EXPLAINABLE DEEPFAKE DETECTION WITH DEEP LEARNING AND LIME

Priyanka Rose Varghese

Columbia University

ABSTRACT

Advancements in deep learning, particularly in computer vision, have led to many technological improvements. However, one downside is the increasing issue of deepfakes, where videos, images, and audio can be manipulated to look real but are actually artificially generated. There is an increasing demand for systems that can detect manipulated images. One of the most effective methods is to train deep learning-based models designed to identify artificially modified media. In this paper, we train three models—VIT-B/16, ConvNeXt Tiny, and a custom CNN network—on OpenForensics, a large dataset containing both real and deepfake images. We achieve an accuracy of 0.91 with the custom CNN network on unseen data. Finally, we use LIME to interpret the model's predictions, addressing the 'black-box' nature of deep learning networks.

***Index Terms*—** Deepfake, Vision Transformers, Deep Learning, ConvNexts, LIME

1. INTRODUCTION

Deepfakes are synthetic media (images, videos, or audio) generated by AI techniques[1]. One of the most common methods for generating deepfakes is using GANs, or Generative Adversarial Networks. Due to the inner workings of GANs, the images they generate appear very realistic and can sometimes be mistaken for real images.

The widespread use of AI and image generators has led to an increase in deepfakes on the web, raising concerns about their social impact. Thus, it has become a need for sophisticated artificially generated image detectors that can differentiate and identify modified images. Deep Learning has proven to be extremely successful in the field of image classification. Its ability to capture features that are otherwise undiscernible has been beneficial in several domains.

In this paper, we train three deep learning models to detect deepfake images: VIT-B/16, ConvNeXt Tiny, and a custom CNN network. Both VIT-B/16 and ConvNeXt Tiny are pretrained models with weights that can be used through transfer learning.

We also aim to address the 'black-box' nature of deep learning models using LIME (Local Interpretable Model-agnostic Explanations). LIME helps to "explain" the model's predictions by highlighting important regions of an image that influenced the outcome. These explanations can be visualized, providing a clearer understanding of the areas on the image that contributed to the model's prediction.

The remainder of the paper is organized as follows. Section I provides the Introduction to the project. Section II presents a literature review of existing systems. Section III provides the Proposed Methodology. Section IV presents the experiments conducted. Section V presents the results and the analysis of the proposed system. Section VI concludes the work.

2. LITERATURE SURVEY

Advancements in Deep Learning in recent years have given rise to highly efficient models that can efficiently detect anomalies in the images. Several studies for DeepFake detection with Deep Learning have been conducted in the past and this section discusses some of these.

Darius et al. [2] proposed two shallow network architectures—Meso4 and MesoInception-4—to detect artificially generated image frames from videos. Meso4 is a CNN architecture consisting of four convolutional and pooling blocks, followed by a dense fully connected network. MesoInception-4 builds upon the original Meso4 architecture by appending an Inception Module. These models were trained on the DeepFake and Face2Face datasets, achieving an accuracy of 98% on the DeepFake test data and 95% on Face2Face.

Another study [3] utilizes deep learning architectures such as DenseNet121, ResNet18, SqueezeNet, and VGG11 for classifying DeepFakes. These models were trained and evaluated on the OpenForensics dataset. With a batch size of 32, the VGG11 model performed the best, achieving an accuracy score of 94.46%, while DenseNet121 achieved an accuracy of 93.89%. It also utilized GradCAM to interpret the model's predictions.

Nirkin et al. [4] proposed an Xception architecture-based network that works in 2 phases. The first phase deals with the identification and segmentation of the face, and the second phase has a context recognition network. The network is trained on the VGGFace2 dataset [5].

This study [6] proposes a novel Hierarchical Memory

Fig. 1. Overview of the proposed methodology

Network Module that works even on unseen image manipulation techniques. It achieves an accuracy of 84.12% on unseen data in the DeepFake dataset and 86.53% on the Face2Face dataset.

One study [7] evaluates the performance of capsule networks for digital forgery detection in images. Their architecture consists of the initial layers of a VGG19 for extracting features, which are the inputs to the capsule network. A novel autoencoder architecture is introduced in [8]. The encoder is trained to learn separate encodings for fake and real images.

In summary, DeepFake detection has been a focus of concern and study, and deep learning models have demonstrated immense potential and success in the detection.

3. PROPOSED METHODOLOGY

3.1. Overview

This section outlines the methods utilized for this project, which can be divided into the following phases: Data Preprocessing, Model Selection & Training, and Segmentation of Fake Regions Using LIME. Fig. 1 highlights the aforementioned steps in a flow chart. Each of these steps is explained in detail below.

3.2. Data Preprocessing

Data preprocessing is an essential step in any data-modeling problem including image classification. In this project, each image in the dataset is fed into a preprocessing pipeline defined using Pytorch transforms.

The data is augmented by the following steps: randomly cropping a region and resizing it to a fixed size (224x224), randomly flipping the image horizontally, and adjusting the color, saturation, hue, and contrast. Data augmentation supplements the existing dataset with variations that help improve the model's performance on unseen data

The images are processed into Pytorch tensors and are normalized. Normalization centers the pixel values to 0 and improves the speed of convergence while training [9].

3.3. Model Selection & Training

For this Project, three models where selected and trained - VIT-B/16[10], ConvNeXt Tiny[11] and a custom CNN architecture.

The original weights of ViT-B/16 and ConvNeXt Tiny were retained, and the last layer was adapted to fit our binary classification problem. All the weights, except those of the final layer, were frozen, and the weights of the final layer were computed using backpropagation. The ConvNeXt Tiny model was further fine-tuned to improve its performance.

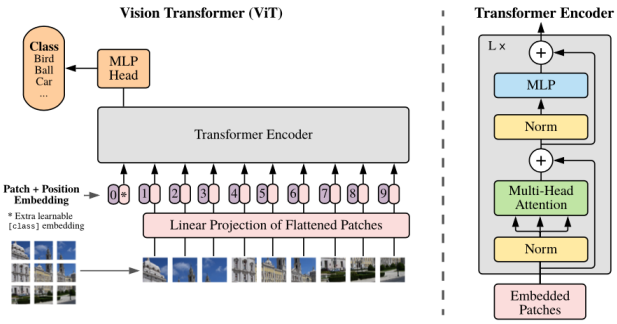

Fig. 2. The architecture of Vision Transformers[10]

The final model, a custom CNN architecture, was trained from scratch.

3.3.1. VIT-B/16

VIT-B/16[10] is a model that belongs to the family of Vision Transformers. It is a pretrained model trained on the ImageNet21k [12] data and fine-tuned on the ImageNet 2012 dataset[13]. The 'B' refers to 'Base' which indicates that the model is medium-sized compared to the 'H' and 'L' versions.

In Vision Transformers, image classification tasks are treated as a language task. An image is divided into patches, which are then fed into the transformer sequentially. These patches are first projected into a lower-dimensional vector space. A learned positional embedding is added to the projected form to help the model understand the spatial relationships between two patches. This is then sequentially fed into the transformer encoder architecture[14], which extracts features from the image. These features are then passed to the Multi-Linear Perceptron Head for classification.

Vision Transformers are known to perform exceptionally well, often outperforming popular CNN architectures like Resnets[15]. This model was selected for the purpose of this project because it excels on large datasets and retains more spatial information than CNN architectures like ResNets[16]. The architecture of Vision Transformers is given in Fig. 2.

3.3.2. ConvNeXt Tiny

ConvNeXt Tiny[11] is a model from the ConvNeXt family, which consists of pure CNN-based networks known to perform as well as Vision Transformers. ConvNeXts are trained on the ImageNet-1K dataset, pre-trained on the ImageNet-22K, and are fine-tuned again on the ImageNet-1K dataset.

ConvNeXt attempts to refine pure CNN blocks (as the ones seen in Resnets) to improve performance and reduce computation. The comparison of block designs of Swin

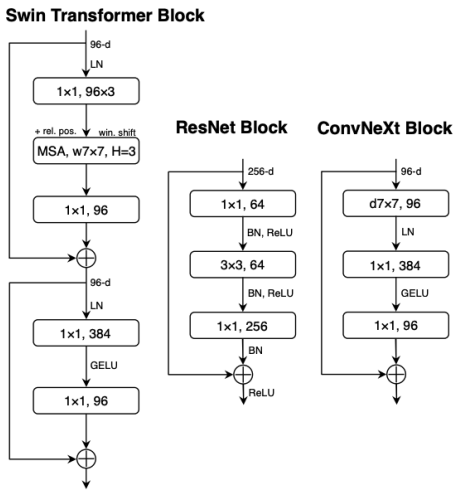

Fig. 3. Comparision of block designs of Swin transformers, ConvNeXt and Resnets[11]

Transformers, Resnets and ConvNeXt can be seen in Fig.3. This network uses GELU[17] and layer normalization to reduce the computation required. The term Tiny means that has less trainable parameters leading to faster convergence time.

3.3.3. Custom CNN arcitecture

The last model is a Custom CNN network consisting of a feature extractor and a classifier. The feature extractor is responsible for extracting spatial features from the input data and the classifier analyzes those features to classify the image as 'real' or 'fake'.

The feature extractor consists of 5 blocks. The first block contains a 2D Convolutional Layer with a 3x3 kernel, a ReLU activation function, followed by a MaxPooling Layer that reduces the dimensionality by half. The second block includes a 2D Convolutional Layer with a 3x3 kernel, a ReLU activation function, and a MaxPooling Layer that again reduces the dimensionality by half (i.e., from 112x112 to 56x56). The third and fourth blocks follow the same structure, each with one 2D Convolutional Layer, a ReLU activation function, and a MaxPooling Layer. After these four blocks, the spatial size is reduced to 14x14. The fifth block consists of a 2D Convolutional Layer with a 3x3 kernel, a ReLU activation function, and an Adaptive Average Pooling Layer. This layer ensures a fixed output size of 7x7, regardless of the input dimensions.

The classifier consists of a flattening layer and a fully connected layer with 4096 hidden neurons. This layer uses a ReLU activation function and a dropout layer to prevent overfitting. The final layer is the output layer, which has a sigmoid activation function that maps to a single output. Fig. 4 shows the block diagram of the above architecture.

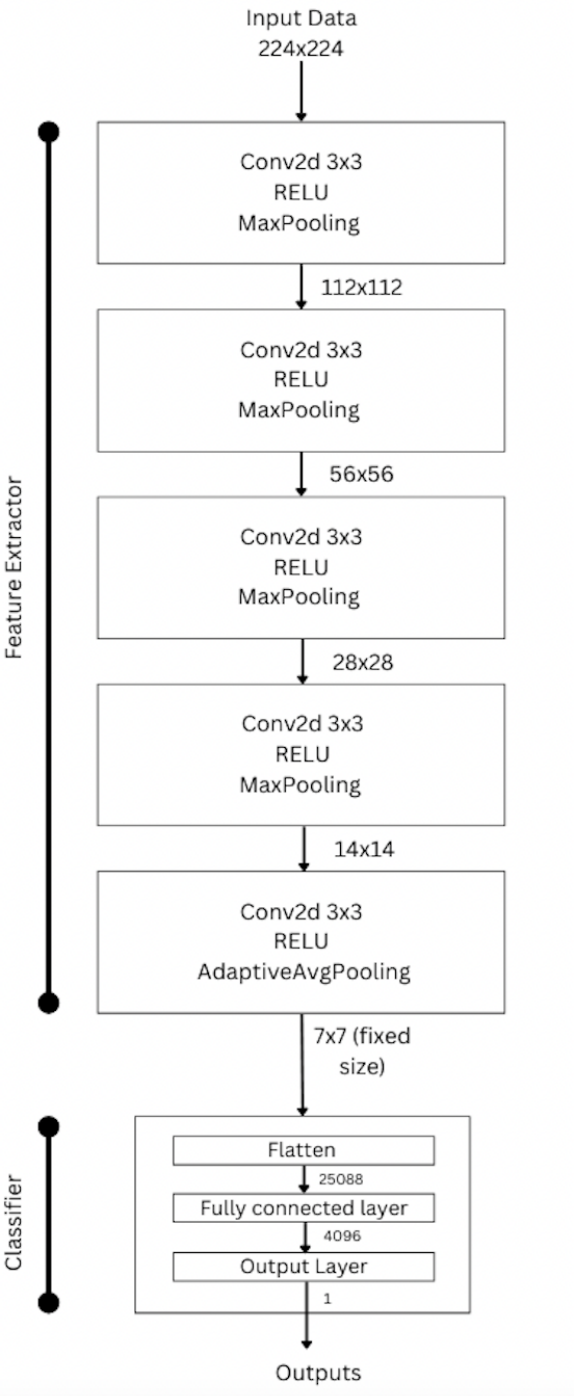

Fig. 4. Architecture of the Custom CNN model

3.4. Segmentation of Fake Regions Using LIME

The final step involves highlighting the "fake" regions that the model identifies. This segmentation brings interpretability to the otherwise black-box model. This project uses LIME [18] or Local Interpretable Model-agnostic Explanation.

LIME works by splitting the image into superpixels (groups of pixels with similar brightness or intensities), which are usually contiguous regions in the image. These superpixels are replaced by the mean color of the pixels (mean replacement). For each generated example, the superpixels are randomly switched 'on' or 'off' based on a set of random variables. LIME then generates a linear model using these examples, with the superpixels serving as interpretable input features and the model's predictions as the output[19].

In this project, images are fed into each model, and a LIME explanation is generated for the top classification label. This explanation is then visualized by overlaying a mask on the image, which highlights the important areas that influenced the LIME explanation.

4. EXPERIMENTS

This section provides a detailed description of the dataset, the evaluation metrics, and the overall experimental setup implemented for the project.

4.1. Dataset Description

The training data for the models in this project is sourced from the Open Forensics dataset[20]. It is one of the largest detection and segmentation datasets [20]. Compared to other datasets, which often contain repeated backgrounds, OpenForensics includes a wide array of images with distinct backgrounds and diverse scenes. This makes it more challenging to train, but the resulting model would be able to generalize well and detect deepfakes in unseen data. This dataset also includes perturbed images to simulate real-world scenarios.

A version of this dataset, organized into train, validation, and test directories, is available on Kaggle and was used as the source for training this model. The training data contains 70,000 real images and 70,000 fake images, the validation directory contains 19,800 real images and 19,600 fake images, and the test directory contains 5,413 real images and 5,492 fake images. The dataset distribution is shown in the table below.

Table 1. Dataset Distribution by Split and Category

Split	Real Images	Fake Images	Total
Train	70,001	70,001	140,002
Validation	19,800	19,600	39,400
Test	5,413	5,492	10,905

4.2. Experimental Setup

The last layer of the Vision Transformer model (VIT-B/16) was replaced with a fully connected layer containing a sigmoid activation function. After freezing all the previous weights, these two layers were trained for 5 epochs with a learning rate of 0.01. Similarly, the replaced layers of the ConvNeXt Tiny model were trained for 5 epochs with a learning rate of 0.001 and a weight decay of 1e-4 to prevent overfitting. The entire model was then fine-tuned for an additional 3 epochs. The final custom CNN model was also trained for 5 epochs with a learning rate of 0.001. Since it is a binary classification problem, Binary Cross-Entropy was used as the loss function, with Adam as the optimizer.

All the models were trained on Google Colab with GPU A100 for accelerated training.

4.3. Evaluation Metrics

Accuracy, precision, recall and F1 score are the metrics used to evaluate the model's performance. Accuracy, precision, and recall are three of the most commonly used performance metrics for medical imaging analysis[21].

Accuracy is simply a measure of how well our predicted values or events in the future match real cases. It can be represented as ,

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

Precision is defined as the ratio of true positive cases (actual positives) to the total number of predicted positives by the classification model. Mathematically, it can be expressed as,

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

The proportion of true positive predictions, out of all the positive instances in the dataset, is called recall. This can be expressed as,

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

The F1 score is the harmonic mean of precision and recall, hence it produces a single score which takes into account both these metrics. The F1 score can be calculated as,

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

5. RESULTS

This section discusses the results and findings of the project.

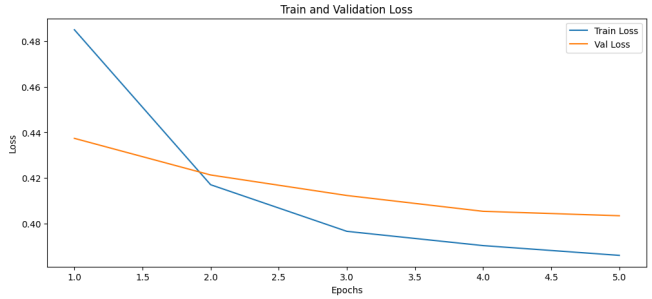

Fig. 5. Training and Validation Loss Curve (VIT-B/16)

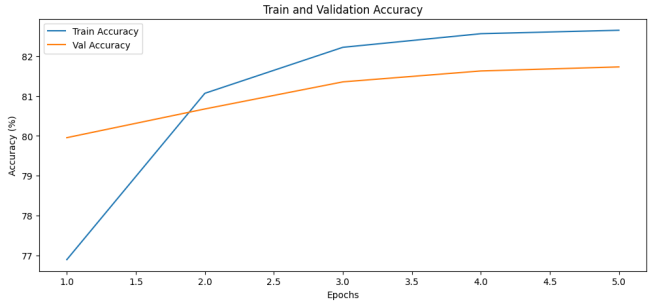

Fig. 6. Training and Validation Accuracy Curve (VIT-B/16)

5.1. Analysis

In this project, three models are trained and evaluated with four evaluation metrics mentioned in the previous sections. The performance of each model and its analysis is detailed below.

5.1.1. Results of VIT-B/16

The VIT-B/16 model was trained on the training set for 5 epochs and validated using the validation dataset. At the end of 5 epochs, the model achieved a training accuracy of 85.65% and a validation accuracy of 81.73%. Training the model for more than 5 epochs resulted in overfitting, which led to a reduction in validation accuracy. Therefore, 5 epochs were selected as the standard to prevent overfitting.

The training and validation loss and accuracy versus epochs curves are shown in Fig. 5-6.

The classification report (Fig. 7) shows that the model predicts the fake class with a precision of 0.72 and a recall of 0.83, while the real class is predicted with a precision of 0.80 and a recall of 0.67. The overall accuracy is 0.75. This suggests that the model is better at correctly identifying the fake class (indicated by the higher recall).

The confusion matrix is visualized in Fig. 8.

Analysis: ViT-B/16 is a highly complex network. It was selected for this study due to its effectiveness in extracting subtle features, which are characteristic of deepfake images. Additionally, it performs well on large datasets. The training

	precision	recall	f1-score	support
Fake	0.72	0.83	0.77	5492
Real	0.80	0.67	0.73	5413
accuracy			0.75	10905
macro avg	0.76	0.75	0.75	10905
weighted avg	0.76	0.75	0.75	10905

Fig. 7. Classification Report(VIT-B/16)

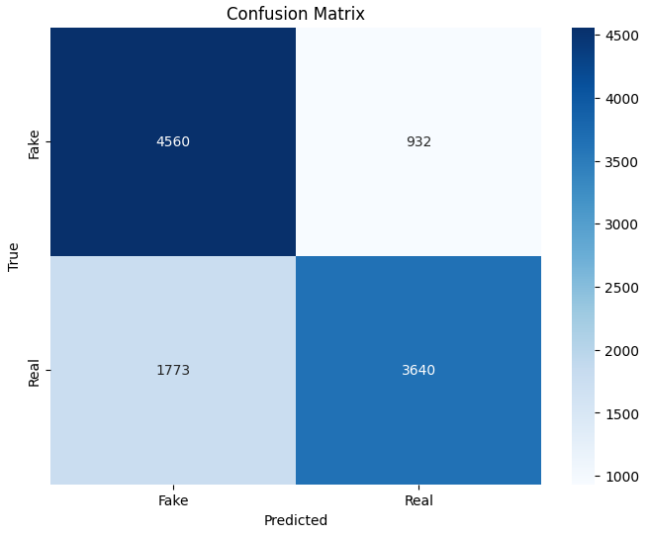

Fig. 8. Confusion Matrix (VIT-B/16)

Fig. 9. LIME explanations (VIT-B/16)

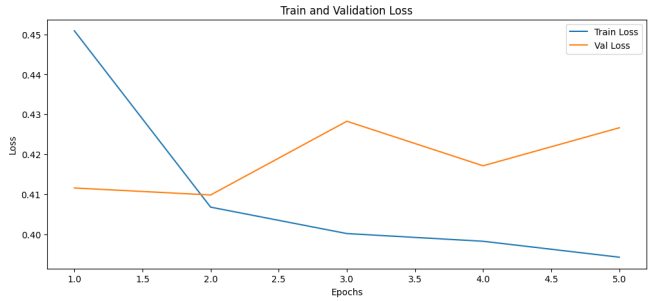

Fig. 10. Training and Validation Loss Curve (ConvNeXt Tiny)

dataset used in this study contained over 100k images. However, the model overfitted on training data due to its overly complex architecture, resulting in only 75% accuracy on unseen data.

Using LIME, the model's prediction can be interpreted. Using the mechanism mentioned in the previous sections, we can map the regions that influence a model's decision. The regions marked red are those that contribute strongly to the model's classification. The cool-colored regions contribute weakly to the model's prediction. In deepfake images, the eyes and mouth typically show strong signs of modification. VIT-B/16 (shown in Fig. 9) successfully identifies the eyes and a very small part of the mouth as reliable indicators of image manipulation.

5.1.2. Results of ConvNeXt Tiny

The pre-trained weights are frozen except for the last fully connected layer, and the model is trained on the training data for 5 epochs. The training accuracy is 82.08% and the validation accuracy is 80.33%. The training and validation loss and accuracy curves (shown in Fig. 10-11) show that the model starts overfitting at the end of the 5 epochs.

This model was also fine-tuned for another 2 epochs, by unfreezing all the layers. This improved the validation accuracy to 81.72%.

The classification report (Fig. 12) yields results similar to

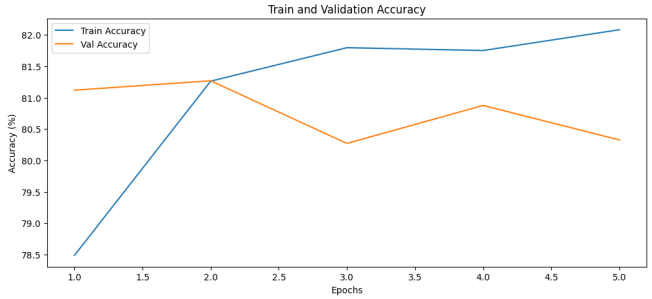

Fig. 11. Training and Validation Accuracy Curve (ConvNeXt Tiny)

	precision	recall	f1-score	support
Fake	0.73	0.80	0.77	5492
Real	0.78	0.70	0.74	5413
accuracy			0.75	10905
macro avg	0.76	0.75	0.75	10905
weighted avg	0.76	0.75	0.75	10905

Fig. 12. Classification Report(ConvNeXt Tiny)

the previous model. For the fake class, the precision is 0.73 and the recall is 0.80. For the real class, the precision is 0.78 and the recall is 0.70. This indicates that the model is better at detecting fake instances than real instances.

The confusion matrix is visualized in Fig. 13.

Analysis: ConvNeXt models are known to be on par with benchmark Vision Transformers while being less complex. In this case, the ConvNeXt model achieves the same accuracy score (75%) as the previous Vision Transformer model. Although this network is slightly less deep and complex than ViT-B/16, it still appears to be overly complex for this problem. The model overfits after 5 epochs without any improvement in validation accuracy.

The model's decisions can be interpreted using LIME. As seen in Fig. 14, the model almost entirely focuses on the areas where modifications are clearly visible. A red mask indicates the region that contributes very strongly to the prediction.

5.1.3. Results of Custom CNN model

This is a custom network trained from scratch. It is trained on the training data for 5 epochs. After 5 epochs, the model starts overfitting, indicated by the drop in validation accuracy. At the end of 5 epochs, the training accuracy is 95.74% and the validation accuracy is 92.88%.

The training and validation loss and accuracy versus epochs curves are shown in Fig. 15-16.

The classification report (Fig. 17) indicates that this is a well-performing model. It predicts the fake class with a

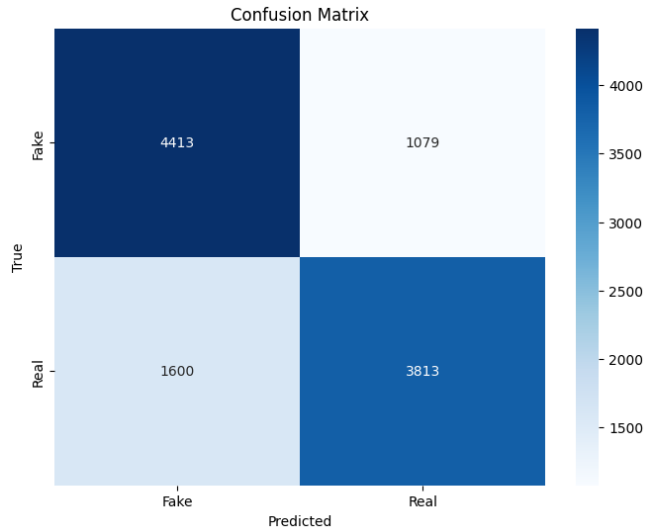

Fig. 13. Confusion Matrix(ConvNeXt Tiny)

Fig. 14. LIME explanations (ConvNeXt Tiny)

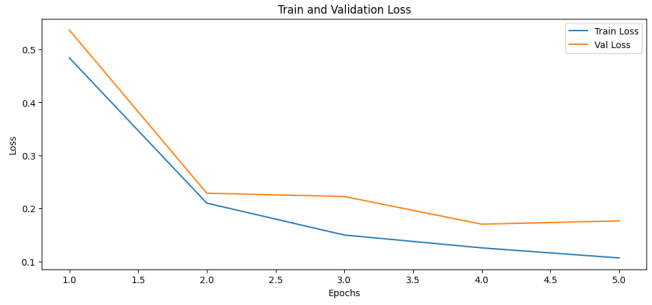

Fig. 15. Training and Validation Loss Curve (Custom Model)

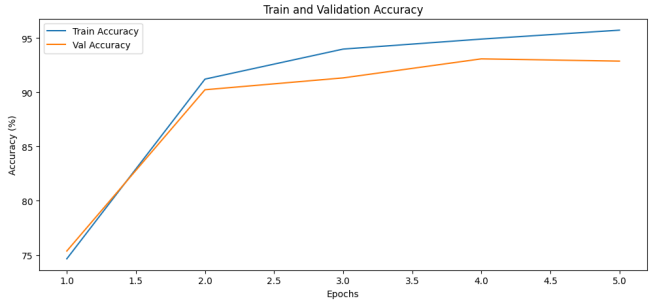

Fig. 16. Training and Validation Accuracy Curve (Custom Model)

	precision	recall	f1-score	support
Fake	0.90	0.92	0.91	5492
Real	0.92	0.89	0.91	5413
accuracy			0.91	10905
macro avg	0.91	0.91	0.91	10905
weighted avg	0.91	0.91	0.91	10905

Fig. 17. Classification Report(Custom Model)

precision of 0.90 and a recall of 0.92. It predicts the 'Real' class with a precision of 0.92 and a recall of 0.89. The overall accuracy on the test data is 91%.

The confusion matrix can be visualized in Fig. 18.

Analysis: This is the best-performing model among those trained and tested on unseen data. One of the reasons for its performance can be attributed to the network architecture. Its simplicity, compared to the other models, likely helped prevent overfitting. The network also had Dropout layers which contributed to the model not overfitting on the training data. However, it wasn't overly simple, allowing the model to capture the subtle differences and details in the images. All these reasons combined helped the model perform way better than the other network architectures.

Using LIME (Fig. 19), we can map the important regions that lead to a particular prediction. This is a 'Fake' image sample from the test dataset. LIME highlights most of the facial regions as contributing the most to the prediction, with red indicating strong indicators. The model appears to have captured even subtle features that suggest modification. This is clearly superior and more sophisticated compared to the regions marked by ViT-B/16, thus justifying its high accuracy.

5.2. Comparision with Prior Work

There has been prior work on the topic of deepfake detection. Table 2 shows some of the best models trained on various datasets.

Table 2. Comparision of Models

Author	*Dataset*	*Algorithms*	*Accuracy(in %)*	*Precision*	*Recall*	*F1 Score*
Nirkin et.al[22]	Celeb-DF-v2	XceptionNet	66.0%			
	FF-DF	XceptionNet	99.7%	-	-	-
Badale, et.al. [23]	Custom Dataset	Dense-CNN	91%	-	-	-
Saealal et.al[24]	OpenForensics	VGG11	94.46%	-	-	-
		DenseNet 121	93.89%	-	-	-
Proposed Model	OpenForensics	VIT-B/16	75.0%	0.75	0.76	0.75
Proposed Model		ConvNeXt Tiny	75.0%	0.76	0.75	0.75
Proposed Model		Custom CNN	91.0%	0.91	0.91	0.91

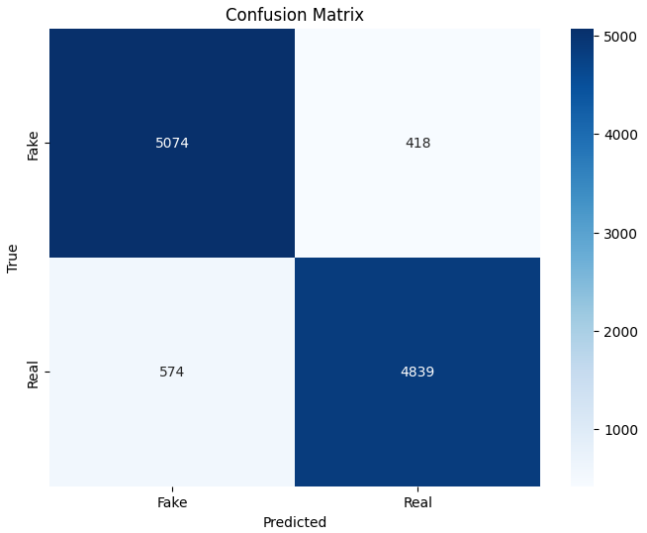

Fig. 18. Confusion Matrix(Custom Model)

Fig. 19. Lime Explanation (Custom Model)

6. CONCLUSION

DeepFakes are a growing alarm. With the rise of sophisticated image generators, it is crucial to have good detection systems in place. This project aims to build a deepfake detection system capable of accurately identifying artificially generated images. For training the models, the large open-sourced OpenForensics dataset was utilized. Three models were trained and evaluated: ViT-B/16, ConvNeXt Tiny, and a custom CNN network. The predictions on the test dataset were interpreted using LIME, providing an idea as to whether the trained model successfully identifies distortions and subtle modifications in the images.

7. REFERENCES

This section contains the list of publications and articles referred for the sake of this project

8. REFERENCES

[1] Wikipedia Contibutors, "Deepfake," Accessed from https://en.wikipedia.org/wiki/Deepfake on 2024-11-05.

[2] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," 12 2018, pp. 1–7.

[3] Muhammad Salihin Saealal, Mohd Zamri Ibrahim, Mohd Ibrahim Shapiai, and Norasyikin Fadilah, "In-the-wild deepfake detection using adaptable cnn models with visual class activation mapping for improved accuracy," in *2023 5th International Conference on Computer Communication and the Internet (ICCCI)*, 2023, pp. 9–14.

[4] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner, "Deepfake detection based on the discrepancy between the face and its context," 2020.

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, 2018, pp. 67–74.

[6] Tharindu Fernando, Clinton Fookes, Simon Denman, and Sridha Sridharan, "Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks," 2019.

[7] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," 2018.

[8] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva, "Forensictransfer: Weakly-supervised domain adaptation for forgery detection," 2019.

[9] Xinyi Chen, Xiang Liu, Yuke Wu, Zhenglei Wang, and Shuo Hong Wang, "Research related to the diagnosis of prostate cancer based on machine learning medical images: A review," *International Journal of Medical Informatics*, vol. 181, pp. 105279, 2024.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[11] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, "A convnet for the 2020s," 2022.

[12] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor, "Imagenet-21k pretraining for the masses," 2021.

[13] ImageNet Large Scale Visual Recognition Challenge, "Imagenet large scale visual recognition challenge 2012," https://www.image-net.org/challenges/LSVRC/2012/, 2012.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," 2023.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.

[16] V7 Labs, "Vision transformer: What it is how it works [2024 guide]," https://www.v7labs.com/blog/vision-transformer-guide, 2022.

[17] Dan Hendrycks and Kevin Gimpel, "Gaussian error linear units (gelus)," 2023.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," 2016.

[19] Damien Garreau and Dina Mardaoui, "What does lime really see in images?," 2021.

[20] Trung-Nghia Le, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen, "Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild," 2021.

[21] Leila Cristina C. Bergamasco and Fátima L.S. Nunes, "Intelligent retrieval and classification in three-dimensional biomedical images — a systematic mapping," *Computer Science Review*, vol. 31, pp. 19–38, 2019.

[22] Yuval Nirkin, Lior Wolf, Yosi Keller, and Tal Hassner, "Deepfake detection based on the discrepancy between the face and its context," 2020.

[23] Anuj Badale, Lionel Castelino, Chaitanya Darekar, and Joanne Gomes, "Deep fake detection using neural networks," in *15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, 2018, vol. 2.

[24] Muhammad Salihin Saealal, Mohd Zamri Ibrahim, Mohd Ibrahim Shapiai, and Norasyikin Fadilah, "In-the-wild deepfake detection using adaptable cnn models with visual class activation mapping for improved accuracy," in *2023 5th International Conference on Computer Communication and the Internet (ICCCI)*, 2023, pp. 9–14.