AI vs. Human Generated Images: Interpretable Binary Classification with Convolutional Networks

Nikolaos Karantaglis

1 INTRODUCTION

In recent times, the explosion of generative AI has led to an increasing number of people using large generative models for content creation. There is a significant trend in generating images for both personal and commercial purposes. Image generation models have become more powerful than ever, producing highly plausible content. Consequently, there arises a need to differentiate between AI-generated and human-generated images. While individuals may find this distinction seemingly straightforward, achieving clear interpretability of their decision-making process can be challenging. It is crucial to distinguish AI-generated images from those created by humans through a method that is fast, accurate, and explainable. This research aims to develop an efficient model that classifies images as either AI-generated or human-generated and provides an explainable basis for its decisions. In this work, we trained an end-to-end Convolutional Neural Network (CNN) binary classifier and employed the Integrated Gradients method to interpret the classifier's decisions. Additionally, to enhance the model's performance, we applied data transformation and image denoising techniques.

2 RELATED WORKS

A significant amount of effort has been made to address the challenge of distinguishing between human-generated and AI-generated images, much of which is not formally published but instead shared as code repositories. Among the published works, one notable study [1] introduced a uniquely constructed dataset named CIFAKE, which was derived by using the CIFAR dataset for human-generated images and employing an image generation model to produce images representing AI-generated counterparts. This approach facilitated a focused examination of binary classification with an interpretability perspective. However, this study primarily focused on dataset creation, employing relatively simple architectures to tackle this intricate problem. Moreover, it utilized images of 32x32 resolution, constrained by the CIFAR dataset's limitations. This approach is suboptimal as the task of distinguishing AI from human-generated images necessitates more detailed and higher-resolution imagery to capture nuanced image characteristics effectively. In contrast, our work leverages a different dataset with higher resolution and deploys more sophisticated models to discern more complex image features. Additionally, we incorporate advanced image denoising techniques, which are better suited to the task at hand.

3 PROPOSED METHOD

In this section, we describe the methodology adopted to develop an interpretable CNN model capable of classifying images as either human-made or AI-generated. Initially, we applied an image preprocessing strategy to convert images into a suitable format for neural network input. Subsequently, we focused on designing an optimal CNN architecture. Finally, we utilized the Integrated Gradients method as an interpretability technique to understand the classifier's decisions by identifying the image regions significantly influencing its judgments.

3.1 Wavelet Transform for Noise Reduction

Throughout our cycle of tuning, training, evaluation, and interpretability, we encountered instances where images contained noise, such as film grain or fur, leading to a higher misclassification rate, often labeling images as AI-generated. We hypothesize that this occurs because diffusion image generation models, like DALL-E, initiate with random noise to create an image based on a prompt, causing the classifier to mistake the noise pattern for AI-generated art, thus leading to confusion. To address this, we incorporated a wavelet transform technique [3] to diminish the noise in images.

The method utilizes discrete wavelet transforms (DWT) to effectively reduce image noise, separating content into frequency bands for precise noise elimination while retaining crucial details. Through decomposing images into approximation and detail coefficients, and selectively thresholding the latter, our approach minimizes noise without distorting the image, streamlining the denoising process with key steps and equations.

1. **Wavelet Decomposition**: The image I is decomposed into approximation (A) and detail (D) coefficients at level L using the discrete wavelet transform:

$$(A_L, D_L, D_{L-1}, \dots, D_1) = \mathrm{DWT}(I, \text{wavelet}, L) \tag{1}$$

where wavelet specifies the wavelet function and L the level of decomposition.

2. **Noise Estimation**: The standard deviation of the noise, σ, is estimated from the median absolute deviation (MAD) of the detail coefficients at the finest scale:

$$\sigma = \frac{\text{median}(|D_1 - \text{median}(D_1)|)}{0.6745} \tag{2}$$

3. **Thresholding**: A threshold T is computed using the universal thresholding rule, which is a function of σ and the size of the image (N):

$$T = \sigma\sqrt{2 \log N} \tag{3}$$

Detail coefficients smaller than T are reduced, aiming to remove noise while keeping significant details.

4. **Image Reconstruction**: The denoised image I_{denoised} is reconstructed from the modified coefficients:

$$I_{\text{denoised}} = \mathrm{IDWT}(A_L, \tilde{D}_L, \tilde{D}_{L-1}, \dots, \tilde{D}_1, \text{wavelet}, L) \tag{4}$$

where $\tilde{D}_i$ represents the thresholded detail coefficients.

This denoising strategy is particularly effective for images where noise and image details are mixed in the frequency domain. By applying this method, we enhance the classifier's ability to accurately differentiate between human-made and AI-generated images, facilitating more reliable and interpretable outcomes.

Ai, Auth,
2024.

3.2 CNN architecture

Our CNN architecture, designed for binary classification, is structured to incrementally extract and refine features through a series of convolutional and pooling layers before making a prediction. Initially, the model processes input images through four convolutional layers, each followed by ReLU activation for introducing non-linearity and max pooling for dimensionality reduction and spatial hierarchy. The convolutional layers progressively increase in depth, starting from 16 channels and doubling at each step up to 128 channels, allowing the model to capture a wide range of features from basic edges to more complex textures and patterns. After the convolutional stages, the model flattens the output and feeds it into a fully connected layer, which further processes the features before reaching the output layer. The output layer uses a sigmoid activation function to produce a probability, indicating the likelihood of the image being AI-generated. The model utilizes the Adam optimizer with a learning rate of 0.001 and is trained to minimize the binary cross-entropy loss over 8 epochs, balancing efficiency and performance in distinguishing between human-made and AI-generated images. Fig. 1 illustrates the 4Conv4Pool model proposed in this work

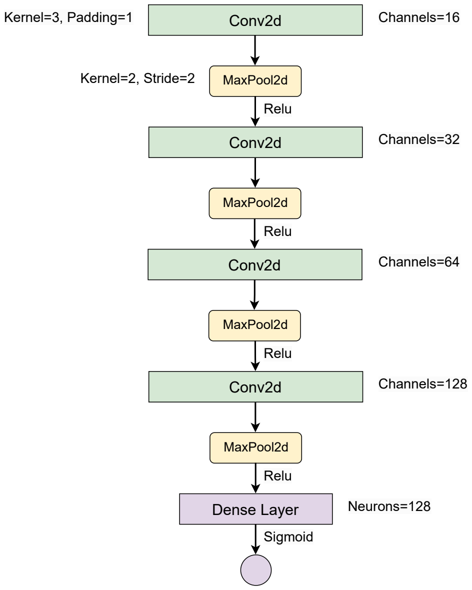

Figure 1: 4Conv4Pool Model

3.3 Interpretability

For enhancing the interpretability of our CNN model's decisions, we have adopted the Integrated Gradients [2] method from Captum's IntegratedGradients module [1]. This method is pivotal for shedding light on the specific areas within each image that significantly influence the model's classification outcome. Integrated Gradients achieves this by attributing the prediction output to the input features of the image, essentially highlighting the pixels that play a crucial role in distinguishing between human-made and AI-generated images.

Integrated Gradients calculates how each pixel influences the model's output by tracing the gradient from a baseline (often a black image) to the input image. This reveals the pixel contributions towards the model's decision, providing a clear, visual explanation of which image parts are most significant. This technique boosts transparency in model predictions, helping identify key decision drivers and potential biases.

4 EXPERIMENTAL EVALUATION

In this section, we describe the dataset utilized in our study, detail the experiments conducted to identify the optimal architecture, elaborate on the hyperparameter tuning process, and explore the impact of the image denoising process on our model's performance. Finally, we will discuss the outcomes of applying the interpretability method.

4.1 DALL-E Recognition Dataset

In this work, we utilized the DALL-E Recognition Dataset from Kaggle [2], which is comprised of two parts: images generated by AI image generation models such as DALL-E and Midjourney, and real images confirmed to be created by humans. The initial dataset contains 22,000 RGB images of various resolutions, with the smallest resolution being $224x224$. Fig. 2 illustrates some example images from the DALL-E Recognition Dataset.

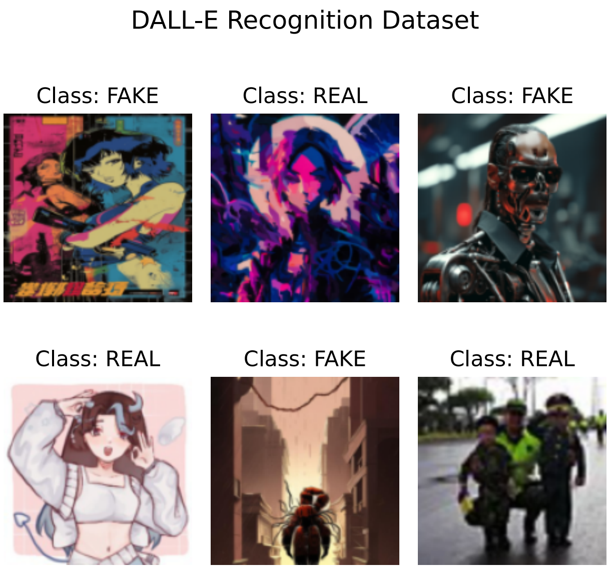

Figure 2: DALL-E Recognition Dataset Example

[1] https://captum.ai

[2] https://www.kaggle.com/datasets/superpotato9/dalle-recognition-dataset

4.1.1 Data Preprocessing. Due to constraints on memory capacity, we opted for a subset of 7,300 images. For consistency and to accommodate our system's limitations, all images in our study were resized to a resolution of 128x128, using the 3 channels (RGB). The dataset was divided into training and testing sets, with the training set containing 5,100 images (2,550 for each class) and the test set comprising 2,200 images (1,100 for each class). During the data loading phase, images were organized into batches of 64, resulting in a data tensor shape of $[64, 3, 128, 128]$. Here, the first dimension represents the batch size, the second dimension corresponds to the three channels of the RGB color model, and the final two dimensions denote the resolution of each image in the batch.

4.2 Evaluation of CNN Architectures

We divided the training dataset into training and validation sets, adhering to an 80:20 split for training and validation, respectively. Several experiments were conducted using six distinct architectures. Our approach was to begin with a simple architecture and incrementally introduce more layers and complexity in subsequent models. The models utilized are described below:

- **2Conv1Pool:** This model comprises 2 convolutional layers followed by a single max pooling layer, leading into a dense layer and culminating with an output layer.
- **3Conv2Pool:** Incorporates 3 convolutional layers interspersed with 2 max pooling layers, transitioning into a dense layer and an output layer.
- **3Conv3Pool:** Features 3 convolutional layers and 3 max pooling layers in sequence, followed by a dense layer and an output layer.
- **4Conv4Pool:** Consists of 4 convolutional layers and 4 max pooling layers, leading into a dense layer and concluding with an output layer.
- **4Conv4PoolBN:** Similar to the 4Conv4Pool model, this architecture includes 4 convolutional and 4 max pooling layers, but integrates batch normalization after each convolutional layer before proceeding to a dense layer and an output layer.
- **4Conv4PoolBND:** Builds on the 4Conv4PoolBN model by adding a dropout layer after the final pooling step to reduce overfitting, followed by a dense layer and an output layer.

The Table 1 presents the validation accuracy, precision, recall, and F1-score for each model at its most efficient training epoch accoring to F1-score. Fig. 3 depicts the F1-score metrics for all models across the epochs.

Table 1: Evaluation of CNN Architectures

Model	Accuracy	Precision	Recall	F1-Score
2Conv1Pool	0.629	0.588	0.895	0.710
3Conv2Pool	0.672	0.630	0.818	0.710
3Conv3Pool	0.724	0.738	0.717	0.726
4Conv4Pool	**0.727**	**0.706**	**0.793**	**0.746**
4Conv4PoolBN	0.667	0.626	0.864	0.725
4Conv4PoolBND	0.622	0.623	0.875	0.693

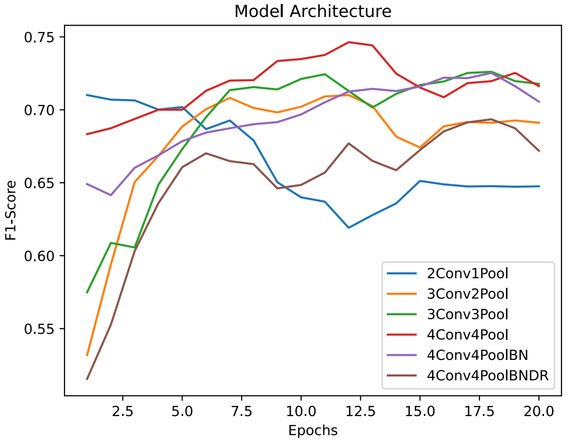

Figure 3: F1-Score on Validation set

We utilize the F1-score to identify the most efficient architecture, as the F1-score is a highly accurate metric that considers both types of errors—false positives and false negatives—rather than merely counting the number of incorrect predictions. As shown in Table 1 and Fig. 3, the most efficient model is the proposed 4Conv4Pool, which achieved an F1-score of 0.759 on the validation set during epoch 12. To better understand the performance of the proposed model, we compare all calculated metrics for both the training and validation sets. Fig. 4 illustrates this comparison.

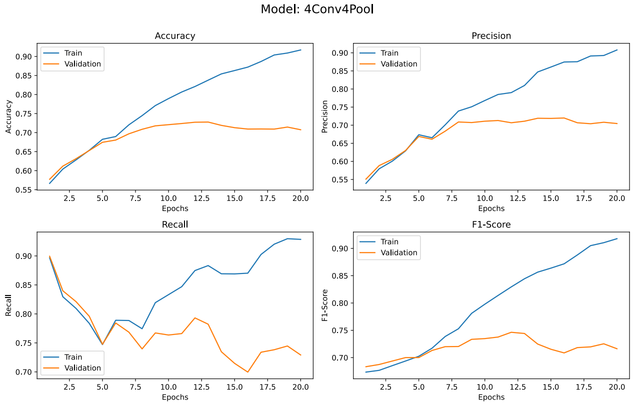

Figure 4: Training vs Validation Metrics

4.3 Optimizing Learning Rate

It is noteworthy that our attempt to prevent overfitting by incorporating a dropout layer into the 4Conv4PoolBND model was unsuccessful. For this reason, we conducted several experiments to identify the optimal learning rate and combat overfitting. Fig. 5 illustrates some of these experiments, showing the performance of our proposed model on the validation set with various learning rates.

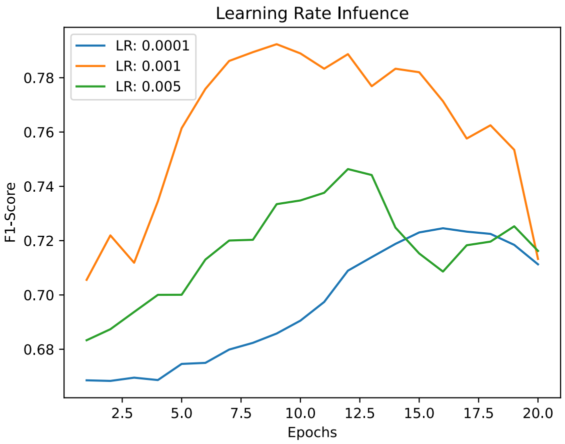

Figure 5: Evaluation of Learning Rate on Validation Set

As can be seen from the Fig. 5, the model underfits when the learning rate is set to 0.0001 and overfits prematurely when the learning rate is set to 0.005. The optimal learning rate was determined to be 0.001, with the best performance occurring at epoch 8. We also tried using a learning rate scheduler, but it did not make the model more efficient.

4.4 DWT Image Denoising Influence

In previous experiments, we identified the 4Conv4Pool model with a learning rate of 0.001 and an optimal performance at epoch 8 as the most efficient. Building on this foundation, we now aim to evaluate the use of Wavelet Transform for noise reduction. Fig. 6 illustrates the impact of the image denoising process on the test set. As shown in Fig. 6, the application of the image denoising process positively influenced the F1-Score.

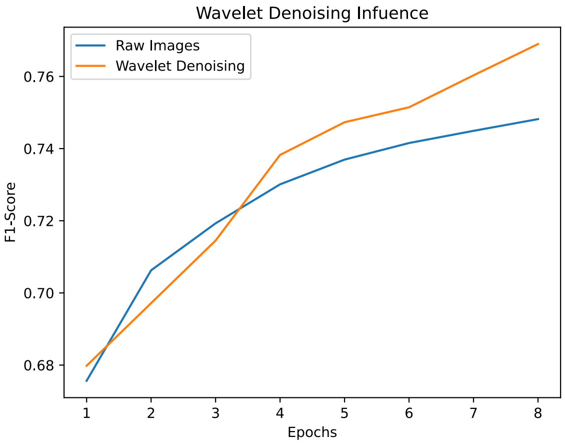

Figure 6: Evaluation of DWT Image Denoising on Test Set

Finally, the results for the proposed model, calculated using the optimal hyperparameters and DWT Image Denoising approach on both the train and test sets, are described in Table 2

Table 2: Summary of Model Performance Metrics

Set	Accuracy	Precision	Recall	F1-score
Train Set	0.819	0.776	0.895	0.831
Test Set	0.744	0.708	0.832	0.764

4.5 Interpretable Insights from CNN Decisions

In applying the Integrated Gradients technique to our CNN model, we gained detailed visual explanations for classifying images as ’REAL’ or ’FAKE’. The heatmaps, particularly from Fig. 7 and Fig. 8, reveal how the model discerns real images by focusing on specific features like texture and edge details. These attributes are crucial for the classification, as shown in the original images and their corresponding attribution heatmaps (Fig. 9 and Fig. 10). The clarity of these visualizations highlights the model’s reliance on subtle yet significant image features that differentiate real photographs from synthetic ones, underscoring the value of Integrated Gradients in making the model’s decision-making process transparent and understandable.

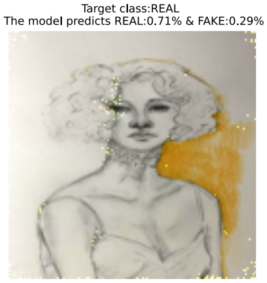

Figure 7: Overlayed Heatmap of Image 1

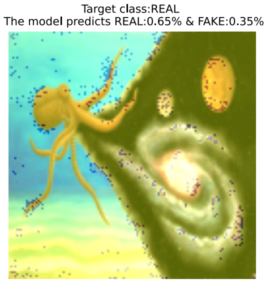

Figure 8: Overlayed Heatmap of Image 2

The heatmaps emphasize natural textural patterns, particularly the variances in lighting and shadow that are characteristic of real objects. These subtleties, such as gradations of light and slight edge imperfections, are less commonly found in AI-generated images, which often exhibit more uniform shading and cleaner transitions. For instance, Fig. 10 and its heatmap show that the model identifies authenticity through abrupt changes in texture and pattern—likely indicators of natural wear or environmental effects on photographic subjects. Such focal points, including complex patterns or noise inherent in photographic processes, are critical cues that the model uses to distinguish real images, highlighting its capability to recognize and evaluate finer details that are generally overlooked or smoothed out in synthetic imagery.

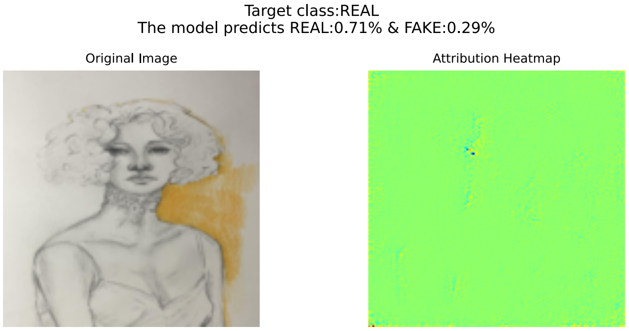

Figure 9: Original vs Heatmap of Image 1

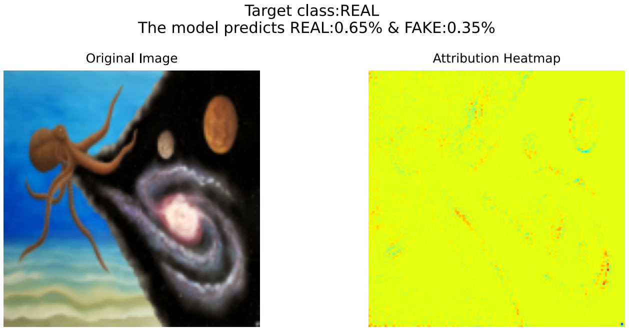

Figure 10: Original vs Heatmap Image 2

5 CONCLUSION

This study has demonstrated the effectiveness of our CNN model in distinguishing between real and AI-generated images, leveraging the advanced capabilities of Integrated Gradients to provide deep interpretative insights into the classification processes. Throughout our experiments, we observed that the model adeptly identifies and utilizes subtle, yet critical, features such as texture variations, edge irregularities, and lighting differences—features that are distinctly more pronounced in real-world imagery compared to their synthetic counterparts. The detailed visualizations provided by the heatmaps not only affirmed the model's accuracy but also enhanced our understanding of the underlying reasons for its decisions, ensuring a high degree of trust in its outputs. Moreover, our findings underscore the importance of model interpretability in AI applications, particularly in fields requiring reliable distinction between real and synthetic content. Future work will focus on refining these interpretative techniques and exploring their applicability across different datasets and in more complex scenarios, aiming to further enhance the transparency and reliability of AI-driven image analysis.

REFERENCES

[1] Jordan J. Bird and Ahmad Lotfi. 2023. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. arXiv:cs.CV/2303.14126

[2] Daniel Lundstrom, Tianjian Huang, and Meisam Razaviyayn. 2022. A Rigorous Study of Integrated Gradients Method and Extensions to Internal Neuron Attributions. arXiv:cs.LG/2202.11912

[3] Rajesh Patil. 2015. Noise Reduction using Wavelet Transform and Singular Vector Decomposition. *Procedia Computer Science* 54 (2015), 849–853. https://doi.org/10.1016/j.procs.2015.06.099