Identifying AI Generated Images: Using a Logistic Regression Model and Feature Extraction

Nidya Vargas
Department of Computer Science
University of Tennessee, Knoxville
Knoxville, USA
nvarges3@vols.utk.edu

Ahmad Tobasei
Department of Computer Science
University of Tennessee, Knoxville
Knoxville, USA
atobasei@vols.utk.edu

Kynnedy Armstrong
Department of Computer Science
University of Tennessee, Knoxville
Knoxville, USA
karmst32@vols.utk.edu

Erik Martinez
Department of Computer Science
University o f Tennessee, Knoxville
Knoxville, USA
ecruz8@vols.utk.edu

***Abstract*— The rise of AI's usage popularity has made it more difficult than ever to discern between whether an image is real or ai-generated. We have developed a project that implements a machine learning framework that uses preprocessing, normalization, and feature extraction to detect any key features found in AI vs human art.**

Keywords— AI, AI art detection, machine learning, AI vs. Human art

I. Introduction

A. Problem

There has been a rising issue with AI getting better at mimicking artwork created by humans, bringing forth a question of ethics: the dilemma of AI art stealing from artists, and lessening the quality and creative aspect of human-made art [1]. This project aims to use machine learning techniques to distinguish between the characteristics of AI generated art and human-created art. Our approach to this issue was to use feature extraction to methods to detect key differences between what is found in AI versus human art, such as symmetry, fine line details, and color distributions [2].

B. Project Goals

The goal of this project is to investigate whether a simpler machine learning approach can distinguish AI-generated artwork from human-created artwork using a limited dataset and hand-engineered features. We leverage a publicly available image dataset from a recent Kaggle competition, **AI vs. Human-Generated Images**, which provides a curated collection of AI-generated and authentic (human-made) artworks. This includes developing a logistic regression model with an increased accuracy percentage, as the current one is at 53%, pinpoint which features should be prioritized in the feature extraction, and find which features best discern the difference between AI vs humans.

C. Dataset

The dataset consists of digital artwork images in two categories: AI-generated art and human-created art. In total there are 18,618 images (roughly balanced between the two classes). For this project we sampled a subset of 5,000 images (approximately 2,500 per class) to use for model training, validation, and testing, given our computational constraints. The images come from a Kaggle competition dataset designed for the AI vs. human image classification challenge [6]. The artwork spans a variety of styles and content. Each image is labeled with its origin (AI or human), which serves as the ground truth for our classifier.

II. Data Exploration

A.Data Analysis

Feature	Details
Resolution	224x224 pixels
Model Performance	66%
Color Spaces	3-channel RGB for color features and 1-channel grayscale for symmetry and edge features
Horizontal Symmetry	AI – Higher symmetry Human – Lower symmetry
Vertical Symmetry	AI – Higher symmetry Human – Less symmetry
Edge Density	AI – Lower edge density Human – Higher edge density
Mean RGB Channels	AI – Higher mean value Human – Lower mean value

Table 1: Features and details of data

B. Preprocessing

Images were loaded from a compressed archive. Each was resized to 224x224 and stored in both RGB (for color features) and grayscale (for symmetry and edge features). This ensures uniformity across data samples [2].

C. Normalization

The normalization steps within the code occur during the execution of the symmetry score and edge density evaluation. We normalized the symmetry score, scaling it to a range of 0 and 1 (0 being completely symmetrical, and 1 being completely asymmetrical), by calculating the mean of the difference in the pixels [3]. As for the edge density normalization, it is ratio based, meaning that it finds the edge density in relation to the image's overall brightness, as brighter colors will have higher intensity, and therefore a greater pixel value [3]. This step is done to detect AI's tendency to oversaturate images.

A. Trends

The symmetry trends have shown that AI is more than likely to have a score closer to 0, and human-made art is going to be closer to 1 during the symmetry score normalization. As for edge density trends, AI art is going to yield a lower edge density to convey that there are smoother transitions happening, while human art is going to have a higher edge density to simulate brushstrokes, or textures during the human error process of creating art [3].

III. Baseline Solution

A. Prior research and baseline model:

In this field of AI generated images versus real images there are two primary types of prior research, traditional machine learning methods and deep learning techniques that rely on neural networks and very large datasets. Traditional methods tend to rely on features such as symmetry measures and texture descriptors. Deep learning methods tend to use architecture that learn to distinguish features on their own from raw images. One existing solution utilizes pixel level forensic features rather than conventional visual features [2]. Their model uses Photo Response Non-Uniformity and Error Level Analysis as input to convolutional neural networks to classify photorealistic AI images versus photographs taken with cameras. PRNU captures sensor imperfections unique to real cameras, while ELA highlights JPEG compression inconsistencies. Their approach achieves over 95% classification accuracy, demonstrating the power of low-level forensic signals for this task and the power neural networks hold in this instance [2]. While we will not be employing any neural networks for this task, such studies provide valuable insight on the existing state of the field and potential directions to take our research.

Due to the requirements and scope of our project, we chose to implement logistic regression as our baseline model. A logistic regression model requires fewer computational resources than other approaches and offers easily interpretable results. This makes it an ideal starting point for us because it allows us to analyze the effectiveness of our dataset and determine the most important features for future extensions.

B. Implementation:

We began our modeling process by implementing logistic regression as the baseline classifier due to its simplicity, interpretability, and low computational cost. We selected a small sample of 1,000 labeled images, which we split into a testing set of 200 images and a training set of 800 images. Each image underwent preprocessing and feature extraction to produce six features: horizontal symmetry, vertical symmetry, edge density, and mean values for the red, green, and blue channels. The logistic regression model was trained using these features. The model achieved a classification accuracy of 53% on the test set. Although this performance is only marginally better than random guessing, it validated the value of our handcrafted features and demonstrated that measurable statistical differences do exist between AI and human-generated images. This result established a foundation for further model enhancements and feature analysis.

IV. Model Improvement

A. SVM Classifier with Tuning

To improve upon the logistic regression baseline, we implemented a linear Support Vector Machine (SVM) using LinearSVC. This model is more robust in handling complex decision boundaries and offers effective performance for high-dimensional data [2]. We trained the model on a balanced dataset of 5,000 images and evaluated it using an 80/20 train-test split.

B. Hyperparameter Tuning

To identify the optimal regularization strength, we tested multiple values for the C parameter across several orders of magnitude. We performed 5-fold cross-validation on C values ranging from 0.0001 to 100 [3]. The best performing value was C = 100, which yielded a mean cross-validation accuracy of 64.7%. Testing different values for max_iter had a minimal effect on accuracy.

C. Feature Selection

Using a SequentialFeatureSelector, we determined that five features provided optimal performance. These were horizontal symmetry, vertical symmetry, edge density, mean red channel, and mean green channel.

D. Learning Curve Analysis

Learning curve analysis revealed that model performance stabilized as training data increased, suggesting that further improvements may require feature augmentation rather than additional data alone [3]. The accuracy begins to even out around 65%, which became the benchmark for our final model performance (Figure 1).

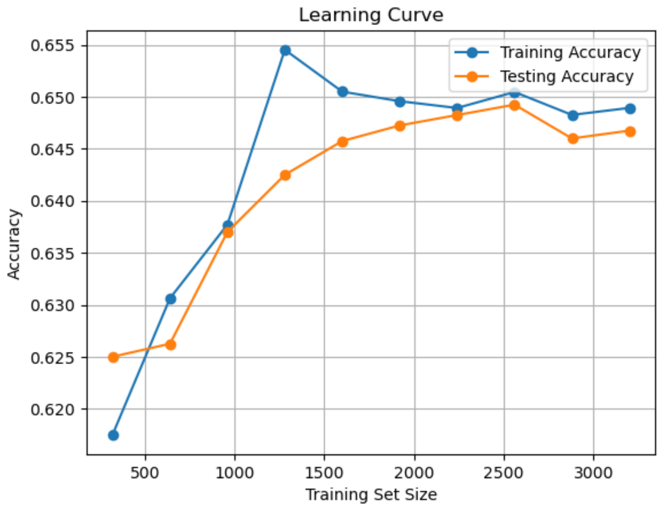

Fig. 1. Learning curve performance

E. *Final Performance*

Metric	Score
Training Accuracy	64.8%
Testing Accuracy	65.9%
Best C Value	100
Selected Features	5 of 6

Table 2: Test results for SVM algorithm

F. *Confusion Matrix (Test Set)*

A normalized confusion matrix showed balanced prediction performance between AI and human classes (Figures 2 and 3), with minor misclassification on certain ambiguous samples.

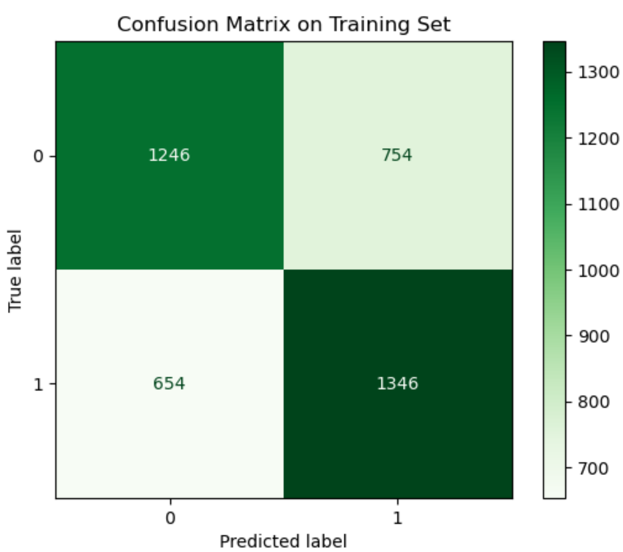

Fig. 2. Training confusion matrix.

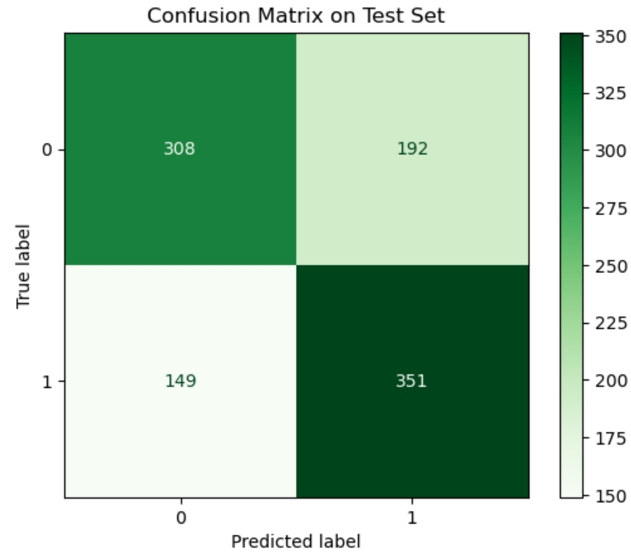

Fig. 3. Test results of confusion matrix

G. *Conclusion*

The SVM model outperformed the logistic regression baseline, improving testing accuracy to 65.9%. Symmetry and color features were most predictive. This work demonstrates that traditional models with handcrafted features can still provide valuable insights in AI-generated image detection.

V. PROPOSED EXTENSION

A. While our baseline solution provides a good foundation, there are several potential extensions or new approaches we can take from this point forward. One such extension is increasing the number of feature vectors. This can be in the form of introducing texture descriptors through local binary patterns or Gabor filters. [4] Other new features can also include more complex analysis of RGB values using color distribution histograms or similar measures. New features and more complex features can help introduce more nuance into our parameters and give our model more data to work with making it more effective.

B. Beyond new features, other potential extensions include using different classifications models. Tree based methods such as random forest can help capture nonlinear interactions among our features. Other approaches include combining predictions from both tree-based models and logistic regression to decrease variance. There is also much room for using various convolutional neural networks though for the purposes of this project and course those will be ignored.

C. Another potential improvement is applying techniques like recursive feature elimination to identify which features are most predictive to potentially get rid of the feature vectors that simply cause unnecessary noise. Diagnosing the features our model misidentifies may help us better understand where some of the oversight lies.

VI. CODE AND WORK DISTRIBUTION

The full Python implementation is available in a Jupyter notebook. The code includes detailed comments documenting preprocessing, feature extraction, training, tuning, and evaluation steps.

Kynnedy Armstrong and Erik Martinez worked together to write the code. Kynnedy implemented most of the model logic, including tuning, evaluation, and feature extraction. Nidya Vargas and Ahmad Tobasei focused on the initial half of the project, such as dataset preparation and baseline modeling, while Erik worked on the second half, including SVM improvements and final testing. All team members reviewed and revised the report collaboratively, checking for consistency, clarity, and correctness.

VII. REFERENCES

[1] C. McCarthy, *AI Art and the Artistic Revolution*, New York, NY, USA: FutureTech Press, 2023.

[2] F. Martin-Rodriguez, R. Garcia-Mojon, and M. Fernandez-Barciela, "Detection of AI-created images using pixel-wise feature extraction and convolutional neural networks," *Sensors*, vol. 23, no. 22, pp. 9037-9050, Nov. 2023.

[3] D. Park, H. Na, and D. Choi, "Performance comparison and visualization of AI-generated-image detection methods," *IEEE Access*, vol. 12, pp. 62609–62627, 2024, doi: 10.1109/ACCESS.2024.3394250.

[4] S. H. Khaleefah, S. A. Mostafa, A. Mustapha, and M. F. Nasrudin, “The ideal effect of Gabor filters and Uniform Local Binary Pattern combinations on deformed scanned paper images,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1960–1969, May 2022. DOI: 10.1016/j.jksuci.2019.07.012

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[6] K. Kannan, “AI and Human Art Classification,” Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/kausthubkannan/ai-and-human-art-classification