarantees platform independence, high accuracy, and user trust. Technically, it makes use of datasets such as CASIA and Columbia, incorporates hashing and LTSA for forgery resilience, and employs CNN-transformer hybrids or GANs for detection and recovery. For maintaining image integrity and authenticity in digital and social media contexts, PFDNet provides a complete, real-time solution.

To insert images in Word, position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with ―Float over text‖ unchecked)(keep text wrapping top-bottom).

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY reserves the right to do the final formatting of your paper.

II. OBJECTIVE

Create PFDNet to safeguard the authenticity of images.

Include modules for Lossless Recovery, Forgery Detection, and Cyber Vaccinator.

Put in place a cyber vaccine that introduces subtle, unchangeable disruptions.

Use PFDNet to identify and pinpoint areasof tampered images.

Use Run-Length Encoding (RLE) for verification and a Lossless Recovery module to restore modified areas.

III. SCOPE

The project scope encompasses all essential components to ensure digital image integrity. It includes image protection and authentication against unauthorized modifications and the Cyber Vaccinator module to embed invisible, tamper-resistant perturbations. PFDNet detects manipulations such as splicing or copy-move, while the Lossless Recovery module, validated by Run-Length Encoding (RLE), restores altered images with high accuracy. The system integrates detection, protection, and recovery into a seamless pipeline, supports platform independence, and ensures compatibility across web and social media platforms. User notifications provide transparency by alerting tampering and recovery outcomes.

Overall, the project prioritizes precision, quality, and trust, offering a robust, user-centric solution for image authenticity verification in digital environments.

MATH

Let I∈RH×W×CI \in \mathbb{R}^{H \times W \times C}I∈RH×W×C represent an RGB image of height HHH, width WWW, and C=3C = 3C=3 channels. The goal is twofold:

Tampering Detection: Learn a function fθ:RH×W×C→{0,1}H×Wf_{\theta} : \mathbb{R}^{H \times W \times C} \rightarrow \{0,1\}^{H \times W}fθ:RH×W×C→{0,1}H×W such that:

fθ(I)ij=1f_{\theta}(I)_{ij} = 1fθ(I)ij=1 if pixel (i,j)(i,j)(i,j) is tampered

fθ(I)ij=0f_{\theta}(I)_{ij} = 0fθ(I)ij=0 otherwise

Tampering Recovery: Learn a function gϕ:RH×W×C→RH×W×Cg_{\phi} : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H \times W \times C}gϕ :RH×W×C→RH×W×C, reconstructing the original content from the tampered image:

I^recovered=gϕ(I)\hat{I}_{\text{recovered}} = g_{\phi}(I)I^recovered=gϕ(I)

where

I^recovered≈Ioriginal\hat{I}_{\text{recovered}} \approx I_{\text{original}}I^recovered≈Ioriginal

IV. REVIEW OF LITERATURE

were used in this study to identify forgeries in compressed and low-resolution images. Although it did not aid in tamper recovery, it did improve tamper localization. This is further enhanced by PFDNet, which incorporates recovery mechanisms.

Park, J., and T. Nguyen (2024) Preventive Protection of Digital Images through Cyber Vaccination presented the idea of incorporating imperceptible disturbances into pictures to prevent manipulation, which forms the basis of the Cyber Vaccinator module in PFDNet. Instead of detection or recovery, it concentrated on tamper resistance.

Zafar, S., and N. Iqbal (2024) DeepTamper: A Comprehensive Deep Learning Model for Image Repair and Forgery Detection created a unified system that uses inpainting to detect tampering and recover partial images. But there was no lossless restoration. PFDNet uses RLE validation and invertible networks to close this gap.

Thomas, J., and V. Reddy (2025) PFDNet: Sturdy Deep Learning-Based Shared Photo Authentication and Recovery This is the main study that suggests PFDNet, which combines Forgery Detection, Lossless Recovery (RLE), and a Cyber Vaccinator to offer a comprehensive solution. It is particularly well-suited for social media content integrity because it achieves high precision in tamper localization and lossless recoveryV. HELPFUL HINTS

SYSTEM CONFIGURATION

A strong hardware configuration is necessary to deploy an image tamper detection and recovery system such as PFDNet. Effective parallel processing is ensured by a minimum quad-core processor (Intel i5/i7 or AMD Ryzen 5/7) with a base speed of 2.5 GHz and a boost speed of up to 4.0 GHz. An octa-core processor is advised for demanding tasks. Although 16 GB guarantees more fluid performance during training or multitasking, at least 8 GB of DDR4 RAM is needed. In image processing, latency is decreased by fast memory speeds. While a 512 GB SSD is better for large datasets, a 256 GB SSD is advised for quick data access, model saving, and real-time detection. Backups can be stored on a secondary HDD. Efficiency, stability, and speed are guaranteed with this setup.

To guarantee smooth operation, the image tamper detection and recovery system combines strong frontend, backend, and deep learning tools. Python 3.6 or later is needed for the system, which runs on Ubuntu or Windows 10/11. Bootstrap facilitates responsive front-end design, while Flask manages back-end development. MySQL oversees the database, and WampServer offers a local Apache server for deployment. Effective image and numerical operations are made possible by Python libraries such as NumPy, Pandas, and OpenCV. Deep learning for detection and recovery is supported by PyTorch and TensorFlow. Machine learning tasks are handled by Scikit-learn, and visualizations are handled by Matplotlib and Seaborn. Pillow facilitates image processing. High performance, flexibility, and cross-platform functionality are guaranteed in both development and deployment environments with this software stack.

PYTHON

Python 3.8 is a high-level, object-oriented programming language that is renowned for being

easy to understand and use. It integrates with libraries such as Flask and TensorFlow and supports dynamic typing. Python boosts productivity and speeds up prototyping, making it perfect for AI, data analysis, and web development. It is appropriate for deep learning and real-time tamper detection systems due to its adaptability and vast ecosystem.

TENSORFLOW

Google's open-source machine learning platform TensorFlow is perfect for deep learning applications like tamper detection and image classification. It provides TensorFlow Lite for mobile use, TensorBoard for visualization, and CPU/GPU support. Building strong neural networks is made easier with high-level APIs like Keras. TensorFlow ensures effective training and inference by powering key models in image forgery detection and recovery.

KERAS

Keras is a high-level API based on TensorFlow that makes it simple to create and train deep learning models. Layers, optimizers, and metrics are among the crucial elements it supports. For tasks like image reconstruction and tamper detection, Keras makes it possible to quickly prototype neural networks. It is ideal for creating intricate models like PFDNet because of its adaptability, multi-backend compatibility, and clear syntax.

PANDAS

Pandas is a Python package for manipulating data with DataFrame and Series structures. It makes data filtering, grouping, and analysis easier. Pandas aids in the management of training logs, datasets, and performance metrics in tamper detection systems. It simplifies data processing and analysis during the model development and evaluation stages by supporting reading and writing from formats like CSV and Excel.

NUMPY

NumPy is a fundamental Python library for scientific computing that provides mathematical functions and multi-dimensional arrays. For image preprocessing operations like resizing and normalization, it is essential. NumPy facilitates quick numerical calculations and tensor operations in tamper detection systems. It facilitates effective processing during model training and feature extraction from image datasets by integrating with TensorFlow and OpenCV.

MATPLOTLIB

A robust library for producing both static and interactive visualizations is Matplotlib. Plotting training curves, confusion matrices, and tamper detection results are among its many applications. It facilitates the visualization of system behavior and model performance when combined with NumPy and Pandas. Because of its customizable features, it is perfect for debugging and plotting of publication quality, improving interpretability and feedback while developing a system.

SCIKIT-LEARN

A Python package called Scikit-learn is used for common machine learning tasks like evaluation, classification, and clustering. By providing techniques like feature selection and cross-validation, as well as metrics like precision and F1-score, it enhances deep learning tools. It supports system robustness and statistical validation in tamper detection by assisting in performance evaluation and comparison with baseline models.

PILLOW

A Python imaging library called Pillow is used to open, modify, and store images. It supports resizing, cropping, and filtering in addition to PNG and JPEG formats. Pillow manages image input/output in image tamper detection, assisting in preprocessing data prior to supplying it to models. It is a flexible image processing tool because it works with NumPy and OpenCV.

OPENCV

A computer vision library called OpenCV provides tools for analyzing images and videos. It carries out preprocessing operations like filtering, edge detection, and noise reduction. OpenCV overlays detection results and prepares images for model input in tamper detection systems. It ensures visual clarity and computational efficiency in the detection pipeline by supporting GPU acceleration and real-time processing.

MYSQL

Structured data can be stored using MySQL, a relational database management system. User information, image metadata, detection logs, and outputs are all stored in this system. MySQL

guarantees safe, scalable, and effective data management through integration with Flask and WampServer. By arranging user interactions and tamper evidence in a trustworthy, query-friendly environment, it preserves system integrity.

V. CONCLUSION

Because forgery threats are on the rise in the digital age, it is crucial to ensure the authenticity of shared images. By fusing deep learning with cutting-edge security features, PFDNet (Protection and Forgery Detection Network) provides a reliable solution. It has a cyber vaccine model that incorporates undetectable information into pictures for recovery and verification later. It enables self-recovery of tampered regions and detects manipulations such as copy-move and splicing using Invertible Neural Networks and Discrete Cosine Transform. Visual fidelity is ensured by quality assurance using PSNR and RLE. Usability is improved by real-time alerts, an easy-to-use user interface, and smooth social media integration. PFDNet is a proactive system that protects image integrity, fosters user trust, and effectively and robustly handles changing image forgery threats.

REFERENCES

[1] C. Dong, X. Chen, R. Hu, J. Cao and X. Li (2023) MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection.
[2] X. Liang, Z. Tang, X. Zhang, M. Yu and X. Zhang (2023) Robust hashing with local tangent space alignment for image copy detection.
[3] X. Liang, Z. Tang, Z. Huang, X. Zhang and S. Zhang (2023) Efficient hashing method using 2D–2D PCA for image copy detection.
[4] X. Lin et al., (2021) Image manipulation detection by multiple tampering traces and edge artifact enhancement.

WEBSITES REFERRED

[5] www.python.org
[6] www.numpy.org
[7] www.Scikit-Learn.
[8] www.seaborn.pydata.org