CommIT Journal 17(2), 151–168, 2023

Classification of Deepfake Images Using a Novel Explanatory Hybrid Model

Sudarshana Kerenalli[1*], Vamsidhar Yendapalli[2], and Mylarareddy Chinnaiah[3]
[1−3]Department of Computer Science and Engineering, School of Technology, GITAM University
Bengaluru 561205, India
Email: [1]ksudarsh@gitam.in, [2]ydhar@gitam.edu, [3]mylarareddyc@gmail.com

***Abstract*—In court, criminal investigations and identity management tools, like check-in and payment logins, face videos, and photos, are used as evidence more frequently. Although deeply falsified information may be found using deep learning classifiers, block-box decision-making makes forensic investigation in criminal trials more challenging. Therefore, the research suggests a three-step classification technique to classify the deceptive deepfake image content. The research examines the visual assessments of an EfficientNet and Shifted Window Transformer (SWinT) hybrid model based on Convolutional Neural Network (CNN) and Transformer architectures. The classifier generality is improved in the first stage using a different augmentation. Then, the hybrid model is developed in the second step by combining the EfficientNet and Shifted Window Transformer architectures. Next, the GradCAM approach for assessing human understanding demonstrates deepfake visual interpretation. In 14,204 images for the validation set, there are 7,096 fake photos and 7,108 real images. In contrast to focusing only on a few discrete face parts, the research shows that the entire deepfake image should be investigated. On a custom dataset of real, Generative Adversarial Networks (GAN)-generated, and human-altered web photos, the proposed method achieves an accuracy of 98.45%, a recall of 99.12%, and a loss of 0.11125. The proposed method successfully distinguishes between real and manipulated images. Moreover, the presented approach can assist investigators in clarifying the composition of the artificially produced material.**

***Index Terms*—Image Classification, Deepfake Images, Explanatory Artificial Intelligence, Hybrid Model**

I. Introduction

FACE images are typically used for face-based payment, face retrieval, face check-in, and other recognition and authentication services in daily life. However, artificial intelligence algorithms have made it simpler to create phony photographs and share incorrect information on social media. Modern data-driven tools have made it easier to create visuals from scratch. For example, Reddit user "deepfakes" employs deep learning algorithms to distribute female face photos into pornographic videos. There is a significant uproar in response to this instance of defamation and disrespect [1]. Moreover, BuzzFeed has ever produced a video with former US President Barack Obama that he gives a speech about the problems of deepfakes on individuals and society as a whole [2].

Artificial Intelligence-based face identification techniques perform noticeably better than human classification methods in identifying the Generative Adversarial Network (GAN) produced images [3]. Most early GAN-generated face image recognition systems depend on deep learning as their primary methodology [4]. Even though deep learning has an excellent track record, it may be difficult to describe how these learning techniques result in inferences and interpret them for forensic investigations [5]. The deep Convolutional Neural Network (CNN) algorithms are used in applications such as Snapchat, Instagram, Facebook, and Reddit to learn and apply visual hints to another picture or video [6].

Deepfake advancements have raised attention to the possible consequences of tampering with artificial faces. Due to the rising priority placed on identifying deep fakes and reducing risks by academic and industry professionals, several spoofing detection methods have been devised. The Deepfake face recognition techniques rely on deep learning to collect signal-level traits and train Deep Neural Network (DNN) classifiers to distinguish between fake and real faces [6]. For example, a forensic face recognizer is built on VGG-Net [7]. In another approach, the Incremental Classifier and Representation Learning (iCaRL) method keeps a small amount of extra information in compact memory to adapt to new image classes without losing the previous knowledge [8]. The residual field signals are important information to classify the data between real and GAN-produced faces [9]. Also, cross-modal, cross-data, and post-processing evaluations are done to detect fake images in realistic settings [10].

Received: July 21, 2022; received in revised form: Nov. 03, 2022; accepted: Nov. 04, 2022; available online: Sep. 06, 2023.
*Corresponding Author

Some researchers have employed the irregularity of the corneal specular highlights between the simulated eyes to identify faces through GAN [11].

As a result, constructing an efficient GAN-face recognition system remains challenging and complex due to issues surrounding its adaptability and decision interpretability. Rather than just adhering to specific datasets and employing complex deep networks, the GAN-face detector needs to be robust, adaptive, and capable of providing a straightforward assessment mechanism for human users, particularly non-artificial intelligence users. Moreover, it is challenging to explain how decisions are made and understood despite having a successful, proven record.

The research proposes a three-phase deepfake analysis approach to efficiently interpret and classify deepfake face images. Firstly, various augmentations are used to improve classifier generality. Then a hybrid EfficientNet and Shifted Window Transformer model is constructed for fake image classification by integrating the CNN and Vision Transformers (ViT). The fake images are the instances of class 1, and the real images are the instances of class 0. Next, the GradCAM technique is used to evaluate the classification model's output and provide a visual explanation of why this image is deepfake for human comprehension. In brief, the proposed approach makes the following contributions to solving a deepfake image detection problem as follows:

1) A three-stage deep learning system is designed to detect fake images produced using human or artificial intelligence techniques. This model combines a CNN with ViT.
2) Computational Intelligence and Photography Lab (CIPL) data sets are combined with a dataset of 140,000 real and synthetic faces to create the training and validation data. After that, this specialized data set is used to train and evaluate the developed model.
3) The research compares several classification metrics to evaluate the domain-neutral augmentations and regularization methods on the proposed framework against the state-of-the-art methods.

II. Literature Review

Table I gives the generation of the deepfake datasets. It describes the dataset name, generation class, unique fake content with or without the user's consent, and the number of perturbations applied.

A. Fake-Image Generation and Deepfake Datasets

Digital face manipulation technologies have advanced rapidly and presented challenging investigative problems. Autoencoders (AE) [1] and GAN [2] are two techniques for making deepfakes. Applications like FakeApp, DeepFaceLab, DFaker, and DeepFake-tf have all adopted autoencoder technology. However, the images generated from such applications are blurry and easily recognizable. Later, 1024×1024 pixels resolution images can be generated using ProGAN [12]. Flow-based generative models make an image with a resolution of up to 2048×1024 pixels [13]. Thus, primarily deepfakes can be classified into four types: synthesis, retouching, reenactment, and replacement. The deepfake datasets are categorized into three generations.

TABLE I
Public Deepfake Datasets.

Dataset	Generation	Unique Fake Dataset	Right	Number of methods
UADFV	1	49	None	1
DFTIMIT	1	640	None	2
FF++	1	4,000	None	4
GDFD	2	3,000	Yes	5
140kRFD+	3	70,000	Yes	- - -
DF-1.0	3	1,000	- -	1
DFDC	3	104,500	Yes	8

Note: Deepfake TIMIT (DFTIMIT), FaceForensics (FF), Google Deepfake Dataset (GDFD), DeeperForensics-1.0 (DF-1.0), Deepfake Detection Challenge (DFDC), and 140K Real and Fake Faces Dataset (140kRFD).

1) UADFV Dataset: There are 98 videos in total in the UADFV dataset [14]. About 49 videos are authentic, and the other 49 videos are fraudulent. In a desktop environment, the dataset is manageable and compact. The 98 videos are 147 MB each, and after being converted to frames, they make up 1.33 GB with 669 MB of fraudulent content and 701 MB of real content.

2) Deepfake TIMIT (DFTIMIT) Dataset: Faces have been altered in the video database known as DFIMIT using an open-source GAN-based approach adapted from the original autoencoder-based deepfake algorithm [15]. The database is created by carefully selecting 16 pairs of visually comparable individuals from the VidTIMIT database, which is available to the general public. Every 32 subjects receive training on a model with Lower Quality (LQ) with a 64×64 input/output size and a model of Higher Quality (HQ) with a 128×128 size. Given that each person has 10 movies in the VidTIMIT database, there are 320 generated videos for each version, totaling 620 videos with the faces altered. However, the audio channel is left unchanged.

3) The FaceForensics (FF) Dataset: The FaceForensics dataset is massive and in raw data form [16]. Video data on the c23 has a total of 30.9 GB and

is generally good quality. FaceForensics++ is developed to standardize face modification detection and evaluation. This benchmark dataset is built on the well-known face alteration techniques Deep-Fakes, Face2Face, FaceSwap, and NeuralTextures. The benchmark is public and contains a database of over 1.8 million modified images as well as a hidden test set.

4) Google Deepfake Dataset (GDFD): Google published a massive collection of visual deepfakes in conjunction with Jigsaw [17]. It is integrated into the new FaceForensics benchmark developed by the Technical University of Munich and the University Federico II of Naples. This collection contains video recordings of paid and consenting performers. Hundreds of deepfakes are made from these films using publicly accessible deepfake creation methods. This dataset is available to the scientific community without charge as a component of the FaceForensics benchmark.

5) DeeperForensics-1.0 (DF-1.0) Dataset: The largest face forgery detection dataset is DeeperForensics-1.0 [18]. It is 10 times larger than current datasets and consists of 60,000 videos with a total of 17.6 million frames. The total dataset contains 48,475 original films and 11,000 modified videos. About 100 paid and consenting actors are featured in the source recordings. They are from 26 different nations. The modified films are produced using DeepFake-Variational Auto Encoder, a recently suggested many-to-many and end-to-end face-swapping technique. The seven types of real-world perturbations at five intensity levels are used to achieve a bigger scale and more diversity.

6) The Deepfake Detection Challenge (DFDC) Dataset: The DFCD is used to measure progress in deepfake detection methods. In 2019, over 100,000 videos were created and released in the DFDC dataset [17]. The DFDC has made it feasible for experts from all around the world to work together, benchmark their deepfake detection models, test out new concepts, and learn from each research. The DFDC dataset comes in two different variations. First, the preview dataset consists of 5,000 videos with two facial modification algorithms. Second, the full dataset has 124,000 videos with eight facial modification algorithms. The whole dataset is used by competitors in a Kaggle competition to create novel algorithms for detecting altered content. Facebook produces this dataset using paid actors who give their consent.

7) The 140K Real and Fake Faces Dataset (140kRFD): The 140kRFD dataset includes a large number of high-quality face images of people with various genders, ages, and real-world fake faces. This dataset contains 70,000 real faces from the Flickr dataset gathered by Nvidia and 70,000 from the 1 million fake faces. They are produced using StyleGAN and contributed by Bojan. All the images are resized to 256 pixels, and the data are divided into the train, validation, and test sets [19].

TABLE II
INPUT IMAGE DATASET.

Dataset	Real	Fake	Total
140k_Real_Fake_Faces	70,000	70,000	140,000
CIPL Dataset s	1,081	960	2,041
Total	71,081	70,960	142,041

8) Computational Intelligence and Photography Lab (CIPL) Dataset: High-quality and professionally edited facial images produced are included in the CIPL dataset. The photos are composites of several faces, divided by the eyes, mouth, nose, or entire face. The reason why people need these expensive photographs in addition to images created automatically by computers may be a question. It can be a situation where one wants to train a classifier to differentiate between real and fake face photos [20].

Using generative models like GAN, producing fake facial images is simple and clear. Once trained using such images, a classifier can reliably differentiate between real and fake face images. It is logical to assume that the classifier finds some pattern among the GAN-generated pictures. However, such patterns may not be effective in front of human professionals since expert counterfeits are made through different techniques. Thus, in the research, the researchers combine both kinds of images with training the model and making it a general one. Table II shows the input image dataset.

B. Fake-Image Detection Methods

The following are the current works on detecting deepfake images. Deepfake face recognition approaches based on deep learning collect signal-level information and train Deep Neural Network (DNN) classifiers to distinguish between fake and real faces. Previous research presents a VGGNet-based forensic face recognizer [21]. It improves the model to distinguish between real and fake images. To create diverse synthetic faces at various sizes and resolutions, they build Progressive Growing-Generative Adversarial Network (PGGAN) and Deep Convolutional Generative Adversarial Network (DCGAN). Then, tests are run utilizing the validation data from the AI Challenge to ensure improved results. Creating flexible training data to meet the test data set for the AI Challenge competition is one of the important contributions to the previous research. A deep learning face recognition

network extracts the face attributes and detects real or fake faces with 80% accuracy.

When face detection and editing algorithms are closely examined, it is found that many of their abnormalities closely resemble classical computer vision issues. It looks at many facial editing methods that are now available, as well as numerous distinctive processing issues. Since the approaches focus on visual qualities, they are easy to comprehend even for non-technical experts. The techniques are easy to apply and enable rapid adaption to new manipulation patterns with little data. The Receiver Operating Characteristic Curve (ROC-AUC) for the approach is 0.866 [21].

Another research shows that residual field signals are crucial criteria for discriminating between real and GAN-produced phony faces [22]. Following high-pass filter analysis of the input faces, the residuals are fed into deep networks for GAN-face recognition. It presents an augmented Xception model for identifying faces produced locally using GAN. Some improvements over Xception are as follows: (1) four leftover blocks are removed to reduce overfitting, (2) the convolution layer in the Xception's pre-processing module is replaced with the Inception block with dilated convolution, (3) for the final choice, a feature pyramid network is used to get multi-level features. It develops the dataset for Locally GAN-based Generated Faces (LGGF) using the pluralistic image completion method and the FFHQ dataset as a reference. It comprises 952,000 images in various sizes and shapes for the regions that have been produced. According to experimental results, the proposed model performs better for faces with small generated areas.

The previous researchers classify GAN-produced images using the iCaRL method [8]. By keeping a small amount of extra information in compact memory, iCaRL adapts to new image classes without losing the previous knowledge. They investigate a dual-task challenge, including GAN-image recognition and classification. They also introduce a new binary loss component to the existing classification loss. They manage to achieve an accuracy of about 89% and utilize 2,400 StarGAN-generated fake images.

Another research offers a framework for evaluating detection algorithms in realistic settings, which includes cross-modal, cross-data, and post-processing evaluations [23]. The suggested framework is then applied to assess cutting-edge detection algorithms. Researchers also investigate the effectiveness of common image pre-processing approaches. Finally, the authors used an online poll to measure human performance and the variables influencing detection performance. Their findings suggest that CNN-based detection algorithms are not yet trustworthy enough to be used in real-world scenarios.

Previous researchers also propose an end-to-end network that is efficient and responsive. The model can identify GAN-generated faces by examining eye inconsistencies to address issues related to the limited number of publicly available datasets. It is because the existing image databases do not adequately reflect real-world scenarios regarding perspective variations and data distributions. Furthermore, modern systems are incapable of evaluating detection data and do not generalize well to real-world issues. Guo's model learns to distinguish abnormal corneal features by automatically localizing and comparing iris artifacts between the two eyes. Deep networks address the imbalance learning difficulties by taking into account the AUC loss and the standard cross-entropy loss [24].

Another previous research employs semantically significant characteristics to recognize faces using a GAN. The eyes, nose, skin, and mouth are facial landmark sites that display abnormalities in GAN-synthesized faces [25]. The GAN-based face synthesis approach functions similarly to players in a game of Fukuwarai. It possesses all the components of a face but falls short of arranging them naturally and cohesively, much as in a real face. There are various anomalies in the facial structures' arrangement. The positions of the facial landmark points are automatically recognized in face pictures. GAN-synthesized faces are displayed using these facial landmarks. It employs a basic SVM classifier with features that are the normalized positions of these face landmarks. On face images created with PGGAN, the landmark location-based Support Vector Machine(SVM) classifier is evaluated for classification using simple components.

Similarly, another research employs the irregularity of the corneal specular highlights between the simulated eyes to identify faces through GAN. When the eyes are oriented straight at the camera, they receive the same view of an item as when the light sources or reflections in the environment are relatively far away from the subject. Furthermore, correlations may be seen in the corresponding corneal specular highlight. GAN-generated faces, on the other hand, are compatible with the portrait configuration. As a result, there are differences in the corneal specular highlights of the eyes. This approach aligns and compares corneal specular highlights from separated eyes. The findings reveal that the similarity score distributions of genuine and GAN-generated faces differ substantially. They can be used to quantitatively identify them from one another [26].

Previous research develops a deep CNN to detect forensic faces with good performance on AI Challenge validation data using GANs for data augmentation

TABLE III
RECENT APPROACHES FOR DEEPFAKE IMAGE CLASSIFICATION.

Article	Detection Method	Image Generation Algorithm	Result (Accuracy in %)
[8]	Incremental Classifier	StarGAN	>81.5
[21]	VGGNet	DCGAN, ProGAN	ACC: 80
[22]	Xception	PGGAN	71.3 to 97.7
[23]	ForensicTransfer	StyleGAN ProGAN	1 to 100
[24]	Residual Attention	StyleGAN2	AUC: 1
[25]	Landmark Locations	PGGAN	AUC: 0.9413
[26]	Gaze Tracking	StyGAN	Acc. 80 to 88.5
[27]	CNN	PGGAN, DCGAN	80
[28]	CNN	StarGAN	99
[29]	Out-of-Context Object Detection	StyleGAN	80
[30]	Deep Neural Net	StyleGAN2	>88
[31]	Color Components Disparities	StyGAN, ProGAN	99.7
[32]	CNN	StyleGAN	98.5
[33]	Corneal Specular Highlight	StyleGAN2	AUC: 0.94
[34]	Eye Color	ProGAN, Glow	AUC: 0.70∼0.85
[11]	Irregular Pupil Shape	StyleGAN2	AUC: 0.91
[35]	Deep Neural Networks	StyleGAN2	Acc. 84
[36]	HRNet	StyleGAN2	Acc. 79.41
[37]	ResNet+ Inception	StyleGAN2	Acc. 96.69

and deep face recognition for feature extraction [27]. Using co-occurrence matrices and deep CNN, a unique method for identifying GAN-generated false pictures is proposed [28]. This method achieves over 99% accuracy on various GAN datasets with strong generalization. Next, with a one-shot GAN-produced fake face detection approach based on out-of-context object identification, previous research also addresses recognizing novel fake faces in shifting data settings and outperformed earlier methods on Style-GAN-generated false faces [29]. Then, another research creates FakeSpotter, a technique that tracks neuronal activity to detect AI-generated fake faces [30]. It has been proven successful and resilient against a variety of fake faces and perturbation attacks. A feature set to recognize deep neural network-generated pictures is also proposed using color image statistics [31].

On both natural and GAN-generated datasets, the proposed approach achieves a high detection accuracy of over 0.99 by combining global and local features, increasing learning on significant face regions with key points, and using metric learning for feature extraction [32]. With straightforward yet successful qualitative and quantitative evaluations, another approach uses discrepancies in corneal specular and highlights between the eyes in GAN-synthesized faces to discriminate between genuine and synthetic faces [33]. Another technique with strong AUC values up to 0.866 uses straightforward visual artifacts from face-editing processes to identify manipulations like Deepfakes and Face2Face [34]. Using a Residual Attention Network (RAN) for corneal specular highlight comparison, a framework in previous research examines eye irregularities to detect GAN-synthesized faces, producing a reliable and understandable model that works well even in settings with unbalanced data [11].

In a comparison of different deep learning-based face-detection classifiers, it is found that VGG19 performs the best, obtaining 95% accuracy on an upgraded dataset [35]. Moreover, iCaps-Dfake introduces a unique deepfake detection method integrating Local Binary Patterns (LBP), modified high-resolution networks (HRNet), and capsule nets, yielding a notable 20.25% increase in AUC above leading models [36]. Then, the proposed deepfake detection model called "DeepfakeNet" is influenced by the structure of ResNet and Inception [37]. It outperforms conventional models in terms of accuracy and cross-dataset detection capabilities.

A summary of previous studies is provided in Table III. It is evident from the discussion that many previous researchers have tried to address a particular question rather than the overall problem. Therefore, in this investigation, the we attempt to incorporate the robustness, explainability, and trustworthiness of the deepfake image detector into the discussion. To solve the robustness issue, augmentation methodologies with two independent datasets reflecting machine intelligence and human intelligence are combined. Next, an EfficientNet B0 is integrated with Shifted Window transformers to construct a reliable classifier. It is also shown how crucial gradient-based class activation maps are for determining the key areas to interpret the classification outcome.

III. RESEARCH METHOD

A. Overview

With an intention to develop a reliable, adaptable, and interpretable deepfake detection model, the we

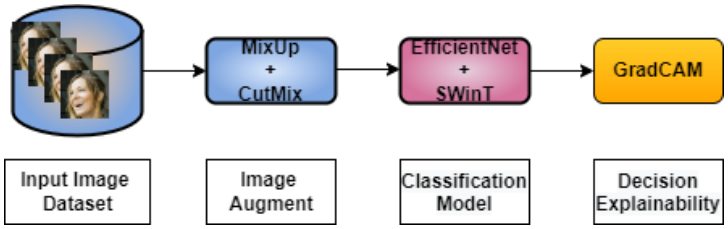

Fig. 1. Overview of the proposed method.

propose a three-phase model with augmentation, classification, and explanation modules. The various image augmentation methods are applied to generate data with wider variations. The objective is to develop a more generalizable dataset so the classifier can learn better during the training. Then, the researchers combine the EfficientNet and Hierarchical Shifted Windows transformer models in the classification module to build a reliable classifier to distinguish between the fake and real images. Next, the GradCAM module considers the Features Maps and Attention maps to provide visual hints for classifier decision-making so that the non-AI user can also understand the classifier's decision. The overall system details are described in Fig. 1.

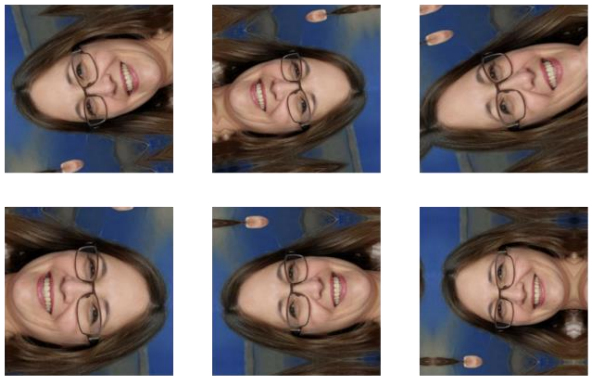

Fig. 2. Results of the regular augmentation methods.

B. Augmentations

In the augmentation module, the researchers apply various image augmentation methods to generate data with wider variations to develop a more generalizable dataset so that the classifier can learn better during the training. In particular, the researchers apply random CutMixUp augmentation. It combines lighter augmentation operations, such as resizing, flipping, and others, with heavy augmentation methods, including CutMix, and MixUp.

1) Regular Augmentations: The dataset is subjected to image augmentation methods to produce extra data for the model's training. The example can be seen in Fig. 2. It consists of several things as follows.

1) Resize: Each color image is resized to 384×384 pixels.
2) Random Flip: Each color image is randomly flipped using the horizontal and vertical axes.
3) Random Zoom: Each color image is randomly zoomed by 20% along the horizontal axis and 30% along the vertical axis.
4) Random Rotation: Every color image is randomly rotated by 30% over the vertical axis and 20% along the horizontal axis with reflection.

2) Mix-Up: A domain-neutral data augmentation strategy called MixUp relies on the assumption that linear interpolations on feature vectors should generate linear interpolations of associated targets to boost the training distribution [38]. MixUp augmentation is easy to implement and has a lighter process. MixUp makes the neural network more robust when dealing with ambiguous inputs or learning from incorrect labels. Integrating two images with class labels essentially averages the two images with the corresponding labels as new data. The researchers provide various actual and fake photographs made by GAN and the labels (Real, Fake).

Let x_i, x_j be the raw input vectors and y_i, y_j be the one-hot label encodings, then (x_i, y_i) and (x_j, y_j) be the examples drawn randomly from training data, and $\lambda \in [0,1]$. Then, it is $\lambda \sim$Beta(α, α), for $\alpha \in (0,\infty)$. By restoring the Empirical Risk Minimization principle as $\alpha \to 0$, the MixUp hyper-parameter α regulates the interpolation strength between feature-target pairings. It is shown in Eq. (1).

$$\begin{aligned} x^1 &= \lambda x_i + (1-\lambda)x_j, \\ y^1 &= \lambda y_i + (1-\lambda)y_j. \end{aligned} \tag{1}$$

3) CutMix: Since MixUp samples are rare and spatially ambiguous, they confuse the training model, particularly for localization. Rather than just eliminating pixels as in traditional image editing, CutMix [39] replaces the deleted areas with a patch from another

Fig. 3. Results of the MixUp augmentation method.

Fig. 4. Results of the CutMix augmentation method.

image. The class labels are also mixed based on the total number of pixels in the merged pictures. Even if uninformative pixels are absent during training in CutMix, the benefit of regional dropout, which focuses on non-discriminative parts of objects, is still present, boosting training effectiveness. The new patches have substantially improved localization capability, as the model can now identify the object from a partial viewpoint. The costs associated with training and inference remain consistent. The example can be seen in Fig. 3.

Let W, H, and C represent the width, height, and number of channels of an image x with the shape $W*H*C$. Let y be the ground truth label. Combining the two samples ($x1$, $y1$) and ($x2$, $y2$) yields a new sample ($x3$, $y3$). The produced training sample ($x3$, $y3$) is applied to train the model with its original loss function. In Eq. (2), it has $M \in 0, 1^{W \times H}$, $\bigodot$ as element-wise multiplication, and λ as the Beta distributed hyperparameter and represents a combination ratio between two sampled data points from the beta distribution.

$$\begin{aligned} x^1 &= M \bigodot x_A + (1 - M) \bigodot xB, \\ y^1 &= \lambda y_A + (1 - \lambda) y_B. \end{aligned} \tag{2}$$

There are steps to sample the binary mask M:

1) Sample the bounding box coordinates B = (r_x, r_y, r_w, r_h) on x_A and x_B,
2) Remove the region B in x_A,
3) Fill in with the cropped patch from B of x_B,
4) Sample rectangular masks M whose aspect ratio is proportional to the original image,
5) Sample the box coordinates uniformly,
6) In each training iteration, generate a CutMixed sample (x^1, y^1) (see Fig. 4) by combining randomly selected two training samples in a mini-batch. It is shown in Eq. (3).

$$\begin{aligned} r_x &\sim \text{Unif}(0, W), \\ r_w &= W\sqrt{(1-\lambda)}, \\ r_y &\sim \text{Unif}(0, H), \\ r_h &= H\sqrt{(1-\lambda)}, \\ \text{Cropped Ratio} &= (r_w r_h)/(W * H) \\ &= 1 - \lambda. \end{aligned} \tag{3}$$

C. Ensembled Model for Deepfake Image Detection

The ViT has demonstrated its ability to successfully replace the role of de facto CNN models on computer vision tasks. Recent research has shown that Transformer models outperform other CNN models on several computer vision tasks [40]. The CNN is mainly concerned with operating locally, while the Transformer blocks can improve feature activation globally across the relevant object. Thus, the researchers consider the EfficientNet B0 [41] (Fig. 5) and Shifted window Transformer models for ensembling to build a reliable and robust classifier [42].

1) EfficientNet: EfficientNet is a family of neural nets. It is introduced to scale the CNN model for increased computational performance intelligently using a neural architecture search. According to the compound scaling strategy, systematically scaling all three model parameters (depth, breadth, and resolution) yields outstanding performance compared to scaling only one parameter. The EfficientNet B0 model features a multi-objective neural network search to enhance precision and floating-point operations. It is a lightweight framework with 11M learnable parameters. Seven inverted residual blocks, each with a unique arrangement, are used in this architecture. In the inverted residual block, skip connections separate wider layers while interconnecting narrower levels. This approach

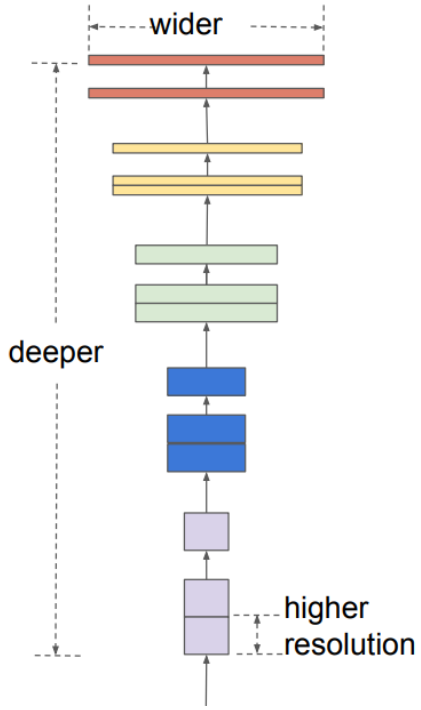

Fig. 5. Compound scaling in EfficientNet architecture [41].

drastically reduces the number of trainable parameters. These blocks integrate excitation with squeeze and swish activation. The MBConv block takes two inputs: data block and block of arguments. The data are outputs from the last layer. A block argument is a collection of attributes used inside an MBConv block, including kernel_size, number of repetitions, input filters, output filters, expansion ratio, id_skip, se_ratio, and others. The four phases of an MBConv block are expansion, depth wise convolution, squeeze and excitation, and output [40].

Linear and sigmoid activations are multiplied to produce swish activation. The squeeze and excitation block gives each channel a different weighting rather than treating them identically. It shows $\alpha \geq 1, \beta \geq 1$, and $\gamma \geq 1$, and grid search yields Alpha, Beta, and Gamma scaling multipliers for depth, width, and resolution. It is shown in Eq. (4).

$$\begin{aligned} \text{Depth} &= d = \alpha^{\varphi}, \\ \text{Width} &= w = \beta^{\varphi}, \\ \text{Resolution} &= r = \gamma^{\varphi}, \\ \text{Such that}, &\alpha^1 \beta^2 \gamma^2 \approx 2. \end{aligned} \tag{4}$$

D. Shifted Window Transformer

The Swin Transformer is a deep learning model built on Transformer that is more precise and efficient. One disadvantage of Vision Transformer is their quadratic computational cost, which is evident for high-resolution images. The fixed-size tokens in ViT are ineffective for visual tasks since the image size varies. The Swin Transformer is presented as a solution to both of these limitations. Swin Transformer introduces a hierarchical feature map with shifting window attention abilities. The Swin Transformer preserves cross-window connections while restricting focus to the local region using shifted windows. Hierarchical feature maps are the intermediate tensors produced and concatenated from one layer to another. The spatial dimension of feature maps is significantly minimized layer-wise. Because of the hierarchical feature maps, the Swin Transformer can be used for fine-grained prediction. The two major building blocks of the Swin Transformer, as in Fig. 6, are patch merging and the Swin Transformer block [42].

1) Patch Merging: The essential element of a feature map is a patch. The application of patch merging in Swin Transformer enables convolution-free down sampling. The steps included in the patch merging process are as follows.

1) Split the input image into a group of 2×2 patches.
2) Arrange the neighbouring patches depth-wise.
3) Combine the stacked groups. It is illustrated in Fig. 7.

2) Swin Transformer Block: In Swin Transformer, a transformer block substitutes the ViT conventional MSA module with a Window MSA (W-MSA) and a Shifted Window MSA (SW-MSA) module. The Swin Transformer block has two critical sub-units. A Normalization Layer1, an Attention Unit, a Normalization Layer2, and a Multi-Layer Perceptron (MLP) Layer constitute each sub-unit. The first sub-unit executes a W-MSA, whereas the second performs a SW-MSA [42].

3) Computing Self Window Attention: The standard multi-head self-attention unit of the ViT captures the relationship between the patches and global self-attention. The MSA for a high-resolution image is computationally very costly to capture. The W-MSA method is applied to address MSA difficulties. Fixed window size is used throughout the network to reduce the W-MSA computational complexity to linear from the quadratic complexity of standard MSA for the number of image patches. The network's modeling potential is limited if the self-attention computation is confined to the window. The SW-MSA module is used after the W-MSA module to avoid this issue [42]. Figure 8 shows the attention within every window using the W-MSA method.

4) Shifted Window Self-Attention: Let M be the size of the window. A SW-MSA shifts the windows

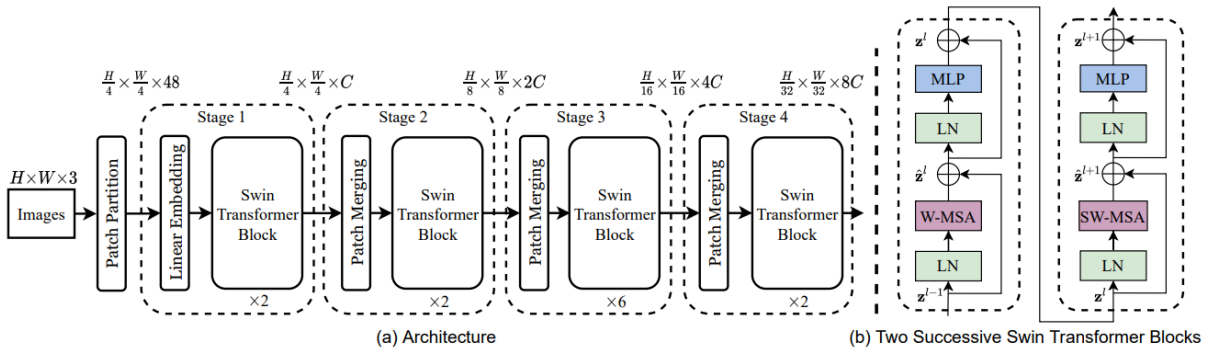

Fig. 6. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows [42].

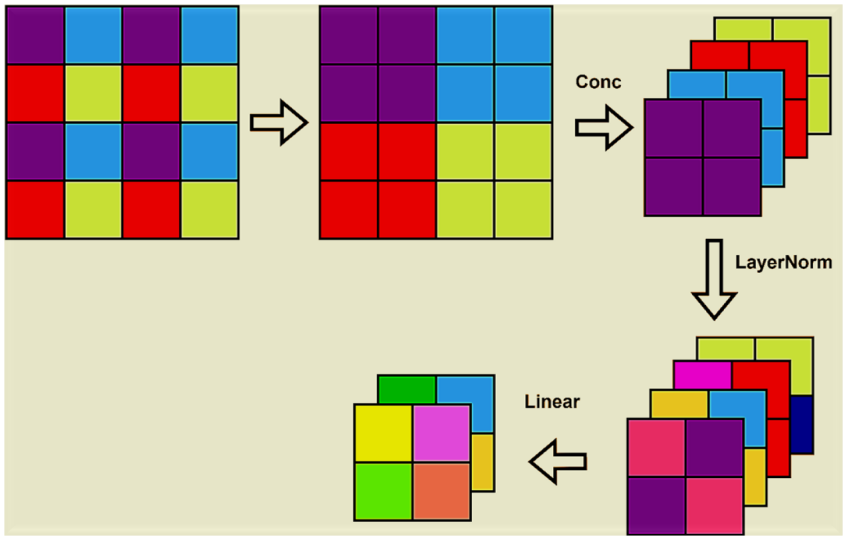

Fig. 7. Patch merging phase.

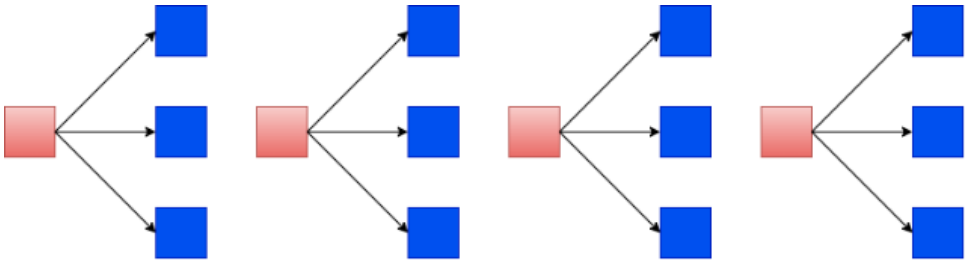

Fig. 8. Attention within every window using the Window MSA method.

by a factor of $M/2$ towards the bottom right corner to generate the cross-window connections. The Swin Transformer employs a 'Cyclic Shift' method to apply 'orphaned' patches into windows with incomplete patches. This moving window approach creates essential cross-connections between windows. It has been demonstrated to improve network performance. Figure 9 shows the computation of SW-MSA.

5) Ensembling the EfficientNet and SWinT: The architecture of the deepfake image classification technique is presented in Fig. 10. The EfficientNet B0 model and the Swin Transformer are combined to boost performance in the proposed image classification approach. To begin with, it divides the input picture collection into train and test sets. The train set is used to train the ensemble model, which includes

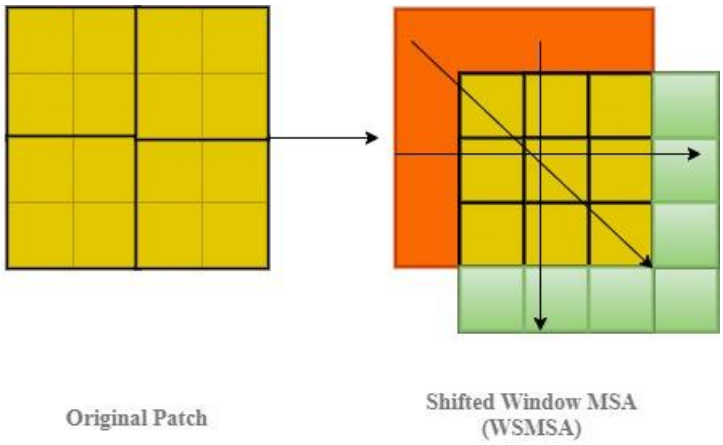

Fig. 9. Computation of Shifted Window MSA.

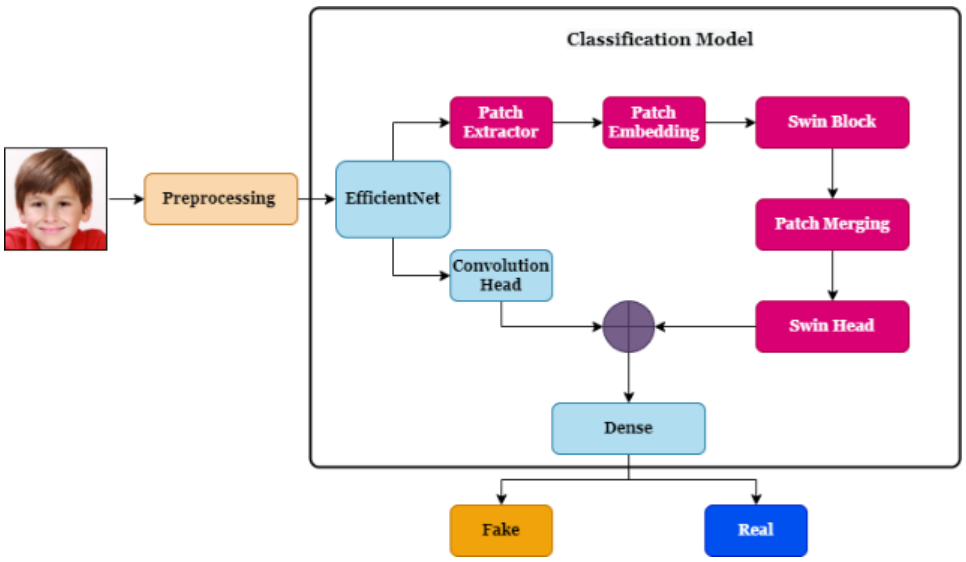

Fig. 10. Architecture of the deepfake image classification technique.

the EfficientNet B0 and Swin Transformer. Various augmentations are done to each input image in the train set, and the photos are scaled to 384×384×3. The resized photos are then loaded into the EfficientNet B0 model, which generates feature maps.

Furthermore, the intermediate layer block6a_expand_activation generates a 24×24 feature map that is forwarded to the Swin Transformer for further refining of the learned representations. The Swin Transformer uses the patch extractor, patch embedding, and Swin blocks to process the 24×24 feature map. Following patch merging, the resultant patches from the Swin block are sent to the Swin head with the features. These characteristics are combined with the features from the convolutional head of the EfficientNet model.

By combining the outputs of the EfficientNet B0 and Swin Transformer, the model acquires a complete comprehension of the input image. This integration is critical since both designs complement each other in feature extraction, resulting in better picture representations. The concatenated feature map is then supplied into a thick layer responsible for classification. Based on the learned attributes, the model can now make educated judgments on the class label of each image in this final classification stage. Combining the use of Swin Transformer with EfficientNet B0 offers benefits in terms of efficiency and attention-based capabilities, improving classification performance. The model is a strong and effective solution for image classification problems due to its capacity to handle complicated pictures in an efficient manner. Overall, the suggested

technique displays the ability to significantly enhance picture categorization by exploiting varied architectures.

E. GradCAM: Explainability

The black-box character of the classifier is addressed by the research using the pixel attribution method. The gradient of the model is used to quantify how each change in pixel affects the prediction of the model. This gradient-based class activation mapping technique illustrates how the classifier arrived at its conclusions. The last layer of the model receives backward propagation of visual reasoning for decisions. Attention and feature maps are taken into account when assessing the predictions produced by earlier model layers. The key portions of the input image are highlighted using a heatmap to provide a clearer understanding [43]. These heatmaps provide very abstract, high-dimensional data that is pertinent to each class. The gradient for the relevant class is used to weigh each pixel in the feature map. The average of the weighted feature maps is computed, yielding pixel values ranging from -1 to +1. After that, the values are transmitted via linear Rectified Units (ReLU). The ReLU function sets non-negative values to 1. The gradient areas involved in the prediction are chosen, decreased, resized, and rescaled. Finally, the heatmap is superimposed over the original image to provide visual insight into the model's decision-making process. This method aids in determining which portions of the input picture are important for the classifier's final prediction. Figure 11 shows the gradient class activation maps. Then, Algorithm 1 shows the GradCAM for deepfake image classification.

IV. Results and Discussion

A. Experimental Details

The proposed method has been implemented on a Dell Precision 5820 Tower Workstation running Windows 11 Pro with an Intel Xeon W-2225 quad-core CPU clocked at 4.1 GHz. The Keras framework and the Python programming language are used to program the algorithm. On the custom dataset, the model is trained across 30 epochs.

B. Results

The researchers have taken 14,204 images for the validation set. There are 7,096 fake photos and 7,108 real images. Let T_+ be the number of true positive samples, T_- be the true negative samples, F_+ be the false positive samples, and F_- be the number of false negative samples, respectively. Then, the metrics employed to evaluate the classifier's performance are accuracy (A), precision (P), recall (R), and F1 score (F_1). They are defined in Eq. (5).

$$\begin{aligned} A &= [\frac{(T_+ + T_-)}{T}], \\ P &= [\frac{(T_+)}{T_+ + F_+}], \\ R &= [\frac{(T_+)}{T_+ + F_-}], \\ F_1 &= [\frac{2PR}{P + R}]. \end{aligned} \tag{5}$$

The accuracy and loss of the model are recorded during a 30-epoch training period. The researchers use validation accuracy to monitor the model training. The accuracy and loss values of the model are shown in Figs. 12 and 13. During training, an intriguing pattern emerges in the performance of the CutMixUp training model. As can be seen from Figs. 12 and 13, the diminishing loss and rising accuracy in the early epochs make the model perform better and better. The accuracy rate drops to 84.05% at the seventh epoch. It is the lowest value on the validation set, and the loss abruptly drops to 0.4108. This situation can be explained by the validation data, including ambiguous patterns or difficult samples that the model finds difficult to categorize appropriately. Misclassifications and a reduction in overall accuracy may result from such patterns. Following the seventh epoch, the model begins to recover and improves in subsequent epochs. This pattern indicates that the model is adjusting to the complicated data distribution in the validation set by learning from the difficult samples. As a consequence, its performance continuously improves, resulting in more accuracy and better results in successive epochs.

C. Discussion

The research objective is to build a classification model that is reliable, generic, and simpler to comprehend. The EfficientNet B0 and Swin Transformer techniques are integrated to create a reliable classifier. The proposed method produces good classification results, as demonstrated by experiments on a customized fake image dataset. Then, the classification performance of the proposed model is examined using the basic model without any enhancements to the data. The effect of regular and lighter augmentation on the dataset is evaluated. The dataset is subjected to procedures, such as randomly rescaling, rotating, and flipping, to create the regular augmentation model. Following that, MixUp augmentation is performed, and statistics are recorded. The previous lighter augmentation is merged with MixUp and CutMix methods to build the

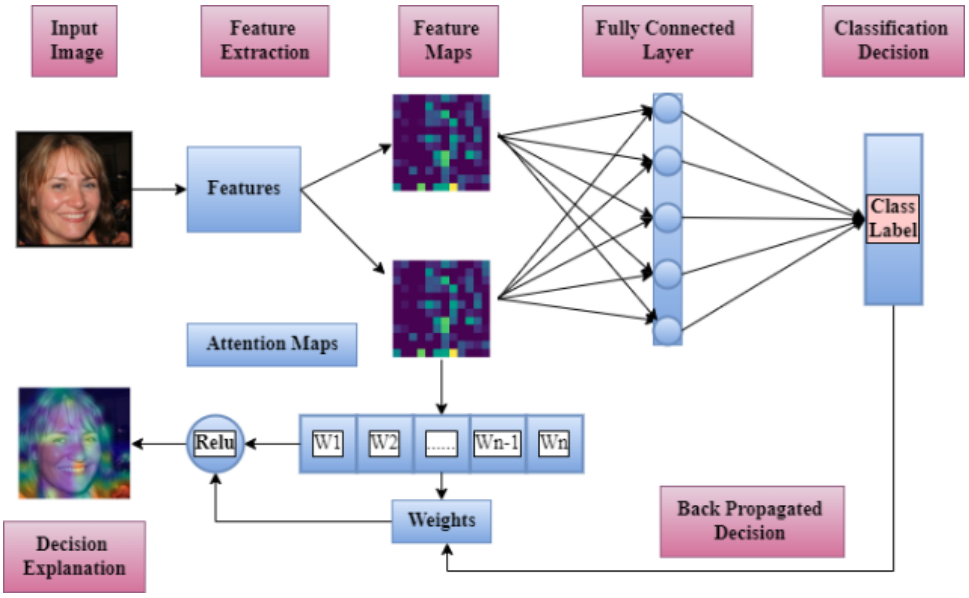

Fig. 11. Computing the gradient class activation maps.

Algorithm 1: GradCAM for Deepfake Image Classification

Input:

- Trained Efficient-Swin Transformer model with final prediction layer and intermediate convolutional layers.
- Input image for classification.

Output:

- Heatmap highlighting the significant regions of the input image that contribute to the model's binary classification decision.

Step1. Start

Step2. Forward Pass:

a. Input the image into the trained binary classification model

b. Perform a forward pass through the model to obtain the final prediction

Step3. Backward Pass

a. Calculate the gradients of the final prediction with respect to the intermediate convolutional feature maps.

b. These gradients indicate the importance of each feature map for the binary classification decision.

Step4. Global Average Pooling (GAP):

a. Average the gradients over the spatial dimensions of each feature map.

b. This process results in a weight value for each feature map representing its significance for the binary classification task.

Step5. Heatmap Generation:

a. Multiply each feature map by its corresponding weight to get the weighted feature maps.

b. Combine the weighted feature maps element-wise using addition or summation to create a heatmap.

c. The heatmap highlights the regions in the input image that have a strong influence on the model's prediction.

Step6. Activation Map Visualization:

a. Overlay the generated heatmap onto the original input image.

b. The heatmap's intensity at each pixel represents its importance for the model's binary classification decision.

c. The areas with higher intensity in the heatmap correspond to the regions that significantly contribute to the target class prediction.

Step7. End

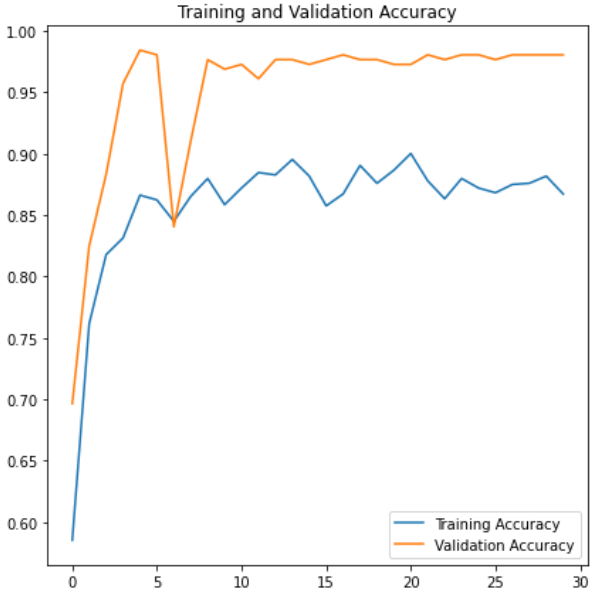

Fig. 12. Training and validation accuracy.

Fig. 13. Training and validation loss.

model CutMixUp. The experimental results are given in Table IV.

The model functions well even in the absence of data augmentation, with a classification accuracy of 95.78%. The precision and recollection statistics suggest that the model can successfully discriminate between true and misleading images. However, there is a need for improvement, particularly in terms of decreasing false positives and negatives. The model is trained without any data augmentation that suffers from

TABLE IV
RESULTS ON TESTING SETS.

Model	Accuracy	Precision	Recall	F1 Score
No Augmentation	0.9578	0.9728	0.9968	0.9904
Regular Augmentation	0.9612	0.9654	0.9965	0.9924
MixUp	0.9672	0.9762	0.9958	0.9889
CutMix	0.9682	0.9769	0.9957	0.9886
CutMixUp	0.9844	0.9845	0.9912	0.9827

a lack of training data diversity. As a result, it may become susceptible to overfitting when the machine remembers the training data but fails to generalize to new and previously unknown data. Compared to the supplemented models, this limitation dramatically reduces accuracy, precision, recall, and F1 score.

Variations in the training data are introduced through regular augmentation techniques, such as random rotations, flips, and transformations. These modifications make the model more resistant to changes in the orientation and location of items in the images. The performance has increased because the model can tolerate variances in the test data and generalize better than the no augmentation scenario. The model performs better on all measures when routine data augmentation strategies are used. The accuracy rises to 96.12%, indicating improved overall performance. Both precision and recall levels also improve, indicating that data augmentation helps the model to generalize better and generate more accurate predictions.

By linearly interpolating between pairs of actual and artificial images and the labels that go with them, MixUp augmentation produces fresh examples for training. The model learns a more generalized decision boundary as a result of the sample blending, which improves generalization performance. The model's performance is further improved by MixUp augmentation, which produces better accuracy, precision, and recall values. It indicates the potency of MixUp in enhancing the model's capacity to discriminate between authentic and fake photos. Hence, it lowers both false positives. The improved model robustness is evidenced by the greater accuracy, precision, recall, and F1 score.

Moreover, CutMix augmentation combines patches from multiple images to generate additional training examples. As a result, the model is forced to focus on important portions of the pictures and acquire more discriminative features. Compared to the ordinary augmentation example, the model benefits from the variety offered by CutMix, resulting in better accuracy, precision, recall, and F1 score. CutMix augmentation, like MixUp augmentation, improves model performance. The model achieves greater accuracy, precision, and recall scores, demonstrating that CutMix aids the

model's generalization and prediction accuracy.

Among all procedures tested, the combination of CutMix and MixUp (CutMixUp) produces the most favorable results. The model achieves excellent precision and recall levels and an overall accuracy of 98.44%. With the use of these two strategies for data augmentation, the model can take advantage of the EfficientNet B0 model's efficiency as well as the Swin Transformer's attention-based capabilities, which substantially enhances classification performance. It includes a variety of samples with blended patches and encourages linear interpolations between actual and false images. This combination improves the model's capacity to handle complicated and varied data, resulting in the highest levels of reliability, precision, recall, and F1 score of the approach tested.

The influence of data augmentation on the model's training process is the primary cause of the differences in the assessment outcomes as a result. Augmentation approaches assist the model in learning more robust and broad characteristics, allowing it to perform better on previously unknown data. The higher performance of the CutMixUp approach implies that combining several data augmentation procedures is advantageous for getting cutting-edge outcomes in deepfake image classification tasks, notably discriminating between real and fabricated images.

When the proposed EfficientNet Swin Transformer Net is compared to existing state-of-the-art approaches, it is clear that the model outperforms them. It reaches an accuracy of 98.45%, which is comparable to the FiD approach. Furthermore, it surpasses other approaches in terms of accuracy, recall, and F1 score, demonstrating a great capacity to categorize both actual and fake images accurately. Then, when compared to Taeb's technique, the EfficientNet Swin Transformer Net improves accuracy (84.05% vs. 96.69%), precision (87.00% vs. 98.45%), and recall (79.00% vs. 99.12%). However, the comparison does not supply the recall statistic for Taeb's technique. Similar to this, the suggested model outperforms iCaps-Dfake in all metrics: accuracy (84.05% vs. 79.41%), precision (87.00% vs. 90.12%), and recall (79.00% vs. 76.45%). DeepfakeNet outperforms the proposed algorithm in terms of accuracy, but it lacks information on precision, recall, and F1 score. Overall, the EfficientNet Swin Transformer Net has great performance and resilience in identifying real and forged photos. Its ability to compete with other cutting-edge approaches implies that the combined EfficientNet B0 and Swin Transformer architecture, coupled with CutMixUp augmentation, gives benefits in processing complicated picture input and delivering reliable classification results. The comparison of the proposed model illustrates that the performance is in line with state-of-the-art methods. It is shown in Table V.

TABLE V
COMPARISON OF EFFICIENTNET SWIN TRANSFORMER NETWORK AGAINST THE STATE-OF-THE-ART METHODS.

Model	Accuracy	Precision	Recall	F1 Score
FiD [24]	0.9845	0.9845	0.9845	0.9845
Taeb [35]	0.8400	0.8700	0.7900	-
iCaps-Dfake [36]	0.7941	0.9012	0.7645	0.8270
DeepfakeNet [37]	0.9669	-	-	-
EfficientNet Swin Transformer Network	0.9845	0.9845	0.9912	0.9827

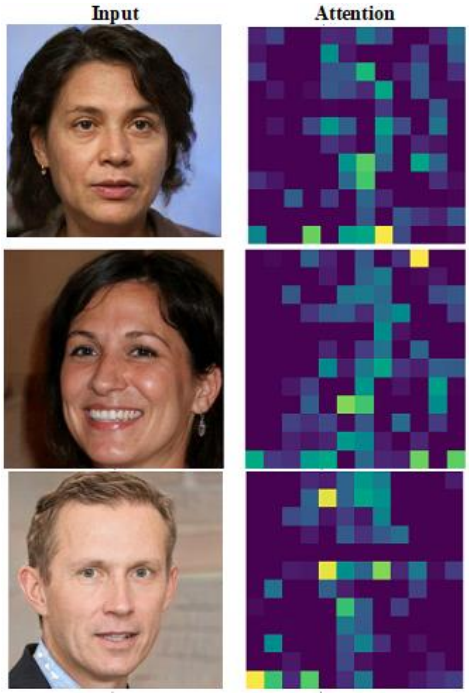

Fig. 14. Visualizing the activation maps for fake images.

D. Model Explainability: GradCAM Analysis

The pixel attribution approach is significant in the context of forensic analysis because it makes it easier for users to understand how the model makes decisions. Using this method, the researchers enable human analysts to acquire greater insights into how the model arrives at its predictions, which is especially useful when working with complicated block-box classifiers. When a human user looks at the heat maps and attention maps produced by the pixel attribution approach, as seen in Fig. 14, they may recognize the areas of interest that the model focuses on while determining the factual accuracy of the image. Thanks to this visual explanation, the human user may better comprehend

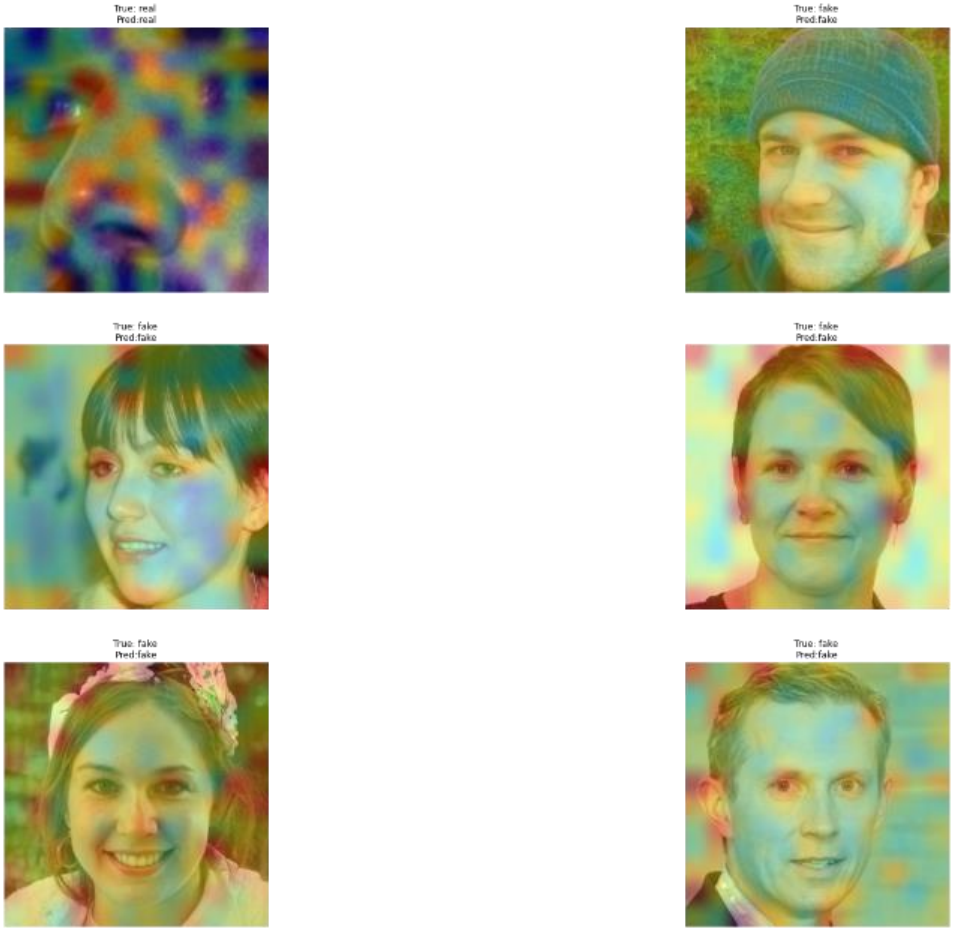

Fig. 15. Heat maps for prediction description.

which particular aspects in the image have affected the model's categorization choice. It gives them important context.

A clear and understandable representation of the model's thought process is provided by the overlay of attention maps and prediction results on the source images, as shown in Fig. 15. Human users may quickly identify the crucial regions that the model evaluates, making the conclusion more visible and understandable. Human analysts can undertake a thorough forensic examination of the photos under investigation with the help of this understanding of the model's behavior. Furthermore, the attention maps depict the model's evaluation in an intelligible manner, encapsulating a more complete image analysis than just selected areas. This broader viewpoint allows for a better comprehension of the model's conclusions since it is in line with how human analysts naturally observe and evaluate images. By including the human user in the interpretation process, the pixel attribution approach bridges the gap between the model's complicated decision-making and the human user's capacity to grasp and validate the findings. This synergy improves the model's dependability and trustworthiness in forensic analysis, making it a useful tool for discriminating between true and fraudulent information with great precision and transparency.

V. Conclusion

Today, a variety of methods and tools are available for processing multimedia. It is due to the recent rapid breakthroughs in artificial intelligence, machine learning, and deep learning. Some criminal groups have exploited these techniques and tools to spread rumours and false information, stir political unrest and hate, or even intimidate and threaten the public by creating convincing and plausibly fabricated text, images, audio, and videos.

Several deep learning and machine learning techniques have been employed in the early works to iden-

tify fake information. However, the capacity to generalize and explain the classification decision remains challenging. Thus, the research examines the visual assessments of an EfficientNet and Swin Transformer hybrid model based on CNN and Transformer architectures. By combining the lighter data augmentations used for model training with the domain-independent heavy augmentation CutMix and regularisation techniques like MixUp, the researchers attempt to overcome the generalization problem. The deepfake images are visually interpreted for human comprehension using the GradCAM technology. GradCAM employs visual class activation maps to draw attention to the veracity or falsity of the information.

The proposed method successfully distinguishes between real and manipulated images. The proposed method obtains an accuracy of 98.45% with a loss of 0.11125 on the custom dataset. The dataset is composed of real, GAN-generated, and human-altered images. The model obtains an F1 score of 0.9827, a precision of 0.9845, and a recall of 0.9912 on the test dataset. The process also offers an easier way to understand the model's decisions. This understanding helps in different forensic investigation works. The results of the proposed model show that the model can be used to understand the nature of fake images.

There are a few limitations to the research that should be noted. First, it ignores the assessment of adversarial image attacks, which are increasingly prevalent in the deepfake generation. It concentrates entirely on artificially and manually created images. Future research should attempt to improve the proposed method's ability to deal with such assaults successfully. Second, the research only examines image data, and it is critical to expand the investigation to include deepfake videos, which are as essential in disseminating deception. Real-time video data analysis might be investigated to address the issues faced by live deepfake material. Furthermore, future research may include constructing lighter appropriate models for less processing-powered devices and providing greater accessibility and use of deepfake detection technologies.

References

[1] H. Murfi, N. Rosaline, and N. Hariadi, "Deep autoencoder-based fuzzy c-means for topic detection," *Array*, vol. 13, pp. 1–9, 2022.

[2] A. Kammoun, R. Slama, H. Tabia, T. Ouni, and M. Abid, "Generative adversarial networks for face generation: A survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[3] A. Sharma, V. Sharma, M. Jaiswal, H. C. Wang, D. N. K. Jayakody, C. M. W. Basnayaka, and A. Muthanna, "Recent trends in AI-based intelligent sensing," *Electronics*, vol. 11, no. 10, pp. 1–39, 2022.

[4] S. Li, V. Dutta, X. He, and T. Matsumaru, "Deep learning based one-class detection system for fake faces generated by GAN network," *Sensors*, vol. 22, no. 20, pp. 1–23, 2022.

[5] T. Bollé, E. Casey, and M. Jacquet, "The role of evaluations in reaching decisions using automated systems supporting forensic analysis," *Forensic Science International: Digital Investigation*, vol. 34, pp. 1–13, 2020.

[6] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q. V. Pham, and C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey," *Computer Vision and Image Understanding*, vol. 223, 2022.

[7] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face x-ray for more general face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, June 14–19, 2020, pp. 5001–5010.

[8] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, July 21–26, 2017, pp. 5533–5542.

[9] P. S. Sisodiya, "DeepFake detection using various deep learning techniques," Ph.D. dissertation, Delhi Technological University, 2022.

[10] H. Zhao, W. Zhou, D. Chen, W. Zhang, and N. Yu, "Self-supervised transformer for deepfake detection," 2022. [Online]. Available: https://arxiv.org/abs/2203.01265

[11] H. Guo, S. Hu, X. Wang, M. C. Chang, and S. Lyu, "Robust attentive deep neural network for detecting GAN-generated faces," *IEEE Access*, vol. 10, pp. 32 574–32 583, 2022.

[12] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *ICLR 2018*, Vancouver, Canada, April 30–May 3, 2018.

[13] L. Ma, K. Huang, D. Wei, Z. Y. Ming, and H. Shen, "FDA-GAN: Flow-based dual attention GAN for human pose transfer," *IEEE Transactions on Multimedia*, vol. 25, pp. 930–941, 2021.

[14] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-*

nition (CVPR), Virtual, June 14–19, 2020, pp. 3207–3216.

[15] P. Korshunov and S. Marcel, "Deepfakes: A new threat to face recognition? assessment and detection," 2018. [Online]. Available: https://arxiv.org/abs/1812.08685

[16] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 27–Nov.2, 2019, pp. 1–11.

[17] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, "The DeepFake Detection Challenge (DFDC) dataset," 2020. [Online]. Available: https://arxiv.org/abs/2006.07397

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, June 14–19, 2020, pp. 2889–2898.

[19] NVlabs, "Flickr-Faces-HQ Dataset (FFHQ)," 2019. [Online]. Available: https://archive.org/details/ffhq-dataset

[20] Computational Intelligence and Photography Lab, Yonsei University, "real-and-fake-face-detection," 2019. [Online]. Available: https://archive.org/details/real-and-fake-face-detection

[21] 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)*. San Jose, CA, USA: IEEE, March 28–30, 2019, pp. 506–511.

[22] H. Mo, B. Chen, and W. Luo, "Fake faces identification via convolutional neural network," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, Innsbruck, Austria, June 20–22, 2018, pp. 43–47.

[23] N. Hulzebosch, S. Ibrahimi, and M. Worring, "Detecting CNN-generated facial images in real-world scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Virtual, June 14–19, 2020, pp. 642–643.

[24] G. Tang, L. Sun, X. Mao, S. Guo, H. Zhang, and X. Wang, "Detection of GAN-synthesized image based on discrete wavelet transform," *Security and Communication Networks*, vol. 2021, pp. 1–10, 2021.

[25] X. Yang, Y. Li, H. Qi, and S. Lyu, "Exposing GAN-synthesized faces using landmark locations," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, Paris, France, July 3–5, 2019, pp. 113–118.

[26] I. Demir and U. A. Ciftci, "Where do deep fakes look? Synthetic face detection via gaze tracking," in *ACM symposium on eye tracking research and applications*, Virtual, May 25–27, 2021, pp. 1–11.

[27] N. T. Do, I. S. Na, and S. H. Kim, "Forensics face detection from GANs using convolutional neural network," *ISITC*, vol. 2018, pp. 376–379, 2018.

[28] L. Nataraj, T. M. Mohammed, S. Chandrasekaran, A. Flenner, J. H. Bappy, A. K. Roy-Chowdhury, and B. S. Manjunath, "Detecting GAN generated fake images using co-occurrence matrices," 2019. [Online]. Available: https://arxiv.org/abs/1903.06836

[29] H. Mansourifar and W. Shi, "One-shot GAN generated fake face detection," 2020. [Online]. Available: https://arxiv.org/abs/2003.12244

[30] R. Wang, F. Juefei-Xu, L. Ma, X. Xie, Y. Huang, J. Wang, and Y. Liu, "FakeSpotter: A simple yet robust baseline for spotting AI-synthesized fake faces," 2019. [Online]. Available: https://arxiv.org/abs/1909.06122

[31] H. Li, B. Li, S. Tan, and J. Huang, "Identification of deep network generated images using disparities in color components," *Signal Processing*, vol. 174, 2020.

[32] B. Chen, W. Tan, Y. Wang, and G. Zhao, "Distinguishing between natural and GAN-generated face images by combining global and local features," *Chinese Journal of Electronics*, vol. 31, no. 1, pp. 59–67, 2022.

[33] 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)*. Toronto, ON, Canada: IEEE, June 6–11, 2021, pp. 2500–2504.

[34] 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)*. Waikoloa, HI, USA: IEEE, Jan. 7–11, 2019, pp. 83–92.

[35] M. Taeb and H. Chi, "Comparison of deepfake detection techniques through deep learning," *Journal of Cybersecurity and Privacy*, vol. 2, no. 1, pp. 89–106, 2022.

[36] S. S. Khalil, S. M. Youssef, and S. N. Saleh, "iCaps-Dfake: An integrated capsule-based model for deepfake image and video detection," *Future*

Internet, vol. 13, no. 4, pp. 1–19, 2021.

[37] D. Gong, Y. J. Kumar, O. S. Goh, Z. Ye, and W. Chi, "Deepfakenet, an efficient deepfake detection method," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 201–207, 2021.

[38] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017. [Online]. Available: https://arxiv.org/abs/1710.09412

[39] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 27–Nov.2, 2019, pp. 6023–6032.

[40] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "CMT: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, New Orleans, Louisiana, June 19–24, 2022, pp. 12 175–12 185.

[41] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, PMLR*. Long Beach, California, USA: PMLR, June 9–15, 2019, pp. 6105–6114.

[42] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Virtual, Oct. 11–17, 2021, pp. 10 012–10 022.

[43] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.