# The "Genes" of Paintings: Style Transfer and Style Classification of Pastiches

Evan Yan

Franklin Regional Senior High
Murrysville PA
21yane@franklinregional.k12.pa.us

**Abstract**

Painting recognition and pastiche creation are tasks that are traditionally done by experts, such as art historians or artists. However, these tasks have been increasingly delegated to computers. A 3-step method was proposed and utilized to compare the relative importance of style and content in distinguishing important art periods. The 3-steps respectively are: style classification, style transfer, and style classification on pastiches. A dataset of 80 thousand digitized paintings from WikiArt, the most common dataset for art-related computer vision research, was used. A total of 27 different art styles were combined and trimmed to the amount of 20 styles. The 3-step method was implemented with the Python language by integrating two machine learning libraries, FastAI and TensorFlow. First, a Convolutional Neural Network (CNN) was trained using the WikiArt dataset, with the goal of identifying the art period of any given painting as accurately as possible. After 7 epochs of training, a painting recognition accuracy of 57% was achieved with the style classifier, comparable to results in the literature of 60%. Then, the style transfer, which also employs neural networks and the WikiArt dataset, was utilized to create pastiches. 3800 pastiches were produced from the 20 art periods. Lastly, the style classifier (from step 1) was used upon all the pastiches to determine the similarity percentage, with the goal of finding the amount of resemblance these children paintings would bear to their parents. The results showed that the intrinsic value or prominence of each style had little effect upon the amount of resemblance. The relationship between the percentage and dataset size is found to be the most significant pattern.

## I. Introduction

Painting has existed for thousands of years, and its techniques have been constantly evolving. When comparing today's art to the art of the past, there are obvious shifts in the paintings. These shifts are used by art historians to distinguish art periods: some of these art periods are very similar to one another, while others may be very different.

The intersection of computer science and humanities has attracted much attention in the machine learning community. Machine learning is the study of computers being able to automatically learn through experience. Researchers view creative work and machine learning as a possible method to answer philosophical questions or gain insight on potential applications [1]. As stated by Goodfellow et al., "ironically abstract and formal tasks that are among the most difficult mental undertakings for a human being are among the easiest for a computer" [2]. This suggests that abstract activities such as decoding works of art may be accomplished effectively with machine learning.

Style and content are typically considered as distinctive aspects of any work of art—not only for paintings, but also for other mediums such as music or literature or film. *Style* refers to a set of qualities unique to a particular art movement such as color, line, light, space, and composition, whereas *content* refers to the objects in a painting. Style exists in art with our own eyes, even if it is an abstract concept. As shown in Figure 1, both images show the same content of dogs, yet the style of each painting is clearly different.

The overall goal of this study is to gain a better understanding of the style and content "genes" of paintings from different art periods, using machine learning. As style and content are rather abstract constructs, different paintings' "genes" will be mixed together to synthesize pastiches which are more concrete and easier for us to decipher the meaning of these "genes" and how they contribute to artistic style by looking for any special patterns. [1] [2]

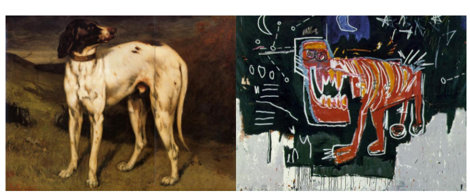

Fig. 1: Comparison of style

[1]The pronoun we is used throughout this paper. Please note that this is only for stylistic reasons, and that this research was purely independent.

[2]There are two usages of the term "style" throughout this paper. In the first use case, "style" and "art period" are used interchangeably. They refer to a class or movement of art (e.g., "Expressionism"). In the second use case, "style" refers to more abstract artistic characteristics.

## II. Related Works

There is a history of collaboration between computer vision and art. This is done synthetically (such as style transfer) and analytically (such as painting recognition).

### *A. Style Transfer*

Style transfer is the act of applying the style from an artistic image onto another image. Traditionally, recreating paintings in a particular style has required skill and substantial time [3]. However, as more computer scientists have become more interested in the field of art, new developments in techniques have enabled effective synthetic paintings (also known as pastiches).

Early work in style transfer came from texture synthesis. Image-based artistic rendering techniques such as stroke-based rendering, region-based rendering, example-based rendering, have all achieved limited success [3]. Recently, however, Neural Style Transfer (NST) [4] has shown breakthrough results using Convolutional Neural Networks (CNNs). The algorithm iteratively optimizes an image by simultaneously minimizing the style and content reconstruction loss. Gatys et al.'s algorithm is computationally expensive to run due to the optimization loop [5]. This seminal work has inspired other papers to improve on NST; for example, Johnson et al. aim to reduce the computational burden [6].

### *B. Painting Recognition*

Painting recognition can be defined as the task of successfully predicting the attributes of a painting (e.g., style or artist) given the image. Painting recognition is a task that is traditionally done by experts, such as art historians, museum curators, and collectors. However, there has been an increase in painting recognition by computers. Research in computer vision has proven the capabilities of machines to classify artworks to their style, with acceptable accuracy [7].

Automatically identifying the style is a challenging problem. Compared to other similar tasks such as facial recognition, which uses recognizable landmarks such as the eyes, style classification cannot depend on definite and reliable features [1], which leads to lower identifying accuracy.

The recognition accuracy is optimized by most researchers through either classical or deep learning methods. Classical methods use pre-computed features such as color histograms, or low-level features such as color, shading, or texture, in order to recognize art [3]. Deep learning methods focus on the use of deep neural networks. As Cetinic et al. states, the recent breakthroughs in computer vision achieved by CNNs, demonstrate the dominance of deep learning compared to classical methods for numerous image classification tasks [8]. Chen and Deng also prove that deep learning is superior to classical machine learning methods for art recognition, specifically for artist recognition [9].

Studies involving art recognition typically aim to get the best accuracy at predicting an attribute, such as the artist [1], [8], [10]. Aside from optimizing accuracy, some studies also approach the task from novel perspectives. For example, one paper utilizes machine learning to discover similar patterns in large collections of artworks [11]. Another example by Garcia et al. uses paintings together with art analysis commentary—a multi-modal approach—in order to explore the connection between the aesthetics of art and underlying meaning [12]. The mix of vision and language yielded good classification accuracy, comparable to human performance.

Our study is inspired by Hicsonmez et al.'s study [13]. Their study uses style transfer of children's book authors' styles onto normal pictures and then tries to classify the stylized pictures by author. We do not use children books, but we do follow a similar pattern of style transfer followed by style classification. This process has not been copied in the computer vision field, as far as we know. Thus, our study is unprecedented in that there are no studies that use a style transfer to style recognition process, as well as utilizing the WikiArt dataset.

## III. Methodology

A 3-step method was proposed and utilized in this paper to analyze the style and content of each important art period. As outlined in Figure 2, these 3-steps are style classification, style transfer, and style classification on pastiches, respectively. The 3-step method was implemented with the Python language by integrating two machine learning libraries, FastAI and TensorFlow.

First, a Convolutional Neural Network (CNN) was trained using the WikiArt dataset, with the goal of identifying the art period of any given painting as accurately as possible. Then, the style transfer, which also involves neural networks and the WikiArt dataset, was utilized to create pastiches. Lastly, the style classifier was used upon all the pastiches to determine the similarity percentage, with the goal of finding the amount of resemblance these children paintings would bear to their parents.

There are multiple layers in a neural network. As shown in Figure 3, Each layer consists of nodes, which give a signal to the next layer. This process is highly similar to and is inspired by the brain's neurons, which is also a network of signals. The quality of the output is dependent on both the quality of input and the quality of the neural network. To improve the quality of the output, the neural network model must be trained through multiple iterations. This entire process can be thought of as a "guess-and-check" process, in which the neural network steadily improves by learning from its mistakes.

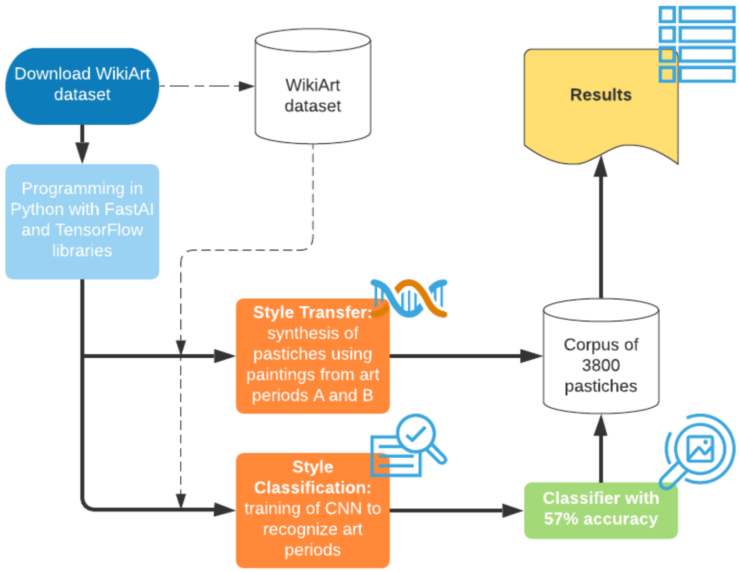


Fig. 2: Flowchart of experimental process

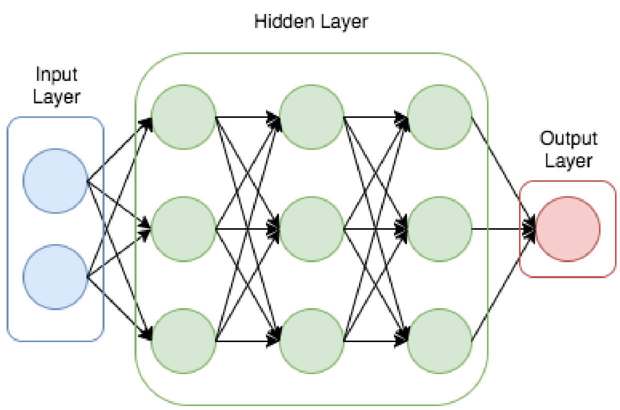


Fig. 3: Visualization of a neural network [14]

Currently, the WikiArt dataset is the most common dataset for art-related computer vision research [8]. We use a dataset of 80,000 digitized paintings from WikiArt, which originally includes a total of 27 different art styles, for style transfer. We combine styles, trimming the amount into 20 styles, similar to Elgammal et al. [7]. Specifically, we combined Post Impressionism and Pointillism; Realism, Contemporary Realism and New Realism; Cubism, Analytical Cubism and Synthetic Cubism; and Abstract Expressionism and Action Painting.

Our final 20 styles were: Abstract Expressionism, Art Nouveau Modern, Baroque, Color Field Painting, Cubism, Early Renaissance, Expressionism, Fauvism, High Renaissance, Impressionism, Mannerism Late Renaissance, Minimalism, Naive Art Primitivism, Northern Renaissance, Pop Art, Post Impressionism, Realism, Rococo, Romanticism, and Ukiyo-e.

Figure 4 depicts the number of paintings for each of the 20 art periods. Famous examples from each art period are given in Figure 5.

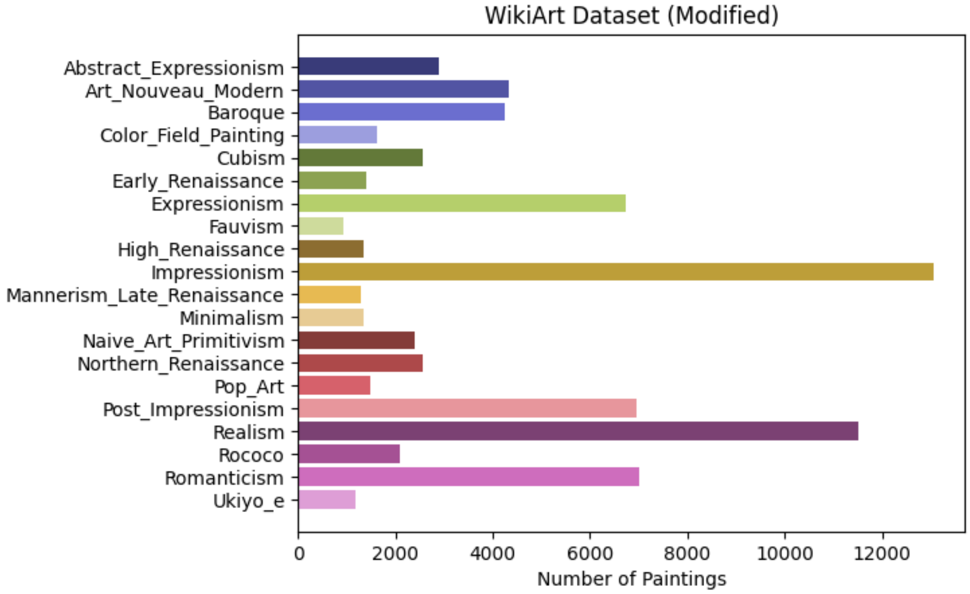


Fig. 4: Style distribution

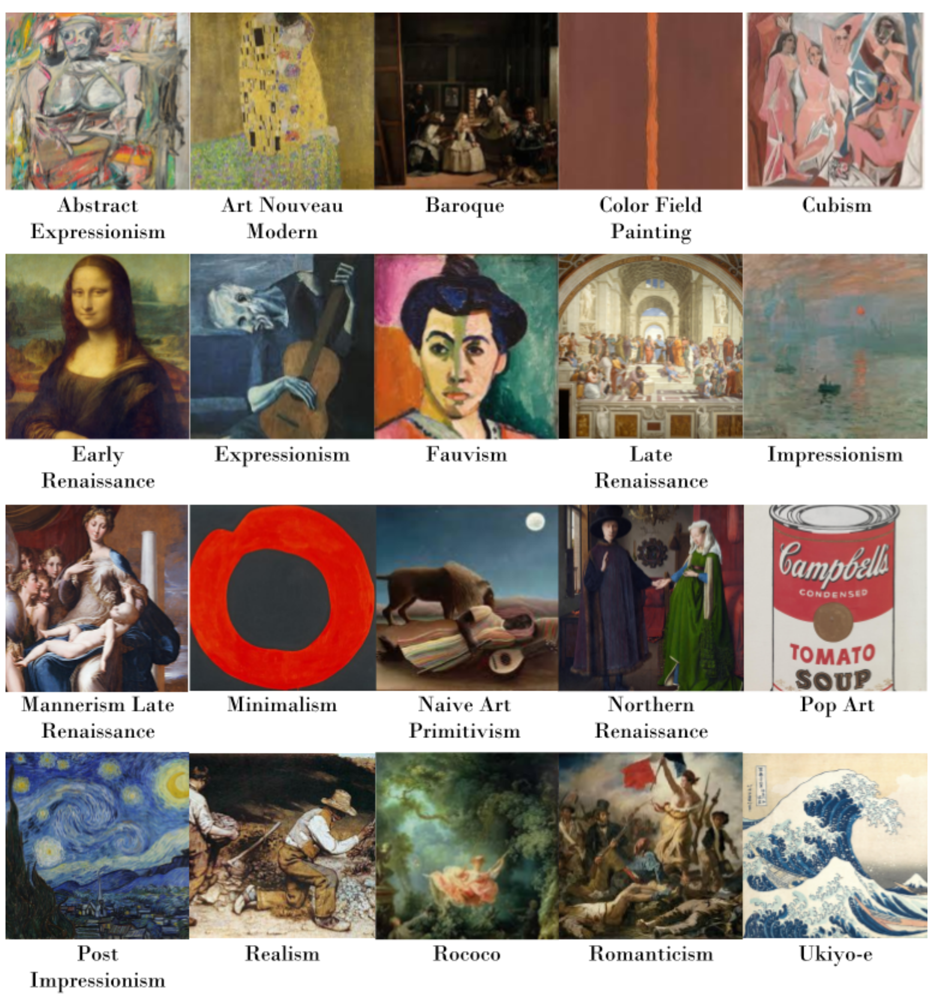


Fig. 5: Famous example paintings for each style

## IV. Results and Discussion

### A. Style Classification

Throughout the development process, we used the Python programming language as well as the FastAI and TensorFlow libraries for machine learning. The dataset from WikiArt was used to train, cross-validate and test a neural network devoted to the purpose of style classification. Specifically, we divide this collection into a 60:20:20 for training, cross-validation, and testing, respectively, as it is the standard practice.

We use the residual network (ResNet) architecture for our classification neural network, as it performs better in comparison to other networks such as AlexNet or VGGNet [15]. The specific metric we optimize for is accuracy, which is defined as the ratio of the number of correct predictions and the total number of predictions.


Fig. 6: Increased accuracy over epochs

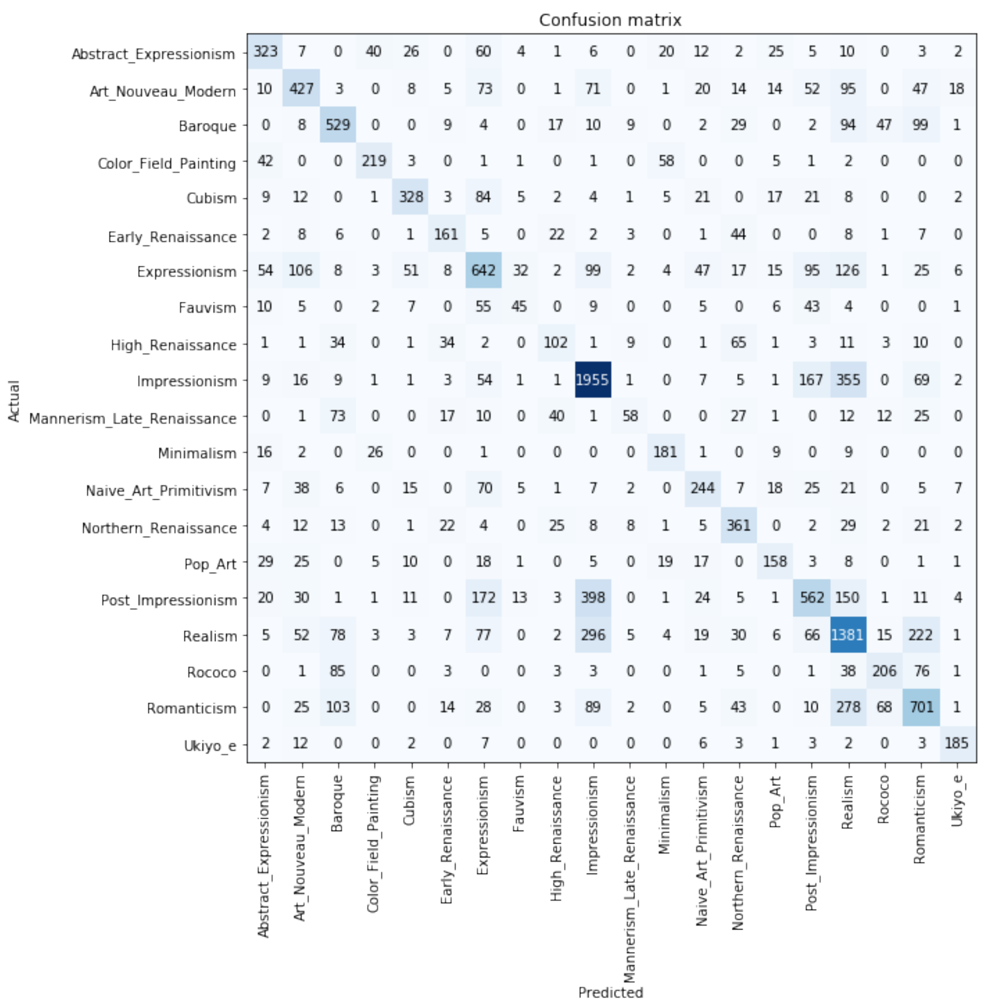


Fig. 7: Confusion matrix by style classification

Google Colab, a free computational service primarily targeted for researchers, was used to run the style classification program. Each training took an average of 4 hours and 13 minutes. Times ranged from 2 and half hours to 6 and half hours because of the variable availability of faster GPUs. Sometimes, GPUs were just unavailable, as resources are not guaranteed. Due to this reason, only 7 epochs (training cycles) were conducted. As shown in Figure 6, the accuracy of the model on cross-validation dataset increased from 48% to 57% after 7 epochs.

The confusion matrix shown in Figure 7 demonstrates the number of mistakes the classifier made during the last epoch (7th iteration). The classifier made a total of 15382 predictions, and 8768 correct predictions. This yields the prediction accuracy of 57%, as shown in Figure 6. The 3 most commonly mistaken pairs were Realism and Romanticism, Realism and Impressionism, Post Impressionism and Impressionism. It should be noted that these classes also contained higher amounts of paintings, thus likely resulting in more common mistakes.

ResNet is originally pre-trained for object recognition on the ImageNet dataset, a crowdsourced large database containing images of objects. We adapt this neural network by training it to recognize art styles. This method is called transfer learning and allows us to fine-tune some layers of the neural network rather than having to completely train the network, which would be computationally expensive.

Lecoutre et al.'s study empirically found that the best method for training the WikiArt dataset classifier is done by retraining about 20 layers of the neural network, which suggests that transfer learning and then fine-tuning is superior to a full training of the network [1]. While fine-tuning requires less data than complete training, a relatively large amount of images is still necessary [8], [16].

### *B. Style Transfer*

Style transfer typically involves a painting as the style image and a digital photograph as the content image. A unique method was utilized in this study by using paintings for both parents. 10 paintings from one art style are used as the content image, and 10 paintings from another art style are used as the style image. In total, we create 10 modified paintings for each of the 380 unique pairs of styles. This number of 380 pairs is calculated by multiplying 20 styles and 20 styles and excluding 20 same style pairs (e.g., Abstract Expressionism — Abstract Expressionism). This means a total of 3800 pastiches. Figure 8 shows a sample of 380 pastiches randomly selected from the total of 3800 pastiches generated in this study.

Figure 9 provides example parents (content image and style image) and an example pastiche. The example content image is *Pencerrig* and the example style image is *Madonna*, both taken from the WikiArt dataset. Notice that the pastiche is a blend of the two paintings: the content image's landscape is still present whereas the style image's colors and brushstrokes take over.

When creating these pastiches, we use Ghiasi et al's variant of style transfer [17] rather than other alternatives because of its efficiency. We wanted to emphasize both speed and quality. Ghiasi et al's version is trained on a collection of 80,000 paintings and is able to classify and generalize to unobserved paintings. They train the model specifically on the *Kaggle Painting By Numbers* dataset, which is notable because the majority of the dataset is related to the WikiArt dataset. This means that their model has already observed most of the WikiArt paintings, suggesting that it is specialized for our task.

Ghiasi et al. 's algorithm relies on data; therefore, the stylization quality is highly dependent on the varieties of training styles [3]. Because WikiArt and *Kaggle Painting By Numbers* training styles are essentially the same, the quality should be high. While Gatys et al.'s model is considered the gold standard, it is slow. For a 256 x 256 pixels pastiche, Gatys et al.'s model takes 14.32 seconds to construct, whereas Ghiasi et al.'s model takes only 0.014 seconds. The tradeoff is that Ghiasi et al.'s model is not effective at producing complex style patterns [3].

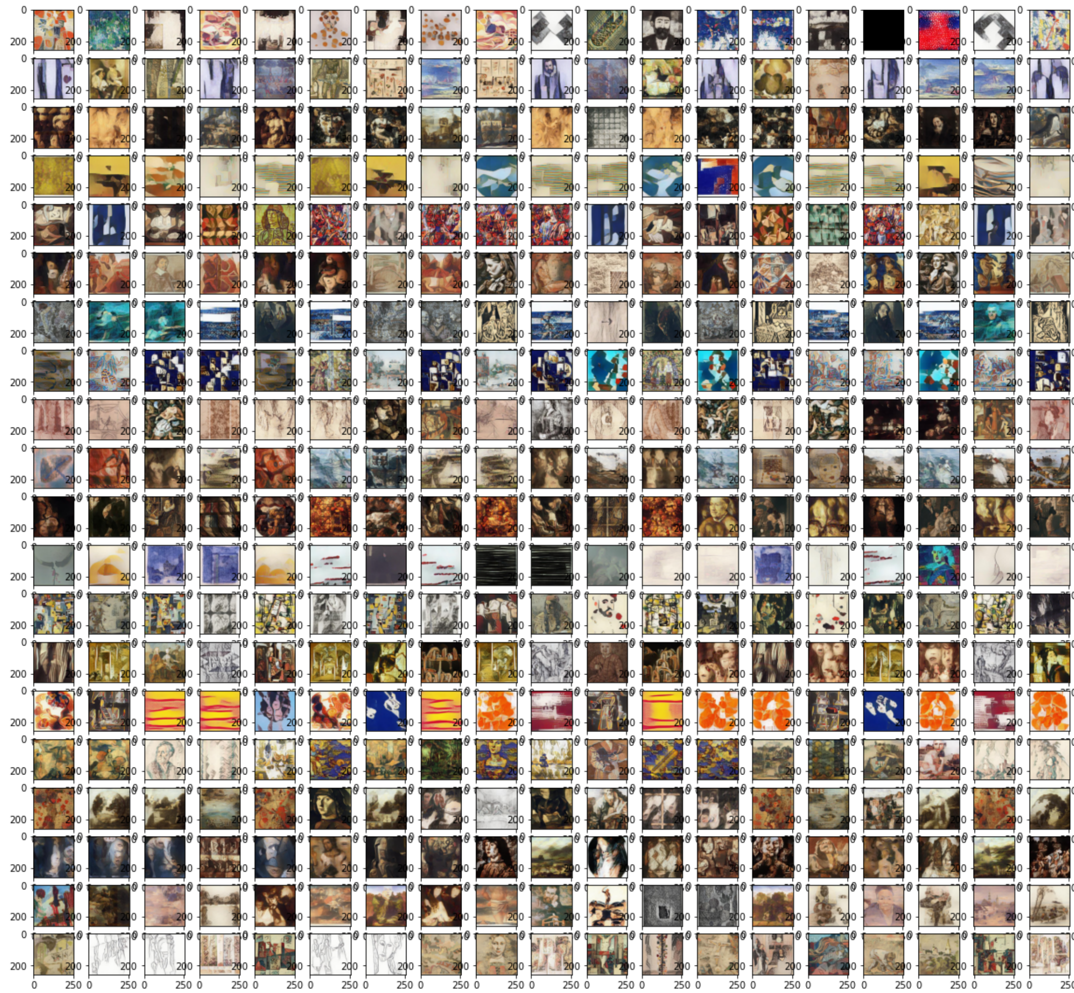

Fig. 8: 380 selected samples from 3800 pastiches


Fig. 9: Style transfer

### C. Style Recognition of Pastiches

Once we synthesize the modified paintings and complete training the neural network on the original paintings, we then test the modified paintings. The classification network will go over the 3800 modified paintings and will predict what style the modified painting belongs to.

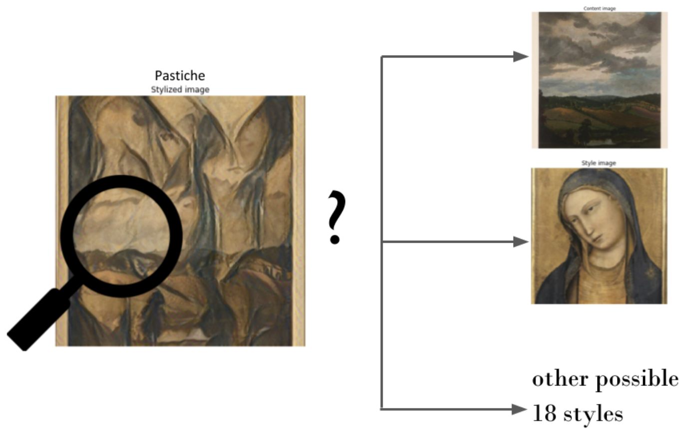


Fig. 10: Possible options for style classification model examining pastiche

Figure 10 provides a visual example of the style recognition of the pastiches, which refers back to the pastiche created in Figure 9. In this example, the content art period was Romanticism, and the style art period was the Early Renaissance. In this instance, we want to determine if the most closely related art period of the pastiche is Romanticism (based on content), Early Renaissance (based on style), or neither (looks like another art period).

We are interested in how often the model will classify the modified painting by content painting style versus by style painting style. There is a heatmap for each classification benchmark (one for content and one for style), which contains all of the relevant percentages, for each hybrid combination. As there are many values, we have moved those results to the appendix (Figures 12 and 13). Figure 11 provides a simplification by taking the average percentages for each art period and plotting the data as a bar graph.

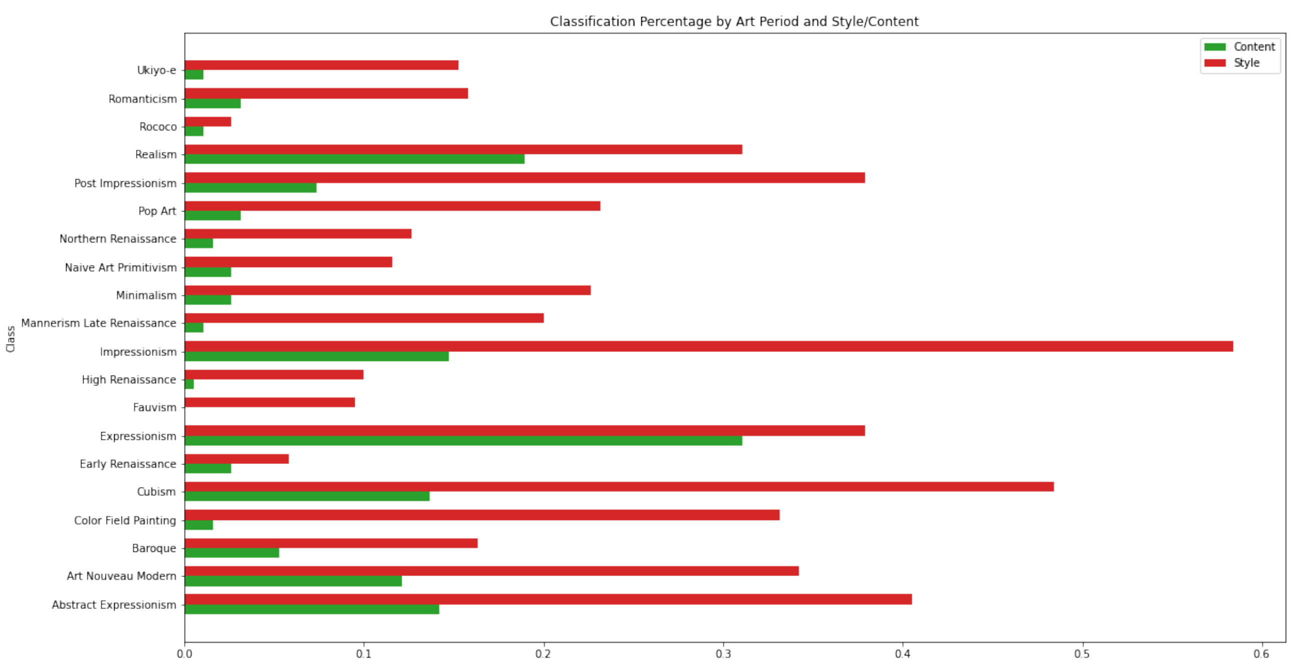


Fig. 11: Comparison of average classification accuracy by style/content

It is clear from this plot that pastiches are recognized with higher percentage (>30%) for the following 8 style art periods (indicated by the red): Abstract Expressionism, Art Nouveau Modern, Color Field Painting, Cubism, Expressionism, Impressionism, Post Impressionism and Realism. Their respective average percentages were 40.5%, 34.2%, 33.1%, 48.4%, 37.8%, 58.4%, 37.8%, and 31%. When classifying with content art period (indicated by the green), there were no values greater than 30%, except for Expressionism (31.1%).

The pastiches' higher affinity to their style parents than their content parents suggests that the style "gene" is more important than the content "gene", when classifying the art periods. This also suggests that the style of a painting might be influenced by its content, but not significantly. We thought styles with a distinctive visual appearance, such as Ukiyo-e, Minimalism, or Color Field Painting, would have achieved higher percentages, as demonstrated by Lecoutre et al.'s testing [1]. However, this was not the case.

No certain relationship between specific style and the classification of the pastiche can be found in Figures 11 to 13. For example, we could not determine if Impressionism is a more prominent style than Fauvism. The size of each class might be a confounding variable. When training the classifier, we found that the bigger the class, the better the classification. Goodfellow et al. corroborates this, estimating that 5000 examples per class is needed to achieve approximate human-level performance [2]. Unfortunately, as seen from Figure 4, there is great disparity in class sizes (e.g., Impressionism is 10 times greater than High Renaissance). This means that the classifier was stronger for bigger art periods, which likely affected the classification of the pastiches. For example, larger classes such as Expressionism (6736 paintings) had higher style and/or content percentages (37.8% and 31.1%, respectively). Meanwhile, the classes that did not reach the 5000 paintings threshold did not have high percentage averages.

Overlap might be another possible cause for the lower percentages. For example, styles such as High Renaissance, Mannerism Late Renaissance, and Early Renaissance are all connected to one central movement, meaning that there is great similarity between these classes. This could impact the classifier's performance if the distinctions between classes were not clear.

## V. Conclusions

In this paper, a novel multi-step approach was developed and utilized to analyze the components of a pastiche. The approach was implemented with the Python language by integrating two machine learning libraries, FastAI and TensorFlow. Our classifier was able to achieve 57% accuracy, which is comparable with other studies that also achieve 60% [1].

3800 pastiches were created in this study by using paintings for both style and content. WikiArt dataset, the most common dataset for art-related computer vision research, was used in this study by combining and trimming a total of 27 different art styles to the amount of 20 styles. We found that the style "gene" is more prominent than the content "gene" when classifying the pastiche.

Our results ultimately did not show any significant pattern in the data between specific art period parents and classification of the pastiche, except for size of each class. This demonstrates our results were more dependent on our trained classifier than on the pastiches themselves.

We hope our research accelerates the field of computer vision, as well as providing some insight to creatives. This research is of particular value to all creatives - novelists, composers, artists. Our findings on the relationship between content/style and the classification of style may affect their future works and their approach to the creative process. For example, for a musician developing their own personal style: how much focus should be placed on genre (e.g., becoming jazzier; the style) versus on the content (e.g., the notes of the song)? Our study suggests that creatives should prioritize style over content (but not neglect content), when developing their own brand.

We also believe our research holds worth primarily in the style recognition. To state briefly, this research could help museums with quickly annotating paintings. Style recognition has practical use in indexing online museum databases, as manually creating metadata can be laborious and inefficient. Additionally, manual style recognition needs the expertise of art historians. Automated recognition provides a fast and cheap solution, if properly implemented.

The only metric we could compare to other studies was the accuracy of our classification model. However, because we use a novel approach, our overall results cannot be compared.

There are still many possibilities when pursuing better results. For future work, we would like to address all the limitations listed below. We believe that if we address these issues, we will be able to obtain more substantial results.

## VI. Limitations

Throughout our process, there were many setbacks that have affected the overall experimentation. We encountered frequent bugs, meaning that we constantly had to adapt to new solutions. For example, we transitioned half of our code from using the TensorFlow library to the FastAI library. Because of the learning curve involved in understanding each library, our development time was inevitably prolonged.

There were also errors with the computer that we used. We used Google Colab, a free computational service primarily targeted for researchers. When unzipping the WikiArt dataset, we encountered frequent errors, primarily due to the zip file's large size of 25 gigabytes. Our primary method of storage was mounting Google Drive to Google Colab, a very makeshift solution, which led to occasional input/output errors. In general, when doing any long computation, Google Colab constantly timed out, significantly extending the testing time.

Because of this, we considered switching services from Colab to another service. Google Cloud seemed to be a viable solution, as it gave 300 dollars in free credits. However, Google Cloud's limit on graphical processing units (GPUs) for free users meant our computations would be lengthened. GPUs are an essential part of machine learning, due to their immense computational power. Additionally, all of our progress in Colab would be essentially erased by changing services. Therefore, we decided to stick with Colab.

In the end, we were only able to run 7 epochs in total. Each epoch took an average of 4 hours and 13 minutes. Times ranged from 2 and half hours to 6 and half hours because of the variable availability of faster GPUs. Sometimes, GPUs were just unavailable, as resources are not guaranteed. As stated on Google's website: "In order to be able to offer computational resources for free, Colab needs to maintain the flexibility to adjust usage limits and hardware availability on the fly. The GPUs available in Colab often include Nvidia K80s, T4s, P4s and P100s. There is no way to choose what type of GPU you can connect to in Colab at any given time" [18]. This meant that attempting to run more epochs would have been unreliable.

Ideally, we would want to train the model many more epoches (e.g., around 100 times). This would theoretically minimize inaccuracies in the classification neural network as much as possible. However, we saw declining reward in training more epochs and the infeasibility of time required.

One frequent finding of machine learning is that more data means better performance. As mentioned previously, Goodfellow et al. recommends at least 5000 examples for training each class. The number of examples per class varied greatly; ideally, each class would have been balanced equally. Our average number of examples per class was 3846, which was nonoptimal and lower than desired.

One possible way we could have increased the strength of our classifier would be to simply alter the training images. This is done by transforming, resizing, flipping, distorting, or other manipulations, in order to make the classifier more robust. This method can be effective, as it does not require more new data; it simply utilizes the dataset given. However, with this method, our dataset would be doubled or multiplied by a greater factor, which would also increase the training time needed.

Another way we could have increased the strength of our classifier would have been to use another neural network architecture. We could have also increased the depth of the neural network (e.g., ResNet50 or ResNet101). Residual neural networks with more layers are uniquely able to better fit the data and predict more accurately. When adding layers for other previous models, there is a degradation problem: accuracy increases and then degrades rapidly [15]. Once again, the main disadvantage is the high computational costs, which would extend the training time. This was why we compromised with ResNet34 which had 34 layers.

As for miscellaneous errors, one mistake was that we accidentally created double the amount of pastiches. Essentially, we created 4 matrices instead of just the 2 illustrated matrices. We simply ignored the 2 duplicates. This was not necessarily a limitation as we gained more results that could have been used; however, extra unplanned time was spent in the process due to this error.

One issue with machine learning and paintings is the differing ratios. Most convolutional neural networks limit the input images to a fixed number of pixels. This means that the paintings are cropped or distorted to fit a particular aspect ratio, which may lead to a loss of details in both the brushstrokes and the composition. This information is crucial for identifying style, as losing these details can lead to worse performance.

Finally, while WikiArt is widely used in similar papers [1], [7], [10] due to its large size, WikiArt might be susceptible to incorrect labelling. Their website relies on crowdsourcing, and not necessarily professional and reliable art historians. If the data is compromised with wrong labels, then the classifier will naturally also be compromised. If incorrect styles were frequent, both our model accuracy and our matrix percentage results would have been lower than expected.

## REFERENCES


[1] A. Lecoutre, B. Negrevergne, and F. Yger, "Recognizing art style automatically in painting with deep learning," in *Proceedings of the Ninth Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M.-L. Zhang and Y.-K. Noh, Eds., vol. 77. PMLR, 15–17 Nov 2017, pp. 327–342. [Online]. Available: http://proceedings.mlr.press/v77/lecoutre17a.html

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song, "Neural style transfer: A review," *CoRR*, vol. abs/1705.04058, 2017. [Online]. Available: http://arxiv.org/abs/1705.04058

[4] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015. [Online]. Available: http://arxiv.org/abs/1508.06576

[5] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *CoRR*, vol. abs/1610.07629, 2016. [Online]. Available: http://arxiv.org/abs/1610.07629

[6] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," *CoRR*, vol. abs/1603.08155, 2016. [Online]. Available: http://arxiv.org/abs/1603.08155

[7] A. M. Elgammal, M. Mazzone, B. Liu, D. Kim, and M. Elhoseiny, "The shape of art history in the eyes of the machine," *CoRR*, vol. abs/1801.07729, 2018. [Online]. Available: http://arxiv.org/abs/1801.07729

[8] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," *Expert Systems with Applications*, vol. 114, pp. 107 – 118, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417418304421

[9] J. Chen and A. Deng, "Comparison of machine learning techniques for artist identification," 2018. [Online]. Available: http://cs229.stanford.edu/proj2018/report/41.pdf

[10] C. Sandoval, E. Pirogova, and M. Lech, "Two-stage deep learning approach to the classification of fine-art paintings," *IEEE Access*, vol. 7, pp. 41 770–41 781, 2019.

[11] X. Shen, A. A. Efros, and M. Aubry, "Discovering visual patterns in art collections with spatially-consistent feature learning," 2019.

[12] N. Garcia, B. Renoust, and Y. Nakashima, "Understanding art through multi-modal retrieval in paintings," *CoRR*, vol. abs/1904.10615, 2019. [Online]. Available: http://arxiv.org/abs/1904.10615

[13] S. Hicsonmez, N. Samet, F. Sener, and P. Duygulu, "Draw: Deep networks for recognizing styles of artists who illustrate children's books," 2017.

[14] [Online]. Available: https://morningpicker.com/technology/deep-learning-is-one-of-the-hottest-tech-topics-right-now-heres-everything-you-need-to-know-about-dl-67474/

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[16] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," 2017.

[17] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," 2017.

[18] "Colaboratory," https://research.google.com/colaboratory/faq.html, accessed: 2020-05-26.

# VII. Appendix


Fig. 12: Percentage of pastiches recognized as style art period

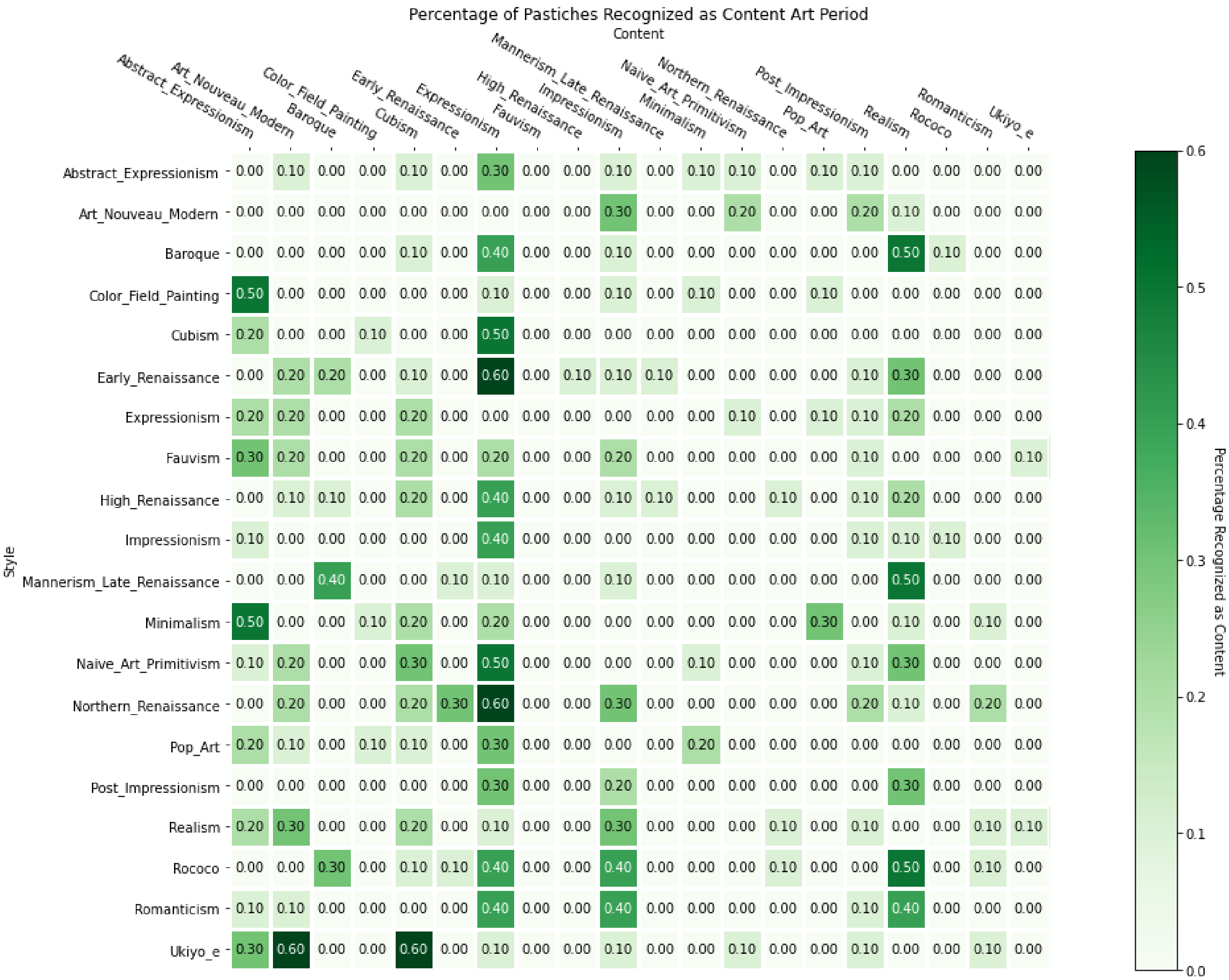


Fig. 13: Percentage of pastiches recognized as content art period