International Journal on Advanced Computer Theory and Engineering
ISSN: 2319 - 2526
Volume 15 Issue 2s (2026) | Pages: 241-246

DeepFake Face Detection with Handcrafted Features and Logistic Regression

Lina Chaudhari[1], Dhanashree Bansode[2], Purvi Patil[3], Samruddhi Magdum [4]

[1]Assistant Professor Department of Artificial Intelligence & Machine Learning, Genba Sopanrao Moze College of Engineering, Pune
[2,3,4] Student, Department of Artificial Intelligence & Machine Learning Genba Sopanrao Moze College of Engineering, Pune

Peer Review Information

Type: *Article*
Received:*23 February 2026*
Revised: *24 March 2026*
Accepted: *22 April 2026*
Published: *20 May 2026*

Abstract

The proliferation of deepfake media, generated by sophisticated generative adversarial networks (GANs), poses significant threats to digital trust, privacy, and information integrity. This paper presents a lightweight deepfake face detection system that leverages handcrafted facial features and a binary logistic regression classifier, deliberately avoiding convolutional neural networks (CNNs) to achieve real-time performance on resource-constrained hardware. The proposed pipeline extracts Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and geometric facial landmark metrics (e.g., Eye Aspect Ratio) from video frames using OpenCV and DLib, concatenating them into a unified feature vector fed into a gradient-descent-trained logistic regression model. The system is deployed via a Flask web interface enabling browser-based inference. Evaluation on standard benchmarks demonstrates approximately 90% accuracy on FaceForensics++ and 80–85% on Celeb-DF, with inference running at over 30 FPS on a standard CPU. The approach substantially outperforms CNN-based methods in training speed and inference efficiency while achieving competitive detection accuracy on moderate-quality deepfakes. Limitations due to absent temporal modeling and sensitivity to compression artifacts are discussed along with directions for future enhancement.

Keywords: Deepfake Detection; Handcrafted Features; Histogram of Oriented Gradients (HOG); Local Binary Patterns (LBP); Facial Landmarks; Logistic Regression; Flask; FaceForensics++; Lightweight Detection

How to Cite This Article

Chaudhari, L., Bansode, D., Patil, P., & Magdum, S. (2026). *DeepFake Face Detection with Handcrafted Features and Logistic Regression*. *International Journal on Advanced Computer Theory and Engineering*, 15(2s), 241–246.

Introduction

The rapid advancement of generative models, particularly Generative Adversarial Networks (GANs), has enabled the creation of highly convincing synthetic face-swap videos, commonly referred to as deepfakes. These manipulations, once confined to well-equipped research labs, are now accessible through consumer-grade applications, enabling widespread misuse for disinformation, fraud, non-consensual synthetic media, and political manipulation [6]. The societal consequences include erosion of trust in digital media, compromised personal reputations, and challenges to evidentiary reliability in legal contexts.

Automated deepfake detection has therefore emerged as a critical research priority. The majority of recent detection systems rely on deep convolutional neural networks (CNNs) that achieve high benchmark accuracy by learning complex spatial and temporal representations from millions of training samples [4]. However, CNN-based systems demand large computational budgets, specialized GPU hardware, and extensive training time, rendering them impractical for deployment in real-time or resource-limited environments such as mobile devices, edge nodes, or low-bandwidth video streaming platforms.

This paper presents an alternative approach: a lightweight deepfake detection pipeline centered on handcrafted facial features and a binary logistic regression classifier. By extracting well-established visual descriptors—Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and geometric facial landmark metrics—and training a linear model via gradient descent, we achieve competitive detection performance with dramatically reduced computational overhead. The system is integrated into a Flask-based web interface, enabling zero-installation, browser-based inference. This work demonstrates that classical computer vision techniques retain meaningful utility in the deepfake detection landscape, especially for deployment scenarios where efficiency and interpretability are prioritized over marginal accuracy gains.

Literature Survey

The deepfake detection literature spans two broad paradigms: classical handcrafted approaches and deep learning-based methods, each with distinct trade-offs in accuracy, generalizability, and computational cost.

Traditional and Handcrafted Approaches

Early deepfake detection exploited low-level visual artifacts specific to face manipulation pipelines. Texture-based descriptors such as LBP and HOG have been widely applied to capture unnatural pixel distributions in synthetic regions [1]. Facial landmark analysis—examining geometric consistency in eye positions, mouth shape, and contour alignment—has demonstrated effectiveness in detecting coarse manipulations where GAN generators fail to preserve anatomical plausibility. Feature-fusion strategies combining HOG, LBP, and keypoint-derived metrics with classical classifiers (SVM, Random Forest, Logistic Regression) have reported accuracies approaching 92% on FaceForensics++ under controlled conditions [2]. These methods offer fast training, interpretable decisions, and CPU-deployable inference, but remain sensitive to compression artifacts and novel manipulation techniques outside their training distribution.

Deep Learning-Based Approaches

State-of-the-art detectors predominantly leverage CNN architectures. XceptionNet, adapted for forgery detection, achieves detection accuracy exceeding 95% on FaceForensics++ by learning hierarchical spatial features [4]. Recurrent architectures and attention mechanisms have extended detection to temporal consistency modeling, exploiting inter-frame artifacts that static methods cannot capture. Ensemble approaches combining spatial CNNs with frequency-domain analysis have further pushed benchmark performance. However, these systems are computationally expensive: training can require tens of hours on multi-GPU clusters, and inference on standard CPUs runs at single-digit frame rates, making real-time deployment infeasible.

Lightweight Hybrid Methods

Recent work has revisited lightweight detection as a practical alternative. Yasir and Kim [2] demonstrated that a multi-feature fusion approach combining HOG, LBP, and gradient-boosted trees achieves 92–96% accuracy across FaceForensics++ and Celeb-DF datasets. AlMuhaideb et al. [8] showed that compact architectures such as MobileNet combined with GRU layers can achieve competitive accuracy with substantially smaller parameter counts. These studies motivate our work, which pushes further toward simplicity by employing a purely linear classifier on handcrafted descriptors, prioritizing deployment efficiency over marginal accuracy improvements.

Table 1: Comparison of deepfake detection approaches

Method	**Accuracy**
XceptionNet [4]	>95% (FF++)
Multi-Feature Fusion [2]	92-96% (FF++)
LightFakeDetect [8]	~91% (FF++)
Proposed (Ours)	~90% (FF++)

Methodology

The proposed system implements a four-stage pipeline: face detection, feature extraction, classification, and web-based deployment. Each stage is designed for efficiency and modularity, enabling frame-level deepfake classification without neural network components.

Face Detection and Alignment

Input video frames or uploaded images are processed through OpenCV's Haar Cascade face detector to localize facial regions. Detected faces are cropped, aligned to a canonical orientation using eye keypoints derived from DLib's 68-point landmark predictor, and resized to a standardized 224×224 pixel patch. Color normalization (mean subtraction, variance scaling) is applied per-frame to mitigate photometric variation. This alignment step is critical for ensuring that HOG descriptors and landmark geometry are computed in a consistent reference frame, reducing spurious feature variation due to pose differences.

Feature Extraction

Three complementary feature sets are extracted from each aligned face patch:

- HOG Descriptors: Gradient orientation histograms are computed over the face image using a cell size of 8×8 pixels and block normalization with 2×2 cell blocks. HOG captures macro-level edge and shape patterns, exploiting the tendency of GAN-synthesized faces to exhibit unnaturally uniform gradient distributions in hair, skin boundary, and eye regions.

- LBP Texture Features: Local Binary Patterns are computed over the grayscale face patch using a radius of 1 pixel and 8 neighboring points. LBP encodes micro-texture by comparing each pixel to its neighbors, producing a rotation-invariant texture histogram that is sensitive to the fine-grained pixel inconsistencies introduced by face synthesis and blending artifacts.

- Facial Landmark Metrics: DLib's 68-point predictor localizes anatomical keypoints including eye corners, brow edges, nose tip, and mouth boundaries. Derived metrics include Eye Aspect Ratio (EAR), mouth aspect ratio, inter-ocular distance normalized by face width, and jaw contour symmetry score. These geometric features detect spatial inconsistencies in landmark placement that often persist in GAN outputs due to limitations in 3D-consistent face generation.

The three feature vectors are concatenated into a single unified descriptor, which is standardized using z-score normalization prior to classification to ensure balanced gradient magnitudes during logistic regression training.

Classification Model

A binary logistic regression classifier maps the concatenated feature vector to a probability score $P(\text{fake} \mid x) \in [0,1]$. The model is trained by minimizing binary cross-entropy loss via mini-batch gradient descent with the L-BFGS optimizer (quasi-Newton method), implemented using scikit-learn's LogisticRegression with the 'lbfgs' solver. L2 regularization is applied to prevent overfitting on the high-dimensional feature space. The decision threshold is set at 0.5 for binary classification; post-hoc threshold tuning on a validation set can be applied to optimize the precision-recall trade-off for specific deployment contexts.

The gradient descent update rule is $\mathbf{w} \leftarrow \mathbf{w} - \boldsymbol{\alpha}\nabla\mathbf{J(w)}$, where α is the learning rate and $J(w)$ denotes the cross-entropy loss over the training batch.

Web Interface

The complete pipeline is served via a Flask web application. Users upload video files or images through a browser interface; the backend extracts frames, detects and aligns faces, computes features, and returns per-frame predictions with probability scores. Results are displayed with annotated bounding boxes and a frame-level verdict (Fake / Real). The Flask design ensures zero client-side installation and cross-platform accessibility.

Dataset And Preprocessing

We evaluate the system on three standard deepfake detection benchmarks that collectively cover a spectrum of manipulation quality and video diversity:

- FaceForensics++ [4]: A large-scale benchmark comprising 1,000 original videos and their manipulations produced by four methods (Deepfakes, Face2Face, FaceSwap, NeuralTextures), yielding over 1.8 million manipulated frames. FF++ is the primary training and evaluation dataset due to its scale and diversity of manipulation types.
- DFDC (DeepFake Detection Challenge) [5]: The largest publicly available face-swap video dataset, containing approximately 100,000 video clips with hundreds of actors. DFDC provides high visual quality and diverse demographics, making it a challenging generalization benchmark.
- Celeb-DF [6]: A dataset of 5,639 celebrity face-swap videos generated using an improved synthesis pipeline, producing more realistic manipulations than earlier benchmarks. Celeb-DF tests robustness to high-quality fakes that are difficult to distinguish visually.

Preprocessing follows a standardized protocol: frames are sampled at 2–5 FPS to balance coverage and computational load. Each detected face is aligned (eye-level), resized to 224×224, and color-normalized. HOG and LBP features are computed on the grayscale patch. Landmark coordinates are normalized by face bounding box dimensions to achieve scale invariance. The full feature pipeline is applied consistently across train and test splits to avoid data leakage.

Table 2: Dataset statistics used for evaluation

Dataset	**Quality**	**Challenge**
FaceForensics++ [4]	Mixed	Scale
DFDC [5]	High	Diversity
Celeb-DF [6]	High	Realism

Results And Evaluation

Detection Performance

Table 3 summarizes detection accuracy and ROC-AUC scores on held-out test sets for each benchmark dataset.

Table 3: Detection performance across benchmark datasets

Dataset	**Accuracy**
FaceForensics++	~90%
Celeb-DF	80–85%
DFDC (subset)	~78%

The proposed system achieves approximately 90% accuracy and an AUC of 0.90 on FaceForensics++, competitive with multi-feature fusion methods [2] and only marginally lower than CNN-based detectors. Performance degrades on Celeb-DF (80–85%), reflecting the increased visual realism of that benchmark's manipulation pipeline. On the DFDC subset, accuracy drops further to approximately 78%, attributable to heavy video compression that degrades HOG and LBP descriptor quality.

Computational Efficiency

A key advantage of the proposed method is its computational profile. Table 4 contrasts training and inference efficiency against representative CNN-based systems.

Table 4: Computational efficiency comparison

Metric	***Proposed Method***
Training Time (100k samples)	*~5 minutes (CPU)*
Inference FPS (CPU)	*>30 FPS*
GPU Required	*No*
Model Size	*<1 MB*

The proposed system trains in approximately five minutes on a standard CPU using 100,000 training samples, compared to 8–12 hours for XceptionNet on GPU hardware. Inference exceeds 30 FPS on CPU, confirming real-time deployability. Model size is under 1 MB, enabling deployment on mobile or embedded systems. These efficiency gains come at a cost of approximately 5–10 percentage points in accuracy relative to state-of-the-art CNN detectors on the hardest benchmarks.

Failure Analysis and Limitations

Several systematic failure modes were identified during evaluation. First, heavily compressed video frames degrade HOG and LBP descriptor quality, as compression artifacts mask the subtle manipulation traces these descriptors target. Second, the absence of temporal modeling means that frame-level predictions are independent; a sequence mixing real and fake frames may yield inconsistent verdicts. Third, the method exhibits sensitivity to partial occlusion, extreme head poses, and very low face resolution, all of which compromise feature extraction quality. Finally, the model generalizes poorly to novel GAN architectures not represented in the training data, a known limitation of feature-based approaches that do not benefit from deep representations learning directly from data.

Research Gaps and Future Scope

Current Limitations

The present system addresses moderate-quality deepfake detection under controlled conditions. Several capability gaps limit its applicability to production-grade scenarios. Temporal consistency modeling is absent; sequential analysis of facial motion, blinking patterns, and expression dynamics remains unexploited. The feature set is static—frequency-domain descriptors (DCT, DFT), color channel inconsistency metrics, and compression noise analysis are not incorporated. The system does not support audio-visual consistency detection, which has shown promise for identifying mouth movement to speech mismatches.

Identified Research Opportunities

Several directions represent meaningful research contributions. First, feature augmentation combining HOG/LBP with frequency-domain descriptors (e.g., high-frequency noise patterns specific to GAN up-sampling artifacts) may close the accuracy gap to CNN-based methods without sacrificing efficiency. Second, integrating dimensionality reduction via PCA before logistic regression training could improve generalization by filtering noise from the high-dimensional concatenated

feature vector. Third, temporal aggregation strategies—majority voting, Bayesian sequential updating, or lightweight recurrent post-processing of frame-level scores—could substantially improve sequence-level detection reliability. Fourth, adversarial augmentation during training (exposing the classifier to compressed, blurred, or noise-added fake samples) may improve robustness to the distribution shifts

observed in challenging benchmarks like DFDC.

Future Enhancement Directions

Planned extensions to the system include: (1) incorporating color histogram and discrete cosine transform (DCT) features to complement existing texture descriptors; (2) applying PCA-based dimensionality reduction to the concatenated feature vector, trading marginal information loss for improved generalization; (3) implementing a hybrid architecture using a compact deep feature extractor (e.g., MobileNetV2 penultimate layer) as an additional feature source while retaining the logistic regression classifier, combining interpretability with improved representational capacity; (4) extending the Flask interface to support real-time webcam inference and batch video processing with frame-level timeline visualization. Additionally, federated learning approaches for training on sensitive institutional datasets without data sharing represent a promising direction given the privacy-critical nature of face manipulation detection.

Conclusion

This paper presented a lightweight deepfake face detection system based on handcrafted facial features and logistic regression, demonstrating competitive performance against more complex approaches while achieving substantial advantages in training speed, inference efficiency, and deployment accessibility. The system achieves approximately 90% accuracy on FaceForensics++, runs at over 30 FPS on a standard CPU without GPU hardware, and requires under 1 MB of model storage, making it viable for real-time, resource-constrained deployment scenarios where CNN-based systems are impractical.

The work confirms that classical computer vision techniques—HOG, LBP, and facial landmark geometry—retain meaningful discriminative power for deepfake detection, particularly against moderate-quality manipulations. The Flask-based web interface ensures zero-installation deployability, broadening accessibility beyond the research community.

Future work will focus on addressing the identified limitations—particularly temporal inconsistency detection and robustness to compression—through feature augmentation and lightweight hybrid architectures. The trade-off between detection accuracy, computational efficiency, and model interpretability explored in this work offers a complementary perspective to the dominant CNN-centric paradigm, with practical relevance for edge deployment, educational tools, and resource-limited detection systems.

Acknowledgement

The authors express gratitude to Professor Parinita Walivadekar for guidance and mentorship throughout the development of this work. We acknowledge the Department of Artificial Intelligence and Machine Learning, Genba Sopanrao Moze College of Engineering, Pune, for providing the necessary infrastructure and computational resources. We thank the open-source communities behind OpenCV, DLib, scikit-learn, and Flask whose tools formed the foundation of this system, and the research teams who publicly released the FaceForensics++, DFDC, and Celeb-DF benchmark datasets.

References

1. Aslam et al., "Extracting Facial Features to Detect Deepfake Videos Using Machine Learning," Int. J. Advanced Computer Science and Applications, vol. 16, no. 4, pp. 834–845, 2025.
2. S. M. Yasir and H. Kim, "Lightweight Deepfake Detection Based on Multi-Feature Fusion," Applied Sciences, vol. 15, no. 4, p. 1954, 2025.
3. S. Banerjee, S. K. Yadav, A. Dhara, and M. Ajij, "A Survey: Deepfake and Current Technologies for Solutions," in Proc. 6th Int. Doctoral Symposium on Intelligence Enabled Research (DoSIER), Jalpaiguri, India, Nov. 2024.
4. Rössler et al., "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2019, pp. 1–11.
5. Dolhansky et al., "The DeepFake Detection Challenge (DFDC) Dataset," arXiv preprint arXiv:2006.07397, 2020.
6. 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. CVPR, 2020, pp. 3207–3216.
7. V. Dorau et al., "Advances in DeepFake Detection Algorithms: Exploring Fusion and Handcrafted Features," Signal Processing, vol. 187, 2021, pp. 1–17.
8. AlMuhaideb et al., "LightFakeDetect: A Lightweight Model for Deepfake Detection in Videos," Mathematics, vol. 13, no. 19, 2025, Article 3088.