Artistic Insights: Classifying and Interpreting Artwork Styles Using Deep Learning

194.077 Applied Deep Learning (VU 2,0) 2024W

Viktoriia Ovsianik (12217985)

January 2025

1 Introduction & Problem Statement

This project aimed to develop a deep learning-based model to classify artworks into various styles and determine whether it was created by a human or generated by AI. The topic of the project is interesting because with the growing prevalence of AI-generated art, distinguishing between AI-generated and human-created artwork has become increasingly relevant. Additionally, accurately classifying artwork styles is challenging due to overlapping features between styles, such as Impressionism and Post-Impressionism. Addressing this problem has applications in art authentication, curation, and digital art markets.

2 Dataset

2.1 Dataset description

For this task, I used AI-ArtBench [1] which is a dataset that contains 180,000+ art images. 60,000 of them are human-drawn art (14th to 21st century) that was directly taken from the ArtBench-10 dataset and the rest is generated by AI using Latent Diffusion and Standard Diffusion models. The human-drawn art is in 256x256 resolution and images generated using Latent Diffusion and Standard Diffusion have 256x256 and 768x768 resolutions respectively. The dataset is already split into train and test parts and contains artworks from 10 different artistic styles:

- Art nouveau
- Baroque
- Expressionism
- Impressionism
- Ukiyo-e
- Post impressionism
- Realism
- Renaissance
- Romanticism
- Surrealism

2.2 Dataset preprocessing

Since the number of AI-generated artworks was higher in comparison to human ones, to balance the dataset I decided to select 5.000 images for each artistic style for each creation type (human, AI). Therefore, the final size of the training subset was (5.000 x 10 x2). Then I split the training subset into train and validation (90/10) setting the seed to provide reproducibility and resized all the images to 256x256. Labels were created with the logic ‘AI/human_{artistic_style_name}”. Finally, I had 90.000 images for training, 10.000 for validation and 30.000 for testing. The example of the images in the dataset can be found in Figure 1 1.

[1] Dataset source: https://www.kaggle.com/datasets/ravidussilva/real-ai-art

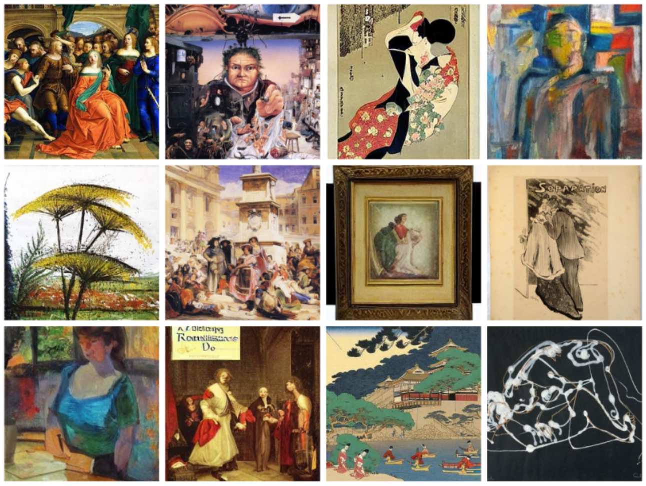

Figure 1: Dataset example

3 Approach & Methods

3.1 Models Selected

3.1.1 Simple CNN

As a baseline, I decided to start with a lightweight convolutional neural network designed for image classification. The model features: 2 convolutional layers with ReLU activation, max pooling after each convolutional layer, and a fully connected layer with 20 output units. Input images were resized to 32 × 32.

3.1.2 ResNet18

After reading literature I decided to consider more advanced architecture. Even though most articles focused on ResNet50, I decided to start with a simpler model that would take fewer resources to train as well as would be less prone to overfitting. ResNet118 is a residual neural network designed for image classification. It features: 7 × 7 convolutional layer with ReLU activation, followed by max pooling; 4 groups of residual blocks, each consisting of convolutional layers with skip connections to address the vanishing gradient problem; global average pooling layer to reduce spatial dimensions while preserving feature richness; a fully connected layer mapping to 20 output units, corresponding to the classification output classes. Input images were resized to 32 × 32. For all models described below, I focused on Transfer Learning by fine-tuning the pre-trained ResNet18 model from the Torchvision.models package.

1. **ResNet18: base** - for the base version, layers ['conv1', 'bn1', 'layer1', 'layer2'] were frozen during training to focus on deeper feature extraction.

2. **ResNet18: with aggressive augmentation** - for this version, I decided to add the following data augmentation techniques: Random Horizontal Flip, Random Rotation(30), Random Resized Crop (32), ColorJitter (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.2)

3. **ResNet18: with augmentation** - for this version, I decided to unfreeze all layers and apply less aggressive data augmentation: Random Horizontal Flip, Random Vertical Flip, Random Rotation(30)

4. **ResNet18: best performance** - for this version, I decided to unfreeze all layers, apply less aggressive data augmentation: Random Horizontal Flip, Random Vertical Flip, Random Rotation(30) and use images of higher resolution (resized to 224 × 224).

3.2 Training setup

The training setup was similar for all models:

- **Epochs:** 10 — A limited number of epochs was used to ensure the model could converge sufficiently during early testing phases without excessive training time.

- **Patience:** 3 — Early stopping with patience allowed the training to stop if no improvements were observed over three validation checks, reducing overfitting.

- **Validation Frequency:** 1 — Validating after every epoch provided immediate feedback on model performance and ensured that overfitting could be detected early.
- **Learning Rate:** 0.001 — A small learning rate allowed the model to make gradual improvements during training.
- **Optimizer:** SGD Optimizer — The Stochastic Gradient Descent optimizer was chosen for its simplicity and effectiveness in training models for image classification tasks.
- **Loss Function:** Cross Entropy Loss Function — This loss function is standard for multi-class classification problems.
- **Training Resources:** Google Colab GPU

3.3 Evaluation metrics

Since the main task was related to multi-class classification the following metrics were used:

- For tracking training process: accuracy, multi-class accuracy
- For evaluating results of the test set: accuracy, multi-class accuracy, precision, recall, f1-score, confusion matrix

The main metric was overall accuracy and the goal was to find an approach that would outperform the baseline.

3.4 Explainability

To increase models' results explainability I decided to add Grad-CAM (Gradient-weighted Class Activation Mapping). This is a technique used to visualize the areas of an image that contribute most to a model's prediction. By computing the gradients of the predicted class score concerning the feature maps of the final convolutional layer, Grad-CAM creates a heatmap that highlights regions of interest in the image.

4 Results

4.0.1 Simple CNN

The performance of the Simple CNN baseline model is summarized below:

Metric	Value
Test Loss	1.7142
Test Accuracy	45.12%
Overall Precision	44.83%
Overall Recall	45.12%
Overall F1-Score	44.03%

Despite the overall low performance, the model highlights difficulties not only in distinguishing between artistic styles but also in identifying whether the artwork is AI-generated or human-created.

4.0.2 ResNet18

ResNet18: base The performance of the ResNet18 base model is summarized below:

Metric	Value
Test Loss	1.0641
Test Accuracy	63.36%
Overall Precision	63.60%
Overall Recall	63.36%
Overall F1-Score	63.21%

While this model performs better than the Simple CNN baseline, it still shows uneven performance across classes. The model faces challenges with classes exhibiting subtle or overlapping features, likely due to similarities in artistic elements. Notably, the model's primary difficulty lies in identifying the correct artistic style, rather than distinguishing originality. AI-generated images are generally recognized as AI-generated but are often misclassified into a different style. This insight can potentially guide further performance improvements.

Additionally, it is important to note that the training process was stopped before reaching the 10th epoch because the validation loss started to increase, signalling overfitting.

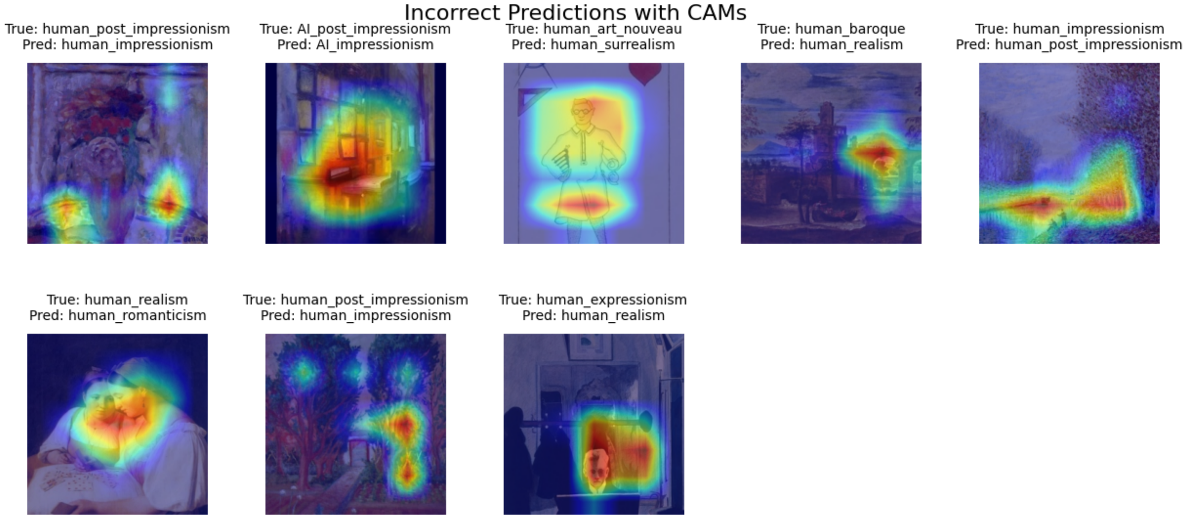
Incorrect Predictions with CAMs
True: human_post_impressionism
Pred: human_impressionism
True: AI_post_impressionism
Pred: AI_impressionism
True: human_art_nouveau
Pred: human_surrealism
True: human_baroque
Pred: human_realism
True: human_impressionism
Pred: human_post_impressionism
True: human_realism
Pred: human_romanticism
True: human_post_impressionism
Pred: human_impressionism
True: human_expressionism
Pred: human_realism

Figure 2: Incorrectly classified images with CAM heatmaps

ResNet18: with aggressive augmentation

For this version of the model training results worsened further. I realized that predicting artistic styles might require preserving the original appearance of the image, so I removed ColorJitter as it alters the image significantly.

ResNet18: with augmentation

The performance of the ResNet18 model with fine-tuning and less aggressive data augmentation is summarized below:

Metric	**Value**
Test Loss	0.9344
Test Accuracy	67.04%
Overall Precision	66.55%
Overall Recall	67.04%
Overall F1-Score	66.56%

Compared to the previous version of ResNet18, this model performs better by avoiding overfitting and making more generalizable predictions. The model continues to struggle with the same classes as the previous version, indicating that these classes have similar characteristics or subtle features that make classification challenging. However, the use of data augmentation has helped the model generalize better, leading to improved overall performance.

ResNet18: best performance

The performance of the ResNet18 model with improved resolution is summarized below:

Metric	**Value**
Test Loss	0.4600
Test Accuracy	82.90%
Overall Precision	82.87%
Overall Recall	82.90%
Overall F1-Score	82.90%

Based on the evaluation results of the previous models, it became clear that the model struggled to identify certain artistic movements such as Impressionism, Post-Impressionism, and Expressionism, which often share overlapping visual traits. These movements are not always easy to distinguish, even for humans. In an attempt to improve the quality of the results, I decided to increase the resolution of the images to 224x224 pixels, as this would allow more details to be captured, which could be crucial in distinguishing between the styles. This decision led to a significant improvement in performance: the improved resolution allowed the model to better capture the subtle differences between the artistic styles, resulting in better performance overall. However, the model still struggles with certain classes that exhibit overlapping features, particularly in the cases of human_impressionism and AI_post_impressionism.

Outcomes of classification on the test set using the best-performing model. Incorrectly (Figure 2) and Incorrectly (Figure 3) classified images are plotted together with heatmaps that identify regions important for classification.

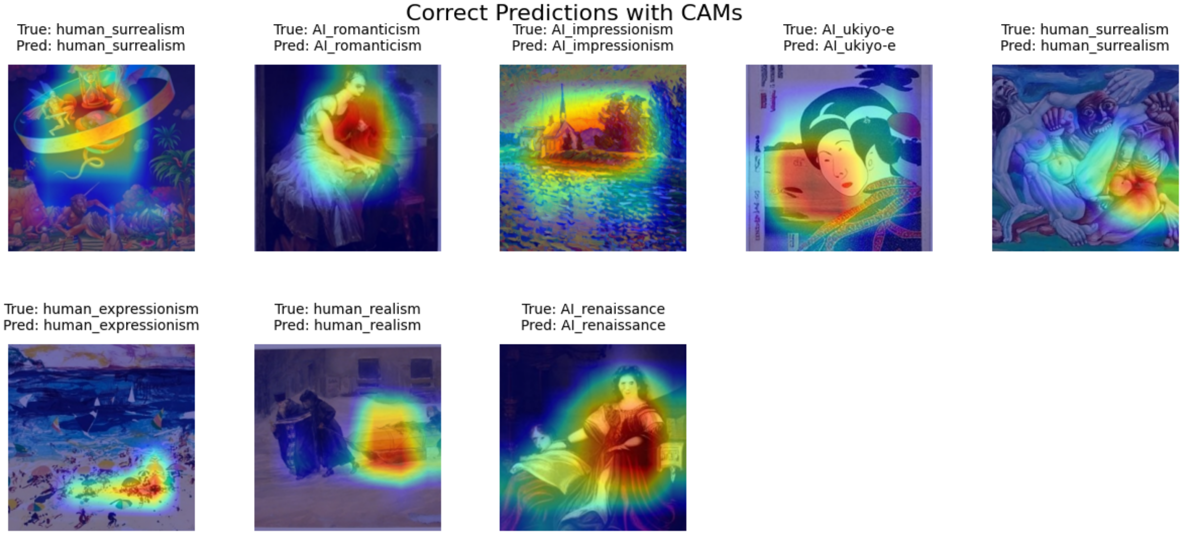

Figure 3: Correctly classified images with CAM heatmaps

5 Conclusion & Future work

In this project, I aimed to build a model capable of classifying artworks based on their origin and artistic style. To achieve this, I worked with the AI-ArtBench dataset and evaluated SimpleCNN (baseline) as well as the more advanced ResNet18 model and its variations. Throughout my experiments, I was able to improve accuracy from 45% to almost 83%. As a further development, I believe that including more variable data (especially for AI-generated images) would help the model to become better at generalization. Also experimenting with more advanced models like ResNet50 might be worth it.

6 Take-aways & Insights

- **Trying simpler models might be worth it**: Many studies focused on ResNet50 however, simpler models, such as ResNet18 also provided competitive results for my task. Therefore, simpler models should not be overlooked when trying to balance computational efficiency and accuracy.

- **Impact of Image Resolution on Training:**: One simple decision as increasing the image resolution did not significantly extend the training time, yet it resulted in a notable performance improvement. This demonstrates the importance of providing higher-quality input data for the model.

- **Data Augmentation and Task-Specific Effects:** Not all data augmentation techniques are universally effective across different tasks. In this project, I observed that aggressive augmentation methods, such as brightness, saturation and hue changes modified fundamental features of the image leading to incorrect artistic style detection.

- **Availability of resources for deep learning tasks**: It is really important to have proper computational resources when working on deep learning tasks.

7 Literature

1. Jordan J. Bird, Ahmad Lotfi. *CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images*, 2023. (https://arxiv.org/abs/2303.14126).

2. Lingquan Zeng. *Improved Painting Image Style Classification of ResNet based on Attention Mechanism*, IEEE, 2021. (https://ieeexplore.ieee.org/document/10512005).

3. Wentao Zhao, Dalin Zhou, Xinguo Qiu, and Wei Jiang. *Compare the performance of the models in art classification*, 2021. (https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0248414).

4. Ravidu Suien Rammuni Silva, Ahmad Lotfi, Isibor Kennedy Ihianle, Golnaz Shahtahmassebi, Jordan J. Bird. *ArtBrain: An Explainable end-to-end Toolkit for Classification and Attribution of AI-Generated Art and Style.* (https://arxiv.org/abs/2412.01512).

8 Time Distribution

Task	**Expected Effort**	**Real Effort**
Dataset collection	2 hours	2 hours
Dataset preprocessing	4 hours	12 hours
Designing & building network	32 hours	24 hours
Fine-Tuning	32 hours	12 hours
Additional Fix	5 hours	12 hours
Demo App	32 hours	12 hours
Report & Presentation	10 hours	4 hours