**CS 726: Advanced Machine Learning** **Jan-May 2019**

# Project Report :
# Unsupervised Learning of Artistic Styles with Archetypal Analysis

*Contributors : Ayush Khandelwal, Sneha Bhakare, Videsh Suman*

## 1 Main Goal

In this project, we try to use an unsupervised learning approach i.e. archetypal analysis [1] to automatically discover, summarize, and manipulate artistic styles from a collection of paintings [3]. We learn archetypal representations of style from an image dataset of Van Gogh's paintings. Having established that archetypal analysis is a natural tool for unsupervised learning of artistic style [3], we also show that it provides a latent parametrization allowing to manipulate style by extending the universal style transfer technique of [2]. Also, we try to avoid artifacts on stylization by improving over the update function for style transfer as discussed in [3].
Although after having been able to generate the archetypes in the form of style descriptors, we cannot visualize the archetypes due to the incomplete information provided in [3].

## 2 Related Literature

Artistic style transfer consists of manipulating the appearance of an input image such that its semantic content and its scene organization are preserved, but a human may perceive the modified image as having been painted in a similar fashion as a given target painting.
The main advantage of archetypal analysis over these other methods is mostly its better interpretability, which is crucial to conduct applied machine learning work that requires model interpretation. Archetypes are simple to interpret since they are related to convex combinations of a few image style representations from the original dataset, which can thus be visualized. Moreover, archetypal analysis offers a dual interpretation view: if on the one hand, archetypes can be seen as convex combinations of image style representations from the dataset, each images style can also be decomposed into a convex combination of archetypes on the other hand. Then, given an image, we may automatically interpret which archetypal style is present in the image and in which proportion, which is much richer information than what a simple clustering approach would produce.

## 3 Method

### 3.1 Learning a latent low-dimensional representation of style

The first part of the problem is to extract the feature maps of all the images in the dataset. For that, we used the first 3 layers of a pre-trained VGG-19 network. The VGG-19 encodes an image into a 3D feature map, that is flattened out to get a 2D representation of the form $\mathbb{R}^{p_l \times m_l}$ where $p_l$ is the number of channels and $m_l$ is the number of pixel positions in the feature map at layer $l$. Then, we obtain the 1st order and 2nd order statistics $\{\mu\mathbf{1}, \mathbf{\Sigma_1}, ..., \mu_{\mathbf{L}}, \mathbf{\Sigma_L}\}$ of each image's feature map using the following formulae:

$$\boldsymbol{\mu}_l = \frac{1}{m_l} \sum_{j=1}^{m_l} \mathbf{F}_l[j] \ \in \mathbb{R}^{p_l} \quad \text{and} \quad \mathbf{\Sigma}_l = \frac{1}{m_l} \sum_{j=1}^{m_l} (\mathbf{F}_l[j] - \boldsymbol{\mu}_l)(\mathbf{F}_l[j] - \boldsymbol{\mu}_l)^\top \ \in \mathbb{R}^{p_l \times p_l}$$

where $\mathbf{F}_l[j]$ represents the column in $\mathbb{R}^p_l$ that carries the activations at position $j$ in the feature map $\mathbf{F}_l$. A style descriptor is then defined as the concatenation of all parameters from the collection $\{\mu\mathbf{1}, \mathbf{\Sigma_1}, ..., \mu_\mathbf{L}, \mathbf{\Sigma_L}\}$, normalized by the number of parameters at each layer—that is, $\mu_1$ and $\Sigma_l$ are divided by $p_l(p_l+1)$, which was found to be empirically useful for preventing layers with more parameters to be over-represented. The resulting vector is very high-dimensional, but it contains key information for artistic style. Then, we apply a singular value decomposition on the style representations from the paintings collection to reduce the dimension to 512 while keeping more than 99% of the variance. Next, we show how to obtain a lower-dimensional latent representation.

## 3.2 Archetypal style representation

Given a set of vectors $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ in $\mathbb{R}^p_n$, archetypal analysis [1] learns a set of archetypes $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_k]$ in $\mathbb{R}^p_k$ such that each sample $\mathbf{x}_i$ can be well approximated by a convex combination of archetypes—that is, there exists a code $\alpha_i$ in $\mathbb{R}^k$ such that $\mathbf{x}_i \approx \mathbf{Z}\alpha_i$, where $\alpha_i$ lies in the simplex. Conversely, each archetype $\mathbf{z}_j$ is constrained to be in the convex hull of the data and

$$\Delta_k = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^k \text{ s.t. } \boldsymbol{\alpha} \geq 0 \text{ and } \textstyle\sum_{j=1}^k \boldsymbol{\alpha}[j] = 1 \right\}$$

there exists a code $\beta_j$ in $\Delta_n$ such that $\mathbf{z}_j = \mathbf{X}_j$. The natural formulation resulting from these geometric constraints is then the following optimization problem which can be addressed efficiently

$$\min_{\substack{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n \in \Delta_k \\ \boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_k \in \Delta_n}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{Z}\boldsymbol{\alpha}_i\|^2 \quad \text{s.t.} \quad \mathbf{z}_j = \mathbf{X}\boldsymbol{\beta}_j \quad \text{for all } j = 1, \ldots, k$$

with dedicated solvers [1]. Note that the simplex constraints lead to non-negative sparse codes $\alpha_i$ for every sample $\mathbf{x}_i$ since the simplex constraint enforces the vector $\alpha_i$ to have unit $l_1$-norm, which has a sparsity-inducing effect. As a result, a sample $\mathbf{x}_i$ will be associated in practice to a few archetypes, as observed in our experimental section. Conversely, an archetype $\mathbf{z}_j = \mathbf{X}_j$ can be represented by a non-negative sparse code $\beta_j$ and thus be associated to a few samples corresponding to non-zero entries in $\beta_j$. Encoding an image style into a sparse vector $\alpha$ allows us to obtain interesting interpretations in terms of the presence and quantification of archetypal styles in the input image. Next, we show how to manipulate the archetypal decomposition by modifying the universal feature transform of [2].

## 3.3 A new variant of Universal Style Transfer

We assume, in this section only, that we are given a content image $I_c$ and a style image $I_s$. We also assume that we are given pairs of encoders/decoders $(d_l, e_l)$ such that $e_l(I)$ produces the $l$-th feature map previously selected from the VGG network and $d_l$ is a decoder that has been trained to approximately invert $e_l$that is, $d_l(e_l(I)) \approx I$. Universal style transfer builds upon a simple idea. Given a content feature map $\mathbf{F}_c$ in $\mathbb{R}^{p\times m}$, making local features match the mean and covariance structure of another style feature map $\mathbf{F}_s$ can be achieved with simple whitening and coloring operations, leading overall to an affine transformation:

$$C^s(\mathbf{F}_c) := \mathbf{C}^s\mathbf{W}^c(\mathbf{F}^c - \boldsymbol{\mu}^c) + \boldsymbol{\mu}^s$$

where $\mu_c, \mu_s$ are the mean of the content and style feature maps, respectively, $C^s$ is the coloring matrix and $W^c$ is a whitening matrix that decorrelates the features. This operation can be simply summarized as a single function $C^s := \mathbb{R}^{p\times m} \to \mathbb{R}^{p\times m}$.

Of course, feature maps between network layers are interconnected and such coloring and whitening operations cannot be applied simultaneously at every layer. For this reason, the method produces a sequence of stylized images $\hat{I}_l$, one per layer, starting from the last one $l = L$ in a coarse-to-fine manner, and the final output is $\hat{I}_1$. Given a stylized image $\hat{I}_{l+1}$ (with $\hat{I}_{L+1} = I^c$), we propose the following update, which differs slightly from [2].

$$\hat{I}_l = d_l\left(\gamma\left(\delta C_l^s(e_l(\hat{I}_{l+1})) + (1-\delta)C_l^s(e_l(I^c))\right) + (1-\gamma)e_l(I^c)\right)$$

where $\gamma$ in $(0, 1)$ controls the amount of stylization since $e_l(I^c)$ corresponds to the $l$-th feature map of the original content image. The parameter $\delta$ in $(0, 1)$ controls how much one should trust the current stylized image $\hat{I}_{l+1}$ in terms of content information before stylization at layer $l$.

In contrast, the update in the original Universal Style Transfer paper [2] involves a single parameter $\gamma$ and is of the form: Notice that here the original image $I^c$ is used only once at the

$$\hat{I}_l = d_l\left(\gamma\left(C_l^s(e_l(\hat{I}_{l+1}))\right) + (1-\gamma)e_l(\hat{I}_{l+1})\right)$$

beginning of the process, and details that have been lost at layer $l + 1$ have no chance to be recovered at layer $l$. We present in the experimental section the effect of our variant. Whenever one is not looking for a fully stylized imagethat is, $\gamma < 1$, content details can be much better preserved with our approach.

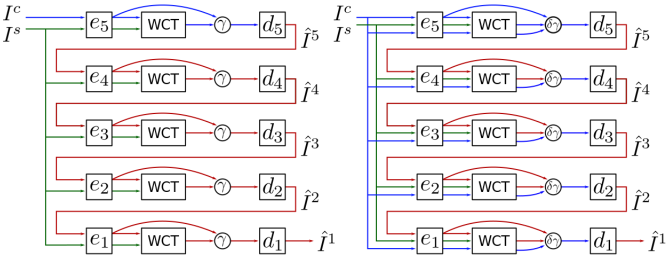

Figure 1: Multi-level Stylization: the left workflow corresponds to the parametrization used in [2], while the next one corresponds to that used in [3].

# 4 Experiments

The implementation has been carried out on Pytorch. To solve the constraint optimisation problem using archetypal analysis, we used the spams toolbox in Python. We started from the code implementation of [2]. Our code repository can be accessed from this link:
**https://github.com/sumanvid97/archetypal_style_analysis**.

We produced 32 archetypes from the dataset of 2046 images containing Vincent Van Gogh's paintings . We have successfully implemented the new variant of style transfer, also comparing it with the original variant in Figure 4. Also, we have the analyzed the semantic quality of the stylized images with respect to the values of $\gamma$ and $\delta$ in the Figure 5. We have obtained 32 archetypes from the Van Gogh's paintings' dataset in the form of latent vectors i.e. style descriptors. However, we couldn't reconstruct the images from the latent vectors using the pre-trained decoder network. Also, the information provided in the paper is incomplete regarding this reconstruction. We believe that it's only possible if we train a new decoder for this task, though there's no mention of it too in the paper.

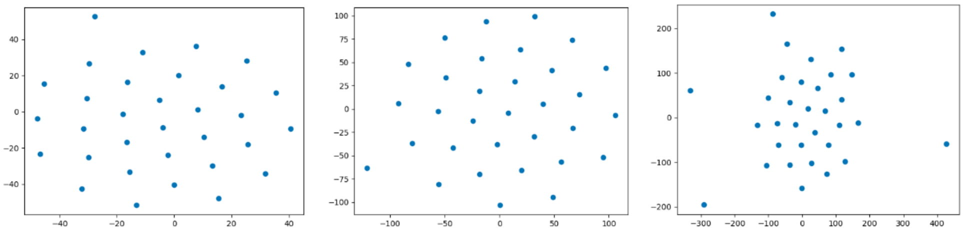

Figure 2: t-SNE embeddings of 32 archetypes computed on the Van Gogh paintings collection of 2046 images. Three scatter plots have been generated corresponding to 3 different configurations where we took feature maps from first, first two, and first 3 layers of the pre-trained VGG network. As the number of features increases the spread of 32 archetypes increase, thus indicating that the archetypes produced from higher number feature maps are more distinct amongst themselves.

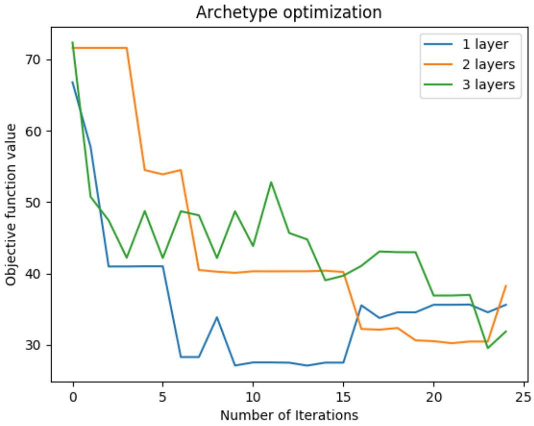

Figure 3: The convergence plot of objective functions corresponding to the 3 variants of style descriptors. As we couldn't run the optimization algorithm for sufficient number iterations, it is difficult to make any conclusion from this graph.

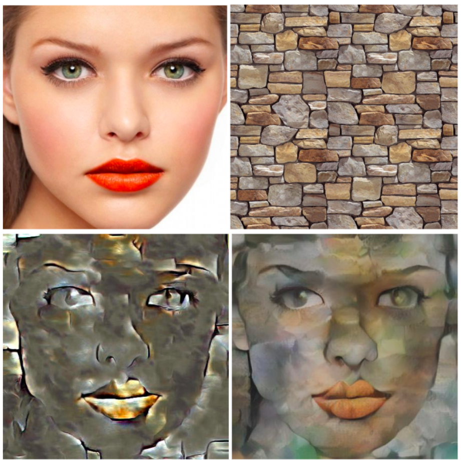

Figure 4: Top left: Content Image; Top right: Style Image; Bottom left: Image obtained by the original Universal Style Transfer model ($\gamma = 0.8$); Bottom right: Image obtained by its new variant ($\gamma = 0.8, \delta = 0.5$) after incorporating the contribution of the original content image in all the feature maps. There is significant difference in the semantic quality of the two variants, where clearly the second model produces the stylized image with much lesser artifacts.

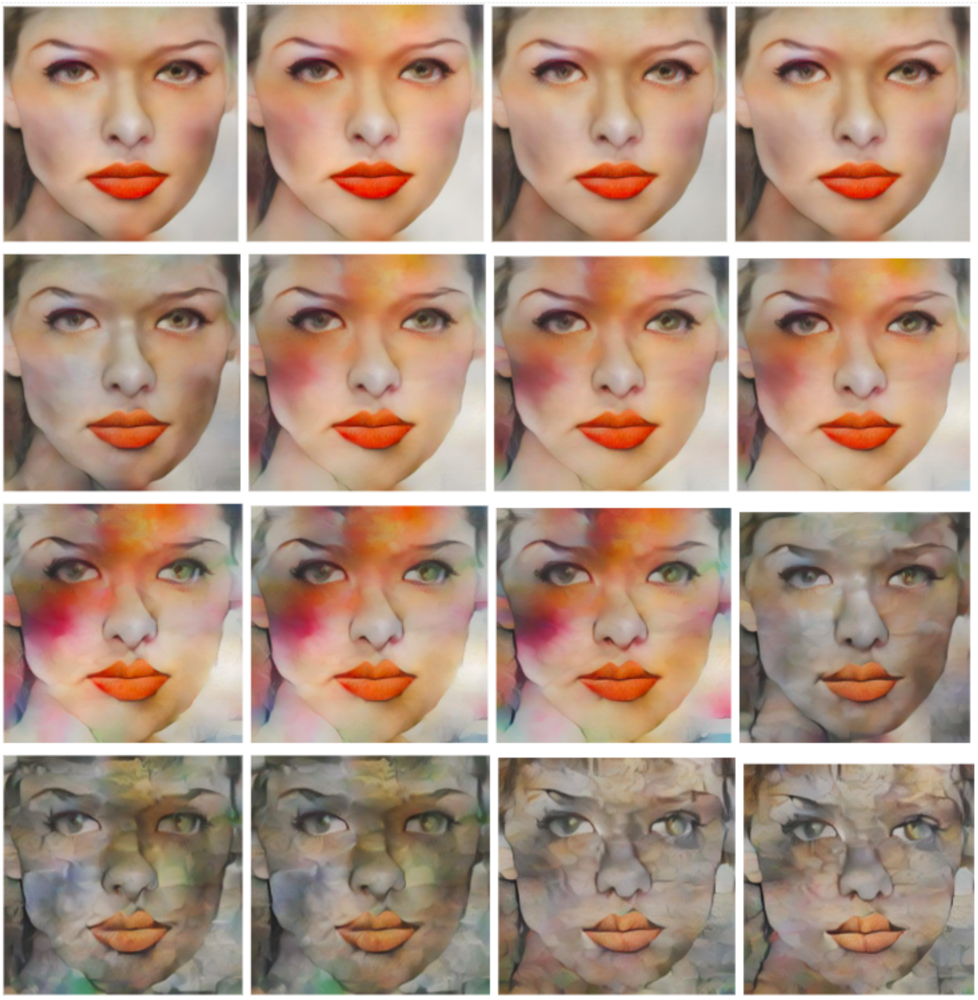

Figure 5: This grid shows the variation of stylization with different values of $\gamma$(taking values 0.2, 0.4, 0.6, 0.8 top to bottom) and $\delta$(taking values 0.2, 0.4, 0.6, 0.8 left to right). We find that as value of $\gamma$ increases (top to bottom), the semantics of the content image become less distinct. As the value of $\delta$ increases (left to right), the features of the original image are better conserved. These images have been generated by the new variant of Universal Style Transfer model. For a given $\gamma$, $\delta$ represents the trade-off between the stylized image and the original content image, and the optimum stylization is observed for $\gamma \approx \delta$.

## 5 Efforts

The major parts were a) computing style descriptor and dimensionality reduction, b) solving optimization problem for archetypes, c) reconstructing image from the archetype, d) implementing new variant of style transfer. The fraction of time spent in each part was 0.2, 0.15, 0.4, 0.25 respectively. The most challenging part was reconstructing image from archetype respresentation. We were not able to figure this and hence we performed limited experiments. All the three members contributed equally to this project.

# References

[1] Yuansi Chen, Julien Mairal, and Zaid Harchaoui. Fast and robust archetypal analysis for representation learning. *arXiv*, 2014. `https://arxiv.org/pdf/1405.6472.pdf`.

[2] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Universal style transfer via feature transforms. *NIPS*, 2017. `https://papers.nips.cc/paper/6642-universal-style-transfer-via-feature-transforms.pdf`.

[3] Daan Wynen, Cordelia Schmid, and Julien Mairal. Unsupervised learning of artistic styles with archetypal style analysis. *NeurIPS*, 2018. `https://papers.nips.cc/paper/7893-unsupervised-learning-of-artistic-styles-with-archetypal-style-analysis.pdf`.