IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

SMART DEEPFAKE DETECTION SYSTEM USING CNN FOR DIGITAL IMAGE AUTHENTICITY

K. V. S. Radha Madhavi[1], Dr. D. Kishore Babu[2]

Department of Computer Science and Engineering, Bapatla Engineering College, ANU, India[1]

Guide, Department of Computer Science and Engineering, Bapatla Engineering College, ANU, India[2]

Abstract: The rapid development of generative artificial intelligence has increased the availability of realistic synthetic and manipulated facial images. This paper presents a Smart Deepfake Detection System Using Convolutional Neural Network (CNN) for automated binary classification of facial images into real and fake classes. The work extends the supplied project manuscript by providing a fuller IEEE-style treatment of the problem, preprocessing pipeline, CNN architecture, web deployment, evaluation methodology, security considerations, and reproducibility requirements. The implemented system resizes and normalizes an uploaded image, performs CNN inference, and displays the predicted class and confidence through a Flask-based interface. The supplied project evidence demonstrates two functional cases: a representative genuine face is returned as "Real Face Detected," while a representative manipulated/synthetic face is returned as "Fake Face Detected." Because the source material does not provide dataset counts or statistical metrics, this paper deliberately does not fabricate accuracy, precision, recall, F1-score, ROC-AUC, or confusion-matrix values. Instead, it specifies a publication-ready evaluation protocol that can be populated from the actual experimental run.

Index Terms: Deepfake detection, convolutional neural network, facial image forensics, synthetic media, image classification, Flask, digital security, deep learning.

I. INTRODUCTION

Digital photographs are increasingly used as evidence, identity representations, communication media, and inputs to automated decision systems. At the same time, generative models and face-manipulation tools can produce visually convincing images with limited technical expertise. This creates a trust problem: visual plausibility is no longer sufficient evidence that a facial image is authentic. Deepfake detection therefore sits at the intersection of computer vision, cybersecurity, digital forensics, and responsible artificial intelligence.

The research literature shows a progression from handcrafted forensic cues toward learned representations. CNN-based systems learn local textures, edges, blending traces, and other discriminative patterns directly from training examples. FaceForensics++ established a widely used benchmark for manipulated facial imagery, while Celeb-DF and DFDC demonstrated the difficulty of detecting high-quality manipulations and cross-dataset examples [4], [7], [9]. Other work has investigated warping artifacts, blending boundaries, and generated-image fingerprints [5], [6], [8].

The supplied project is intentionally focused on an implementable image-level detector rather than a large video-forensics platform. Its central contribution is an end-to-end pipeline that connects a CNN classifier with a Flask web interface. This makes the system understandable to students and practitioners while providing a foundation for later extensions such as transfer learning, transformer models, video-frame aggregation, and cross-dataset testing.

II. PROBLEM

STATEMENT AND OBJECTIVES

The problem addressed is binary classification of a user-provided facial image as authentic or manipulated. The detector must transform an arbitrary uploaded image into the input representation expected by the trained network and return a clear decision to the user. The principal objectives are to automate feature extraction, reduce dependence on handcrafted rules,

provide a simple prediction interface, and establish an evaluation procedure suitable for reproducible experimentation.

• Develop a CNN-based binary image classifier for real/fake facial images.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

- Apply consistent resizing and pixel normalization during training and inference.
- Integrate the trained model with a Flask application for practical prediction.
- Record confidence scores together with predicted labels.
- Define quantitative evaluation using accuracy, precision, recall, F1-score, ROC-AUC, and a confusion matrix.
- Assess robustness using controlled changes in compression, resolution, and manipulation source.

III. RELATED WORK

Goodfellow et al. describe deep learning as a framework for learning hierarchical representations, while Krizhevsky et al. demonstrated the impact of deep CNNs on image classification [1], [2]. VGG introduced a deep architecture based on repeated small convolutional filters, establishing a useful transfer-learning backbone [3]. ResNet and EfficientNet later improved the efficiency and scalability of image representation learning [20], [19].

For deepfake forensics, Li and Lyu showed that face-warping artifacts can provide useful CNN-detectable cues [6]. Rössler et al. introduced FaceForensics++ as a benchmark covering several facial manipulation methods [4]. Wang et al. demonstrated that CNN-generated images can contain detectable statistical traces [5]. Face X-ray sought a more general representation based on blending boundaries rather than dependence on a specific manipulation algorithm [8]. These studies motivate a detector that learns from diverse examples while acknowledging that generalization remains a major challenge.

Recent work increasingly explores transformer-based detectors. M2TR combines multi-scale transformer processing with frequency information [10], DFDT uses transformer representations to model local and global relationships [11], and comparative studies have examined CNN and ViT generalization under cross-forgery conditions [13]. DeepfakeBench further emphasizes standardized preprocessing and evaluation protocols [16]. These developments motivate the future extensions described in this manuscript.

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed architecture contains five stages: dataset preparation, preprocessing, CNN feature learning, classification, and web deployment. During training, real and fake images are assigned labels and converted into a common input size. The CNN extracts hierarchical features through convolution and pooling operations. The learned representation is converted to a class decision by dense layers and a final probability layer. During inference, the Flask application receives an image, applies the same preprocessing, loads the saved model, performs forward propagation, and presents the resul

TABLE I. SYSTEM MODULES AND DATA FLOW

Module	Input	Processing	Output
Dataset	Real/fake images	Labeling and split	Train/validation/test sets
Preprocessing	Raw image	Resize + normalize	Model-ready tensor
CNN	Image tensor	Convolution + pooling + dense layers	Class probabilities
Inference	Uploaded image	Model prediction	Real/Fake + confidence
Web layer	Prediction result	HTML response	User-visible decision

V. METHODOLOGY

A. DATASET PREPARATION

The source manuscript states that real and fake images are organized for model development but does not report the exact number of samples. For publication, the authors should record the number of real images, fake images, identities, source datasets, and train/validation/test counts. Identity leakage must be avoided: images belonging to the same person should not be distributed across training and testing subsets when identity-specific cues could influence the decision.

B. PRE PROCESSING

Each image is resized to the fixed spatial dimensions required by the trained network and converted to a numerical tensor. Pixel normalization reduces scale variation and makes training more stable. Optional augmentation can include horizontal

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

flipping, small rotations, brightness variation, blur, and JPEG compression. Augmentation should be applied only to the training data unless the experiment explicitly studies robustness.

C. CNN CLASSIFIER

The CNN uses convolutional layers to learn local spatial patterns. Pooling layers reduce spatial dimensions while preserving dominant responses. The flattened or globally pooled representation is passed to dense layers, followed by a two-class output. For binary classification, the model can be formulated as $p(y|x)=\text{softmax}(z)$, where z represents the final logits. The predicted class is $\arg\max p(y|x)$. Training minimizes categorical or binary cross-entropy depending on the selected output formulation.

The key design principle is consistency between training and deployment. If training images are normalized in one manner and uploaded images are normalized differently, the model may exhibit a distribution shift that reduces reliability. Therefore, preprocessing functions should be shared or replicated exactly between the training notebook and Flask inference code.

D. FLASK DEPLOYMENT

The web application provides an upload endpoint, validates the file type, stores or streams the image for inference, invokes the trained H5 model, and renders the predicted label and confidence. Input validation should restrict acceptable image formats and enforce file-size limits. Uploaded files should not be executed or treated as trusted content. For deployment beyond a local demonstration, authentication, logging, rate limiting, secure storage, and HTTPS should be considered.

VI. EXPERIMENTAL DESIGN

The supplied project material contains two representative functional demonstrations. These establish that the application can execute an upload-to-prediction path, but they are not sufficient to estimate population-level performance. A publication experiment should use a held-out test set that is never used for model selection. The same test set should be retained for all model comparisons.

TABLE II. RECOMMENDED EVALUATION METRICS

Metric	Purpose	Formula/Interpretation
Accuracy	Overall correctness	(TP+TN)/(TP+TN+FP+FN)
Precision	Reliability of fake predictions	TP/(TP+FP)
Recall	Fake detection sensitivity	TP/(TP+FN)
F1-score	Balance of precision/recall	2PR/(P+R)
ROC-AUC	Threshold-independent ranking	Area under ROC curve

VII. FUNCTIONAL RESULTS

The source project reports two observed interface outputs. In the first case, a representative genuine face is classified as “Real Face Detected.” In the second case, a representative manipulated/synthetic face is classified as “Fake Face Detected.” These observations verify the functional path from upload through preprocessing and CNN inference to user-visible classification.

TABLE III. SUPPLIED FUNCTIONAL TESTS

Test	Input	Observed output	Evidence type
T1	Sample genuine face	Real Face Detected	Functional demonstration
T2	Sample manipulated/synthetic face	Fake Face Detected	Functional demonstration

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

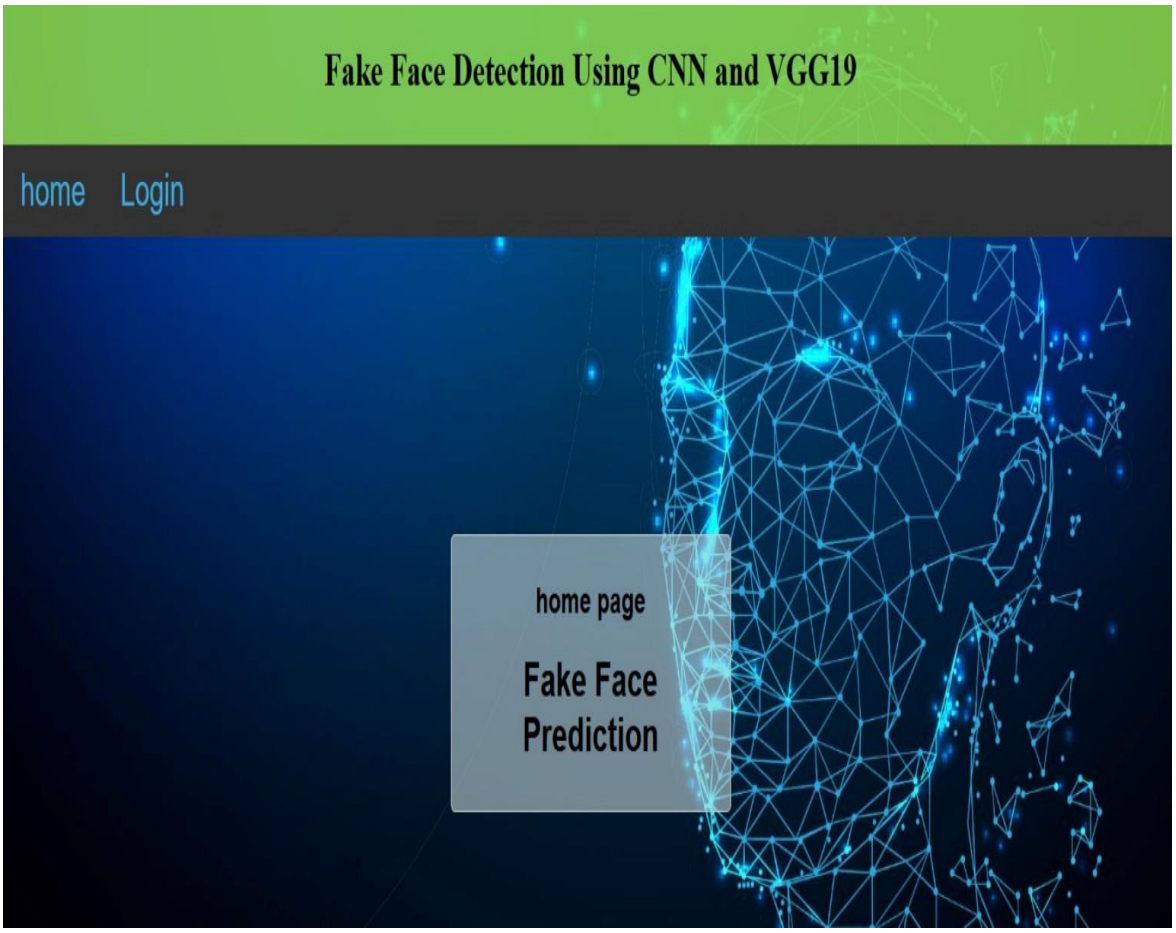

Fig. 1. Home page showing the Fake Face Prediction module.

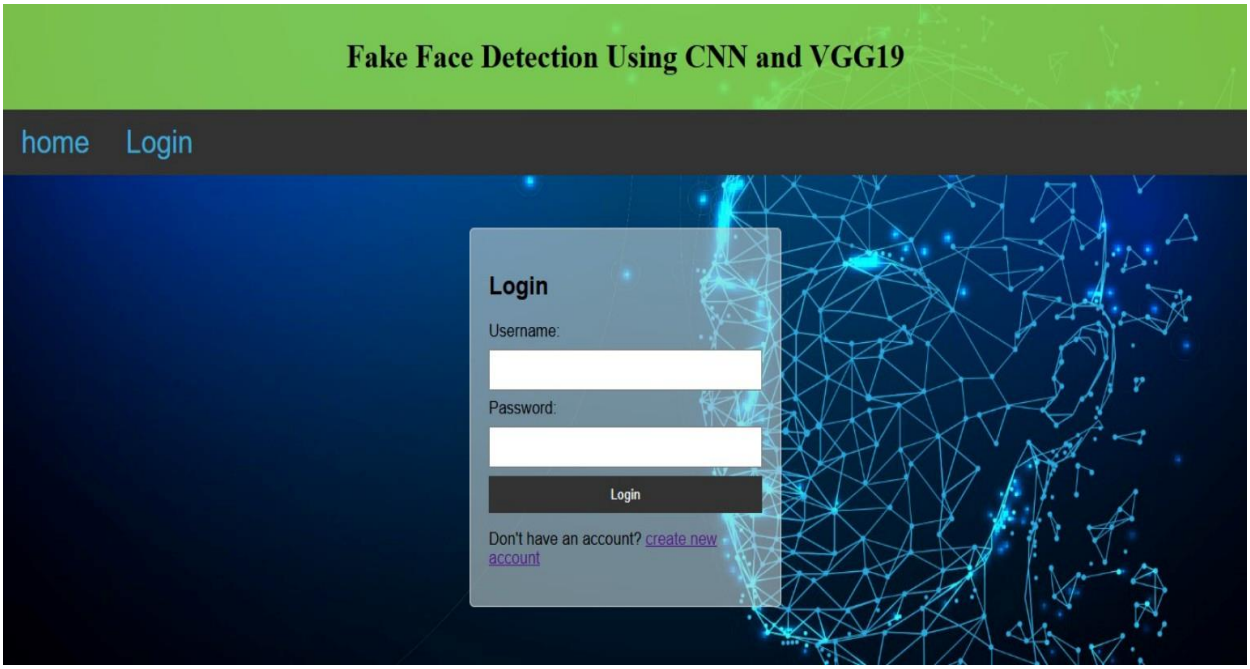

Fig. 2. Login interface of the proposed deepfake detection web application.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

Fig. 3. Image-upload interface used to submit an input image for prediction.

The following screenshots provide additional functional evidence of the deployed interface, including user authentication, the prediction home page, and the image-upload stage.

No numerical performance values are reported because the supplied material does not provide the dataset size, number of epochs, batch size, optimizer, learning rate, train/test split, accuracy, precision, recall, F1-score, ROC-AUC, or confusion matrix. In a publication submission, these fields should be populated from the actual training log and test run rather than estimated or copied from unrelated studies.

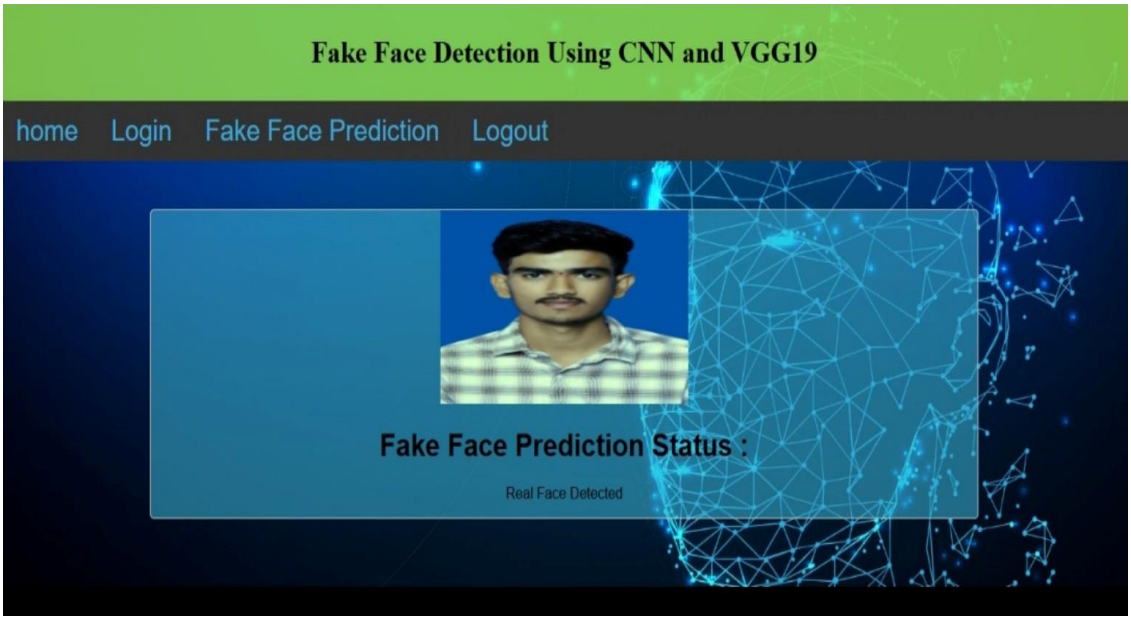

Fig. 4. Functional prediction result for a representative genuine face: Real Face Detected.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

Fig. 5. Functional prediction result for a representative manipulated/synthetic face: Fake Face Detected.

VIII. SECURITY AND ROBUSTNESS CONSIDERATIONS

A deepfake detector should be treated as a decision-support component rather than an infallible authenticity oracle. An attacker can modify image compression, resize the image, crop the face, add noise, or generate new manipulation methods that differ from the training distribution. The system should therefore be evaluated under controlled perturbations and cross-source testing.

A useful robustness matrix includes original test images, JPEG-compressed versions, downsampled versions, blurred versions, brightness-adjusted versions, and samples from a different manipulation family. Reporting performance separately for each condition can reveal whether the detector has learned general forensic traces or dataset-specific shortcuts.

IX. LIMITATIONS

- The supplied dataset description is incomplete and does not provide sample counts or source composition.
- The supplied evidence contains functional examples but no statistically sufficient test-set results.
- Generalization to unseen manipulation methods is not established by the two demonstrated cases.
- The current work focuses on images and does not model temporal inconsistencies in videos.
- The Flask demonstration is suitable for a prototype; production deployment requires additional security controls.

X. FUTURE EXTENSIONS

A first extension is transfer learning using VGG19, ResNet, or EfficientNet to compare feature reuse against a custom CNN. A second extension is a hybrid CNN–Vision Transformer architecture that combines local texture cues with longer-range spatial relationships. A third extension is video-level detection using frame sampling followed by temporal aggregation. Dataset diversity, cross-dataset evaluation, explainability, and adversarial robustness should be treated as core research questions rather than optional additions.

XI. CONCLUSION

This paper presents a structured IEEE-style formulation of the Smart Deepfake Detection System Using CNN. The system learns visual representations from real and fake facial images and exposes the trained classifier through a Flask web interface. The supplied implementation demonstrates successful functional predictions for representative real and fake examples. The principal research gap is not the ability to produce a label, but the need to quantify reliability under controlled and unseen conditions. The evaluation framework in this paper provides the required structure for a reproducible publication study without inventing unavailable measurements.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 9, September 2026

DOI: 10.17148/IARJSET.2026.13921

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2012.
[3] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in Proc. Int. Conf. Learn. Representations (ICLR), 2015.
[4] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, FaceForensics++: Learning to detect manipulated facial images, in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019.
[5] S. Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, CNN-generated images are surprisingly easy to spot... for now, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020.
[6] Y. Li and S. Lyu, Exposing DeepFake videos by detecting face warping artifacts, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2019, pp. 46–52.
[7] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, Celeb-DF: A large-scale challenging dataset for DeepFake forensics, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 3207–3216.
[8] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, Face X-ray for more general face forgery detection, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 5000–5009.
[9] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton Ferrer, The DeepFake Detection Challenge (DFDC) dataset, arXiv:2006.07397, 2020.
[10] J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, S.-N. Lim, and Y.-G. Jiang, M2TR: Multi-modal multi-scale transformers for DeepFake detection, arXiv:2104.09770, 2021.
[11] A. Khormali and J.-S. Yuan, DFDT: An end-to-end DeepFake detection framework using vision transformer, Applied Sciences, vol. 12, no. 6, Art. no. 2953, 2022.
[12] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, Deepfake detection scheme based on vision transformer and distillation, arXiv:2104.01353, 2021.
[13] D. A. Coccomini, R. Caldelli, F. Falchi, C. Gennaro, and G. Amato, Cross-forgery analysis of vision transformers and CNNs for deepfake image detection, arXiv:2206.13829, 2022.
[14] Z. Wang, Z. Cheng, J. Xiong, X. Xu, T. Li, B. Veeravalli, and X. Yang, A timely survey on vision transformer for deepfake detection, arXiv:2405.08463, 2024.
[15] B. Ghita, I. Kuzminykh, A. Usama, T. Bakhshi, and J. Marchang, Deepfake image detection using vision transformer models, in Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom), 2024, pp. 332–335.
[16] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, DeepfakeBench: A comprehensive benchmark of deepfake detection, arXiv:2307.01426, 2023.
[17] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning, 2021.
[18] F. Pedregosa et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[19] M. Tan and Q. V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 6105–6114.
[20] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

