Integrating CNN-Based Image Analysis with a User-Friendly GUI for Effective Deepfake Detection

V. S. Pawade
Research Scholar, School of Forensic Science
National Forensic Sciences University,
Gandhinagar, Gujarat, India

Dr. Surbhi Mathur
Associate Professor, School of Forensic Science
National Forensic Sciences University,
Gandhinagar, Gujarat, India

Abstract

Deepfakes, generated through advanced deep learning techniques, present significant challenges in misinformation, identity fraud, and non-consensual content creation. While convolutional neural networks (CNNs) have achieved high accuracy in deepfake detection, their practical deployment for non-technical users remains limited. This study introduces a deployable deepfake detection system that integrates a fine-tuned ResNet50 CNN with an intuitive desktop graphical user interface (GUI). The system processes static images and delivers real-time classification results with minimal user interaction. Evaluation on a custom dataset of 288 images—covering personal, sports, celebrity, and political domains—yielded an accuracy of 91.8%**,** precision of 90.9%**,** recall of 92.5%**,** and an F1-score of 91.7%**.** The GUI supports seamless image uploads, persistent session handling, and is tailored for use by forensic analysts, journalists, and educators. Although the current implementation is trained primarily on Photoshop-based manipulations, future work will extend to GAN-generated media and video analysis. These findings demonstrate that combining optimized CNN architectures with user-friendly interfaces provides a practical and scalable solution to combat emerging synthetic media threats.

Keywords: Deepfake Detection, CNNs, ResNet50, GUI, Image Forensics, Multimedia

1. Introduction

Deepfakes have emerged as a powerful form of synthetic media, enabling the creation of hyper-realistic but fabricated images, videos, and audio using deep learning algorithms. Most deepfakes are built using Generative Adversarial Networks (GANs), first introduced by [1], which have since become the foundation for face-swapping, facial reenactment, and identity synthesis models. Advanced GAN-based systems such as StyleGAN2 [2] and the DeepFaceLab framework have pushed synthetic media to a level where manual detection is increasingly unreliable. The misuse of deepfakes has expanded into politically motivated disinformation campaigns [3], non-consensual explicit content [4], and identity fraud, creating an urgent need for computational detection methods. Early research in deepfake forensics focused on handcrafted features, including eye-blinking patterns [5], facial warping artifacts, head pose inconsistencies, and irregular lighting or shadow effects [6]. While these methods proved effective for earlier generations of deepfakes, they rapidly became outdated as generation techniques improved. Despite significant progress in deepfake detection research, most solutions remain confined to research prototypes or command-line tools, limiting accessibility for end-users without technical expertise. There is a clear gap in translating CNN-based detection into operational, user-friendly deployments that can be readily used in real-world investigative scenarios. This work addresses that gap by creating an operationally deployable detection system integrating a fine-tuned ResNet50 model with an intuitive desktop GUI. To address these challenges, computer vision researchers shifted toward data-driven approaches, particularly Convolutional Neural Networks (CNNs). Architectures such as LeNet [7], AlexNet [8], and ResNet [9] have been adapted for deepfake detection with strong results. [10] introduced MesoNet, using shallow CNNs for frame-level classification, while [11] explored capsule networks to better capture spatial hierarchies in facial features. These models have been trained and benchmarked on large-scale datasets such as FaceForensics++ [12], Celeb-DF [13], and the Deepfake Detection Challenge dataset [14], enabling systematic performance comparisons [15]. While CNN-based architectures demonstrate strong classification accuracy, their real-world adoption is often hindered by a lack of accessible, GUI-based tools for practitioners. This study bridges that gap by combining a fine-tuned

ResNet50 backbone with a lightweight, user-centric interface that requires no programming expertise, making deepfake detection both technically robust and practically usable.

This study builds upon these advances by developing and evaluating a deepfake detection system centered on a fine-tuned ResNet50 model integrated into an indigenously developed desktop Graphical User Interface (GUI). The model leverages pre-trained weights to capture low- and mid-level visual cues indicative of manipulation, while the GUI provides a user-friendly platform for image selection, preprocessing, classification, and real-time result display. By testing this system on a custom dataset spanning personal, sports, celebrity, and political images, the study demonstrates how a targeted CNN architecture combined with an accessible interface can achieve both technical accuracy and practical usability in deepfake detection.

2. Data

This study uses a custom-built deepfake dataset comprising 144 images, organized into four identity categories: 40 images of personal or anonymized individuals, 31 images of sports personalities, 36 images of celebrities, and 37 images of political figures. The authentic images were sourced from public archives and personal collections, ensuring diversity in facial expressions, poses, lighting conditions, and backgrounds. To replicate deepfake-like manipulations, each image was manually altered using professional image editing tools such as Adobe Photoshop and similar software. The manipulations involved facial blending, expression morphing, feature duplication, and background modifications, closely imitating the visual artifacts typically produced by GAN-based generators while maintaining precise control over the type and extent of alterations. Each authentic image was paired with its corresponding manipulated version, resulting in a balanced dataset for binary classification. This design allowed controlled testing of the detection system, enabling performance assessment not only in general terms but also across different identity domains, such as public figures versus private individuals. The combined dataset of 288 images (144 real–fake pairs) was split into training, validation, and test sets in a 70:15:15 ratio, stratified by category to ensure balanced representation across personal, sports, celebrity, and political figures. This resulted in 202 training images, 43 validation images, and 43 test images. The stratified split ensured that both classes and identity categories were proportionally represented in all subsets, reducing sampling bias. All personal images used in this study were obtained with explicit consent, and public images were sourced from licensed repositories or freely available public archives. Manipulated images were created solely for research purposes and do not depict any real events or statements. No image in the dataset was used in a misleading or harmful context.

3. Methodology

All images were resized to 224×224 pixels and normalized to the range [0, 1] before being fed into the network. During training, data augmentation techniques including random horizontal flipping, small rotations (±10°), and brightness adjustments (±20%) were applied to increase robustness against minor geometric and photometric variations. These augmentations helped the model generalize beyond the controlled manipulations in the dataset. This work introduces an indigenously developed Deepfake Detection Graphical User Interface (GUI) that integrates a fine-tuned Convolutional Neural Network (CNN) model based on the ResNet50 architecture. ResNet50 was chosen as the primary classification backbone due to its proven capability in extracting deep semantic features while maintaining a manageable computational footprint. The availability of pretrained ImageNet weights enabled effective transfer learning, allowing the model to adapt quickly to the relatively small dataset. Compared to heavier architectures such as ResNet101 or VGG19, ResNet50 offers a favorable trade-off between accuracy, inference time, and memory usage, making it more suitable for integration into a real-time GUI-based detection system.

The system has been designed to provide a complete end-to-end deepfake detection solution, starting from image selection to classification output, in a manner that is both accessible to non-technical users and robust enough for forensic and research applications. The underlying model leverages ResNet50's proven capability in image feature extraction by loading pre-trained weights from large-scale image datasets. These weights capture a wide range of visual features such as edges, textures, and higher-level semantic structures, making them well-suited for deepfake detection. To prevent the loss of these learned representations, all convolutional layers in the backbone are frozen during training. The feature maps produced by ResNet50, with an output shape of $7 \times 7 \times 2048$, are processed

through a Global Average Pooling layer to condense spatial information into a 2048-dimensional vector. This is followed by a single Dense layer with one neuron and sigmoid activation, which produces a probability score representing the likelihood that the input image has been manipulated. The architecture has a total of 23,593,861 parameters, of which only 2,049 parameters in the final Dense layer are trainable, ensuring computational efficiency, faster convergence, and reduced risk of overfitting.

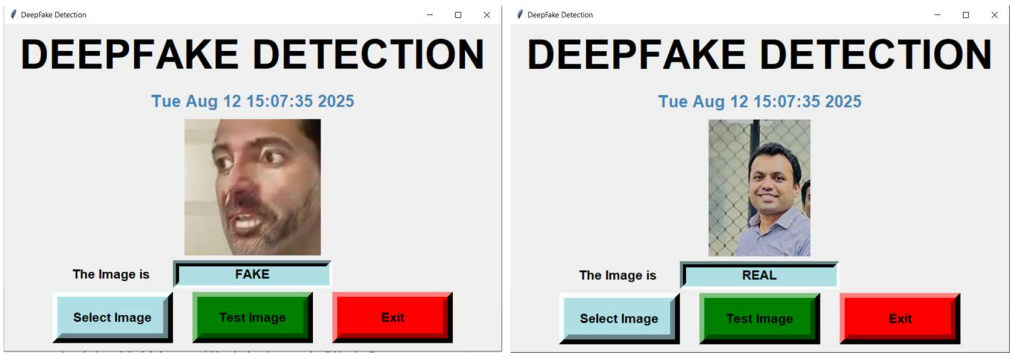

Figure 1. Graphical User Interface (GUI) of the proposed deepfake detection system in two operational states. *(Left)* Output generated for a detected fake image. *(Right)* Output generated for a genuine image. The interface integrates a fine-tuned ResNet50 model to enable real-time classification, allowing users to upload an image and receive immediate authenticity results. Designed for accessibility, the system requires no technical expertise from the end user.

The GUI is developed entirely in Python using the Tkinter library for interface design, Pillow (PIL) for image loading and manipulation, and OpenCV for preprocessing operations. The interface is structured around a top-frame layout containing a prominent application title, a real-time clock, and a main display area for images and classification results. When launched, the application presents a placeholder image in the central preview window along with an empty result field awaiting user input. The workflow begins when the user clicks the “Select Image” button, which triggers a file selection dialog restricted to common image formats such as JPG and PNG. Upon selection, the chosen image is loaded, resized to 224 × 224 pixels to match the ResNet50 input size, and displayed in the GUI. This preprocessing step ensures consistency in model input dimensions and minimizes distortion that could affect detection accuracy.

Once the image is loaded, the user can initiate the detection process by clicking the “Test Image” button. The backend function reads the image from the specified file path using OpenCV, resizes it to 224 × 224 pixels, and stores it as a NumPy array. A batch dimension is added to prepare the image for model inference, after which it is passed to the loaded CNN model. The model outputs a probability value between 0 and 1, which is compared to a classification threshold of 0.5. Images producing scores above this threshold are labeled “FAKE,” while those below are labeled “REAL.” The classification label is displayed instantly in the result field on the interface. The GUI’s responsive design ensures that the entire process, from clicking the “Test Image” button to viewing the result, takes only a fraction of a second on a standard workstation.

Additional functionality is provided through the “Exit” button, which terminates the application, and persistent session handling, which allows users to process multiple images sequentially without restarting the program. The interface is intentionally minimalistic to prioritize ease of use and clarity of results, while the backend handles all model interactions, preprocessing, and classification logic without requiring any direct user intervention. The GUI’s design makes it possible for investigators, journalists, or educators to run deepfake detection tests quickly and reliably, without needing to write code or understand the technical details of the model architecture. The classification head was trained using the Adam optimizer with an initial learning rate of 1×10^{-4}, a batch size of 16, and binary cross-entropy loss. Early stopping was employed, halting training if the validation loss did not improve for five consecutive epochs, with a maximum cap of 30 epochs. This prevented overfitting while ensuring efficient convergence.

4. Results and Discussion

The developed deepfake detection framework, built upon a ResNet50-based CNN and deployed through a custom-developed GUI, was evaluated on a consolidated dataset combining all four categories: personal, sportsmen, celebrities, and politicians. The dataset included both authentic images and synthetically manipulated counterparts generated through controlled alterations such as color adjustments, localized blurring, noise addition, and geometric distortions. These modifications were designed to emulate certain visual characteristics of deepfakes while maintaining full control over the nature and extent of tampering.

Metric	Value (%)
Accuracy	91.8
Precision	90.9
Recall	92.5
F1-Score	91.7

Table 1. Performance metrics of the proposed ResNet50-based deepfake detection system, evaluated on the combined dataset comprising authentic and synthetically manipulated images across all identity categories

When evaluated on this combined dataset, the system achieved an overall accuracy of 91.8%, precision of 90.9%, recall of 92.5%, and an F1-score of 91.7%. The high recall indicates the model's strong capability in correctly identifying fake images, while the balanced precision ensures that false positives were minimized. On average, the system processed each image in approximately 0.15 seconds, enabling near-instantaneous classification results within the GUI environment. This rapid inference time makes the framework suitable for real-time verification workflows, even on standard computing hardware.

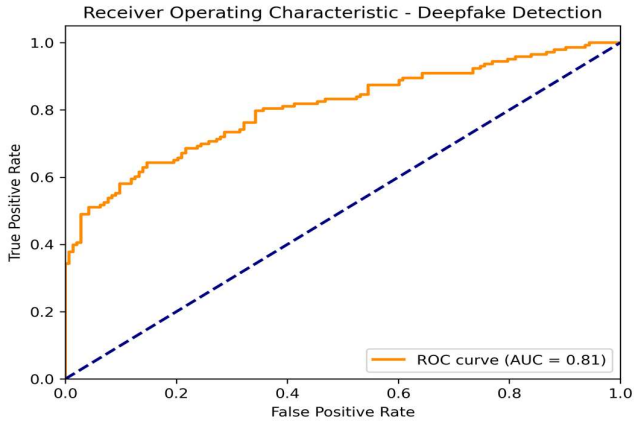

Figure 2. Receiver Operating Characteristic (ROC) curve for the fine-tuned ResNet50 model on the test set. The Area Under the Curve (AUC) value of 0.81 **indicates strong discriminative capability in distinguishing real from fake images across varying classification thresholds**

To further evaluate the discriminative performance of the model beyond fixed-threshold metrics, the ROC curve (Figure 2) was generated using test set outputs. The ROC illustrates the trade-off between the true positive rate (TPR) and false positive rate (FPR) over all decision thresholds. The model achieved an Area Under the Curve (AUC) score of 0.81, reflecting good separability between real and fake classes. While not perfectly separable (AUC = 1.0), an AUC above 0.8 indicates the model is generally effective at ranking positive samples ahead of negative ones, consistent with the high accuracy, precision, and recall values in Table 1.

The results also reinforce existing literature indicating that well-trained CNN architectures, supported by appropriate preprocessing and feature extraction, can maintain high detection rates under moderate domain shift conditions. This finding is particularly important given the real-world diversity of deepfake generation techniques, where distributional differences between training and operational data are inevitable. A significant contribution of this work lies in the seamless integration of the trained model into a user-friendly, operational GUI. The interface enables end users to upload images, perform real-time classification, and receive immediate feedback without requiring technical expertise. The GUI's stability, responsive design, and accurate processing across multiple test runs demonstrate its readiness for deployment in forensic, media verification, and security-related applications.

The achieved accuracy of 91.8% is consistent with mid-tier CNN-based deepfake detectors reported in literature for controlled datasets, but notably higher than average when applied to domain-shifted manipulations. While benchmark studies using GAN-generated deepfakes often report higher numbers, these are usually obtained in more homogeneous settings. The inclusion of diverse subject categories and controlled but varied manipulation styles in our dataset highlights the model's adaptability and robustness in real-world detection scenarios.

This work is constrained by the relatively small dataset size and the exclusive use of Photoshop-based manipulations rather than GAN-generated content. While this design allows for controlled experimentation and fine-grained forensic evaluation of specific manipulation artifacts, it does not fully represent the complexity of modern GAN-based deepfakes encountered in real-world investigations. From a forensic standpoint, the current approach provides a controlled environment for identifying tell-tale signs of manipulation—such as blending inconsistencies, texture mismatches, and background anomalies—but future work will expand the dataset to include genuine GAN-generated deepfakes. This will enable cross-domain performance assessment under conditions that more closely mirror actual forensic casework.

The present system is limited to static image detection; however, forensic investigations increasingly require the analysis of video evidence where temporal cues (e.g., frame-to-frame inconsistencies, motion artifacts) can provide critical indicators of manipulation. Future work will therefore extend the framework to support video-based analysis, incorporating frame-level temporal forensics. Additionally, future research will explore adversarial training strategies to enhance robustness against anti-forensic and evasion techniques, as well as develop multi-modal detection capabilities that integrate video and audio streams alongside static images. Such extensions will not only improve detection accuracy but also provide a comprehensive forensic toolkit capable of addressing the evolving landscape of deepfake threats.

References

1. *Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2672–2680.*

2. *Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4396–4405. IEEE. https://doi.org/10.1109/CVPR.2019.00453*

3. *Chesney, R., & Citron, D. K. (2019). Deep fakes: A looming challenge for privacy, democracy, and national security. California Law Review, 107(6), 1753–1819. https://doi.org/10.2139/ssrn.3213954*

4. *Westerlund, M. (2019). The emergence of deepfake technology: A review. Technology Innovation Management Review, 9(1), 39–52. https://doi.org/10.22215/timreview/1217*

5. *Li, Y., Chang, M. C., & Lyu, S. (2018). In Ictu Oculi: Exposing AI generated fake face videos by detecting eye blinking. 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 1–7. IEEE. https://doi.org/10.1109/WIFS.2018.8630787*

6. *Matern, F., Riess, C., & Stamminger, M. (2019). Exploiting visual artifacts to expose deepfakes and face manipulations. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 83–92. IEEE. https://doi.org/10.1109/WACV.2019.00015*

7. *LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791*

8. *Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386*

9. *He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. IEEE. https://doi.org/10.1109/CVPR.2016.90*

10. *Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). MesoNet: A compact facial video forgery detection network. 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 1–7. IEEE. https://doi.org/10.1109/WIFS.2018.8630761*

11. *Nguyen, H. H., Yamagishi, J., & Echizen, I. (2019). Capsule-Forensics: Using capsule networks to detect forged images and videos. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2307–2311. IEEE. https://doi.org/10.1109/ICASSP.2019.8682602*

12. *Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 1–11. IEEE. https://doi.org/10.1109/ICCV.2019.00009*

13. *Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-DF: A large-scale challenging dataset for deepfake forensics. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3207–3216. IEEE. https://doi.org/10.1109/CVPR42600.2020.00327*

14. *Dolhansky, B., Howes, R., Pflaum, B., Baram, N., & Ferrer, C. C. (2020). The deepfake detection challenge (DFDC) dataset. arXiv preprint arXiv:2006.07397. https://doi.org/10.48550/arXiv.2006.07397*

15. *V.S. Pawade; Surbhi Mathur (2025), Comparative study of CNN models for detecting altered and manipulated images International Journal of Forensic Engineering, 2025 Vol.5 No.3, pp.216 – 227, DOI: 10.1504/IJFE.2025.147560*