

A Review of AI-Based Face-Swap Deepfake Video Detection and Suspicious Frame Analysis

¹Tazmeen Rashid Karbhari, ²Sabiha Shabbir Chikte, ³Radha Subodh Anerao
⁴S. Sankareswari

^{1,2,3}*Department of Information Technology Finolex Academy of Management and Technology
Ratnagiri, India*

⁴*Professor, Department of Information Technology Finolex Academy of Management and
Technology Ratnagiri, India*

Abstract—Face-swap deepfakes are now easy to make and hard to spot by eye, and they are already being used for misinformation, impersonation, and fraud. This paper reviews deep-learning methods for detecting such videos: benchmark datasets, frame-level CNN detectors, artifact-based and physiological-signal methods, and models that add temporal information. Detection is clearly possible, but results depend heavily on the dataset, the compression level, and the manipulation method, and most detectors return only a final label with no hint of where the manipulation is. Based on these gaps, we describe a prototype that classifies faces frame by frame, combines the frame scores into a Real/Deepfake decision with a confidence value, and shows the most suspicious frames on a web dashboard. We trained a baseline CNN, Xception, and EfficientNetB0 on the same data split and compared them. EfficientNetB0 gave the highest accuracy (59.61%) and ROC-AUC (0.6419) at about one-third of Xception’s inference time, so it was chosen for the prototype, although Xception had higher recall and F1-score. The scores are preliminary and leave clear room for improvement.

Index Terms—Deepfake Detection, Face-Swap, FaceForensics++, EfficientNet, Convolutional Neural Network, Transfer Learning, Suspicious Frame Localization, Web Dashboard

I. INTRODUCTION

Face-swapping tools have improved to the point where a manipulated video can pass for real footage. In a face-swap deepfake, one person’s face is placed onto another person’s body while the rest of the video stays untouched, so the clip appears to show someone doing or saying something they never did. Such videos are easy to produce and hard to verify, and they can be

used for misinformation, impersonation, fraud, and harassment. Trust in video as evidence is becoming a real concern.

Researchers have responded with large benchmark datasets [1]–[4], CNN-based detectors [5], [6], methods that look for the blending and warping artifacts left by face-swapping [8], and models that add temporal information [16]–[19]. Even so, a detector that scores well on one dataset can fail on another dataset, a different compression level, or a manipulation method it has not seen. A bare Real/Fake label is also of limited use to a person who has to decide whether to trust it. In this paper we review the main lines of work and use the gaps we find to shape a prototype detection system.

II. REVIEW METHODOLOGY

We looked at published work on face-swap deepfake detection and selected papers that dealt either with detecting manipulated face videos or with building a usable detection system. Each paper was read for its aim, its technique, its main contribution, and its limitations, and the papers were then compared to find limitations they had in common. Those shared limitations guided the prototype described in Section VI. Only a limited number of papers have been covered so far, and the review will be extended before submission.

III. LITERATURE REVIEW

A. Benchmark Dataset for Facial Manipulation Detection (ICCV, 2019)

Rössler et al. [1] built FaceForensics++, a public benchmark covering four manipulation methods (DeepFakes, Face2Face, FaceSwap, and NeuralTextures) at several compression levels. It holds over 1.8 million manipulated images and keeps its test set hidden. The authors found that domain-specific knowledge helps forgery detection even under heavy compression, and that trained detectors clearly beat human observers. The benchmark extends the earlier FaceForensics video dataset [2]. Its limitation is that performance on unseen manipulations and real-world conditions remains hard to guarantee.

B. Challenging Deepfake Datasets (CVPR, 2020)

Li et al. [3] noticed that earlier datasets looked nothing like the deepfakes circulating online, and released Celeb-DF, with 5,639 high-quality deepfake videos of celebrities. Dolhansky et al. [4] followed with the DeepFake Detection Challenge (DFDC) dataset: over 100,000 clips from 3,426 paid actors, made with several deepfake, GAN-based, and non-learned methods. They concluded that deepfake detection is extremely difficult and still unsolved. The lesson from both is that a detector has to be judged on data it was not trained on.

C. Frame-Level CNN Detection

Afchar et al. [5] proposed MesoNet, two compact networks with few layers that look at mesoscopic image properties, and reported detection rates above 98% for DeepFakes and 95%

for Face2Face. XceptionNet [6] is a widely used CNN backbone for this task, while EfficientNet [7] is a family of CNNs scaled to balance accuracy and computational cost. We chose EfficientNet for the prototype because of that balance. Frame-level detectors classify each face image on its own, which keeps them simple, but they ignore how the video changes over time.

D. Artifact-Based Detection (CVPRW, 2019)

Li and Lyu [8] observed that the face-warping step in deepfake generation leaves resolution inconsistencies, and trained a CNN to pick up on them. Other researchers have relied on different cues: eye blinking [9], capsule networks [10], inconsistent head poses [11], two-stream networks for tampered faces [12], person-specific cues for protecting world leaders [13], and training on self-blended images [14]. These methods mostly work on single frames, so they make limited use of the temporal information in a video.

E. Physiological-Signal-Based Detection (TPAMI, 2020)

Ciftci et al. [15] proposed FakeCatcher, which separates real from synthetic portrait videos using biological signals together with spatial-temporal patterns. It is a clever idea, but it is more complex than a plain frame classifier and depends on video quality and on how clearly the face is visible, which makes it harder to deploy.

F. Deepfake Detection Using Recurrent Neural Networks (AVSS, 2018)

Güera and Delp [16] extracted frame-level features with a CNN and passed them to a recurrent network that decides whether the whole video has been manipulated. Tested on deepfake videos collected from several video websites, the pipeline gave competitive results with a simple architecture. Many later spatiotemporal methods follow this CNN-plus-recurrent-network pattern, using an LSTM [27] or a GRU [28] for the temporal part.

G. An Integrated Spatiotemporal-Based Methodology for Deepfake Detection (Neural Computing and Applications, 2022)

Ismail et al. [17] join CNN/Xception spatial features with recurrent temporal modeling. Combining the two kinds of information helps, but it costs more computation and makes the model harder to deploy and to interpret.

H. Deepfake Detection via Spatial–Temporal Deep Networks (Discover Applied Sciences, 2026)

Jugade et al. [18] compared several CNN backbones with LSTM temporal modeling on FaceForensics++ using a range of evaluation metrics. Their results support adding temporal modeling on top of CNN features. Generalization across datasets and efficient deployment are still left open.

I. Robust Deepfake Video Detection Using Spatio-Temporal Features and Dynamic Difference Learning (Scientific Reports, 2026)

AbdElfattah et al. [19] pair spatial and temporal analysis in one detector. Facial landmarks from Dlib feed a Transformer encoder that models facial motion, a Dynamic Difference Module highlights abrupt changes between frames, and VGG16 CNNs with LSTMs capture spatial features and temporal dependencies. Evaluated on FaceForensics++, UADFV, and DFDC, the framework reports 100% accuracy on FaceForensics++ and DFDC and 99.6% on UADFV. These are strong numbers, but the design has several parallel branches, and the authors themselves say that further validation in real-world deployment settings is needed.

J. Surveys and Face-Swap Generation Tools

Verdoliva [20] and Tolosana et al. [21] give broader overviews of media forensics, face manipulation, and fake detection. On the generation side, Face2Face [22] performs real-time face capture and reenactment, deferred neural rendering [23] synthesizes images from neural textures, and DeepFaceLab [24] is an open-source face-swapping framework. Knowing how face-swap videos are made helps explain why artifacts appear around the face and why they differ from tool to tool.

IV. COMPARATIVE ANALYSIS

The studies reviewed here cover different parts of the problem: building benchmarks, classifying single frames, detecting artifacts, reading physiological signals, and modeling change over time. They differ in aim, in technique, in computational cost, and in how well they hold up outside their own dataset.

Rössler et al. [1] supply the benchmark on which many detectors are tested, and Li and Lyu [8] concentrate on warping artifacts in single frames. Ciftci et al. [15] take a different route through physiological signals. Ismail et al. [17], Jugade et al. [18], and AbdElfattah et al. [19] all combine spatial features with temporal modeling, building on the CNN-plus-recurrent structure introduced by Güera and Delp [16].

Taken together, the studies show that detection works, but three things stand out. Most report only a final decision, which gives an analyst little to check. The spatiotemporal methods tend to be heavier to run. And performance reported on one dataset does not automatically carry over to other datasets, compression levels, or unseen manipulation methods. What is missing is a simple detection workflow that classifies frames, gives a confidence score for the whole video, and shows which frames triggered the decision.

Table I. Comparative Analysis of Reviewed Studies

Ref.	Application	Main Technique	Major Contribution	Major Limitation
[1]	Benchmark Dataset	FF++ with four manipulation methods	Studies compression and data-driven detection	Generalization to unseen manipulations

[8]	Artifact-Based Detection	CNN on face-warping artifacts	Detects warping artifacts	Mainly spatial; limited temporal use
[15]	Physiological-Signal Detection	Biological signals + spatial-temporal patterns	Real/synthetic separation	Complex; sensitive to video quality
[17]	Spatiotemporal Detection	CNN/Xception + recurrent modeling	Joint spatial and temporal features	Complexity; deployment and interpretability
[18]	Spatiotemporal Detection	CNN backbones + LSTM	Backbone comparison on FF++	Cross-dataset generalization
[19]	Spatiotemporal Detection	Landmarks + Transformer, Dynamic Difference Module, VGG16 + LSTM	Motion and spatial cues combined; strong reported results	Multi-branch; real-world validation needed

V. CHALLENGES AND RESEARCH GAP

Deep learning has clearly moved deepfake detection forward, but several problems get in the way of using these detectors in practice.

The first is generalization. A detector trained on one dataset may do much worse on another dataset, on a different compression level, or on a manipulation method it has never seen. The second is interpretability. A single Real/Deepfake label gives an analyst very little to inspect, whereas a system that also points to suspicious frames is easier to verify. The third is cost: spatiotemporal models improve detection but are heavier to train and run, and physiological-signal methods are sensitive to video quality and face visibility. Finally, a detector that lives in a research notebook is hard for ordinary users to apply, so a simple workflow from video upload to result is needed.

The gap we address is therefore a practical one. We want a lightweight detector, built on a single well-chosen backbone, that turns frame-level scores into a video-level confidence value and shows the suspicious frames through a web dashboard.

VI. PROTOTYPE METHODOLOGY

The prototype is built around a frame-level EfficientNet classifier. A user uploads a video through a web dashboard; the system extracts frames, detects and preprocesses the faces, scores each face with the EfficientNet model, and combines the scores into a video-level decision. The flow is: Video → Frame Extraction → Face Detection → Preprocessing → EfficientNet Model → Frame-Level Analysis → Video-Level Result → Dashboard, as shown in Fig. 1. An LSTM or

GRU stage can be added between frame-level analysis and the video-level result as an optional extension.

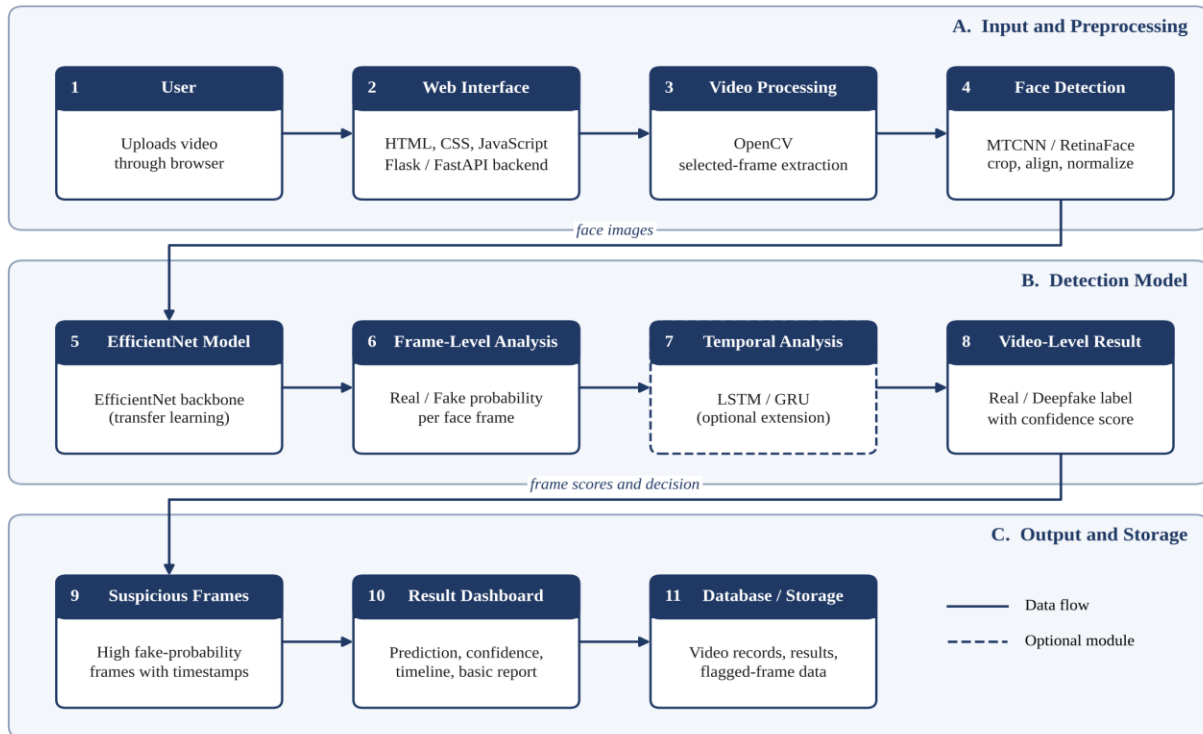


Fig. 1. Proposed system architecture of the face-swap deepfake video detection system. The dashed box marks an optional extension.

A. Dataset Preparation

Real and manipulated videos, mainly from FaceForensics++ [1], are collected and split into training, validation, and test sets. Because the full dataset is large, a representative subset is used, sized to the available storage and compute. Videos that share a source are kept on the same side of the split so that the test results are not inflated.

B. Frame Extraction and Face Preprocessing

OpenCV extracts selected frames from each validated video, and every frame keeps its position in the video so that a suspicious frame can later be placed correctly on the timeline. Faces are found with MTCNN [25] or RetinaFace [26], then cropped, aligned, resized, and normalized, so the model looks at the region where face-swap manipulation actually happens. If no face is found, the system shows a warning and stops.

C. EfficientNet Frame-Level Classification

Each preprocessed face is passed to an EfficientNet model [7] that outputs a Real or Fake label with a probability score. We use transfer learning from pre-trained weights, which makes

training possible on limited GPU, CPU, RAM, and storage. EfficientNet was chosen because it is designed to give good accuracy at a modest computational cost.

D. Video-Level Decision and Suspicious-Frame Localization

The frame-level scores are combined into a single Real/Deepfake decision with a confidence value. Frames with a high fake probability are marked as suspicious and listed with their time stamps on a timeline. If the video is classified as Real, no frames are marked.

E. Optional Temporal Extension

If frame-level scoring alone proves insufficient, the sequence of frame features can be passed to an LSTM [27] or GRU [28] to capture inconsistencies that a single frame does not show. This stage is kept optional so that the benefit of temporal information can be measured against the frame-level baseline.

F. Dashboard and Data Storage

A web dashboard (HTML, CSS, JavaScript) with a Flask or FastAPI backend shows the prediction, the confidence score, the suspicious frames, and a basic detection report. Uploaded video records, results, and flagged-frame data are stored so that results can be reopened later. Privacy of uploaded videos is a stated requirement, and the output is meant as decision support, not as legal proof.

VII. EXPERIMENTAL EVALUATION

To check whether EfficientNet is a sensible choice for the prototype, we trained three frame-level models on the same data split and tested them on the same held-out data: a baseline CNN, Xception, and EfficientNetB0. Each model was scored on accuracy, precision, recall, F1-score, and ROC-AUC, and we also recorded the inference time per image and the total training time, since the prototype has to run on limited hardware. The results are given in Table II.

Table II. Performance comparison of the evaluated models (best value in each column in bold)

Model	Accuracy (%)	Precision	Recall	F1-score	ROC-AUC	Inference (ms/image)	Training time (s)
Xception	56.68	0.5322	0.6319	0.5778	0.6128	266.86	1878.36
EfficientNetB0	59.61	0.6111	0.3819	0.4701	0.6419	89.50	640.30
CNN	53.09	0.0000	0.0000	0.0000	0.4604	35.30	868.77

EfficientNetB0 reached the highest accuracy (59.61%), precision (0.6111), and ROC-AUC (0.6419). It also trained fastest (640.30 s) and needed about 89.50 ms per image, roughly one-third of the time Xception took (266.86 ms) and about one-third of its training time (1878.36 s). Xception, on the other hand, had the best recall (0.6319) and F1-score (0.5778), which means it

caught more of the fake frames, at a much higher computational cost. The baseline CNN was the fastest at inference (35.30 ms) but was not usable as a detector: its accuracy was the lowest (53.09%), its ROC-AUC (0.4604) was below chance, and it produced no true positives, so its precision, recall, and F1-score are all zero.

Taken together, EfficientNetB0 gives the best balance of detection quality and cost among the three models, which is why it is used in the prototype. It is not the best on every metric, though: its recall is clearly lower than Xception's, so it misses more fake frames. The absolute scores are also modest, with accuracy under 60% and ROC-AUC of 0.64, so these should be read as preliminary baseline results rather than as a finished detector. The suspicious-frame output has not yet been measured quantitatively, and a suitable measure of how well the flagged frames match the manipulated part of a video will be decided with the project guide.

VIII. FUTURE DIRECTIONS

The most obvious next step is to raise the detection scores. Possible routes are fine-tuning more of the EfficientNet layers, training on a larger share of the dataset for longer, using larger EfficientNet variants, and balancing the trade-off between precision and recall so that fewer fake frames are missed. The optional LSTM/GRU stage can then be added and compared with the frame-level baseline. Testing on Celeb-DF [3] and DFDC [4] would show how well the model generalizes to data it has not seen.

The system itself can be extended with explainability views such as Grad-CAM, a fuller detection report, and a status view for videos that are being processed. Audio manipulation and real-time streaming detection are outside the current scope and are left as later extensions. The literature review will also be widened and the research gap re-checked before submission.

IX. CONCLUSION

The reviewed studies show that deep learning can detect face-swap deepfakes, whether through CNN-based, artifact-based, or spatiotemporal methods. They also show where current work falls short: most detectors give only a final label, and their results may not carry over to new datasets or unseen manipulation methods.

Based on these gaps, we designed a prototype that combines face detection and preprocessing, EfficientNet frame-level classification, a video-level decision with a confidence score, and a dashboard that shows suspicious frames. In our comparison of a baseline CNN, Xception, and EfficientNetB0, EfficientNetB0 gave the highest accuracy and ROC-AUC at a much lower inference and training cost than Xception, although with lower recall. The overall scores are still modest, so improving detection performance, adding temporal modeling, and testing on other datasets remain future work.

ACKNOWLEDGMENT

The authors would like to thank their project guide, Prof. S. Sankareswari, their industry mentor, and the faculty members for their guidance and support during this review and the design of the prototype. The authors also acknowledge the research papers and publicly available datasets that made this study possible.

REFERENCES

- [1] 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.
- [2] A. Rössler et al., "FaceForensics: A Large-Scale Video Dataset for Forgery Detection in Human Faces," arXiv:1803.09179, 2018.
- [3] 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.
- [4] B. Dolhansky et al., "The DeepFake Detection Challenge (DFDC) Dataset," arXiv:2006.07397, 2020.
- [5] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network," in Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS), 2018.
- [6] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in Proc. IEEE CVPR, 2017.
- [7] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. Int. Conf. Mach. Learn. (ICML), 2019.
- [8] Y. Li and S. Lyu, "Exposing DeepFake Videos by Detecting Face Warping Artifacts," in Proc. IEEE/CVF CVPR Workshops, 2019.
- [9] Y. Li, M.-C. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking," in Proc. IEEE WIFS, 2018.
- [10] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos," in Proc. IEEE ICASSP, 2019.
- [11] X. Yang, Y. Li, and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in Proc. IEEE ICASSP, 2019.
- [12] P. Zhou et al., "Two-Stream Neural Networks for Tampered Face Detection," in Proc. IEEE/CVF CVPR Workshops, 2017.
- [13] S. Agarwal et al., "Protecting World Leaders Against Deep Fakes," in Proc. IEEE/CVF CVPR Workshops, 2019.
- [14] S. Shiohara and T. Yamasaki, "Detecting Deepfakes with Self-Blended Images," in Proc. IEEE/CVF CVPR, 2022.

- [15] U. A. Ciftci, I. Demir, and L. Yin, “FakeCatcher: Detection of Synthetic Portrait Videos Using Biological Signals,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
- [16] D. Güera and E. J. Delp, “Deepfake Video Detection Using Recurrent Neural Networks,” in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, 2018.
- [17] A. Ismail et al., “An Integrated Spatiotemporal-Based Methodology for Deepfake Detection,” *Neural Comput. Appl.*, Springer, 2022.
- [18] Y. Jugade et al., “Deepfake Detection via Spatial–Temporal Deep Networks: Leveraging CNNs and LSTMs for Enhanced Accuracy,” *Discover Appl. Sci.*, Springer, 2026.
- [19] E. AbdElfattah, H. M. Mousa, A. Elsis, N. Mahmoud, et al., “Robust Deepfake Video Detection Using Spatio-Temporal Features and Dynamic Difference Learning,” *Scientific Reports*, vol. 16, Art. no. 16990, Jun. 2026, doi: 10.1038/s41598-026-53545-w.
- [20] L. Verdoliva, “Media Forensics and DeepFakes: An Overview,” *IEEE J. Sel. Topics Signal Process.*, 2020.
- [21] R. Tolosana et al., “DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection,” *Information Fusion*, 2020.
- [22] J. Thies et al., “Face2Face: Real-Time Face Capture and Reenactment of RGB Videos,” in *Proc. IEEE/CVF CVPR*, 2016.
- [23] J. Thies, M. Zollhöfer, and M. Nießner, “Deferred Neural Rendering: Image Synthesis Using Neural Textures,” *ACM Trans. Graph.*, vol. 38, no. 4, 2019.
- [24] I. Perov et al., “DeepFaceLab: Integrated, Flexible and Extensible Face-Swapping Framework,” *arXiv:2005.05535*, 2020.
- [25] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” *IEEE Signal Process. Lett.*, vol. 23, no. 10, 2016.
- [26] J. Deng et al., “RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild,” in *Proc. IEEE/CVF CVPR*, 2020.
- [27] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, 1997.
- [28] K. Cho et al., “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014.