ECGR 4105 Final Project - AI Versus Human Art Classification

Cole Fredrick, Joelle Bailey, Noah Telfer

Abstract**—— GitHub Repository: https://github.com/Cole-Fredrick/ITML-FINAL_PROJECT**

I. Introduction

At the Colorado state fair annual art competition this year (2022), one entrant by the name of Jason M. Allen, created and submitted an AI art piece utilizing an AI program by the name of Mid-Journey. His art piece, ”Théâtre D'Opéra Spatial”, won first place in the digital art contest and single-handedly marked a footprint for the future of artificial intelligence art. This did not come without repercussions as the backlash of winning this contest provides further clarity to a viewpoint from artists that AI art should not be allowed to compete or alter platforms that have primarily consisted of human made art.

The impact of AI art on human artists is a topic of debate and discussion that has grown steadily over the course of recent years. Some believe that the rise of AI art has the potential to challenge and even replace human artists, as AI algorithms can create new works of art quickly and without the need for human intervention. This could potentially disrupt the traditional art market, as well as the careers of human artists. On the other hand, people also argue that AI art can enhance and broaden the work of human art.

In the context of AI art, image classification can be used to identify whether a given image was created by a human or an AI algorithm. This overall, is a challenging task as AI algorithms have become sophisticated enough to generate highly realistic images that can be difficult to distinguish from that made by humans. With the goal of overcoming this challenge, the team tested two neural models with the aim of reaching the highest possible validation and training accuracy for classification. The two networks are ResNet-18 and ResNet-34.

II. Data Set

A. Human Art Data

To gather data for the non-AI class, a portion of a data-set was utilized: GANgogh developed by Github User rkjones4. This data-set acts as a training set to help GAN networks learn how to create art. It utilizes a plethora of images pulled from wikiart.org categorized by different styles and scene types with a variety of different image resolutions. The team gathered 944, landscape category images from the set with the lowest image resolution being 512x264.

B. AI Art Data

To gather the images for the AI class of the data set, the first goal was to choose a program that allowed us to mass produce images that could generate art as similar to the human data gathered as possible. Originally, there was some utilization of Mid-Journey, but given the amount of data needed and the limitations of images you could generate on a free plan, it was ultimately decided against. The team decided on using Stable Diffusion. Stable diffusion is an AI art program that uses a mathematical concept called ”diffusion-limited aggregation” to generate complex, organic-looking patterns. The program is called ”stable” because the resulting patterns are often aesthetically pleasing and well-balanced. This software allows a user to enter a prompt and select from a wide variety of options to generate an image of their liking. What was primarily favored is that generating images could be done in batch operations allowing the team to gather the images we needed quickly. A total of 945 images were generated with the following properties listed below:

Stable Diffusion Image Properties:

- Prompt: Landscape, People, Epic Scene, (Location Variable)
- Locations: Ruins, Forest, Village, Farmhouse, Manor, Swamp, Lake, Bridge, Meadow, Mountain, Ocean, Castle
- Sampling Step (Inference Steps): 55
- Image Resolution: 512 x 512
- Images per category batch: ≈ 78

After gathering the data needed for both the AI and non-AI classes, the data-set was established on Roboflow to handle back-end processing and to give the team a way to easily annotate images and import the data to be used in training. The properties of the data-set are listed below:

AI Versus Non AI Classification Data-Set Properties:

- Total Image Count: 1,889
- Classes: NON-AI and AI
- Train Split: 70 - (1318 Images) - (659 NON, 659 AI)
- Validation Split: 20 - (374 Images) - (187 NON, 187 AI)
- Test Split: 10 - (197 Images) - (98 NON, 99 AI)
- Prepossessing: Image Stretch - 512 x 512
- Augmentations: NONE

III. Training Models

A. Data Initialization

All notebook and script work was done in Google Colab using the free plan. Before training can begin, the data must be loaded into the notebook. The data-set is already fully class annotated due to the onset capabilities of the website. All that's left was to prepossess and normalize the training and validation sets. The team calculated the standard deviation and mean for the image set and normalized the image tensors with those results. The loaded data set was not augmented beyond the previous statements for both the ResNet18 and ResNet34 models.

TABLE I: Mean and Standard Deviation for Image Reprocessing

	Evaluated Normalization Weights
Mean	0.4758, 0.4473, 0.3603
STD	0.2669, 0.2644, 0.2747

B. ResNet18 Overview

Resnet is a convolutional Neural network that is trained on the Image-Net dataset, a large dataset of images that are commonly used to train and evaluate image classification models. ResNet maintains the usage of residual connections, which allows the network to learn more effectively by enabling the flow of information from earlier layers to enter later layers in the network. This allows the network to learn and improve on things more easily in regards to image classification tasks. In the context of ResNet18 whose architecture is 18 layers deep, the model parameters roughly equals 11,174,000. ResNet18 was used as the team wanted to not only utilize it as an introductory model for image classification and convolutional networks but it also boasts a lite training framework due to a low layer count and model complexity. To train the model ResNet18 was initialized with random parameters (untrained) and with predefined parameters (pre-trained).

1) ResNet18 Untrained: The Untrained version of ResNet18 was imported with randomly initialized parameters. Essentially this means that the model with have to start learning from scratch. Table 2 shows the values in which the model has been initialized to. The learning rate is the rate at which the the weights of the model are adjusted. In this case the learning rate has been set to 0.001. The stochastic gradient descent optimizer was used because of its computation efficiency and wide use as a baseline throughout the industry.

2) ResNet18 Pre-Trained: The Pre-Trained version of ResNet18 parameters updated to reflect the ResNet18 model trained off the data set ImageNet. ImageNet is a large collection of images that have been hand labeled. By using ResNet18 parameters that have already been pre-trained off ImageNet it gives the model a base knowledge and a starting point to begin learning. The parameters used to train this version of ResNet18 are listed in Table 2.

TABLE II: Untrained ResNet18 Training Setup

ResNet18 (Untrained) Training Setup	
Learning Rate	0.001
Transformations:	Normalized Data
Batch	64
Optimizer	Stochastic Gradient Descent
Loss Function	Cross Entropy
Number of Epochs	10
Momentum	0.9
Runtime Device Used	Google Colab GPU

TABLE III: Pre-Trained ResNet18 Training Setup

ResNet18 (Pretrained) Training Setup	
Learning Rate	0.001
Transformations	Normalized Data
Batch	64
Optimizer	Stochastic Gradient Descent
Loss Function	Cross Entropy
Number of Epochs	10
Momentum	0.9
Runtime Device Used	Google Colab GPU

C. ResNet34

Extending from this class of convolutional neural networks, ResNet34 is a deeper version of the same model concept. With 34 layers and 21,797,672 parameters, the main difference between the two models is the added architectural complexity and potential for more abstraction. However, some of the inherent drawbacks of this model are traced back to this complexity. With a greater number of layers the computational power needed to train the model increases. When increasing the number of parameters, the back propagation similarly has to do more computation. The objective of the inclusion of this model as a comparison to the ResNet18 model was to add a reference point for the impact of adding complexity to the solution and the cost overall.

The ResNet34 models were implemented with similar statistics to the ResNet18 models. The two models trained from the architecture differed only in whether the initial parameter weights were loaded from a ResNet34 model trained on ImageNet or untrained with random initial weights. ImageNet is a collection of images with objects in the world that a general model for image recognition could use to classify a variety of objects. The theory for including these weights is that a model trained to recognize images may be able to avoid local loss minima and may better find a generalized solution. The training setup statistics are shown in the tables below.

TABLE IV: Untrained ResNet34 Training Setup

ResNet34 (Untrained) Training Setup	
Learning Rate	0.001
Transformations	Normalized Data
Batch	64
Optimizer	Stochastic Gradient Descent
Loss Function	Cross Entropy
Number of Epochs	10
Momentum	0.9
Runtime Device Used	Google Colab GPU

TABLE V: Trained ResNet34 Training Setup

ResNet34 (Pretrained) Training Setup	
Learning Rate	0.001
Transformations	Normalized Data
Batch	64
Optimizer	Stochastic Gradient Descent
Loss Function	Cross Entropy
Number of Epochs	10
Momentum	0.9
Runtime Device Used	Google Colab Premium GPU

The 0.001 learning rate was selected for a slower and more stable training and to increase the time to overfitting while maintaining a comparable epoch number. Other transformations explored included resizing and center cropping the images. Ultimately these where not used as these functions were redundant from the transformations Roboflow applied to the images. The batch size was selected in relation to how many epochs expected and the size of the dataset. This number allowed the full dataset to be seen and fit on to the computational devices used while still being small enough to generalize the problem. Overall, these parameters were selected to allow for insight into the specific model changes made, namely the ResNet18 versus ResNet34 difference.

IV. Results

A. ResNet18 Results

The results shown below in Table 6 and 7 represent the ResNet18 Untrained and Pre-trained performance. The accuracy achieved by the ResNet18 Untrained model was 74.86%. Using the parameters defined in Table 2 the models loss and accuracy seemed to fluctuate heavily during the training. This implies that the model is traversing multiple local minima and cannot find or has yet to find the global minima. Hypothetically this issue is being caused by a lack of data ore the lack of variety in data in the generated data set. The lack of variety in the data set could also be causing overfitting which was seen in the form of a downward trend in training loss while the inverse was true for the validation loss. These issues were solved however, when the model pre-trained. This is likely to do with the fact that the model was trained using ImageNet so it had some knowledge of what an image was. This seemed to give the model the same results that it would have had if the dataset had been padded. Overall the ResNet18 Pr-Trianed had the best accuracy out of each instance include the 34 however this may have changed if the models had been trained over more epochs.

TABLE VI: Untrained ResNet18 Training Results

ResNet18 (Untrained) Results	
Training Time	351.54s
Training Accuracy	92.33%
Validation Accuracy	74.86%

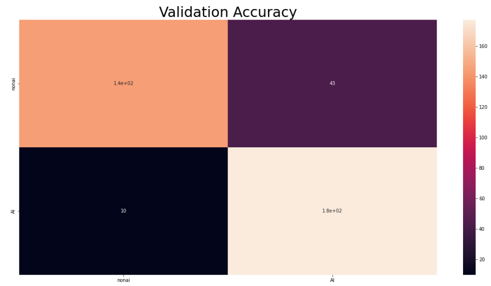

Fig. 1: ResNet18 (Untrained) Confusion Matrix

TABLE VII: Untrained ResNet18 Training Results

ResNet18 (Pre-Trained) Results	
Training Time	292.23s
Training Accuracy	99.84%
Validation Accuracy	99.73%

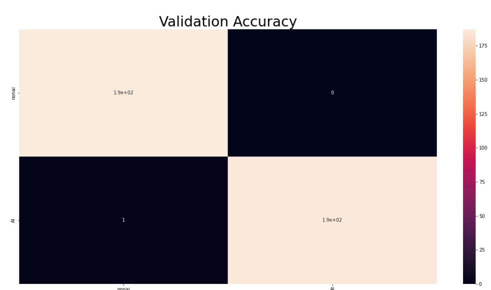

Fig. 2: ResNet18 (Pre-Trained) Confusion Matrix

B. ResNet34 Results

The ResNet34 results were obtained in a similar fashion to the ResNet18 results. The timing of each run is expected for the computational settings used and the increase in accuracy from the untrained to the pre-trained version of the model remains the same. The results of the untrained ResNet34 results align with the expectation that the trained weights may allow for the model to reach a more generalized solution as the initial settings may draw the model away from local minima. Because the model continued to increase and did not stop at a high error zone, the resulting weights found can be said to have a more general solution and most likely has less symptoms of overfitting.

ResNet34 (Untrained) Results	
Training Time	373.0389s
Training Accuracy	92.87%
Validation Accuracy	91.18%

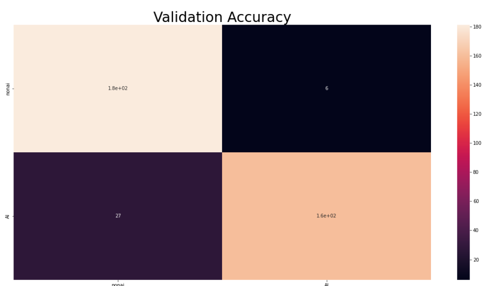

Fig. 3: ResNet34 (Untrained) Confusion Matrix

The untrained ResNet34 model skews slightly to non-AI with the classification of more AI generated images as non-AI than the other way around. Because the model is quite accurate the comparative skew would have to be further examined with more validation data.

ResNet34 (Pretrained) Results	
Training Time	102.2524s
Training Accuracy	100.00%
Validation Accuracy	99.20%

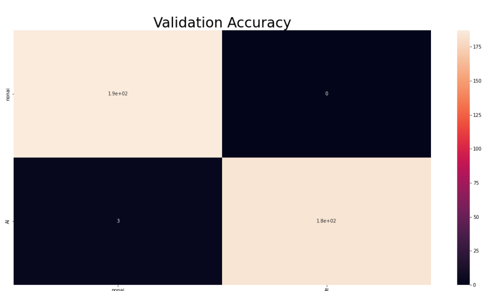

Fig. 4: ResNet34 (Pretrained) Confusion Matrix

The timing for the pretrained ResNet34 model is due to the premium GPU used for the training setup of this model.

Fig. 5: Test Case with Similar Objects

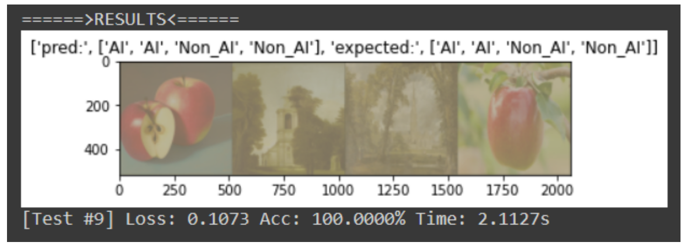

Fig. 6: Test Case with Similar Objects

Using a selection of similar objects a blind validation was done to see if the model trained (in this case ResNet34 with pretrained weights) would be able to generalize. This is by no means a conclusive test and will need to be a further area of study. However, the initial tests runs as shown in the selection about were promising for the achievability of this result.

V. Conclusion

The classification of AI generated versus non-AI generated art has become an increasingly needed resource as the structures in place to monetize art are slow to adapt to the new technology introduced. To protect small-content creators who rely on AI art flags and non-AI forums and create a fair grounds for these art mediums to be judged on, this project aimed to take a step in the way of content verification turning AI against AI.

To accomplish this end, four residual networks were used on a dataset comprised of both artificially generated and human-made art images in equal number. These networks allowed for an understanding of the potential benefits of this form of classification with correct classification. The pretrained models resulted in the best accuracy with the conclusion that general image weights at initialization improve the overall performance of the model.

Further research on this topic could explore the building of a more generalized model with the access to more varied dataset generation prompts and general AI art databases to find specific image and pixel to pixel indicators of AI-generation.

VI. References

G. L. Team, "Introduction to resnet or residual network," *Great Learning Blog: Free Resources what Matters to shape your Career!*, 22-Mar-2022. [Online]. Available: https://www.mygreatlearning.com/blog/resnet/. [Accessed: 16-Dec-2022].

"RoboFlow", *Roboflow*. [Online]. Available: https://roboflow.com/. [Accessed: 16-Dec-2022].

None, "Gangogh Training Data Set," *Academic Torrents*. [Online]. Available: https://academictorrents.com/details/1d154cde2fab9ec8039becd03d9bb877614d351b. [Accessed: 16-Dec-2022].