ConvNext-PNet: An interpretable and explainable deep-learning model for deepfakes detection

Hafsa Ilyas
University of Engineering and Technology, Taxila, Pakistan
hafsailyas97@gmai1.com

Ali Javed
University of Engineering and Technology, Taxila, Pakistan
ali.javed@uettaxila.edu.pk

Khalid Mahmood Malik
University of Michigan-Flint, MI, USA
drmalik@umich.edu

Abstract

The evolution of artificial intelligence (AI) techniques in recent years has increased the generation of fake content including AI-generated text, images, audio, and videos. Among which the fake visual content commonly known as deepfakes has imposed a great threat to society due to its negative impacts. To mitigate the adverse aspects of deepfakes, the research community has introduced various deepfakes detection methods. However, these deepfakes detection methods lack the interpretability and explainability of the decision-making process. The interpretable model increases trustworthiness as it provides the reasoning for classifying outcomes as real or fake. Therefore, in this paper, we have introduced ConvNext-PNet, which is a prototypical-based learning framework for the interpretable and explainable detection of visual deepfakes. In the proposed framework, prototype learning is incorporated into the modified ConvNext model that improves the discriminative features learning capability of the proposed framework along with the explainability aspect. The performance of ConvNext-PNet is evaluated on challenging datasets including FaceForensics++ (FF++), CelebDF, DFDC-P, and DeepFakeFace (DFF) datasets. The robustness of the proposed model is validated through various experiments along with the interpretability analysis. The quantitative results demonstrate the effectiveness of the model for the detection of visual manipulation, whereas the model interpretability and explainability aspect increases the trustworthiness via providing reasoning for the model predictions.

Keywords: ConvNext-PNet, ConvNext, ProtoPNet, Interpretable deepfakes detection, Explainable deepfakes detection.

1. Introduction

With the advancement in artificial intelligence (AI), generative AI has also evolved tremendously in recent years. Nowadays generative AI has been utilized for creating synthetic content including text, images, audio, and videos. With the introduction of variational autoencoders (VAEs), generative adversarial networks (GANs), and more recent diffusion models, highly realistic synthetic images and videos, also known as deepfakes, can be created [1]. Deepfakes generation has gained substantial attention as it can be used for performing various malicious activities such as propagating disinformation, defaming famous personalities, and misguiding individuals to create chaos. Besides the negative impacts, it has numerous positive aspects including its application in movie production, virtual meetings, and entertainment [1]. However, the unethical use of deepfakes necessitates the development of countermeasures.

Existing literature on deepfakes detection mainly focuses on the effective, accurate, and generalizable deepfakes detection methods, however, less attention is given to the intrinsically interpretable deepfakes detection approaches. There is a lack of deepfakes detection models that are interpretable by design along with built-in self-explainability. In other words, current deepfakes detection approaches can identify fake and real content but are unable to explain the decision-making process of the model. To overcome this knowledge gap, we introduce a prototype learning-based deepfakes detection framework namely ConvNext-PNet. More specifically, we present an intrinsically interpretable deepfake detection approach where the modified ConvNext model is incorporated with the prototypical part network (ProtoPNet). The modified ConvNext better learns the discriminative features of real and fake images, while the prototypical part network provides the explainability aspect of the model. The main contributions of the proposed research work are as follows:

1. We propose a novel prototype learning-based approach, namely ConvNext-PNet for the interpretable and explainable detection of deepfakes.
2. We introduce ConvNext as the base architecture in our framework to reliably capture more crucial features specific to synthetic data.

3. We performed extensive experimentation to show the effectiveness of the proposed explainable framework.

2. Related Works

This section presented the discussion on the existing deepfakes detection approaches including the explainable methods implementing the post-hoc explainability analysis. Along with, the methods that are interpretable by design are also described in this section.

2.1. Deepfakes Detection Methods

Deepfakes detection models have gone through significant advancement from hand-crafted feature-based methods [2, 3] to deep learning-based approaches [4, 5] and unified detection techniques [6]. Zhang et al. [4] introduced a two-stream neural network combining image spatial and residual domains to detect the tampering artifacts in deepfakes videos. The spatial stream captured the tampering artifacts while the residual stream detected tampering traces from the image residual. Deepfakes detection method namely the localized artifact attention network (LAA-Net) was presented in [5] that incorporated an attention mechanism and enhanced feature pyramid network (E-FPNet). This method [5] is sensitive to structural perturbations and performs worst when tested on noisy images. The traditional hand-crafted-based techniques lack the generalization ability; however, the deep learning-based methods are not interpretable and vulnerable to adversarial attacks.

2.2. Explainable Deepfakes Detection Approaches

Because of the non-transparent decision-making, and opaque nature of deep learning models, explainable AI (XAI) has emerged as a field focused on making AI models understandable and hence trustworthy [7]. XAI techniques include post-hoc explainability analysis and intrinsically interpretable methods. In post-hoc analysis, trained convolutional neural networks (CNNs) are interpreted by highlighting the parts of the input that contribute most to the final prediction. Techniques of post-hoc analysis include saliency map visualization, activation maximization, and deconvolution. Mostly in the deepfakes detection, post-hoc explainability analysis is performed on trained CNNs to provide the explainability aspect. For instance, to detect deepfakes images, a supervised contrastive learning approach was introduced in [8], providing the model explainability through heatmaps generated corresponding to the model's last layer. In [9], a ResNet-Swish-Dense54 model was introduced for effective deepfakes detection along with explainability. The explainability power of the model was analyzed via generation heatmaps corresponding to the last layer of the trained model. A pairwise learning approach along with color space exploitation was presented in [10] for the generalizable deepfakes detection. More specifically, the multi-channel Xception network was employed with attentive pairwise learning. This approach [10] then utilized t-SNE and class activation maps to explain the decision-making process of the model. The above-mentioned approaches implementing the post-hoc visualization analysis provide insights of the input image regions contributing to the final prediction. However, these approaches are unable to explain how the models arrive at the prediction decision.

2.3. Intrinsically Interpretable Methods

Intrinsically interpretable methods involve the modification of the model's architecture before training to develop such frameworks that are interpretable by design with self-explaining ability. These methods provide predictions that are more understandable. One of the XAI intrinsic methods in the image recognition field is a prototype-based approach where explainability is provided based on visual similarity. The prototype-based approach implements case-based reasoning identical to the humans' way of recognizing objects. This approach highlights the regions of the input image under examination, along with providing the prototypical cases identical to those regions. Chen et al. [11] introduced the prototypical part network that incorporates prototype learning into CNNs for image recognition. In [12], a model was presented that organized prototypes hierarchically and performed prediction at every level of taxonomy. To provide more explainability to ProtoPNet, [13] introduced the framework that specified the textual quantitative explanation for the model decision. of finding the similarity between prototypes and image patches. The textual information determined the influence of characteristics such as hue, texture, saturation, contrast, and shape on the model decision.

3. Methodology

In this section, in-depth details of the architecture and training process of the proposed framework ConvNext-PNet in the context of visual deepfakes detection are provided. The overall architecture of the proposed framework is presented in Figure 1.

3.1. Pre-processing

Pre-processing involves the extraction of facial frames from the videos using Multi-task Cascaded Convolutional Neural Networks (MTCNN) [18]. MTCNN detects and extracts facial features with high accuracy from the given images having faces of distinct orientations and sizes. The extracted facial frames are resized to $224 \times 224 \times 3$ and then input to the proposed framework for further processing.

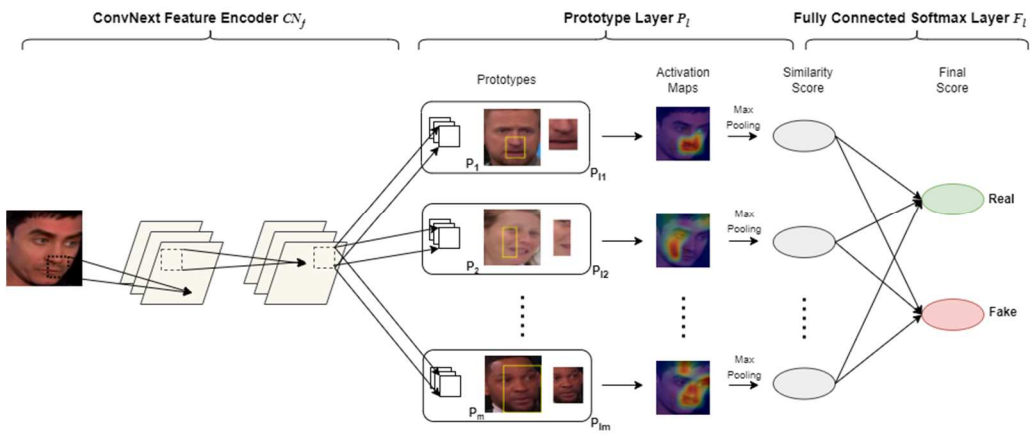

Figure 1: ConvNext-PNet framework.

3.2. ConvNext-PNet

The proposed ConvNext-PNet framework is comprised of 3 main components, ConvNext feature encoder CN_f with parameters ω_{CN_f}, prototype layer P_l, and fully connected layer F_l with weights ω_{F_l}. Let $X = \{x_j, y_j\}_{j=1}^{N}$ be the given facial frames dataset, where x_j is the input facial frames with $y \in \{0, 1\}$ representing the label for fake and real. These facial frames x_j are sampled as input to the ConvNext-PNet.

ConvNext Feature Encoder. ConvNext feature encoder CN_f initialized with ImageNet pretrained weights, is introduced as the backbone architecture of the proposed framework ConvNext-PNet. The design choices for ConvNext architecture are adapted from Swin Transformer retaining the simplicity and efficiency of CNNs [19] and thus can better capture the crucial features from synthetic images. ConvNext proved to outperform the Swin Transformer in terms of performance and simplicity [19]. Primarily, the ConvNext is based on the architecture of ResNet that is modernized towards the design of Swin Transformer in the following ways:

- The stem cell in ResNet is replaced with the patchify stem implemented using 4×4 non-overlapping convolutional layers. Also, the stage computation ratio is adjusted to 1:1:3:1.
- Depth-wise convolutions are employed by adopting the ResNext design and adjusting the network width to 96.
- Inverted bottlenecks are adopted from the Transformer block design.
- Large kernel-sized convolutions are utilized.
- Fewer activation functions and normalization layers are utilized.

Additionally, we utilized the LeakyReLU activation function in each ConvNext block. LeakyReLU is computationally efficient and its non-zero gradient for negative inputs enables ConvNext to learn from both negative and positive neurons thus ensuring subtle features learning during training. Finally, the two successive 1×1 convolution layers are then added to the end of the network, after the last ConvNext block. The first convolution layer is followed by ReLU, and the second convolution layer is followed by the sigmoid activation function. The architecture of modified ConvNext is shown in Figure 2.

For a given input image x, the ConvNext feature encoder CN_f, encodes the hidden representation $v \in \mathbb{R}^{H \times W \times C}$, where $v = CN_f(x)$. In other words, ConvNext encoder extracts the salient features $CN_f(x)$ that are further utilized for prediction. $H \times W \times C$ is the shape of the extracted output features $CN_f(x)$, where the spatial dimension is $H = W = 7$ and the number of output channels C is selected from these potential values: 128, 256, 512, via cross-validation.

Prototype Layer. The prototype layer P_l of the ConvNext-PNet enables the framework to learn a set of prototypes $P = \{p_i\}_{i=0}^{n}$ with the shape of $H_1 \times W_1 \times C$, where $H_1 = W_1 = 1$. This indicates that the depth C of the prototype is similar to the output features $CN_f(x)$, however, the height and width of the prototype are smaller compared to that of output features v. Each prototype p_i in latent space learns the discriminative prototypical parts (consisting of facial regions) from each class (real/fake). In the prototype layer P_l, each prototype unit P_{li} computes the L_2 distance between the prototypical part p_i and the latent patches $\hat{v}$ (having a shape similar to p_i) of the given feature vector $v = CN_f(x)$ and then converts the distances to similarity scores. This generates the similarity scores activation map indicating the strength of the presence of the prototypical part in the image. Then the corresponding activation map is up-sampled to input image size to visualize as a heatmap, identifying the part of the image most similar to the learned

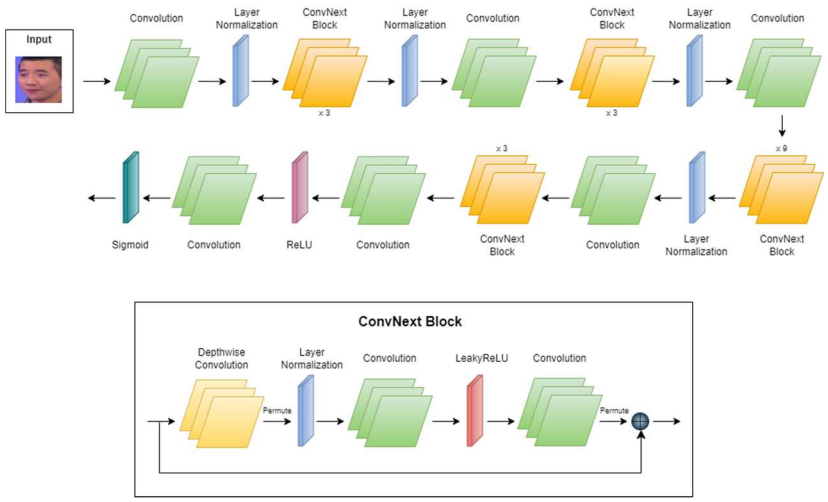

Figure 2: Architecture of modified ConvNext.

prototype. Mathematically, the prototype unit P_{li} computes the following.

$$D_{dist} = \|\hat{v} - p_i\|_2^2 \quad (1)$$

$$P_{l_i}(v) = max_{\hat{v} \in patches(v)} \log\left(\frac{D_{dist}+1}{D_{dist}+\varepsilon}\right) \quad (2)$$

where ε represents the very small positive value to avoid zero division.

The similarity scores activation map resulting from each prototype unit P_{li} is then reduced to a single similarity score utilizing the max pooling layer. The highest similarity score represents the stronger presence of the prototypical part in the patch of the input image.

It is the objective to learn the meaningful and discriminative features representing manipulation, during the training phase. Therefore, to learn appropriate latent space, the image patches $\hat{v}$ must be clustered around semantic similar prototypes P based on the L_2 distance and the cluster of distinct classes (real or fake) must be well-separated. For this purpose, ConvNext feature encoder weights ω_{CN_f}, and prototypes $P = \{p_i\}_{i=0}^{n}$ from prototype layer P_l are optimized jointly utilizing stochastic gradient descent (SGD) before the fully connected last layer F_l, while the last layer weights ω_{F_l} are kept fixed.

Consider $X = \{x_j, y_j\}_{j=1}^{N}$ be the given facial frames dataset, where x_j is the input facial frames for training, the optimization problem is given by:

$$min_{P,\omega_{CN_f}} \frac{1}{n} \sum_{j=1}^{n} CE\,(F_l \circ P_l \circ CN_f(x_j), y_j) + \lambda_{clus} C_{clus} + \lambda_{sep} S_{sep} \quad (3)$$

where $CE(.)$, $C_{clus}(.)$, and $S_{sep}(.)$ refer to the cross entropy, clustering, and separation loss, respectively, whereas λ_{clus} and λ_{sep} represents the hyperparameters.

The clustering loss promotes the minimization of the L_2 distance between the patches in the training image and the closest prototype from its class. For instance, it encourages that each fake image has some patch that is close to some or at least one prototype of the fake class and vice versa. Clustering loss is defined as:

$$C_{clus} = \frac{1}{n} \sum_{j=1}^{n} min_{i:\, p_i \in P_{y_j}} \; min_{\hat{v} \in patches\,(CN_f(x_j))} \|\hat{v} - p_i\|_2^2 \quad (4)$$

The separation loss encourages the separation of the patches of training image of a class from the prototypes of other class. For instance, it promotes that every patch of real image remains away from the prototypes of fake class and vice versa. Separation loss is defined as:

$$S_{sep} = -\frac{1}{n} \sum_{j=1}^{n} min_{i:\, p_i \notin P_{y_j}} \; min_{\hat{v} \in patches\,(CN_f(x_j))} \|\hat{v} - p_i\|_2^2 \quad (5)$$

Fully Connected Softmax Layer. Finally, the fully connected layer F_l computes the weighted sum of the similarity scores $\alpha = \omega_{F_l} P_l(v)$, where $\omega_{F_l} \in \mathbb{R}^{k \times m}$ represents the weights and $k=2$ denotes the number of classes. The softmax function is used to produce the predicted score for the given input image belonging to the classes (real and fake) as follows.

$$\hat{y_i} = \frac{e^{\alpha_i}}{\sum_{j=1}^{k} e^{\alpha_j}} \quad (6)$$

The optimization is performed on the weights ω_{F_l} of last layer F_l to introduce the sparsity property in our ConvNext-PNet framework. The sparsity property reduces the reliance of the framework on the process of negative reasoning (the facial frame belongs to class *fake* because it contains the patch that is not prototypical of class *real*). The optimization is convex as the parameters of ConvNext feature encoder and prototype layer are kept fixed. The optimization problem is as follows:

$$min_{\omega_{F_l}} \frac{1}{n} \sum_{j=1}^{n} CE\,(F_l \circ P_l \circ CN_f(x_j), y_j) + \lambda \sum_{k=1}^{K} \sum_{i:\, p_i \notin P_k} \left|\omega_{F_l}{}^{(k,i)}\right| \quad (7)$$

where $CE(.)$ represents the cross loss and λ indicates the hyperparameters.

4. Experiments and Results

This section presents the description of the datasets utilized, details of the experiments conducted, and results discussion to assess the performance of the ConvNext-PNet.

4.1. Datasets

The performance of the proposed ConvNext-PNet is evaluated utilizing the standard deepfakes detection datasets including FaceForensics++ (FF++) [20], CelebDF [21], DFDC-P [22], and DeepFakeFace (DFF) [23] dataset. FF++ dataset includes Real subset and four different manipulated subsets namely Faceswap, DeepFake, Face2Face, and NeuralTextures, each containing 1000 videos. CelebDF dataset contains face-swapped videos of celebrities including distinct ethnicity, age, and gender. This dataset contains 590 real videos corresponding to which 5639 face-swapped videos are generated. DFDC-P dataset is more challenging in the deepfakes detection domain, it includes 5000 videos of paid actors, and the manipulated videos are generated using different deepfakes generation algorithms. DFF dataset contains a total of 120K images including 30K real images and 90K fake images. Fake images are further split into 3 subsets having 30K images. Each subset comprises the fake images generated through distinct Stable Diffusion models. Specifically, the DFF dataset includes the fake images synthesized using Stable Diffusion Inpainting, Stable Diffusion v1.5, and InsightFace. Overall, the datasets used for ConvNext-PNet evaluation include the diverse deepfakes generated using different algorithms having distinct illumination conditions, viewpoints, and background settings.

4.2. Performance Evaluation of Proposed Model

To assess the effectiveness of the proposed ConvNext-PNet for the detection of deepfakes, we performed the experiments in three stages. In the first stage, ConvNext-PNet is evaluated for the faceswap deepfakes, in the second stage, performance of the model is analyzed for facial reenactment deepfakes. Whereas, in the third stage, the model is evaluated for deepfakes generated via Stable Diffusion. The performance of the proposed model is evaluated using standard evaluation matrices including accuracy and area under curve (AUC). The details and results discussion of these experiments are provided in the subsequent subsections.

Evaluation on Faceswap Deepfakes. To show the performance of the proposed ConvNext-PNet for faceswap deepfakes, we conducted experiments where we evaluated ConvNext-PNet on FaceSwap and DeepFake subsets of FF++ dataset, CelebDF, and DFDC-P datasets. For this, we trained and tested the model for the real and fake samples from the datasets. The results of the experiment are presented in Table 1.

It can be seen from Table 1 that the proposed interpretable model has shown robust performance for the detection of faceswap deepfakes in the FF++ dataset, generated using two different techniques. Specifically, the model has achieved an accuracy of 98.70% and 98.67% on the DeepFake and FaceSwap subsets of the FF++ dataset. For the CelebDF and DFDC-P datasets, the ConvNext-PNet has achieved an accuracy of 97.09% and 90.87%, respectively. CelebDF dataset includes high-quality visual manipulations with minimal flickering and color discrepancies. However, the DFDC-P dataset includes huge variations in lightning conditions, making it more difficult to detect the manipulation. The results in Table 1 indicate that the presented models can detect the faceswap visual manipulation generated using diverse algorithms having variations such as different ethnicity, illumination conditions, viewpoint, and background variations.

Evaluation on Facial Reenactment Deepfakes. To analyze the facial reenactment deepfakes detection capability of ConvNext-PNet, experiments are conducted on Face2Face and NeuralTextures subsets of the FF++ dataset. These two subsets involve facial manipulation where the source face is transferred to the target face while preserving the identity and appearance of the target face. The ConvNext-PNet is evaluated on the real and fake samples and the obtained results in terms of accuracy and AUC are shown in Table 1.

From Table 1, it is observed that the model attained an accuracy of 97.78% and 92.64% for the Face2Face and NeuralTextures subsets of the FF++ dataset. Facial reenactments generated using NeuralTextures are most difficult to detect as these involve alternation to the mouth region only. However, our proposed ConvNext-PNet detects such manipulation quite effectively with 92.64% accuracy. Overall, the results validated that the model has the potential to accurately identify facial reenactment manipulation. This indicates the better ability of the model to capture the complicated patterns that exist in the real and fake samples.

Evaluation for Diffusion Models Generated Deepfakes. To evaluate the performance of the proposed ConvNext-PNet for the deepfakes images generated using diffusion models, we utilized the DeepFakeFace dataset. We trained and tested the proposed model for the real and fake images of the subsets of DFF dataset, separately. The attained accuracy and AUC on the dataset subsets are presented in Table 1.

The results in Table 1 indicate the remarkable performance of the ConvNext-PNet for detecting deepfakes generated via Stable Diffusion models. Stable Diffusion v1.5 subset

Table 1. Performance of ConvNext-PNet for deepfakes detection.

Dataset	Accuracy (%)	AUC (%)
Faceswap Deepfakes		
DeepFake (FF++)	98.70	99.80
FaceSwap (FF++)	98.67	99.78
CelebDF	97.09	98.99
DFDC-P	90.87	93.62
Facial Reenactment Deepfakes		
Face2Face (FF++)	97.78	99.25
NeuralTextures (FF++)	92.64	95.88
Deepfakes Images Generated Via Stable Diffusion Models		
InsightFace	90.51	94.12
Stable Diffusion v1.5	98.80	99.89
Stable Diffusion Inpainting	93.75	96.78

comprises fake images that are entirely constructed from scratch (including background elements and facial attributes). The accuracy of 98.80 % on the Stable Diffusion v1.5-based deepfakes indicates the ability of the model to detect fully synthesized deepfakes images. Stable Diffusion Inpainting subset contains deepfakes images where only the facial area is synthesized while retaining the background elements. The proposed model effectively identifies the inpainted deepfakes images with an accuracy of 93.75%. Moreover, 90.51% accuracy is achieved for the deepfakes generated via InsighFace toolbox representing the ability of our model for better classification of identity-swapped fake images. Overall, the ConvNext-PNet performs remarkably in the detection of entirely synthetic deepfakes (generated using Stable Diffusion v1.5) compared to the partially synthesized deepfakes (generated using InsightFace and Stable Diffusion Inpainting).

4.3. Comparison with Existing Methods

To elaborate on the deepfakes detection performance of the proposed framework, we compared it against other existing contemporary methods [14, 15, 16, 17, 24, 25] employing the same datasets. Among the comparative methods, [14] introduced the interpretable model utilizing the graph neural network (GNN) for deepfakes detection. However, [15, 16] utilized the weighted attention mechanism module in the presented models and applied the LayerCAM technique to the different layers of introduced networks. [17] introduced the ensemble of models including standard and attention-based networks and utilized GradCAM, to demonstrate the interpretability aspect. Method [24, 25] are the transformer-based approaches for deepfakes detection. Table 2 presents the comparative results in terms of accuracy. Specifically, for the FF++ dataset, the proposed model is evaluated against the approaches [14, 15, 16, 24, 25], while it is compared with methods [14, 17, 24, 25] for the DFDC-P dataset. However, the model performance is compared with [14, 16, 17, 24] for the CelebDF dataset, where our model is the best performer among the contemporary methods with 97.09% accuracy. In the case of DFDC-P dataset, DFGNN, a graph-based neural network with complex architecture and high computational cost, is the best performer with 92.05% accuracy. However, our model performance is satisfactory on the DFDC-P dataset which is the most challenging dataset in the deepfakes detection domain. The proposed model performance is compared for each subset of the FF++ dataset. Our ConvNext-PNet attained the highest accuracy among the contemporary methods for FaceSwap, Face2Face, and NeuralTextures subsets, while for the DeepFake subset, our model is the second-best performer with 98.70% accuracy. From the comparative analysis, it is evident that our ConvNext-PNet performs remarkably well for detecting deepfakes generated utilizing distinct deepfakes generation algorithms. The results above 90.00% on all the datasets indicate the powerful feature learning capability of modified ConvNext along with the prototype learning.

4.4. Cross-corpora Evaluation

To analyze the generalizability of the ConvNext-PNet, a cross-corpora evaluation is conducted. In this experiment, the proposed model trained on one dataset is evaluated on another distinct dataset. For instance, the ConvNext-PNet trained on the FF++ dataset is evaluated on DFDC and CelebDF datasets and vice versa. The results of cross-corpora evaluation and their comparison with existing approaches [9, 14, 24] are provided in Table 3. The results depict the robustness of prototype-learning-based ConvNext model for the unseen samples of entirely different deepfakes datasets. From Table 3, it is evident that the performance of the ConvNext-PNet is degraded for cross-corpora evaluation compared to the intra-dataset

Table 2. Performance comparison against the existing contemporary methods in terms of accuracy.

Models	FF++				CelebDF (%)	DFDC-P (%)
	DeepFake (%)	FaceSwap (%)	Face2Face (%)	NeuralTextures (%)		
DFGNN [14]	98.97	98.07	62.49	75.09	93.90	92.05
MRT-Net [15]	96.70	96.76	97.67	90.25	---	---
AW-MSA [16]	98.05	97.79	97.60	91.28	96.12	---
Ensemble [17]	---	---	---	---	93.64	92.00
ViXNet [24]	89.10	66.00	78.10	84.00	94.40	86.30
CviT [25]	93.00	69.00	---	60.00	---	91.50
ConvNext-PNet	98.70	98.67	97.78	92.64	97.09	90.87

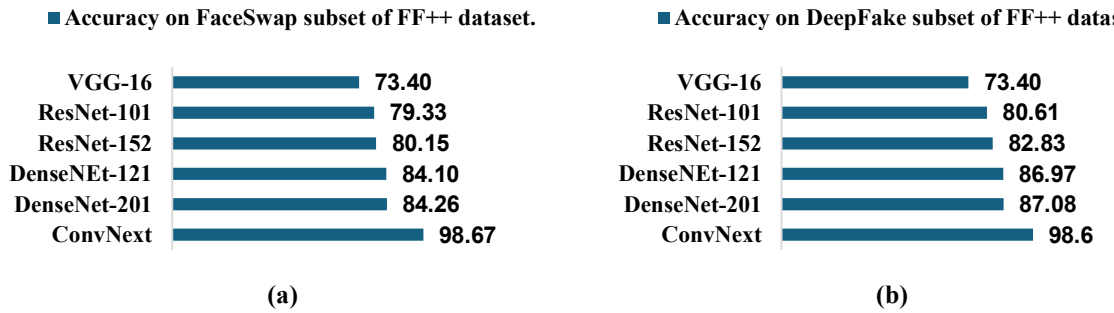

Figure 3: Ablation study for the evaluation of the base architectures.

experiment. The basic reason is the diversity present in the different datasets in terms of generation algorithms, viewpoint variations, and diverse illumination conditions. For instance, the FF++ dataset includes face swapped and face reenactment manipulation generated using four different algorithms. Likewise, DFDC comprises face-swapped deepfakes generated via various unknown face-swapping methods.

Table 3. Cross-Corpora evaluation.

Methods	Tested Dataset (Accuracy (%))		
	FF++	CelebDF	DFDC-P
Trained on FF+			
DFGNN [14]	--	73.40	71.01
ResNet-Swish-Dense54 [9]	--	70.04	81.51
ViXNet [24]	--	69.30	--
ConvNext-PNet	--	68.45	75.42
Trained on CelebDF			
DFGNN [14]	69.60	--	61.30
ViXNet [24]	68.00	--	--
ConvNext-PNet	41.28	--	60.00
Trained on DFDC-P			
DFGNN [14]	68.90	72.12	--
ResNet-Swish-Dense54 [9]	70.12	67.14	--
ConvNext-PNet	79.67	60.06	--

It is observed from Table 3 that ConvNext-PNet model trained on FF++ dataset attained reasonable performance in cross-corpora settings and achieved an accuracy of 68.45% and 75.42% on CelebDF and DFDC-P datasets, respectively. This is because the FF++ dataset is more diverse in terms of generation algorithms, therefore model trained on the FF++ dataset has greater generalization aptitude compared to the other datasets. The ConvNext-PNet trained on the CelebDF dataset attained the lowest performance when tested on FF++. The reason is that CelebDF consisted of only face-swapped deepfakes, therefore the model is unable to accurately identify face reenactment deepfakes present in the FF++ dataset. Overall, for the cross-corpora evaluation, ConvNext-PNet has attained satisfactory results, however, these results should be further improved, compared to the existing models.

4.5. Ablation Study

We conducted an ablation study experiment to analyze the impact of different base architectures in the proposed prototype-based framework, The main goal of this study is to assess the performance of the latest ConvNext in the prototype-based framework for the task of deepfakes detection against other existing deep learning models including VGG, ResNet, and DenseNet. Specifically, we compared the performance of DenseNet-121, DenseNet-201, ResNet-101, ResNet-152, VGG-16, and ConvNext as the base architecture of the prototypical part network. This experiment is conducted utilizing the faceswap deepfakes (DeepFake and FaceSwap subsets) from the FF++ dataset. The results in terms of accuracy are shown in Figure 3.

From Figure 3, it can be clearly observed that ConvNext outperforms the other comparative models for the detection of deepfakes. The lowest performance is reported by the VGG-16 with an accuracy of 73.98% and 73.40% on DeepFake and FaceSwap subsets of the FF++ dataset, respectively. DenseNet-201with an accuracy of 87.08% on the DeepFake subset and 84.26% on the FaceSwap subset, is the second-best performer. ConvNext has attained the best performance with 98.00% accuracy on both DeepFake and FaceSwap subsets. This is mainly due to the modernized architecture of ConvNext having a resemblance to the Transformers. ConvNext has the simplicity of CNN architectures and design resemblance with the Swin Transformer without having modules like shifted windows and thus leads to better performance. On the other hand, models like VGG, ResNet, and DenseNet have the standard CNNs architectural design leading to comparatively low performance. So, the results indicate that the introduction of ConvNext as base architecture to the proposed prototypical-based framework is more robust for the detection of visual manipulation compared to the other deep learning models. The reliable performance of ConvNext is due to the model's potential to learn the distinct features. This eventually leads to the better ability of the model to deal with the transformation changes involved in the visual manipulation.

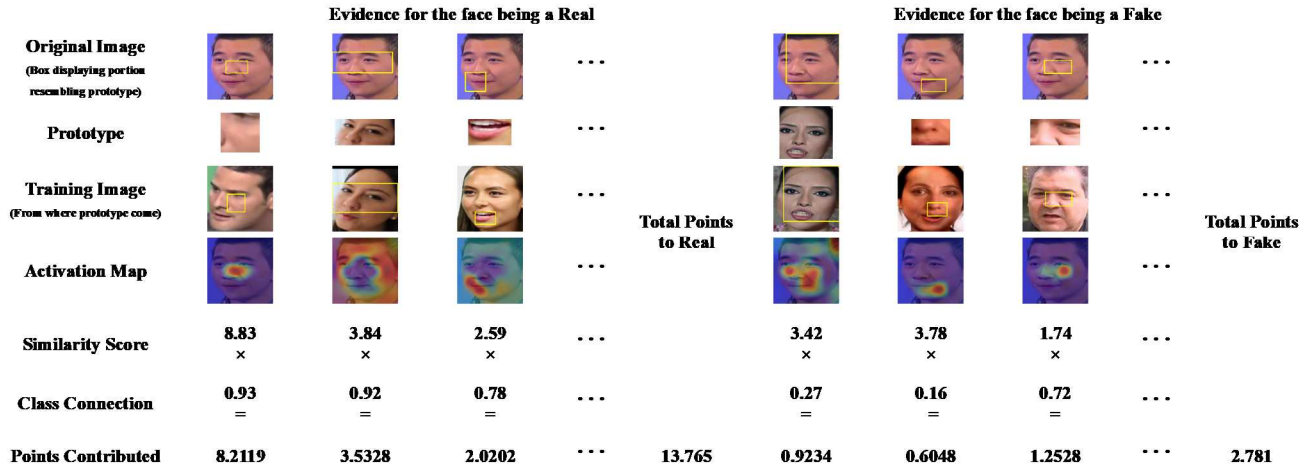

Figure 4: Reasoning process of ConvNext-PNet.

5. Explainability of ConvNext-PNet

In this section, it is depicted how the ConvNext-PNet reaches the prediction decision by explaining the reasoning process of the model shown in Figure 4. The goal is to highlight the interpretability and explainability aspects of the proposed ConvNext-PNet. Consider a test image x, for which the model captures the latent features $v = CN_f(x)$. The model has already learned the prototypes p_i of the classes (real and fake). The patch representations $\hat{v}$ of the latent features are compared against the learned prototypes p_i of each class to find the evidence that the input image x belongs to that class.

Consider Figure 4, where our ConvNext-PNet tries to find the proof that the input facial image belongs to a fake or real class. For this, the model compares the patch representations of latent features of the input image with every learned prototype of the fake and real class. The similarity score against each prototype is computed which is then up-sampled and overlaid on the input image. This produces the activation map indicating the part of the input image activated by the respective prototype. For instance, the left side of Figure 4 represents the network's ability to find evidence for the real class by comparing image patches with every prototype of that class. The resultant similarity score map toward each prototype was superimposed on the given input image to highlight the part activated by each prototype (shown in the Activation Map row). Additionally, the bounding box on the input image (shown in the Original Image row) represents the most activated part of the input image for each prototype. This indicates that our model considers that the image patch looks like the corresponding prototype. The similarity score between the learned prototype and image patch is shown in the Similarity Score row. The class Connection row indicates the degree to which a specific prototype is associated with a particular class. A similarity score is multiplied with a class connection to obtain the points contributed. These contributed points are summed up to find the final similarity score (representing total points to a specific class). The highest final similarity score for the class represents that the input image belongs to that class. In our case, the final similarity scores for the real and fake classes are 13.765 and 2.781, respectively. This indicates that the input image belongs to the real class.

6. Conclusion

This paper has presented a prototype-based learning framework namely ConvNext-PNet for the interpretable and explainable detection of visual deepfakes. Precisely, in the proposed framework, prototype-based learning is incorporated into the modified ConvNext model. Performance of the ConvNext-PNet on the FF++, CelebDF, DFDC-P, and DeepFakeFace datasets highlights the effectiveness of the prototype-based learning model for the identification of visual deepfakes generated using distinct approaches. The ablation study outcome also signifies the effectiveness of ConvNext incorporated with the prototype learning framework for deepfakes detection. Overall, the results emphasize the significance of explainable models for deepfakes detection to increase the trustworthiness of the model prediction. In the future, we plan to further investigate the performance of the prototype-based learning framework for deepfakes detection under the occurrence of post-processing attacks (i.e., size transformation, blurring, and noise) and adversarial attacks (i.e., FGSM and PGD). We also intended to extend the implementation of interpretable models for the audio-visual deepfake detection task along with improving the interpretability and explainability aspects of such models.

References

[1] Pei, Gan, Jiangning Zhang, Menghan Hu, Guangtao Zhai, Chengjie Wang, Zhenyu Zhang, Jian Yang, Chunhua Shen,

and Dacheng Tao. "Deepfake generation and detection: A benchmark and survey." *arXiv preprint arXiv:2403.17881*, 2024.

[2] Han, Xintong, Vlad Morariu, and Peng IS Larry Davis. "Two-stream neural networks for tampered face detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 19-27, 2017.

[3] He, Peisong, Haoliang Li, and Hongxia Wang. "Detection of fake images via the ensemble of deep representations from multi color spaces." In *2019 IEEE international conference on image processing (ICIP)*, pp. 2299-2303. IEEE, 2019.

[4] Sandhya, and Abhishek Kashyap. "A Light Weight Depthwise Separable Layer Optimized CNN Architecture for Object-Based Forgery Detection in Surveillance Videos." *The Computer Journal* (2024): bxae005.

[5] Nguyen, Dat, Nesryne Mejri, Inder Pal Singh, Polina Kuleshova, Marcella Astrid, Anis Kacem, Enjie Ghorbel, and Djamila Aouada. "LAA-Net: Localized Artifact Attention Network for High-Quality Deepfakes Detection." *arXiv preprint arXiv:2401.13856*, 2024.

[6] Ilyas, Hafsa, Ali Javed, and Khalid Mahmood Malik. "AVFakeNet: A unified end-to-end Dense Swin Transformer deep learning model for audio–visual deepfakes detection." *Applied Soft Computing* 136, 110124, 2023.

[7] Speith, Timo. "A review of taxonomies of explainable artificial intelligence (XAI) methods." In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pp. 2239-2250, 2022.

[8] Xu, Ying, Kiran Raja, and Marius Pedersen. "Supervised contrastive learning for generalizable and explainable deepfakes detection." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 379-389. 2022.

[9] Nawaz, Marriam, Ali Javed, and Aun Irtaza. "ResNet-Swish-Dense54: a deep learning approach for deepfakes detection." *The Visual Computer* 39, no. 12, 6323-6344, 2023.

[10] Xu, Ying, Kiran Raja, Luisa Verdoliva, and Marius Pedersen. "Learning pairwise interaction for generalizable deepfake detection." In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 672-682, 2023.

[11] Chen, Chaofan, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. "This looks like that: deep learning for interpretable image recognition." *Advances in neural information processing systems* 32, 2019.

[12] Hase, Peter, Chaofan Chen, Oscar Li, and Cynthia Rudin. "Interpretable image recognition with hierarchical prototypes." In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 7, pp. 32-40, 2019.

[13] Nauta, Meike, Annemarie Jutte, Jesper Provoost, and Christin Seifert. "This looks like that, because... explaining prototypes for interpretable image recognition." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 441-456. Cham: Springer International Publishing, 2021.

[14] Khalid, Fatima, Ali Javed, Hafsa Ilyas, and Aun Irtaza. "DFGNN: An interpretable and generalized graph neural network for deepfakes detection." *Expert Systems with Applications* 222, 119843, 2023.

[15] Yadav, Ankit, and Dinesh Kumar Vishwakarma. "AW-MSA: Adaptively weighted multi-scale attentional features for DeepFake detection." *Engineering Applications of Artificial Intelligence* 127, 107443, 2024.

[16] Yadav, Ankit, and Dinesh Kumar Vishwakarma. "MRT-Net: Auto-adaptive weighting of manipulation residuals and texture clues for face manipulation detection." *Expert Systems with Applications* 232, 120898, 2023.

[17] Silva, Samuel Henrique, Mazal Bethany, Alexis Megan Votto, Ian Henry Scarff, Nicole Beebe, and Peyman Najafirad. "Deepfake forensics analysis: An explainable hierarchical ensemble of weakly supervised models." *Forensic Science International: Synergy* 4, 100217, 2022.

[18] Xiang, Jia, and Gengming Zhu. "Joint face detection and facial expression recognition with MTCNN." In *2017 4th international conference on information science and control engineering (ICISCE)*, pp. 424-427. IEEE, 2017.

[19] Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. "A convnet for the 2020s." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976-11986, 2022.

[20] Rossler, Andreas, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. "Faceforensics++: Learning to detect manipulated facial images." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1-11, 2019.

[21] Li, Yuezun, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. "Celeb-df: A large-scale challenging dataset for deepfake forensics." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3207-3216, 2020.

[22] Dolhansky, Brian, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. "The deepfake detection challenge (dfdc) preview dataset." *arXiv preprint arXiv:1910.08854*, 2019.

[23] Song, Haixu, Shiyu Huang, Yinpeng Dong, and Wei-Wei Tu. "Robustness and generalizability of deepfake detection: A study with diffusion models." *arXiv preprint arXiv:2309.02218*, 2023.

[24] Ganguly, Shreyan, Aditya Ganguly, Sk Mohiuddin, Samir Malakar, and Ram Sarkar. "ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection." *Expert Systems with Applications* 210,118423, 2022.

[25] Wodajo, Deressa, and Solomon Atnafu. "Deepfake video detection using convolutional vision transformer." *arXiv preprint arXiv:2102.11126*, 2021.