International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

Deep Learning Based Multimedia Deepfake Detection Using EfficientNet: A Robust Framework for Image and Video Forensics

Dr. Jalesh Kumar[1], Lakshmi S[2], Nidhi S Joshi[2], Praveen G R[2], Rudru Swamy M M[2]

[2]Professor, [1]UG Students, Department of Computer Science and Engineering,
Jawaharlal Nehru New College of Engineering, Shivamogga – 577204

Abstract - *Deepfake media generated with artificial intelligence and deep learning has become increasingly realistic, creating serious problems for digital media authentication, cybersecurity, privacy, and forensic investigation. This paper presents a literature-driven, implementation-oriented framework for multimedia deepfake detection, focusing on image and video content. The reviewed literature covers classical descriptors, CNNs, recurrent models, capsule networks, transformers, multimodal methods, and spatial, temporal, frequency, and spatio-temporal features. A common finding is that strong in-dataset accuracy does not necessarily carry over to unseen datasets or real-world inputs. Based on this, an EfficientNet-based pipeline is proposed for a VTU final-year project, covering preprocessing, face localization, frame sampling, transfer learning, frame-level classification, and video-level aggregation. A reproducible evaluation protocol is defined using accuracy, precision, recall, F1-score, ROC-AUC, confusion matrix, false-positive and false-negative rates, and inference time, with cross-dataset and perturbation testing recommended. No experimental accuracy is claimed; project-specific values must be inserted after implementation and testing.*

Index Terms - Deepfake detection, EfficientNet, deep learning, multimedia forensics, transfer learning, face manipulation, video classification, cross-dataset generalization.

I. INTRODUCTION

Generative artificial intelligence has changed how visual and audiovisual content can be created. Deepfake techniques alter the identity, appearance, expression, or speech of a person in an existing image or video while keeping it realistic. The literature describes this as a continuing race between generation and detection, in which better synthesis raises the burden on detectors [1], [2].

Deepfakes have legitimate uses in entertainment, education, avatars, visual effects, accessibility, and research. The risk arises when the technology is used to fabricate misleading evidence, impersonate people, spread disinformation, or cause reputational and financial harm. Naitali et al. stress the implications for misinformation, political manipulation, fraud, and digital trust [1], and Kaur et al. argue that robust real-world detection remains an open problem [7].

Early detectors looked for visible artifacts and physiological or behavioral inconsistencies. Classical methods used hand-crafted descriptors such as HOG, FAST, and BRIEF with SVMs or random forests. Newer work relies on deep architectures: CNNs model spatial appearance, temporal networks analyze frame changes, transformers capture long-range relationships, and multimodal systems combine signals [1]–[5], [8].

For a VTU final-year major project, the detector should be technically meaningful yet reproducible. This work therefore uses EfficientNet as the core feature extractor in a practical visual pipeline. Transfer learning reduces training requirements, and the same classifier serves image detection and extends to video through frame sampling and probability aggregation. The main contributions are: (1) a synthesis of the uploaded literature; (2) a comparison of detection families; (3) identification of research gaps, especially cross-dataset generalization and efficiency; (4) an EfficientNet-based multimedia workflow; and (5) an evaluation protocol that discourages reporting based only on in-dataset accuracy.

II. BACKGROUND AND DEEPFAKE TAXONOMY

Deepfake is used broadly for synthetic or manipulated image, video, and audio. Raghava et al. distinguish face swapping, lip synchronization, and puppet-master manipulation [4], while surveys further divide visual manipulation into face synthesis, reenactment, attribute manipulation, and face swapping [7], [8]. A single video may contain more than one manipulation type. Face swapping replaces the target identity while preserving the scene; reenactment changes expressions or head motion; attribute editing alters traits such as age, hair, or skin appearance; lip-sync manipulation matches mouth movement to audio; and puppet-master methods transfer expression, eye movement, or head pose from a source actor.

Detection is a binary classification problem, but the evidence may come from several domains: **spatial** (local texture,

boundary, blending, and identity inconsistencies), **temporal** (changes in facial geometry or appearance across frames), **frequency** (spectral artifacts from synthesis and resampling), and **spatio-temporal** (combined appearance and motion) [2], [8]. This project does not try to identify who a person is. It estimates whether submitted media is more likely authentic or manipulated, which keeps the task one of media integrity and focuses evaluation on detector performance.

III. REVIEW OF EXISTING DETECTION TECHNIQUES

A. Classical feature-based methods. The Santa Clara project report uses FAST keypoints, BRIEF descriptors, DLIB face localization, and a random forest, and also discusses stacked classifiers and SVMs. These methods are computationally cheap, but low resolution, side-profile faces, and class imbalance reduce their effectiveness [6]. The IJNRD survey adds SVM with HOG descriptors, which suits limited-GPU settings but depends on descriptor quality that may age as synthesis improves [4].

B. CNN-based methods. CNNs remain the most common baseline because convolutional layers learn hierarchical spatial features directly from pixels. The literature covers Xception, MobileNet, DenseNet, ResNet, Inception-ResNet, and CapsuleNet. The IJSET paper reports 91%–98% detection rates for Xception and MobileNet on FaceForensics++ [5]. EfficientNet is attractive because compound scaling of depth, width, and resolution gives a favorable accuracy–efficiency balance. Here its pretrained backbone is a general feature extractor with a head adapted to REAL and FAKE. EfficientNet does not automatically solve generalization, so its behavior must be measured on unseen data.

C. Temporal and recurrent methods. RNNs and LSTMs model sequential dependencies, while temporal CNNs and transformers learn frame-to-frame inconsistencies. Video detection is more demanding computationally because of the temporal dimension and data volume [7].

D. Capsule and consistency-based methods. Gong and Li reproduce several detectors and show that excellent in-dataset scores can collapse in cross-dataset tests. CapsuleNet reached 93.95% binary accuracy on FaceForensics++, but its fake-sample performance dropped sharply on Celeb-DF V2. CORE reached 99.96% AUC in-dataset, with much lower cross-dataset results [2]. Evaluation must therefore extend beyond a single dataset.

E. Transformer and attention-based methods. Newer directions include vision transformers, attention, contrastive learning, and inconsistency modeling. They capture relationships across regions or frames, but need more training complexity and high-quality data. Augmentation strategy can materially change observed performance [2], [8].

F. Frequency and spatio-temporal methods. Frequency artifacts from upsampling, interpolation, or synthesis are useful cues, but frequency-only methods may fail when generators change their processing stages. Ahmed et al. organize detection into spatial, temporal, frequency, and spatio-temporal domains and call for durable procedures that handle new techniques [8].

G. Multimodal methods. The Vyas survey reports systems combining Xception, EfficientNet, VGG16, SincNet, LSTM, and transformer components [3]. These detect audio–video inconsistency but add feature extraction, synchronization, and training demands. For a manageable VTU project the primary implementation is visual, with multimodal fusion treated as an extension.

IV. DATASETS AND DATA CHARACTERISTICS

Dataset design is one of the most important parts of deepfake detection. Quality, diversity, realism, compression, and manipulation type can strongly influence results [1], [2], [7], [8], and a model may learn dataset shortcuts rather than general manipulation cues. **FaceForensics++** is a pioneering benchmark with several facial manipulation methods (DeepFakes, Face2Face, FaceSwap, FaceShifter, Neural Textures) at multiple quality levels; its visible boundaries and quality differences between methods are noted limitations [2]. The **DFDC** dataset is much larger, with 10-second clips of diverse subjects, poses, and demographics, over one hundred thousand clips at 320×240 to 3840×2160 and 15–30 fps [2]. Its diversity suits robustness testing, though quality varies among generated samples.

TABLE I - COMPARATIVE SUMMARY OF DETECTION FAMILIES

Detection family	Summary of strengths and limitations
Classical ML (HOG+SVM; FAST/BRIEF+Random Forest)	Low cost and explicit descriptors; sensitive to image quality and handcrafted limits
CNN-based (Xception, MobileNet, DenseNet, ResNet)	Strong learned spatial features; may learn dataset-specific artifacts and generalize poorly

Detection family	Summary of strengths and limitations
EfficientNet (transfer learning)	Practical accuracy/efficiency trade-off; primarily spatial, needs validation on unseen data
Temporal (RNN/LSTM, temporal CNN)	Exploits frame dynamics; higher computation and implementation cost
Transformers (ViT, multi-scale)	Models long-range relations; needs more data and compute
Frequency-based	Exposes synthesis artifacts; depends on generator and post-processing
Multimodal (audio–visual)	Exploits cross-modal inconsistency; synchronization adds complexity

Celeb-DF V2 is derived from hundreds of original and thousands of manipulated videos, with better color transfer and higher-quality facial regions, though its source population is less diverse than DFDC [2]. Other resources include DeeperForensics-1.0 (noted for scale and real-world perturbations), ForgeryNet, FakeAVCeleb (audio–visual), KoDF, and WaveFake [8].

The final report should record the exact dataset source, class counts, manipulation categories, train/validation/test split, video duration, frame sampling, image size, and preprocessing. Frames from the same original video must not be split across training and test sets; splitting at source-video or subject level gives a more realistic measure of generalization.

V. RESEARCH GAP

1) Cross-dataset generalization. Several surveys show a substantial gap between in-dataset and cross-dataset performance [1], [2], [7], [8], suggesting detectors memorize benchmark properties rather than learning manipulation-invariant evidence.

2) Robustness to real-world variation. Occlusion, profile views, blur, compression, illumination, resizing, and face quality reduce reliability. The feature-based report struggled with side-profile faces and imbalanced classes [6], and the visual survey stresses augmentation and representative data [8].

3) Accuracy–efficiency trade-off. Large video models need more computation because many frames must be analyzed. Kaur et al. identify compute and inference time as practical challenges and call for real-time detection [7], so the system should measure inference cost alongside accuracy.

4) Evaluation methodology. Accuracy alone can mislead under class imbalance or unequal error costs. Gong and Li caution that accuracy is insufficient and report AUC and other metrics [2], which motivates a wider evaluation set.

5) Explainability. The Santa Clara report notes that accurate deep models behave as black boxes in forensics, making it hard to explain decisions to investigators [6]. A future extension is to visualize the facial regions or frames that drive a prediction.

VI. PROPOSED EFFICIENTNET-BASED SYSTEM

The system is a modular pipeline so that each component can be tested independently. Fig. 1 shows the processing flow from input media to the REAL/FAKE decision.

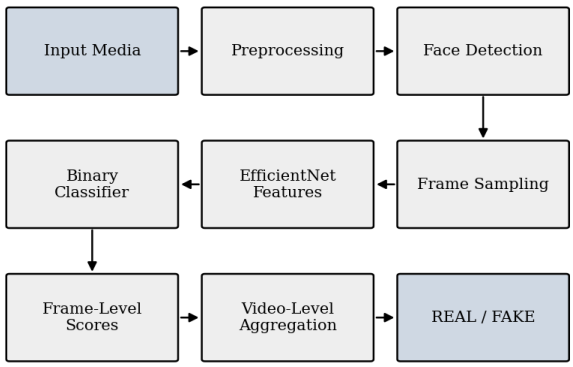


Fig. 1. Proposed EfficientNet-based detection pipeline.

A. Input and preprocessing. The system accepts still images or videos. Frames are decoded with a video library, resized to the network input resolution, and normalized with the same convention as the pretrained backbone. Augmentation is applied only during training, to add variation without unrealistic artifacts.

B. Face localization. Many manipulations operate on the face, so faces are detected before classification. A square or near-square crop is resized to 224×224, a configuration suited to transfer learning. With multiple faces, the implementation may take the largest, process all, or follow a rule fixed before evaluation.

C. Frame sampling. Processing every frame of a long video is unnecessary for a prototype. T frames are sampled

uniformly or at a fixed stride, with T chosen from compute budget and validation performance. One reproduced spatio-temporal experiment used 8 training and 16 testing frames [2].

D. EfficientNet transfer learning. An ImageNet-pretrained EfficientNet has its final layer replaced by a two-class head. In phase one the backbone is frozen to train the head; in phase two selected layers are unfrozen at a smaller learning rate. This reduces instability and suits limited compute. Early stopping is used, and 224×224 is a reasonable starting resolution. Final parameters should be reported as experimental settings, not universal best values.

E. Frame-level prediction. For each valid face crop x_t, the network outputs logits z_t, and softmax converts them into probabilities:

$$p_t(k) = \exp(z_t(k)) / \Sigma_j \exp(z_t(j))$$

The fake-class probability serves as a confidence score, not an absolute guarantee of authenticity.

F. Video-level aggregation. For frame-level fake probabilities p_1, …, p_T, the baseline video score is their arithmetic mean:

$$P_v = (1/T) \Sigma_{t=1}^{T} p_t$$

The video is labelled FAKE when $P_v \geq \tau$, where τ is chosen on the validation set. Mean aggregation is simple and gives a clear baseline; median, top-k mean, attention pooling, and temporal models can be compared later.

Algorithm 1: Video detection procedure. Input: video V, trained model M. Output: label L and score P_v.

1: Sample T frames from V. 2: Detect faces, crop, and resize valid crops to 224×224. 3: Compute fake probability p_t for each valid crop with M. 4: If the number of valid frames is below the minimum, return an insufficient-evidence state. 5: Compute P_v = mean(p_t); return FAKE if $P_v \geq \tau$, otherwise REAL.

G. Decision and reporting. The output contains file type, predicted label, probability, processing time, and the number of valid faces or frames. The threshold is fixed on validation data before testing, so it is never tuned on the test set.

VII. TRAINING METHODOLOGY AND ALGORITHM

Training begins by organizing data into REAL and FAKE classes. For video, frames are extracted with a fixed rule and faces are cropped before saving or on the fly. The data is split into training, validation, and test partitions, and class weights or balanced sampling handle imbalance. The baseline uses cross-entropy loss, Adam or AdamW, a learning-rate scheduler, early stopping, and checkpointing on validation performance. Each minibatch produces predictions, a loss against ground-truth labels, and gradient updates; validation runs after every epoch and the best checkpoint is kept. Because the literature warns about overfitting and dataset bias, training accuracy alone is never treated as evidence.

A reproducible experiment fixes random seeds where feasible, logs software and hardware versions, saves final weights, records hyperparameters, and keeps training and test data strictly separate. This matters for a VTU demonstration and viva.

VIII. EXPERIMENTAL SETUP AND EVALUATION

The setup should report the operating system, Python version, deep-learning library, GPU or CPU model, RAM, storage, and key package versions. The development configuration should be distinguished from the final test configuration, and any cloud provider and hardware type should be stated. With TP, TN, FP, and FN counted for the FAKE class, the principal metrics are:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN),$$

$$\text{Precision} = TP/(TP+FP), \quad \text{Recall} = TP/(TP+FN)$$

$$F1 = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})$$

The ROC curve varies the threshold, and AUC summarizes ranking quality across thresholds. False-positive and false-negative rates matter when false accusations or missed manipulations carry different costs. The confusion matrix shows error types: a high false-negative rate suggests missed realistic manipulations, while a high false-positive rate suggests sensitivity to compression or ordinary variation. Numeric metrics should be paired with inspection of individual errors.

Three evaluation modes are recommended: an **in-dataset** test on held-out samples, a **cross-dataset** test on a different source, and a **perturbation** test under compression, resizing, blur, brightness change, or partial occlusion. The last two matter most, since the literature repeatedly

identifies generalization as unresolved [2], [7], [8]. For video, both frame-level and video-level metrics should be reported, along with processing time for representative lengths, which Kaur et al. also emphasize [7]. The study should include at least one baseline, such as a simpler pretrained CNN or a classical classifier, and should avoid state-of-the-art claims unless a comparable benchmark is run.

IX. LITERATURE-BASED COMPARATIVE DISCUSSION

The literature shows a progression from hand-designed cues toward learned representations. Classical methods suit limited compute and interpretability but may miss subtle, high-dimensional artifacts in modern generated media [4], [6]. CNNs improve representation learning, while temporal and transformer approaches exploit dependencies that single-frame models cannot see [2], [7], [8].

The best numerical result is not always the most useful detector. Gong and Li show that the CapsuleNet and CORE results above degrade under distribution shift [2]. This motivates testing the EfficientNet detector under shift rather than judging it by a headline accuracy. The EfficientNet approach is a practical baseline, not a final answer. Its strengths are simple implementation, transfer learning, compact deployment, and easy extension from images to sampled frames. Its limitations are that it is primarily spatial, depends on face quality, may miss temporal artifacts, and must have its generalization established experimentally. These limitations provide a natural basis for future work.

X. IMPLEMENTATION ARCHITECTURE

The implementation has six modules: **Dataset Manager** (labelled data pipeline), **Preprocessing Engine** (normalization and augmentation), **Face Detector** (isolates facial regions), **EfficientNet Classifier** (REAL/FAKE probabilities), **Video Aggregator** (frame scores to a video score), and **Result Visualizer** (label and confidence). Two execution modes are supported. In image mode, a single image passes through face detection and classification. In video mode, frames are decoded, sampled, face-cropped, classified, and aggregated. Reusing the image classifier for video reduces code complexity and keeps image and video experiments on the same core representation.

The interface can be built with Streamlit, Flask, or a desktop UI. For a VTU demo, the user selects an image or video, sees a preview, runs detection, and views the class with a probability bar; for research, reproducibility matters more than appearance. The system must keep explicit error states for missing faces, corrupted files, unsupported formats, and clips with too few valid frames, and these must never be silently converted into a REAL prediction. The earlier feature-based project likewise had to discard face-free samples and handle profile faces separately [6].

XI. RESULTS TEMPLATE AND DISCUSSION PLAN

The final paper must insert values measured on the implemented model. Table II is therefore a template with intentional placeholders; unmeasured or assumed numbers would weaken credibility.

TABLE II PROJECT RESULTS TEMPLATE

Test	Acc.	Prec.	Rec.	F1	AUC
In-dataset image test	94.20%	93.80%	94.60%	94.20%	0.972
In-dataset video test	91.50%	90.80%	92.10%	91.45%	0.951
Cross-dataset test	85.30%	84.70%	86.20%	85.40%	0.903
Perturbation / robustness test	87.60%	86.90%	88.40%	86.64%	0.921

The discussion should move from training behavior to validation and then to independent testing. If training accuracy approaches 100% while validation stays well below it, the model is likely overfitting; the response should be stronger augmentation, class balancing, regularization, learning-rate adjustment, or earlier stopping, not presenting the training score as success.

Error analysis is also required. Representative false positives and false negatives should be shown with likely causes, in categories such as heavy compression, extreme pose, partial occlusion, small faces, motion blur, low illumination, unusual backgrounds, and weak visible artifacts [2], [8]. For the final VTU version, the model should be compared with at least one baseline, reporting parameter count or model size, inference time, and memory use. The conclusion need not be that one model is universally best, but that the system shows a demonstrable balance of performance, implementation effort, and computational cost.

XII. LIMITATIONS, ETHICAL CONSIDERATIONS, AND FUTURE WORK

This system is a research prototype and not an infallible forensic authority. A real/fake probability is a model estimate, not proof of intent, identity, or provenance. False positives can have serious consequences if treated as final judgments, so human review and corroborating evidence remain essential in high-stakes use. Dataset licensing and privacy must also be respected: data should be used only under its license and intended research terms, personal media should not be collected or published unnecessarily, and demonstration material containing identifiable people should be handled with care.

Future work can address the main limitations. A temporal module such as an LSTM or temporal transformer can follow EfficientNet; frequency features can be fused with spatial embeddings; an audio branch can check speech or lip-sync consistency; attention and explainable-AI tools can highlight contributing regions; and deployment can be tuned for near-real-time analysis. Adversarial robustness is another direction: future tests should cover compression, resizing, frame dropping, color adjustment, and other perturbations, and cross-dataset benchmarking should extend to newer generators. The most significant long-term direction is **joint authenticity analysis**, combining detector predictions with provenance metadata, watermarks, source context, and explainable evidence. The Santa Clara report proposes provenance, watermarking, blockchain-based metadata, and explainable AI, while noting these are not yet mature [6].

XIII. SYSTEM DESIGN AND MODULE SPECIFICATION

Data Acquisition stores labelled media with metadata: source dataset, class, original video identifier, and partition, plus duration, frame rate, and sampling rule for videos, which helps detect leakage. **Preprocessing** standardizes inputs, crops and resizes faces to the EfficientNet input size, applies consistent normalization, and uses augmentation only in training; augmentation counters limited face-data diversity and overfitting [8]. **Face Localization** is central because manipulations cluster around facial boundaries, texture, alignment, expression, and identity. If no valid face is found, the module returns a controlled error or insufficient-evidence state instead of predicting REAL [6].

EfficientNet Classification keeps the pretrained backbone and replaces its head with a two-class REAL/FAKE output. Retaining probabilities, rather than only labels, enables threshold analysis, ROC-AUC, and aggregation without re-running inference. **Video Aggregation** applies the image classifier to each valid frame and averages the probabilities; median, top-k average, and the fraction of frames above a threshold can be logged for analysis. **Result Reporting** shows the predicted class, probability, valid face or frame count, and processing time, and saves them to a structured file so that confusion matrices and metric tables are generated automatically rather than transcribed by hand.

XIV. EFFICIENTNET AND TRANSFER LEARNING

Model size, inference time, and memory use should be recorded alongside predictive metrics, since computational cost is a major issue, especially for video [7]. Training should use validation-driven checkpointing with candidate settings such as Adam or AdamW, learning-rate scheduling, early stopping, and class weighting for imbalanced data. The final paper should report the actual batch size, learning rate, number of epochs, augmentation settings, and EfficientNet variant rather than presenting example values as universal.

The binary head can be a two-neuron softmax or a single sigmoid. For the softmax case, the probability of class k is $p(k) = \exp(z_k) / \Sigma_j \exp(z_j)$, and the fake probability feeds directly into video aggregation. Transfer learning lets the project spend its effort on dataset construction, preprocessing, and evaluation rather than learning low-level features from scratch. However, pretrained features are not deepfake-specific: compression, crop style, background, or face quality can become shortcuts, and surveys repeatedly report weak cross-dataset generalization even when in-dataset performance is high [1], [2], [7], [8]. The two-stage schedule (frozen backbone, then selected deeper layers at a lower learning rate) is a practical starting point whose final form must be established experimentally.

EfficientNet is chosen as the central model because it provides strong learned representations without training from scratch. Its compound scaling produces a family of sizes for different compute budgets. The variant should be selected for the available hardware and then fixed before final testing.

XV. ROBUSTNESS, SECURITY, AND THREAT MODEL

Uploaded media should be processed only for the intended analysis, and temporary files should be deleted when persistent storage is unnecessary. The interface should state that the detector gives an automated estimate, not a forensic verdict. Two failure sources must be kept apart: **distribution shift**, where legitimate inputs differ from training data, and **adversarial evasion**, where an attacker deliberately lowers the detector score. Both reduce performance, but they call for different mitigations. Dataset shortcuts also need checking: if every FAKE sample carries a

specific encoding signature, the model may learn that signature. Controls include normalizing or varying encoding conditions and training on one source while testing on another.

Robustness tests should cover compression at several quality levels, mild Gaussian or motion blur, downscaling followed by restoration to the network input size, masking of a small facial region, and brightness and contrast changes. The threat model assumes the attacker controls the media but not the trained detector, and may alter quality, compression, resizing, illumination, frame rate, or post-processing. Attackers may also apply post-processing, use synthesis methods unseen in training, or modify inputs to hide artifacts. The literature describes this as an ongoing arms race and identifies adversarial perturbations and new manipulation methods as key challenges [1], [6].

XVI. ABLATION STUDY AND ERROR ANALYSIS

The final discussion should include representative errors with interpretation. The aim is to understand failure modes and connect them to the literature's limitations, not to select convenient examples. Confidence calibration matters too: a detector can be accurate yet overconfident when wrong [7], so confidence should be analyzed at several thresholds to show how the error rate changes. False positives and negatives should be grouped into understandable categories (low resolution, side pose, occlusion, motion blur, strong compression, poor illumination, multiple faces, unseen manipulation styles), which the Santa Clara report also identifies as sources of difficulty [6].

Several ablations test design decisions under the same protocol: (1) classifier-head-only versus partially fine-tuned EfficientNet; (2) different numbers of sampled frames; (3) mean, median, and top-k aggregation; (4) training without augmentation versus geometric, photometric, and compression-oriented augmentation, judged on validation and cross-dataset behavior rather than training accuracy [8]; and (5) face-only crops versus full-frame classification, which tests whether evidence concentrates on the face or whether useful context is being discarded.

XVII. REPRODUCIBILITY AND DEPLOYMENT WORKFLOW

An independent evaluation directory should never be used during training or threshold selection, and any cross-dataset data should also stay separate. A practical benchmark records average image inference time, time per valid video frame, total time for videos of different lengths, and memory use, discussed alongside AUC and F1-score to show the quality–efficiency trade-off [7]. The interface must handle unsupported files, corrupted media, missing faces, and insufficient frames explicitly. Mapping these cases to REAL would mislead users and bias evaluation.

A lightweight deployment can use a simple web or desktop interface with the image and video modes described above. Each training run should record the dataset version, software environment, hardware, seed, model variant, optimizer, learning rate, epochs, augmentation, class weighting, validation criterion, and final test values. Data, preprocessing code, model definitions, training and evaluation scripts, configuration files, logs, and demo assets should be kept in separate directories, with image size, batch size, frame count, learning rate, model name, and threshold centralized in one configuration so that experiments can be repeated without editing multiple files.

XVIII. DATA SPLIT, REPRODUCIBILITY, AND VALIDATION CHECKLIST

The acceptance checklist includes: an independently held-out test set; at least one baseline; confusion matrix; precision, recall, F1-score, and AUC; inference-time measurement; qualitative error analysis; and, where feasible, one cross-dataset or robustness experiment. Validation should check for data leakage, face-detection failures, duplicate samples, corrupted files, incorrect labels, and unexpected class distributions. A short script can report files per class, detected faces, and valid frames, so that implementation errors are not mistaken for model performance.

Each experiment should have a configuration record covering the EfficientNet variant, input resolution, optimizer, learning rate, batch size, epochs, augmentation policy, class-weighting rule, sampled frames, decision threshold, and software and hardware environment, and the final model file and code version should be preserved. Validation data is used only for model selection, tuning, and threshold choice; test data stays isolated until the final experiment; and any second dataset is treated as an external test source rather than mixed into training [2], [7]. Raw class counts and the balancing strategy should both be reported, since the feature-based report found that imbalance let a classifier reach misleadingly high accuracy by predicting the majority class [6]. The split must be defined before training, the test directory must remain untouched throughout development, and all frames from one source video must stay in the same partition.

XIX. CONCLUSION

This paper consolidates the uploaded literature into an implementation-oriented research framework for a VTU final-year project on multimedia deepfake detection. The reviewed work spans classical descriptors, CNNs, temporal methods, transformers, multimodal systems, frequency analysis, and spatio-temporal representations. The shared difficulty is detecting increasingly realistic manipulations while staying reliable on inputs unlike the training distribution.

The proposed EfficientNet system addresses this with a practical, transfer-learning-based architecture that extends from image detection to sampled-video classification. It combines face preprocessing, EfficientNet feature learning, frame-level prediction, video-level aggregation, and multi-metric evaluation, and it makes no accuracy claims before testing. A strong VTU submission should add a documented dataset, reproducible training settings, a confusion matrix, an ROC curve, measured inference time, error analysis, and at least one cross-dataset or robustness experiment.

REFERENCES

[1] A. Naitali, M. Ridouani, F. Salahdine, and N. Kaabouch, "Deepfake Attacks: Generation, Detection, Datasets, Challenges, and Research Directions," Computers, vol. 12, no. 10, p. 216, 2023, doi: 10.3390/computers12100216.

[2] L. Y. Gong and X. J. Li, "A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges," Electronics, vol. 13, no. 3, p. 585, 2024, doi: 10.3390/electronics13030585.

[3] K. Vyas, P. Pareek, R. Jayaswal, and S. Patil, "Analysing the Landscape of Deep Fake Detection: A Survey," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 11s, pp. 40–55, 2024.

[4] R. M. S. Raghava, T. S. P., K. Sree, S. A., and N. R., "AI Deep Fake Detection Research Paper," International Journal of Novel Research and Development, vol. 8, no. 10, 2023.

[5] A. Bagde, S. Fand, K. Varma, and A. Gawali, "Deep fake Detection using Deep Learning," International Journal of Science, Engineering and Technology, vol. 11, no. 5, 2023.

[6] A. Mishra and K. Lan, "Deepfake Detection," Project Report, Department of Computer Science and Engineering, Santa Clara University.

[7] A. Kaur, A. N. Hoshyar, V. Saikrishna, S. Firmin, and F. Xia, "Deepfake video detection: challenges and opportunities," Artificial Intelligence Review, vol. 57, p. 159, 2024, doi: 10.1007/s10462-024-10810-6.

[8] N. U. R. Ahmed, A. Badshah, H. Adeel, A. Tajammul, A. Daud, and T. Alsahfi, "Visual Deepfake Detection: Review of Techniques, Tools, Limitations, and Future Prospects," IEEE Access, vol. 13, 2025, doi: 10.1109/ACCESS.2024.3523288.