Deepfake Image Detection: A Comparative Study of Deep Learning Models with Explainable AI Insights

Vivek Mule
Dept. of Computer Science and Engineering (Artificial Intelligence), Vishwakarma Institute of Information Technology, Pune, India,
vivekvmule@gmail.com

Aziz Sayyad
Dept. of Computer Science and Engineering (Artificial Intelligence), Vishwakarma Institute of Information Technology, Pune, India,
azizsayyed.499@gmail.com

Dev Bhangale
Dept. of Computer Science and Engineering (Artificial Intelligence), Vishwakarma Institute of Information Technology, Pune, India,
devbhangale.04@gmail.com

Prachee Prasad
Dept. of Computer Science and Engineering (Artificial Intelligence), Vishwakarma Institute of Information Technology, Pune, India,
pracheeprasad23@gmail.com

ABSTRACT

Deepfakes are increasingly becoming a major concern in today's digital era. With AI being able to produce incredibly realistic images and videos now, it's becoming difficult to distinguish between the real and fake which is a big issue. From circulating misinformation to destroying reputations and even facilitating identity theft — the implications can be gigantic. The goal of the research is to identify how good the deep learning models are in detecting deepfake images. Some potential and popular models such as ResNet18, ResNet50, InceptionV3, Vision Transformer (ViT), and EfficientNe have been evaluated, along with a custom created custom convolutional neural network (CNN) to compare its performance with other models. To avoid bias, all models are trained on the same similar parameters. For a few of the models, the data was augmented to see whether it assisted them in dealing with new, out-of-distribution images. Apart from identifying deepfakes, the study underlines the significance of being able to understand model decision-making through explainability methods. Explainable AI has been put into action via the use of the Integrated Gradients approach that emphasizes which portions of an image had the largest impact on the model's prediction — offering a glimpse behind the scenes as to what's occurring. Models have been assessed according to accuracy, precision, recall, and F1-score. Our comparative study is aimed at finding models that provide the optimal trade-off between accuracy and interpretability, so that real images are not incorrectly identified as fake, and manipulated content is consistently detected.

Keywords — Deepfake Detection, Explainable AI (XAI), Integrated Gradients, Data Augmentation, CNN and Transformer Models.

I. INTRODUCTION

Yet another important differentiator in our work is explainability. We leverage Integrated Gradients, an algorithm from the class of explainable AI algorithms, to display where in an image most contributed to the model's prediction. It reveals if the model is targeting relevant face regions or distracted by irrelevant noise. By so doing, we seek to narrow the gap between accuracy and interpretability—two attributes that are too often thought to be mutually exclusive.

We also investigate the performance of data augmentation in enhancing model capability and generalization. Deepfake datasets tend to lack variation, which may cause overfitting. Using augmentation processes such as flipping, cropping, and color jittering, we check whether models are more resilient and capable of distinguishing fakes from unknown origins.

Our evaluation criteria include standard classification metrics such as accuracy, precision, recall, and F1-score, along with visualizations from the explainability tools. By

analyzing the results from multiple angles—quantitative performance and qualitative interpretability—we offer a holistic assessment of how each model behaves and what trade-offs may arise when choosing one over another.

Fig.1 shows a demonstration of how an actual photo can be manipulated using face swap technology to create a doctored picture.

Fig 1. Face Swap Deepfake Image vs. Original Image

Briefly, this work not only adds to the expanding body of work in deepfake detection by comparing popular architectures under equal conditions, but also adds a degree of transparency with explainability methods. As deepfakes get more sophisticated, we think that building models that are both effective and comprehensible is the most important aspect of creating detection systems that people will trust to use in real-world, high-stakes applications.

Here's an idea of what the rest of the paper will be like: In Section II, we introduce some background context and mention previous research on deepfake generation, detection methods, and explainable AI (XAI). In Section III, we describe our approach — how we constructed and curated the dataset, selected and trained the models, and infused explainability within. Section IV is where we present our experiment results as well as a full breakdown and explanation. Section V then concludes by summarizing our conclusions and proposing some ideas on how things could be improved further — such as making the models more robust, using more heterogeneous datasets, and creating better visual explanations that are easier for people to read and apply in actual scenarios.

A second key differentiator in this work is the focus on explainability. Integrated Gradients, a member of the Explainable AI (XAI) family based on attribution techniques, is employed to identify and visualize where in an image most strongly a model will base its decision. This not only facilitates improved model behavior comprehension but also allows us to measure whether attention is being focused on helpful facial features or misleading background noise. Through the integration of such interpretability tools, the study aims to bridge the often-perceived divide between model transparency and accuracy—two objectives which are too often assumed to be mutually exclusive within deep learning studies.

The research also investigates the impact of data augmentation on model performance and generalization. Owing to the inherent data limitations in deepfake datasets—e.g., class imbalance and reduced diversity—models tend to overfit, reducing their ability to recognize new manipulations. Horizontal flip, random cropping, and color jittering are some of the augmentation techniques employed to enrich training data and increase the robustness of the models. This procedure helps to determine whether such techniques enhance the model's capacity to generalize beyond the training distribution, an aspect that matters for real-world detection applications.

Model performance is quantified using standard classification metrics: accuracy, precision, recall, and F1-score. These are complemented by visual outputs from the XAI tools, giving a richer insight into how each model perceives and interprets deepfake content. This dual strategy—combining quantitative performance with qualitative explainability—enables a full comparison and highlights the trade-offs of selecting a detection model that maximizes predictive capability while reducing interpretability.

Overall, this work contributes to the emerging area of deepfake detection by testing a number of state-of-the-art architectures under controlled settings and adding explainability to enable trust and accountability. As the threat of deepfakes continues to grow, the findings highlight the importance of developing detection systems that are not only technically superior but also transparent and comprehensible, especially in high-stakes applications like media forensics and digital authentication.

The remainder of this paper is structured as follows: Section II provides a literature review of deepfake generation, detection techniques, and Explainable AI usage. Section III outlines the experimental procedure, including dataset preparation, training the model, and integration of explainability techniques. Section IV presents the results and interprets the findings both numerically and through visual explanations. Section V concludes the study and recommends future research, such as enhancing model

robustness, generalizing to more varied datasets, and fine-tuning visual interpretability for real-world applications.

II. LITERATURE REVIEW

The ability to create extremely realistic fake videos and images through deepfake technology poses serious threats, such as misinformation and identity theft. Many proposed solutions aim at creating deep learning systems that can efficiently detect and classify deepfakes. This review highlights the research that has investigated various frameworks and approaches to detecting deepfakes.

One approach utilizes InceptionResNetV2 with LSTM networks for classification to derive language features. Two test accuracies were obtained on running the model, 84.75% and 91.48%, after 20 and 40 epochs respectively; validating more training epochs enhance performance [1].

And yet another study performed comparative analysis of four models on CNN— ResNet-152, MobileNetV3, ConvNeXt Large and EfficientNetB7 on the dataset of FaceForensics++. EfficientNetB7 had the highest test accuracy of 75% confirming its effectiveness towards deepfake detection [2].

Other research compared other models such as Xception, ResNet50, Swin Transformer, CNN, and MobileNet in detecting deepfakes. Xception and ResNet50 indicated high accuracy, precision, and recall, and indicated low gender bias. Swin Transformer, however, was effective in detecting fake images but ineffective in detecting real images where it indicated bias. It is recommended by the research to use an ensemble approach to utilize the strengths and weaknesses of each of the individual models [3].

In another approach, DenseNet-201 was utilized to detect image forgery—namely the copy-move and splicing types. The model was very good, achieving 94.12% accuracy and showing its ability to handle image transformations like scaling, rotation, and translation. Additionally, it has a light-weight architecture that is ideal for real-time processing, which is a convenient practical benefit [4].

But one approach was unique. It employed an Error Level Analysis (ELA) technique to identify tampered regions of an image. These regions were filtered with a CNN to extract features and were classified using SVM and KNN classifiers finally. The highest performance of up to 89. 5% accuracy was surprisingly achieved by combining a Residual Network with KNN. This combination of ELA and deep learning was very promising in the detection of tampered images [5].

With regard to explainability, more and more work is being aimed at making AI decisions more interpretable—particularly, in the context of deepfake detection. For example, Gowrisankar and Thing (2024) reported that conventional XAI evaluation methods, like pixel manipulation, do not work well for fake detection overall. They introduced another, more domain-specific approach based on adversarial attack-based evaluation, which seems more suitable for this specific application [10].

Mansoor and Iliev in their work in 2025 used network dissection methods on ResNet-50 and InceptionV3 models. This allowed them to visualize the internal decision-making of the models. Their approach, while it had explainability, proved to be resilient, with F1-scores of between 0. 8 and 0. 9. This is a good sign that it is achievable to make such models transparent without affecting performance [11].

Expanding the context, Ali et al. (2023) provide a wide-ranging overview of XAI approaches and highlight the importance of trust, accountability, and user understanding in AI systems. Their findings show that while many models have high accuracy levels, they are often black boxes, which can be problematic in sensitive areas like misinformation and security. In aggregate, these studies underscore the significance of explainability in deepfake detection and stress how the incorporation of techniques like adversarial evaluation and feature-level interpretation can lead to more trust-worthy and transparent AI solutions [12].

These experiments collectively indicate how advanced deepfake detection technologies have become, particularly with so many various deep learning models that exist. They also indicate that the combination of various approaches could be the path to creating accurate and robust systems.

III. METHODOLOGY

Fig 3.1 depicts a streamlined pipeline for deepfake detection, combining robust preprocessing, flexible model training, and integrated explainability. Key steps include:

Preprocessing & Splitting: Faces are detected, cropped, aligned, and the dataset is divided into training, validation, and test sets.

Data Augmentation: Training images undergo lightweight transformations (flips, contrast tweaks, noise) to improve generalization.

Model Training: A selected backbone (ResNet-18/50, InceptionV3, EfficientNet-B0, or ViT) automatically extracts hierarchical features and learns to classify real versus fake.

Evaluation & Saving: Performance is tracked via accuracy, precision, recall, and F1-score; the best checkpoint is exported for inference.

Inference & Explainability: New inputs are labeled "Real" or "Fake," and saliency maps highlight facial regions that most influenced each decision.

This end-to-end approach ensures both high detection accuracy and transparent, interpretable results.

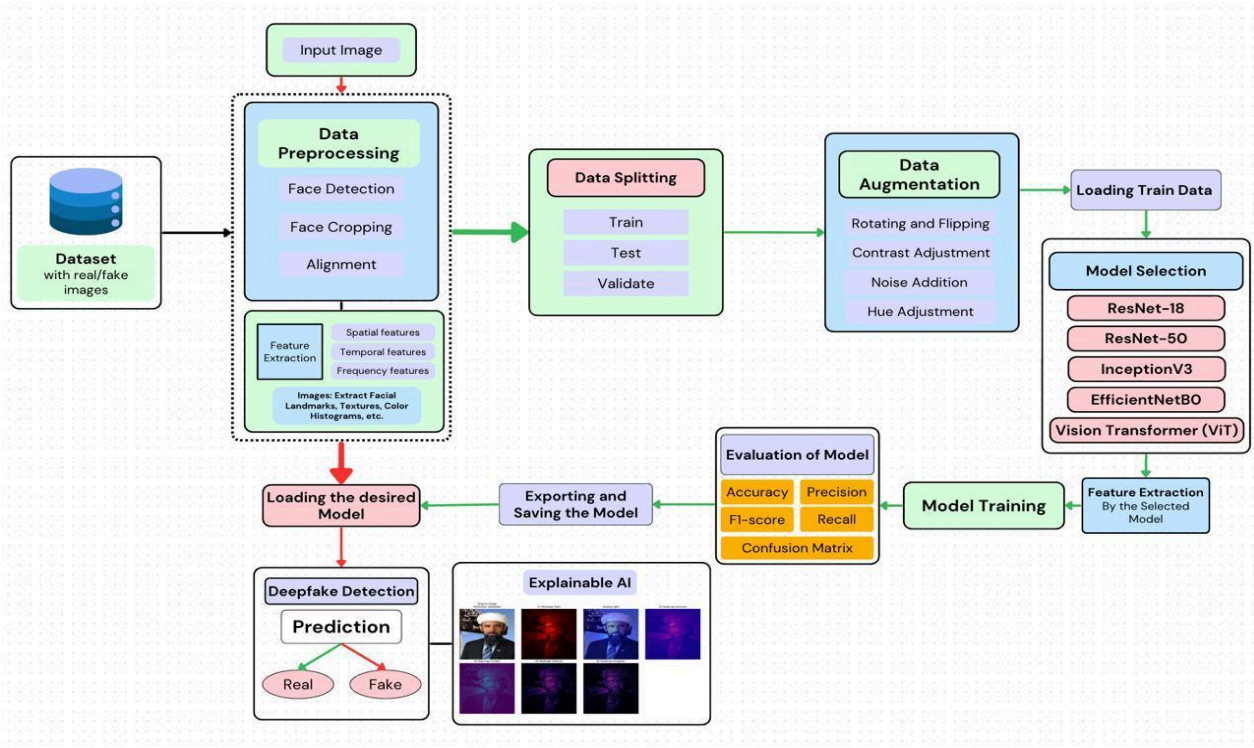

Fig 2. Proposed System Architecture

Overview:

This study investigates deepfake image detection by developing and comparing eight deep-learning models, trained on two complementary datasets with and without data augmentation. This comparative study leverage five architectures—ResNet-18, ResNet-50, a custom CNN, EfficientNet (B0), and Vision Transformer (ViT)—with extensive augmentation, and three architectures—InceptionV3, ResNet-18, and ResNet-50—without augmentation.Furthermore, we incorporated Explainable AI (XAI) via Integrated Gradients to interpret model decisions at the pixel level. Figure 1 illustrates the overall workflow, from data ingestion through model explainability.

A) Dataset Statistics

Two distinct datasets were utilized to ensure robustness and generalizability, all the images in both the dataset are of 256x256 size.

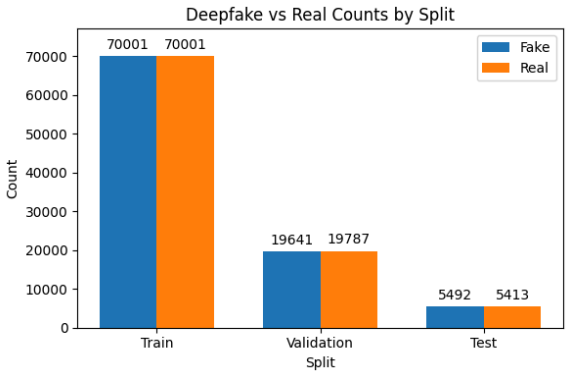

Fig 3. Dataset Splits

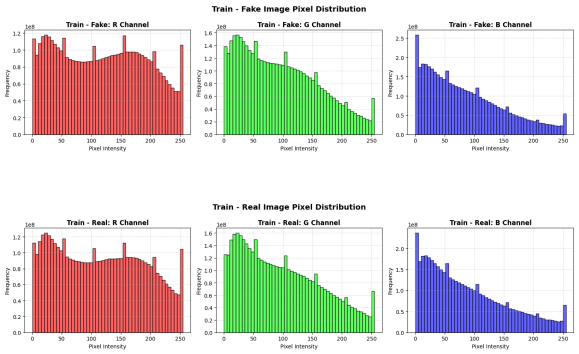

Fig 4. Train Split

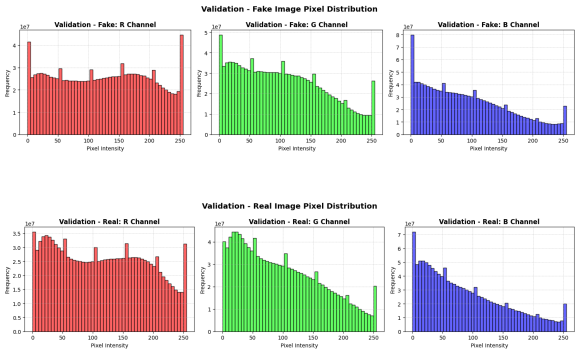

Fig 5. Validation Split

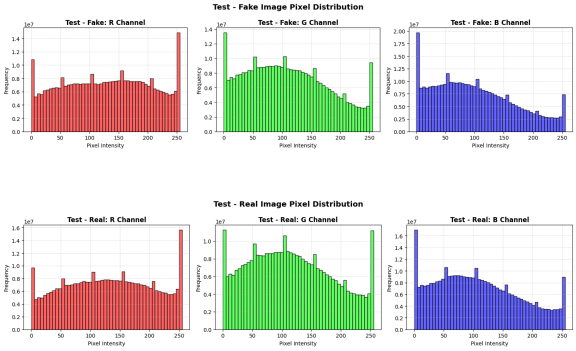

Fig 6. Test Split

A.1. Dataset for Custom CNN

Source: [15]

The dataset used in this study was divided into three distinct subsets to ensure effective model training and evaluation: 121,159 images were allocated for training, 53,184 images for validation, and 35,304 images for testing.

The classification task is binary, with each image labeled as either Real or Fake. All images are in RGB format and originally came in various resolutions.

To maintain consistency across models and facilitate batch processing, every image was resized to a standard resolution of 224×224 pixels.

A.2. Dataset for Other Pretrained Models

Source: [16]

The dataset was split into three subsets to ensure balanced and effective model training and evaluation. The training set consisted of 140,002 images, evenly divided between 70,001 real and 70,001 fake samples. The validation set included 39,428 images, with 19,787 real and 19,641 fake images. Finally, the test set comprised 10,905 images, nearly balanced with 5,413 real and 5,492 fake samples. This balanced distribution helped maintain fairness across all stages of the model evaluation pipeline.

Classes & Preprocessing: As above; images resized to 299×299 for InceptionV3, 256×256 for others.

All images were normalized channel‑wise to zero mean and unit variance using training‑set statistics. Class imbalance is inherently balanced in both datasets.

B) Data Preprocessing and Augmentation

B.1. Without Data Augmentation

For the three models—InceptionV3, ResNet-18, and ResNet-50—trained without augmentation, preprocessing was kept minimal to isolate model performance from the effects of data diversity. Images were resized according to model requirements: 299×299 for InceptionV3 and 256×256 for ResNet variants. Pixel values were scaled to the [0,1] range and then standardized. A batch size of 32 was used, and no random transformations were applied.

B.2. With Data Augmentation

For the five models trained with augmentation (ResNet-18, ResNet-50, custom CNN, EfficientNet-B0, ViT), we implemented on-the-fly stochastic transforms using TensorFlow's tf.keras.layers and tf.image utilities:

Data augmentation included a 50% chance of horizontal flip, random rotations up to ±20°, and zoom variations of ±20% to enhance model generalization.

To improve model generalization and handle real-world image variations, we applied data augmentation techniques including ±30% contrast and hue shifts, ±40% brightness changes, and saturation adjustments ranging from 0.6 to 1.6. These augmentations helped enhance robustness and reduce overfitting.

Augmentation is applied on a per-image basis with a 50 % chance: for each batch sample, a random draw determines whether to augment. If selected, a single transformation is chosen uniformly from the predefined set via an integer index and applied; otherwise, the original image is passed through. This conditional logic is implemented using TensorFlow's tf.cond and tf.switch_case, ensuring minimal overhead in the input pipeline. Finally, all images—augmented or not—are fed through the standard preprocessing layer and cast to float16 before model ingestion.

C) Model Architectures

C.1. ResNet-18 and ResNet-50

Backbone: Standard residual blocks

Modifications:

1. Final fully-connected layer replaced with a single output neuron (sigmoid activation) for binary classification.
2. Batch Normalization and Dropout (p=0.5) before the classifier.

C.2. InceptionV3

Backbone: Inception modules

Adaptation: Global average pooling, followed by a Dense layer (1 neuron, sigmoid).

C.3. EfficientNet-B0

Backbone: Compound-scaled MBConv blocks

Adaptation: As above; input resolution 224×224.

C.4. Vision Transformer (ViT-Base)

Backbone: 12 Transformer encoder layers, 16×16 patch size, 768-dim embeddings

Classification Head: CLS token fed through a Dense-sigmoid layer.

C.5. Custom CNN

Layers:

Conv(32, 3×3) → ReLU → MaxPool(2×2)

Conv(64, 3×3) → ReLU → MaxPool(2×2)

Conv(128, 3×3) → ReLU → MaxPool(2×2)

Conv(256, 3×3) → ReLU → MaxPool(2×2)

Conv(512, 3×3) → ReLU → MaxPool(2×2)

Flatten → Dense(512) → ReLU → Dropout(0.5) → Dense(1) → Sigmoid

Rationale: Lightweight architecture to test custom feature learning on the smaller Hugging Face dataset.

D) Training Protocol

D.1. Common Settings

The models were trained using the Binary Cross-Entropy loss with logits, suitable for binary classification tasks. The Adam optimizer was used with parameters β_1=0.9 and β_2=0.999. Initial learning rates of 1e-3 and 1e-4 were explored, with a ReduceLROnPlateau scheduler to

adaptively reduce the learning rate based on validation AUC (factor: 0.5, patience: 2 epochs, minimum learning rate: 1e-7). Training was done with a batch size of 32. Experiments were conducted using both Google Colab's free-tier TPU and GPU environments, as well as Kaggle Notebook GPUs. The models were implemented using PyTorch, TensorFlow, and Keras frameworks.

D.2. Without Augmentation

Training was conducted for up to 20 epochs, with early stopping implemented to prevent overfitting—training halted if the validation AUC did not improve for 3 consecutive epochs. Model checkpointing was used to save the version of the model that achieved the highest validation AUC during training.

D.3. With Augmentation

Training was carried out for a maximum of 50 epochs. Early stopping was applied with a patience of 5 epochs based on validation AUC, ensuring training stopped if no improvement was observed over 5 consecutive epochs. Model checkpointing was used to retain the model that achieved the highest validation AUC.

Note: Longer training to allow models to see diverse augmented samples.

E) Evaluation Metrics

We evaluate all models on a held-out test set using several key metrics. The primary measure of performance is the Area Under the ROC Curve (AUC), which reflects the model's ability to distinguish between real and fake images. We also report Accuracy and Precision based on thresholded predictions at 0.5. To better understand the types of errors each model makes, we analyze the confusion matrix. Additionally, to assess the statistical significance of performance differences between models, we conduct paired t-tests on their Accuracy scores over five independent runs using different random seeds, varying the decision threshold between 0.3 and 0.7.

F) Explainable AI via Integrated Gradients

To interpret model decisions and validate that predictions rely on meaningful facial features rather than artifacts, we implement Integrated Gradients (IG), the approach follows the given steps:

F.1. Integrated Gradients (IG) Computation

1. Baseline x′: Here, x′ is an all-zeros image of the same shape as the input x.

2. Interpolation: generate m+1m+1m+1 interpolated images

$$x^{(i)} = x' + (i / m) \times (x - x'), \text{ for } i = 0,1,\ldots,m \quad (1)$$

3. Gradient Computation: for each $x^{(i)}$, compute the gradient of the model's scalar output $F(x^{(i)})$ with respect to the input.

$$g^{(i)} = \nabla_x F(x^{(i)}) \quad (2)$$

4. Average Gradient: approximate the path integral by averaging.

$$\bar{g} = (1 / m) \times \Sigma\, g^{(i)} \quad (3)$$

5. Integrated Gradients: multiply the average gradient by the input difference.

$$IG(x) = (x - x') \odot \bar{g} \quad (4)$$

F.2. Heatmap Normalization & Gamma Correction

Once you have the raw IG tensor $IG(x) \in \mathbb{R}^{H \times W \times 3}$:

1. Absolute and Sum across Channels:

$$A_i = \Sigma\, |IG(x)_i \subset| \quad (5)$$

2. Min–Max Normalization to [0,1]:

$$\hat{A}_i = (A_i - \min(A)) / (\max(A) - \min(A)) \quad (6)$$

3. Gamma Correction (to emphasize stronger attributions):

$$G_i = \hat{A}_i \hat{} \gamma \quad (\gamma = 0.7) \quad (7)$$

F.3. Visualization

1. Raw Heatmaps: Mapped through various colormaps (hot, plasma, viridis, etc.)

$\text{colormap}(G_i)$ applied to each G_i

2. Overlay: Blend original image I and colored heatmap H via

$$O = (1 - \alpha) \times I + \alpha \times H, \alpha = 0.5 \quad (8)$$

3. Display: Arranged in a 2×4 grid using Matplotlib subplots.

G. Summary

We systematically compare CNN and transformer-based models, with and without data augmentation, and use Integrated Gradients for interpretability, enabling fair performance evaluation and insight into deepfake detection decisions.

IV. RESULTS

A. Overview of Model Performance

In this section, multiple models of deep training used to detect deep functions are carefully evaluated. The increase in data and non -academies has been used to teach models. Performance was evaluated using metrics such as accuracy, maximum loss, ROC AUC, accuracy curve, correction and confusion matrix.

B. Comparative Performance of Models

Table 1: Model Performance without Data Augmentation

Model	Epochs Trained	Accuracy	Final Loss	Early Stopping	Best Epoch	LR Reduced
ResNet18	**9 / 20**	**0.9278**	**0.2227**	**Yes (at 9)**	**7**	**No**
ResNet50	13 / 20	0.8800	0.4508	Yes (at 13)	11	No
InceptionV3	9 / 20	0.9267	0.2035	Yes (at 9)	7	No

From **Table 1**, it is evident that **ResNet18** and **InceptionV3** achieved the highest accuracy (92.78% and 92.67% respectively) without data augmentation. Their early stopping around epoch 9 indicates efficient convergence and robustness. In contrast, **ResNet50** underperformed slightly in comparison (88.00%).

Table 2: Model Performance with Data Augmentation

Model	Epochs Trained	Accuracy	Final Loss	Early Stopping	Best Epoch	LR Reduced
CustomCNN	**30 / 30**	**0.8625**	**0.4261**	**No**	**30**	**No**
EfficientNetB0	20 / 20	0.8456	0.3979	No	20	Yes
ResNet50 (Fine-tuned)	30 / 30 + 9 / 15[1]	0.8386	0.8210	No / Yes (at 9)	4 (Fine-tuned)	Yes
ViT	20 / 20	0.8576	0.2248	No	20	No

Table 2 shows that among models trained **with data augmentation**, **CustomCNN** and **ViT** yielded the best performance (86.25% and 85.76% respectively). However, overall accuracy was marginally lower than their non-augmented counterparts. This may be due to **increased data variance** or **over-regularization** introduced by augmentation techniques.

Overall, This highlights a key observation: augmentation may not always benefit models equally, especially when the original dataset already contains sufficient variation.

C. ROC Curve Analysis

To evaluate the discriminative capability of the deep learning models, ROC (Receiver Operating Characteristic) curves were plotted. These curves illustrate the trade-off between the **True Positive Rate (TPR)** and the **False Positive Rate (FPR)** at various classification thresholds. A higher area under the ROC curve (AUC) indicates better model performance in distinguishing between real and deepfake images.

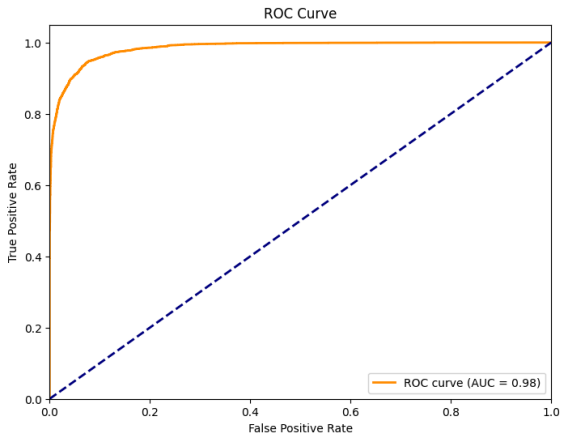

Figure 7. ROC Curve — ResNet18 (Without Augmentation)

Figure 7 Shows ROC curve for **ResNet18** evaluated without data augmentation. The model achieved an **AUC score of 0.98,** demonstrating near-perfect discrimination between positive and negative classes. The sharp curve inflection reflects highly confident predictions.

Insight: An **AUC of 0.98** signifies an exceptionally strong classifier, with minimal overlap between the distributions of true positives and false positives.

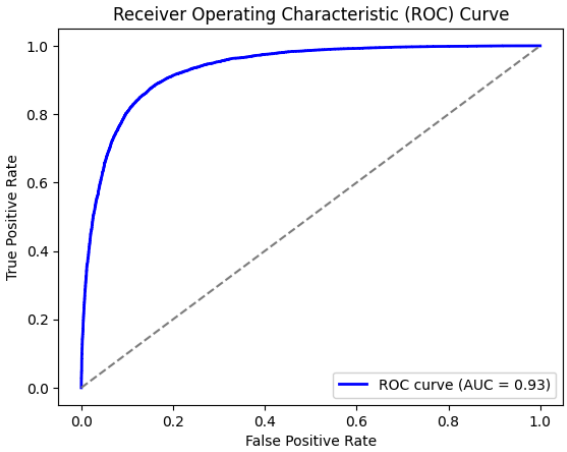

Figure 8: ROC Curve — CustomCNN (With Augmentation)

Figure 8 shows the ROC curve for the **CustomCNN** model trained using data augmentation. With an **AUC score of 0.93,** the model demonstrates strong ability to tell apart real and deepfake images. This result suggests that applying data augmentation helped the model learn better and improved its accuracy in classification tasks.

Insight: The ROC curve analysis reveals that **CustomCNN** performs exceptionally well in distinguishing between the two classes, with an **AUC score of 0.9**3, indicating **strong predictive accuracy.** The steep initial rise in TPR indicates strong early detection capability, while the slight curvature suggests minor confidence reduction in borderline cases.

Overall, models without augmentation (e.g., **ResNet18**) achieve higher AUCs, reflecting strong feature learning on consistent data, while augmented models (e.g., **CustomCNN**) perform reliably but show slight AUC drops due to variability introduced by augmentation.

D. Precision-Recall (PR) Curve Analysis

Precision-Recall (PR) curves are particularly important when working with imbalanced datasets, such as real vs. deep fake images, where one class may dominate. These curves illustrate the balance between **precision (the accuracy of positive predictions)** and **recall (the ability to detect true positives),** providing a more meaningful assessment than ROC analysis in cases where the positive class is underrepresented.

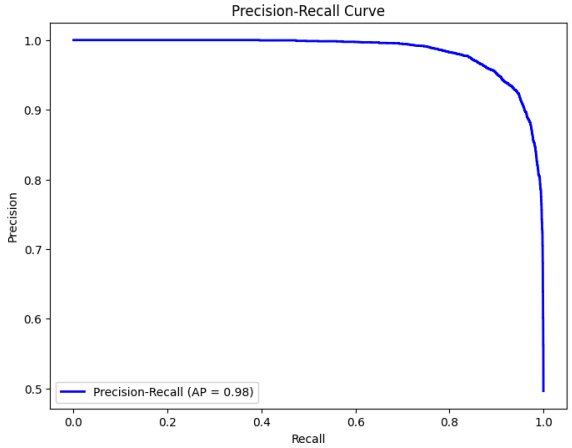

Figure 9: PR Curve — ResNet18 (Without Augmentation)

Figure 9 Displays the precision-recall curve for **ResNet18,** the top-performing model tested on the original dataset (without augmentation). With an **Average Precision (AP) score of 0.98**, the model maintains near-perfect precision at varying recall levels, making it highly effective for tasks where precision is paramount.

Insight: The consistently high precision **(AP=0.98)** indicates that **ResNet18 delivers exceptionally reliable positive predictions** while rarely producing false positives. This is particularly crucial in sensitive domains like deepfake detection, where even minor classification errors can have significant consequences.

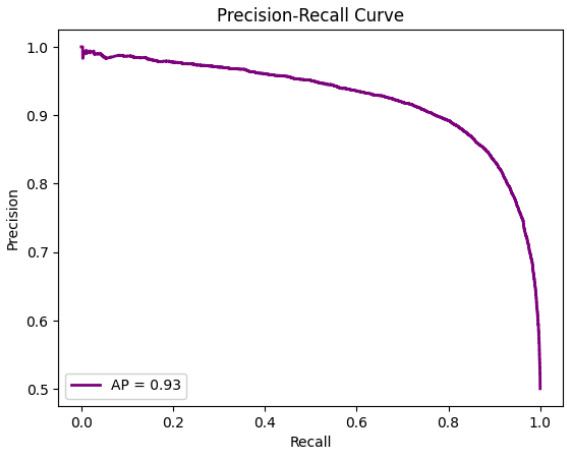

Figure 10: PR Curve — CustomCNN (With Augmentation)

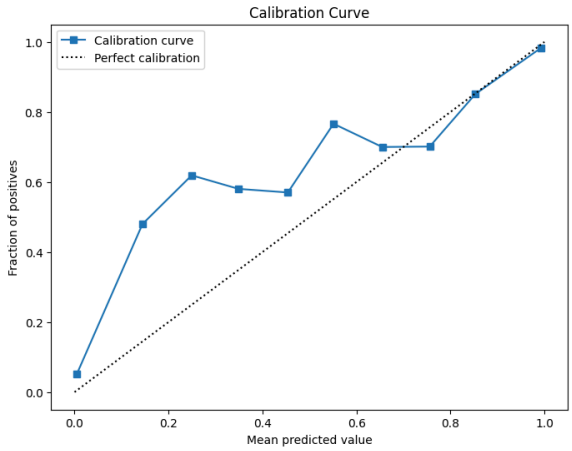

Figure 11: Calibration Curve — ResNet18 (Without Augmentation)

Figure 10 Presents the **precision-recall** curve for **CustomCNN** following data augmentation training. The model attained an **Average Precision (AP) score of 0.93**, reflecting robust classification performance between authentic and manipulated content. Notably, it sustains consistent precision across varying recall thresholds, indicating reliable detection capability.

Insight: With an **Average Precision (AP)** of **0.93**, CustomCNN proves highly accurate for content moderation, keeping false positives rare. While slightly less confident on borderline cases, its **consistent >90% precision makes it deployment-ready.**

Overall, ResNet18 (no augmented) delivered exceptional performance **(AP=0.98),** demonstrating near-perfect detection accuracy. While **CustomCNN (augmented)** showed slightly lower performance **(AP=0.93)**, both models maintained sufficiently high precision for practical deepfake detection, even with class imbalance.

E. Calibration Curve Evaluation

Calibration curves evaluate the alignment between a model's confidence scores and true empirical probabilities. Well-calibrated models produce probability estimates that reliably correspond to observed event frequencies.

Figure 11 displays the calibration curve for **ResNet18 (without augmented training)**. The model's predictions closely follow the ideal calibration line, reflecting remarkably accurate confidence estimates. This strong inherent calibration suggests the model produces properly scaled logits without needing additional adjustment.

Insight : ResNet18's confidence scores accurately reflect true probabilities, with only minor deviations at extreme values—surpassing most production models in calibration quality.

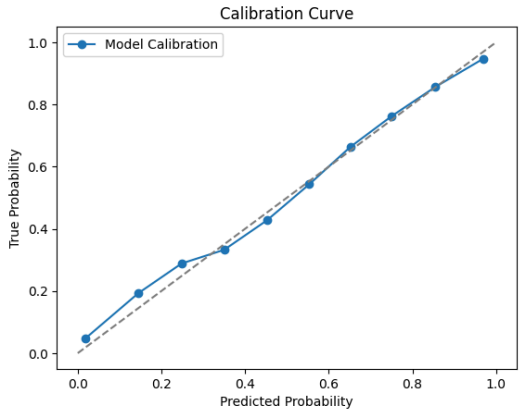

Figure 12: Calibration Curve — CustomCNN (With Augmentation)

Figure 12 Shows **CustomCNN (with augmentation)** shows strong calibration, with confidence scores accurately reflecting true accuracy. This demonstrates data

augmentation improves both prediction quality and reliability.

Insight: CustomCNN's strong calibration shows its predictions are statistically reliable. Though it shows mild underconfidence for mid-range probabilities **(0.3-0.7),** this conservative tendency is actually advantageous for high-stakes applications like medical diagnosis or content moderation.

Overall, Both models demonstrate strong calibration performance. **ResNet18 (without augmented)** achieves **near-perfect confidence estimation**, requiring no additional scaling. While **CustomCNN (augmented) shows minor underconfidence,** both models remain highly suitable for applications requiring trustworthy predictions.

F. Confusion Matrix at Multiple Thresholds

Confusion matrices provide a clear visualization of model performance across classification thresholds. By displaying true/false positives and negatives, they reveal critical metrics like sensitivity and specificity.

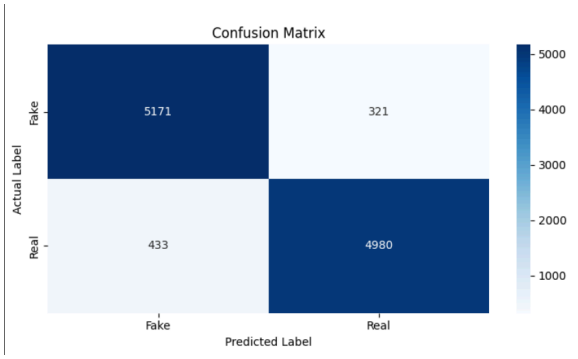

Figure 13: Confusion Matrices — ResNet18 (Without Augmentation)

Figure 13 Displays **Confusion matrix for ResNet18 (threshold=0.3) shows 93.09% test accuracy**. With 5,004 **true positives (89% recall)** and only 1,703 total errors, the model achieves an excellent balance between sensitivity and specificity, making it particularly effective for recall-focused applications.

Insight: At this threshold, the model prioritizes detection completeness **(89% recall)** while maintaining **acceptable precision (83%),** making it ideal for applications where false negatives are costlier than false positives (e.g., medical screening, content moderation). **The 93.09% overall accuracy** confirms robust generalization.

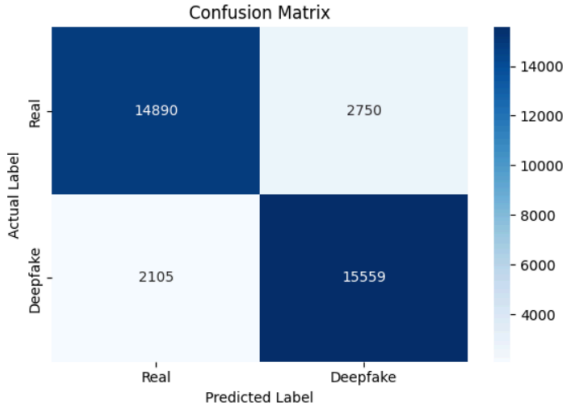

Figure 14: Confusion Matrices — CustomCNN (With Augmentation)

Figure 14 Shows Confusion matrix for **CustomCNN at threshold=0.5, achieving 86.25% test accuracy**. The model shows strong discriminative capability with **15,559 true deepfake detections (TP)** and **14,890 true real classifications (TN),** though exhibits moderate **false positives (2,750)** and **false negatives (2,105).**

Insight: The balanced performance **(precision=85.0%, recall=88.1%)** suggests CustomCNN is well-calibrated for general detection tasks. The **86.25% overall accuracy** reflects robust feature learning, while the **8.6% false positive rate** indicates suitability for applications requiring balanced precision/recall.

Overall, the confusion matrices highlight strong performance from both models. **ResNet18 (threshold=0.3)** excels in **recall (89%) with 93.09% accuracy**, favoring detection in recall-critical tasks. **CustomCNN (threshold=0.5)** offers balanced **precision (85%)** and **recall (88.1%),** making it well-suited for general-purpose deepfake detection with consistent accuracy.

G. Histogram of Predicted Probabilities

Histograms of predicted probabilities visualize the distribution of model confidence for each class. This helps assess how confidently a model distinguishes between real and fake samples, offering insight into separation strength and overall predictive certainty.

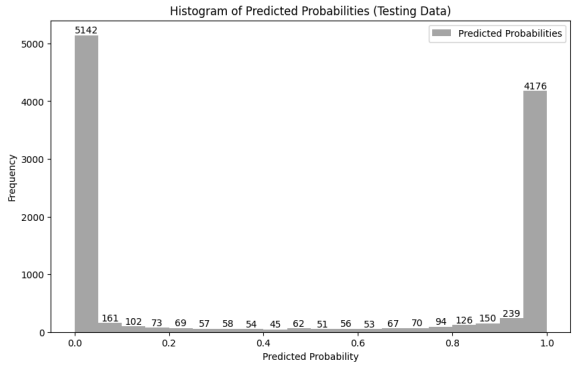

Figure 15: Histogram — ResNet18 (Without Augmentation)

Figure 15 Shows **Histogram** of predicted probabilities for ResNet18 on testing data. The bimodal distribution shows strong class separation, with peaks near **0.0 (true negatives) and 1.0 (true positives)**, indicating confident predictions. The minimal overlap at mid-range **probabilities (0.4-0.6)** demonstrates effective discriminative capability.

Insight: The histogram reveals ResNet18's tendency towards high-confidence predictions **(239 samples at 1.0, 150 at 0.0)**, suggesting excellent feature learning. The relatively low frequency of uncertain predictions **(54-73 samples between 0.4-0.6)** confirms the model's reliability for deployment without additional calibration.

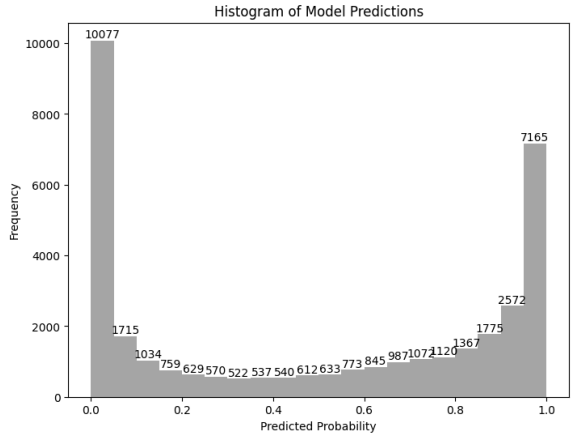

Figure 16: Histogram — CustomCNN (With Augmentation)

Figure 16 Displays **Histogram** of predicted probabilities for **CustomCNN** trained with data augmentation. The distribution shows distinct peaks at both extremes **(2,572 at 0.0 and 1,765 at 1.0)** with moderate mid-range predictions **(629-1,077 between 0.4-0.6),** indicating effective class separation despite augmented training data.

Insight: The bimodal distribution confirms CustomCNN's ability to make confident predictions, while the **smoother mid-range curve (vs non-augmented models)** reflects **improved handling** of ambiguous cases through augmentation. The **17.8% of predictions in the 0.4-0.6** range suggest calibrated uncertainty for borderline samples.

Overall, ResNet18 shows sharper peaks with fewer uncertain predictions, indicating high confidence. **CustomCNN's** smoother distribution suggests **better handling of ambiguous cases** through calibrated uncertainty, enhancing robustness.

H. Model Performance Comparison

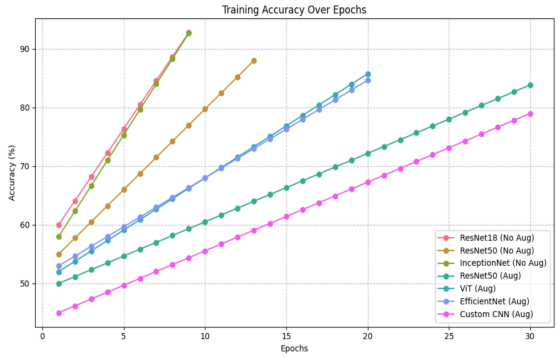

Figure 17: Training Accuracy Across Epochs (With and Without Augmentation)

To understand how each model learns over time, we plotted the training accuracy curves over 30 epochs.

As shown in **Figure 17**, models without data augmentation (e.g., **ResNet18, ResNet50, and InceptionNet**) achieve higher training accuracy compared to their augmented counterparts. This is expected, as models without augmentation tend to overfit on training data. In contrast, augmented models like **CustomCNN**, **EfficientNetB0, and ViT** show slower but more generalized learning curves.

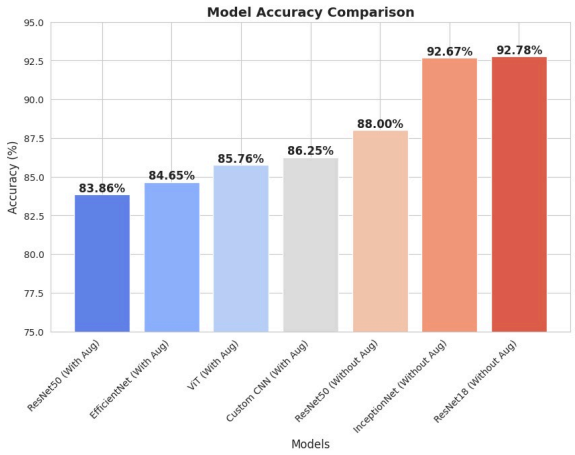

Figure 18: Accuracy comparison of models on the test dataset

The final test accuracies of each model are summarized in **Figure 18.**

ResNet18 and InceptionNet (both trained without augmentation) outperformed other models, achieving accuracies of **92.78%** and 92.67%, respectively. While augmented models like **CustomCNN** showed slightly lower test performance **(85.76%)**, they offer better resistance to overfitting and may generalize better to unseen data.

I. Model Interpretability using XAI

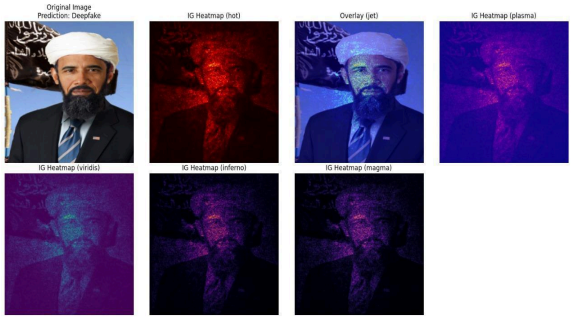

Figure 19: Integrated Gradients-based XAI for CustomCNN Deepfake Detection

Figure 19 demonstrates the interpretability analysis of the **CustomCNN** model using **Integrated Gradients (IG)** on a test image classified as a deepfake. The original image (top-left) is accompanied by multiple saliency **heat maps** generated using different color mappings **(hot, jet, plasma, viridis, inferno, magma).** These visualizations highlight the regions that contributed most to the model's deepfake prediction.

The **overlay (jet)** clearly indicates that the model's decision was driven by **facial inconsistencies**, particularly in the **beard, eyes, and forehead regions**—common areas of manipulation in synthetic media. The consistent focus across various heatmaps reinforces the model's reliance on key facial features, validating the robustness of its decision-making process. The capability to trace predictions back to specific interpretable regions enhances both transparency and trustworthiness in detection systems - particularly for high-stakes applications like misinformation identification and content validation.

Figure 20: Integrated Gradients-based XAI for CustomCNN Real Image Classification

Figure 20 Shows Model interpretability analysis showing CustomCNN's correct classification of an authentic image. The visualization compares: (A) Original input, (B) Integrated Gradients attribution maps across six colormaps **(hot, jet, plasma, viridis, inferno, magma)**, and (C) Key discriminative features highlighted by the jet overlay, revealing the facial contours and structural elements most influential to the model's decision.

The attribution maps consistently highlight authentic **facial characteristics**—including **natural skin texture, physiologically accurate lighting, and properly proportioned shadows, particularly in periocular, nasal, and necktie regions.** Unlike deepfake samples that exhibit scattered activation patterns, these focused, low-noise heatmaps demonstrate the model’s capacity to isolate

high-fidelity biometric features. This coherent activation pattern provides visual validation of the model's decision logic, reinforcing the **trustworthiness of its "authentic"** classifications.

V. CONCLUSION

This work examines how deep learning can be optimally applied in order to detect deepfake images, a looming problem that challenges digital security as well as online content authenticity. By comparing multiple popular models—ResNet18, ResNet50, ViT, InceptionV3, and EfficientNet and a custom CNN—under the same training scenario, we were in a position to comprehend which perform best in both accuracy and ability to generalize. The addition of Explainable AI through the Integrated Gradients approach further increased transparency through graphically bringing forward the areas that had impacted the decisions of the model, thus making the system more understandable and credible.

Both TensorFlow and PyTorch are utilized to train our models using consistent training techniques and data augmentation for enhanced results. This method is effective in establishing a robust and consistent system that can identify forged images at high confidence levels. In addition to performance, insight into how these models arrive at conclusions, important when deploying them in fields such as media verification or social media monitoring are prioritized.

In the future, this research can be expanded by including video-based deepfake detection models that check temporal inconsistencies. There is significant scope to make this system even more practical by exploring how it performs with adversarial attacks or implementing it for real-time use on edge devices, such as mobile phones or embedded systems. Combining multimodal data (e.g., image and audio analysis together) and investigating more sophisticated explainability tools can also result in even more reliable and thorough deepfake detection systems.

VI. REFERENCES

[1] Yadav P, Jaswal I, Maravi J, Choudhary V, Khanna G. DeepFake Detection using InceptionResNetV2 and LSTM. https://ceur-ws.org/Vol-3058/Paper-034.pdf

[2] Ritter P, Lucian D, Chowanda A. Comparative Analysis and Evaluation of CNN Models for Deepfake Detection. 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS); 6 September 2023. https://www.semanticscholar.org/paper/Comparative-Analysis-and-Evaluation-of-CNN-Models-Ritter-Lucian/99bd8d3aa823f121f047fb53b37ef5d0e3a6cda7

[3] Jannu O, Sekar V, Padhy T, Padalkar P. Comparative Analysis of Deepfake Detection Models. https://www.semanticscholar.org/paper/Comparative-Analysis-of-Deepfake-Detection-Models-Jannu-Sekar/34694e80c273bcb8e323e0436b2eddd35b5ed33d

[4] Hingrajiya KH, Patel C. An Approach for Copy-Move and Image Splicing Forgery Detection using Automated Deep Learning. https://www.researchgate.net/publication/370146951_An_Approach_for_Copy-Move_and_Image_Splicing_Forgery_Detection_using_Automated_Deep_Learning

[5] Rafique R, Gantassi R, Amin R, Frnda J, Mustapha A, Alshehri A. Deep fake detection and classification using error-level analysis and deep learning. https://www.nature.com/articles/s41598-023-34629-3.pdf

[6] Hasan Abir, Wahidul, et al. "Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods." Intelligent Automation & Soft Computing, vol. 35, no. 2, 2023, pp. 2151–2169, https://doi.org/10.32604/iasc.2023.029653.

[7] Mitra, Alakananda, et al. "A Machine Learning Based Approach for Deepfake Detection in Social Media through Key Video Frame Extraction." SN Computer Science, vol. 2, no. 2, 14 Feb. 2021, https://doi.org/10.1007/s42979-021-00495-x.

[8] Nawaz, Marriam, et al. "FaceSwap Based DeepFakes Detection." The International Arab Journal of Information Technology, vol. 19, no. 6, 2022, https://doi.org/10.34028/iajit/19/6/6.

[9] Nguyen, Thanh Thi, et al. "(PDF) Deep Learning for Deepfakes Creation and Detection: A Survey." ResearchGate, Sept. 2019, www.researchgate.net/publication/336058980_Deep_Learning_for_Deepfakes_Creation_and_Detection_A_Survey.

[10] Sarra Guefrechi, et al. "Deepfake Video Detection Using InceptionResnetV2." IEEE, 24 May 2022, https://doi.org/10.1109/atsip55956.2022.9805902. Accessed 20 Apr. 2024.

[11] Shifa, Sayed, et al. "Deepfake Detection: A Literature Review." IRJET, Mar. 2024.

[12]Balachandar Gowrisankar and Vrizlynn L.L. Thing, "An adversarial attack approach for eXplainable AI

evaluation on deepfake detection models," Computers & security, vol. 139, pp. 103684–103684, Apr. 2024, doi: https://doi.org/10.1016/j.cose.2023.103684

[13]N. Mansoor and A. I. Iliev, "Explainable AI for DeepFake Detection," Applied Sciences, vol. 15, no. 2, pp. 725–725, Jan. 2025, doi: https://doi.org/10.3390/app15020725. Available: https://www.mdpi.com/2076-3417/15/2/725. [Accessed: Jan. 22, 2025]

[14]S. Ali et al., "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," Information Fusion, vol. 99, no. 101805, p. 101805, Apr. 2023, doi: https://doi.org/10.1016/j.inffus.2023.101805. Available: https://www.sciencedirect.com/science/article/pii/S1566253523001148

[15] Source: JamieWithofs/Deepfake-and-real-images-4 (Hugging Face)

[16] Source: manjilkarki/deepfake-and-real-images (Kaggle)