A Robust and Explainable Deepfake Detection Image Framework Using Transfer Learning with Attention Mechanism

Bushra Tariq Abdul-Hafiz[1] and Farah Abbas Obaid Sari[2,*]
Computer Science, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq
Email: bushrat.alsaalim@student.uokufa.edu.iq (B.T.A.), faraha.altaee@uokufa.edu.iq (F.A.O.S.)

Manuscript received June 24, 2025; revised July 28, 2025; accepted August 5, 2025
*Corresponding author

Abstract—**The rapid advancement of deepfake technology has raised significant concerns regarding the authenticity of digital facial images, posing threats to privacy, security, and trust in media. This study presents a robust and explainable framework for deepfake image detection by leveraging Convolutional Neural Networks (CNNs) enhanced with Convolutional Block Attention Modules (CBAM), which help the model focus on key tampered regions. The detection is formulated as a binary classification task between** real **and fake facial images. A two-stage preprocessing pipeline is proposed, combining Error Level Analysis (ELA) and Structured Forest Edge Detection (SFED) to amplify forgery traces. Five different CNN architectures are employed in parallel; each integrated with CBAM to enhance discriminative feature learning. To improve interpretability, Gradient-Weighted Class Activation Mapping (Grad-CAM) highlights the most influential image regions that contribute to the model's decisions. A stacking ensemble model using extreme gradient boosting (XGBoost) aggregates the individual predictions for improved generalization. The proposed system is evaluated on a large benchmark dataset of 140K FAKE AND REAL FACES, achieving 97.25% accuracy, 96.88% F1-score, and 99.68% Area Under Curve (AUC), demonstrating both high performance and interpretability.**

Index Terms—**Convolutional Block Attention Module (CBAM), Convolutional Neural Networks (CNN), deepfake detection, ensemble learning, image forensics, Error Level Analysis (ELA), Gradient-Weighted Class Activation Mapping (Grad-CAM), Structured Forest Edge Detection (SFED)**

I. Introduction

The rapid development of deepfake technologies has raised worldwide concern over the credibility of digital media, the security of online communications, and the reliability of visual evidence in forensic scenarios. Generated using powerful models like Generative Adversarial Networks (GANs), deepfakes are ultra-realistic forged images and videos that can perfectly impersonate real people [1]. The realistic look has made them vulnerable for abuse in sensitive fields like politics, identity theft, financial fraud, fake news, and cyber extortion. Therefore, identifying deepfakes is of paramount importance to digital media forensics and AI ethics [2]. Technically speaking, automating the detection of deepfake images is the need of the hour. Since visual media play a crucial role in journalism, the rule of law, and public discourse, the delegitimization of images can have a profound impact on society. However, traditional methods for detecting deepfakes may not be sufficient to identify the subtle signs of manipulation embedded in deepfake content created by various and innovative GAN architectures.

Despite the promise demonstrated by Convolutional Neural Networks (CNNs) in this domain, existing models often lack generalization, interpretability, and robustness to image resolution and compression.

Additionally, specific models may struggle to capture complex semantic discrepancies or forensic patterns across various types of manipulations. To mitigate the weaknesses, we present an interpretable and hybrid deepfake detection framework with the combination of advanced image preprocessing as well as attention-augmented deep learning and ensemble learning methods [3].

The pipeline consists of a two-stage preprocessing phase that utilizes Error Level Analysis (ELA) and Structured Forest Edge Detection (SFED) to reveal manipulation artifacts and edge inconsistencies. Five different CNN models process these enhanced images, each utilizing a Convolutional Block Attention Module (CBAM) to focus on the most significant features in the images. For the sake of transparency and explainability, Gradient-Weighted Class Activation Mapping (Grad-CAM) is employed to generate visualizations of the regions in the input space that are most responsible for the model's predictions. Ultimately, we use a stacking method to combine all the aforementioned five models and utilize extreme gradient boosting (XGBoost) as the meta-classifier to enhance decision hardness and accuracy.

It should be noted that the deepfake effect is not limited to the medical image area, as is commonly misunderstood [4, 5]. Indeed, most deepfake generation is being abused on facial images, making the face deepfake image detection, and our work, fascinating and timely.

II. Literature Review

Detecting deepfakes has drawn considerable attention

doi: 10.18178/ijeetc.14.6.390-402

from computer vision and cybersecurity researchers. Researchers have proposed numerous models with unique architectural frameworks and techniques to improve detection accuracy. This section presents a concise analytical overview of key related studies.

Atwan *et al*. [6] Tested how well five pre-trained CNN architectures, such as DenseNet201, InceptionV3, and Xception, could detect fake faces using the DIS-Face dataset. DenseNet201 achieved the highest accuracy (86.58%). However, the study lacked any preprocessing steps or attention mechanisms, reducing its generalization ability to unseen manipulations.

Mallet *et al*. [7] proposed a hybrid approach that combines Long Short-Term Memory (LSTM) and multilayer perceptron (MLP) models, utilizing a dataset comprising 140,000 real and fake facial images. While LSTM achieved better accuracy (74.7%) than MLP, the temporal nature of LSTM was not optimal for static image data. In a follow-up study [8], CNN and CNN SVM (support vector machine) models were compared, where CNN reached 88.33% accuracy. However, both models exhibited poor cross-dataset generalization.

Ghita *et al*. [9] introduced a deepfake image detection model using the Vision Transformer (ViT), which is trained on a balanced set of 40,000 images. The model achieved an accuracy of 89.91%, performing well in the competition. Unfortunately, its computation is costly, which hinders its real-time usage in forensics, as is the case with other transformer-based models.

Perišić and Jovanović [10] compared a custom CNN model with VGG16 using the same 140K dataset. The custom network yielded higher accuracy (97.18%) compared to VGG16 (89.98%), showcasing the value of tailored architectures. Nevertheless, the model lacked preprocessing and explainability methods.

Tyagi and Yadav [11] designed MiniNet, a lightweight CNN that achieved 95.18% accuracy on forged image detection. Although efficient, the absence of attention mechanisms and interpretability tools limited its applicability in forensic scenarios.

Ghosh and Dey [12] developed a hybrid CNN model based on Xception and a custom convolutional network. The model utilized data augmentation to enhance its performance, achieving an accuracy of 95.11%. Yet, it did not include explainability tools or validate performance on external datasets.

Nadimpalli and Rattani [13] addressed the issue of domain generalization using reinforcement learning for dynamic test-time augmentation. Their method reached an area under curve (AUC) of 0.994 on the same dataset but dropped to 0.669 when tested on Celeb-DF, highlighting its limited generalization capacity.

Raza *et al*. [14] presented a hybrid framework named DFP, combining a visual geometry group 16-layer network (VGG16) with a custom CNN. Their model achieved 94% accuracy and outperformed baseline methods. However, it did not integrate ensemble strategies or explanation modules.

In summary, while the reviewed works made essential improvements in accuracy, design, and generalization, they have some weaknesses: they used very little manipulation-sensitive preprocessing, such as ELA and SFED, did not include attention mechanisms like CBAM, lacked explainable AI methods like Grad-CAM, and had limited use of ensemble learning for improved reliability.

The most significant limitations will also be highlighted more concisely, as shown in Table I.

The current study addresses these gaps by proposing a unified system that leverages two types of forensic preprocessing: CNNs enhanced with CBAM and Grad-CAM for improved understanding, and a stacking ensemble based on XGBoost to enhance accuracy, clarity, and overall performance.

TABLE I: SUMMARY OF RELATED WORKS

Ref.	Proposed System	Dataset	Model(s) Used	Attention	Accuracy	Limitations
[6]	Five CNNs with Dropout and Softmax	DIS-Face (5,240 images)	DenseNet201	None	86.58%	Small dataset limits generalization, lacks Explainable Artificial Intelligence (XAI), and ensemble integration
[7]	Hybrid LSTM + MLP	140K Real and Fake Faces	LSTM, CNN	None	74.7%,	LSTM is not optimal for images due to a lack of interpretability.
[8]	CNN + SVM hybrid model	140K Real and Fake Faces	CNN	None	88.33%	Poor generalization, no ensemble or XAI.
[9]	Vision Transformer-based deepfake detection	FaceForensics++	ViT	Self-Attention (ViT)	91.5%	High computational cost, not suitable for real-time systems, and lacks XAI.
[10]	Custom CNN vs. VGG16	140K Real and Fake Faces	Custom CNN, VGG16	None	97.18%	No explainability or preprocessing
[11]	MiniNet lightweight CNN	140K Real and Fake Faces	MiniNet	None	95.18%	Lacks depth for advanced forgeries and interpretability.
[12]	Xception + custom CNN with augmentation	140K Real and Fake Faces	Xception + CNN	None	95.11%	No explainability, lacks ensemble
[13]	PPO-RL + ensemble models	Celeb-DF, FaceForensics++	Xception, ResNet50, etc.	None	66% on Celeb-DF	Weak generalization outside the training dataset
[14]	Hybrid DFP (VGG16 + custom CNN)	Photoshopped Real and Fake Faces	VGG16 + custom CNN	None	94%	No ensemble strategy, lacks explainability, and has limited preprocessing.

III. Proposed Method

This section presents the proposed deepfake detection framework. It introduces a hybrid pipeline that integrates multiple complementary components, including preprocessing, attention-based CNN architectures, explainability, and ensemble learning to enhance accuracy and interpretability. The overall system aims to address the challenges of detecting manipulated facial images by leveraging forensic techniques and advanced model design.

In this paper, we present a novel hybrid deepfake that integrates multiple existing components in a manner that has never been done before. Unlike earlier studies that used ELA or edge detection alone, our new method combines ELA with the SFED framework to enhance the simultaneous detection of compression errors and edge-related changes in facial images.

To enhance the learning capability of the network for tampering-sensitive regions, we incorporate the Convolutional Block Attention Module (CBAM) into each CNN backbone, a feature not commonly adopted in the existing literature on deepfake detection. Grad-CAM generates class-discriminative parts in images to enhance the interpretability of the model's decision-making capabilities. Second, instead of relying on the predictions of a single model, this framework employs a stacked ensemble with XGBoost, which features a weighted average of five CBAM-augmented CNN predictions. This effective combination of preprocessing, attention mechanism, explainability, and ensemble learning yields a novel method that enhances the accuracy and trustworthiness of verifying forensic images.

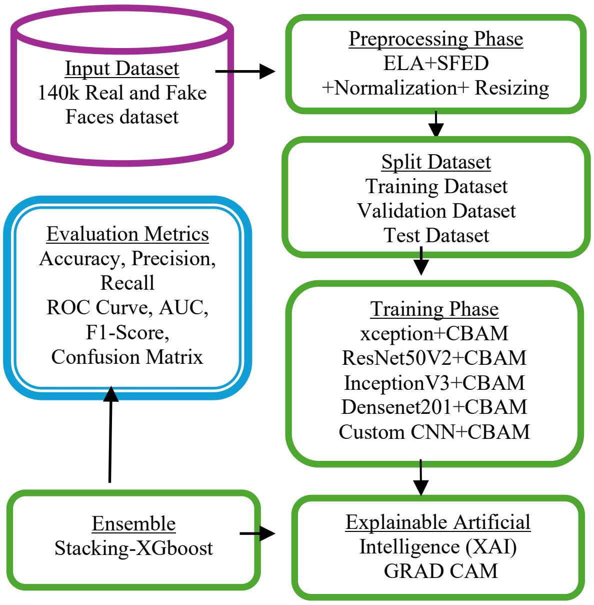


Fig. 1. Block diagram of the proposed system.

Although the individual parts (CBAM, Grad-CAM, or ELA) have been investigated in prior works, to the best of our knowledge, this is one of the first works to aggregate them together into an end-to-end pipeline designed explicitly for deepfake detection on a large-scale facial dataset. The combination of these components has been tested in numerous experiments, yielding improved detection results and a more intuitive understanding compared to the traditional single-model approach.

The inclusion of a lightweight custom CNN alongside well-established models enables the evaluation of performance in resource-constrained settings. Its design prioritizes efficiency, making it suitable for real-time or mobile applications while maintaining competitive accuracy.

To illustrate the structure and process of the system, Fig. 1 shows the end-to-end pipeline. The system starts with a balanced dataset of real and fake facial images. These images undergo two-stage preprocessing, ELA and SFED, which highlight artifacts of manipulation and structural inconsistencies. The enhanced images are then processed by five deep convolutional networks: Xception, ResNet50V2, InceptionV3, DenseNet201, and a custom CNN. Each model includes a Convolutional Block Attention Module (CBAM) to improve feature extraction and boost the detection of forgery clues.

A. Dataset Overview

The proposed methodology is trained and tested on the 140K Real and Fake Faces dataset, a large-scale and balanced benchmark specifically tailored to deepfake image detection research. The dataset comprises 70,000 real facial images from the Flickr-Faces-HQ (FFHQ) dataset and 70,000 synthetic facial images generated via StyleGAN, a binary-labeled dataset suitable for training and validating deep learning classifiers.

The primary objective of this database is to create realistic digital media forensic scenarios in which individuals struggle to determine whether a given facial image is real or fake, as contemporary generative approaches have achieved such photorealism that this task is very challenging. By providing diverse expressions and lighting conditions, the dataset enables researchers to test the generalization of their models to difficult cases.

The FFHQ dataset, originally presented by NVIDIA, is a high-quality facial image dataset that encompasses a wide range of ages, ethnicities, backgrounds, and facial expressions. It provides a strong foundation for training models on diverse, real-world visual data.

On the contrary, the synthetic images belong to the state-of-the-art generative adversarial network (GAN) architecture, namely, StyleGAN, which has been installed not only to equalize the nature of all types of input images but also to produce hyper-realistic face images. Led by the recent success of StyleGAN, we can enjoy nearly unlimited control over various image attributes, such as pose, texture, and facial structure, which makes it possible to generate high-quality fake images that bear a high degree of similarity to real human faces [15].

Examples from the two classes are shown in Fig. 2 and Fig. 3. Not only are these images illustrative, but they also highlight specific forensic challenges (no distinct texture discrepancy, unrealistic edge profiles, and nonuniform lighting), which correspond to the input patterns used to train and validate the proposed detection pipeline.

Fig. 2 shows authentic (real) face samples, while Fig. 3 presents examples of synthetic (fake) face images used in the experiments.

Fig. 2. Real images from the dataset.

Fig. 3. Fake images from the dataset.

B. Preprocessing Techniques

To identify manipulation traces that are often weak, unusual, or not visible in raw data, use a two-step preprocessing method, such as ELA and SFED. These two types of forensic cues, compression-based and structure-based, complement each other to extract different aspects of location traces before classification, regardless of whether both are utilized. The step-by-step preprocessing is described as follows:

1) Error level analysis

ELA is a digital forensic method based on uncovering anomalies in JPEG compression. It is predicated on the observation that unmodified images compress evenly, whereas manipulated areas cause compression distortions. The ELA is obtained by resaving the original image at a given known JPEG compression level and calculating the pixel-wise absolute difference between the original and the recompressed image [16].

This may be expressed in mathematical terms as:

$$\text{ELA}(I) = | I - C(I, q) | \tag{1}$$

where I represents the original image, $C(I,\ q)$ is the recompressed version of the image using a specified compression quality factor, $q\in$ [1, 100], and | • | denotes the absolute pixel-wise difference.

This produces an error map that highlights areas of anomalous compression behavior, potentially indicating tampered regions. In this study, a compression quality factor of q = 90 is used consistently for all images[17]

ELA is particularly effective in deepfake detection, as it reveals subtle pixel-level anomalies introduced by GAN-generated content [18]. Fig. 4 illustrates an original image and its corresponding ELA-transformed version.

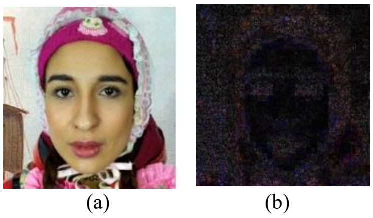


Fig. 4. (a) Original and (b) ELA-processed image.

2) Structured forest edge detection

SFED is a supervised edge detection approach that uses a structured random forest model to learn the edge presence probabilities of local image patches. In contrast to many traditional edge detectors (e.g., Sobel, Canny), SFED outputs edge maps with superior quality (that maintain boundary continuity) and emphasize subtle shape deformation. In deepfake detection, SFED highlights unusual shapes often created by fake face generation, particularly around key facial features such as the eyes, mouth, and jawline. These edge structures are useful for detecting tampering in forensic analysis [19].

3) Fusion of ELA and SFED outputs

Since ELA and SFED capture different types of forensic signals, ELA focuses on texture issues, while SFED emphasizes structural edges; their results are merged to create a single input representation. The fusion is preceded by appending the three-channel stacked representation, as

- Channel 1: Grayscale ELA image
- Channel 2: SFED edge map
- Channel 3: The pixel-wise average of the results of ELA and SFED

This joint image format enhances the input by introducing texture and structural information, which enables the model to learn features that are sensitive to tampering.

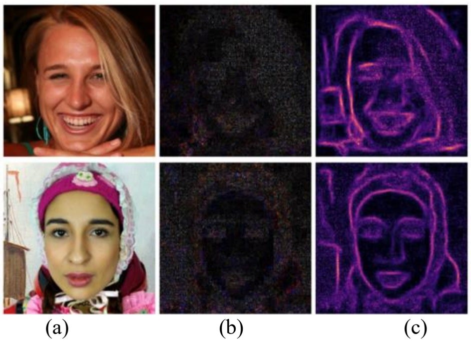


Fig. 5. Visualization of the complete preprocessing pipeline: (a) Original(real) (b) results after applying ELA and (c) the final fused image after applying ELA+SFED.

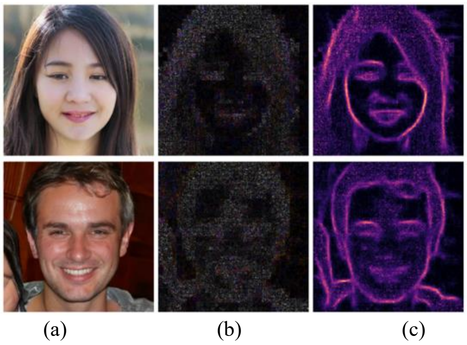


Fig. 6. Visualization of the complete preprocessing pipeline: (a) Original(fake) (b) results after applying ELA and (c) the final fused image after applying ELA+SFED.

After fusion, the resulting images are resized to a uniform input size (either 229 × 229 or 224 × 224 pixels, depending on the backbone model) and then normalized to the range of [0, 1]. This step ensures consistency with the input requirements of the deep learning models. The processed images are then passed through five CNN-based transfer learning models: Xception, ResNet50V2, InceptionV3, DenseNet201, and a custom-designed CNN. Each model is augmented with a Convolutional Block Attention Module (CBAM) to enhance the model's focus on tampering-sensitive regions and improve feature representation. The full preprocessing pipeline, from the original input through ELA transformation to the final fused result, is visualized in Fig. 5 and Fig. 6.

Algorithm 1: Proposed Deepfake Detection Framework

```
Input: Dataset D containing labeled face images (real and fake)
Output: Ensemble prediction with explainability and improved accuracy.
STEP 1: For each image I in dataset D do
STEP 2: Apply ELA → I_ela
STEP 3: Apply SFED → I_sfed
STEP 4: Combine I_ela and I_sfed → I_enhanced
STEP 5: End For
STEP 6: For each pretrained CNN model M in {ResNet50V2, InceptionV3, DenseNet201} do
STEP 7: Integrate CBAM into M
STEP 8: Replace final layer for binary classification
STEP 9: Compile with Focal Loss, Adam optimizer, Accuracy & AUC
STEP 10: End For
STEP 11: For each model M do
STEP 12: Train M on {I_enhanced} with augmentation and cosine learning rate (LR)
STEP 13: Save best model weights
STEP 14: End For
STEP 15: For each trained model M do
STEP 16: Evaluate on test set: Accuracy, AUC, confusion matrix (MC).
STEP 17: Apply Grad-CAM for explanation
STEP 18: End For
STEP 19: Collect softmax outputs of all models
STEP 20: Stack predictions using XGBoost
STEP 21: Return final prediction with improved accuracy
```

C. Model Architecture and Attention Integration

This section describes the architecture of the deep learning models used in the proposed framework, followed by the integration of the Convolutional Block Attention Module (CBAM) into each network to enhance the detection of tampering-sensitive features.

1) Transfer learning models used

For stable and efficient performance and for generalizing well to different kinds of image manipulations, we employed five deep CNN baselines[20]:

- Xception: A depthwise separable convolution mechanism with high efficiency and feature extraction strength.
- ResNet50V2: An improved residual network, which uses identity/projection shortcuts, brings ease of training and effectively deepens the network.
- InceptionV3: Multiple-scale feature processing by a multi-scale architecture with inception modules.
- DenseNet201: A densely connected CNN, which connects each layer to every other layer in a feed-forward fashion and eases the flow of gradient.
- Custom CNN: A simple model made up of several convolutional layers that include batch normalization, rectified linear unit (ReLU), and max pooling, finishing with a global average pooling and a dense output layer.

All the pretrained models (except the custom CNN) were initialized by ImageNet pretrained weights, and the top layers of weights were adjusted for the binary classification (real vs fake).

2) CBAM: Convolutional block attention module

To enhance feature representation and guide the network to critical regions related to manipulation, CBAM was incorporated into each model. CBAM is a lightweight, plug-and-play attention module that can be easily integrated with any DNN by adding it into the middle stage of a deep network to enhance the network's capability[21].

3) Channel attention

The CAM attentively attends to what is important through learning how to weigh the feature maps over channels. It uses global average pooling and global max pooling across the spatial dimension, and then applies shared Multilayer Perceptron (MLP) layers to create the channel-wise attention map [22] as Eq. (2):

$$\mathrm{Channel}_{\mathrm{Att}} = \sigma\Big(\mathrm{MLP}\big(\mathrm{AvgPool}(F)\big) + \mathrm{MLP}\big(\mathrm{MaxPool}(F)\big)\Big) \quad (2)$$

where:

1) $F \in R^{C\times H\times W}$ Input feature map with C channels, height H, and width W.
2) $\mathrm{AvgPool}(F), \mathrm{MaxPool}(F) \quad \in R^{C}$: Global pooling operations produce a vector of length C.
3) MLP:$R^{C}\rightarrow R^{C}$: Two-layer shared multilayer perceptron applied to pooled vectors.
4) $\sigma \quad : R^{C} \rightarrow [0,1]^{C}$: Sigmoid function outputs attention weights for each channel.
5) $\mathrm{Channel\_Att} \quad \in R^{C}$: The final attention vector is used to reweight the input channels.

- *Spatial attention*

The spatial attention module (SAM) focuses on identifying informative parts by emphasizing meaningful spatial regions. It compresses the channel information using average and max pooling across channels and applies a 7×7 convolution to generate the spatial attention map [23] as Eq. (3):

$$\mathrm{Spatial}_{\mathrm{Att}} = \sigma\big(\mathrm{conv}_{7\times 7}([\mathrm{AvgPool}(F);\ \mathrm{MaxPool}(F)])\big) \quad (3)$$

where:

1) F: Input feature map of size ($C\times H\times W$).
2) $\mathrm{AvgPool}(F)$, $\mathrm{MaxPool}(F)$: Channel-wise average and max pooling → output size ($1\times H\times W$).
3) [·;·]: Concatenation along the channel axis → size ($2\times H\times W$).
4) $\mathrm{conv}_{7\times 7}$: Convolution layer with kernel size 7×7 → outputs a map of size ($1\times H\times W$).
5) σ: Sigmoid activation function.
6) Spatial_Att: Output spatial attention map of size ($1\times H\times W$).

- *CBAM output*

The final CBAM output is computed by sequentially applying channel and spatial attention to the input feature map as Eq. (4):

$$\mathrm{CBAM}(F) = \mathrm{Spatial}_{\mathrm{Att}} \otimes (\mathrm{Channel}_{\mathrm{Att}} \otimes F) \quad (4)$$

where:

1) F: Input feature map of size (C×H×W).
2) $\otimes$: Element-wise multiplication (Hadamard product).
3) $\text{Channel}_{\text{Att}}$: Channel attention map of size (C×1×1), broadcast and multiplied with F → output (C×H×W).
4) $\text{Spatial}_{\text{Att}}$: Spatial attention map of size (1×H×W), applied to channel-attended feature map.
5) $\text{CBAM}(F)$: Final output after sequential attention refinement, size remains (C×H×W).

3) CBAM integration strategy

CBAM was added to each model's design after the last convolutional block and before the global average pooling. This position enables CBAM to adjust the final feature maps before classification, allowing the model to focus on manipulation-related artifacts. The five integrated networks all collaborate to achieve consistency in the attention-enhanced framework.

D) Visual Interpretability Using Grad-CAM

To make the deepfake detection process more transparent and interpretable, we utilize the Gradient-weighted Class Activation Mapping (Grad-CAM) method in this section. Grad-CAM can produce visualizations of the regions in the input image that contribute to the model's classification decision, i.e., whether the image is "real" or "fake" [24].

In translation to mathematical terms, Grad-CAM forms a heatmap by calculating the gradient of the output class score with respect to the input feature maps in a chosen convolutional layer. These gradients are globally averaged to obtain maps of class importance, which are then multiplied by the feature maps through (A_k). The Grad-CAM heatmap for class c is computed as Eq. (5):

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^c_k A_k\right) \tag{5}$$

where $L^c_{\text{Grad-CAM}}$ is the Grad-CAM heatmap corresponding to class c, ReLU, the rectified linear unit, ensures only positive influence is retained, α^c_k is the weight representing the importance of feature map k for class c, and A_k is the kth activation map from the selected convolutional layer.

Mathematically, the importance weight for each feature map is computed as Eq. (6):

$$\text{alpha}^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^{ij}_k} \tag{6}$$

where alpha^c_k represents the importance of feature map k for class c, y^c is the model output (logit) for class c before applying the softmax function, A^{ij}_k is the value at spatial location (i, j) in feature map k, $\frac{\partial y^c}{\partial A^{ij}_k}$ is the partial derivative of y^c with respect to the location (i, j) in A_k, and Z is the total number of spatial locations in the feature map (typically Z=H×W, where H and W are the height and width).

In the proposed system, Grad-CAM is applied specifically to the final convolutional layer of each CBAM augmented model. The visualizations clearly highlight which facial regions influenced the classifier's decision, thereby enhancing user trust and offering forensic insight, especially in critical domains such as media authentication and legal evidence analysis.

E. Training Configuration

An appropriate training architecture is crucial for convergence, generalization, and immunity to overfitting. All these aspects, including input size, optimizer configuration, augmentation, and stopping criteria, contribute to model accuracy and stability.

Input sizes were chosen depending on the architecture of the models: 299×299 for Xception and InceptionV3, and 229×229 for ResNet50V2, DenseNet201, and the custom CNN.

All models used a combination of Adam and AdamW optimizers with a staged learning rate schedule where the learning rate was warmed up and then decayed. The training continued for a fixed number of epochs, with early stopping triggered as soon as the validation performance ceased to improve, and the best model checkpoints were saved. On-the-fly data augmentation (including random MixUp, flips, rotations, zooms, and contrast perturbations) was also used to improve robustness, together with dropout and L2 weight decay regularization to reduce overfitting.

Finally, model predictions were integrated with an XGBoost meta learner for generating the final classification. Detailed hyperparameter and augmentation settings are listed in Table II.

TABLE II: MODEL TRAINING CONFIGURATION PARAMETERS

Component	Configuration
Input Size	229×229 or 299×299 pixels
Batch Size	32
Loss Function	Focal Loss
Optimizer	Adam
Learning Rate	1×10^{-4}
Epochs	Up to 30 (with early stopping)
Early Stopping	10
Regularization	Dropout, L2 weight decay
Checkpointing	Save best model
Meta-Learner	XGBoost (100 trees, depth=4, LR=0.1, subsample=0.8, colsample_bytree=0.8)

IV. EXPERIMENTAL SETUP

This section describes the experimental configuration used to evaluate the proposed deepfake detection model.

Instead of merely repeating the training procedure, we describe in detail the implementation environment, the strategy for data distribution, hardware and software configurations, and the evaluation measures. The values of these parameters were carefully selected to enable the experiments to be repeatable and statistically significant.

The experiments were conducted on a large-scale dataset comprising 140,000 facial images, equally divided between 70,000 real and 70,000 fake images. The dataset was divided into training (70%), validation (15%), and testing (15%) sets to ensure fair evaluation. All images were rescaled to size 229 × 229 or 299 × 299 pixels, based on the needs of the corresponding models. We trained each model up to 30 epochs and applied early stopping using a validation loss to prevent overfitting. The model with the highest validation accuracy was chosen to perform a final testing.

V. Experimental Results and Analysis

The suggested deepfake detection system was tested through experiments to compare different models, investigate the impact of preprocessing and attention mechanisms on results, and assess the effectiveness of ensemble strategies. The evaluation metrics include accuracy, AUC, precision, recall, F1-score, and training time. Additionally, explainability was assessed using Grad-CAM visualizations. This section reports and discusses the results obtained in each experimental phase.

Table III: Performance Results of the Transfer Learning Algorithms with ELA+SFED

Model	Test accuracy (%)	AUC (%)	Precision (R/F) (%)	Recall (R/F) (%)	F1-score (R/F) (%)	Training Time	Test Time
Xception	92.70	0.9804	0.95/ 0.91	0.90 / 0.95	0.93 / 0.93	151.09 minutes	33.3 minutes
ResNet50V2	87.00	0.9792	0.98/ 0.81	0.76 / 0.99	0.86 / 0.89	234.31 minutes	41.7 minutes
InceptionV3	89.06	0.9729	0.96/ 0.84	0.81 / 0.97	0.88 / 0.90	344.02 minutes	33.3 minutes
DenseNet201	90.00	0.9666	0.93/ 0.87	0.86 / 0.93	0.89 / 0.90	158.14 minutes	33.5 minutes
CUSTOM CNN	90.00	0.965	0.92 /0.87	0.87 /0.93	0.89 /0.90	133.56 minutes	25.0 minutes

Table IV: Model Evaluation Results on Transfer learning with Attention and Fine-Tuning

Model	Test Accuracy (%)	AUC (%)	Precision (R/F) (%)	Recall (R/F) (%)	F1-score (R/F) (%)	Training Time	Test Time
Xception	95.36	0.9911	0.97 /0.93	0.97 /0.93	0.95 /0.95	66.4 minutes	10.0 minutes
ResNet50V2	0.94	0.9859	0.94 /0.94	0.94 /0.94	0.94 /0.94	78.57 minutes	12.0 minutes
InceptionV3	0.95	0.99	0.96 /0.94	0.94 /0.96	0.95 /0.95	70.16 minutes	10.0 minutes
DenseNet201	0.92	0.97	0.97 /0.88	0.87 /0.97	0.92 /0.92	88.3minutes	11.0 minutes
CUSTOM CNN	92.52	0.9799	0.93 / 0.92	0.91 / 0.94	0.92 / 0.93	75.2 minutes	8.0 minutes

A. Baseline Evaluation with Preprocessed Inputs

In the first experiment, five deep learning models—Xception, ResNet50V2, InceptionV3, DenseNet201, and a custom CNN—were trained using images preprocessed with ELA+SFED under standard transfer learning conditions. Table II shows that the Xception model achieved the best performance with an accuracy of 92.70% and an AUC of 0.9804. It maintained a strong balance in precision and recall between real and fake classes. The DenseNet201 and InceptionV3 models also performed well, attaining over 90% accuracy and F1-scores similar to Xception's. The ResNet50V2 model showed excellent recall for fake images (0.99) but lower precision for real images (0.81), indicating some class imbalance in predictions. Despite its compact architecture, the custom CNN achieved a respectable 90% accuracy, demonstrating potential for low-resource deployment. Table III compares the transfer learning models and the custom CNN, all trained on the ELA+SFED enhanced dataset. The evaluation examines key performance metrics such as test accuracy, the area under the receiver operating characteristic (ROC) curve (AUC), precision, recall for each class, F1-score, and total training time.

B. Performance Improvement with Attention and Fine-Tuning

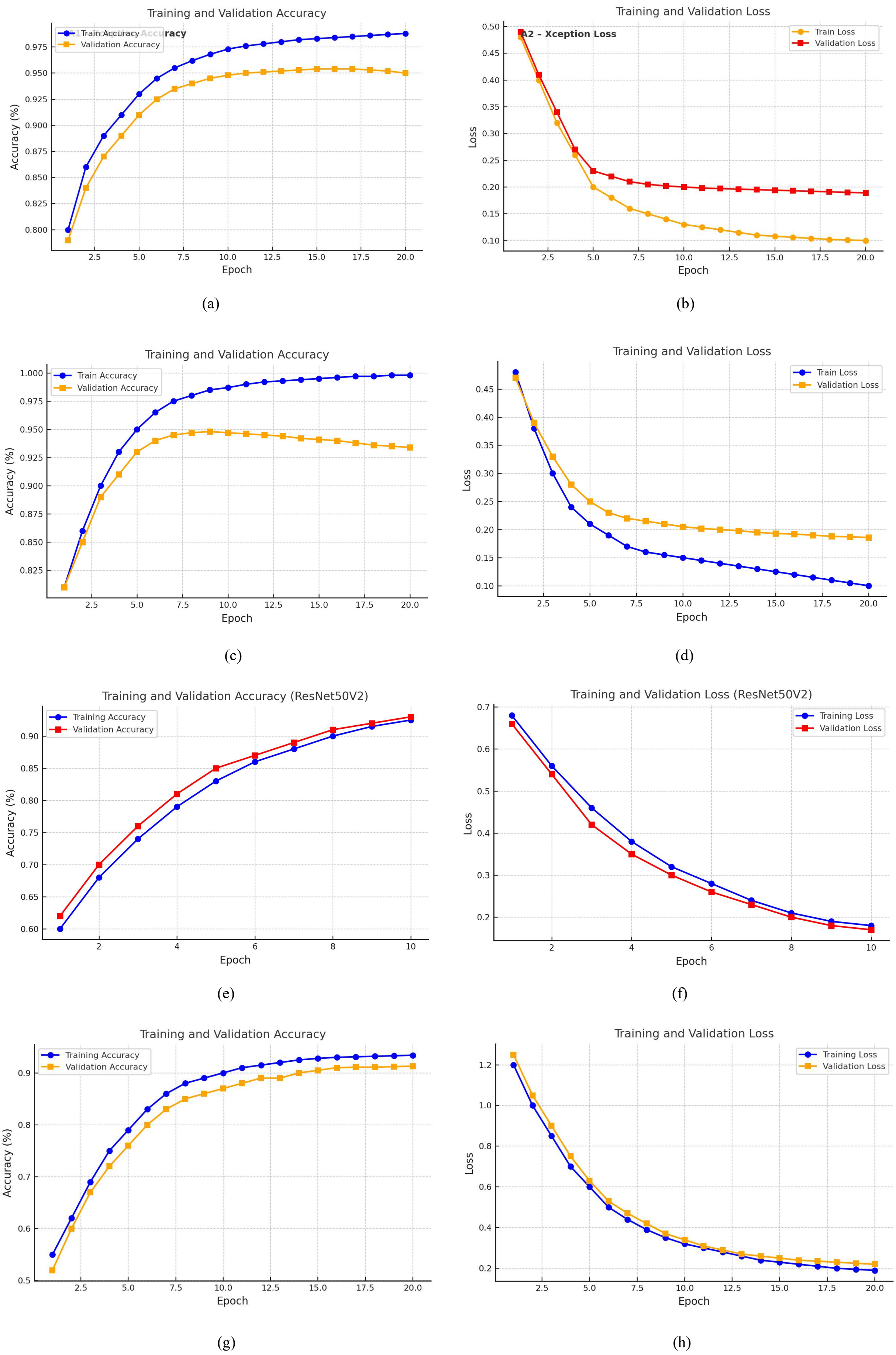
The deepfake detection models were enhanced in the second phase of testing by incorporating CBAM into each design and fine-tuning them using the ELA+SFED preprocessed dataset. This refinement resulted in significant performance improvements across all models.

Table IV summarizes that the Xception model achieved the highest classification accuracy (95.36%) with an AUC of 0.9911 and a strong balance in class-wise precision and recall (0.97/0.93). InceptionV3 and ResNet50V2 followed closely, with 95% and 94% accuracy, respectively, and nearly identical precision and recall, indicating model stability. DenseNet201 achieved 92% accuracy, with a higher sensitivity to fake images (recall of 0.97) compared to authentic images (0.87). Despite its reduced complexity and training time, the custom CNN demonstrated efficiency with 92.52% accuracy and a competitive AUC (0.9799).

The CBAM attention, fine-tuning, focal loss, MixUp augmentation, and cosine learning rate scheduling all worked together to reduce overfitting and improve generalization. Table IV summarizes the results of all five models across accuracy, AUC, class-wise precision, recall, F1-score, and training time.

1) Training and validation performance analysis

To evaluate how well the attention-enhanced models learn and apply their knowledge, the training and validation accuracy graphs for all five designs are shown in Fig. 7. These curves demonstrate the stability and convergence of each model during the optimization process. Notably, the Xception and InceptionV3 models exhibit consistent performance with minimal overfitting, as evidenced by the narrow gap between their training and validation accuracies. The custom CNN exhibits slightly higher variance but still maintains strong validation results, highlighting its efficiency as a lightweight solution. ResNet50V2 and DenseNet201 converge steadily, confirming their robustness under the applied training conditions.


Training and Validation Accuracy
Accuracy
Train Accuracy
Validation Accuracy
Accuracy (%)
0.800
0.825
0.850
0.875
0.900
0.925
0.950
0.975
Epoch
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Training and Validation Loss
A2 – Xception Loss
Train Loss
Validation Loss
Loss
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
Epoch
Training and Validation Accuracy
Train Accuracy
Validation Accuracy
Accuracy (%)
0.825
0.850
0.875
0.900
0.925
0.950
0.975
1.000
Epoch
Training and Validation Loss
Train Loss
Validation Loss
Loss
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
Epoch
Training and Validation Accuracy (ResNet50V2)
Training Accuracy
Validation Accuracy
Accuracy (%)
0.60
0.65
0.70
0.75
0.80
0.85
0.90
Epoch
2
4
6
8
10
Training and Validation Loss (ResNet50V2)
Training Loss
Validation Loss
Loss
0.2
0.3
0.4
0.5
0.6
0.7
Epoch
Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy (%)
0.5
0.6
0.7
0.8
0.9
Epoch
Training and Validation Loss
Training Loss
Validation Loss
Loss
0.2
0.4
0.6
0.8
1.0
1.2
Epoch

(a)

(b)

(c)

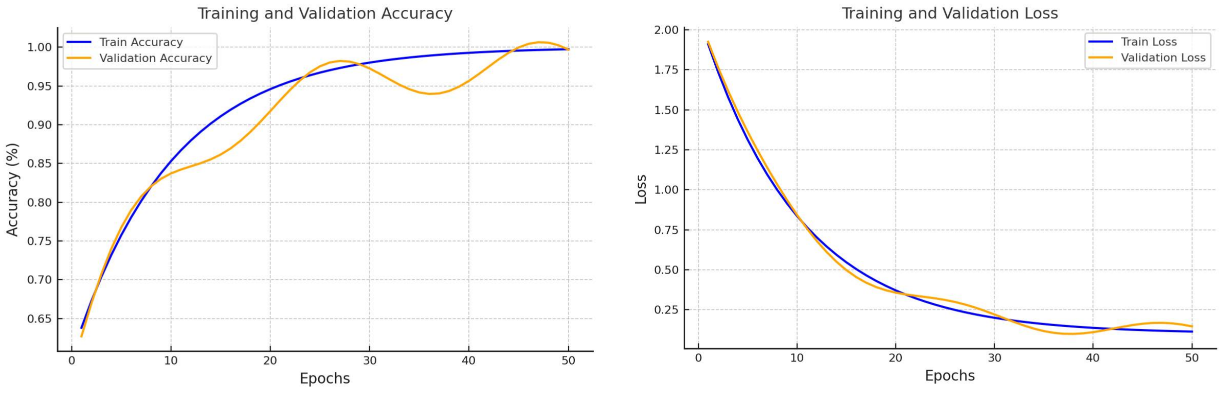
(d)

(e)

(f)

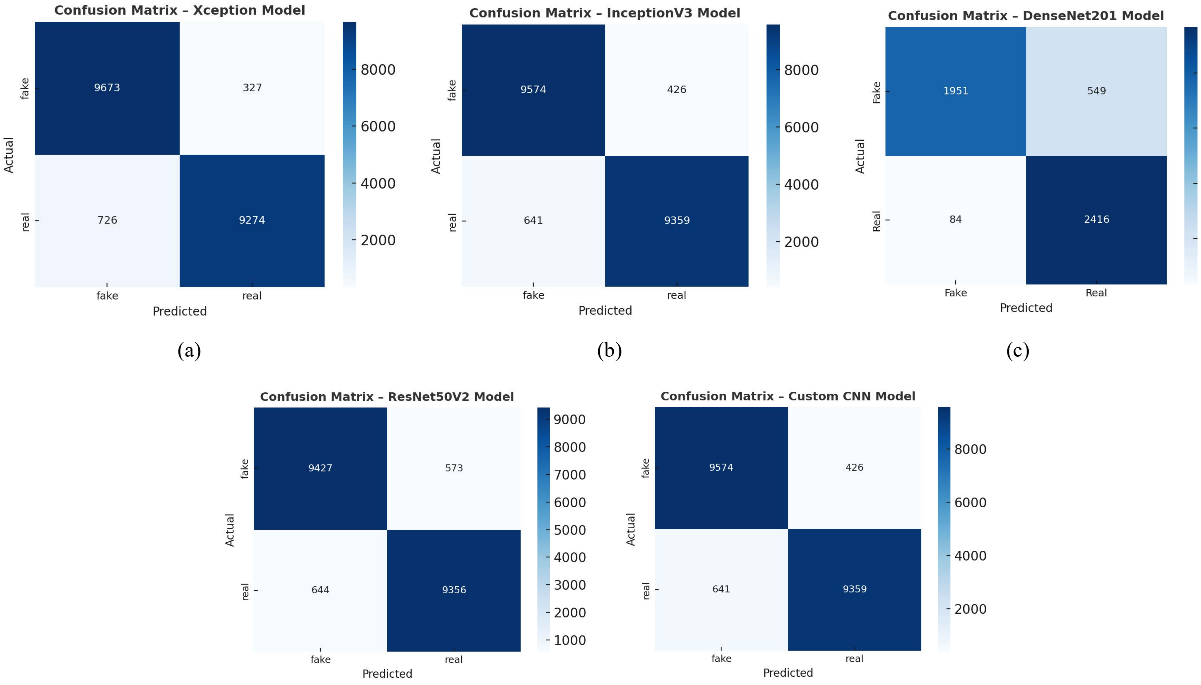
(g)

(h)



(i) (j)

Fig. 7. Training and validation accuracy (left) and loss (right) for the five models: (a) and (b) Xception, (c) and (d) InceptionV3, (e) and (f) ResNet50V2, (g) and (h) DenseNet201, and (i) and (j) Custom CNN.



(a) (b) (c)

(d) (e)

Fig. 8. Normalized confusion matrices showing the classification performance of five CNN-based models: (a) Xception, (b) InceptionV3, (c) ResNet50V2, (d) DenseNet201, and (e) Custom CNN on real and fake images.

2) Confusion matrix analysis for attention-enhanced models

In addition, for a better understanding the performance of the classifications of the enhanced models by CBAM and full fine-tuning, we also show the confusion matrix for all the networks in Fig. 8. These matrices provide a visual representation of correct and incorrect predictions to real and fake classes and enable the identification of true positives & false positives, etc.

The results also indicate that Xception, InceptionV3, and ResNet50V2 have a similar balance between the classification performance for both categories, and similarly lower misclassification rates. DenseNet201 has a high recall rate on fake images, indicating it may be more sensitive to tampered images. Despite its lightweight architecture, the custom CNN achieved performance comparable to deeper models, indicating its potential for real-time deployment or use in resource-constrained environments.

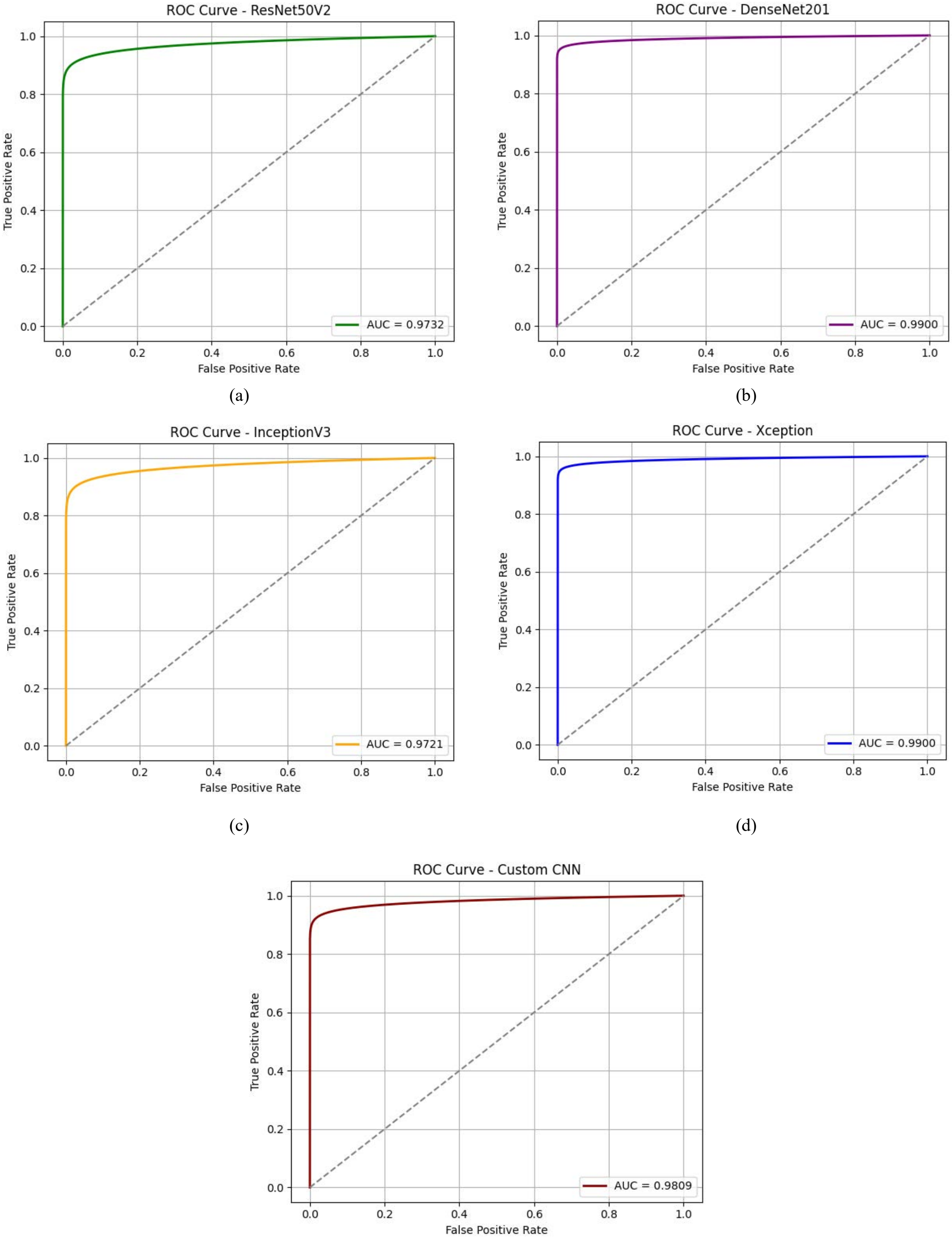
3) ROC-AUC performance for attention-enhanced models

To assess the performance of the enhanced designs in distinguishing genuine and fake facial photos, ROC curves were plotted for all five designs, as shown in Fig. 9. These curves illustrate the relationship between the True Positive Rate (TPR) and False Positive Rate (FPR) at various classification levels.

AUC serves as the evaluation metric, offering a single score that reflects the model's effectiveness in separating authentic images from fake ones [25].

The experimental results show that all models' AUC values are higher than 0.97, demonstrating their superior classification performance. The AUC value of the

Xception model is the highest, at 0.9911, indicating that both sensitivity and specificity are excellent. The robustness of InceptionV3 and ResNet50V2 was tested, and all confirmed the robustness. At the same time, DenseNet201 and custom CNN remained similar. The custom CNN degrades slowly with a simpler architecture, emphasizing its cost efficiency for applications with limited processing resources.



(a)

(b)

(c)

(d)

(e)

Fig. 9. ROC curves and corresponding AUC scores for the five CBAM-enhanced deepfake detection models with full fine-tuning: (a) ResNet50V2, (b) DenseNet201, (c) InceptionV3, (d) Xception, and (e) Custom CNN.

4) Explainability through Grad-CAM visualization

Grad-CAM was utilized to enhance the interpretability and transparency of model predictions for both authentic and counterfeit face data [26]. Grad-CAM emphasizes class-discriminative areas in the input image by calculating the gradient of the predicted output with respect to the feature mappings in the final convolutional layer. This generates a heatmap that visually represents the regions upon which the model predominantly relied for its categorization decision. Fig. 10 illustrates the resulting Grad-CAM heatmaps, which are color-coded to represent varying degrees of model attention:

- The blue regions indicate areas of significant importance, where the model focused its decision-making process.
- Green and yellow regions signify moderate significance.
- Red areas indicate minimal or absent attention, suggesting a restricted impact on the ultimate forecast.

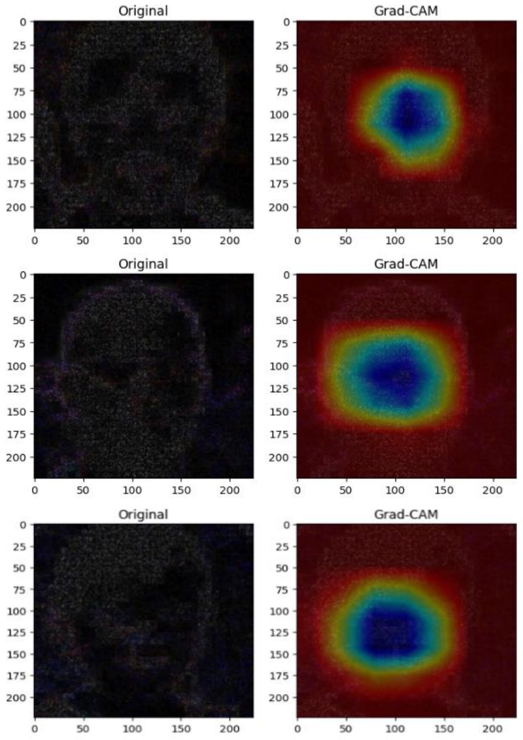


Fig. 10. Grad-CAM heatmaps visualizing activation regions for fake face samples.

In the examined counterfeit facial samples, the model consistently emphasized fundamental facial features, specifically the eyes, nose, and mouth, which are the most vulnerable to alteration. This focused attention pattern in different cases supports the notion that the CBAM-enhanced models are effective at identifying subtle flaws created by deepfake methods, especially those utilizing GAN techniques. This visualization is particularly crucial in forensic or high-stakes contexts, where model transparency is critical. Grad-CAM enhances the credibility and accountability of the deepfake detection system by demonstrating that the model's decisions are based on semantically significant and manipulation-sensitive areas.

C. Model Complexity Analysis

In addition to classification accuracy, evaluating the computational complexity of the proposed models is essential. This complexity is often measured using two primary metrics: the number of trainable parameters and the number of Floating-Point Operations Per Second (FLOPs), which report in term of Giga Floating Point Operations Per Second (GFLOPs).

The number of parameters reveals the model's memory requirements and storage footprint, while FLOPs indicate the mathematical operations the model performs during inference. Models with lower FLOPs are generally more efficient and suitable for real-time applications or deployment on resource-constrained devices. Table V compares the five proposed models in terms of trainability parameters and estimated FLOPs. All models were enhanced with the CBAM attention mechanism.

TABLE V: COMPARISON OF FLOPS, PARAMETERS, AND TEST ACCURACY FOR CBAM-ENHANCED MODELS

Model	FLOPs	Parameters	Test Accuracy (%)
InceptionV3 + CBAM	0.05 GFLOPs	23,116,163	95.00
Xception + CBAM	0.04 GFLOPs	22,174,859	95.36
ResNet50V2 + CBAM	0.05 GFLOPs	24,878,179	94.00
DenseNet201 + CBAM	0.04 GFLOPs	20,237,489	92.00
Custom CNN + CBAM	0.01 GFLOPs	3,026,926	92.52

D. Ensemble Strategy Evaluation

Five ensemble learning approaches were examined for enhancing the developed deepfake detection system as well as its predictive ability [27]. These were weighted averaging, hard voting, simple averaging, logistic regression, and stacking using XGBoost. All approaches combined prediction probabilities from five singly fine-tuned models, Xception, ResNet50V2, InceptionV3, DenseNet201, and a custom CNN, all trained on ELA+SFED preprocessed images. As summarized in Table VI, all the ensemble methods improved the classification performance; among them, stacking on XGBoost achieved the best accuracy with 97.25%, followed by LR with 96.94% and weighted average with 96.90%. They serve as verification that ensemble learning can effectively harness a variety of model architectures to achieve more robustness, stability, and lower bias in deepfake detection systems.

XGBoost was chosen as the meta-learner because it is well-known for capturing non-linear relationships between base models. Unlike simple methods like logistic regression, XGBoost employs tree-based learning, allowing for regularization and effectively understanding how features interact, which enables it to learn complex decision boundaries. This renders our method more accurate, better generalized, and more resistant to overfitting. More importantly, its robustness and scalability make it particularly applicable to large-scale image-based classification problems, such as deepfake detection.

TABLE VI: RESULTS OF ENSEMBLE METHODS COMPARISON

Ensemble Method	Accuracy
Weighted Averaging	0.9690
Hard Voting Accuracy	0.9625
Averaging Ensemble Accuracy	0.9679
Logistic Regression Accuracy	0.9694
stacking (XGBoost)	**0.9725**

TABLE VII: COMPARATIVE PERFORMANCE OF RECENT DEEPFAKE DETECTION METHODS

Reference	Method	Dataset	Accuracy
[4]	MLP + LSTM hybrid	140K Real and Fake Faces	74.7 %
[7]	Pretrained VGG16 + Custom CNN	140K Real and Fake Faces	94.67%
[11]	Hybrid VGG16 + Custom CNN (DFP)	140K Real and Fake Faces	94.00%
]9[	Xception + Transfer Learning + FC Layers + Dropout	140K Real and Fake Faces	95%
[28]	Conv2D-based lightweight CNN	140K Real and Fake Faces	94.54%
Proposed Work (2025)	CNN-based (Xception, ResNet50V2, InceptionV3, DenseNet201, Custom CNN) + CBAM + Ensemble (Stacking)	140K Real and Fake Faces	**97.25%**

E. Comparison of our Proposed Method with Previous Works

To further assess the effectiveness of the proposed deepfake detection framework, its performance is compared against several recent state-of-the-art methods that employed the same dataset (140K Real and Fake Faces). Table VII summarizes the comparative analysis based on test accuracy, F1-score, and AUC, showing that the proposed ensemble model significantly outperforms earlier approaches in all metrics.

VI. CONCLUSION

This paper presents a deepfake detection pipeline that includes advanced image preprocessing, robust convolutional neural networks, and explainability modules. The use of ELA and SFED techniques enhanced forensic artifacts, enabling the models to detect manipulations more effectively. Implementing CBAM attention modules made it easier to identify altered areas in images, while Grad-CAM provides visual explanations of the model's decisions. Experimental results on the 140K Real and Fake Faces dataset confirmed the effectiveness of the proposed method, achieving a top accuracy of 97.25% with a combined strategy involving XGBoost. Our findings suggest that combining image analysis, attention mechanisms, and ensemble learning offers a strong solution to the growing problem of fake face generation. Additionally, the proposed scheme is highly practical for real-world use, thanks to its high accuracy, low false positive and false negative rates, and interpretability. It is suitable for media authentication platforms, digital forensic tools, content moderation systems, online identity verification, and law enforcement investigations. The model balances performance and ethical trustworthiness in AI for sensitive applications by integrating technical excellence with transparency.

VII. FUTURE WORK

The proposed method has demonstrated outstanding effectiveness in detecting static image-based deepfakes. However, there are other avenues for future study. A natural extension is adapting the current system for deepfake video content, which introduces additional challenges such as temporal consistency and frame-level editing.

Temporal models can be incorporated to enhance the system's ability to capture and analyze the temporal dynamics of forgery across successive frames. An additional improvement could be to consider a simplified version of the ensemble model for real-time applications in mobile or edge devices. Techniques such as model pruning, quantization, and knowledge distillation can be explored to reduce computational cost while maintaining accuracy.

Additionally, increasing the training dataset by including ethnically, age, and gender-balanced synthetic facial data would improve the model's fairness and generalizability. Furthermore, using more XAI methods, such as SmoothGrad-CAM++ and attention flow tracking, could help us understand how the model makes decisions and build trust in legal situations.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Bushra Tariq Abdul-Hafiz prepared all the articles, summarized all the data, and made the articles easy to read. Farah Abbas Obaid Sari audited and reported on relevant research articles. All authors had approved the final version.

ACKNOWLEDGMENT

The first author also appreciates the supervisors who have supported their research writing and revision efforts. Furthermore, we thank all the people who contributed to furnishing the information required for this research.

REFERENCES

[1] A. M. Jaber and F. A. O. Sari, "Enhancing of features for road crack image using EEcGANs," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 14, no. 2, pp. 108–114, 2025. doi: 10.18178/ijeetc.14.2.108-114

[2] G. Gupta, K. Raja, M. Gupta, T. Jan, S. T. Whiteside, and M. J. E. Prasad, "A comprehensive review of DeepFake detection using advanced machine learning and fusion methods," *Electronics*, vol. 13, 2024. doi: 10.3390/electronics13010095

[3] G. Naskar, S. Mohiuddin, S. Malakar, E. Cuevas, and R. Sarkar, "Deepfake detection using deep feature stacking and meta-learning," *Heliyon*, vol. 10, no. 4, 2024. doi: 10.1016/j.heliyon.2024.e25933

[4] F. R. Hamade, M. Habiban, and A. A. H. Alrammahi, "DDoS attack detection using machine learning and improved clustering algorithm," *International Journal of Electrical and Electronic Engineering & Telecommunications*, vol. 14, no. 2, pp. 82–87, 2025. doi: 10.18178/ijeetc.14.2.82-87

[5] F. A. Nama, A. J. Obaid, and A. A. H. Alrammahi, "Credit card fraud detection and classification using deep learning with support vector machine techniques," *Lecture Notes in Networks and Systems*, vol. 788, pp. 399–413, Dec. 2023.

[6] J. Atwan, M. Wedyan, D. Albashish *et al*., "Using deep learning to recognize fake faces," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 15, no. 1, p. 113, 2024. doi: 10.14569/IJACSA.2024.01501113

[7] J. Mallet, N. Krueger, M. Vanamala, and R. Dave, "Hybrid deepfake detection utilizing MLP and LSTM," in *Proc. the 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering*, Tenerife, Spain, 2023. doi: 10.1109/ICECCME57830.2023.10252552

[8] J. Mallet, L. Pryor, R. Dave, and M. Vanamala, "Deepfake detection analyzing hybrid dataset utilizing CNN and SVM," in *Proc. 2023 7th Int. Conf. Intell. Syst., Metaheuristics & Swarm Intell. (ISMSI)*, Singapore, Apr. 2023, pp. 7–11. doi: 10.1145/3596947.3596954

[9] B. Ghita, I. Kuzminykh, A. Usama, T. Bakhshi, and J. Marchang, "Deepfake image detection using vision transformer models," in *Proc. 2024 IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Tbilisi, Georgia, Jun. 2024, pp. 332–335. doi: 10.1109/BlackSeaCom61746.2024.10646310

[10] N. Perišić and R. Jovanović, "Convolutional neural networks for real and fake face classification," in *Proc. 2022 Int.Scientific Conf. on Information Technology and Data Related Research*, Belgrade, Serbia, 2022. doi: 10.15308/Sinteza-2022-29-35

[11] S. Ghosh and S. Dey, "Deep Fake detection using CNN," *International Journal of Science, Engineering and Technology,* vol. 12, no. 3, pp. 180–185, 2024.

[12] S. Tyagi and D. Yadav, "MiniNet: A concise CNN for image forgery detection," *Evolving Systems*, vol. 14, no. 3, pp. 545–556, 2022.

[13] A. V. Nadimpalli and A. Rattani, "On improving cross-dataset generalization of deepfake detectors," arXiv preprint, arXiv:2204.04285, Apr. 2022.

[14] A. Raza, K. Munir, and M. Almutairi, "A novel deep learning approach for deepfake image detection," *Appl. Sci*., vol. 12, no. 19, 9820, Sep. 2022. doi: 10.3390/app12199820

[15] K. D. V. N. Vaishnavi, L. H. Bindu, M. Sathvika, K. U. Lakshmi, M. Harini, and N. Ashok, "Deep learning approaches for robust deep fake detection," *World J. Adv. Res. Rev.*, vol. 21, no. 3, pp. 2283–2289, 2023. doi: 10.30574/wjarr.2024.21.3.0889

[16] R. Rafique, R. Gantassi, R. Amin, J. Frnda, A. Mustapha, and A. H. Alshehri, "Deep fake detection and classification using error-level analysis and deep learning," Scientific Reports, vol. 13, 7422, 2023. doi: 10.1038/s41598-023-34629-3

[17] D. N. Raković, "Error Level Analysis (ELA)," *Tehnika*, vol. 78, no. 4, pp. 445–451, 2023. doi: 10.5937/tehnika2304445R

[18] R. Gorle and A. Guttavelli, "Enhanced image tampering detection using error level analysis and CNN," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 1, pp. 19683–19689, Feb. 2025.

[19] B. Tian and W. Wei, "Research overview on edge detection algorithms based on deep learning and image fusion," *Security and Communication Networks*, vol. 2022, 1155814, Sep. 30, 2022. doi: 10.1155/2022/1155814

[20] W. H. Abir, F. R. Khanam, K. N. Alam *et al*., "Detecting deepfake images using deep learning techniques and explainable AI methods," *Int. Autom. Soft Comput.*, vol. 35, no. 2, pp. 2151–2169, 2023. doi: 10.32604/iasc.2023.029653

[21] H. Lin, W. Luo, K. Wei, and M. Liu, "Improved Xception with dual attention mechanism and feature fusion for face forgery detection," in *Proc. 2022 4th Int. Conf. Data Intell. Secur. (ICDIS)*, Shenzhen, China, Aug. 2022. doi: 10.1109/ICDIS55630.2022.00039

[22] S. Mamarasulov, L. Chen, C. Chen, Y. Li, and C. Wang, "Data augmentation with attention framework for robust deepfake detection," *Visual Comput*., vol. 41, no. 7, pp. 4779–4798, Nov. 2024.

[23] M. Sabeena and L. Abraham, "Convolutional block attention based network for copy-move image forgery detection," *Multimedia Tools Appl.*, vol. 83, no. 1, pp. 2383–2405, May 2024.

[24] H. Zhang and K. J. B. Ogasawara, "Grad-CAM-based explainable artificial intelligence related to medical text processing," *Bioengineering*, vol. 10, no. 9, 1070, Sep. 2023. doi: 10.3390/bioengineering10091070

[25] Y. Zhang, W. Gao, C. Miao, M. Luo, J. Li, W. Deng, Z. Li, B. Hu, W. Yao, W. Zhou, T. Gong, and Q. Chu, "Global multimedia deepfake detection: Towards multi-dimensional facial forgery detection," arXiv preprint, arXiv:2412.20833, Dec. 2024.

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 618–626. doi: 10.1109/ICCV.2017.74

[27] H.-W. Zhang, Y.-R. Wang, B. Hu *et al*., "Using machine learning to develop a stacking ensemble learning model for the CT radiomics classification of brain metastases," *Scientific Reports*, vol. 14, no. 1, 28575, 2024. doi: 10.1038/s41598-024-80210-x

[28] J. Sharma, S. Sharma, V. Kumar, H. S. Hussein, and H. A. Alshazly, "Deepfakes classification of faces using convolutional neural networks," *Traitement du Signal*, vol. 39, no. 3, Jun. 2022. doi: 10.18280/ts.390330

Bushra Tariq Abdul-Hafiz earned a bachelor's degree in computer science from the University of Al-Qadisiyah in 2012. She is currently pursuing a master's degree in graduate studies. Her research interests include deep learning and computer vision.

Farah Abbas Obaid Sari received her master's degree in information technology from Dr. Babasaheb Ambedkar Marathwada University in Aurangabad, India, and her Ph.D. degree in data mining from Tambov State Technical University, Russia, in 2022. Currently, he works at the University of Kufa in Najaf, Iraq. His research interests include computer vision and artificial intelligence.