# 6.7900 Final Project: Exploring Artwork Style-Content Interaction in Image Classification Models

Katie Chen and Andrew Liu

December 15, 2023

## Abstract

In this project, we study the problem of classifying artwork by *style* using artistic images in the ArtBench dataset [7]. We find that deep convolutional neural networks (CNNs) like ResNet50 [4] and VGG19 [8] are much better suited to learning artwork style than diffusion-based classifier models, where ResNet50 achieves 97% classification accuracy on a binary classification task, VGG19 achieves 99%, and a diffusion-based model achieves 81%. We further investigate the success of VGG19 in its classification task with a series of manual feature extraction tasks. In particular, by training linear probes with features extracted at different layers in the network, and introducing visual heuristics with Principal Component Analysis (PCA), we find that the depth of VGG19 plays a key role in its success classifying artwork. The models' ability to confidently express style-related learning almost directly correlates with model depth, ultimately suggesting that *style* in artwork is a layered visual characteristic.

## 1 Introduction

The field of artwork classification via neural networks has gained significant traction propelled by advancements in deep learning. The interplay between *style* and *content* in artwork has emerged as a subject of significant interest and challenge. We are interested in exploring which modern architectures are best suited to learn image *style*, and how architectures which are effective are able to perceive and process the style-content interaction in artwork. The first notions of style-content interaction in image classification were introduced by Gatys et al. [3], in their paper introducing the Neural Style Transfer algorithm.

*Style*, as defined by Gatys et al., is the "texturised versions of the input image that capture its general appearance in terms of colour and localised structures" [3]. *Content* refers to the discernible objects and figures depicted within the artwork, independent of artistic style. Artwork analysis is particularly complex because nuances in style and content make learning artistic images not just a matter of recognizing pixel-based patterns, as is the case for most image classification tasks; instead, there are also intangible and subjective elements at play.

In our project, we first compare how deep CNN architectures, particularly ResNet50 and VGG19, compare to diffusion model classifiers in the style classification task. We fine-tune these deep CNNs on the ArtBench dataset [7], and implement the diffusion classifier in [6]. After showing that neural net architectures are more effective, we then further investigate how they are able to break down the style-content representation of artwork effectively through feature extraction of the networks. We select five different layers in the network to extract features from and train linear probing classifiers to study the evolution of how the network perceives the image. We also visualize this evolution by employing PCA on the extracted features. Ultimately, we find that deeper layers in the network capture predominantly style-related information to more content-specific details, suggesting a layered understanding to the visual characteristics that define an artwork.

## 2 Related Works

A notable contribution in this domain is the work by Gatys et al. (2016) [3], which elegantly demonstrated how CNNs can separate and recombine the content and style of natural images, laying the groundwork for automated artistic style transfer. Work like Elgammal et al. [2] then furthered the narrative by applying deep learning to classify artworks into historical periods, leveraging the nuanced feature extraction capabilities of CNNs to discern artistic styles.

The exploration of style and content interaction in artworks using neural networks has opened new avenues in understanding the interpretability of these models. Zeiler and Fergus [9] made early strides by visualizing the intermediate layers of CNNs, providing insights into the hierarchical feature learning process. This methodology shed light on how neural networks perceive artistic content and style. Studies such as those by Bau et al. [1] have delved deeper into interpreting the layers of neural networks, particularly in the context of generative models.

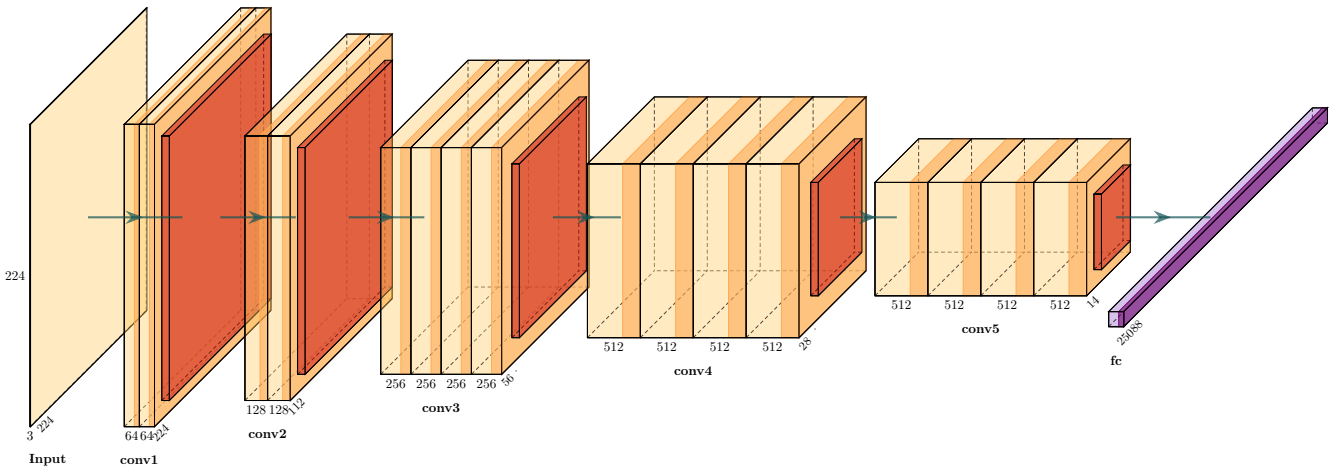



Figure 2: VGG19 architecture. We fine-tuned pre-trained weights with ArtBench [7], where the final connected layer is modified to a binary classifier, or multiclass classifier. We trained linear probes on the features at the output of the five max pooling layers (dark orange) directly following each convolutional block: `conv1`, `conv2`, `conv3`, `conv4`, and `conv5`. In total, this network has 140M parameters.

The goal of our project is to combine the methods established by these works to produce similar results on a dataset that, unlike most other datasets, is specifically separated by style, rather than content.

# 3 Methodology

## 3.1 Dataset

We use ArtBench-10, first introduced by Liao et al [7]. It is the first class-balanced, high quality, cleanly annotated and standardized dataset for benchmarking artwork generation. It inlcudes 60,000 images, segmented into 10 artistic styles: `art_nouveau`, `baroque`, `expressionism`, `impressionism`, `post_impressionism`, `realism`, `renaissance`, `romanticism`, `surrealism`, `ukiyo_e`. Each style has 5,000 training images and 1,000 testing images. Crucially, because this dataset is purely stratified by style, we make the assumption that content is distributed uniformly across all style classes. We justify this with the advantages that the Artbench dataset asserts over previous artwork datasets, which is that it maintains class-balance and a high-quality image and annotation regime [7]. This allows us to reason about networks' ability to learn style only.

For training, we use the 256x256 resolution images, cropped to 224x224 resolution, for the CNN architectures and the 32x32 resolution images for the diffusion classifier. For all architectures, we employ data augmentations like horizontal and vertical flips to the training set to help reduce overfitting.

## 3.2 Models

We consider the following models: the deep CNNs ResNet50 [4], VGG19 [8], as well as the diffusion classifier architecture introduced by Li et. al [6]. ResNets incorporate skip connections to enable training of very deep networks, leading to a more layered understanding of features and better performance on complex tasks, ultimately incentivizing us to investigate whether they could represent style effectively. The VGG architecture is used in the original Neural Style Transfer algorithm [3]; see Figure 2 for the exact architecture we use.

Using a diffusion model as a classifier was a novel approach to image classification, diverging from the typical discriminative model paradigm. Since CNNs and other architectures that specialize in image recognition are typically content-based learners, we were motivated to try a novel architecture. Diffusion models as classifiers could offer an advantage in handling the variability within artistic styles. Styles like surrealism exhibit a high degree of intra-class variation, which is suitable for diffusion models' generative nature. This characteristic could be particularly beneficial in differentiating artworks that share content features but diverge stylistically.

## 3.3 Classification Task

For each of the CNN architectures, we modified the last fully connected layer in two ways: (1) as multiclass classifiers, to identify the style belonging to a given input image, and (2) as binary classifiers, to identify whether a given image was `ukiyo_e` or `surrealism`. We then fine-tuned pre-trained versions of the weights across the whole network. For the diffusion classifier, we closely followed the procedure described in [6]; more details about our exact training procedure are listed in Appendix B.

| model | art | bar | exp | imp | pos | rea | ren | rom | sur | uki |
|---|---|---|---|---|---|---|---|---|---|---|
| acc | 0.58 | 0.70 | 0.45 | 0.37 | 0.38 | 0.36 | 0.74 | 0.46 | 0.71 | 0.96 |

Table 1: VGG19 test accuracy by style

Testing accuracy results after training for the three models are listed in Table 2.

| | ResNet50 | VGG19 | Diffusion |
|---|---|---|---|
| binary | 0.97 | 0.98 | 0.81 |
| multi | 0.48 | 0.57 | 0.28 |

Table 2: Fine Tuned model results (test accuracy)

Although results from the diffusion model were promising, interpretability in the context of classification is a challenge, in part because inference for the procedure described in [6] is very slow, on the order of minutes per image. The model architecture is also much less transparent than the CNN architectures, since it does not produce any intermediate features that are directly optimized for classification; they are instead optimize for a loss metric that does not lend itself to interpretability (see Appendix B for details). It is worth noting that, when trained on MNIST, our implementation of the diffusion classifier obtained 97% testing accuracy, which suggests that the diffusion classifier is capable of learning the content aspect of images well. This intuitively makes sense given that the architecture for backwards inference, the UNet, is essentially another CNN that is optimized for a different image metric. Ultimately, due to the lack of easily interpretable features, we decided not to study this architecture further.

The ResNet50 managed to achieve better results. Despite this, there was a large gap in testing vs. training accuracy in the multiclass case; the testing accuracy was 92%, which suggests that the model was overfit. Our best attempt to explain this overfitting, given that VGG19 and ResNet50 are both deep neural architectures who we initially expected to behave similarly, is that the ResNet50 was overfitting on content differences in the classes and not really interpreting style correctly. This is easily reinforced by skip connections, since features closer to the original image are emphasized many times over the course of learning.

VGG19 had the best classification accuracy in both tasks. This model is well known for its transfer learning capabilities, and it intuitively makes sense why it might perform better than the other models: it has very large depth, with roughly 140M trainable parameters compared to ResNet50s 23M, and style is a complex trait that we expect deeper networks to be able to learn better.

Thus, we further analyze the VGG19 architecture to investigate aspects of the network that makes it successful at learning style. Our decision to simplify the rest of our analysis to binary classfiication was motivated by the results in Table 1; the model seems to recognize `surrealism` and `ukiyo_e` most effectively, so for the remainder of our analysis, we only consider these styles. Henceforth, `VGG_10_class` refers to the fine-tuned 10 class VGG, and `VGG_2_class` refers to the fine-tuned binary classifier.

# 4 Results and Analysis

## 4.1 Feature Extraction

We extracted features from `conv1_1`, `conv2_1`, `conv3_1`, `conv4_1`, `conv5_1`, corresponding to the layers `features.0`, `features.5`, `features.10`, `features.19`, `features.28` from the `VGG_2_class` classifier (see Figure 2). We chose to analyze these layers to match the layers chosen in the paper by Gatys et al. [3].

Using these extracted features, we train a simple logistic classifier. Our results are shown in Table 3.

| conv1_1 | conv2_1 | conv3_1 | conv4_1 | conv5_1 |
|---|---|---|---|---|
| 0.82 | 0.82 | 0.93 | 0.95 | 0.97 |

Table 3: Linear Probing Classifier Results

The increasing accuracy suggests that higher-level features are more effective in distinguishing between `surrealism` and `ukiyo_e`. Early layers like `conv1_1` and `conv2_1` tend to capture stylistic features like edges and textures. The deeper layers like `conv5_1` are capable of capturing high-level content. The way objects are depicted in `surrealism` is inherently different from `ukiyo_e`, not just in color or texture but also in form and composition. This might explain why deeper layers, which capture these complex interactions, perform better in classification. Different art styles also emphasize different spatial hierarchies. `ukiyo_e`, for example, has a distinct flatness and pattern-oriented approach, while surrealism often plays with perspective and depth in unconventional ways. Deeper layers of the network, which capture higher-order spatial hierarchies, are likely better at identifying these style-specific spatial characteristics.

In addition to our linear probes, we also create a visual heuristic for style understanding with PCA. For each of the five layers that we probed, we fit a PCA matrix on features extracted from the testing set; the two principal components are plotted in Figure 6. The PCA results from layer `conv5_1` shows the two classes almost perfectly

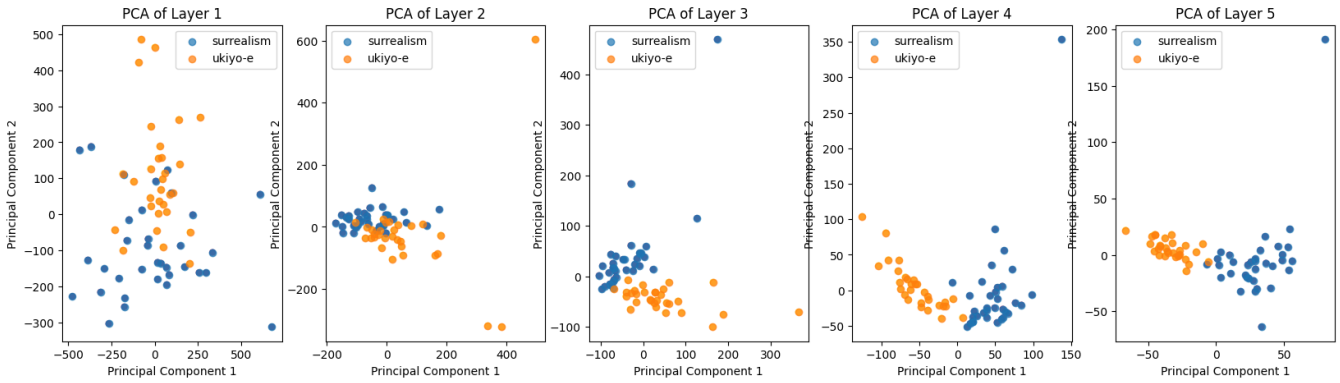



Figure 6: First two Principal Components from manually extracted features

separated along the first principal component, which suggests the extracted features provides a highly discriminative representation of the two art styles. Surrealism also apprears to be more spread out in the `conv5_1`. We suspect that this is caused by the inherent style variability; surrealism is known for its broad range of styles and techniques. This diversity could result in a wider spread of feature representations even after being abstracted by the deeper layers of the network.

### 4.2 Content and Style Reconstruction

We implement the methods outlined by Gatys et. al [3] and reconstructed style and content from `conv1_1`, `conv2_1`, `conv3_1`, `conv4_1`, `conv5_1`, using both `VGG_10_class` and `VGG_2_class`. We use a surrealist image that the model has not seen before; results are shown in Figure 9. We observe that at `conv2_1`, `VGG_2_class` had more finer details when reconstructed, where as at `cov3_1` `VGG_10_class` has more finer details. We hypothesize this might be due to class-specific fine-tuning and feature generalization, in a sense that `VGG_2_class` is fine-tuned specifically on surrealism and ukiyo-e, which may lead it to develop more specialized filters at earlier layers for distinguishing between these two styles. In contrast, `VGG_10_class` is exposed to a greater variety of images during training. This exposure can lead to the development of more robust and varied feature maps at deeper layers to accommodate the broader variance in the dataset, resulting in richer detail captured at the `conv3_1` layer.

### 4.3 Case Study with Neural Style Transfer

Neural Style Transfer [3] is an algorithm that takes a content image and style image as input, and produces an image that has similar content to the first image in the style of the second image. Further, arbitrary Neural Style Transfer [5] is a modification that allows us to combine the content image with a weighted factor $\alpha$ of the style of the style image, where $0 \leq \alpha \leq 1$. In Figure 10, we visualize an example of an arbitrary `ukiyo_e` content image processed with arbitrary neural style transfer from a surrealist style image, with varying levels of $\alpha$. In Figure 11, we track the location of this image representation in the PCA space across different levels of the network. Note that in all five layers, the starting location ($\alpha = 0$) is always in the cluster corresponding to the `ukiyo_e` style, while the final location ($\alpha = 1$) is always in the cluster corresponding to the `surrealism` style. Further, the trajectory taken from $\alpha = 0$ to $\alpha = 1$ appears to almost always be a direct path. This ultimately suggests that at all points in the network there is some understanding of the differences between `surrealism` and `ukiyo_e`. Note further that the distance travelled in the PCA space in the first layer is much larger than that of the other layers, suggesting a higher level of variability in earlier layers, a result that we expect. This is an informal visual heuristic that reaffirms our result that style is learned better in deeper layers.

## 5 Conclusion

In this project we explored the problem of style-content interaction in image classification networks as it pertains to artwork. In particular, we find that, despite the fact that deep CNNs are not known to learn intangible stylistic features as effectively as they are for tangible shapes, they still outperform an alternate model of image classification with diffusion. The VGG19 network achieved test accuracy of 98% after minimal fine tuning, which attests to its expressive power and depth.

We used manual feature extraction with a number of different tasks to investigate aspects of the VGG19 network that contributed to its expressive power in style. By training linear probes at 5 different points in the network, we find that the training accuracy increases in later layers, which suggests that deeper layers have a better understanding of image style. We visually confirmed this result with PCA.

# References


[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. *Network Dissection: Quantifying Interpretability of Deep Visual Representations*. arXiv:1704.05796 [cs]. Apr. 2017. DOI: 10.48550/arXiv.1704.05796. URL: http://arxiv.org/abs/1704.05796 (visited on 11/17/2023).

[2] Ahmed Elgammal, Marian Mazzone, Bingchen Liu, Diana Kim, and Mohamed Elhoseiny. *The Shape of Art History in the Eyes of the Machine*. Feb. 2018. URL: https://arxiv.org/abs/1801.07729 (visited on 12/15/2023).

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. *A Neural Algorithm of Artistic Style*. arXiv:1508.06576 [cs, q-bio] version: 2. Sept. 2015. URL: http://arxiv.org/abs/1508.06576 (visited on 11/15/2023).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 [cs]. Dec. 2015. URL: http://arxiv.org/abs/1512.03385 (visited on 11/30/2023).

[5] Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. Mar. 2017. URL: https://arxiv.org/abs/1703.06868 (visited on 12/15/2023).

[6] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. *Your Diffusion Model is Secretly a Zero-Shot Classifier*. en. (Visited on 11/15/2023).

[7] Peiyuan Liao, Xiuyu Li, Xihui Liu, and Kurt Keutzer. *The ArtBench Dataset: Benchmarking Generative Models with Artworks*. arXiv:2206.11404 [cs]. June 2022. URL: http://arxiv.org/abs/2206.11404 (visited on 11/01/2023).

[8] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. en. Sept. 2014. URL: https://arxiv.org/abs/1409.1556v6 (visited on 11/16/2023).

[9] Matthew D. Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*. arXiv:1311.2901 [cs]. Nov. 2013. DOI: 10.48550/arXiv.1311.2901. URL: http://arxiv.org/abs/1311.2901 (visited on 11/17/2023).


# A Fine-tuning Loss and Training accuracy on deep CNN architectures

The following figures depict training and test loss and accuracies for both of the deep CNN architectures that we fine-tuned for the baseline classification task. All graphs depicted are for the models fine-tuned for binary classification.

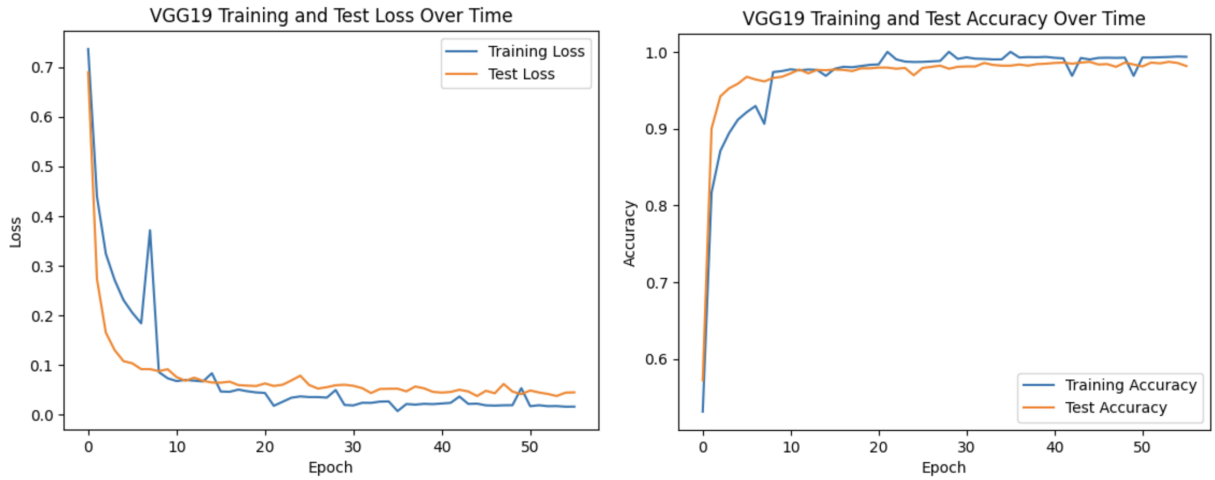



Figure 7: VGG19 fine-tuning training/testing loss, and training/testing accuracy

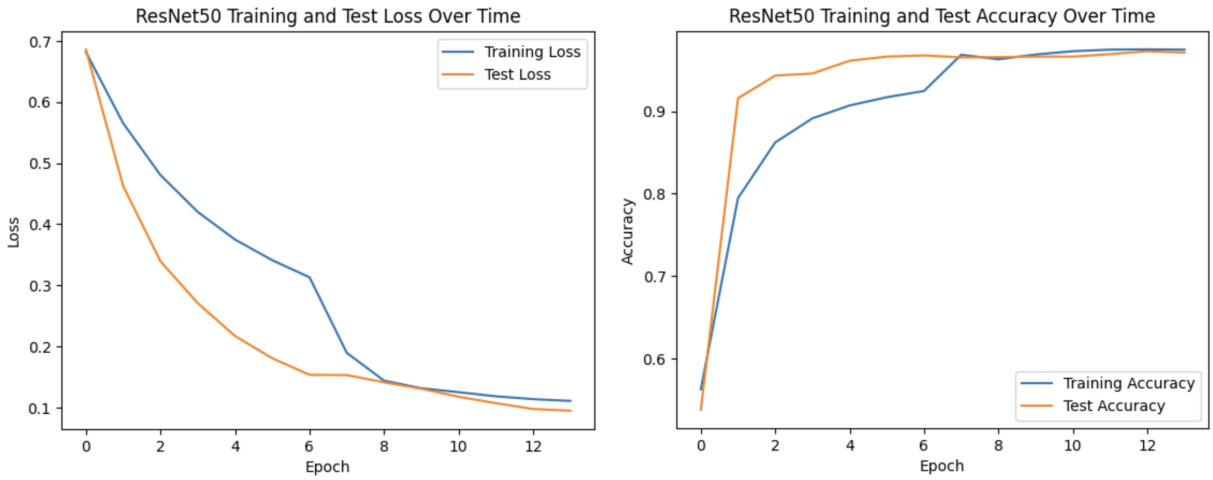



Figure 8: ResNet50 fine-tuning training/testing loss, and training/testing accuracy

# B Diffusion training details

The procedure for turning any diffusion model into a classifier, as described in [6], is as follows. Normal class-conditional diffusion models are trained to optimize the ELBO objective, given by

$$\mathcal{L}(c_i) = -\mathbb{E}_{t,e}[\|\varepsilon - \varepsilon_\theta(x_t, c_i)\|^2],$$

where $\varepsilon$ is the true backwards noise at timestep $t$, given image $x_t$ and class $c_i$. In order to predict the class given an image, we want

$$p_\theta(c_i|x) = \frac{p(c_i)p_\theta(x|c_i)}{\sum_j p(c_j)p_\theta(x|c_j)}.$$

Since the loss (i.e., approximately the ELBO) is the log likelihood of the probability of any given input, we can approximate

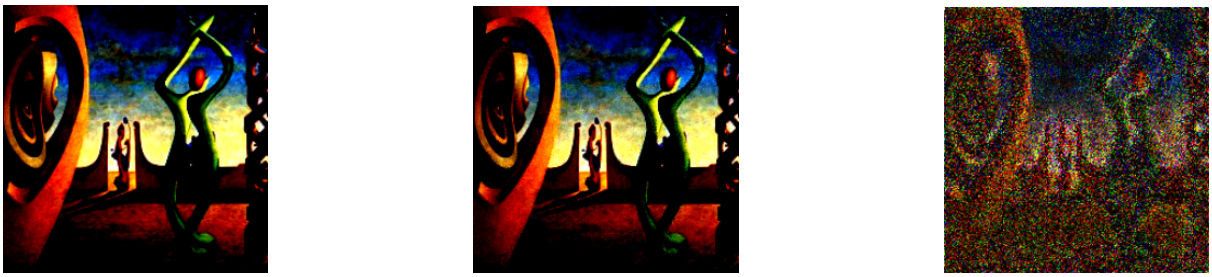
$$p_\theta(c_i|x) \approx \frac{\exp\left(\mathcal{L}(c_i)\right)}{\sum_j \exp\left(\mathcal{L}(c_j)\right)}.$$

Each $\mathcal{L}(c_i)$ can be approximated with a Monte Carlo sampling from $(t_i, \varepsilon_i)$ to approximate the expected value, thus giving us a classifier.

With this method, we found training and inference to be a large bottleneck. For inference, accurate Monte Carlo estimates requires the number of samples to be on the order of thousands before results are not nonsensical. Each image prediction took our network roughly 3 minutes on the smaller image size (32x32), with 1000 samples per image. When we originally attempted to implement this method with larger image sizes (256x256), as with the other methods, training was on the order of an hour per epoch. After training the larger classifier for around 20 hours, we realized that any attempt to optimize or interpret the model would not be realistic.

We attempted to implement the classifier with a number of different architectures, including UNets with and without self attention, $T$ timesteps equal to 1000 and 4000, and with linear and cosine scheduling. Out of all of these combinations, we found that $T = 4000$, cosine scheduling, and self attention produced the best results. Since we were restricted to 32x32 images due to the time and compute required to train a full model, our classification results were technically not a fair comparison with the deep CNN classifier results.

## C Style-content reconstruction



(a) VGG_2_class conv1_1 (b) VGG_2_class conv2_1 (c) VGG_2_class conv3_1

(d) VGG_10_class conv1_1 (e) VGG_10_class conv2_1 (f) VGG_10_class conv3_1

Figure 9: Top Row: content reconstructed from VGG_2_class. Bottom row: content reconstructed from VGG_10_class. Notably, the 2 class reconstructions are sharper, since the network is more finetuned.

## D Neural Style Transfer Trajectories

Figure 10: Arbitrary neural style transfer with a surrealist style image (very right), on a ukiyo_e content image, with alpha values $\alpha = 0, 0.25, 0.5, 0.75, 1$ from left to right.

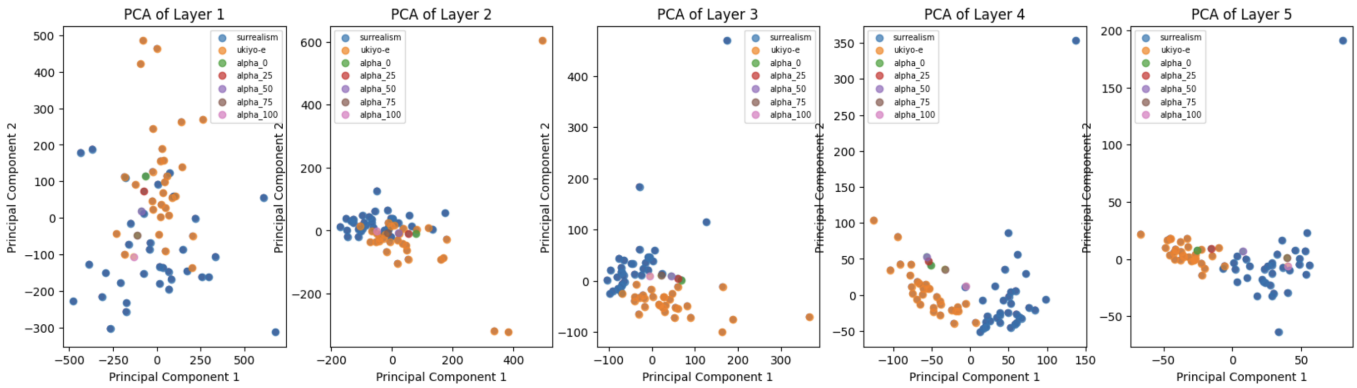



Figure 11: PCA trajectory of the arbitrary neural style transfer depicted in Figure 10

It is noteworthy that the pattern observed in Section 4.3 becomes even more pronounced when we repeat this procedure with multiple images:

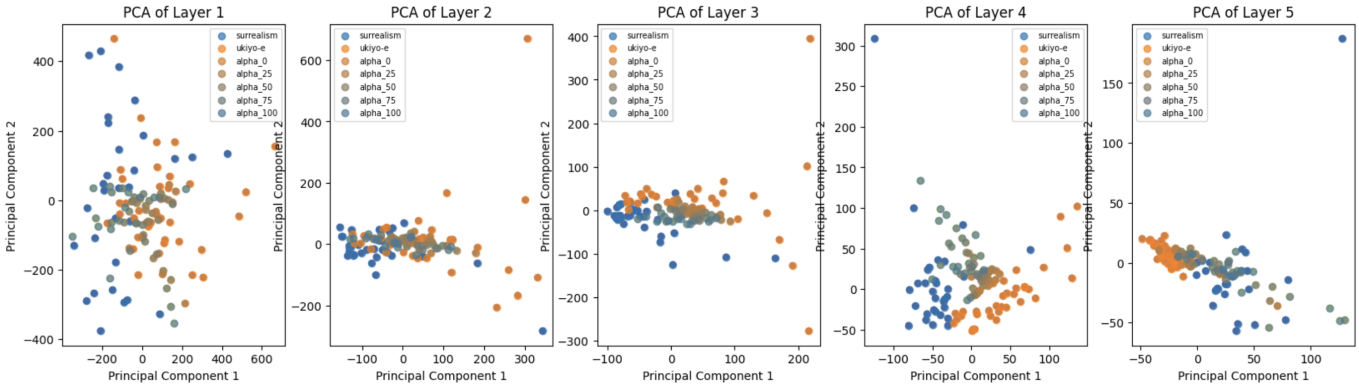



Figure 12: PCA trajectories

In Figure 12, we repeat the procedure in Figures 10 and 11 for twelve (`surrealist` style, `ukiyo_e` content) image pairs, where each pair contributes five data points ($\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$) to each of the five layers. The results are almost the same, but more pronounced, than the results in Section 4.3. In particular, there is a lot of variance in the first layer, which agrees with our hypothesis that style representations are not yet reliable early in the network. However, by layer 4, we observe a clear gradient from the orange `ukiyo_e` images to the blue `surrealism` images, with increasing $\alpha$ values creating linear layers. This suggests that this point in the network has very strong and reliable interpretations of style.