True Vision: Image Forgery Detection with Deep Learning Techniques

Harshwardhan Patil[1], Rohit Dongare[2]
[1,2]*School of Computing, MIT Art, Design and Technology University Pune, India*

***Abstract*—The rapid growth of digital image manipulation tools has significantly increased the prevalence of image forgeries, posing serious challenges to media credibility, digital forensics, and information security. This project presents a deep learning–based image forgery detection framework designed to identify and classify tampered images with high reliability. The proposed approach employs convolutional neural network (CNN) architectures to automatically learn discriminative features from forged and authentic images, eliminating the need for handcrafted feature extraction. The model is trained and evaluated on a labelled dataset containing both genuine and manipulated images, including common forgery types such as copy–move and splicing attacks. Experimental results demonstrate that the deep learning–based method achieves superior accuracy and robustness compared to traditional image forensics techniques. Additionally, visual explanation methods are used to highlight manipulated regions, improving interpretability and trust in model decisions. The proposed system shows strong generalisation capability and offers an effective, scalable solution for real-world image forgery detection applications.**

***Index Terms*— Image Forgery Detection, Deep Learning, Convolutional Neural Networks, Digital Image Forensics, Image Manipulation Detection.**

I. INTRODUCTION

Image forgery has emerged as a critical challenge in the digital era dominated by advanced image editing software and deep learning–based content generation tools. With the widespread availability of powerful manipulation techniques, forged images can now be created with high realism, making it extremely difficult for the human eye to distinguish between authentic and manipulated content. Although image editing technologies have legitimate applications in areas such as media production, education, and creative design, their misuse poses serious threats including misinformation, legal disputes, reputation damage, and digital fraud.

Image forgeries primarily involve intentional alterations to visual content in order to misrepresent reality. Common types of image forgery include copy–move forgery, where a region of an image is duplicated and pasted within the same image to conceal or replicate objects, and image splicing, in which regions from multiple images are combined to form a deceptive composite. Additional manipulations such as retouching, rescaling, and object removal further complicate the detection process. These manipulations often leave subtle statistical and structural inconsistencies that are imperceptible to human observers but can be identified through computational analysis.

To combat the growing threat of image forgery, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have gained significant attention in the field of digital image forensics. CNN-based models are capable of automatically learning hierarchical features that capture spatial, texture, and frequency-based inconsistencies introduced during image manipulation. Unlike traditional hand-crafted feature–based methods, deep learning approaches offer improved robustness and adaptability when trained on large and diverse datasets of forged and authentic images.

In this project, we propose an image forgery detection framework based on deep learning techniques that effectively distinguishes between genuine and manipulated images. The system is trained on a labelled dataset containing both authentic and forged images and is evaluated across multiple forgery scenarios. The proposed approach demonstrates high detection accuracy and strong generalisation performance, making it suitable for real-world

applications such as digital forensics, social media monitoring, and multimedia authentication.

II. DEEPFAKE.IMAGE FORGERY

The primary tools used for creating image forgery content include advanced image editing software and deep learning–based models, particularly Convolutional Neural Networks (CNNs) and generative techniques. These modern technologies enable the manipulation of digital images through operations such as copy–move, splicing, object removal, and retouching with high visual realism. As a result, forged images often appear authentic and are difficult for the human eye to distinguish from original images. The increasing sophistication of image manipulation techniques has significantly raised concerns in areas such as digital forensics, media integrity, and information security, thereby highlighting the need for robust image forgery detection methods.

2.1 Autoencoders for Image Forgery Detection

Autoencoders are widely used deep learning models in image forgery detection due to their ability to learn compact and meaningful representations of image data. Unlike their use in content generation, autoencoders in forgery detection are trained to model the normal characteristics of authentic images and identify anomalies introduced during manipulation. An autoencoder consists of two main components:

1. Encoder: The encoder compresses the input image into a low-dimensional latent representation by extracting essential structural and textural features while suppressing redundant information.
2. Decoder: The decoder attempts to reconstruct the original image from the latent representation. For authentic images, reconstruction error remains low, whereas forged images typically produce higher reconstruction errors due to inconsistencies caused by manipulation.

In image forgery detection pipelines, autoencoders are trained primarily on genuine images so that the network learns the natural distribution of real image data. When a forged image such as one created through copy–move or splicing is passed through the model, reconstruction artifacts and pixel-level discrepancies emerge. These anomalies are then analysed to detect and localize forged regions within the image.

Advanced CNN-based encoders, including ResNet and Efficient Net variants, are often integrated into autoencoder frameworks to improve feature extraction capabilities. This approach enhances sensitivity to subtle spatial, texture, and boundary inconsistencies introduced during image manipulation. As a result, autoencoder-based methods provide an effective and scalable solution for detecting complex image forgeries in digital forensics applications

2.2 GAN-Based Image Forgery and Detection

Generative Adversarial Networks (GANs) represent a powerful class of deep learning models capable of learning complex image distributions and generating highly realistic synthetic content. In the context of image forgery, GANs are often used to create manipulated images for tasks such as image splicing, object insertion, and texture modification. These forgeries pose a significant challenge for digital forensics due to their high visual fidelity.

A GAN consists of two core components that are trained simultaneously through adversarial learning:

Generator (G): The generator produces forged or synthetic images by learning the statistical distribution of real images.

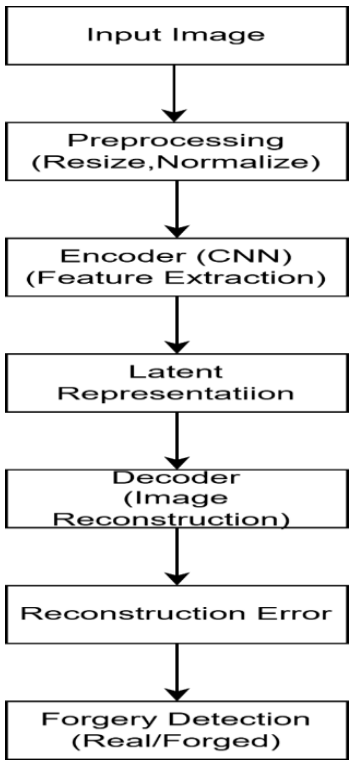


Figure 1.1: Autoencoder-based Image Forgery Detection Framework

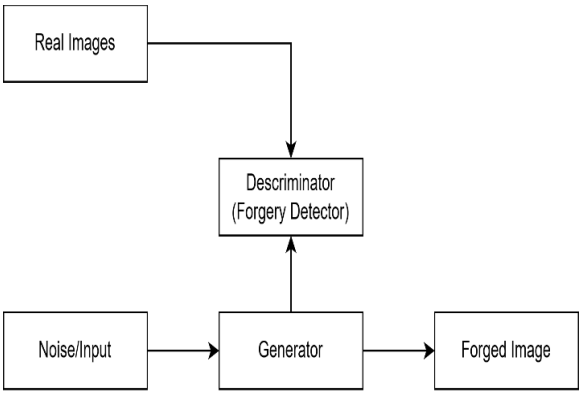


Figure 1.2: GAN-based Image Forgery and Detection Framework

A GAN consists of two core components that are trained simultaneously through adversarial learning:

1. Generator (G): The generator produces forged or synthetic images by learning the statistical distribution of real images from a training dataset. It takes random noise or input conditions and generates image samples that resemble authentic images.
2. Discriminator (D): The discriminator acts as a binary classifier that distinguishes between real (authentic) images and forged (generated) images. Its objective is to correctly identify whether an input image is genuine or manipulated.

During training, the generator attempts to deceive the discriminator by producing increasingly realistic forged images, while the discriminator improves its ability to detect subtle artifacts and inconsistencies. This interaction is modelled as a minimax optimization problem, where the generator minimizes detection probability and the discriminator maximizes classification accuracy.

Although GANs are primarily designed for image synthesis, their discriminator networks are highly valuable for image forgery detection. The discriminator learns rich feature representations that capture anomalies in texture, noise patterns, edge continuity, and lighting inconsistencies introduced during manipulation. These learned features are exploited in forgery detection systems to classify images as real or forged.

Several GAN variants have influenced the evolution of image forgery techniques, including:

- DCGAN: Improved stability in image generation using convolutional layers.
- WGAN: Enhanced training convergence through Wasserstein loss.
- StyleGAN: Enabled high-quality image synthesis with fine-grained control over visual attributes.

While these models improve forgery realism, they also introduce detectable artifacts. Deep learning–based detection frameworks leverage CNNs trained on both authentic and GAN-generated images to identify these artifacts. However, GAN-based systems require large datasets and careful tuning to avoid issues such as mode collapse and overfitting. Therefore, robust detection models must generalize well across unseen manipulation techniques and datasets.

The adversarial relationship between generation and detection highlights the importance of continuously improving deep learning–based image forgery detection techniques to maintain media integrity

2.3 Creation of Deepfake: Tools

Creation of Image Forgery: Tools

Recent advancements in deep learning and computer vision have significantly influenced the creation and manipulation of digital images. The availability of sophisticated image editing software, combined with artificial intelligence–based techniques, has enabled the production of highly realistic forged images. These tools allow users to modify image content through operations such as object insertion, removal, duplication, and enhancement, often leaving minimal visible traces of manipulation.

Traditional image forgery tools include professional photo editing software such as Adobe Photoshop, GIMP, and CorelDRAW, which provide powerful functionalities like layer manipulation, cloning, blending, and retouching. These tools enable copy–move and splicing forgeries by seamlessly combining regions within or across images. Due to their widespread use and ease of access, such tools have played a major role in the growth of forged visual content across digital platforms.

With the emergence of deep learning, AI-based image manipulation tools have further increased the realism of forged images. Techniques based on Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) are now used for image synthesis, super-resolution, inpainting, and texture generation. These methods can automatically learn complex image patterns and generate visually consistent results, making forgery detection more challenging. Models

such as StyleGAN and DCGAN have been widely adopted for high-quality image generation, contributing indirectly to the sophistication of image forgeries.

Mobile and web-based applications have made image manipulation accessible to non-expert users. Tools such as Photoshop Express, Remini, and AI-powered photo editors allow users to alter images with minimal technical knowledge. As a result, forged images are increasingly shared on social media and digital platforms, raising concerns related to misinformation, digital fraud, and content authenticity.

Despite the impressive capabilities of modern image forgery tools, they often introduce subtle artifacts related to texture inconsistencies, edge discontinuities, noise patterns, and lighting variations. These artifacts form the basis for deep learning–based image forgery detection techniques, which aim to identify manipulated images by learning discriminative features that distinguish forged content from authentic images.

Tool Name	Description
Adobe Photoshop	Professional image editing software widely used for copy–move forgery, splicing, object removal, and retouching.
GIMP	Open-source image manipulation tool that supports cloning, layering, and blending for creating forged images.
CorelDRAW	Graphic design tool used for image composition, editing, and manipulation.
Photoshop Express	Mobile-based image editing application enabling quick retouching and content alteration.
Remini	AI-powered image enhancement tool that can modify image details and textures.
StyleGAN	GAN-based deep learning model used for high-resolution image synthesis and manipulation.
DCGAN	Deep convolutional GAN used for generating synthetic images with realistic texture patterns.
Image Inpainting Tools	AI-based tools used to remove or fill objects in images, often leaving subtle artifacts.
Copy-Move Forgery Tools	Software utilities that duplicate and paste image regions within the same image to conceal or replicate objects.
Online AI Image Editors	Web-based platforms that provide automatic image manipulation with minimal user effort.
MyHeritage Deep Nostalgia	AI-based service that animates old photos to simulate realistic movement.
Synthesia	AI-based video generation platform to create avatars for video content.
Deep Video Portraits	Technology that enables realistic facial reenactment in videos.
VoCo (Adobe)	Voice manipulation software that allows speech to be edited or generated.
GANPaint Studio	Interactive tool for editing GAN-generated scenes and images.
Modulate.ai	AI-driven platform for real-time voice modulation and transformation.
FSGAN	Real-time face swapping and reenactment tool.
Reflect.tech	Face-swapping tool that emphasizes realism and fine detail.
SynthEyes	Motion tracking tool for deepfake video creation.
FakeApp	One of the earliest tools for creating deepfake videos through face swapping.

Table 1 Creation Tools Description

Detection Method	Year	Description
FaceForensics++	2019	A large-scale dataset and benchmark for detecting manipulated facial content.
XceptionNet	2019	CNN-based method fine-tuned on deepfake datasets for high accuracy in fake image detection.
Capsule Networks	2020	Uses dynamic routing between capsules to identify subtle inconsistencies in facial textures and expressions.
Multi-task Learning (MTL)	2020	Simultaneously learns to detect deepfakes and predict the manipulation method used.
Spatial Artifacts Detection	2020	Identifies spatial irregularities in images, such as blending artifacts or mismatched textures.
FFT-based Detection	2020	Uses frequency-domain analysis to detect anomalies introduced by GANs.
Two-Stream Networks	2021	Combines spatial and temporal information for improved video-based deepfake detection.

FWA (Face Warping Artifacts)	2019	Detects artifacts caused by face warping in GAN-based deepfake videos.
DeepRhythm	2021	Uses biological signals like heartbeat rhythms visible in subtle skin color changes to identify fakes.
Siamese Neural Networks	2021	Compares similarity between frame sequences to detect fake facial movements.
Patch-based CNN	2021	Analyzes small patches of images independently to spot inconsistencies.
EYE Blink Detection	2018	Detects unusual eye blink patterns, which are often absent or unnatural in deepfakes.
Audio-Visual Inconsistency	2021	Compares facial movements with audio tracks for mismatches.
Vision Transformer (ViT)	2022	Leverages transformer-based architecture for pixel-level analysis of manipulated regions.
Video Transformer Networks	2022	Uses attention mechanisms to track inconsistencies across video frames.
Attention-based CNN	2022	Focuses on important facial regions during analysis to improve detection accuracy.
Lip Sync Error Detection	2022	Identifies misalignment between lip movements and spoken audio in videos.
Physics-based Detection	2023	Uses physical constraints like light reflection and shadow consistency for verification.

III. DEEPFAKE DETECTION

3.1 Image Forgery Detection

Detecting forged images plays a crucial role in maintaining the authenticity and reliability of digital media. With the rapid advancement of image editing software and deep learning–based image generation techniques, forged images have become increasingly realistic and difficult to distinguish from genuine content. Modern manipulation methods such as copy–move, splicing, object removal, and AI-assisted image synthesis introduce subtle artifacts that often go unnoticed by the human eye. As a result, traditional image forgery detection techniques based on handcrafted features and statistical analysis are no longer sufficient to address current forgery challenges. To overcome these limitations, deep learning–based approaches have emerged as powerful solutions for image forgery detection. Convolutional Neural Networks (CNNs) are capable of automatically learning hierarchical feature representations that capture inconsistencies in texture, color distribution, noise patterns, and structural composition introduced during image manipulation. These models eliminate the need for manual feature engineering and offer improved robustness across different forgery types and datasets.

In this project, we address the image forgery detection problem using deep learning techniques that effectively analyze visual artifacts caused by image manipulation. The proposed approach focuses on extracting discriminative features from images to differentiate between authentic and forged content. By leveraging CNN-based architectures, the system detects subtle anomalies related to lighting variations, boundary discontinuities, and texture irregularities that are difficult to identify through visual inspection. This section presents an overview of the progression from conventional image forgery detection methods to modern deep learning–based forensic techniques. It also describes the methodologies employed in our work to achieve accurate and reliable image forgery detection, highlighting their applicability in real-world digital forensics and multimedia authentication scenarios.

3.1.1 Datasets

For robust image forgery detection, datasets are generally divided into general image forensics datasets and deep learning-based forgery datasets, each serving to evaluate different detection scenarios.

3.1.2 Traditional Forensics Datasets

Traditional datasets, such as the Dresden Image Database and MICC series, focus on detecting classic image manipulations like splicing, copy-move, and inpainting. They are created in controlled settings with limited diversity, making them less suitable for evaluating modern AI-generated forgeries.

3.1.3 Deepfake Datasets

Modern Deepfake datasets are largely generated using GANs or DNN-based face-swapping techniques and include realistic face manipulations. Key examples include:

- Face Forensics++ (FF++): 1,000 real and 4,000 manipulated videos using four forgery methods.

- DFDC (Deepfake Detection Challenge): Over 100K videos from diverse actors with multiple manipulation methods.
- Celeb-DF and DFD: High-quality deepfakes with improved facial realism.
- DeeperForensics-1.0: 17.6 million frames from 60K videos, simulating real-world compression and perturbations.
- Wild Deepfake (WDF): Web-crawled dataset with 707 real-world deepfake videos.
- DFFD and DF-TIMIT: Datasets focusing on low and high-quality face swaps across subjects.

These datasets laid the groundwork for benchmarking detection algorithms, but most focus on single-face, video-based scenarios.

3.1.4 Open Forensics Dataset

For our project, we use the Open Forensics (OF) dataset [ICCV 2021] — a state-of-the-art benchmark for multi-face deepfake image detection and segmentation in the wild. Unlike prior datasets:

- It contains 115K high-resolution images and over 334K faces from varied scenes.
- Supports detection in multi-face, cluttered, occluded, and real-world conditions.
- Includes pixel-level forgery annotations and natural post-processing effects.

We use a curated subset of the dataset:
40,000 images are used for validation, 140,000 for training, and 20,000 for testing.This dataset directly supports our project goal of robust image-based deepfake detection by providing large-scale, complex, and diverse training data, enabling our ensemble model (Xception, ResNeXt50, EfficientNetB1) to generalize effectively.

3.2 Traditional Forensic-Based Techniques

Traditional forensic methods detect image manipulations using techniques such as copy-move (splicing), resampling (scaling, rotation), and object addition/removal. These methods are broadly divided into active and passive approaches.

- Active methods embed watermarks or signatures during image creation, allowing later verification of authenticity. However, they require prior embedding, which is often impractical for general image analysis.
- Passive methods rely on detecting statistical inconsistencies, such as noise patterns, compression artifacts, or lighting anomalies, without any prior embedding. These are widely used when source metadata or hardware protection is unavailable.

In addition, anti-spoofing techniques are crucial to prevent attacks using manipulated images, including deepfakes or hyper-realistic masks. Common approaches include eye blink detection, CNN-based classification, facial landmark analysis, and feature extraction.

3.3 CNN/DNN based Technique

Researchers' attention has shifted to sophisticated multimedia forensic techniques for efficient detection due to the growing threat posed by deepfakes and their capacity to convincingly alter visual content. These detection techniques typically rely on two main types of evidence: temporal artefacts, such as irregular motion patterns, physiological signal mismatches, and frame-level synchronization discrepancies, and spatial artefacts, such as irregular facial blending, unnatural textures, and distinctive GAN fingerprints. More robust Deepfake detection models have recently been developed using features.

YEAR	DATASET	ORIGINAL IMAGES	VIDEOS	FAKE IMAGES
2011	MICC-F220, MICC-F2000. MICC-F600	110, 1300, 440	/ / /	110, 700. 160
2013	IEEE IFS-TC	1050	/	450
2015	WWD [45]	13.5K	/	/
2015	CELEBA [46]	202K	/	/
2017	VISION [47]	34.4K	1914	
2018	UADFV [48]	17.3K	49	17.3K
2018	DF-TIMIT [49]	34.0K	320	68.0K
2018	FF [50]	500.0K	1004	521.4K
2019	FF++ [51]	509.9K	1,000	509.0K
2019	DFFD [28]	58.7K	1,000	240.3K
2019	DFD [52]	315.4K	363	2,242.7K
2019	DFDC-P [53]	488.4K	1,131	1,783.3K

2020	DFDC [54]	/	23K	/
2020	CELEB-DF [55]	225.4K	590	2,116.8K
2020	DF-1.0 [56]	12.6M	50,000	5.0M
2020	WDF [57]	11.8M	/	7,314
2021	OF [58]	16K	/	173K

Table 3 Publicly Available Forgeries Detection Datasets

3.3.1.1 Preprocessing Pipeline

To ensure consistency and model interoperability across the ensemble-based deepfake detection system, a robust preprocessing pipeline was employed on the dataset comprising facial images labeled as either Real or Fake.

The dataset consists of high-resolution facial images extracted from video frames sourced from publicly available deepfake detection datasets. Each image is explicitly labeled as:

- Real: Authentically recorded facial images.
- Fake: Synthetically generated or altered faces using deepfake techniques.

These images serve as input to the ensemble composed of three deep convolutional neural network (CNN) models: EfficientNetB1, ResNet50, and Xception.

3.3.1.1.1 IMAGE STANDARDIZATION

Since each model has different native input dimensions (e.g., 224×224 for ResNet50 and EfficientNet, 299×299 for Xception), all images were uniformly resized to 256×256 pixels. This intermediate resolution offers a trade-off between detail preservation and computational efficiency. Standardization of input shape is critical to:

- Ensure seamless parallel inference through the ensemble pipeline.
- Prevent feature misalignment due to mismatched input resolutions.
- Optimize memory utilization during batch training and evaluation.

3.3.1.1.2 Preprocessing Techniques

All image samples underwent the following preprocessing steps:

- Rescaling: Pixel values were normalized from the default 0–255 range to the 0–1 range using:

$$\text{Normalized Pixel} = \frac{\text{Pixel Value}}{255}$$

This normalization accelerates convergence and stabilizes training.

- Data Augmentation: To enhance dataset diversity and minimize overfitting, a random horizontal flip with a 50% probability was employed. This data augmentation technique allows the model to better generalize to variations in left-right facial orientation. Furthermore, various random geometric transformations were applied during training to further enrich the dataset.
- Resizing
- Rotation
- Reflection
- Shear
- Translation

These transformations were applied using an augmentedImageDatastore to ensure that during each training epoch, the model saw a different augmented version of the same image, improving generalization and robustness to variations.

Resizing Option	Data Format	Resizing function	Sample Code
Rescaling	3-D array representing a single color or multispectral image 3-D array representing a stack of grayscale images, 4-D array representing a stack of images	imresize	im = imresize(I,outputSize);
	4-D array representing a stack of images Image Datastore table	augmentedImageDatastore	auimds = augmentedImageDatastore(outputSize,I);
Cropping	3-D array representing a single color or multispectral image	imcrop (Image Processing	im = imcrop(I,rect);im = imcrop3(I,cuboid);

		Toolbox) and imcrop3	
	3-D array representing a stack of grayscale images, 4-D array representing a stack of color or multispectral images	augmentedImageDatastore	auimds = augmentedImageDatastore (outputSize,I,'OutputSizeMode', m);

Table 4 Resize Images Using Rescaling and Cropping

- Label Encoding: Binary class labels (Real or Fake) were transformed into one-hot encoded vectors:

Real → [1, 0]
Fake → [0, 1]
This encoding facilitates the use of categorical cross-entropy loss and supports multi-output neural architectures.

3.3.1.2 Model Architectures

The suggested ensemble model combines three sophisticated deep convolutional neural network (CNN) architectures for deepfake detection: Xception, ResNet50, and EfficientNetB1. Together, these models improve the system's overall performance by providing unique advantages in computational efficiency, feature extraction, and detection accuracy.

3.3.1.2.1 ExceptionNetB1

EfficientNetB1 is a member of the EfficientNet family, which employs a novel neural architecture search (NAS) strategy to optimize network depth, width, and resolution simultaneously. This multi-objective approach ensures that EfficientNetB1 achieves a high accuracy-to-parameter ratio, making it an efficient model that strikes a balance between accuracy and computational complexity. Specifically, the architecture utilizes a compound scaling method, allowing it to scale uniformly across all dimensions (depth, width, and resolution). This results in fewer parameters compared to traditional architectures while maintaining or even surpassing their performance. EfficientNetB1 is particularly well-suited for deployment in resource-constrained environments, such as mobile devices or edge computing platforms, where computational resources are limited.

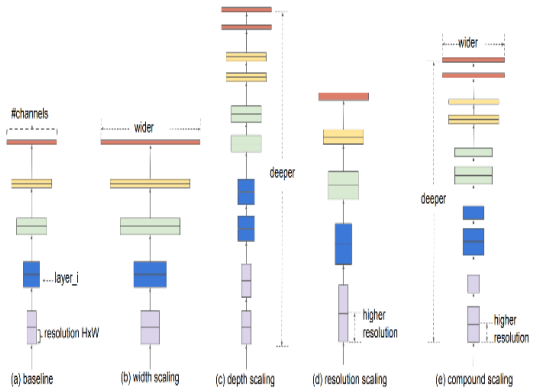


Figure 5 Efficient-Net Scaling [Source: EfficientNet: Rethinking Convolutional Neural Network Model Scaling]

3.3.1.2.2 ResNet50

In order to solve the issue of vanishing gradients in deep neural networks, ResNet50 presents the idea of residual learning by utilizing residual connections. The network can learn more intricate representations thanks to these residual connections without experiencing the degradation issue that deeper models usually have. ResNet50's primary innovation is its capacity to make training extremely deep networks easier by permitting gradient flows through identity mappings, thereby avoiding some of the network's non-linearities. ResNet50 is therefore very good at extracting hierarchical features from images, which allows it to identify both low-level and high-level patterns. For image classification tasks like deepfake detection, where identifying minute facial variations is crucial, its resilience in learning from deep layers makes it especially useful.

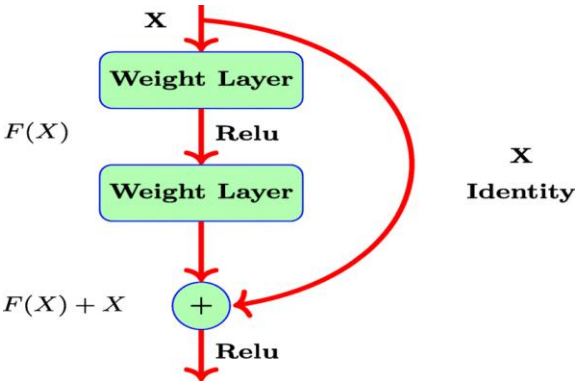


Figure 6: Residual Block [[Source: EfficientNet: Rethinking Convolutional Neural Network Model Scaling]]

3.3.1.2.3 Xception

Depth wise separable convolutions are used in Xception, an advancement of the Inception architecture, to improve feature extraction and drastically lower computational costs. This method separates spatial and channel-wise operations, allowing for more efficient processing than traditional convolution layers. With less computational work, Xception can now capture fine details in photos thanks to this design. Because of this, Xception performs exceptionally well in challenging image recognition tasks like deepfake detection, where it is essential to discern minute variations between real and fake faces.

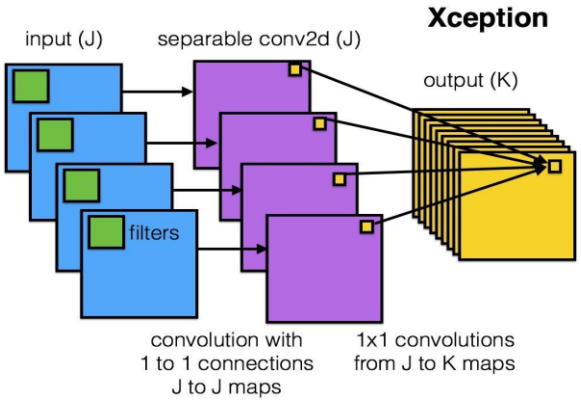


Figure 7 Separable Convolutions by Depth [Source: Xception — With Depth wise Sep. Convuliton - Linkedln]

3.3.1.2.4 Training Methodology – Ensembling

Ensemble learning combines predictions from several models to produce a final prediction. In the context of deepfake detection, assembling allows us to integrate predictions from the EfficientNetB1, ResNet50, and Xception models all of which were trained independently on the same dataset. The goal of using an ensemble model is to maximise each model's strengths while compensating for its flaws. The three models in the ensemble are able to recognize different patterns and nuances in the dataset due to their distinct architectures, which raises the detection accuracy overall.

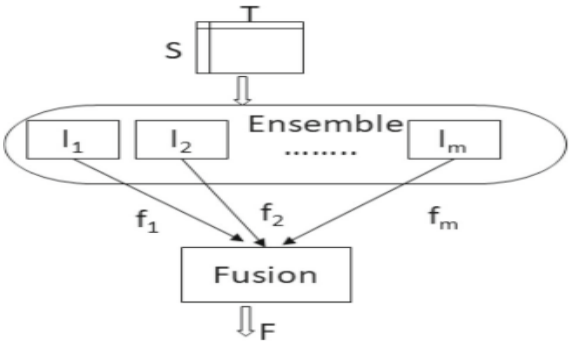


Figure 8 Ensemble Fusion diagram (illustrating the process of aggregating outputs from multiple models). [Source: Google Search]

Benefits of Using Ensemble Methods:

Enhanced Generalization: Merging several models typically leads to superior generalization compared to single models. This is vital in deepfake detection, where detecting subtle nuances between authentic and manipulated faces is essential.

Lower Risk of Overfitting: By combining outputs from multiple models, ensemble approaches lessen the chance of overfitting to specific data subsets.

Greater Stability: Ensembles are less affected by mistakes from individual models, resulting in more consistent and dependable performance

3.3.1.2.4.1 Soft Voting:

Rather than using a simple majority vote of the models predicted labels, soft voting is an ensemble approach in which the final decision is made by averaging the probability outputs from all models. The likelihood of each class (such as Real or Fake) is provided by each model, and the final prediction is calculated by averaging these probabilities:

$$P_{\text{class}_i}(x) = \frac{1}{n}\sum_{k=1}^{n} P_k^{\text{class}_i}(x)$$

The probability for class I given input x is computed mathematically for a classification task with C classes and n models as follows:

$$\hat{y} = \arg\max_i P_{\text{class}_i}(x)$$

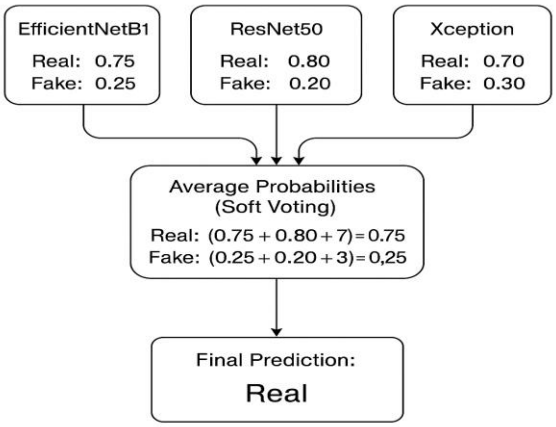
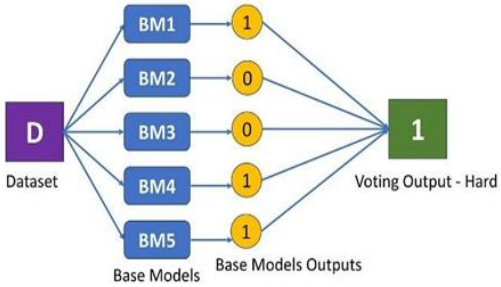


Figure 9 Soft Voting [Source: Medium]



3.3.1.3 Evaluation Ensemble

To find out how well the ensemble-based deepfake detection system distinguishes between real and manipulated facial images, it is imperative to evaluate it. It is crucial to use a comprehensive evaluation approach because deepfake detection is complicated and the distinctions between real and fake visuals are frequently very subtle. This aids in precisely determining the ensemble model's strengths and weaknesses.

The evaluation metrics, procedures, and experiments used to gauge the effectiveness of the ensemble made up of the ResNet50, Xception, and EfficientNetB1 models are described in this section.

Evaluation Metrics:

- Accuracy: Accuracy quantifies the percentage of true and false predictions that the model makes. When working with imbalanced dataset s, which is common in deepfake detection, accuracy may not always be enough, even though it offers a broad picture of model performance.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Real images that the model accurately recognises as real are known as True Positives (TP).

True Negative (TN): False pictures that the model accurately detects as such.

False Positives (FP) are phoney photos that the model incorrectly classifies as authentic.

False Negative (FN): Actual photos that the model mistakenly classifies as fraudulent.

- Area Under the ROC Curve (AUC-ROC): This measure assesses how well the model can differentiate between the real and fake classes. Plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold values is done by the Receiver Operating Characteristic (ROC) curve. The likelihood that the model will rank a randomly selected real image higher than a randomly selected fake image is represented by the AUC.

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR})$$

- Confusion matrix visually represents how well the ensemble model performs by showing the comparison between actual labels and the model's predictions. It breaks down the counts of true positives, true negatives, false positives, and false negatives, giving clear insight into classification accuracy and errors

IV. CHALLENGES FOR DEEPFAKE DETECTION

Even with significant improvements in DeepFake detection accuracy, a number of issues still need to be addressed. The efficacy of current detection techniques is hampered by a number of factors, such as the scarcity of diverse datasets, the emergence of unknown and emerging forms of media attacks, the difficulty of aggregating information over time, and the existence of substantial amounts of unlabelled data.

- Absence of DeepFake datasets: The size and diversity of the datasets used for training have a significant impact on how well a DeepFake detection model performs. It becomes difficult to create a model that can detect novel or invisible forms of manipulation when it is tested on media that doesn't contain examples of those manipulations. Furthermore, as web-based platforms have grown in popularity, DeepFake videos frequently go through postprocessing procedures like cropping, blurring, smoothing, and temporal artefact removal in an effort to trick detection systems.Unknown type of attack
- Another challenging task is creating a robust DeepFake detection model that can withstand unknown attack types, such as the fast gradient sign method (FGSM) [129] and the Carlini and Wagner L2 norm attack (CW-L2) [130]. These attacks deceive classifiers in their actual output. An example of a DeepFake creation using source and target faces with adversarial perturbations is presented in Figure 13. DeepFakes are correctly classified as fake by a DeepFake detector, but adversarially perturbed DeepFakes are classified as real.
- Temporal Aggregation: Current DeepFake detection algorithms use binary frame-level classification, which determines whether each video frame is authentic or fraudulent. However, because these methods do not take interframe temporal consistency into account, they may encounter issues such as displaying temporal abnormalities and real/artificial frames occurring in consecutive intervals. Furthermore, these approaches necessitate an extra step to calculate the video integrity score, which needs to be integrated for every frame in order to obtain the final result.
- Unlabeled data: DeepFake detection models are usually trained on large datasets. However, in some cases, such as journalism or law enforcement-based DeepFake detection, there may only be a limited dataset available. Additionally, labelling the score that corresponds to the type of forgery used in this type of dataset requires more effort. Thus, further investigation is required to understand instances of forgery in law enforcement or journalism. Most DeepFake detection models, particularly those based on deep learning techniques, lack this type of explanation because they are black-boxed. As a result, developing a DeepFake detection model using a small, unlabelled dataset is challenging.

V. CONCLUSION

Using the advantages of the EfficientNetB1, ResNet50, and Xception models, this paper concludes by presenting an ensemble framework based on deep learning for efficient DeepFake image detection. This ensemble greatly improves detection robustness and accuracy when compared to individual models. Even with these advancements, problems like domain adaptation and dataset diversity still exist. Our method contributes to a more dependable forensic system by introducing Grad-CAM for model interpretability, a soft voting mechanism, and a standardised preprocessing pipeline. Protecting the authenticity of digital media will require improving detection techniques and integrating multi-modal analysis as DeepFake generation becomes more complex.

REFERENCES

[1] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 39–57, doi: 10.1109/SP.2017.49.

[2] T. K. Moon, “The expectation-maximization algorithm,” *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, 1996, doi: 10.1109/79.543975.

[3] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-generated images are surprisingly easy to spot for now,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 8692–8701, doi: 10.1109/CVPR42600.2020.00872.

[4] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[5] G. Huang, Z. Liu, L. Van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf.*

Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[6] U. Scherhag, L. Debiasi, C. Rathgeb, C. Busch, and A. Uhl, "Detection of face morphing attacks based on PRNU analysis," *IEEE Trans. Biom. Behav. Identity Sci.*, vol. 1, no. 4, pp. 302–317, 2019, doi: 10.1109/TBIOM.2019.2942395.

[7] J. Galbally, S. Marcel, and J. Fierrez, "Biometric ant spoofing methods: A survey in face recognition," *IEEE Access*, vol. 2, pp. 1530–1552, 2014, doi: 10.1109/ACCESS.2014.2381273.

[8] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "Deepfake detection based on discrepancies between faces and their context," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6111–6121, 2022, doi: 10.1109/TPAMI.2021.3093446.

[9] S. Hu, Y. Li, and S. Lyu, "Exposing GAN-generated faces using inconsistent corneal specular highlights," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 2500–2504, doi: 10.1109/ICASSP39728.2021.9414582.

[10] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, 2018, pp. 67–74, doi: 10.1109/FG.2018.00020.

[11] S. Fernandes *et al.*, "Detecting deepfake videos using attribution-based confidence metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020, pp. 1250–1259, doi: 10.1109/CVPRW50498.2020.00162.

[12] L. Guarnera, O. Giudice, and S. Battiato, "Deepfake detection by analysing convolutional traces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020, pp. 2841–2850, doi: 10.1109/CVPRW50498.2020.00341.

[13] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, "Detecting deepfake videos from appearance and behavior," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, 2020, pp. 1–6, doi: 10.1109/WIFS49906.2020.9360904.

[14] S. Fernandes *et al.*, "Predicting heart rate variations of deepfake videos using neural ODE," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2019, pp. 1721–1729, doi: 10.1109/ICCVW.2019.00213.

[15] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, 2019, pp. 83–92, doi: 10.1109/WACVW.2019.00020.

[16] N. Yu, L. Davis, and M. Fritz, "Attributing fake images to GANs: Learning and analyzing GAN fingerprints," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 7555–7565, doi: 10.1109/ICCV.2019.00765.

[17] F. Marra, D. Gragnani Ello, L. Verdoliva, and G. Poggi, "Do GANs leave artificial fingerprints?" in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval (MIPR)*, 2019, pp. 506–511, doi: 10.1109/MIPR.2019.00103.

[18] S. McCloskey and M. Albright, "Detecting GAN-generated imagery using saturation cues," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2019, pp. 4584–4588, doi: 10.1109/ICIP.2019.8803661.

[19] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing AI-created fake videos by detecting eye blinking," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, 2018, pp. 1–7, doi: 10.1109/WIFS.2018.8630787.

[20] Y. Zhang, L. Zheng, and V. L. L. Thing, "Automated face swapping and its detection," in *Proc. IEEE Int. Conf. Signal Image Process. (ICSIP)*, 2017, pp. 15–19, doi: 10.1109/SIPROCESS.2017.8124497.

[21] S. Fung, X. Lu, C. Zhang, and C.-T. Li, "DeepfakeUCL: Deepfake detection via unsupervised contrastive learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9534089.

[22] H. Khalid and S. S. Woo, "OC-FakeDect: Classifying deepfakes using one-class variational autoencoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020, pp. 2794–2803, doi: 10.1109/CVPRW50498.2020.00336.

[23] Z. Liu, X. Qi, and P. H. S. Torr, "Global texture enhancement for fake face detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 8057–8066, doi: 10.1109/CVPR42600.2020.00808.

[24] X. Wu, Z. Xie, Y. Gao, and Y. Xiao, "SSTNet: Detecting manipulated faces through spatial, steganalysis and temporal features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 2952–2956, doi: 10.1109/ICASSP40776.2020.9053969.

[25] Gandhi and S. Jain, "Adversarial perturbations fool deepfake detectors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9207034.

[26] Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, 2012, pp. 1–7.

[27] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 2307–2311, doi: 10.1109/ICASSP.2019.8682602.

[28] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS)*, 2018, pp. 1–6, doi: 10.1109/AVSS.2018.8639163.

[29] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, 2018, pp. 1–7, doi: 10.1109/WIFS.2018.8630761.

[30] M. Zampoglou, S. Papadopoulos, and Y. Kompa Tsiaris, "Detecting image splicing in the wild (WEB)," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2015, pp. 1–6, doi: 10.1109/ICMEW.2015.7169839.

[31] T.-N. Le, H. H. Nguyen, J. Yamagishi, and I. Echizen, "Open Forensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 10097–10107, doi: 10.1109/ICCV48922.2021.00996.

[32] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "DeeperForensics-1.0: A large-scale dataset for real-world face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 2886–2895, doi: 10.1109/CVPR42600.2020.00296.

[33] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for deepfake forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 3204–3213, doi: 10.1109/CVPR42600.2020.00327.

[34] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch, "Face recognition systems under morphing attacks: A survey," *IEEE Access*, vol. 7, pp. 23012–23026, 2019, doi: 10.1109/ACCESS.2019.2899367.

[35] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632.

[36] Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1967–1974, 2019, doi: 10.1109/TNNLS.2018.2875194.