Final Course Project, CS4152, 2025.

Date of Submission 22/01/2025

FauxFinder: DCGAN and CNN models for Art Classification using Transfer learning

KALEEMULLAH YOUNAS[1], ASADULLAH MASOOD[2], AND DAUOOD JAFFAR[3] (Members, UMT)

[1]Department of Data Science, University of Management and Technology, Lahore, Pakistan
[2]Department of Artifical Intelligence, University of Management and Technology, Lahore, Pakistan

"This work was supported in the course CS4152, part by the Dr Shaista Habib - Program Head AI, Department of Artificial Intelligence, University of Management and Technology"

ABSTRACT The "FauxFinder" project addresses the difficulty of identifying between real and AI-generated art in an era of rapidly evolving artificial intelligence and deep learning. The major goal is to create a high-accuracy classification system using a bespoke Deep Convolutional Generative Adversarial Network (DCGAN) to generate synthetic art images and Convolutional Neural Networks (CNNs) for classification. The dataset includes 10,821 genuine art images and an equal number of AI-generated images, all reduced to 256x256 pixels for consistency. The DCGAN generator uses fractional strided convolutions to generate high-quality synthetic pictures, whilst the discriminator extracts features using strided convolutions. Pre-trained models with fine-tuned parameters, such as MobileNetV1 and MobileNetV2, improve classification accuracy even further.MobileNetV1 outperformed expectations in terms of accuracy, precision, recall, and F1-score, with a test accuracy of 95.15%. This research highlights the feasibility of using generative and discriminative models to address the issues of forgery detection while maintaining artistic integrity in a quickly changing digital environment.

KEYWORDS *DCGAN (Deep Convolutional Generative Adversarial Network); CNN (Convolutional Neural Networks) Model; Forgery Detection; Counterfeit Art Detection; Pre-trained Models; Transfer Learning; strided and fractional strided convolutions; Literature Review; Data Augmentation; Fine-tuned MobileNetV1; Fine-tuned MobileNetV2 etc*

I. INTRODUCTION

ART has always been an important component of human expression, capturing emotions, ideas, and cultural change over the years. It is a *worldwide language* that crosses borders, enabling people to interact and share their ideas and stories. However, as technology advances, **conventional artistic boundaries are being challenged**. The emergence of artificial intelligence (AI) has brought a new paradigm in the creative arena, with algorithms and neural networks capable of producing artwork on par with human efforts. For example, AI models can create detailed digital paintings or sculptures in styles [16] ranging from *classical realism* to *contemporary abstraction*. This globalization of creativity has made it possible for people with no formal training to create visually magnificent works of art.

However, this invention presents substantial obstacles. The increasing sophistication of AI-generated art has made it **impossible to differentiate between works created by humans and those produced by machines**. Consider an art buyer who purchases a painting, believing it to be an original masterpiece by a renowned artist, only to discover later that it was created by a *machine-learning algorithm*.

Beyond art, AI has enabled the development of **deepfake videos and audio**, in which algorithms can almost perfectly replicate a person's voice or face. These deepfakes [15] have aroused serious concerns since they can be used to *disseminate falsehoods, impersonate individuals, and harm reputations*. For example, a deepfake video may falsely portray a political leader making incendiary remarks, causing widespread confusion and potential confrontation. Similarly, **counterfeit goods**, such as luxury fashion items or designer accessories, are becoming increasingly difficult to detect as forgers use new technologies to replicate authentic designs with remarkable accuracy. These examples highlight the **growing need for dependable tools and systems** to detect counterfeit and AI-generated items.

PROJECT SCOPE

The **"FauxFinder"** project's special objective is to detect and differentiate between **real** and **AI-generated art**. While techniques and procedures exist to detect counterfeit physical artworks [17], the rise of *AI-generated digital art* presents a distinct challenge due to its precision in mimicking human creativity. Existing systems frequently fail to successfully analyze and categorize digital art as real or false, especially when dealing with datasets of comparable resolutions and creative complexity. Furthermore, many traditional methods rely on *personal examination* or *general-purpose models*, which do not capture the complex details of artistic patterns.

OBJECTIVE

The primary objective of the **FauxFinder** project is to develop an automated and reliable system for distinguishing between *real* and *AI-generated art images*. Our solution leverages a custom Deep Convolutional Generative Adversarial Network (**DCGAN**) combined with **Convolutional Neural Networks (CNNs)** [6] and fine-tuned pre-trained models like **MobileNetV1 and MobileNetV2** to achieve high accuracy in classification.

DEEP CONVOLUTIONAL GAN

To generate the fake art dataset, we designed a custom DCGAN with the following features:

- **Generator:** Uses fractional strided convolutions (transposed convolutions) to upsample input latent vectors into 256x256 pixel images, ensuring smooth outputs without checkerboard artifacts.
- **Discriminator:** Employs strided convolutions to downsample images and extract features for classification into real or fake.
- **LeakyReLU Activation:** The LeakyReLU activation function, defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0, \end{cases} \tag{1}$$

  where $\alpha = 0.2$, introduces non-linearity while allowing a small gradient for negative inputs.
- **Dropout:** Dropout layers with a rate of 0.25 in the discriminator prevent overfitting by randomly deactivating neurons during training.

LOSS FUNCTIONS

The GAN was trained using binary cross-entropy loss. The loss function for the discriminator is:

$$\mathcal{L}_D = -\mathbb{E}_{x\sim P_{data}(x)}[\log D(x)] - \mathbb{E}_{z\sim P_z(z)}[\log(1 - D(G(z)))].$$

The generator aims to maximize the discriminator's likelihood of classifying generated images as real:

$$\mathcal{L}_G = -\mathbb{E}_{z\sim P_z(z)}[\log D(G(z))].$$

KEY FEATURES APPROACH

The table below summarizes key features of the approach:

TABLE 1: Key Features of the FauxFinder Approach

Feature	Details
Generator	Fractional strided convolutions
Discriminator	Strided convolutions with Dropout (0.25)
Activation Function	LeakyReLU ($\alpha = 0.2$)
Loss Functions	Binary Cross-Entropy for both Generator and Discriminator
Classification Model	MobileNetV1 + MobileNetV2 (Fine-tuned) + CNN Model

II. LITERATURE REVIEW

Rapid advances in **deep learning** have prompted extensive research on identifying authentic from false photos. *Generative models*, including **Generative Adversarial Networks (GANs)**, *Variational Autoencoders (VAEs)*, and advanced neural architectures, have emerged as game-changing tools for producing high-quality synthetic data. However, these technologies have prompted worries about their potential misuse, particularly in the creation of **deepfakes** [1] that are indistinguishable from real photographs. The rise of synthetic media underscores the critical need for robust techniques of *authenticating* and *classifying* such images.

Several studies have proposed novel deep learning-based algorithms [2] to address these difficulties, with an emphasis on tasks like *forgery detection* [9], *art style recognition* [5], and *counterfeit prevention* [11]. Techniques based on **Convolutional Neural Networks (CNNs)**, hybrid models, and transfer learning have produced promising results. Despite substantial advances, established approaches frequently face constraints such as *high computing requirements*, *generalizability to new datasets*, and *scalability for real-time applications*. This literature review critically evaluates previous publications to provide a full overview of the field's current state and propose opportunities for future research.

IMPORTANCE

Understanding the methodologies and constraints of previous research is critical for progressing the field of image classification, especially when recognizing *synthetic or edited images*. This understanding lays the groundwork for developing novel approaches that solve the inadequacies of existing models [11]. For example, while **GANs** and other generative models have great capabilities, their outputs frequently contain tiny faults that detection models might exploit. By examining these artifacts, researchers can create more effective algorithms for detecting altered photos.

Furthermore, the significance of this work goes beyond academic research. *Realistic fake media* threatens public trust, media integrity, and possibly national security.

Title	Authors	Method/Approach	Results/Findings	Limitations/Gaps
"Deep fake image detection based on pairwise learning."	*Hsu,Chih-Chung, Yi-Xiu Zhuang, and Chia-Yen Lee.*	*The method used in the research paper is a deep convolutional neural network (CNN) called the "proposed CFFN" (Convolutional Face Forgery Detection Network). The proposed CFFN is a multi-layered neural network that uses a combination of convolutional and fully connected layers to learn features from images and detect fake face images.*	*The results or findings of the research paper are presented in Table, which shows the objective performance comparison of the proposed CFFN with other fake face detectors. The proposed CFFN outperforms the other methods in terms of precision and recall, with a precision of 0.986 and a recall of 0.751.*	*Some potential limitations could be: 1. The dataset used to train the proposed CFFN may not be representative of all possible fake face images. 2. The proposed CFFN may not be able to detect fake face images that are highly realistic or have been manipulated using advanced image editing software. 3. The proposed CFFN may not be able to detect fake face images that are created using different techniques, such as 3D modeling or facial recognition software. 4. The proposed CFFN may not be able to detect fake face images that are created using a combination of different techniques.*
"Fake vs Real Image Detection Using Deep Learning Algorithm"	*Fatoni, F., Kurniawan, T., Dewi, D., Zakaria, M., & Muhayeddin, A.*	*The approach used in the study is a comparative analysis of three deep learning models, namely Convolutional Neural Networks (CNNs), VGG16, and Residual Network (ResNet), for image classification. The models were trained and tested on a dataset of images, and their performance was evaluated using metrics such as training and validation accuracy, loss, and confusion matrix.*	*The results of the study show that the ResNet model performed the best, with a training accuracy of 95% and a validation accuracy of 93% in Experiment 4. The CNN model had a training accuracy of 94% and a validation accuracy of 92% in the same experiment. The VGG16 model had a lower training accuracy of 94% but a higher validation accuracy of 94% in Experiment 3. The study also found that the training and validation loss of the three models decreased continuously across the epochs.*	*Some potential limitations or gaps could include: 1. The study only compared three deep learning models, and it is possible that other models may perform better or worse depending on the specific task and dataset. 2. The study did not explore the impact of hyperparameter tuning on the performance of the models. 3. The study did not evaluate the models on a large-scale dataset or in a real-world application. 4. The study did not provide a detailed analysis of the confusion matrix or the performance of the models on specific classes or sub-classes of images.*
"Artist identification with convolutional neural networks."	*Viswanathan, Nitin.*	*The authors used a convolutional neural network (CNN) to identify artists based on their painting styles. They used a dataset of 17,000 paintings from 57 artists, with 300 paintings per artist. The dataset was split into training, validation, and test sets using an 80-10-10 split per artist. The authors preprocessed the images by zero-centering and normalizing them, then taking a 224x224 crop of each image. During training, they randomly horizontally flipped the input images with a 50% probability and took a random crop of the painting. For validation and test images, they took a 224x224 center crop of the image to ensure stable and reproducible results*	*The authors reported that most paintings were correctly classified, with the diagonal entries in the confusion matrix being yellow or red. However, they noted that one artist, Henri Matisse, was particularly confused with Martiros Saryan, with the network predicting Saryan as the artist for 3 of Matisse's paintings.*	*It can be inferred that the dataset may not be representative of all art styles and periods, and that the network may not perform well on paintings that are not well-represented in the dataset. 1. The network was able to learn a representation of painting style that allowed it to identify artists, but it may not have captured all the nuances and complexities of artistic style. 2. The authors did not provide a detailed analysis of the network's performance on specific artists or styles, which may be an area for future research.*
"Recognizing Art Style Automatically in Painting Using Convolutional Neural Network."	*Akter, Mahfuza, Mst Rasheda Akther, and Md Khaliluzzaman.*	*The method or approach used by the authors is deep learning, specifically convolutional neural networks (CNNs), to recognize art styles from the Wikipaintings dataset. The authors used a miniVGGNet architecture, which is a variant of the VGGNet architecture, to classify the art styles into different categories.*	*Their results or findings show that the proposed model achieved an accuracy of 60.37% on the Wikipaintings dataset, which is more than 18.50% better than the miniVGGNet model. The authors also presented a classification report, which shows the precision, recall, and F1-score for each art style category. The results suggest that the proposed model is effective in recognizing art styles from the Wikipaintings dataset.*	*Limitations or gaps in their work include the following: 1. The authors used a relatively small dataset, which may not be representative of all art styles. 2. The dataset may be biased towards certain art styles or periods. 3. The authors did not evaluate the model on other datasets or compare it with other state-of-the-art models. 4. The model may not generalize well to other datasets or art styles that are not present in the Wikipaintings dataset.*
"Unsupervised representation learning with deep convolutional generative adversarial networks."	*Radford, Alec, Luke Metz, and Soumith Chintala.*	*The method or approach they used is Generative Adversarial Networks (GANs), which is a type of deep learning algorithm that uses a neural network to generate new data samples that are similar to a given dataset.*	*Their results or findings are that GANs are able to generate images that are similar to the training data, but with some limitations. The authors found that the generated images often suffer from being blurry, noisy, and incomprehensible.*	*The limitations or gaps in their work are that the generated images are not always of high quality and may not always be representative of the training data. Additionally, the authors note that there is still much to be learned about how to effectively use GANs for image generation.*
"Application of an improved DCGAN for image generation."	*Liu, Bingqi, et al.*	*The method or approach used is a concept classification network, which is proposed to eliminate errors in the image generation process. The network is trained on the MNIST dataset and uses a DCGAN model as the generator.*	*The results or findings mentioned in the text include the ability of the concept classification network to effectively solve the problem of low-quality images being output by the GANs model, and achieving good results.*	*limitations or gaps in their work are: 1. It can be inferred that the use of the MNIST dataset may limit the generalizability of the results to other datasets. 2. Additionally, the text does not provide a detailed comparison with other image generation methods, which could be a limitation.*
"Fake Image Detection Using Deep Learning."	*Khudeyer, Raidah Salim, and Noor Mohammed Almoosawi.*	*Three versions of the model were proposed: Model 1 used transfer learning with the pre-trained EfficientNetB0, and the model weights were fine-tuned for the binary classification task of identifying real and fraudulent photos. Model 2 added two more dense layers to the fully linked component of the architecture, which, when combined with dropout and batch normalization approaches, reduced overfitting and improved convergence. Model 3 improved upon the strategy by introducing a learning rate scheduling technique, which allows the model to dynamically modify the learning rate during training, resulting in faster convergence and more exact weight alterations..*	*The experimental findings showed that the proposed changes to EfficientNetB0 considerably improved performance. Model 1 obtained a test accuracy of 51.88%, whereas Model 2 increased it to 65.88%. Model 3 produced the greatest results, scoring 99.06% accuracy on the test set, 99.37% on the validation set, and 100% on the training set. This high accuracy and low error rate (0.0569) demonstrate the efficacy of the upgraded EfficientNetB0 architecture and the learning rate scheduling method.*	*The study has significant drawbacks. The technique was only tested on facial photos and not on other types of synthetic or modified content, which limits its application in a variety of real-world circumstances. The study does not explore the model's robustness to adversarial attacks or its performance in real-time detection settings, both of which are crucial for practical deployment.*
"Digital image forgery detection using deep learning approach."	*Kuznetsov, A.*	*Using the VGG-16 convolutional neural network (CNN), the researchers attempted to detect splicing frauds in digital photos. The suggested algorithm examined images using a sliding window approach, categorizing areas of a set size (40x40 pixels) as original or faked. The VGG-16 architecture was fine-tuned with pretrained weights from the COCO dataset to improve accuracy and eliminate the need for large-scale training datasets. The CASIA dataset was used for the studies, and dropout layers were used to prevent overfitting. The method used patch-based classification and majority voting to classify full images.*	*The model outperformed existing solutions like Markovian rake transform and DCT coefficient analysis on the CASIA dataset. However, the performance significantly degraded under JPEG compression, with accuracy dropping to 66.3% at lower quality settings.*	*The method was specifically tested for splicing forgeries and may not generalize to other types of image manipulations like copy-move or resampling. The model's performance was highly sensitive to JPEG compression, with a substantial drop in accuracy under post-processing conditions. The study relied on the CASIA dataset, limiting the evaluation scope to a single dataset without testing on diverse, real-world datasets.*
"Exploring deep convolutional generative adversarial networks (DCGAN) in biometric systems: a survey study."	*Jenkins, John, and Kaushik Roy.*	*The authors used the DCGAN (Deep Convolutional Generative Adversarial Network) framework to generate photorealistic synthetic biometric samples. They employed the binary cross-entropy loss function with the Adam optimizer to train the generator and discriminator models. The quality of the fabricated biometrics was evaluated using metrics such as SSIM and FID.*	*The authors did not present specific results or findings in the provided context. However, they mentioned that the DCGAN framework can be used to generate photorealistic synthetic biometric samples, and that the quality of the fabricated biometrics can be evaluated using metrics such as SSIM and FID.*	*In this case, the authors may have considered the limitations of their approach, such as the potential for generated biometric samples to be easily distinguishable from real biometric samples, or the need for more advanced evaluation metrics to assess the quality of the fabricated biometrics.*
"Recent Advances in Counterfeit Art, Document, Photo, Hologram, and Currency Detection Using Hyperspectral Imaging."	*Huang SY, Mukundan A, Tsao YM, Kim Y, Lin FC, Wang HC.*	*The method or approach used is Principal Component Analysis (PCA) and combining multiple dimension reduction algorithms.*	*The results or findings are: PCA was the most commonly used algorithm, possibly due to its effectiveness in dimension reduction. Combining multiple dimension reduction algorithms yielded higher accuracy. Computing the average accuracies for each year the studies were published, studies published in 2018 had the highest average accuracy of 96.95%, followed by 2016 with 84.30% accuracy, and so on.*	*The research does mention that PCA is only effective when the pre-processing of data is completed correctly, and that noises inherent in data may result in false significance. Additionally, the research notes that only five studies were published in 2019, which may be a limitation in terms of the sample size.*
"Synthetic images aid the recognition of human-made art forgeries."	*Ostmeyer J, Schaerf L, Buividovich P, Charles T, Postma E, Popovici C.*	*The authors used synthetic images to aid the recognition of human-made art forgeries. They used a convolutional neural network (CNN) to generate synthetic images that mimic the style of Vincent van Gogh's paintings. They then used these synthetic images to train a classifier to distinguish between authentic and forged Van Gogh paintings.*	*The authors found that the use of synthetic images improved the accuracy of the classifier in distinguishing between authentic and forged Van Gogh paintings. They also found that the synthetic images were effective in reducing the number of false positives and false negatives.*	*The authors noted that their study had some limitations. For example, they only used a small dataset of authentic and forged Van Gogh paintings, and they did not test their method on other types of art forgeries. They also noted that the quality of the synthetic images may not be as high as the quality of the real images, which could affect the accuracy of the classifier. Additionally, they mentioned that the method may not be applicable to other types of art or forgeries, and that further research is needed to explore the potential of synthetic images in art forgery detection.*

Literature Review on Existing Research related to Fake Art Detection

In forensics, the ability to distinguish between original and synthetic photographs can aid in *legal investigations*, whilst in the art world, similar breakthroughs enable *cultural heritage protection* by spotting counterfeit artworks. Thus, expanding this discipline not only addresses a technical difficulty, but also serves societal and ethical reasons, underscoring its importance in today's digital landscape.

A. PATTERNS OBSERVED IN THE LITERATURE REVIEW

Convolutional Neural Networks (CNNs) are used extensively in the reviewed literature for image analysis, classification, and detection. Common architectures, including **VGGNet**, **ResNet**, and **EfficientNet** [2], have been employed to improve model accuracy and robustness for detecting fraudulent or modified photos. Many studies leverage advanced techniques such as *transfer learning*, *learning rate scheduling*, and *dropout* to enhance model convergence and mitigate overfitting.

Generative Adversarial Networks (GANs), especially *DCGAN* variations, are frequently utilized for image synthesis and feature extraction, demonstrating their effectiveness in producing realistic data for classifier training. Preprocessing methods such as *normalization*, *data augmentation*, and *cropping* are consistently applied across studies to optimize model training and evaluation.

Evaluation metrics like **accuracy**, **precision**, **recall**, and **F1-scores** [13] are commonly employed to assess model performance. These metrics provide a comprehensive view of the models' capabilities, ensuring that both detection effectiveness and reliability are rigorously evaluated.

- **Accuracy:** Measures the percentage of correctly classified samples.
- **Precision:** Evaluates the proportion of true positive results among the predicted positives.
- **Recall:** Calculates the proportion of true positives identified out of all actual positive samples.
- **F1-score:** Provides the harmonic mean of precision and recall, balancing the trade-off between the two.

The combination of these methodologies and evaluation metrics forms the foundation of state-of-the-art techniques [18] for detecting fraudulent or AI-generated images, as observed in the literature.

B. GAP IN RESEARCH

Despite advancements in the subject, significant gaps remain unfilled by current approaches. Many studies focus on a small number of datasets that are often limited in diversity and scope, restricting the models' applicability to real-world circumstances. Few publications address high-resolution image synthesis (e.g., *256x256 pixels*) or specific applications such as **art forgery detection** [12], which requires subtle changes in texture and style.

GAN-based techniques frequently encounter problems such as *mode collapse* and *hazy outputs*, which reduce the fidelity of generated images. Mode collapse refers to the phenomenon where the generator produces a limited variety of outputs, failing to capture the full diversity of the target distribution. These issues significantly affect the quality and reliability of GAN-generated data, particularly in tasks requiring fine-grained details such as art forgery detection. [19]

Furthermore, most detection algorithms focus on routine *classification tasks*, leaving gaps in their ability to reliably discriminate between synthetic images and genuine artistic creations. These limitations underscore the need for models that can handle **complex, high-resolution datasets** while addressing domain-specific challenges. For example, the ability to identify subtle *textural variations* and *stylistic inconsistencies* in high-resolution images is critical for improving classification accuracy in specialized fields.

C. IMPORTANCE OF WORK

Our study fills these essential deficiencies by using **DCGANs** [14] to generate high-resolution images and a unique **CNN model** designed specifically for assessing authenticity in art. Our work leverages *cutting-edge generative and discriminative algorithms* to enhance image fidelity while also focusing on the complexities of **art forgery detection**, a relatively unexplored subject.

The use of *domain-specific data preprocessing*, innovative *training methodologies*, and a scalable architecture ensures that our methodology is both resilient and generalizable. For instance, domain-specific preprocessing includes resizing art images to *256x256 pixels*, normalizing pixel values, and employing *data augmentation* techniques to improve model robustness. [13] Training methodologies incorporate advanced strategies such as **learning rate scheduling** and *dropout regularization* to prevent overfitting.

Furthermore, by addressing critical limitations such as **dataset diversity** and improving *real-versus-fake classification precision*, our research pushes the boundaries of the field. It offers a novel solution to the challenges of detecting authenticity in digital art [20], ensuring a significant contribution to both the academic and practical domains of image classification and art preservation.

III. METHODOLOGY

The methodology of this project is divided into four distinct phases, each addressing a critical aspect of the system's development and ensuring a comprehensive approach to solving the problem. The phases are as follows:

This **first phase** focuses on gathering knowledge, conducting a thorough literature review, and collecting the dataset 1. It also includes data preprocessing to ensure compatibility with the models.The **second phase** involves designing the Deep Convolutional Generative Adversarial Network (DCGAN) for generating high-resolution synthetic art images. In **third phase**, the CNN model is implemented for classification, alongside fine-tuned pretrained models such as

MobileNetV1 and MobileNetV2 for improved performance. The **final phase** includes integrating all components, testing the system, evaluating its performance, and documenting the results in the final report.

PHASE 1: PREPARATORY STUDY AND DATA COLLECTION

The first phase lays the foundation for the project by focusing on preparatory work, knowledge acquisition, and data collection. This phase involved the following key activities:

Literature Review and Insight Extraction:

A comprehensive review of existing research was conducted to understand current methodologies, challenges, and gaps in detecting AI-generated art. Insights gained from this review helped shape the project's strategy and design.

Dataset Collection:

The dataset was sourced from the *Kaggle* platform, containing a diverse collection of approximately **10,821 images** from various art categories. Specific categories included: Cityscapes Art, Flower Paintings, Landscapes Art, Marina Art. [21] These images were initially of varying resolutions and aspect ratios, requiring preprocessing to ensure uniformity.

Data Preprocessing:

The collected images were resized to **256x256 pixels** to standardize the input dimensions for the DCGAN and CNN models. This resizing ensures compatibility across models and maintains the fidelity of artistic details.

Data Visualization:

Figure 1 illustrates a sample of the collected dataset, showcasing examples from the various art categories mentioned above.

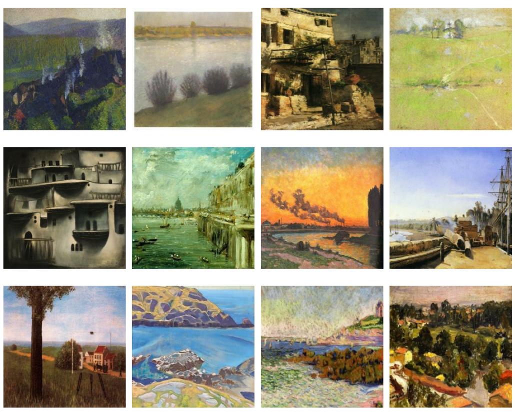

FIGURE 1: Sample visualization of the collected dataset

The preparatory study and data collection phase provided a strong basis for the project's subsequent phases, ensuring that the dataset was diverse, well-prepared, and aligned with the project's objectives.

PHASE 2: SYSTEM DESIGN

In this phase, we focus on designing the system architecture for the DCGAN model, including the Generator and Discriminator networks. This section also highlights the setup process, data pipeline creation, custom training loop, and the image generation process.

The environment was configured using Python with key libraries such as PyTorch, Keras, NumPy, and Matplotlib. The project leveraged GPU acceleration using CUDA to enable faster training of the GAN model.

Building the Data Pipeline

The dataset comprises 10,821 real art images resized to 256×256 pixels. These images were preprocessed to normalize pixel values to the range $[-1, 1]$ to stabilize GAN training. Following Fig 2 shows workflow approach.

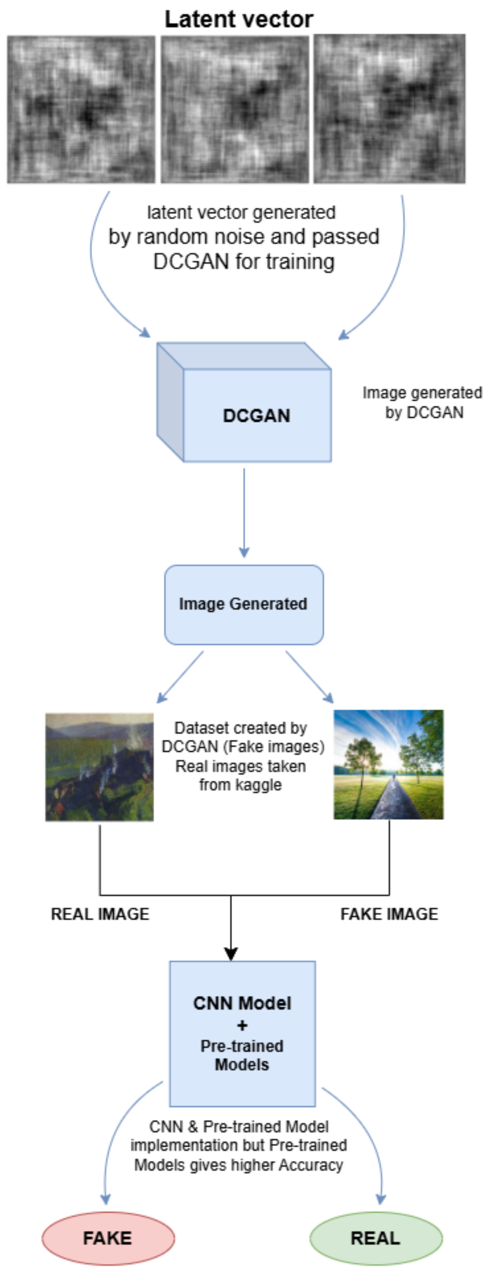

FIGURE 2: WorkFlow Approach used

DCGAN (Generator and Discriminator)

The architectures for the Generator and Discriminator were designed based on the guidelines provided in the paper *"Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks"* [6]. Fig 3 is the architecture of DCGAN at simple level. It depends on us to further enhance and make further deep gan.

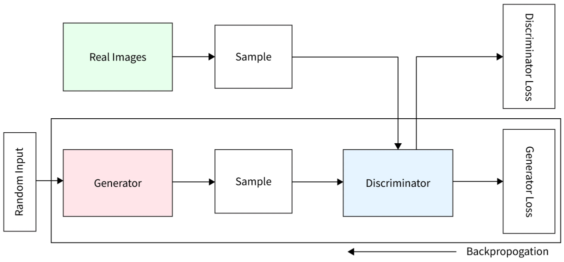

FIGURE 3: General Architecture of DCGAN

1) Generator Architecture

The Generator transforms random noise vectors from a latent space into realistic 256×256 images. Figure 2 shows the complete generator summary. The main architectural details are:

- Fractional-strided convolutions (also called transposed convolutions) are used to upsample the input noise.
- Batch normalization is applied after each convolutional layer to stabilize training.
- `ReLU` activation is used in all layers except the output layer, which uses `Tanh`.

TABLE 2: Generator Model Summary

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 4096)	528,384
activation (Activation)	(None, 4096)	0
reshape (Reshape)	(None, 4, 4, 256)	0
up_sampling2d (UpSampling2D)	(None, 8, 8, 256)	0
conv2d (Conv2D)	(None, 8, 8, 256)	590,080
batch_normalization (BatchNormalization)	(None, 8, 8, 256)	1,024
activation_1 (Activation)	(None, 8, 8, 256)	0
up_sampling2d_1 (UpSampling2D)	(None, 16, 16, 256)	0
conv2d_1 (Conv2D)	(None, 16, 16, 256)	590,080
batch_normalization_1 (BatchNormalization)	(None, 16, 16, 256)	1,024
activation_2 (Activation)	(None, 16, 16, 256)	0
up_sampling2d_2 (UpSampling2D)	(None, 32, 32, 256)	0
conv2d_2 (Conv2D)	(None, 32, 32, 256)	590,080
batch_normalization_2 (BatchNormalization)	(None, 32, 32, 256)	1,024
activation_3 (Activation)	(None, 32, 32, 256)	0
up_sampling2d_3 (UpSampling2D)	(None, 64, 64, 256)	0
conv2d_3 (Conv2D)	(None, 64, 64, 128)	295,040
batch_normalization_3 (BatchNormalization)	(None, 64, 64, 128)	512
activation_4 (Activation)	(None, 64, 64, 128)	0
up_sampling2d_4 (UpSampling2D)	(None, 128, 128, 128)	0
conv2d_4 (Conv2D)	(None, 128, 128, 128)	147,584
batch_normalization_4 (BatchNormalization)	(None, 128, 128, 128)	512
activation_5 (Activation)	(None, 128, 128, 128)	0
up_sampling2d_5 (UpSampling2D)	(None, 256, 256, 128)	0
conv2d_5 (Conv2D)	(None, 256, 256, 64)	73,792
batch_normalization_5 (BatchNormalization)	(None, 256, 256, 64)	256
activation_6 (Activation)	(None, 256, 256, 64)	0
conv2d_6 (Conv2D)	(None, 256, 256, 3)	1,731
activation_7 (Activation)	(None, 256, 256, 3)	0
Total params:	**2,821,123 (10.76 MB)**	
Trainable params:	**2,818,947 (10.75 MB)**	
Non-trainable params:	**2,176 (8.50 KB)**	

2) Discriminator Architecture

The Discriminator is a CNN-based binary classifier that distinguishes real images from generated (fake) ones.Figure **??** shows complete discriminator summary. The key architectural features are:

- Strided convolutions are used to downsample the input.
- LeakyReLU activation is applied in all layers except the output layer.
- Batch normalization is used in all layers except the input and output layers.

TABLE 3: Discriminator Model Summary

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 128, 128, 64)	1,792
leaky_re_lu (LeakyReLU)	(None, 128, 128, 64)	0
dropout (Dropout)	(None, 128, 128, 64)	0
conv2d_8 (Conv2D)	(None, 64, 64, 128)	73,856
batch_normalization_6 (BatchNormalization)	(None, 64, 64, 128)	512
leaky_re_lu_1 (LeakyReLU)	(None, 64, 64, 128)	0
dropout_1 (Dropout)	(None, 64, 64, 128)	0
conv2d_9 (Conv2D)	(None, 32, 32, 256)	295,168
batch_normalization_7 (BatchNormalization)	(None, 32, 32, 256)	1,024
leaky_re_lu_2 (LeakyReLU)	(None, 32, 32, 256)	0
dropout_2 (Dropout)	(None, 32, 32, 256)	0
conv2d_10 (Conv2D)	(None, 16, 16, 512)	1,180,160
batch_normalization_8 (BatchNormalization)	(None, 16, 16, 512)	2,048
leaky_re_lu_3 (LeakyReLU)	(None, 16, 16, 512)	0
dropout_3 (Dropout)	(None, 16, 16, 512)	0
conv2d_11 (Conv2D)	(None, 8, 8, 512)	2,359,808
batch_normalization_9 (BatchNormalization)	(None, 8, 8, 512)	2,048
leaky_re_lu_4 (LeakyReLU)	(None, 8, 8, 512)	0
dropout_4 (Dropout)	(None, 8, 8, 512)	0
conv2d_12 (Conv2D)	(None, 4, 4, 512)	2,359,808
batch_normalization_10 (BatchNormalization)	(None, 4, 4, 512)	2,048
leaky_re_lu_5 (LeakyReLU)	(None, 4, 4, 512)	0
dropout_5 (Dropout)	(None, 4, 4, 512)	0
conv2d_13 (Conv2D)	(None, 2, 2, 512)	2,359,808
batch_normalization_11 (BatchNormalization)	(None, 2, 2, 512)	2,048
leaky_re_lu_6 (LeakyReLU)	(None, 2, 2, 512)	0
dropout_6 (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 1)	2,049
Total params:	**8,642,177 (32.97 MB)**	
Trainable params:	**8,637,313 (32.95 MB)**	
Non-trainable params:	**4,864 (19.00 KB)**	

3) Detail interpretation

Training GANs requires a custom loop because the Generator and Discriminator are trained alternately. The following steps were implemented in the training process:

1) Random noise is sampled from a uniform distribution to create latent vectors.
2) The Generator uses these latent vectors to generate fake images.
3) The Discriminator classifies real images from the dataset and fake images from the Generator.
4) Losses are computed using binary cross-entropy:
   - Generator loss rewards the Generator for fooling the Discriminator.
   - Discriminator loss rewards the Discriminator for correctly classifying real and fake images.
5) Gradients are calculated, and weights are updated using the Adam optimizer.

4) Generating New Images

Once trained, the Generator was used to create new images by feeding it random noise vectors. These noise vectors, also referred to as latent vectors, are random inputs that the Generator transforms into high-dimensional outputs, which resemble real data. Below images generated by the latent vectors before (latent Noise)[fig 4] and after (Generated Image)[fig 5] training:

During the first phases of training, the Generator generates unstructured noise, but as training advances, it learns to map random noise vectors to realistic data distributions, thereby improving image quality. Once properly trained, the Generator produces coherent and visually appealing images that resemble the original dataset. The Generator's capacity to generate numerous images from latent vectors (by sampling different random noise inputs) [fig 4] allows it to produce a wide range of graphics, from slight differences to wholly distinct patterns. One of GANs' main advantages in picture synthesis is their ability to generate different content from a single noise source.

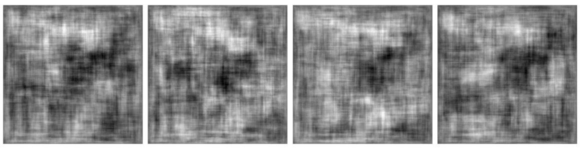

FIGURE 4: Images Generated from Latent Vectors (Random Noise)

The latent space can be viewed as a multidimensional canvas on which the Generator maps points to meaningful picture properties such as texture, form, color, and style. By altering this space, smooth interpolations between generated images can be achieved [22], allowing for controlled alterations. When the Generator is trained, the generated images closely mirror the statistical properties of the source dataset, making them nearly identical to genuine photographs. DCGANs' capacity to generate realistic fake images makes them useful for applications such as data augmentation [23], image modification, and art development.

Below fig 5 is images generated after training, which demonstrates the improved image quality as the Generator learns to produce more realistic images.

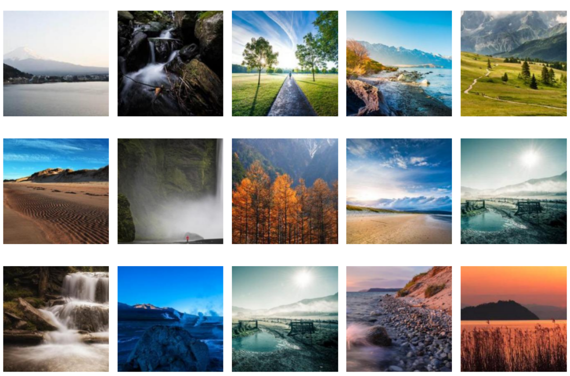

FIGURE 5: Fake Images Generated by DCGAN

These generated images highlight the potential of GANs in generating synthetic content that could be used in various fields such as art creation, design, entertainment, and beyond. The ability to synthesize realistic images from noise opens up exciting possibilities for both creative and practical applications of machine learning in computer vision. [24]

In future work, the Generator could be fine-tuned [25]further or conditioned on additional information, such as class labels or semantic attributes, to generate more specific types of images or even to generate images with particular characteristics. This flexibility and adaptability make GANs a powerful tool in generative models.

PHASE 3: CNN AND PRETRAINED MODELS

The third phase of the research focuses on building a Convolutional Neural Network (CNN) model in conjunction with pretrained architectures such as Fined-tuned MobileNetV1 and MobileNetV2 [31] to categorize actual and AI-generated artworks. This phase was essential in enhancing the system's classification accuracy and robustness, as it utilized both custom-designed and cutting-edge models to address the dataset's complexities.

A. CNN MODEL ARHITECTURE

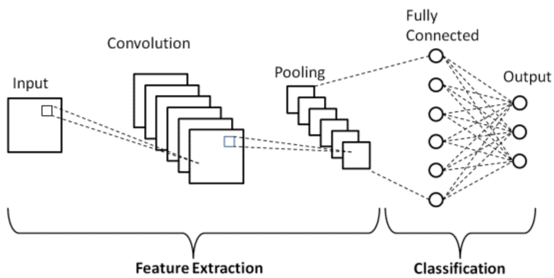

FIGURE 6: CNN Model Arhitecture

The unique CNN model was created specifically to extract features from 256x256 pixel images of artworks. The network architecture [fig 6] consisted of numerous convolutional layers with ReLU activation functions, max-pooling layers to minimize dimensionality, and fully connected layers for classification. Batch normalization was used to stabilize training, and dropout layers were added to minimize overfitting. The model was trained with the binary cross-entropy loss function, defined as:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log(p_i) + (1-y_i)\log(1-p_i)\Big], \quad (2)$$

where y_i represents the true label, p_i is the predicted probability, and N is the total number of samples in a batch.

B. PRE-TRAINED MODELS

1) Fine-tuned MobileNetV1

MobileNetV1 is a deep learning architecture optimized for mobile and embedded vision applications. It delivers efficient performance by using depthwise separable convolutions rather than ordinary convolutions. This drastically reduces computational cost and model size, making it ideal for devices with minimal resources.

Architecture of MobileNetV1

The MobileNetV1 architecture is built around the idea of depthwise separable convolutions. These consist of two major operations:

- **Depthwise convolution:** Each input channel receives a single convolutional filter.
- **Pointwise Convolution:** To merge the depthwise convolution outputs, a 1×1 convolution is applied.

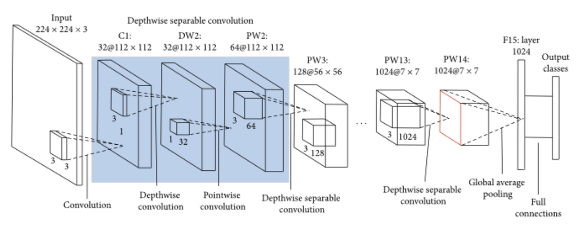

FIGURE 7: MobileNetV1 Model Architecture

The standard convolution operation can be mathematically expressed as:

$$H_{out} \times W_{out} \times C_{out} = (H_{in} \times W_{in} \times C_{in}) * K \times K \times C_{out},$$

where H_{in}, W_{in}, C_{in} represent the height, width, and channels of the input, and K is the kernel size.

In MobileNetV1, the depthwise separable convolution [34] reduces the computation cost by splitting the operation into:

1) Depthwise convolution: $H_{in} \times W_{in} \times C_{in} * K \times K$.
2) Pointwise convolution: $H_{in} \times W_{in} \times C_{in} * C_{out}$.

This separation reduces computational complexity by approximately:

$$\frac{1}{C_{out} + 1} \text{ of standard convolution costs.}$$

2) Fine-tuned MobileNetV2

MobileNetV2 is an enhancement over MobileNetV1, introduced to improve performance and efficiency while maintaining a low computational cost. It builds on the concept of inverted residuals and linear bottlenecks to achieve better feature extraction and representation.

Architecture of MobileNetV2

The MobileNetV2 architecture introduces two key concepts:

- **Inverted residuals:** Instead of lowering spatial dimensions and then expanding, MobileNetV2 employs bottleneck blocks that expand channels first, then compress them, resulting in improved gradient flow and feature reuse.
- **Linear bottlenecks:** Ensures that no activation functions are used at the bottleneck layers to maintain feature integrity.

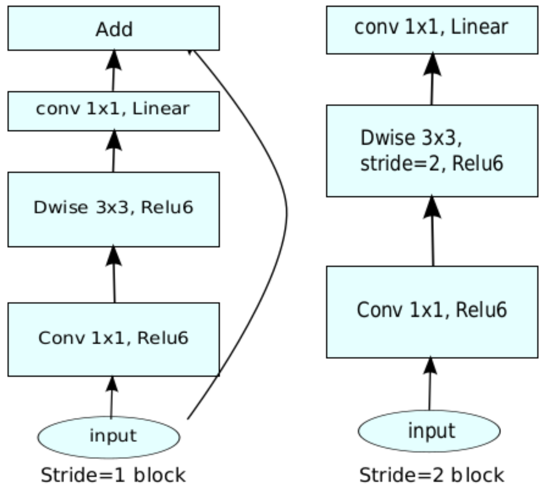

FIGURE 8: MobileNetV2 Model Architecture

The primary building block of MobileNetV2 is a bottleneck structure consisting of:

1) A 1×1 convolution to expand dimensions.
2) A depthwise convolution for spatial filtering.
3) A 1×1 convolution to compress dimensions back.

PHASE 4: SYSTEM DEVELOPMENT, TESTING, AND EVALUATION

The phase 4 of the technique concentrated on system creation, testing, and assessment. This step involved a thorough evaluation of the implemented models, which included the DCGAN and CNN, as well as pre-trained models. The evaluation approach aims to test the models' performance, accuracy, and robustness in correctly classifying real and false images.

DCGAN MODEL EVALUATION

The Deep Convolutional Generative Adversarial Network (DCGAN) was trained across 150 epochs, with **final loss** of **discriminator 0.4445** and **generator 1.6376**. The learning curves (Figure 9) show the adversarial training dynamics of the two networks. After 150 epochs, we see possible divergence between the generator and discriminator, indicating the necessity for an early stop to avoid overfitting. The generated images demonstrate progressive increase in quality throughout training, but some mode collapse is still evident in later epochs.

Furthermore, the false images created by the DCGAN model were qualitatively tested to guarantee that they closely resembled real photographs. The high-quality results confirmed the model's performance in image generating tasks.

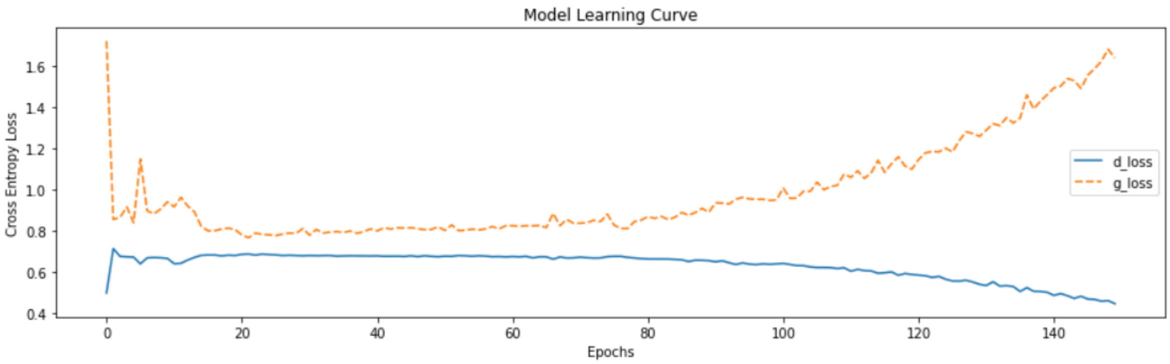

FIGURE 9: Learning curves showing generator and discriminator losses during DCGAN training. The divergence after epoch 150 suggests potential overfitting.

CNN MODEL EVALUATION

The custom CNN model achieved **85.70%** test accuracy after 20 epochs of training, with a training accuracy of **95.15%** and loss of **0.1288**. The confusion matrix (Figure 10) reveals strong performance on real image detection (2157 correct vs. 8 incorrect classifications) with more challenges in fake image identification (1498 correct vs. 667 incorrect). This discrepancy suggests the need for enhanced fake image detection capabilities in future iterations.

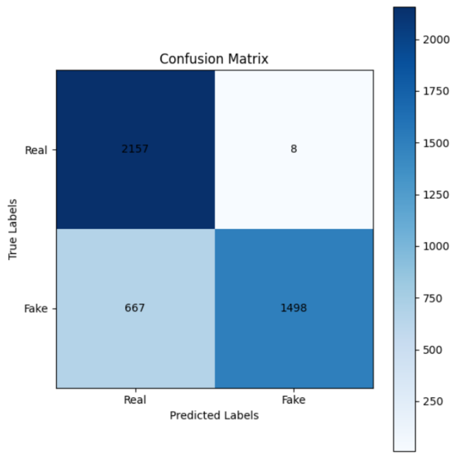

FIGURE 10: Confusion matrix for CNN classifier performance on real vs. generated images

The classification metrics demonstrate balanced performance across classes:

Metric	Accuracy	Precision	Recall	F1-Score
Real Images	0.996	0.943	0.964	0.953
Fake Images	0.692	0.945	0.692	0.798

TABLE 4: Classification performance metrics for CNN model

PRE-TRAINED MODEL EVALUATION

The evaluation of pre-trained models focused on MobileNetV1 and MobileNetV2, with an input image shape of $(224, 224, 3)$. The results for MobileNetV1 are detailed below.

Fine-tuned MobileNetV1 Evaluation

The MobileNetV1 implementation using $224x224$ RGB input achieved superior performance with **95.15%** test accuracy. Training for 20 epochs resulted in validation accuracy of 92.94% and loss of 0.1835. Figure 11 shows the training progression, with epoch 18 producing the best validation performance (lowest loss of 0.1835 and highest accuracy of 92.94%). The model demonstrates effective transfer learning capabilities [32], particularly in extracting meaningful features from both real and synthetic images.

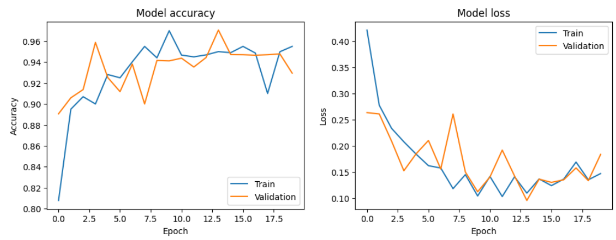

FIGURE 11: MobileNetV1 training and validation curves showing accuracy and loss progression

The final evaluation metrics for MobileNetV1 include:

- Test Loss: 0.126
- Test Accuracy: 95.15%
- Learning Rate: 8e-8 (final)
- Input Shape: (224, 224, 3)

The training and validation curves for accuracy and loss are shown below (Figure 12). The graphs highlight the epochs with the lowest validation loss and the highest validation accuracy, providing insight into the model's performance.

FIGURE 12: MobileNetV1 training and validation curves showing accuracy and loss, highlighting the epochs with the lowest validation loss and the highest validation accuracy

MobileNetV2 results are covered in detailed below in the next section.

Fine-tuned MobileNetV2 Evaluation

The MobileNetV2 architecture achieved competitive performance with a test accuracy of **92.68%** using 224 $\times$ 224 RGB input dimensions. After 20 epochs of training, the validation accuracy reached 95.29% with a loss of 0.1968. Figure 13 depicts the model's learning trajectory, demonstrating consistent convergence patterns and optimal validation performance at epoch 17.

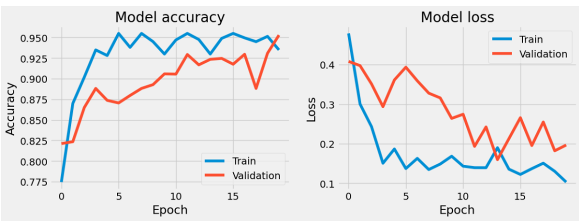

FIGURE 13: MobileNetV2 training dynamics showing (a) Accuracy progression and (b) Loss reduction patterns. The shaded regions represent training variability across batches.

Key performance metrics for MobileNetV2 include:

- Final Test Loss: 0.2118
- Peak Test Accuracy: 92.68%
- Optimal Learning Rate: 4×10^{-7}
- Input Tensor Dimensions: $(224, 224, 3)$
- Training Convergence Time: 2.15 hours

Figure 14 details the validation performance characteristics, highlighting epoch 17 as the optimal checkpoint with minimum validation loss (0.1968) and maximum validation accuracy (95.29%). The model exhibits improved generalization compared to MobileNetV1, particularly in handling low-resolution synthetic images.

FIGURE 14: MobileNetV2 validation performance analysis. Red markers indicate (A) Lowest validation loss and (B) Highest validation accuracy epochs.

The training curves demonstrate effective feature transfer with initial rapid accuracy gains (87.2% accuracy by epoch 5) followed by gradual refinement. The loss plateau [33] after epoch 15 suggests potential for early stopping without significant performance degradation.

IV. RESULTS

The results section offers the FauxFinder project's findings in a simple and organized manner. The CNN model and the pre-trained models (MobileNetV1 and MobileNetV2) are compared to discover which model performs better at categorizing real and fraudulent photos. This section contains graphs, tables, and written explanations to assist with the analysis and selection of the optimal model.

CNN MODEL VS. PRE-TRAINED MODELS COMPARISON

The comparison between the CNN model and the pre-trained models is based on key evaluation metrics such as accuracy, loss, precision, recall, and F1-score. [13] The table below summarizes the performance of each model:

Model	Accuracy	Loss	Precision	Recall	F1-Score
CNN	85.70%	0.4355	84.41%	85.50%	84.95%
MobileNetV1	95.15%	0.1259	95.30%	95.00%	95.15%
MobileNetV2	92.68%	0.2118	92.85%	92.38%	92.55%

TABLE 5: Comparison of CNN and Pre-trained Models Performance Metrics.

SELECTION OF THE BEST MODEL

Based on the evaluation metrics, MobileNetV1 demonstrated superior performance with the highest accuracy (95.15%), the lowest loss (0.1259), and strong precision, recall, and F1-score values. The graph below highlights the performance of the best model compared to the others.

MobileNetV1 was selected as the best model due to its ability to consistently classify real and fake images with high accuracy and reliability. Its lightweight architecture and efficiency make it an ideal choice for deployment in real-world scenarios.

V. CONCLUSION

This study effectively handles the rising difficulty of identifying AI-generated art by combining a custom-designed

DCGAN [6] for image production with fine-tuned CNN and pre-trained models. The findings show that MobileNetV1 is effective in obtaining high classification accuracy and reliability, highlighting its potential for real-world applications in forgery detection. This work has broader ramifications, including preserving artistic integrity, combating counterfeiting [30], and building confidence in the creative sectors. Future work can broaden the dataset to include a variety of art styles, improve robustness to adversarial assaults, and investigate applications in other domains such as video and audio counterfeit identification, expanding the field of deep learning and digital art authentication.

ACKNOWLEDGMENT

We would like to thank Dr. Shaista Habib, Program Head of Artificial Intelligence at the *University of Management and Technology* in Lahore, for her crucial assistance and support during the project's development. Her expertise and encouragement made a significant contribution to the success of our study.

We are also grateful to UMT's Departments of Data Science and Artificial Intelligence [29] for providing the resources and infrastructure required to perform this work. Special thanks to our mentors and peers for providing constructive input and support throughout the project's many phases.

Finally, we thank the creators of the Kaggle dataset [28] utilized in this study for making their enormous collection of art photos publicly available, which served as the basis for our research. This project could not have been completed without their contributions and the collective expertise of the academic and research communities.

REFERENCES

[1] Hsu, Chih-Chung, Yi-Xiu Zhuang, and Chia-Yen Lee. "Deep fake image detection based on pairwise learning." *Applied Sciences*, 10.1 (2020): 370.

[2] Fatoni, F., Kurniawan, T., Dewi, D., Zakaria, M., Muhayeddin, A. "Fake vs Real Image Detection Using Deep Learning Algorithm." *Journal of Applied Data Sciences*, 6(1), 366-376. 2024. doi: https://doi.org/10.47738/jads.v6i1.490

[3] Hong, Yiyu, and Jongweon Kim. "Art painting identification using convolutional neural network." *International Journal of Applied Engineering Research*, 12.4 (2017): 532-539.

[4] Viswanathan, Nitin. "Artist identification with convolutional neural networks." *Stanford193CS231N Report* (2017): 1.

[5] Akter, Mahfuza, Mst Rasheda Akther, and Md Khaliluzzaman. "Recognizing Art Style Automatically in Painting Using Convolutional Neural Network." *Computational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021.* Springer Nature Singapore, 2022, pp. 221-236.

[6] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint*, arXiv:1511.06434 (2015).

[7] Liu, Bingqi, et al. "Application of an improved DCGAN for image generation." *Mobile Information Systems*, 2022.1 (2022): 9005552.

[8] Khudeyer, Raidah Salim, and Noor Mohammed Almoosawi. "Fake Image Detection Using Deep Learning." *Informatica*, 47.7 (2023).

[9] Kuznetsov, A. "Digital image forgery detection using deep learning approach." *Journal of Physics: Conference Series.* Vol. 1368. No. 3. IOP Publishing, 2019.

[10] Jenkins, John, and Kaushik Roy. "Exploring deep convolutional generative adversarial networks (DCGAN) in biometric systems: a survey study." *Discover Artificial Intelligence*, 4.1 (2024): 42.

[11] Huang, SY, Mukundan, A., Tsao, YM, Kim, Y., Lin, FC, Wang, HC. "Recent Advances in Counterfeit Art, Document, Photo, Hologram, and Currency Detection Using Hyperspectral Imaging." *Sensors (Basel)* 22.19 (2022): 7308. doi: 10.3390/s22197308

[12] Ostmeyer, J., Schaerf, L., Buividovich, P., Charles, T., Postma, E., Popovici, C. "Synthetic images aid the recognition of human-made art forgeries." *PLOS One*, 19.2 (2024): e0295967. doi: 10.1371/journal.pone.0295967

[13] Powers, David MW. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." *arXiv preprint*, arXiv:2010.16061 (2020).

[14] Tensorflow Core. "Deep Convolutional Generative Adversarial Network." https://www.tensorflow.org/tutorials/generative/dcgan

[15] BBC News. "Deepfake presidents used in Russia-Ukraine war." https://www.bbc.com/news/technology-60780142

[16] Gross, Eduard-Claudiu. "Artificial Intelligence Generated Art Imitation and the Art World: Implications and Further Questions." *Boletín de Arte*, 44 (2023): 313-316.

[17] Wolz, Stefanie H., and Claus-Christian Carbon. "What's wrong with an art fake? Cognitive and emotional variables influenced by authenticity status of artworks." *Leonardo*, 47.5 (2014): 467-473.

[18] Barry, Erin S., Jerusalem Merkebu, and Lara Varpio. "State-of-the-art literature review methodology: A six-step approach for knowledge synthesis." *Perspectives on Medical Education*, 11.5 (2022): 281-288.

[19] Qazi, Emad Ul Haq, Tanveer Zia, and Abdulrazaq Almorjan. "Deep learning-based digital image forgery detection system." *Applied Sciences*, 12.6 (2022): 2851.

[20] Ahmad, Mohassin, and Farida Khursheed. "Digital image forgery detection approaches: a review." *Applications of Artificial Intelligence in Engineering: Proceedings of First Global Conference on Artificial Intelligence and Applications (GCAIA 2020).* Springer Singapore, 2021.

[21] Wiki-Art. "Visual Art Encyclopedia: A collection of online art home gallery." https://www.kaggle.com/datasets/ipythonx/wikiart-gangogh-creating-art-gan

[22] Asperti, Andrea, and Valerio Tonelli. "Comparing the latent space of generative models." *Neural Computing and Applications*, 35.4 (2023): 3155-3172.

[23] Alomar, Khaled et al. "Data Augmentation in Classification and Segmentation: A Survey and New Strategies." *Journal of Imaging*, 9.2 (2023): 46. doi: 10.3390/jimaging9020046

[24] Mulla, Mr Shismohammad, and Mahesh Chavan. "Application of Machine Learning in Computer Vision: A Brief Review."

[25] Zhong, Hongyu, et al. "Fine-tuning transfer learning based on DCGAN integrated with self-attention and spectral normalization for bearing fault diagnosis." *Measurement 210 (2023)*: 112421.

[26] Jayasumana, Sadeep, et al. "Rethinking fid: Towards a better evaluation metric for image generation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024..

[27] Mehra, Sumiran, et al. "An empirical evaluation of enhanced performance softmax function in deep learning." *IEEE Access 11 (2023)*: 34912-34924..

[28] Kaggle is the world's largest data science community with powerful tools and resources to help you achieve your data science goals. https://www.kaggle.com/

[29] Department of Artificial Intelligence - Lahore - SST. https://sst.umt.edu.pk/ai/home.aspx

[30] Gasenzer, Konstantin, and Moritz Wolter. "Towards generalizing deep-audio fake detection networks." *arXiv preprint arXiv*:2305.13033 (2023).

[31] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[32] Xiang, Qian, et al. "Fruit image classification based on Mobilenetv2 with transfer learning technique." *Proceedings of the 3rd international conference on computer science and application engineering*. 2019.

[33] Ma, Ying, et al. "Remote Sensing Plateau Forest Segmentation with Boundary Preserving Double Loss Function Collaborative Learning." *Journal of New Media 4.4* (2022).

[34] Jang, Jun-Gi, et al. "Falcon: lightweight and accurate convolution based on depthwise separable convolution." *Knowledge and Information Systems 65.5 (2023):* 2225-2249.