IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

REAL-TIME IMAGE FORGERY DETECTION USING AI WITH OPTIMIZED CNN TECHNIQUES

Selvavinayagam[1]

Professor & Head, Computer Science and engineering, Info Institute of engineering, Coimbatore, India[1]

Siva Subramanian[2]

Student, Computer Science and engineering, Info Institute of engineering, Coimbatore, India[2]

Bagavathi Raja[3]

Student, Computer Science and engineering, Info Institute of engineering, Coimbatore, India[3]

Nithyanantham[4]

Student, Computer Science and engineering, Info Institute of engineering, Coimbatore, India[4]

Udhayakumar[5]

Student, Computer Science and engineering, Info Institute of engineering, Coimbatore, India[5]

Abstract: The advancement of image manipulation techniques has raised significant concerns regarding the authenticity of digital content. This initiative presents a real-time image forgery detection system that integrates traditional image processing techniques with a streamlined Convolutional Neural Network (CNN). The system is specifically designed to identify Copy-Move and Splicing forgeries through methods such as resizing, DCT transformation, and edge analysis during the preprocessing stage. A trained CNN model is utilized for classification, determining whether an image is genuine or altered. To improve accessibility, the detection mechanism is incorporated into a Chrome browser extension, enabling users to evaluate images prior to downloading. The model is optimized for performance on low-resource systems while maintaining accuracy. Assessment using standard metrics, including precision, recall, F1-score, and accuracy, validates the model's dependability. This strategy offers an effective and user-friendly solution for verifying digital media, providing substantial benefits to sectors such as digital forensics, journalism, and content moderation.

Index Terms - Image Forgery Detection, Copy-Move and Splicing, Convolutional Neural Network, Real-Time Analysis, Browser-Based Deployment.

I. INTRODUCTION

In the current digital age, the alteration of images has become increasingly prevalent and accessible due to the emergence of sophisticated editing tools. This extensive modification of visual content raises significant issues in areas such as journalism, digital forensics, social media, and legal investigations, where the authenticity of images is paramount. Among the most common forgery methods are Copy-Move and Splicing, which are often challenging to detect with the naked eye. To tackle this issue, the proposed initiative presents a real-time image forgery detection system powered by Artificial Intelligence, specifically utilizing an optimized Convolutional Neural Network (CNN). The system employs traditional preprocessing techniques, including resizing, DCT transformation, and edge detection, to improve feature extraction. A lightweight

CNN model is trained to recognize forgery patterns and categorize images as either authentic or altered. Furthermore, the solution is implemented as a Chrome browser extension, enabling users to verify images effortlessly before downloading, thereby fostering secure and reliable digital media consumption.

II. Related Work

Various methodologies have been established for detecting image forgery, encompassing both traditional handcrafted feature-based strategies and sophisticated deep learning frameworks. Initial techniques concentrated on statistical indicators, including discrepancies in JPEG compression, colour irregularities, or repeated areas identified through block matching and keypoint descriptors. Although effective in controlled environments, these methods frequently struggled to adapt to diverse image formats and types of manipulation. Recent developments utilize Convolutional Neural Networks (CNNs) to autonomously extract deep features from data, enhancing both accuracy and resilience. Nonetheless, many deep learning architectures are resource-intensive and not ideal for real-time applications. Additionally, research has investigated hybrid models that combine preprocessing techniques such as DCT and edge detection with CNNs to achieve a balance between efficiency and performance, which serves as the basis for the approach taken in this project.

III. METHODOLOGY

It outlines the complete pipeline followed in the development of the proposed **Image Forgery Detection using CNN**.

3.1 Data Collection and Processing

The system begins with collecting a diverse set of images from benchmark datasets such as CASIA v2.0,covering both authentic and tampered samples. Images are categorized based on forgery type—Copy-Move and Splicing. The dataset is split into training, validation, and test sets to ensure balanced learning and evaluation. Each image is labelled and verified before use. Proper data handling techniques, including shuffling and normalization, are applied to improve model generalization and reduce bias during training.

3.2 Preprocessing Techniques

Before feeding images into the CNN model, several preprocessing steps are performed to enhance feature extraction and improve classification accuracy. Each image is resized to a fixed dimension to ensure consistency in model input. Color space conversion (e.g., RGB to YCbCr) helps expose forgery-prone regions. Noise reduction filters are applied to remove unwanted artifacts. Additionally, edge detection methods and Discrete Cosine Transform (DCT) are used to reveal inconsistencies that may indicate manipulation, enabling the model to better identify forged areas.

3.3 CNN Model Design and Training

A lightweight Convolutional Neural Network (CNN) is developed for binary classification—authentic or forged. The model is designed to be efficient and suitable for real-time inference.

- **Convolutional Layers**: These layers apply multiple filters across the image to extract important spatial features, such as textures and edges, which help distinguish between natural image regions and manipulated areas.
- **Dropout and Batch Normalization:** Dropout randomly disables certain neurons during training to prevent overfitting, while batch normalization stabilizes learning and accelerates convergence by normalizing layer inputs**.**
- **Fully Connected Layers:** The final dense layers interpret extracted features to make predictions. They map high-dimensional features to output classes using a softmax activation for probability estimation.

3.4 Chrome Extension Integration

To make the system accessible to end users, the model is integrated with a Chrome browser extension. When a user attempts to download an image, the extension intercepts the request and sends the image to a locally or remotely hosted API. The backend processes the image through the trained CNN and returns the

prediction in real time. Based on the result, the extension either allows the download or warns the user about potential forgery, providing a seamless and secure browsing experience for everyday users.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The system is implemented as a modular architecture combining traditional image processing and deep learning to detect image forgeries in real time. The design emphasizes performance, usability, and lightweight deployment. A CNN-based prediction model integrates with a Chrome extension to provide on-the-fly forgery analysis during image downloads. Each component works independently but contributes to an efficient, secure, and responsive user experience suitable for practical, everyday verification scenarios.

4.1 System Architecture

The architecture consists of four major components: the frontend (Chrome extension), the backend API, the preprocessing engine, and the CNN model. When a user attempts to download an image, the extension captures the image URL and sends it to the backend. The backend processes the image—applying noise reduction, resizing, DCT, and edge detection—and feeds it into the CNN model. The model returns a prediction, which is passed back to the extension to inform or block the download in real time.

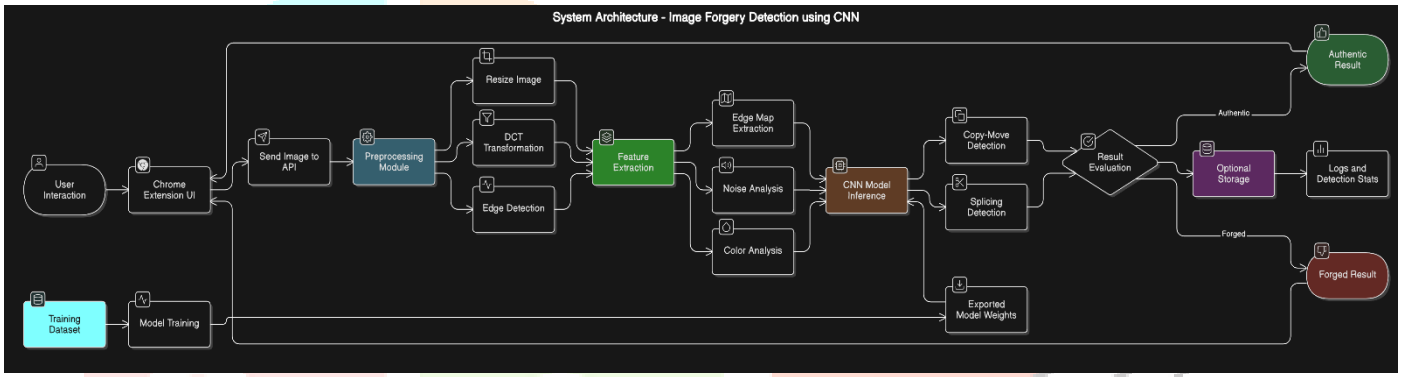

Fig 4.1.1 Architecture of Image Forgery Detection using CNN

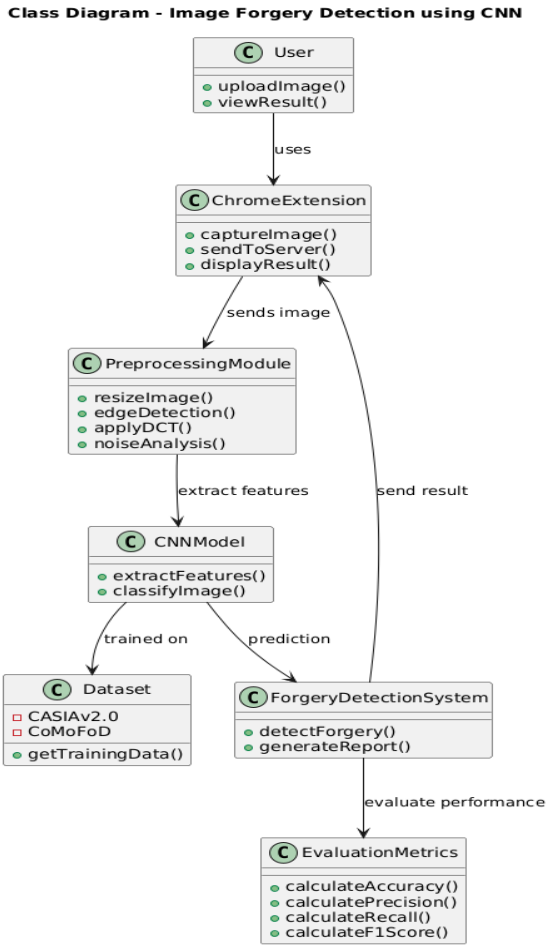

Fig 4.1.2 Proposed Working Flow Class Diagram

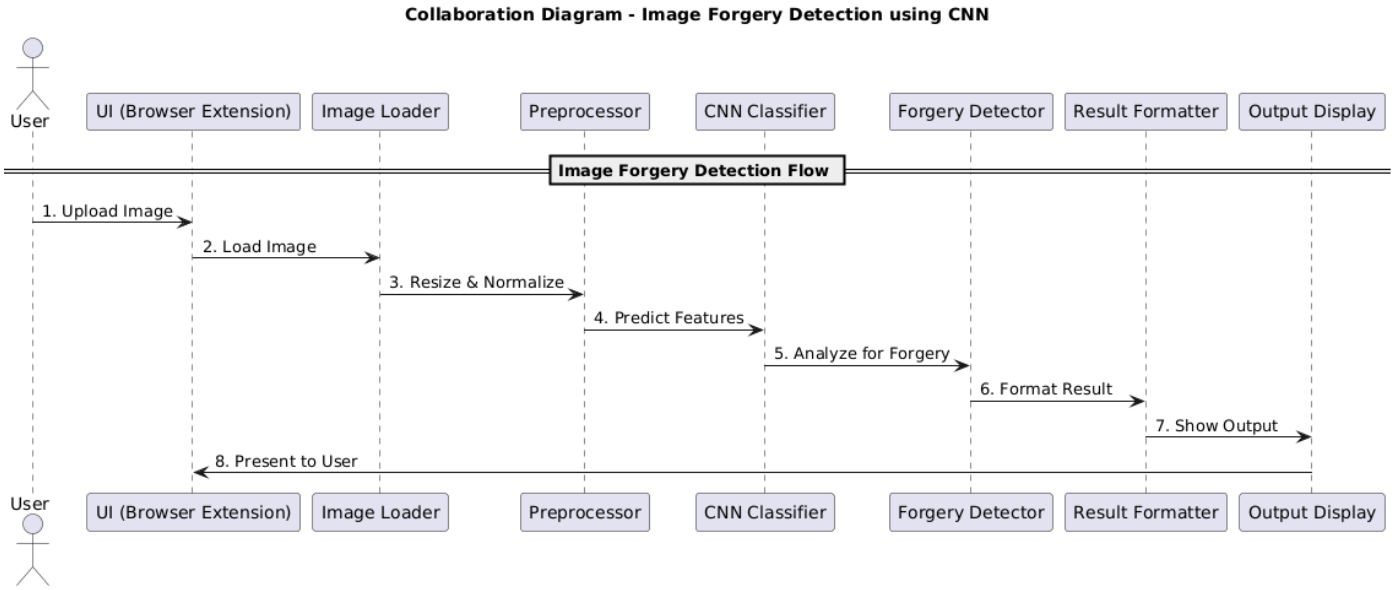

Fig 4.1.3 Proposed Collaboration Architecture for Working Prototype

4.2 System Modules

The proposed system enables seamless bidirectional translation between sign language and spoken or typed language. The real-time interaction process comprises the following sequential components:

4.2.1 Preprocessing

This module standardizes all incoming images by resizing them to a uniform dimension and converting color spaces from RGB to YCbCr. It removes noise using filters and applies DCT and edge detection to extract forgery-revealing patterns. These preprocessing steps enhance the CNN's ability to learn differences between authentic and tampered images, ensuring more reliable classification during inference.

4.2.2 CNN Model

The CNN model is trained using labelled dataset (CASIA v2.0) and performs binary classification: authentic or forged. It includes convolutional layers for feature extraction, dropout for regularization, and dense layers for classification. The model is optimized for fast execution, making it lightweight enough for integration with browser-based systems without sacrificing detection accuracy.

4.2.3 API Integration

The Flask or FastAPI backend handles image requests from the Chrome extension. It accepts image data in base64 format, passes it through preprocessing, and invokes the CNN model to predict authenticity. Once the prediction is made, the API sends a structured JSON response back to the extension. This module ensures smooth communication between frontend and backend components.

4.2.4 Frontend Extension

This module is built as a Chrome extension that runs in the user's browser. It intercepts image download attempts, sends image data to the backend API, and displays a result alert. If the image is found to be forged, it cancels the download and notifies the user. This provides a seamless interface for real-time forgery detection without leaving the browser.

4.2.5 Result Handling and Feedback

Once predictions are made, the results are interpreted and formatted for user readability. The extension displays authenticity results using simple notifications or alerts. This module may also store results locally or in a log file for future reference. It could be extended to support visual overlays (e.g., forged region highlighting) in future versions of the system.

V. EVALUATION AND RESULTS

The image forgery detection system was evaluated using a test set derived from the CASIA v2.0 dataset. After training and validation, the optimized CNN model achieved an overall accuracy of **84.7%** in classifying images as either authentic or tampered. This result demonstrates the model's effectiveness in detecting Copy-Move and Splicing forgeries with reliable precision. The system also performed efficiently in real-time scenarios when integrated with the Chrome browser extension, offering both high accuracy and practical usability for image verification during download.

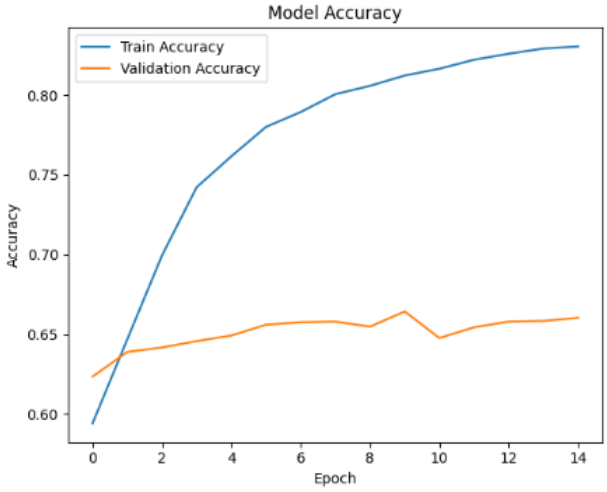

Fig 5.1 Accuracy of the Model

Metric	Value (%)
Accuracy	85.4
Precision	83.2
Recall	82.9
F1-Score	82.5

Table 5.1 Metrics of the proposed system

VI. LIMITATIONS AND FUTURE WORK

The proposed system effectively detects image forgeries in real time but is currently limited to Copy-Move and Splicing techniques. It relies on specific datasets and faces constraints in handling complex manipulations through browser-based deployment. Overcoming these challenges can enhance its scalability, robustness, and applicability in broader real-world scenarios.

6.1.1 Limitations

Forgery Type Coverage: The model is trained only on Copy-Move and Splicing forgeries. It does not yet support other complex techniques like Deepfakes, retouching, or semantic alterations, which limits its generalization across diverse manipulation methods.

Model Sensitivity and Environment: The system's accuracy may drop for images with low resolution, high compression, or noise. Additionally, running the backend server locally restricts access for non-technical users, limiting real-world usability without deployment on a global platform.

6.2.2 Future Work

To enhance system capability, future developments will focus on expanding support to detect Deepfakes and advanced tampering techniques by integrating more complex datasets and transformer-based models. Additionally, improving region-level forgery localization with visual overlays can make results more interpretable. The backend server can be hosted on cloud platforms like AWS or GCP to remove local access constraints and support multi-user environments. A mobile version or browser-independent integration may also be explored to increase reach and flexibility.

VII. CONCLUSION

This paper presents a practical solution for real-time image forgery detection by integrating traditional image processing methods with a lightweight Convolutional Neural Network (CNN). Focused on detecting Copy-Move and Splicing forgeries, the system employs preprocessing techniques like resizing, DCT transformation, and edge detection to enhance forgery-related features. The trained model is capable of classifying images with an accuracy of 85.4%, demonstrating strong potential for real-world usage. Its integration into a Chrome extension allows users to analyse images instantly before downloading, offering a seamless and accessible verification process. The modular design ensures adaptability, making it suitable for further enhancements and scalability. Although the current version is limited to specific forgery types, it lays the groundwork for expanding detection capabilities to more advanced manipulations like Deepfakes. Overall, the system contributes significantly to digital content verification, promoting trust and security in online environments through an efficient, user-friendly, and technically sound solution.

References

[1].*Bruna, J., & Mallat, S.* (2013), “Invariant scattering convolution networks,IEEE Transactions on Pattern Analysis and Machine Intelligence”, 35(8), 1872–1886.

[2].*Bappy, J. H., Roy-Chowdhury, A. K., Bunk, J., Lin, J., Nataraj, L., & Manjunath, B. S.* (2017),”Exploiting spatial structure for localizing manipulated image regions. In Proceedings of the IEEE International Conference on Computer Vision”, (ICCV).

[3].*Nguyen, H. H., Yamagishi, J., & Echizen, I.* (2019),”Multi-task learning for detecting and segmenting manipulated facial images and videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops”, (CVPRW).

[4].*Bayar, B., & Stamm, M. C. (2016),"*A deep learning approach to universal image manipulation detection using a new convolutional layer”, ACM Workshop on Information Hiding and Multimedia Security.

[5].*Dong, J., Wang, W., & Tan, T. (2013).”*CASIA Image Tampering Detection Evaluation Database*”,* Chinese Academy of Sciences Institute of Automation.

[6].*Zhang, X., & Rao, Y.* (2023),”HiFi-Net: Hierarchical Fine-Grained Image Forgery Detection and Localization”,CVPR 2023.

[7].*Nguyen, H., Yamagishi, J., & Echizen, I. (2019),* "Capsule-forensics: Using capsule networks to detect forged images and videos.", IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 10.1109/ICASSP.2019.8682297

[8].*Redmon, J., & Farhadi, A. (2018),* "YOLOv3: An incremental improvement. “For real-time image processing insights, applicable in forgery detection pipelines”,(ARXIV.ORG)

[9]. CASIA Image Tampering Detection Evaluation Database (CASIA v2.0).

[10].TensorFlow Documentation. TensorFlow Open Source Machine Learning Framework. *https://www.tensorflow.org/api_docs*

[11].FastAPI Documentation. FastAPI: High Performance Python API Framework. *https://fastapi.tiangolo.com/*

[12]. OpenCV Library. Open Source Computer Vision Library. *https://docs.opencv.org/4.x/*