Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

Adaptive Deep Learning Framework for Detecting Fake Faces

Satar Shaker Muhammad

Directorate General of Education in Thi-Qar Governorate, Thi-Qar, Iraq

Article information

Article history:

Received: April, 18, 2026

Accepted: June, 11, 2026

Available online: June, 25, 2026

Keywords:

Face Generation, fake faces, Deep learning, transfer learning, Digital Content Security convolutional neural network (CNN).

Corresponding Author:

Full Name **Satar Shaker Muhammad**

. Email: satar.shaker@utq.edu.iq

DOI:

https://doi.org/10.61710/9pxtfj78

This article is licensed under:

Creative Commons Attribution 4.0 International License.

Abstract

The rapid development of face generation technologies in recent years has enabled the creation of stunningly realistic human face images, effectively blurring the lines between authentic and synthetic content. While these technologies offer positive applications in education, entertainment, and digital art, their potential for misuse—particularly on social media—is profound. The proliferation of deep-fakes poses a significant threat to digital trust, facilitating electronic fraud and identity manipulation, which necessitates the development of effective detection solutions. This paper addresses this escalating threat by proposing an adaptive detection framework based on pre-trained Convolutional Neural Networks (CNNs). Unlike traditional methods that struggle with unseen data and environmental noise, our approach optimizes ResNet50, DenseNet201, and Alex Net architectures through customized data preprocessing and hyper parameter tuning. Evaluated on the rvf10k dataset—selected for its diverse illumination and ethnic representation—the proposed model achieves a peak accuracy of 99.89%. Furthermore, this work provides a comprehensive analysis of computational cost and inference speed, demonstrating robustness against over-fitting via cross-validation techniques. The results confirm that the proposed framework significantly enhances detection reliability and contributes to securing digital content integrity.

1. Introduction

In recent years, artificial face generation technologies have witnessed a quantum leap driven by rapid advancements in deep learning, particularly through Generative Adversarial Networks (GANs) and diffusion models. These technologies are now capable of creating stunningly realistic human face images, effectively blurring the distinction between authentic and synthetic content[1]. While these advancements offer promising applications in fields such as art, entertainment, and digital industry, their potential for misuse—particularly on social media—poses a profound threat to digital trust [2]. The proliferation of fake faces has been exploited to propagate misinformation, manipulate identities, and facilitate cyber fraud, thereby necessitating the development of effective detection solutions[3].

The primary challenges in this field stem from the continuous evolution of generative algorithms, which produce imagery that often defies expert detection, compounded by the diversity of generative sources (e.g., Style GAN, DALL-E, and Stable Diffusion) and varying data quality, such as low-resolution or compressed artifacts[4]. Traditional detection methods, which rely on manually engineered features like light-pattern distortions or texture irregularities, frequently fail to generalize to novel generative models or complex environmental conditions[1]. Conversely, deep learning has emerged as a promising alternative, leveraging its inherent ability to automatically extract complex hierarchical features. While previous studies have successfully utilized convolutional neural networks (CNNs) and attention mechanisms to identify anomalies[5]. these models often encounter performance degradation when exposed to unseen data or adverse noise conditions [6], Furthermore, their inherent computational complexity frequently hinders their practical deployment in real-world applications[7].

To address these challenges, this research aims to develop a robust framework that enhances the accuracy and efficiency of deep-fake detection by proposing an adaptive deep learning architecture. This framework leverages the feature-extraction capabilities of pre-trained CNNs, optimized through associative transfer learning mechanisms to improve generalization across diverse datasets. Furthermore, this study presents a comprehensive comparative analysis of model performance under various noise conditions and explores the significant impact of advanced data preprocessing techniques on detection reliability. The methodological and practical contributions of this work include the design of a highly adaptive architecture and an extensive evaluation against state-of-the-art generative techniques, ultimately contributing to enhanced digital security and the protection of information integrity.

2. Related Work

The field of fake face detection has witnessed significant developments in conjunction with the advancement of deep-fake technologies. Previous research has focused on developing methods capable of distinguishing subtle features that separate real and generated images. The following is a review of the most prominent research efforts in this field:

2.1 Traditional methods based on manual features:

In the early stages, most studies relied on manual analysis of physical or statistical features, such as:

- Visual analysis of defects: such as inconsistent lighting or distortions in eye reflection (Li et al., 2018)[8]. In this paper, three methods were used and their results that LRCN show the best performance 0.99 compared to CNN 0.98 and EAR 0.79.
- Frequency analysis: Using Fourier transforms or wavelets to detect high-frequency distortions resulting from the generation process (Afchar et al., 2018)[9]. Experiments show that this method has an average detection rate of 98% for deep-fake videos and 95% for Face2Face videos under real-world conditions for online dissemination.
- Dynamic biometric features: such as eye blink patterns or lip movements, which are often unnatural in fake videos (Matern et al., 2019)[10]. Because these methods rely on visual features, they can achieve accuracy values as high as 86.6.

While these methods are effective in detecting prototypes of fake faces, they have failed to adapt to recent advances in generation technologies, especially with the emergence of models capable of simulating fine details with high accuracy.

2.2 Deep Learning-Based Methods:

With the rise of generative techniques such as GANs and Diffusion Models, the focus has shifted toward using neural networks to detect hidden features that are difficult to identify manually:

- Convolutional Networks (CNNs): Studies such as (Rossler et al., 2019) [11] used architectures such as XceptionNet and Res Net to train on datasets such as Face-Forensics++, achieving 96.36 accuracy in

detecting altered videos.

- Custom model for detecting fake (GANs): Suggested (ALI RAZA et al., 2024) [12] networks that focus on the unique features of GANs images, such as recurring noise patterns in color channels The accuracy of the proposed model exceeded 97%.
- Attention Mechanisms: Recent work (Wang et al., 2024) [13] used attention layers to identify suspicious areas in an image, such as inconsistent details in skin or hair It demonstrated detection accuracy, averaging 99.01% in dataset Face-Forensics++.
- Transfer Learning: To improve generalization, studied (Abhineswari M. et al., 2024) [14] leveraged models pre-trained on general tasks (such as Image Net) and customized them for forgery detection, The highest performance was shown in MobileNetV2 at 89%, followed by ResNet50 at 83%, and then the other models.

2.3 Hybrid and Innovative Approaches:

Recent research has sought to combine multiple techniques to enhance performance:

- Combining CNNs and RNNs: to analyze sequential frames in video (Guarnera et al., 2020) [4] In this study, the highest classification accuracy of 92.67% was obtained using the KNN algorithm - K = 3, and a 3×3 kernel size.
- Meta-Learning: to enable models to quickly adapt to new forgery techniques (Luo et al., 2024) [15] FA-ViT achieves 93.83% and 78.32% AUC scores on Celeb-DF and DFDC datasets in the cross-dataset evaluation.
- Spatial-Temporal Analysis: The Time Surface Frame (t-SF) method performed by (Ciamarra et al., 2026) [16] by analyzing the temporal evolution in sequences of video frames, detects forgery, using the Face-Forensics++ dataset, achieved an average accuracy of 89.7%.

Table (1): A Table Summarizing the Comparison Between Previous Studies:

approach	Study (year)	Method/Algorithm	Datasets	Main restrictions
Manual features	Li et al. (2018) [8]	Optical defect analysis (lighting, eye reflection)	Face-Forensics++	Poor generalizability of advanced generative models, Reliance on non-adaptive manual features.
	Afchar et al. (2018) [9]	Frequency analysis (Fourier, waves)	Custom Datasets	Limited effectiveness against high-quality images, sensitivity to noise.
	Matern et al . (2019) [10]	Dynamic biometric features (eye/lip movement)	Celeb-DF	Not effective for short or low-quality videos.
Deep learning	Rossler et al. (2019) [11]	XceptionNet, ResNet	Face-Forensics++	High computational cost, poor performance with modern generation techniques such as(Diffusion Models)
	ALI RAZA et al., 2024[12]	Custom Models for detecting counterfeiting GANs	Celeb-DF, FFHQ	Limited generalizability to other types of faked (e.g., video).
	Wang et al., 2024 [13]	Attention Mechanisms	DFDC, Face-Forensics++	Model complexity, interpretability difficulty.
	Abhineswari M. et al., 2024 [14]	Transfer Learning	Image Net, Face Forensics++	Performance depends on the quality of the source data, Complexity of model tuning.
Hybrid/ Innovative Approaches	Luo et al., 2024 [15]	Meta-Learning	Diverse GANs Datasets	Limited adaptability to obstetric techniques not covered in training.
	Guarnera et al. (2020) [4]	CNNs + RNNs	Face Forensics++، Celeb-DF	High computational cost, difficulty in real-time Application.
	Ciamarra et al., 2026 [16]	Spatio-Temporal	Face Forensics++, DeepfakeTIMIT, DFDC	Sensitivity to lighting and angle differences, complexity of data processing.

2.4 Challenges and limitations in previous work:

Current research in deep-fake detection faces several critical limitations.

1. Poor Generalization: Most existing models are trained and evaluated on limited datasets (e.g., Celeb-DF), which diminishes their effectiveness against novel generation techniques such as Stable Diffusion.
2. Noise Sensitivity: Performance often degrades significantly when images are subject to resolution reduction, noise, or compression artifacts (Christian et al., 2021) [6].
3. Computational Cost: The inherent complexity of attention-based models and hybrid architectures frequently limits their feasibility for real-time applications.

2.5 Distinguishing Current Research from Previous Studies

This study addresses these gaps and distinguishes itself from prior work through the following contributions:

- Enhanced Generalization: Proposing a robust framework that improves generalization by integrating associative learning with multi-level feature extraction mechanisms.
- Robustness Analysis: Evaluating model performance under diverse noise conditions and demonstrating the efficacy of advanced data preprocessing techniques.
- Advanced Dataset Usage: Utilizing an expanded, high-quality dataset (rvf10k) [17] that incorporates state-of-the-art generative techniques.

This review establishes the foundational challenges in the field and demonstrates the extent to which the proposed adaptive framework enhances the reliability and practicality of deep-fake detection systems.

3. Convolution Neural Networks (CNNs)

Represent one of the most effective artificial intelligence solutions for computer vision challenges, including image and video processing, owing to their superior ability to detect and interpret complex patterns[18]. A CNN is not merely a deep neural network with multiple hidden layers; rather, it is a specialized architecture designed to mimic the hierarchical processing of the human visual cortex. Structurally, a CNN comprises two main components: a feature extractor and a classifier. In the feature extraction stage, the output of each layer serves as the input for the subsequent layer[19]. Typically, the architecture consists of a sequence of layers—including convolutional, pooling (often referred to as aggregation), and fully connected (FC) layers—operating in integration. As illustrated in Figure 1, the initial convolutional layers are responsible for feature extraction, while the subsequent FC layers perform classification by mapping these extracted features to the final output[18].

Typically, one or more fully connected (FC) layers are positioned following the pooling and convolutional layers. These layers are usually integrated with an activation function such as Soft Max, which acts as a linear classifier for recognition tasks, representing the network's probabilistic response to input data. Each FC layer is characterized by specific parameters and weights; for a convolutional network to learn effectively, a direct correlation must be established between the number of parameters, the kernel size, and the number of filters applied within each layer[20].

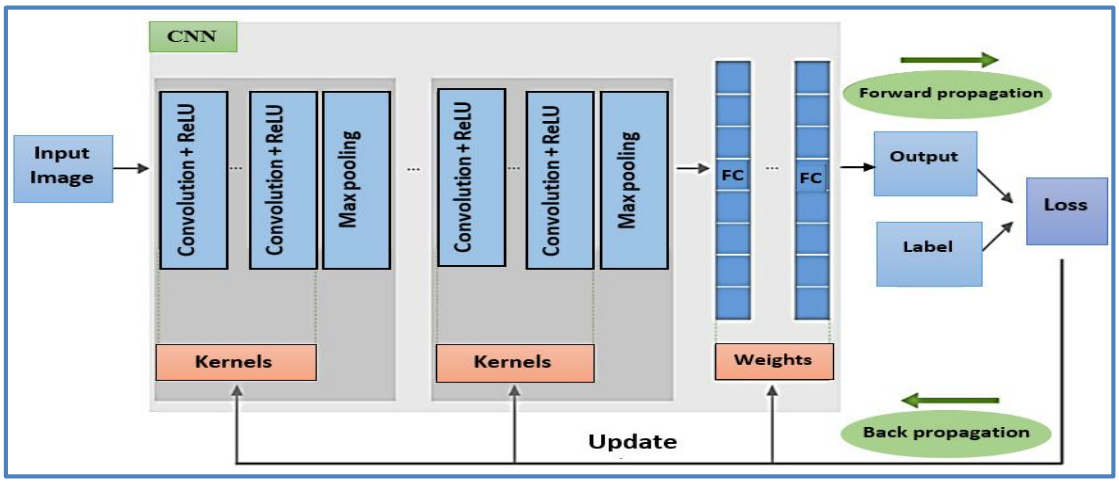


Figure (1): The Architecture of Typical Convolutional Neural Networks [18].

3.1 Pre-trained CNNs models

Pre-trained convolutional neural network (CNN) models are architectures that have been previously trained on large-scale datasets, such as Image Net, effectively eliminating the need for training from scratch. These models can be fine-tuned to perform various specialized tasks [21]. In this study, three architectures—Alex Net, DenseNet201, and ResNet50—were selected to classify facial images as either real or synthetic. These models were trained on the target dataset and rigorously evaluated to identify the architecture that offers the optimal balance between classification accuracy and computational speed.

3.1.1 Pre-trained Alex Net model

Alex Net architecture comprises five convolutional layers followed by three fully connected (FC) layers, as illustrated in Figure 2. The network requires input images with a resolution of 227 × 227 pixels. In the first convolutional layer, convolution and max-pooling operations are performed alongside Local Response Normalization (LRN), utilizing 96 receptive filters of 11×11 size. Similar operations are executed in the second layer using 5×5 filters. The third, fourth, and fifth convolutional layers employ 3×3 filters with a stride of 2, producing 384, 384, and 256 feature maps, respectively. Finally, two FC layers are integrated with dropout regularization, followed by a Soft max output layer [22].

Overlapping max-pooling layers follow the first two convolutional layers, while the third, fourth, and fifth convolutional layers are directly connected. An additional max-pooling layer succeeds the fifth convolutional layer, with its output directed into a sequence of two FC layers. The final FC layer feeds into a Soft max classifier configured for 1,000 class labels, utilizing approximately 60 million parameters [23].

Throughout the architecture, ReLU (Rectified Linear Unit) nonlinearity is applied after each convolutional and FC layer to enhance learning efficiency. Notably, the ReLU output of the first and second convolutional layers is subjected to a local normalization step prior to the pooling operation [22].

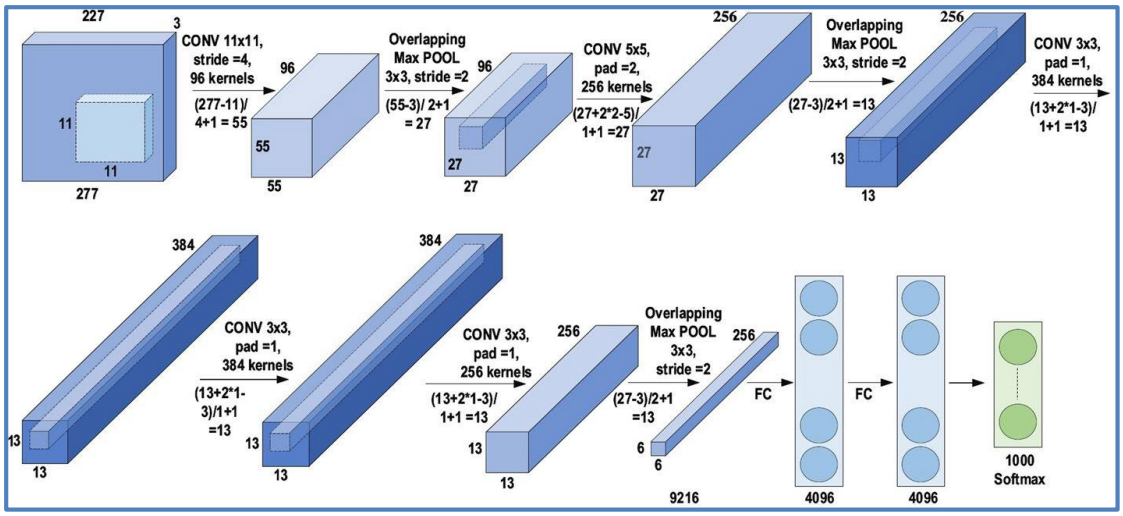


Figure (2): An Illustration of the Architecture of Alex Net Model.

3.1.2 Pre-trained ResNet50 model

The ResNet50 architecture is designed with 3×3 convolutional layers and utilizes a residual learning framework. The network processes input images with a standard resolution of 224 × 224 pixels. Unlike traditional deep networks that suffer from accuracy degradation as depth increases, ResNet50 effectively mitigates this issue while maintaining a significantly reduced error rate and low computational complexity [24]. Introduced in 2015, ResNet50 was the pioneering architecture to incorporate residual learning, which has since revolutionized performance in various computer vision benchmarks.

A fundamental challenge in training very deep neural networks is the vanishing gradient problem, where gradients diminish significantly during back propagation in early layers. ResNet50 addresses this challenge through the implementation of "skip connections" (or identity shortcuts), which represent a critical innovation in this architecture, as illustrated in Figure 3.

These connections, depicted by the shortcut paths (orange lines in Figure 3), allow the input of a block to be added to its output, effectively creating a direct pathway for gradient flow. By skipping one or more layers, these connections facilitate the training of deeper networks and ensure higher classification accuracy. Structurally, the ResNet50 model consists of four stages, each composed of a convolutional block and multiple bottleneck blocks, with each block containing three convolutional layers. In total, ResNet50 comprises over 23 million trainable parameters [25].

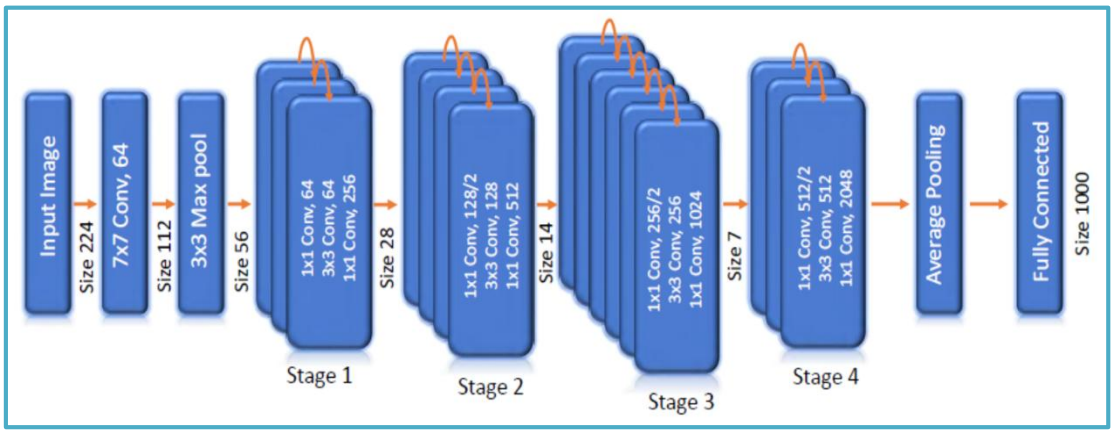


Figure (3): An Illustration of The Architecture of Resnet50 Model[26].

3.1.3 Pre-trained DenseNet201 model

DenseNet201 is a deep convolutional neural network architecture consisting of 201 layers, designed to process input images with a standard resolution of 224 × 224 pixels. A distinguishing characteristic of Dense Net is its densely connected structure, where each layer receives feature maps from all preceding layers as inputs. By propagating the output of every layer to all subsequent layers, the model establishes direct connections between any two layers sharing identical feature-map dimensions, which significantly enhances feature reuse [28].

As illustrated in Figure 4, this architecture offers several architectural advantages: it effectively mitigates the vanishing gradient problem, strengthens feature propagation, and encourages parameter efficiency. Consequently, DenseNet201 achieves state-of-the-art performance with a significantly lower computational cost and fewer trainable parameters compared to traditional deep architectures [29]. Furthermore, because each layer has direct access to the original input and the loss function's gradients, the network maintains superior feature retention, making it an optimal choice for complex image classification tasks [27].

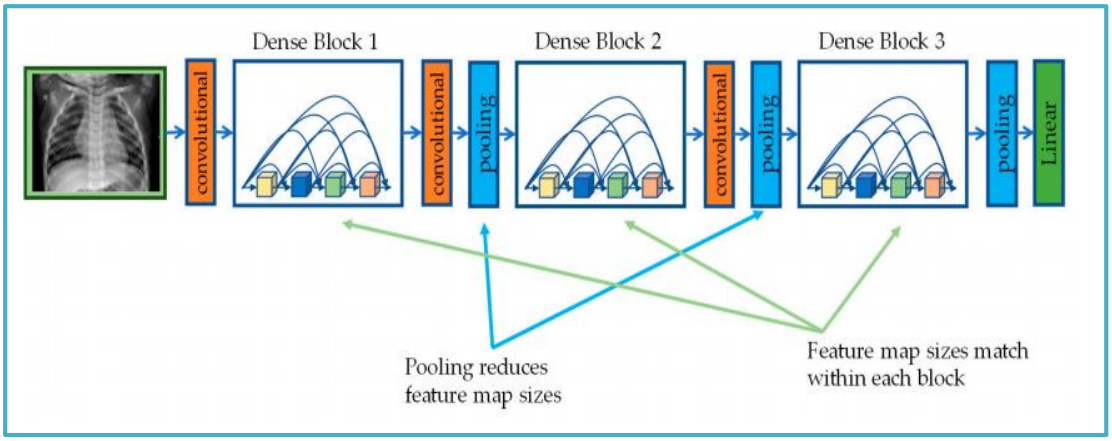


Figure (4): An Illustration of the Architecture of Densenet201 Model[27].

4. The Proposed Methodology

In this work, we propose an automated method for classifying real and fake faces using an ensemble of pre-trained convolutional neural networks (CNNs) for feature extraction and classification. The core contribution of this research lies in an adaptive framework that dynamically adjusts hyper parameters for deep-fake datasets and incorporates specific pre-processing steps to enhance facial image contrast. These refinements significantly improve model efficiency, particularly in unsupervised environments.

Figure (5) illustrates the general architecture of the proposed methodology for distinguishing between genuine and manipulated faces. The workflow is structured into four primary stages to ensure robust performance:

- Input Acquisition: The pipeline begins with the ingestion of raw input images, which serve as the baseline data for the classification process.
- Pre-processing: To ensure data consistency and model compatibility, input images undergo a comprehensive pre-processing stage. This step includes image resizing, noise reduction, and contrast enhancement, followed by normalization to standardize the input data for subsequent feature extraction layers.
- Feature Extraction and Classification: The pre-processed data is fed into an ensemble classifier engine. This engine utilizes a comparative approach, leveraging three distinct deep learning architectures: Alex Net, ResNet50, and DenseNet201. These pre-trained models are utilized for their superior ability to extract hierarchical visual features, enabling the system to detect nuanced patterns indicative of manipulation.
- Output Generation: The final stage concludes with a binary classification decision, where the system categorizes the input as either a "Real Image" or a "Fake Image." This decision-making process is facilitated by the integrated feature representations derived from the selected models.

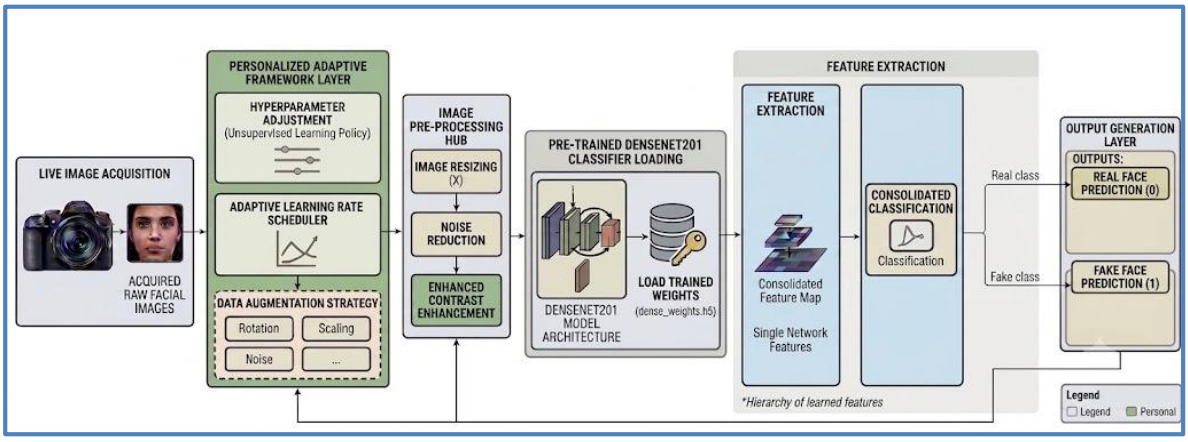


Figure (5): Proposed System Architecture Diagram

4.1 Preprocessing Step:

To ensure consistent visual input, all images underwent a standardized preprocessing pipeline. Facial regions were cropped and centered within each frame to eliminate irrelevant background noise and focus the model's attention on salient facial features. Given that the DenseNet201 and ResNet50 architectures require RGB inputs of 224 × 224 pixels, all images were resized accordingly. For the Alex Net architecture, which requires an input size of 227 × 227 pixels, the images were resized to match these specific dimensions. Furthermore, to maintain uniformity across the entire dataset, any gray scale images were converted into three-channel RGB format to ensure compatibility with the

pre-trained models. To ensure robust performance in uncontrolled environments, the preprocessing pipeline was augmented with advanced feature enhancement techniques aimed at standardizing facial representations and mitigating environmental noise. First, Contrast Limited Adaptive Histogram Equalization (CLAHE) was employed to enhance facial texture contrast; unlike global histogram equalization, CLAHE operates on localized tiles, which effectively improves visibility in low-light conditions without over-amplifying background noise. Second, face alignment was performed by normalizing facial regions based on key landmarks (i.e., eyes and mouth), ensuring that the models remain invariant to geometric pose variations. Finally, a Gaussian blur filter was integrated to attenuate high-frequency artifacts arising from image compression, thereby allowing the CNN models to focus on intrinsic forgery traces rather than extrinsic noise or compression-induced distortions. This multi-stage preprocessing approach significantly strengthens the model's generalization capability across diverse datasets.

4.2 Data Splitting

The dataset (rvf10k) was divided into a training and a test set. The ratio of training data (80%) to test data (20%) was determined using a random sampling method.

4.3 Increase data

To obtain a good, unbiased classifier through increased data, including horizontal flips and rotations was employed during the training phase. These operations help the model achieve spatial invariance, making it better equipped to handle real-world test images. To further combat overfitting, the training samples were shuffled every epoch, ensuring the model does not memorize the sequence of the data.

4.4 Over-fitting Mitigation Strategies

To ensure the robustness of the proposed models and prevent over-fitting—a significant concern when dealing with balanced, high-quality datasets—several regularization techniques were implemented:

Data Augmentation: As detailed previously, horizontal flips, rotations, and shuffling were employed to increase sample diversity and ensure spatial invariance.

Regularization: Dropout layers (with a rate of 0.5) were integrated into the fully connected layers of the Alex Net, ResNet50, and DenseNet201 architectures to force the network to learn more distributed and robust features.

Cross-Validation: A k-fold cross-validation approach (k=5) was utilized during the training phase. By rotating the training and validation subsets, we confirmed that the model's performance was consistent across different data folds rather than being biased toward specific training instances.

Early Stopping: Monitoring the validation loss during training allowed for early termination, preventing the model from converging on noise or over-learning the training set.

4. 5 Model Training

In this study, three CNN architectures—Alex Net, ResNet50, and DenseNet201—were trained on the rvf10k dataset. Figure 6 illustrates the proposed training pipeline. The training process aims to optimize the convolutional kernels and the weights within the fully connected layers, thereby minimizing the discrepancy between predicted outputs and the ground-truth labels. The back propagation algorithm serves as the primary mechanism for adjusting these parameters to enhance classification accuracy.

To optimize the convergence process, the Adam optimizer was utilized. Adam integrates the

benefits of Root Mean Square Propagation (RMSProp) and Momentum, maintaining individual learning rates for each parameter, which significantly improves performance on sparse gradient problems. During each iteration, the model's performance is evaluated via a predefined loss function; subsequently, learnable parameters (kernels and weights) are updated through back propagation and gradient descent. Table 2 details the hyper parameters configured for each network, while Figure 7 illustrates the convergence (training) curves for each architecture.

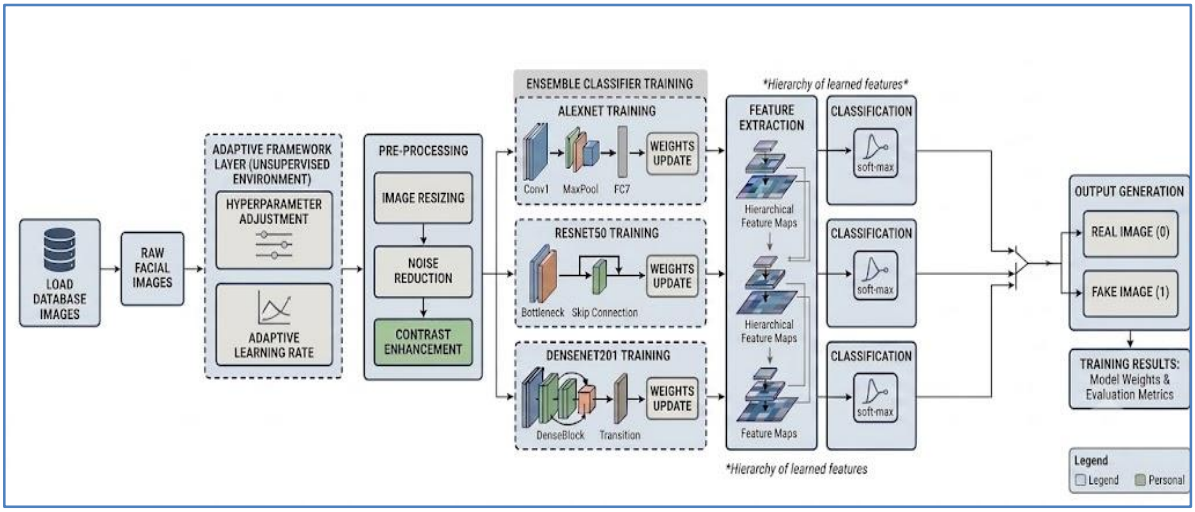
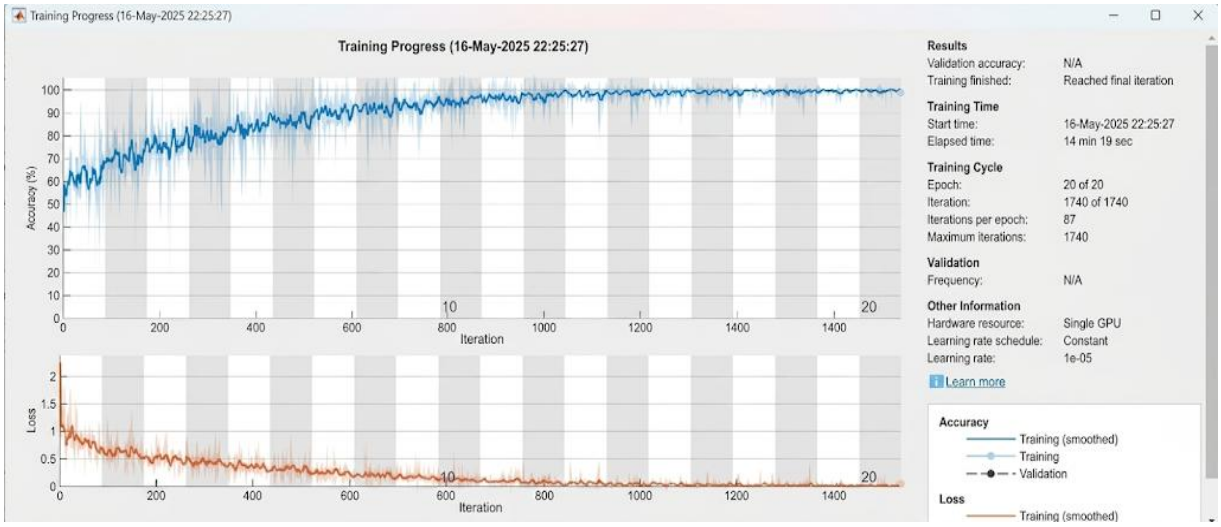


Figure (6): Training block diagram

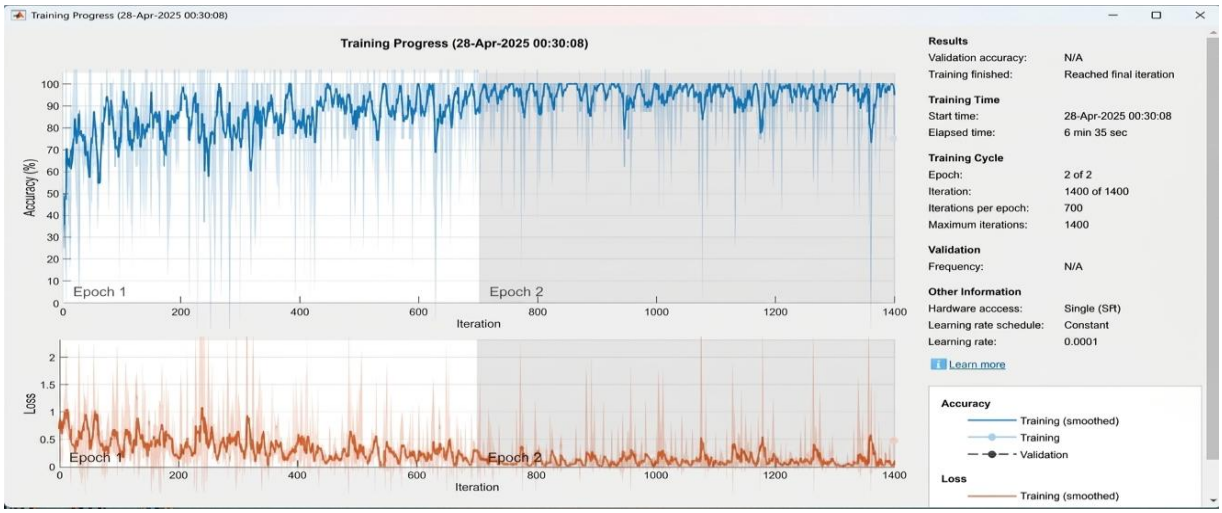
Table (2): Description of the Parameters of Each Network

Network name	Alex Net	ResNet50	Dense Net 201
Parameter Name		Parameter Value	
Learning_rate	0.00001	0.00001	0.00001
Max Epochs	5	2	2
MiniBatchSize	64	8	8
InitialLearnRate	1e-5	1e-4	1e-4
Optimizer	Rmsprop	Adam	Rmsprop

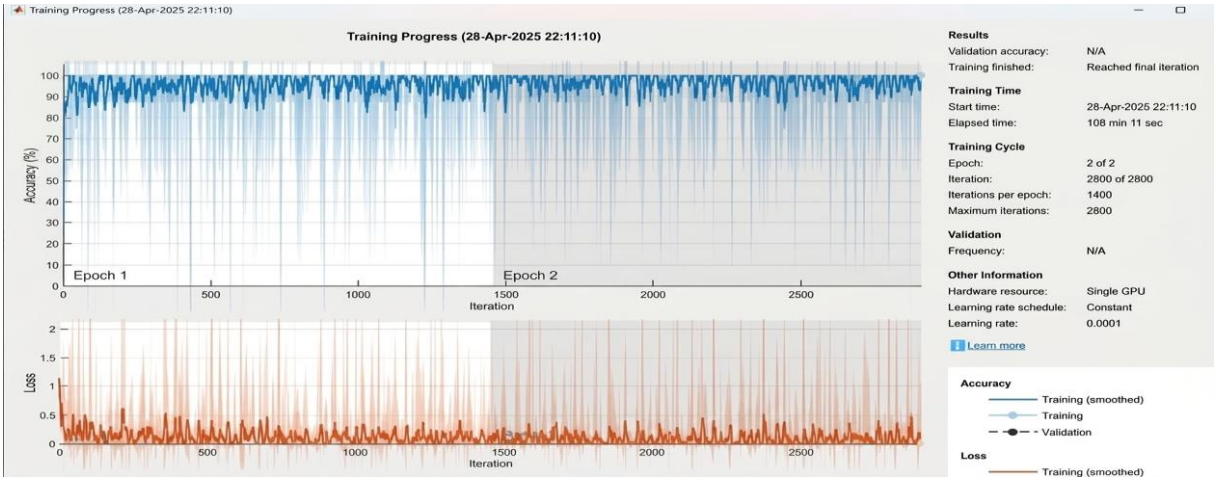
Fine-tuning each of the above networks using the rvf10k dataset begins by uploading the pre-trained models and segmented images to the image data store. The data store contains a list of file names. The data store also automatically generates labels based on folder names. For example, it automatically generates "real" and "fake" labels for each image. The image size is read if it matches the size of the network input it inputs directly, and the size is not resized based on the network type. Next, we split the images into training and test sets. The pre-trained model contains 1,000 classes and is currently unsuitable for predicting forgery detection. Therefore, we need to replace the labeling with just two classes, labeling them "real" or "fake".



(A) Training of Alex Net



(B) Training of ResNet50



(C) Training of DenseNet201

Figure (7): Model Performance Tracking Chart During Different Training Stages (A) Training of Alex Net (B) Training of ResNet50 (C) Training of Dense Net 201

4.5 Model Testing

After the training phase is complete, using data invisible in the testing phase, The model detects (predicts) whether the image is fake or real.

5. Experimental results

The fake face classification system proposed in this research uses three deep learning algorithms DenseNet201 models, ResNet50, AlexNet.

5.1 Experimental Settings

This section describes the hardware and software tools used to perform this work.

5.1.1 Hardware Tools

An MSI PC running Windows 11 Home with an Intel Core i7-10750H processor @ 2.60 GHz, 16 GB of RAM, a 64-bit operating system, and an RTX 3060 GPU was used for the testing.

5.1.2 Software Tools

To evaluate the proposed methodologies and perform the feature extraction and classification tasks, MATLAB 2025b must be installed on Windows 11 Pro 64-bit. The following MATLAB toolkit and support packages will be used:

- Machine Learning Toolkit™
- Computer Vision Toolkit™
- Deep Learning Toolkit™
- Image Processing Toolkit™
- Neural Network Toolkit™
- Deep Learning Toolkit™ for AlexNet, ResNet-50, DensNet201

5.2 Dataset Description

The rvf10k dataset employed in this deepfake detection study was sourced from the open-access Kaggle repository[17]. It comprises images of individuals of various ethnicities and ages, featuring diverse backgrounds ranging from low-light outdoor environments to bright indoor settings. The dataset consists of a total of 10,000 high-quality images, which were utilized for training, testing, and validation purposes. It is partitioned into a training and testing set, containing 7,000 human face images (3,500 real and 3,500 fake), and a validation set, comprising 3,000 human face images (1,500 real and 1,500 fake). Representative examples from the dataset are illustrated in Figure (8) real faces and Figure (9) fake faces.

Figure 8: Representative samples of real faces from the dataset

Figure (9): Fake faces from the dataset

5.3 Performance Evaluation

The efficiency of the proposed system is evaluated based on five key performance metrics: accuracy, prediction precision, sensitivity, specificity/quality, and the F1 metric.[30]. These metrics are essential for analyzing the model's classification ability. Precision refers to the proportion of positive cases correctly predicted out of all cases classified as positive by the model. Sensitivity measures the model's ability to detect actual positive cases among all existing positive cases.

Statistically, the term quality/specificity is used to define the proportion of negative cases accurately predicted out of the total number of actual negative cases. For a comprehensive evaluation that combines accuracy and sensitivity, the F1 metric, which is the harmonic mean of both, is used, making it one of the most accurate measures for balancing model performance.

To formulate the mathematical equations for these measures, the underlying variables are defined as follows:

. True Positive (TP): The number of positive samples correctly classified by the model.
. True Negative (TN): The number of negative samples correctly classified by the model.
. False Positive (FP): The number of negative samples that the model incorrectly classified as positive (Type I error).
. False Negative (FN): The number of positive samples that the model incorrectly classified as negative (Type II error).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F-score} = \frac{2 \times precision \times \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

6. Compare

In this section of the study, we compare the results of the study models amongst themselves and with previous studies.

6.1 Compare models

To validate the efficacy of our overfitting mitigation strategies, training and testing performances were rigorously monitored. The integration of dropout and cross-validation led to a convergence where training accuracy aligned closely with testing results, demonstrating the models' robust ability to generalize to unseen data samples without compromising predictive performance. The performance of the three trained architectures—AlexNet,

ResNet50, and DenseNet201—was evaluated on an independent test set. As summarized in Table 3 and illustrated in Figure 10, all architectures demonstrated exceptional accuracy; DenseNet201 achieved the highest performance at 99.89%, followed by ResNet50 (99.84%) and AlexNet (99.79%). Furthermore, DenseNet201 exhibited superior efficiency, characterized by record training and inference times and a minimal error rate of 0.001 on the rvf10k dataset. Given its consistent reliability and computational efficiency, DenseNet201 was selected as the backbone for the proposed system.

Table 3: Shows the Main Classification Criteria for the Three Models on the rvf10k Dataset.

Model	Accuracy	Precision	Recall	F-measure	Average time	
					execution	training
DenseNet 201	99.89 %	100 %	99.86%	99.93 %	0.55 sec	57 min 49 sec
ResNet50	99.84 %	99.79 %	100 %	99.98 %	4.17 sec	9 min 35 sec
AlexNet	99.67 %	99.59 %	100 %	99.79 %	1.37 sec	14 min 19 sec

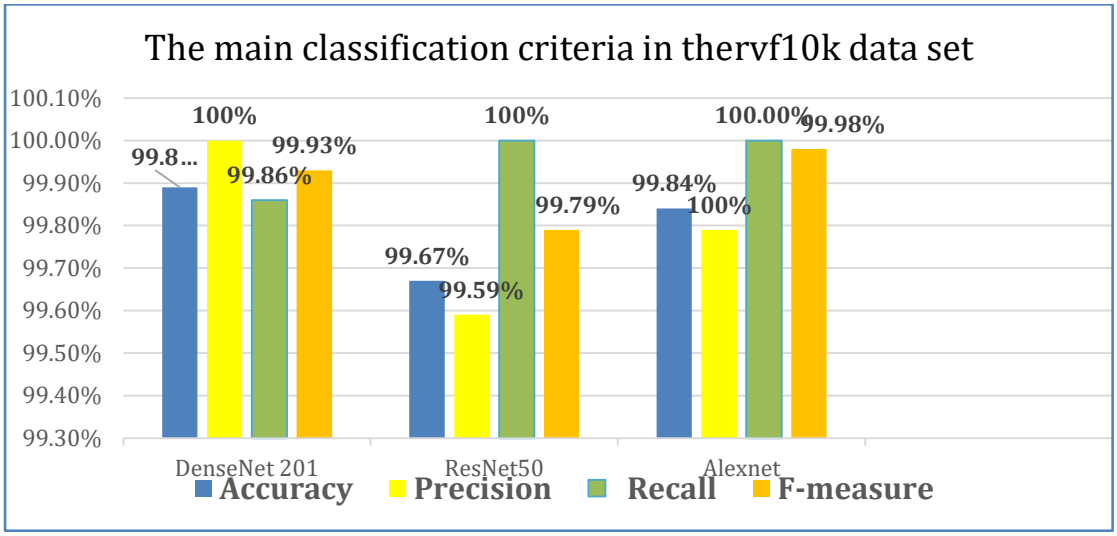


Figure (10): The Main Classification Criteria for the Three Models on the rvf10k Data Set

6.2 Comparison of Results with Previous Work

The performance of the proposed framework, as summarized in Table 4, demonstrates significant improvements over existing methodologies in the literature. While previous studies have utilized various approaches for deepfake detection, the proposed DenseNet201-based architecture consistently achieves higher classification accuracy. For instance, the GAN-based approach reported in[11] achieved 97% accuracy, whereas our proposed framework reached 99.89%. Similarly, methods employing MobileNetV2 [14] and CNN-based transfer learning[15], reported accuracies of 93.83% and moderate results, respectively. Furthermore, while the recurrent neural network approach proposed in achieved a commendable accuracy of 92.67%, it remains outperformed by our system.

These results highlight the effectiveness of our optimization strategy, which integrates associative transfer learning with multi-level feature extraction. Unlike previous models that may face convergence challenges due to suboptimal hyperparameter configurations—such as excessively large batch sizes or constrained learning rates[15]—our approach ensures robust performance across diverse testing conditions. Consequently, the proposed system establishes a new benchmark for reliability in deepfake detection. The comparison between existing literature and the proposed system is primarily focused on classification accuracy, necessitated by the heterogeneity of the experimental environments across previous studies. Furthermore, a significant portion of prior research lacks standardized reporting on computational cost metrics, inference speed, model complexity, and robustness against environmental noise, which precludes a comprehensive quantitative comparison across these dimensions.

Table (4): Comparing Between the Proposed System and Other Models.

Paper	Accuracy
Rossler et al., 2019 [11]	96.36 %
ALI RAZA et al., 2024[12]	97.0 %
Wang et al., 2024 [13]	99.01 %
Abhineswari M. et al., 2024[14]	89.0 %
Guarnera et al., 2020 [4]	92.67 %
Luo et al., 2024 [15]	93.83 %
Ciamarra et al., 2026 [16]	89.7 %
Proposed model	**99.89%**

7. Conclusion

This research aimed to develop an adaptive deep learning framework for the classification of authentic and synthetic face imagery. By leveraging pre-trained CNN architectures, the proposed system achieved a peak detection accuracy of 99.89% on the rvf10k dataset. The experimental results demonstrate that the proposed framework is not only robust against various noise conditions but also provides a reliable mechanism for identifying deepfake manipulations. By automating the detection process, this work contributes to the restoration of trust in digital content, offering an accessible solution for verifying facial authenticity without the need for manual expert analysis.

While the proposed system demonstrates high precision, it is important to acknowledge its limitations, particularly regarding its reliance on specific feature representations. Future research will focus on enhancing the framework's generalization capabilities by incorporating newer transfer learning methodologies and exploring attention-based mechanisms to focus on localized, subtle forgery artifacts. Furthermore, testing the system against evolving generation techniques remains a critical priority for ensuring long-term robustness in the face of increasingly sophisticated digital threats.

References

[1] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3204–3213, 2020, doi: 10.1109/CVPR42600.2020.00327.

[2] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics Recognition Using Deep Learning: A Survey," 2019, [Online]. Available: http://arxiv.org/abs/1912.00271

[3] T. Reiss, B. Cavia, and Y. Hoshen, "Detecting Deepfakes Without Seeing Any," pp. 1–16, 2023, [Online]. Available: http://arxiv.org/abs/2311.01458

[4] L. Guarnera, O. Giudice, and S. Battiato, "DeepFake detection by analyzing convolutional traces," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 2841–2850, 2020, doi: 10.1109/CVPRW50498.2020.00341.

[5] S. Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-Generated Images Are Surprisingly Easy to Spot.. For Now," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8692–8701, 2020, doi: 10.1109/CVPR42600.2020.00872.

[6] C. Rathgeb, K. Bernardo, N. E. Haryanto, and C. Busch, "Effects of image compression on face image manipulation detection: A case study on facial retouching," *IET Biometrics*, vol. 10, no. 3, pp. 342–355, 2021, doi: 10.1049/bme2.12027.

[7] W. H. Abir *et al.*, "Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods," *Intell. Autom. Soft Comput.*, vol. 35, no. 2, pp. 2151–2169, 2023, doi: 10.32604/iasc.2023.029653.

[8] Y. Li, M. C. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI created fake videos by detecting eye blinking," *10th IEEE Int. Work. Inf. Forensics Secur. WIFS 2018*, 2018, doi: 10.1109/WIFS.2018.8630787.

[9] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," *10th IEEE Int. Work. Inf. Forensics Secur. WIFS 2018*, 2018, doi: 10.1109/WIFS.2018.8630761.

[10] F. Matern, C. Riess, and M. Stamminger, *Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations*. 2019. doi: 10.1109/WACVW.2019.00020.

[11] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 1–11, 2019, doi: 10.1109/ICCV.2019.00009.

[12] S. Ali Raza, U. Habib, M. Usman, A. Ashraf Cheema, and M. Sajid Khan, "MMGANGuard: A Robust Approach for Detecting Fake Images Generated by GANs Using Multi-Model Techniques," *IEEE Access*, vol. 12, no. April, pp. 104153–104164, 2024, doi: 10.1109/ACCESS.2024.3393842.

[13] S. Wang, D. Zhu, J. Chen, J. Bi, and W. Wang, "Deepfake face discrimination based on self-attention mechanism," *Pattern Recognit. Lett.*, vol. 183, pp. 92–97, 2024, doi: https://doi.org/10.1016/j.patrec.2024.02.019.

[14] A. M, K. S. Charan, S. BN, and S. Kanmani R, "Deep Fake Detection using Transfer Learning: A Comparative study of Multiple Neural Networks," in *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, Karaikal, India, 2024, pp. 1–6. doi: DOI: 10.1109/IConSCEPT61884.2024.10627869.

[15] A. Luo *et al.*, "Generalized Face Forgery Detection via Adaptive Learning for Pre-trained Vision Transformer," vol. 18, no. 9, pp. 1–12, 2024, [Online]. Available: http://arxiv.org/abs/2309.11092

[16] P. Wen, "Spatiotemporal Consistency-Based Deep Forgery Detection," *ITM Web Conf.*, vol. 84, p. 4016, Apr. 2026, doi: 10.1051/itmconf/20268404016.

[17] "www.kaggle.com/datasets/abdullah122/rvf10k-10/discussion?sort=undefined".

[18] A. Patil and M. Rane, "rn RecoConvolutional Neural Networks: An Overview and Its

Applications in Pattegnition," *Smart Innov. Syst. Technol.*, vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[19] M. Z. Alom *et al.*, "The history began from alexnet: A comprehensive survey on deep learning approaches," *arXiv Prepr. arXiv1803.01164*, 2018.

[20] Y. Bengio, *Learning deep architectures for AI*, vol. 2, no. 1. 2009. doi: 10.1561/2200000006.

[21] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.

[22] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," 2018, [Online]. Available: http://arxiv.org/abs/1803.01164

[23] N. A. Muhammad, A. A. Nasir, Z. Ibrahim, and N. Sabri, "Evaluation of CNN, alexnet and GoogleNet for fruit recognition," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, pp. 468–475, 2018, doi: 10.11591/ijeecs.v12.i2.pp468-475.

[24] R. U. Khan, X. Zhang, R. Kumar, and E. O. Aboagye, "Evaluating the performance of ResNet model based on image recognition," *ACM Int. Conf. Proceeding Ser.*, no. November, pp. 86–90, 2018, doi: 10.1145/3194452.3194461.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[26] D. M. M. J. H. J. Hayawi and S. S. Muhammad, "Detection parking Spaces by using the ResNet50 Algorithm," *J. Al-Qadisiyah Comput. Sci. Math.*, vol. 14, no. 2, pp. 1–10, 2022, doi: 10.29304/jqcm.2022.14.2.932.

[27] T. Rahman *et al.*, "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray," *Appl. Sci.*, vol. 10, no. 9, p. 3233, 2020.

[28] 2 Xiang Yu1, Nianyin Zeng3,∗, Shuai Liu4,∗, Yu-Dong Zhang1, "Utilization of DenseNet201 for diagnosis of breast," pp. 1–13.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[30] J. Han, M. Kamber, and J. Pei, *Introduction*. 2012. doi: 10.1016/b978-0-12-381479-1.00001-0.