# CNN Ensemble Model for Real-Time Deepfake Image Detection

Akshay Jadhav[1], Unmesh Kakuste[2], Satyam Shinde[3], Om Mangate[4], Prof. Sonali Deshpande[5]
[1,2,3,4,5]*School of Computing, MIT Art, Design and Technology University Pune, India.*

*Abstract*— **The proliferation of the 'Deepfakes' is increasingly undermining media authenticity and digital security. This work offers an ensemble-based deepfake detection system using EfficientNetB1, ResNet50, and Xception models. Every architecture is fine-tuned on a labelled dataset of actual and fake facial images; then, using a soft voting technique, they are combined to enhance classification robustness. Our method shows the benefit of architectural variety by attaining greater accuracy and AUC over single models. Moreover, GradCAM visualisations are used to interpret predictions by localising facial areas affecting model decisions. The suggested approach shows good generalisation ability and provides a scalable and understandable solution for deepfake image detection in the real world.**

*Index Terms*—**Ensemble, EfficientNetB1, ResNet50, and Xception, Grad CAM, Media Forensics.**

## I. INTRODUCTION

Data forgery has become a significant issue in the age ruled by artificial intelligence and machine learning. So sophisticated have deepfakes—synthetically produced images and videos closely resembling actual ones—that the human eye often cannot tell them from genuine media. Although such technology can be used for creative and educational reasons, it also raises major concerns including false information, political manipulation, defamation, and cybercrime. Deepfakes most often target the face, which is altered mostly by two methods: identity swapping and expression transfer. While identity manipulation substitutes one person's face with another—usually a celebrity or public figure—to spread false information, expression-based manipulation changes facial gestures in real-time.

To address the issues associated with the increasing threat of deepfakes, Convolutional Neural Networks (CNNs) and sophisticated preprocessing methods have been utilized. These methods include the extraction and alignment of facial features from video sequences where faces are tracked in the frames and pretrained CNNs (VGG16 or OxfordNet) are used to detect patterns differentiating real and fake sequences.

In this paper, we present a novel framework for detecting deep fakes which is based on ensemble face detection using EfficientNetB1, ResNet50, and Xception. Our model is trained on a comprehensive dataset of authentic and manipulated images, achieving over 90% accuracy in face recognition and deepfake detection which is reliable and easily adaptable over the years to come.

## II. DEEPFAKE

The primary tools for producing deepfake content are deep neural networks, particularly Generative Adversarial Networks (GANs) and autoencoders. These cutting-edge machine learning techniques allow for the creation of incredibly lifelike images and videos that, to the untrained eye, are frequently indistinguishable from authentic media.

### 2.1 Autoencoders for Face Swapping

The traditional approach to deepfake creation involves autoencoders, which are neural network architectures trained to learn efficient codings of input data. As described by Juefei-Xu et al. (2022), an autoencoder consists of two components:

1. Encoder: Compresses input images or videos into a latent vector, retaining critical features while eliminating noise.
2. Decoder: Tries to reproduce the original media by reconstructing the input from this compressed latent space.

A dual autoencoder model is frequently used in deepfake pipelines (Fig. 1.1), in which two distinct encoder-decoder pairs are trained: one for the source face and one for the target face. Using both real and fake data, the encoder learns to represent facial features into a common latent space. The target face is then successfully superimposed onto the source's facial expressions and head movements by the decoder, which creates synthetic outputs from this representation (Nguyen et al., 2019).

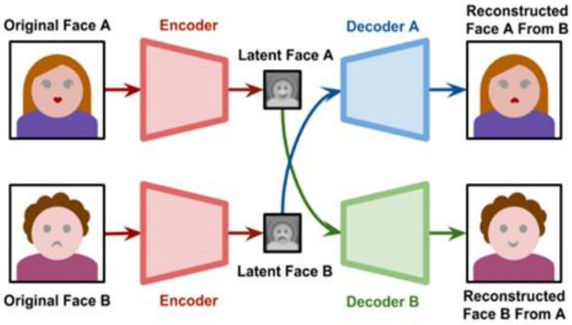


Figure 1 Deepfake model: Dual pair autoencoders

This process is further refined using powerful backbone architectures such as 3D ResNeXt and 3D ResNet, which capture spatiotemporal dynamics in video sequences for enhanced realism (Alanazi & Asif, 2023).

2.2 GAN-Based Content Synthesis

Generative Adversarial Networks (GANs) are a sophisticated class of deep learning models that are intended to produce realistic synthetic samples and replicate intricate data patterns. The Generator (G) and the Discriminator (D), the two primary parts of a GAN, compete with one another during the training process. The discriminator's job is to distinguish between generated and real samples, whereas the generator's job is to produce artificial data samples, like images.

While the discriminator learns to correctly classify the inputs as real or fake, the generator aims to accurately replicate real data distributions. A minimax optimisation problem is the mathematical representation of this adversarial training:

$$G^* = \arg\min_{G}\max_{D} V(G,D) = \arg\min_{G}\max_{D} \left[\mathbb{E}_{x\sim P_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim P_z(z)}[\log(1 - D(G(z)))]\right]$$

In this context, z~Pz(z)z \sim P_z(z)z~Pz(z) represents the random noise input fed into the generator. The discriminator, denoted as D(x)D(x)D(x), predicts the likelihood that the sample xxx comes from the real data distribution PdataP_{\text{data}}Pdata, as opposed to being generated by the model's distribution PgP_gPg.

Over successive iterations, G improves its synthetic outputs to deceive D, while D becomes more proficient at identifying fakes. This adversarial learning framework results in highly realistic data generation, which lies at the heart of modern deepfake creation.

The versatility and power of GANs have given rise to various architectural improvements and training methodologies, such as:

- DCGAN: Introduced convolutional layers for stability in image generation.
- WGAN: Employed Wasserstein loss to improve convergence.
- PGGAN: Used progressive growing for high-resolution image synthesis.
- BigGAN: Focused on class-conditional high-fidelity samples.
- StyleGAN & StyleGAN2: Enabled fine-grained control of facial attributes and styles, widely used in facial deepfake synthesis.

hese advanced GAN variants are crucial for generating photorealistic fake videos and images, where facial expressions, head poses, and even micro-expressions can be manipulated with remarkable precision (Malik et al., 2022).

Despite their strength, GANs need to be trained carefully and with large datasets to prevent issues like gradient instability, mode collapse, and overfitting. Almars (2021) points out that GANs may have trouble with small datasets, so architecture tuning and data augmentation are crucial. Fig. 1.2 shows the adversarial feedback loop between G and D during training, illustrating the fundamental architecture of a GAN model used in deepfake pipelines.

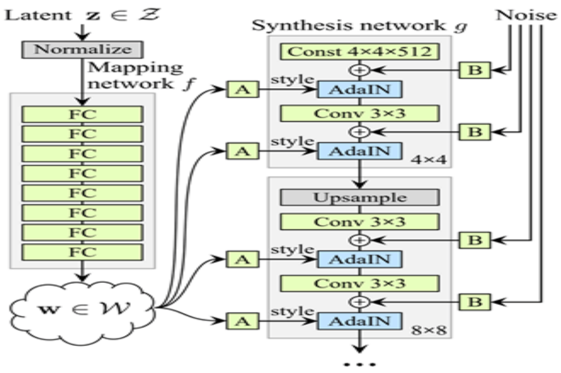


Figure 2 GAN architecture [Source: netpune.ai]

Deepfake technology uses sophisticated neural networks to create incredibly lifelike fake media. There are two levels of analysis for network architecture attribution: architectural and model-specific. Two main approaches are investigated: an AttNet-based approach and one based on learnt features. Especially when comparing fake and real images from the same model and dataset, AttNet successfully separates distinguishing features from GAN-generated images. However, when subjected to novel or altered training circumstances, its accuracy declines. In contrast, studies by support the suggested method's strong discriminative capability across a variety of scenarios, with differences in feature extraction visualised through t-SNE analysis.

## 2.3 Creation of Deepfake Tools

Deep learning advancements have significantly impacted the fields of robotics and computer vision. The integration of Generative Adversarial Networks (GANs) has further enhanced the capabilities of digital face image synthesis and video manipulation, producing highly realistic results. However, achieving disentangled and controllable synthesis within high-resolution contexts remains a challenge. Disentanglement allows for independent control of different features, but implementing this effectively in GANs often requires modifications such as regularization techniques. Table 1 outlines the tools commonly used for generating deepfake images and videos. Mobile applications like ZAO, Auto FaceSwap, and FaceApp have made it accessible for everyday users to create fake images and videos, contributing to the spread of deepfakes. Platforms like YouTube are filled with spoof videos produced through GAN-based face-swapping technologies, highlighting the ease of transferring a face from one image onto another seamlessly.

Face-swapping, face synthesis, face reenactment, and attribute manipulation are rapidly gaining traction due to their realistic outputs. Notably, models like StyleGAN2 and StyleGAN2-Ada have demonstrated remarkable realism, almost indistinguishable from real images. Despite these advancements, certain limitations still exist—StyleGAN, for example, struggles to adjust specific facial attributes like skin tone or eye size without affecting other facial features. Similarly, BigGAN faces challenges in modifying color attributes without altering the overall image structure.

| Tool Name | Description |
|---|---|
| Face App | Mobile app for altering facial features and aging effects. |
| ZAO | Chinese mobile app for face swapping in videos. |
| Deep Face Lab | Open-source tool for creating deepfake videos and images. |
| Face swap | Open-source platform with deep learning for swapping faces in images/videos. |
| First Order Motion Model | Deep learning model for animating still images using motion templates. |
| Reface | Mobile app that lets users swap faces in GIFs and short video clips. |
| Avatarify | Real-time face reenactment for video calls. |
| StyleGAN | GAN-based model for high-resolution face generation and manipulation. |
| Deep Art Effects | AI-powered platform for artistic deepfake transformation. |
| Deep Nude | Controversial app that used GANs to create fake nude images. |
| 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. |
| Fake App | One of the earliest tools for creating deepfake videos through face swapping. |

Table 1 Creation Tools Description

| Detection Method | Year | Description |
|---|---|---|
| Face Forensics++ | 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. |
| Deep Rhythm | 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

A key element in preserving the authenticity and integrity of multimedia material is finding fake facial images. The development of generative models such as autoencoders and GANs has greatly increased this difficulty by allowing the generation of very realistic, yet false, photographs. Our work emphasises finding altered facial photos produced by such models. Against today's deep learning-based fabrications, traditional image forgery detection methods relying on handcrafted features and statistical cues are no longer adequate.

We tackle this difficulty by means of an ensemble of deep learning architectures—including Xception, EfficientNetB1, and ResNeXt50—to efficiently extract and examine minute anomalies produced during deepfake creation. Our approach uses convolutional neural networks (CNNs) to identify texture, lighting, and structural pattern discrepancies that are challenging to perceive with the human eye.This section outlines the evolution of forgery detection from conventional methods to modern deepfake image forensics, and presents the techniques we apply for accurate and reliable deepfake detection.

### 3.1 Datasets

For effective deepfake detection in realistic settings, datasets can be broadly categorized into traditional forensics datasets 1and DeepFake-specific datasets.

### 3.1.1 Traditional Forensics Datasets

Traditional datasets like the Dresden Image Database (DID) and MICC series (F220, F2000, F600) were designed for image forgery detection tasks such as splicing, inpainting, and copy-move forensics. These datasets were built in controlled environments with limited diversity and primarily focus on image alteration rather than AI-generated forgeries. While important historically, their limitations in terms of scale, realism, and manipulation variety make them less suited for modern deepfake detection.

### 3.1.2 DeepFake Datasets

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

- FaceForensics++ (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 Deep fake (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.

### 3.1.3 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 image processing methods like copy-move (splicing), resampling (resize, rotate), and object addition/removal are frequently used to identify manipulated images. Active and passive methods are the two main categories into which forensic techniques for image tampering detection fall.

During the creation process, active methods incorporate digital watermarks or signatures into images. To confirm authenticity and spot areas that have been tampered with, these can subsequently be extracted. These techniques, however, necessitate prior embedding, which is frequently impractical for general image analysis.

Conversely, passive approaches don't need data that has already been embedded. To find evidence of manipulation, they use statistical inconsistencies in the image, such as noise patterns, lighting imbalances, or compression artifacts. When hardware-based protection or source metadata are not available, these techniques are frequently employed.

Anti-spoofing techniques are essential for thwarting biometric attacks like image-based impersonation and hyperrealistic masks in face image detection. These include eye blink detection, CNN-based classification, facial landmark analysis, and feature extraction.There are 334K human faces in 115K unrestricted photos in the OF dataset.These current datasets are summarized in Table 3.

| 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 2 Publicly Available Forgeries Detection Datasets

### 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 synchronisation 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.

#### 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:
  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<br>3-D array representing a stack of grayscale images<br>4-D array representing a stack of images | imresize | im = imresize(I,outputSize); |
| | 4-D array representing a stack of images<br>ImageDatastore table | augmentedImageDatastore | auimds = augmentedImageDatastore(outputSize,I); |
| Cropping | 3-D array representing a single color or multispectral image | imcrop (Image Processing Toolbox) and imcrop3 | im = imcrop(I,rect);<br>im = imcrop3(I,cuboid); |
| | 3-D array representing a stack of grayscale images<br>4-D array representing a stack of color or multispectral images | augmentedImageDatastore | auimds = augmentedImageDatastore(outputSize,I,'OutputSizeMode',m); |

Table 3 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.

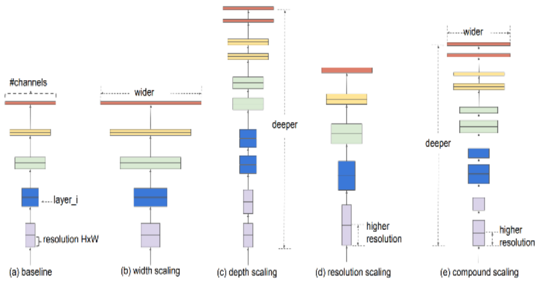


Figure 3 Efficient-Net *Scaling [*Source: Efficient Net: Rethinking Convolutional Neural Network Model *Scaling]*

### 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.

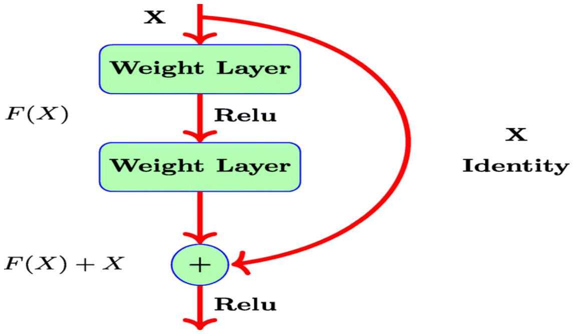


Figure 4 : Residual *Block [*[Source: EfficientNet: Rethinking Convolutional Neural Network Model *Scaling]*]

3.3.1.2.3 XCEPTION

Depthwise 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.

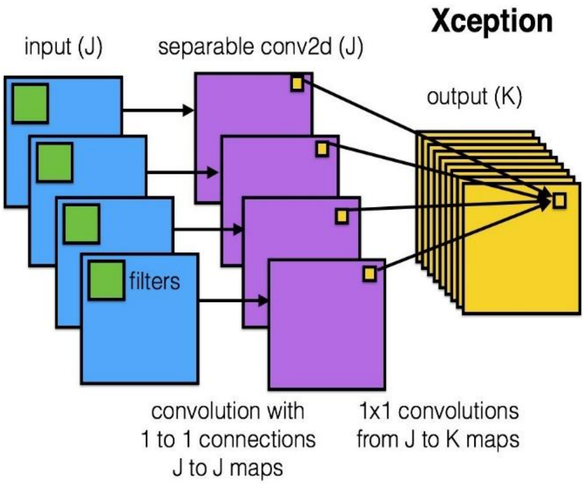


Figure 5 Separable Convolutions by Depth [Source: Xception — With Depthwise Sep. Convuliton - Linkedln]

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 recognise different patterns and nuances in the dataset due to their distinct architectures, which raises the detection accuracy overall.

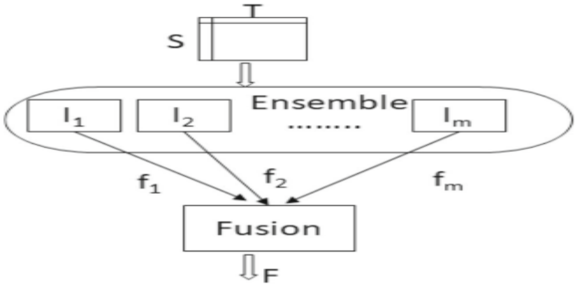


Figure 6 Ensemble Fusion diagram (illustrating the process of aggregating outputs from multiple models) [Source: Google Search]

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

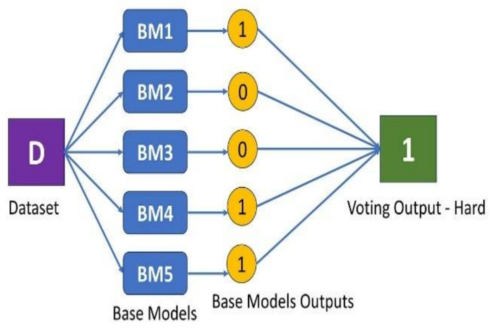
3.3.1.2.4.1 Soft Voting:

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:

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



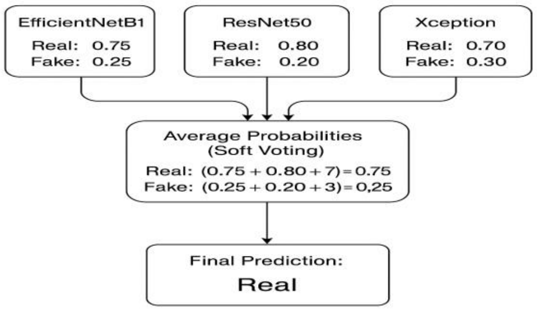


Figure 7 Soft Voting [Source: Medium]

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 DEEP FAKE 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 2017 IEEE Symposium on Security and 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8692–8701. doi: 10.1109/CVPR42600.2020.00872.

[4] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (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 2017 IEEE Conference on Computer Vision and Pattern Recognition (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 Antispoofing 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 ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (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 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 2018, pp. 67–74. doi: 10.1109/FG.2018.00020.

[11] S. Fernandes et al., “Detecting Deepfake Videos using Attribution-Based Confidence Metric,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1250–1259. doi: 10.1109/CVPRW50498.2020.00162.

[12] L. Guarnera, O. Giudice, and S. Battiato, “DeepFake Detection by Analyzing Convolutional Traces,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition 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 Deep-Fake Videos from Appearance and Behavior,” in 2020 IEEE International Workshop on Information Forensics and 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 2019 IEEE/CVF International Conference on Computer Vision Workshop (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 2019 IEEE Winter Applications of Computer Vision 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 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 7555–7565. doi: 10.1109/ICCV.2019.00765.

[17] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, “Do GANs Leave Artificial Fingerprints?” in 2019 IEEE Conference on Multimedia Information Processing and 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 2019 IEEE International Conference on Image Processing (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 2018 IEEE International Workshop on Information Forensics and 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 2017 IEEE 2nd International Conference on Signal and Image Processing (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 2021 International Joint Conference on Neural Networks (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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (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 ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2952–2956. doi: 10.1109/ICASSP40776.2020.9053969.

[25] A. Gandhi and S. Jain, “Adversarial Perturbations Fool Deepfake Detectors,” in 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207034.

[26] I. Chingovska, A. Anjos, and S. Marcel, “On the effectiveness of local binary patterns in face anti-spoofing,” in 2012 BIOSIG - Proceedings of the International Conference of 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 ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (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 2018 15th IEEE International Conference on Advanced Video and 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 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7. doi: 10.1109/WIFS.2018.8630761.

[30] M. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, “Detecting image splicing in the wild (WEB),” in 2015 IEEE International Conference on 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, “OpenForensics: Large-Scale Challenging Dataset for Multi-Face Forgery Detection and Segmentation In-The-Wild,” in 2021 IEEE/CVF International Conference on Computer Vision (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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (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 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5967–5976. doi: 10.1109/CVPR.2017.632.

[36] A. 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.