Detection of AI-Generated Faces Using Deep Learning and Explainable AI Techniques

Final Project Report Submitted to
The Department of Computer Science
Faculty of Computer and Information Technology
Jordan University of Science and Technology
In Partial Fulfillment of the Requirements for the Degree of Bachelors of Science in Computer Science

Prepared by:

Shahd Alrawabdeh[163352]
Ban Alrawabdeh [165854]
Reham Malkwi[162597]

Supervisor:

Dana ElRushaidat

January 2026

نموذج حقوق الملكية الفكرية لمشاريع التخرج في قسم علوم الحاسوب

يتم قراءة وتوقيع هذا النموذج من قبل الطلاب المسجلين لمشاريع التخرج في قسم علوم الحاسوب

تعود حقوق الملكية الفكرية لمشاريع التخرج ونتائجها (مثل براءات الاختراع أو أي منتج قابل للتسويق) إلى جامعة العلوم والتكنولوجيا الأردنية، وتخضع هذه الحقوق إلى قوانين وأنظمة و تعليمات الجامعة المتعلقة بالملكية الفكرية وبراءات الاختراع.

بناءا على ما سبق أوافق على ما يلي:

1) أن أحفظ كافة حقوق الملكية الفكرية لجامعة العلوم والتكنولوجيا الأردنية في مشروع التخرج.
2) أن ألتزم بوضع اسم جامعة العلوم والتكنولوجيا الأردنية و أسماء جميع الباحثين المشاركين في المشروع على أي نشرة علمية للمشروع كاملا أو لنتائجه. و يشمل ذلك النشر في المجلات و المؤتمرات العلمية عامة أو النشر على المواقع الإلكترونية أو براءات الاختراع أو المسابقات العلمية.
3) أن ألتزم بأسس حقوق التأليف المعتمدة في جامعة العلوم والتكنولوجيا الأردنية.
4) أن أقوم بإعلام الجهة المختصة في الجامعة عن أي اختراع أو اكتشاف قد ينتج عن هذا المشروع و أن ألتزم السرية التامة في ذلك و أن أعمل من خلال الجامعة على الحصول على براءة الاختراع التي قد تنتج عن هذا المشروع.
5) أن تكون جامعة العلوم والتكنولوجيا الأردنية هي المالك لأي براءة اختراع قد تنتج عن هذا المشروع و تشمل هذه الملكية حق الجامعة في إعطاء التراخيص و التسويق و البيع كمؤسسة راعية و داعمة لكافة الأنشطة البحثية. ويكون حق للطالب شمول اسمه على براءة الاختراع كأحد المخترعين، و في حال تم إعطاء تراخيص أو تسويق و بيع لأي من منتجات المشروع يمنح المخترعون بما فيهم الطالب نسبة من الإيرادات حسب تعليمات البحث العلمي في جامعة العلوم والتكنولوجيا الأردنية.

إسم الطالب التوقيع

إسم الطالب التوقيع

إسم الطالب التوقيع

إسم المشرف **التوقيع**

تاريخ ....

Detection of AI-Generated Faces Using Deep Learning and Explainable AI Techniques

Dana ElRushaidat
dept. Computer Science and AI
Jordan Univ. of Science and Technology
Irbid, Jordan
dmelrushaidat@just.edu.jo

2nd Shahd Al-Rawabdeh
dept. Computer Science and AI
Jordan Univ. of Science and Technology
Irbid, Jordan
slrawabdeh22@cit.just.edu.jo

3rd Ban Al-Rawabdeh
dept. Computer Science and AI
Jordan Univ. of Science and Technology
Irbid, Jordan
bralrawabdeh22@cit.just.edu.jo

4th Reham Malkawi
dept. Computer Science and AI
Jordan Univ. of Science and Technology
Irbid, Jordan
raalmalkawi22@cit.just.edu.jo

***Abstract*—With the rapid advances in AI models used for generating and/or manipulating facial images, distinguishing true face images from AI-generated ones has become a critical challenge for maintaining digital security and identity verification.**

This work aims to distinguish real face images from AI-generated face images using various deep learning models. Our methodology trains multiple deep learning models, ranging from pretrained to untrained models. We then evaluate their accuracy in distinguishing fake face images and apply Grad-CAM and LIME to understand the details of the results. To assess our model's performance, various metrics are used, including precision, recall, and F1-score. The initial results show the superiority of the pre-trained convolutional neural network models compared to transformer-based untrained models. We are still investigating overfitting and other factors that might provide a wrong accuracy measure. The results obtained so far are promising, and with further testing of multiple deep learning models, supported by Explainable AI, we aim to Provide a better understanding of the fake face image detection problem.

***Index Terms*—Fake face detection, Deep learning, CNN, Image classification.**

I. PROJECT GOALS AND OBJECTIVES

The goal of this project is to develop a deep learning-based system that can distinguish between real face images and fake images generated by GAN models. The project focuses on analyzing and preprocessing a real and fake face dataset, training several deep learning models using both pretrained architectures and models trained from scratch, and evaluating their performance using standard metrics such as accuracy, precision, recall, and F1-score.

This research uses explainable AI, such as Grad-CAM and LIME, to better understand how classification decisions for fake versus true face images are made.

II. INTRODUCTION

The emergence of artificial intelligence (AI) and deep learning over the last decade has enabled applications across many domains. AI-generated face images are considered one domain that introduced a high and serious security threat. The threat introduced by fake images on the digital identity and the possibility of misuse of such images is tremendous. The effect of such threat motivated research in the real-time and accurate detection of fake images and the ability to distinguish fake images from true ones.

Our methodology lies in using different models that do microscopic analysis on the face images. The method then uses explainable AI methods that provide the ability to explain the results obtained from the various models under consideration. The suggested Explainable AI methods are GRAD-CAM and LIME. Those models are supposed to provide us with a better understanding of the visual details in the images that resulted in the classification decision. This knowledge will help in separating real face images from fake AI-generated face images.

The use of Explainable AI models capability provide a great insight into the decision-making process of why the face image is considered fake or true. The Explainable AI will provide us with the visual attributes and features that help in the classification process. This integration between high-accuracy models and the understanding of why those models performed well is an addition to the field of fake face detection. This insight provided by the Explainable AI model will help in the selection of future models that would achieve even higher accuracy.

The selected dataset for the study [1] is publicly available through Kaggle. The dataset consists of 140,000 face images. The dataset is equally divided into two categories: real and fake face images. The real images are taken from the Flickr Faces HQ collection, while the fake faces are generated using StyleGAN [2]. The dataset is well-suited for training deep-learning models that aim to distinguish real face images from fake AI-generated face images.

Research in the accurate detection of fake face images from real ones is a hot ongoing research [3], [4]. Most of the research on the topic focused on comparing pre-trained deep-learning models, while work on untrained models remains limited [5]. The limitation of research in untrained models motivated us to conduct this research. Our research aims to provide a better understanding of the ability of untrained models to provide accurate fake face detection.

A. REVIEW AND ANALYSIS OF RELATED WORK

Recently, researchers have suggested some methods that rely on deep learning to help us distinguish between real and fake images. These methods typically work with models such as convolutional neural networks, long short-term memory networks, Vision Transformers, and some lightweight classifica-

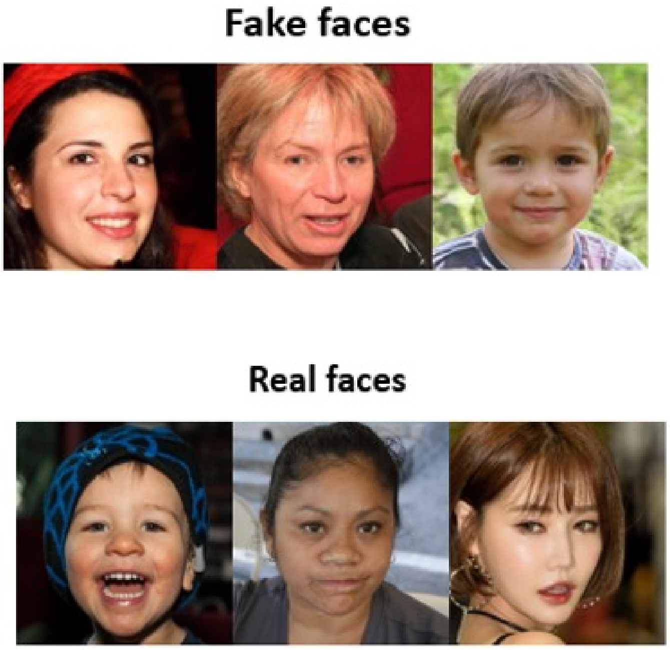

Fig. 1. Sample dataset [1]

tion networks. These models are often used in conjunction with transfer learning techniques and hybrid processing methods.

Mallet et al. [6] presented a method combining multilayer neural networks (MLPs) and long short-term memory networks to classify real and fake images using a dataset of 140,000 real and fake faces. The long short-term memory model improved the multilayer neural network model, achieving an accuracy of 74.7%. However, the model exhibited limited generalization due to training on a single dataset.

In a subsequent study, Mallet et al. [7] evaluated CNN and Support Vector Machine (SVM) classifiers on the same dataset. The CNN-based model achieved a significantly higher accuracy of 88.33% along with better precision, recall, and F1-score. Nevertheless, the authors noted that reliance on a single dataset may reduce real-world performance.

Cınar and Doğan [8] proposed PV-ISM, a patch-based Vision Transformer model designed to detect synthetic images by modeling fine-grained patch relationships using self-attention mechanisms. The model achieved strong results on CIFAKE and RVF-10K datasets; however, performance degradation was observed on facial datasets, particularly when using low-resolution images or insufficiently deep architectures.

Similarly, J.J. and S.P. [9] developed a deepfake detection pipeline using CNNs and several pretrained architectures, including MobileNet, ResNet50, and InceptionV3. Their system achieved up to 98.5% accuracy and was deployed through a web-based interface for real-time detection. Despite these results, the study was limited by hardware constraints, restricting training to only 20K images instead of the full 140K dataset.

Agrawal et al. [10] compared several models, including a fully connected network, a CNN, and the ResNet50 model, on low-resolution (64×64) GAN-generated face images, and their results showed that CNNs performed better than deep models, possibly due to the loss of some high-level features in the low-resolution data.

Naeem et al. [11]proposed a three-category classification system that distinguishes between real faces, GAN-generated fake faces, and AI-generated fake faces. Using a dataset of 140,000, statistical and visual differences were found between these categories; however, the low data resolution and simple models limited the generalizability.

In another work, Naeem et al. [12] presented a Vision Transformer-based system capable of capturing long-range spatial dependencies between image segments. The model demonstrated strong resilience on the FaceForensics++, DFDC, and Celeb-DF datasets, but it consumed significant GPU resources and required longer training times.

Nguyen et al. [13] proposed the FakeFormer model, a lightweight Transformer architecture developed to detect small manipulated areas. While effective with various types of fake faces, the model exhibited a performance decrease when implemented on images designed entirely using a GAN.

Kurt and Jabbarlı [14] introduced LightFFDNet, two lightweight CNN models designed for deepfake detection on resource-limited devices. These models achieved competitive performance compared to larger networks such as ResNet50 and VGG16; however, the limited number of training cycles and the diversity of the dataset negatively impacted the model's generalizability.

Sharma et al. [15] worked with CNN models and models previously trained, such as VGG16 and ResNet50. Their goal was to distinguish real faces from fake ones using a dataset of 140,000 images. They combined several models, which helped them achieve an accuracy close to 98.79. However, since they only used one dataset, this raised some doubts about overfitting and how well the model would perform in real situations.

Finally, Sharma and Sharma [16] introduced a dataset of 140,000 real and fake faces, in which they suggested a CNN-based detection framework. Their comparative analysis with pre-trained models showed the challenges of detecting deepfakes, stressing the need for models capable of generalizing across various kinds of forgeries, varying lighting conditions, and multiple manipulation techniques. Table I

III. APPROACH AND METHODOLOGY

A. Methodology

In this part, we present the steps of implementing our project, starting with selecting and preparing the dataset, followed by applying preprocessing techniques, and finally training multiple deep learning models to classify real and fake faces. The goal of this project is to demonstrate how the data

TABLE I
SUMMARY OF DEEPFAKE DETECTION RELATED WORK

Ref	Author(s)	Year	Dataset	Objective	Gap
[6]	Jacob Mallet, Natalie Krueger, Mounika Vanamala, Rushit Dave	2023	140k Real and Fake Faces	Combine MLP and LSTM to detect deepfakes.	Single dataset; poor generalization.
[7]	Jacob Mallet, Natalie Krueger, Mounika Vanamala, Rushit Dave	2023	140k Real and Fake Faces	Evaluate CNN vs SVM for deepfake detection.	Performance may additionally mislead in real-world scenarios.
[8]	Cınar, B. Do˘gan	2023	CIFAKE, RVF-10K	Patch-based totally Vision Trans-former (PV-ISM) for discovering artificial snapshots.	Low performance on detailed face datasets; wishes for higher decision.
[9]	J.J., S.P.	2023	20k GAN-generated photos	CNN + pre-educated models for deepfake detection.	Limited dataset; high computation required.
[10]	Harshal Agrawal, Ricky Parada, Colin Sullivan	2024	sixty four×64 GAN faces	Compare shallow CNN, FC network, ResNet50.	Low-decision pictures; limited generalization.
[11]	Shahzeb Naeem, Ramzi Al-Sharawi, Muhammad Riyyan Khan, et al.	2024	140K	ThThree-elegance classification: Real, GAN, AI-generated.	Simple fashions; low resolution; restricted generalization.
[12]	Shahzeb Naeem, Ramzi Al-Sharawi, Muhammad Riyyan Khan, et al.	2023	FaceForensics++, DFDC, Celeb-DF	Vision Transformer for deepfake detection.	Requires a massive GPU; longer schooling times.
[13]	Dat Nguyen, Marcella Astrid, Enjie Ghorbel, Djamila Aouada	2024	Multiple forgery datasets	Lightweight Transformer (FakeFormer) for little manipulated areas.	Weaker on completely synthetic GAN pictures.
[14]	Murat Kurt, G¨unel Jabbarlı	2024	224x224 face snap shots	Lightweight CNNs for fast deepfake detection.	Limited generalization; short schooling; low dataset range.
[15]	Jatin Sharma, Sahil Sharma, Vijay Kumar, Hany S. Hussein, Hammam Alshazly	2022	140k Real and Fake Faces	Ensemble CNNs for actual/faux face class.	Dataset dependence; overbecoming hazard; computationally high-priced.
[16]	Jatin Sharma, Sahil Sharma	2021	140k Real and Fake Faces	Develop CNN-based body-paintings for GAN detection.	High computation; fashionable-ization issues; restrained testing.

is handled and how each model is utilized in the classification process.

1) Dataset [1]: We relied on a single dataset for our research. The selection of this dataset was not random; it was chosen because it is one of the most widely used and up-to-date datasets for detecting real and fake faces. It contains a large number of images that are evenly distributed between the two classes, making it suitable for training and accurately evaluating deep learning models. Below, we present the characteristics of the dataset used and the distribution of its classes.

In this research, we used the **140k Real and Fake Faces** dataset, which is publicly available on the Kaggle platform. The dataset contains a total of 140k face images, with 70k real face images (Real) and 70k fake face images (Fake).

The real images were collected from the high-quality Flickr-Faces dataset, while the fake images were generated using the StyleGAN model, one of the most advanced generative adversarial networks capable of producing highly realistic synthetic faces. The dataset consists of two classes only (binary classification): real faces and fake faces.

The images exhibit a high degree of diversity in age, gender, background, and lighting conditions for both real and fake categories. This diversity in the data makes the image dataset suitable for training strong deep learning models that can perform well on new and unseen images, especially for distinguishing between real and fake faces.

Due to the lack of resources needed to use the whole dataset, a balanced subset of the original dataset is used. The data splitting used: training (50,000), split equally between the two classes. Validation (10,000) was split equally between the two classes.

2) Preprocessing: Before training the deepfake face detection model, a few preprocessing steps were carried out on the data, and the accuracy of the results must be ensured, as well as the model's performance. In this stage, we focused on cleaning the images, making them similar in standardized dimensions and form, and introducing greater variation to make these models compatible with deep learning neural networks. We carried out this preparation process step by step for all models, whether they were pre-trained or trained from scratch.

For the pre-trained models, the first step was image clean-up and deleting the ones that were corrupt. There could be corrupted images when uploaded. Thus, the images become unusable for the model training.

To prevent problems, we wrote a small piece of code that checked. Each image in the folders was ensured to be opened. Any corrupted images were immediately removed to avoid problems during training, such as the data loader crashing or the training becoming less stable.

After confirming that all images were opening, we resized them to 224 x 224 pixels, as this is the size commonly used by most networks, such as ResNet, VGG, and MobileNet.

This reduces the computational burden on the computer, making the model run faster, and ensures that all images are of the same size. We then converted the pixels to tensors and adjusted them using the mean and standard deviation from ImageNet data, which are [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225]. These are the same values that most pre-trained networks use for their means and standard deviations, allowing for accelerated training by preventing the corruption of results with large pixel value gradients.

In the model, to avoid storing, in addition to increasing the variety of images, we made a random change in the training images, such as randomly flipping them horizontally, and then rotating ±10 degrees to create images from different angles, applying color adjustments, and stretching the contrast to enhance brightness and achieve ideal contrast. These techniques improve the model's performance on real deepfake videos.

For the models, we trained from scratch, we used the same method, adding. So with a few extra steps, you get the exact estimation of data for the image set. We used the same method to remove corrupted images, as described earlier. Next, the training dataset was loaded and reloaded without changes to compute the mean and standard deviation of the red, green, and blue color channels. Which we then used it for adjustments during training and evaluation.

We resized all the images to 224 x 224 pixels so that all the images are the same size before being fed into the model.

For the training dataset, some modifications were introduced to add variability to the data: random horizontal flip, rotation of ±10°, and limited changes in the brightness and color components. In the validation and test sets, only image resizing, tensor creation, and image normalization were conducted in order to properly evaluate the model during this phase of the experiment.

Finally, we loaded all this data into a data loader with a batch size of 32. The data was shuffled in the training set, but not shuffled in both the validation and testing sets.

3) Models Used for Image Classification: In this project, we used eight deep neural network models to classify images into two groups: real and fake. Below is a detailed description of the steps and parameters used for each model.

a) MobileNetV2 [17]: A pre-trained MobileNetV2 model, initially trained on the ImageNet dataset, was used due to its high efficiency and relatively low computational complexity compared to deeper architectures. To modify the model for detecting fake faces, the final classification layer was modified to produce two outputs, corresponding to the two types of faces used in this dataset (real and fake). The Adam optimizer with a learning rate of 0.0001 was also employed to ensure stable and efficient convergence. The model was trained over 30 epochs, and pre-trained weights were used to refine the learning process and improve the overall performance of the model.

b) ResNet18 [18]: We used a ResNet18 model pre-trained on ImageNet because of its residual connections, which simplify optimization and help stabilize training in deep networks. ResNet18 was chosen over deeper editions (e.G., ResNet50) to balance accuracy and computational cost; its smaller potential is also better appropriate to medium-sized datasets and may minimize overfitting. For the fake face detection task, we replaced the final fully related layer with a 2-class classifier (real vs. Fake). We trained the model using CrossEntropyLoss and optimized it with Adam (lr = 0.0001) for 30 epochs, initializing from pre-training weights to boost up convergence and improve generalization.

c) DenseNet-121 [19] : We used the DenseNet-121 model, a convolutional neural network (CNN) pre-trained on the ImageNet dataset. This model is characterized by its capability to reuse capabilities across layers, which improves learning despite limited data and mitigates the vanishing gradient problem. To modify the binary version of the project (real or fake), the very last layer was replaced with a fully connected layer containing two neurons. The model was trained for 30 epochs using the Cross-Entropy Loss with the Adam optimizer, and an lr(0.0001). DenseNet-121 was selected as it provides a good balance between accuracy and computational performance, and the pre-trained ImageNet weights allow for leveraging prior knowledge to improve overall performance on our task more quickly.

d) EfficientNetB0 [20]: The EfficientNetB0 model was used to train ImageNet for its efficiency, low computational risk, and stability in terms of accuracy and training speed across deeper models. To adapt it for the task of detecting fakes, the final classification layer was modified to produce only two categories: real and fake. The CrossEntropyLoss loss function was used, and the AdamW optimizer was adopted with a learning rate of 0.0001. The model was trained for 30 training epochs.

Table II shows the hyperparameters of pretrained models.

TABLE II
HYPERPARAMETERS OF PRE-TRAINED MODELS FOR FAKE FACE DETECTION

Model	Dataset	Opt.	LR	Loss	Epochs
MobileNetV2	ImageNet	Adam	1e-4	CrossEnt	30
ResNet18	ImageNet	Adam	1e-4	CrossEnt	30
DenseNet121	ImageNet	Adam	1e-4	CrossEnt	30
EfficientNetB0	ImageNet	AdamW	1e-4	CrossEnt	30

e) ViT-Tiny [21]: We used the ViT-Tiny (Vision Transformer) model from the timm library to classify binary images. The model processes images as patches and extracts long-range contextual information. It was initialized without pre-trained weights, and the last layer was modified to produce a single output. The training used the BCEWithLogitsLoss loss function with the Adam optimizer (learning rate of 0.0001) on the available GPU or CPU, allowing for fast and efficient experimentation while taking advantage of the adapter-based features. Figure 2 shows the model architecture.

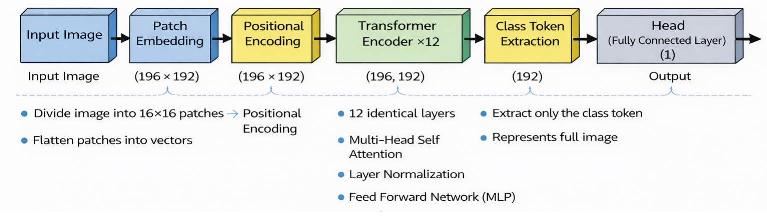

Fig. 2. ViT-Tiny model

f) Convolutional neural network [22] : We used the SimpleCNN model, a simple convolutional neural network model, to classify images into two categories: real and fake. This model consists of three sequential convolutional layers, a max pooling layer, and a dropout layer to help reduce data size and prevent overfitting. The output is then converted to a directional format and passed through two fully connected layers to produce the final result. The model was trained using CrossEntropyLoss and the Adam optimizer at a learning rate of 0.001, implemented on either a GPU or a CPU, depending on availability. Figure 3shows the model architecture.

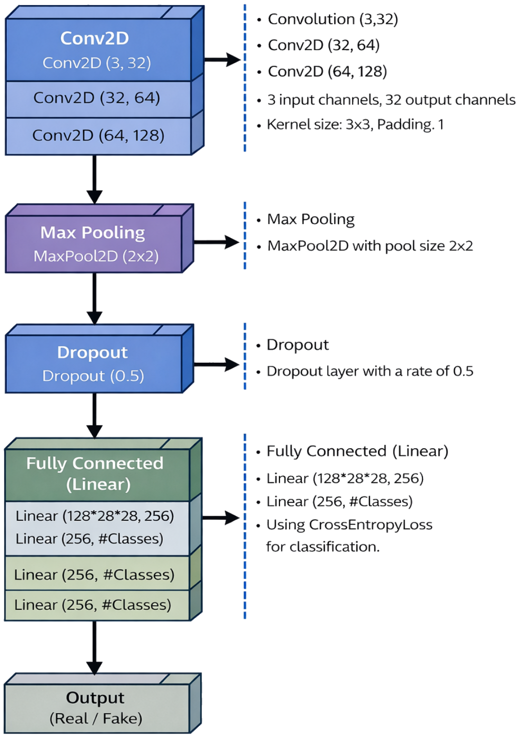

Fig. 3. CNN model

g) HybridCMT [23]: We used a HybridCMT model, combining a CNN and a transformer, to classify images as real or fake. The convolutional part consists of three CNN blocks for spatial feature extraction, followed by a flattening and dimensioning stage to feed the transformer. The transformer consists of two encoder layers and four attention heads to capture long-range relationships. The average is then taken and sent to a fully connected classification head with a dropout to generate the final result. It is trained using cross-entropy loss and ADAM at a learning rate of 0.0003 with the use of a GPU, which enables highly efficient local feature extraction. Figure 4shows the model architecture.

h) PiT-Small [24]: We used the PiT-Small model, a vision converter based on the lightweight converter architecture, for image classification. In PiT, the image is split into multiscale parts and then turned into a sequence for self-attention. The main difference from typical vision transformers is that PiT groups tokens between blocks, which reduces the number of tokens and lowers the computational cost. This helps the model learn both local details and global patterns efficiently, with stable training. We used CrossEntropy Loss and Adam optimizer at a learning rate of 0.001 for training, on a GPU if available; otherwise, on the CPU. Figure 5shows the model architecture.

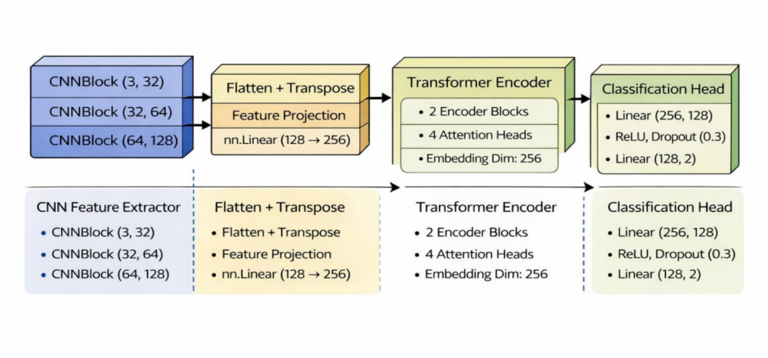

Fig. 4. HybridCMT model

Table III shows the hyperparameters of unpretrained models.

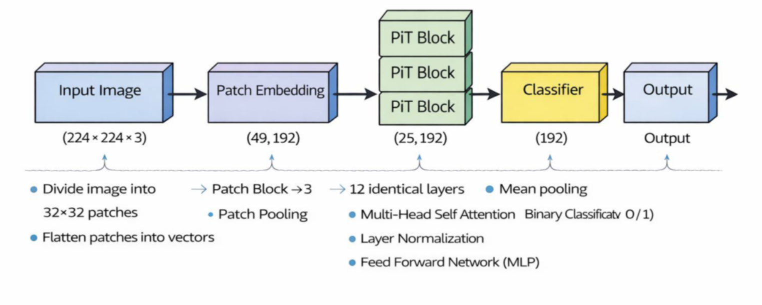

Fig. 5. PiT-Small model

TABLE III
HYPERPARAMETERS OF DEEPFAKE DETECTION MODELS

Model	Type	Loss	Opt.	LR	Epochs
ViT-Tiny	ViT	BCELogits	Adam	1e-4	30
SimpleCNN	CNN	CrossEnt	Adam	1e-3	30
HybridCMT	CNN+Trans	CrossEnt	Adam	3e-4	30
PiT-Small	ViT	CrossEnt	Adam	1e-3	30

Explainable Artificial Intelligence techniques were also utilized within the proposed research for improving the explicability of the results obtained from the deep learning models. Specifically, Grad-CAM and LIME were utilized for the explanation of the areas of the image that contributed the most towards the image classification process.

The Grad-CAM technology has also been applied to convolutional neural networks in order to generate heat maps that highlight the important regions of the input images. This would help in understanding whether the models focus on the relevant or irrelevant regions of the image while distinguishing between real and fake images.

Moreover, the LIME tool was used in this study for the purpose of local interpretability of the developed models. The LIME tool assists in making the outcomes visible in an understandable form by specifying the super-pixels that have an important influence on making the predictions of the model. In this respect, the use of the Grad-CAM tool and the LIME tool assists in conducting an in-depth study on the decision-making process.

4) Model Evaluation: After finishing each training phase, we tested the model using validation or test data, if available. We measured the amount of error and accuracy to see how well the model was performing and whether it was just memorizing the training data. This method helps us ensure that it can also perform well on new data.

a) Performance Metrics : To see our model, we used accuracy, Precision, Recall, and F1-score. We list them with their corresponding equations below: **Accuracy**, **Precision**, **Recall**, and **F1-score**. We list them with their corresponding equations below:

1) **Accuracy:** Accuracy measures the proportion of correctly classified instances:

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

2) **Precision:** The ratio of true positives to all predicted positives:

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

3) **Recall:** The ratio of true positives to all actual positive instances:

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

4) **F1 Score:** The harmonic mean of precision and recall, used to balance both metrics:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

5) EXPECTED RESULTS/OUTPUTS: In this section, we present the experimental results of deep learning models used to classify images into two categories: real and fake. The performance of each model on the test dataset was evaluated using common binary classification metrics, including accuracy, fine-tuning, recall, and the F1 metric. We also provide a comparison between pre-trained models and models trained from scratch. In addition to a comparison between convolutional neural network-based architectures and transformer-based architectures.

B. Results of Pretrained Models

Based on our testing of the four pre-trained models mentioned, the MobileNet model achieved high accuracy, reaching 99.80%, and performed strongly with low computational costs. The ResNet18 model performed reasonably well but was less stable, achieving the lowest accuracy compared to the other models. The DenseNet121 model showed strong and stable results, reaching 99.82% accuracy. Finally, the EfficientNetB0 model achieved the best performance, reaching the highest accuracy of 99.98% with consistently low losses. Based on our results, the EfficientNetB0 and DenseNet121 models were the best among the models tested. As shown in Figure6, the training loss of the best-performing pretrained model decreases steadily over epochs, indicating stable learning and effective convergence. In Figure7, the accuracy curve demonstrates a clear improvement throughout the training process, showing a strong correlation between model convergence and performance stability.

TABLE IV
PERFORMANCE OF PRETRAINED MODELS ON TEST SET (CLASS-WISE METRICS)

Model	Accuracy	Class	Precision	Recall	F1-score
MobileNetV2	99.8%	Fake	1.00	0.99	1.00
		Real	0.99	1.00	1.00
ResNet18	99.4%	Fake	0.99	1.00	1.00
		Real	1.00	0.99	1.00
DenseNet121	99.82%	Fake	1.00	1.00	1.00
		Real	1.00	1.00	1.00
EfficientNetB0	99.98%	Fake	1.00	1.00	1.00
		Real	1.00	1.00	1.00

Table IV shows the results of pretrained models.

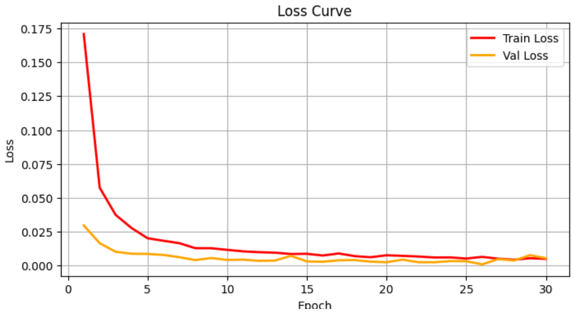

Fig. 6. loss best model

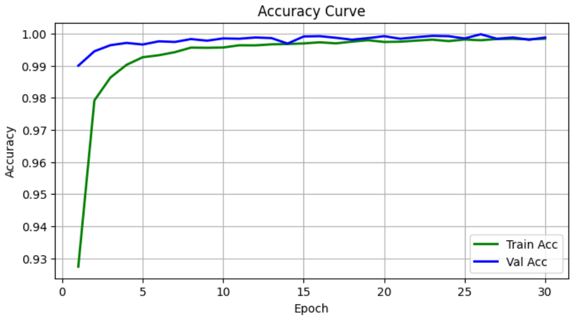

Fig. 7. Accuracy best model

C. Results of Unpretrained Models

We then trained on the previously mentioned untrained models. We started with the Simple CNN model, which performed best, exhibiting rapid convergence, high generalization, and the highest verification accuracy of 95.09% with consistently

TABLE V
PERFORMANCE OF MODELS TRAINED FROM SCRATCH ON TEST SET (CLASS-WISE METRICS)

Model	Accuracy	Class	Precision	Recall	F1-score
ViT-Tiny	88%	Fake	0.90	0.85	0.87
		Real	0.86	0.90	0.88
SimpleCNN	95%	Fake	0.96	0.94	0.95
		Real	0.94	0.96	0.95
HybridCMT	56%	Fake	0.57	0.50	0.53
		Real	0.55	0.62	0.58
PiT-Small	70%	Fake	0.65	0.85	0.74
		Real	0.79	0.55	0.65

low loss. This meant that the CNN model was able to extract important features from images.

The ViT model was a Vision Transformer Tiny Patch 16-224. It achieved the second-best result, with 88.34%. Its performance was stable across channels, leveraging an attention mechanism that detects public relationships between images. The Hybrid CNN+Transformer model performed next, with which we can therefore conclude that poor performance will occur, as verification accuracy ranges between 55% and 60%, and it lacks learning stability. Finally, the PiT model performed moderately by achieving 71% verification accuracy, higher than that obtained previously. In our experience, we conclude that the Simple CNN and the best models of all kinds were the ViT-Tiny models. Figure 8 illustrates the loss behavior of the best-performing untrained model during training, where a gradual reduction in loss indicates successful feature learning. As shown in Figure9, the accuracy of the untrained model improves over time, confirming its ability to generalize and distinguish between real and fake face images. Table V shows the results of untrained models.

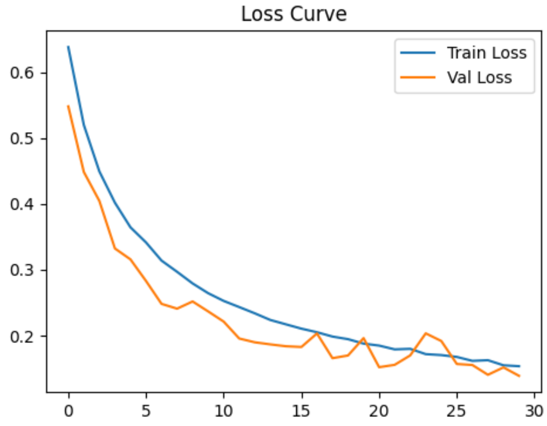

Fig. 8. loss best model

D. CONCLUSION

This project focused on detecting fake images using advanced deep learning models. Some models were trained. We also used a large and diverse dataset, which helped us obtain accurate results and clearly compare the performance of different models.

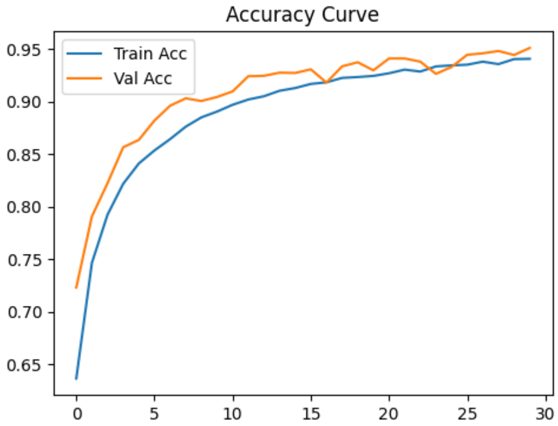

Fig. 9. Accuracy best model

Some models, such as EfficientNet and DenseNet, were pre-trained, while others were trained from scratch, and these models achieved the best performance. They demonstrated excellent accuracy and stability. However, simpler models, such as CNN, also performed remarkably well considering their simplicity.

This suggests that by choosing the right model and applying the data correctly, pre-processing can lead to excellent results without the need for high complexity. There will be no need for complex processes in cases of highly complex data.

We will also apply tools for data analysis, such as Grad-Cam and LIME, in order to better understand how the model works and why it detects a real image or a fake one. This will help improve the accuracy of the outcomes and will allow us to better understand how the model behaves.

Overall, this project will ensure that we understand how to handle deepfakes, the value of data quality, the importance of proper data handling, and how the optimal model needs to be chosen. We believe that this will be just the beginning of further research in the fields of video, complex data, or the improvement of the model's performance in more realistic conditions.

E. LOCATION AND SAFETY CONSIDERATIONS

The work was conducted on a personal computer with a GPU to accelerate model training. All experiments were conducted in a secure environment with no physical risks. Data privacy was considered, and only publicly available datasets were used for training and testing.

REFERENCES

[1] Clément Bisaillon, "140k real and fake faces dataset," https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces, Kaggle, 2026, accessed: 2026-01-11.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1812.04948

[3] L. Gong and X. Li, "A contemporary survey on deepfake detection: Datasets, algorithms, and challenges," *Electronics*, vol. 13, no. 3, p. 585, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/3/585

[4] P. Liu, Q. Tao, and J. T. Zhou, "Evolving from single-modal to multi-modal facial deepfake detection: A survey," *arXiv Preprint*, vol. abs/2406.06965, 2024. [Online]. Available: https://arxiv.org/abs/2406.06965

[5] F. Croitoru, A. Hiji, V. Hondru, N. C. Ristea, P. Irofti, M. Popescu, C. Rusu, R. T. Ionescu, F. Shahbaz Khan, and M. Shah, "Deepfake media generation and detection in the generative ai era: A survey and outlook," *arXiv Preprint*, vol. abs/2411.19537, 2024. [Online]. Available: https://arxiv.org/abs/2411.19537

[6] J. Mallet, N. Krueger, M. Vanamala, and R. Dave, "Deepfake detection using hybrid mlp and lstm networks," *Journal of Artificial Intelligence Research*, 2023.

[7] N. Mallet, J. Vanamala, M. Krueger, and R. Dave, "Deepfake image detection using cnn and svm classifiers," *Pattern Recognition Letters*, 2023.

[8] B. Cınar and B. Doğan, "Pv-ism: Patch-based vision transformer for synthetic image detection," *Expert Systems with Applications*, 2023.

[9] J. J. and S. P., "Deepfake detection using cnns and pretrained models," *Procedia Computer Science*, 2023.

[10] H. Agrawal, R. Parada, and C. Sullivan, "Detecting gan-generated faces at low resolution," *IEEE Access*, 2024.

[11] S. Naeem, R. Al-Sharawi, and M. R. Khan, "Three-class classification of real, gan-generated, and ai-generated facial images," *Applied Sciences*, 2024.

[12] R. Naeem and M. R. Khan, "Vision transformer for robust deepfake detection," *IEEE Transactions on Information Forensics and Security*, 2023.

[13] D. Nguyen, M. Astrid, E. Ghorbel, and D. Aouada, "Fakeformer: A lightweight transformer for deepfake detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[14] M. Kurt and G. Jabbarlı, "Lightffdnet: Lightweight cnns for fast deepfake detection," *Neural Computing and Applications*, 2024.

[15] J. Sharma, S. Sharma, V. Kumar, H. S. Hussein, and H. Alshazly, "Ensemble deep learning models for deepfake face detection," *Multimedia Tools and Applications*, 2022.

[16] J. Sharma and S. Sharma, "140k real and fake faces dataset for gan-generated face detection," *Data in Brief*, 2021.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *arXiv Preprint*, vol. abs/1801.04381, 2018. [Online]. Available: https://arxiv.org/abs/1801.04381

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv Preprint*, vol. abs/1512.03385, 2016. [Online]. Available: https://arxiv.org/abs/1512.03385

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *arXiv Preprint*, vol. abs/1608.06993, 2017. [Online]. Available: https://arxiv.org/abs/1608.06993

[20] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv Preprint*, vol. abs/1905.11946, 2019. [Online]. Available: https://arxiv.org/abs/1905.11946

[21] S. Wang, J. Gao, Z. Li, X. Zhang, and W. Hu, "A closer look at self-supervised lightweight vision transformers," *arXiv Preprint*, vol. abs/2205.14443, 2022. [Online]. Available: https://arxiv.org/abs/2205.14443

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *arXiv Preprint*, vol. abs/726791, 1998. [Online]. Available: https://ieeexplore.ieee.org/document/726791

[23] D. W. Deressa, H. Mareen, P. Lambert, S. Atnafu, Z. Akhtar, and G. Van Wallendael, "Genconvit: Deepfake video detection using generative convolutional vision transformer," *Applied Sciences*, vol. 15, no. 12, p. 6622, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/15/12/6622

[24] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," *arXiv Preprint*, vol. abs/2103.16302, 2021. [Online]. Available: https://arxiv.org/abs/2103.16302