Proxy Learning of Visual Concepts of Fine Art Paintings from Styles through Language Models

Diana Kim[1], **Ahmed Elgammal**[1], **Marian Mazzone**[2]

[1] Department of Computer Science, Rutgers University, U.S.A
[2]Department of Art and Architectural History, College of Charleston, USA
dsk101@rutgers.edu, elgammal@cs.rutgers.edu, marian.mazzone@gmail.com

Abstract

We present a machine learning system that can quantify fine art paintings with a set of visual elements and principles of art. The formal analysis is fundamental for understanding art, but developing such a system is challenging. Paintings have high visual complexities, but it is also difficult to collect enough training data with direct labels. To resolve these practical limitations, we introduce a novel mechanism, called proxy learning, which learns visual concepts in paintings through their general relation to styles. This framework does not require any visual annotation, but only uses style labels and a general relationship between visual concepts and style. In this paper, we propose a novel proxy model and reformulate four pre-existing methods in the context of proxy learning. Through quantitative and qualitative comparison, we evaluate these methods and compare their effectiveness in quantifying the artistic visual concepts, where the general relationship is estimated by language models; GloVe or BERT. The language modeling is a practical and scalable solution requiring no labeling, but it is inevitably imperfect. We demonstrate how the new proxy model is robust to the imperfection, while the other methods are sensitively affected by it.

Introduction

Artists and art historians usually use elements of art, such as line, texture, color, and shape (Fichner-Rathus 2011), and principles of art, such as balance, variety, symmetry, and proportion (Ocvirk et al. 2002) to visually describe artworks. These elements and principles provide structured grounds for effectively communicating about art, especially the first principle of art, which is “visual form” (Van Dyke 1887).

However, in the area of AI, understanding art has mainly focused on a limited version of the principle, through developing systems such as predicting styles (Elgammal et al. 2018; Kim et al. 2018), finding non-semantic features for style (Mao, Cheung, and She 2017), or designing digital filters to extract some visual properties like brush strokes, color, textures, and so on (Berezhnoy, Postma, and van den Herik 2005; Johnson et al. 2008). While they are useful, the concepts do not reveal much about the visual properties of paintings in depth. Kim et al. (2018) suggested a list of 58 concepts that break down the elements and principles of art. We focus on developing an AI system that can quantify such concepts in this paper, and the concepts for art will be referred to as “visual elements”.

The main challenge is that it is not easy to deploy supervised methodology. Art is typically annotated with artist information (name, dates, bio), style, and genre attributes only, but annotating elements of art requires high specialties to identify the visual proprieties of artworks. To resolve the issue, we propose a novel method to learn the visual elements of art through their general relations to styles (period style). While it is difficult to obtain the labels for the visual concepts, there are plenty of available paintings labeled by styles and language resources relating styles to visual concepts, such as online encyclopedia and museum websites. In general, knowing the dominant visual features of a painting enables us to identify its plausible styles. So we have the following questions; (1) what if we can take multiple styles as proxy components to encode visual information of paintings? (2) can a deep Convolutional Neural Network (deep-CNN) help to retrace visual semantics from a proxy representation of multiple styles?

In these previous studies (Elgammal et al. 2018; Kim et al. 2018), existence of the conceptual ties between visual elements and styles is demonstrated by using a hierarchical structure in the deep-CNN. They showed the machine can learn underlying semantic factors of styles from its hidden layers. Inspired by the studies, we hypothetically set a linear relation between visual elements and style. Next, we constrain a deep-CNN by the linear relation to make the machine learn visual concepts from its last hidden layer, while it is trained as a style classifier only.

To explain the methodology, a new concept—proxy learning—is defined first. It refers to all possible learning methods aiming to quantify paintings with a set of finite visual elements, which has no available label, by correlating it to another concept that has abundant labeled data. In this paper, we reformulate four pre-existing methods in the context of proxy learning and introduce a novel approach that utilizes a deep-CNN to learn visual concepts from styles labels and language models. We propose to name it *deep-proxy*.

In the experiment, deep-proxy and four methods in attribute learning—sparse coding (Efron et al. 2004), logistic regression (LGT) (Danaci and Ikizler-Cinbis 2016), Principal Component Analysis method (PCA) (Kim et al.

2018), and an Embarrassingly Simple approach to Zero-Shot Learning (ESZSL) (Romera-Paredes and Torr 2015)—are quantitatively compared with each other. We analyze their effectiveness depending two practical solutions to estimate a general relationship: (1) language models—GloVe (Pennington, Socher, and Manning 2014) and BERT (Vaswani et al. 2017; Devlin et al. 2018)—and (2) sample means of a few ground truth values. The language modeling is a practical and scalable solution requiring no labeling, but it is inevitably imperfect. Finally, we demonstrate how the deep-proxy's cooperative structure learning with styles creates strong resilience to the imperfection from the language models, while PCA and ESZSL are significantly affected by it. On the other hand, as a general relation is estimated by some ground truth samples, PCA performs best in various experiments. Our contributions are as follows.

1. Formulating the proxy learning methodology and applying it to learn visual artistic concepts.
2. A novel and end-to-end method to learn visual elements from fine art paintings without any direct annotation.
3. A new word embedding trained by BERT and a huge art corpus ($\sim 2,400,000$ sentences). This is the first BERT model for art trained by art-related texts.
4. A ground truth set of 58 visual semantics for 120 fine art paintings completed by seven art historians.

Related Work

Attribute Classification For learning semantic attributes, mainstream literature has been based on simple binary classification and fully (Farhadi et al. 2009; Lampert, Nickisch, and Harmeling 2013) or weakly supervision methods (Ferrari and Zisserman 2007; Shankar, Garg, and Cipolla 2015). Support Vector Machine (Farhadi et al. 2009; Lampert, Nickisch, and Harmeling 2013; Patterson et al. 2014) and logistic regression (Farhadi et al. 2009; Danaci and Ikizler-Cinbis 2016) are used to recognize the presence or absence of targeted semantic attributes.

Descriptions by Visual Semantics This paper's method is not designed using a classification problem, but rather it generates real-valued vectors. Each dimension of each vector is aligned with a certain visual concept, so the vectors naturally indicate which paintings are more or less relevant to the concept. As is the case with most similar formats, Parikh et al. (Parikh and Grauman 2011; Ma, Sclaroff, and Ikizler-Cinbis 2012) proposed to predict the relative strength of the presence of attributes through real-valued ranks.

For attribute learning, recently its practical merits have been rather emphasized, such as zero-shot learning (Xian et al. 2018) and semantic (Li et al. 2010) or non-semantic attributes (Huang, Change Loy, and Tang 2016) to boost object recognition. However, in this paper, we focus on attribute learning itself and pursue its descriptive and human understandable advantages, in the same way that Chen et al. (2012) focused on describing clothes with some words understandable to humans.

Incorporating Classes as Learning Attributes Informative dependencies between semantic attributes and objects (classes) are useful; in fact, they have co-appeared in many papers. Lampert et al. (2013) assign attributes to images on a per-class basis and train attribute classifiers in a supervised way. On the other hand, Yu et al. (2014) model attributes based on their generalized properties—such as their proportions and relative strength—with a set of categories and make learning algorithms satisfy them as necessary conditions. The methods do not require any instance-level attributes for training like this paper method, but learning visual elements satisfying the constraints of relative proportions among classes is not related to our goal or methodology. Some researchers (Mahajan, Sellamanickam, and Nair 2011; Wang and Ji 2013) propose joint learning frameworks to more actively incorporate class information into attribute learning. In particular, Akata et al. (2013) and Romera-Paredes et al. (2015) hierarchically treat attributes as intermediate features, which serve to describe classes. The systems are designed to learn attributes by bi-directional influences from class to attributes (top-down) and from image features to attributes (bottom-up) like deep-proxy. However, their single and linear layering, from image features to their intermediate attributes, are different from the multiple and non-liner layering in deep-proxy.

Learning Visual Concepts from Styles Elgammal et al. (Elgammal et al. 2018; Kim et al. 2018) show that a deep-CNN can learn semantic factors of styles from its last hidden layers by using a hierarchical structure of deep-CNN. They interpret deep-CNN's last hidden layer with predefined visual concepts through multiple and separated post-procedures, but deep-proxy simultaneously learns visual elements while machines are trained for style classification. In the experiment, the method proposed by Kim et al. (2018) is compared with deep-proxy as the name of PCA.

Methodology

Linear Relation

Two Conceptual Spaces Styles are seldom completely uniform and cohesive, and often carry forward within them former styles and other influences that are still operating within the work. As explained in *The Concept of Style* (Lang 1987), a style can be both a possibility and an interpretation. It is not a definite quality that inherently belongs to objects, although each of the training samples are artificially labeled with a unique style. Due to the complex variations of the visual properties of art pieces in sequential arrangements of times, styles can be overlapped, blended, and merged.

Based on the idea, this research begins with representing paintings with the two entities: a set of m visual elements and a set of n styles. Two conceptual vector spaces $\mathbf{S}$ and $\mathbf{A}$ for style and visual elements are introduced, whose each dimension is aligned with their semantic. Two vector functions, $f_A(\cdot)$ and $f_S(\cdot)$, are defined to transform input image x into the conceptual spaces in equation (1) below.

$$\begin{aligned} f_A(x) &: x \rightarrow \vec{a}(x) = [a_1(x), a_2(x), ..., a_m(x)]^t \in \mathbf{A} \\ f_S(x) &: x \rightarrow \vec{s}(x) = [s_1(x), s_2(x), ..., s_n(x)]^t \in \mathbf{S} \end{aligned} \quad (1)$$

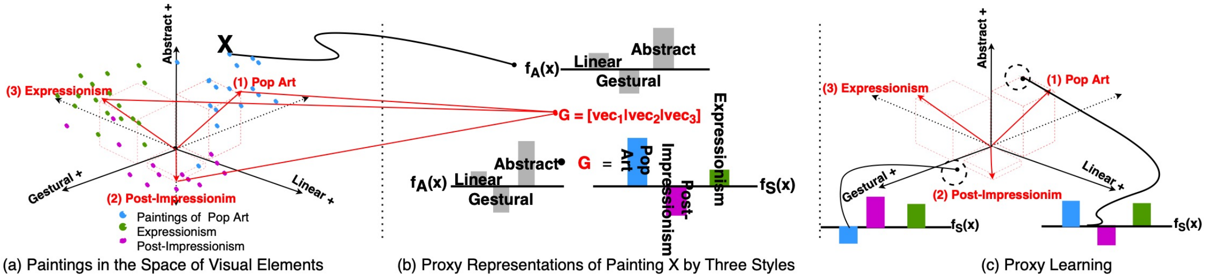


Figure 1: Summary of Proxy Learning: (a) The paintings of three styles (Pop art, Post-impressionism, and Expressionism) are scattered in the space of three visual elements (abstract, gestural, and linear). The red vectors represent typical vectors of the three styles. (b) A painting X, originally positioned in the visual space, can be transformed to the three-style (proxy) representation by computing inner products with each of the typical vectors. (c) Proxy learning aims to estimate or learn its original visual coordinates from a proxy representation and a set of typical vectors.

Category-Attribute Matrix Inspired by a prototype theory (Murphy 2004) in cognitive science, we posit that a set of pre-defined visual elements of art is sufficient for characterizing styles. According to this theory, once a set of attributes are arranged to construct a vector space, a vector point can summarize each of categories. Mathematically modeling the principles, $x_{s_i^*}$ is set to be the typical (best) example for the style s_i, where $i \in \{1, 2, ...n\}$ and n is the number of styles. This is represented by $f_A(x^*_{s_i}) = x \rightarrow \vec{a}(x^*_{s_i})$. By accumulating the $\vec{a}(x^*_{s_i}) \in \mathbb{R}^m$ as columns for all different n styles, a matrix $G \in \mathbb{R}^{m \times n}$ is formed. Matrix G becomes a category-attribute matrix.

Linear System Matrix G ideally defines n typical points for n styles in the attribute space of $\mathbf{A}$. However, as aforementioned, images that belong to a specific style show intra-class variations. In this sense, for a painting x that belongs to style s_i, the $f_A(x^*_{s_i})$ is likely to be the closest to the $f_A(x)$, and its similarities to other styles' typical points can be calculated by the inner products between $f_A(x)$ and $f_A(x^*_{s_i})$ for all n styles, $i \in \{1, 2, ...n\}$. All computations are expressed by $f_A(x)^t \cdot G$ and its output $f_S(x)^t$. This results in the linear equation (2) below.

$$f_A(x)^t \cdot G = f_S(x)^t \tag{2}$$

Definition of Proxy Learning

In equation (2), knowing $f_S(\cdot)$ becomes linearly tied with knowing $f_A(\cdot)$, so we have the following questions: (1) given G and $f_S(\cdot)$, how can we learn the function $f_A(\cdot)$? (2) before doing that, how can we properly encode G and $f_S(\cdot)$ first? This paper aims to answer these questions. We first re-define them by a new concept of learning, named *proxy learning*. Fig. 1 is an illustrative example to describe it.

Proxy learning: a computational framework that learns the function $f_A(\cdot)$ from $f_S(\cdot)$ through a linear relationship G. G is estimated by language models or human survey.

Language Modeling

The G matrix is estimated by using distributed word embeddings in NLP. Two embeddings were considered: GloVe (Pennington, Socher, and Manning 2014) and BERT (Devlin et al. 2018; Vaswani et al. 2017). However, their original dictionaries do not provide all the necessary art terms. Especially for BERT, it holds a relatively smaller dictionary than GloVe. In the original BERT, vocabulary words are represented by several word-pieces (Wu et al. 2016), so it is unnecessary to hold a large set of words. However, the token-level vocabulary words could lose their original meanings, so a new BERT model had to be trained from scratch on a suitable art corpus in order to compensate for the deficient dictionaries.

A Large Corpus of Art To prepare a large corpus of art, we first gathered all the descendent categories (about $6,500$) linked with the parent categories of ``Painting'' and ``Art Movement'' in Wikipedia and scrawled all the texts under the categories by using a library available in public. Some art terms and their definitions presented in TATE museum (http://tate.org.uk/art/art-terms) were also added, so finally, with $\sim 2,400,000$ sentences, a set word embedding set —BERT— is newly trained for art.

Training BERT For a new BERT model for art, the BERT-BASE model (12-layer, 768-hidden, 12-heads, and not using cased letters) was selected and trained from scratch with the collected art corpus. For training, the original vocabulary set is updated by adding some words which are missed in the original framework. We averaged all 12 (layers) embeddings to compute each of final word embeddings.

Estimation of Category-Attribute Matrix G To estimate a matrix G, vector representations were collected and the following over-determined system of equations was set. Let the $W_A \in \mathbb{R}^{d \times m}$ denote a matrix of which each column implies a d-dimensional word embedding to encode one of m visual elements, and the $w_{s_i} \in \mathbb{R}^d$ be a word embedding that represents style s_i among n styles.

$$W_A \cdot \vec{a}(x^*_{s_i}) = w_{s_i} \tag{3}$$

By solving the equation (3) for $i \in \{1, 2, ..., n\}$, the vector $\vec{a}(x^*_{s_i}) \in \mathbb{R}^m$ was estimated, which becomes each column vector of G. It quantifies how the visual elements are

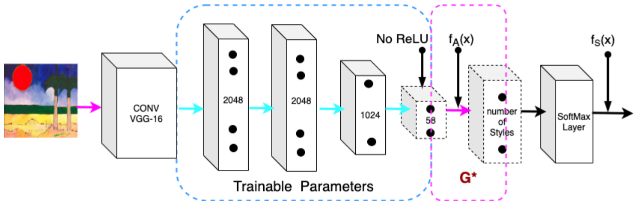


Figure 2: Deep-Proxy Architecture.

positively or negatively related to a certain style in a distributed vector space. In general, word embedding geometrically captures semantic or syntactical relations between words, so this paper postulates that the general relationship among the concepts can be reflected by the linear formulation (3).

Deep-Proxy

We propose a novel method to jointly learn the two multimodal functions, $f_S(x)$ and $f_A(x)$, through a pre-estimated general matrix (G). Its principal mechanism is a category-attribute matrix (G) is hard-coded into the last fully connected (FC) parameters of a deep-CNN, so it is enforced to learn visual elements ($f_A(x)$) indirectly from its last hidden layers, while it is outwardly trained to learn multiple styles ($f_S(x)$). We propose to name this framework *deep-proxy*. In this paper, the original VGG-16 (Simonyan and Zisserman 2015) is adapted for its popularity and modified as a style classifier, as shown in Fig. 2.

Implementation of Deep-Proxy All convolution layers are transferred from the ImageNet as is and frozen, but the original FC layers, $(4096-4096-1000)$, are expanded to the five layers $(2048-2048-1024-58-G^*(58\times n)-$ n number of styles$)$. These FC parameters (cyan colored and dashed box) are updated during training. We also tried to fine-tune convolution parts, but they showed slightly degraded performance compared to the FC-only training. Therefore, all the results presented in the later sections are FC-only trained for $200,000$ steps at 32 batch size by the momentum optimizer (momentum $= 0.9$). The learning rate is initially set as $1.0e-3$ and degraded at the factor of 0.94 every 2 epochs. The final soft-max output is designed to encode the $f_S(x)$, and the last hidden layer (58-D) is set to encode the $f_A(x)$. The two layers are interconnected by the FC block G^* (magenta colored and dashed box) to impose a linear constraint between the two modalities. For the $f_A(x)$, the hidden layer's Rectified Linear Units (ReLU) is removed, so it can have both positive and negative values.

Objective Function of Deep-Proxy Let $I(q,k)$ be an indicator function stating whether or not the k-th sample belongs to style class q. Let $s_q(x|w)$ be the q-th style component of the soft-max simulating $f_S(x)$. Let $f_A(x|w)$ be the last hidden activation vector, where x is an input image and ω is the network's parameters. Then, an objective for multiple style classification is set as in equation (4) below. The λ is added to regularize the magnitudes of the last hidden layer.

$$\min_{\omega}\sum_{k}^{K}\sum_{q}^{Q}-I(q,k)\ \cdot\ \log_e(s_q(x|\omega))\ +\ \lambda\ \cdot\ \|f_A(x|\omega)\|_1 \tag{4}$$

In the next subsections, three versions of deep-proxy are defined depending on how G^* matrix is formulated.

(1) Plain Method $(G^*=G)$ A G matrix is estimated and plugged into the network as it is. Two practical solutions are considered to estimate G, language models and sample means of a few ground truth values. In training for Plain, the G^* is fixed as the G matrix, while the other FC layers are updated. This modeling is exactly aligned with equation (2).

(2) SVD Method A structural disadvantage of Plain method is noted and resolved by using Singular Vector Decomposition (SVD).

It is natural that the columns of a G matrix are correlated because a considerable number of visual properties are shared among the typical styles. Thus, if the machine learns visual space properly, the multiplication of $f_A(x)^t$ with G necessarily produces $f_S(x)^t$, highly valued on multiple styles. On the other hand, deep-proxy is trained by one-hot vectors promoting orthogonality among style and a sharp high value on a specific style component. Hence, learning with one-hot vectors can cause interference on learning visual semantics if we simply adopt the plain method above.

For example, suppose there is a typical Expressionism painting x_*. Then, it is likely to be highly valued both in terms of Expressionism and Abstract-Expressionism under equation (2) because the two styles are correlated visually. But if one hot-vector encourages the machine to value $f_S(x_*)$ highly on the Expressionism axis only, then the machine might not be able to learn visual concepts well, such as gestural brush-strokes or mark-making, and the impression of spontaneity, for those concepts are supposed to be high on both styles. To fix this discordant structure, the G and $f_A(x)$ are transformed to the space where typical style representations are orthogonal to one another. It reformulates equation (2) by equation (5), where T is a transform matrix to the space.

$$[f_A(x)^t\cdot T^t]\cdot[T\cdot G]=f_S(x)^t \tag{5}$$

To compute the transform matrix T, G is decomposed by SVD. As the number of attributes (m) is greater than the number of classes (n) and its rank is n, the G is decomposed by $U\cdot\Sigma\cdot V^t$, where $U(m\times n)$ and $V(n\times n)$ are the matrices whose columns are orthogonal and the Σ $(n\times n)$ is a diagonal matrix. From the decomposition, $V^t=\Sigma^{-1}\cdot U^t\cdot G$, we can use $\Sigma^{-1}{\cdot}U^t$ as the transform matrix T. The $T=\Sigma^{-1}{\cdot}U^t$ transforms each column of G to each orthogonal column of V^t. In this deep-proxy SVD method, the G^* is reformulated by these SVD components as presented in equation (6) below.

$$G^*=T^t\cdot T\cdot G=U\cdot\Sigma^{-2}\cdot U^t\cdot G \tag{6}$$

(3) Offset Method A positive offset vector $\vec{\sigma}\in R^{+m}$ is introduced to learn a threshold to determine a visual concept as relevant or not. Each component of $\vec{\sigma}$ implies an individual threshold for each element, so when it is subtracted from a column of a G matrix, we can take zero as an absolute threshold to interpret whether or not visual concepts are relevant to a style class. Since G is often encoded by the values between zero and one, especially when it is created by human survey (ground truth values), we need a proper offset to shift G matrix. Hence, the vector $\vec{\sigma} \in R^{+m}$ is set as learnable parameters in the third version of deep-proxy. It sets the G^* as $U \cdot \Sigma^{-2} \cdot U^t \cdot (G - \boldsymbol{\mu})$, where $\boldsymbol{\mu} = [\vec{\sigma}|\vec{\sigma}|...|\vec{\sigma}]$ is the tiling matrix of the vector $\vec{\sigma}$. In Offset method, the SVD components U and Σ are newly calculated for the new $(G - \boldsymbol{\mu})$ at every batch in training.

Experiments

In this section, we quantitatively evaluate five proxy methods and compare their effectiveness in quantifying artistic visual concepts.

Four Proxy Methods

Four pre-existing methods—sparse coding (Efron et al. 2004), logistic regression (LGT) (Danaci and Ikizler-Cinbis 2016), Principal Component Analysis (PCA) method (Kim et al. 2018), and an Embarrassingly Simple approach to Zero-Shot Learning (ESZSL) (Romera-Paredes and Torr 2015)—are reformulated in the context of proxy learning and quantitatively compared with each other. We demonstrate how LGT and deep-proxy based on deep-learning are more robust than others when the general relationship (G matrix) is estimated by language models; GloVe (Pennington, Socher, and Manning 2014) or BERT (Devlin et al. 2018; Vaswani et al. 2017). We also show LGT performance is degraded sensitively when G matrix is too sparse.

Logistic Regression (LGT) Each column of G was used to assign attributes to images on a per class basis. When G matrix is not a probabilistic representation, without shifting zero points, the positives were put into the range of 0.5 to 1.0 and the negatives were put into the range of 0.0 to 0.5.

PCA The last hidden feature of a deep-CNN style classifier is encoded by styles and then multiplied with the transpose of G matrix to finally compute each degree of the visual elements.

ESZSL This can be regarded as a special case of the deep-proxy Plain by setting a single FC layer between visual features and $f_A(x)$, replacing the softmax loss with Frobenius norm $\|\cdot\|^2_{Fro}$, and encoding styles with $\{-1, +1\}$. To compute the single layer, a global optimum is found through a closed-form formula proposed by Romera-Paredes et al. (2015).

Sparse Coding It estimates $f_A(\cdot)$ directly from the style encodings $f_S(\cdot)$ and G matrix by solving a sparse coding equation without seeing input images. Its better performance versus random cases proves our hypothetical modeling assuming informative ties between style and visual elements.

WikiArt Data Set and Visual Elements

This paper used the $76,921$ paintings in WikiArt's data set (WikiArt 2010) and merged their original 27 styles into 20 styles, the same as those presented by Elgammal et al. (2018). 120 paintings were separated for evaluation, and the remaining samples were randomly split into 85% for training and 15% for validation. This paper adopts the pre-selected visual concepts proposed by Kim et al. (2018). In the paper, 59 visual words are suggested, but we used 58 words, excluding the "medium" because it is not descriptive.

Evaluation Methods

Human Survey A binary ground truth set was completed by seven art historians. The subjects were asked to choose between one of the following three choices: (1) yes, the shown attribute and painting are related. (2) they are somewhat relevant. (3) no, they are not related. Six paintings were randomly selected from each of 20 styles, and art historians made three sets of ground truths of 58 visual elements for the 120 paintings first. From the three sets, a set was determined based on the majority vote. For example, if a question is marked by three different answers, the (2) 'somewhat relevant' is determined as the final answer. The results show $1,652$ (24%) as relevant, 782 (11%) as somewhat, and $4,526$ (65%) as irrelevant. In order to balance positive and negative values, this paper considered the somewhat answers as relevant and created a binary ground truth set. The 120 paintings will be called "eval" throughout this paper.

AUC Metric The **A**rea **U**nder the receiver operating characteristic **C**urve (AUC) was used for evaluation. When we say AUC@K, it means an averaged AUC score, where the K denotes the number of attributes to be averaged. A random case is simulated and drawn in every plot as a comparable baseline. Images are sorted randomly without the consideration of the machine's out values and then AUCs are computed.

Plots To draw a plot, we measured 58 AUC scores for all 58 visual elements. The scores were sorted in descending order, every three scores were grouped, and 19 ($\lfloor 58/3 \rfloor$) points of AUC@3 were computed. Since many of the visual concepts were not learnable (AUC ≤ 0.5), a single averaged AUC@58 value did not differentiate performance clearly. Hence, the descending scores were used, but averaged at every three points for simplicity. Regularization parameters were written in the legend boxes of plots if necessary.

SUN and CUB SUN (Patterson et al. 2014) and CUB (Wah et al. 2011) are adopted to understand the models in general situations. All experiments are based on the standard splits, proposed by Xian et al. (2018). For evaluation, mean Average Precision (AP) is used because the ground truth of the data sets is imbalanced by very large negative samples (the mean of all the samples is 0.065 for SUN and 0.1 for CUB at the binary threshold of 0.5). For G matrix, their ground truth samples are averaged.

Estimation of Category-Attribute Matrix

Two ways to estimate G matrix are considered. First, from the two sets of word embeddings—GloVe and BERT—two G matrices are computed by equation (3). This paper will refer to the BERT matrix as G_B and to the GloVe matrix as G_G. The G_G is used only for 12-style experiments in a

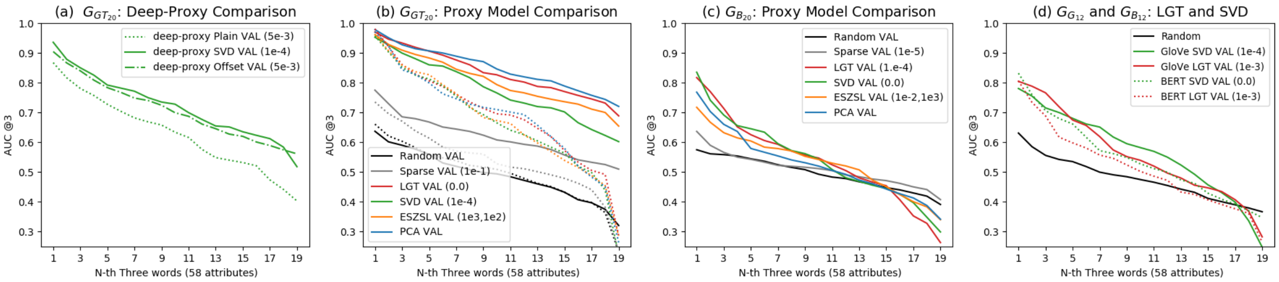


Figure 3: (a) Three deep-proxy models by $G_{GT_{20}}$ are compared on "eval". SVD is selected as the best model for art. (b) Proxy models by $G_{GT_{20}}$ are compared. The solid-lines are evaluated by "eval-G", and the dotted-lines are evaluated by "eval-NG". (c) Five proxy models by $G_{B_{20}}$ are evaluated by "eval". (d) SVD and LGT by $G_{B_{12}}$ and $G_{G_{12}}$ are compared by "eval-VAL".

section later because the vocabulary of GloVe does not contain the all terms for the 20-style. As necessary, they will be written with the number of styles involved in experiments like $G_{B_{20}}$ or $G_{B_{12}}$. Second, 58-D ground truths of the three paintings, randomly selected among six paintings of each style, were averaged and accumulated into columns, and the ground truth matrix G_{GT} was also established. To do so, we first mapped the three answers of the survey with integers: "relevant" = $+1$; "somewhat" = 0; and "irrelevant" = -1. The 60 paintings of "eval" used to compute G_{GT} will be called "eval-G" and the others will be called "eval-NG".

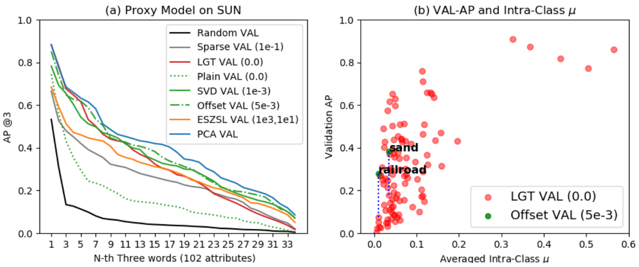


Figure 4: SUN experiment: (a) Validation results for all models are shown. (b) The relationship between validation-AP (y-axis) and intra-class μ (x-axis) for LGT is presented for each attribute (each red dot). Points of Offset (two green dots) are drawn only when the AP-gap between Offset and LGT is more than 0.2. The higher scores on the green dots show that Offset is less affected by the sparsity of G matrix than LGT. As the AP-gap gets lower to 0.15, 19 words were found, and Offset worked better than LGT for all the 19 words.

Model Selection for Deep-Proxy

To select the best deep-proxy for art, the three versions of Plain, SVD, Offset by $G_{GT_{20}}$ are compared. For Offset, the $G_{GT_{20}}$ is pre-shifted by $+1.0$ to make all components of $G_{GT_{20}}$ matrix positive, and let machines learn a new offset from it. For the regularization λ in equation (4), $1\text{e}{-4}$, $5\text{e}{-4}$, $1\text{e}{-3}$, and $5\text{e}{-3}$ are tested. In Fig. 3 (a), SVD achieved the best rates and outperformed the Plain model. Offset was not as effective as SVD. Since $G_{GT_{20}}$ was computed from the ground truth values, its zero point was originally aligned with "somewhat", so offset learning may not be needed.

For a comparable result with SUN data, and Offset is shown as the best in Fig. 4 (a). SUN's G matrix is computed by "binary" ground truths, so it is impossible to gauge the right offset. Hence, Offset learning becomes advantageous for SUN. However, for CUB, SVD and Offset were not learnable (converged to a local minimum, whose recognition is the random choice of equal probabilities). Since CUB's G matrix has smaller eigenvalues than other data sets, implying subtle visual difference among bird classes, the two deep-proxy methods happen to be infeasible by demanding a neural net to capture fine and local visual differences of birds first, in order to discern the birds as orthogonal vectors. However, for the neural net especially in the initial stage of learning, finding the right direction to the high goal is far more challenging compared to the case of art and SUN, whose attributes can be found rather distinctively and globally in different class images.

Analysis of Proxy Models

Proxy models by $G_{GT_{20}}$ and $G_{B_{20}}$ are evaluated in Fig. 3 (b) and (c). To avoid the bias by the samples used in computing G matrix, for the models by $G_{GT_{20}}$, validation (solid-line) and test (dotted-line) are separately computed based on "eval-G" and "eval-NG " each.

Sensitive Methods to Language Models High sensitivity to $G_{B_{20}}$ is observed for PCA and ESZSL. In Fig. 3 (b), PCA and LGT show similar performance on "eval-NG", but on "eval-G", PCA performs better than LGT. The same phenomenon is observed between ESZSL and SVD again. The better performance on "eval-G" indicates somewhat direct replication of G matrix into outcomes. This hints that ESZSL and PCA can suffer more degradation than other models if G matrix is estimated by language models, so its imperfection straightly act on their results, as shown in Fig. 3 (c). Since they compute visual elements through direct multiplications between processed features and G matrix, and particularly for ESZSL, it finds a global optimum given a G matrix, they showed the highest sensitivity to the condition of G matrix.

Robust Methods to Language Models Deep-learning makes LGT and deep-proxy slowly adapt to the information given by G matrix, so the models are less affected by language models than ESZSL and PCA, as shown in Fig. 3

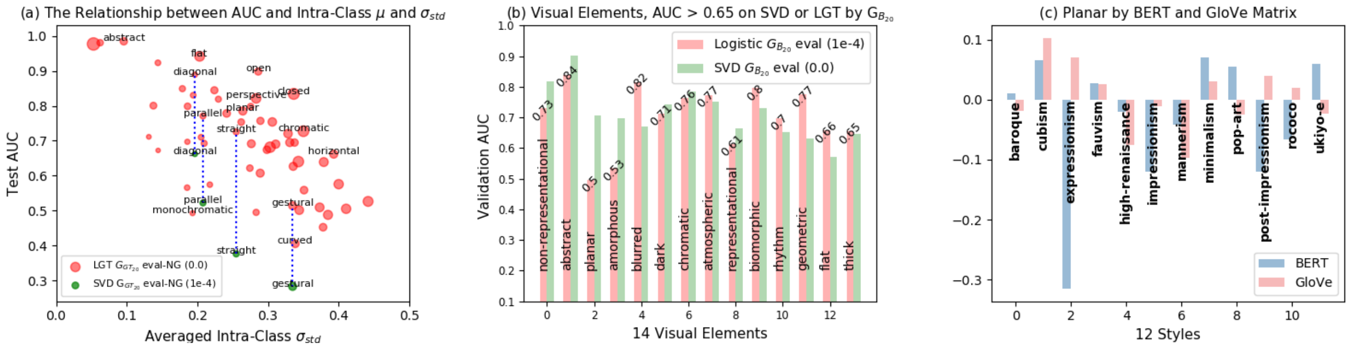


Figure 5: (a) The relationship between test-AUC (y-axis) and σ_{std} (x-axis) and μ (size of dots) for LGT (by $G_{GT_{20}}$ and "eval-NG") is shown for each attribute (each red dot). Offset points (four green dots) are drawn only when the AUC-gap between the SVD and LGT is more than 0.2. Each performance gap is traced by the blue dotted-lines. (b) Visual elements scored more than 0.65 AUC by SVD or LGT (by $G_{B_{20}}$ and "eval") are presented. (c) Style Encoding of BERT and GloVe for the word "planar".

(c). LGT can learn some visual elements through BERT or GloVe, even when not all style relations for the elements are correct in the models. For example, for G_B, 'expressionism' is encoded as more related with "abstract" than 'cubism' or 'abstract-expressionism', which is false. But despite the partial incorrectness, LGT gradually learns the semantic of "abstract" at the rate of 0.84 AUC using the training data in a larger range of styles, correctly encoded; northern-renaissance (least related to "abstract") < rococo < cubism < abstract-expressionism (most related to "abstract") etc (abstract AUCs of SVD, PCA, and ESZSL by $G_{B_{20}}$: $0.9, 0.8, 0.7$).

Deep-proxy more actively adjusts some portion of G matrix. Suppose there is a neural net trained with $G' = (G+\Delta G)$ distorted by ΔG. By equation (2), this $f'_A(x)^t \cdot (G+\Delta G) = f^t_S(x)$ is a valid convergent point of the net, and we also can see this $(f'_A(x)^t + b^t(x)) \cdot G = f_S(x)^t$ as another possible point, where $b(x)^t \cdot G = f'_A(x)^t \cdot \Delta G$. If the bottom of the neural net approximately learns $f'_A(x)^t + b^t(x)$, it would work as if a better G is given, absorbing some errors. This adjustment could explain the robustness to the imperfection of language models than others, and also the flexibility to the sparse G matrix that is shown to be directly harmful for LGT. This will be discussed in the next section.

Logistic and Deep-Proxy on G_{GT} Two factors are analyzed with LGT and SVD performance: intra-class's standard deviation (σ_{std}) and mean (μ). The intra-statistics of each style are computed with "eval" and averaged across the styles to estimate σ_{std} and μ for 58 visual elements. For LGT and SVD by $G_{GT_{20}}$, AUC is moderately related with σ_{std} (Pearson correlation coefficient $r_{LGT} = -0.65$ and $r_{SVD} = -0.51$), but their performance is not explained solely by σ_{std}. As shown in Fig. 5 (a), "monochromatic" (AUC of LGT and SVD: 0.49 and 0.58) scored far less than "flat" (AUC of LGT and SVD : 0.94 and 0.92) even if both words' σ_{std} are similar and small. Since the element of monochromatic was not a relevant feature for most of styles, it was consistently encoded as very small values across the styles in G_{GT} matrix. The element has small variance within a style, but does not have enough power to discriminate styles so failed to learn. LGT can be more degraded with the sparsity because the information encoded that is close to zero for all styles cannot be back-propagated properly. As shown in Fig. 4 (b), intra-class μ of 102 attributes in SUN are densely distributed between 0.0 and 0.1, so LGT is lower ranked compared to art. LGT AP is most tied in the sparse μ to others ($r_{LGT} = 0.43$, $r_{Offset} = 0.36$, $r_{PCA} = 0.33$, $r_{ESZSL} = -0.15$ at $\mu < 0.3$).

For SVD by $G_{GT_{20}}$, its overall performance is lower than LGT by $G_{GT_{20}}$. When the words "diagonal", "parallel", "straight", and "gestural" (four green dots in Fig. 5 (a)) were found by the condition of $|AUC_{(SVD)} - AUC_{(LGT)}| > 0.2$, LGT scored higher than SVD for all the words. Since SVD is trained by an objective function for multiple style classification, the learning visual elements can be restricted by the following cases. Some hidden axes could be used to learn non-semantic features to promote learning styles. Or, some necessary semantics for styles could be strewn throughout multiple axes. Hence, LGT generally learns more words than SVD when G matrix is estimated by some ground truths as shown in Fig. 3 (b), but G matrix should not be too sparse for LGT.

Logistic and Deep-Proxy on BERT and GloVe For language models, it is a bit hard to generalize the performance of LGT and SVD. As shown in Fig. 5 (b), it was not clear which is better with BERT. We needed another comparable language model to understand their performance. GloVe is tested after dividing 20 styles into train (12 styles) [1] and test (8 styles). Aligned with the split, the "eval" was also separated into "eval-VAL" and "eval-TEST " (8 unseen styles in training). Here, the "eval-VAL" was used to select hyper parameters. On the same split, the models by BERT $G_{B_{12}}$ were compared, too. Depending on each language model, the ranking relations were differently shown. In Fig. 3 (d), SVD by $G_{B_{12}}$ scored better than LGT at all AUC@3 points. However, LGT by $G_{G_{12}}$ was better than SVD for the first top 15 words, but for second 15 words, SVD scored better than LGT. To figure out a key factor of the different performance,

[1] Baroque, Cubism, Expressionism, Fauvism, High-Renaissance, Impressionism, Mannerism, Minimalism, Pop-Art, Ukiyo-e

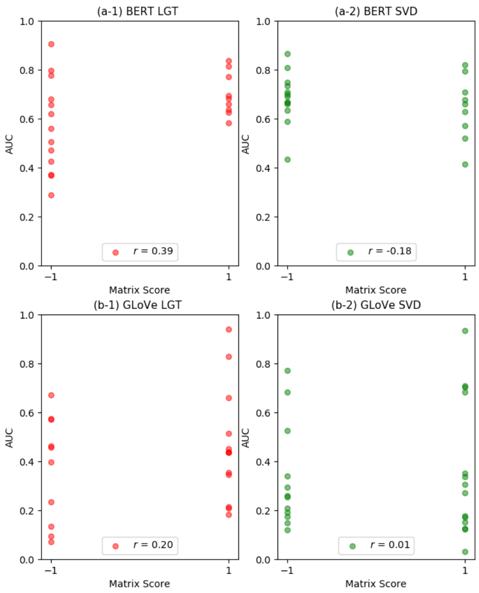


Figure 6: Correlation analysis between AUCs and matrix scores. The BERT plots (a-1) and (a-2) are drawn based on the 22 visual elements which scored more than AUC 0.6 by any of SVD or LGT (by $G_{B_{12}}$). The GloVe plots (b-1) and (b-2) are drawn based on the 28 visual elements which scored more than AUC 0.6 by any of SVD or LGT (by $G_{G_{12}}$). This shows LGT is more sensitively affected by the quality of language models.

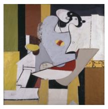
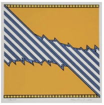
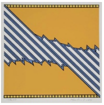
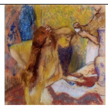
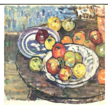
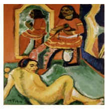
abstract	chromatic	planar	representational	perspective
				
				
				
				
0.90	0.79	0.71	0.67	0.46

Table 1: Descending ranking results (top to bottom) based on the prediction $f_A(x)$ of SVD ($G_{B_{20}}$ and $\lambda = 0.0$). The three most ($1-3$ rows) and three least ($4-6$ rows) relevant paintings are shown as the machine predicted. The last row indicates the AUC score of each visual element.

we scored the quality of BERT and GloVe with $\{-1, +1\}$ for each visual element and conducted correlation analysis between the scores and the AUC results. Pearson correlation coefficient r between AUCs and the scores are computed. The results are shown in Fig. 6.

In the analysis, GloVe scored higher than BERT, and LGT showed the stronger correlation than SVD between AUCs and scores. This proves the robustness of SVD to the imperfection of language models along with the results of Fig 3 (d). As a specific example, the word "planar" is incorrectly encoded by BERT, quantifying some negatives on expressionism, impressionism, and post-impressionism as shown by Fig. 5 (c). The wrong information influenced more on LGT, so its AUC scored 0.38 (eval-TEST: 0.47) by BERT but 0.77 (eval-TEST: 0.76) by GloVe, while SVD learned "planar" by the similar rates of 0.73 (eval-TEST: 0.58) by BERT and 0.78 (eval-TEST: 0.68) by GloVe on "eval-VAL". For LGT, the defective information is directly provided through training data, so it is more sensitively affected by noisy language models. However, SVD can learn some elements even when it is trained by a G matrix that is not perfect if the elements are essential for style classification possibly through the adjustment operation, as aforementioned.

Descending Ranking Results of SVD by $G_{B_{20}}$ To present some example results, 120 paintings of "eval" are sorted based on the activation values $f_A(x)$ of SVD by $G_{B_{20}}$. Table 1 presents some results of words that achieved more than 0.65 or less than 0.65 with BERT model. This shows how the "eval" paintings are visually different according to each output-value of SVD by G_B for the selected five visual elements (abstract, chromatic, planar, representational, and perspective).

Conclusion and Future Work

Quantifying fine art paintings based on visual elements is a fundamental part of developing AI systems for art, but their direct annotations are very scarce. In this paper, we presented several proxy methods to learn the valuable information through its general and linear relations to style, which can be estimated by language models or human survey. They are quantitatively analyzed to reveal how the inherent structures of the methods make them robust or weak on the practical estimation scenarios. The robustness of deep-proxy to the imperfection of language models is a key finding. For future work, we will look at more complex systems. For example, a non-linear relation block learned by language models could be transferred or transplanted to a neural network to learn visual elements through the deeper relation with styles. Furthermore, direct applications for finding acoustic semantics for music genres or learning principle elements for fashion designs would be interesting subjects for proxy learning. Their attributes are visually or acoustically shared to define higher level of categories, but their class boundaries could be softened as proxy representations.

References

Akata, Z.; Perronnin, F.; Harchaoui, Z.; and Schmid, C. 2013. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 819–826.

Berezhnoy, I. E.; Postma, E. O.; and van den Herik, J. 2005. Computerized visual analysis of paintings. In *Int. Conf. Association for History and Computing*, 28–32.

Chen, H.; Gallagher, A.; and Girod, B. 2012. Describing clothing by semantic attributes. In *European Conference on Computer Vision (ECCV)*, 609–623. Springer.

Danaci, E. G.; and Ikizler-Cinbis, N. 2016. Low-level features for visual attribute recognition: An evaluation. *Pattern Recognition Letters*, 84: 185–191.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R.; et al. 2004. Least angle regression. *The Annals of statistics*, 32(2): 407–499.

Elgammal, A.; Liu, B.; Kim, D.; Elhoseiny, M.; and Mazzone, M. 2018. The shape of art history in the eyes of the machine. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2183–2191. AAAI press.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1778–1785. IEEE.

Ferrari, V.; and Zisserman, A. 2007. Learning visual attributes. *Advances in Neural Information Processing Systems (NIPS 2007)*, 20: 433–440.

Fichner-Rathus, L. 2011. *Foundations of art and design: An enhanced media edition*. Cengage Learning.

Huang, C.; Change Loy, C.; and Tang, X. 2016. Unsupervised learning of discriminative attributes and visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5175–5184.

Johnson, C. R.; Hendriks, E.; Berezhnoy, I. J.; Brevdo, E.; Hughes, S. M.; Daubechies, I.; Li, J.; Postma, E.; and Wang, J. Z. 2008. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4): 37–48.

Kim, D.; Liu, B.; Elgammal, A.; and Mazzone, M. 2018. Finding Principal Semantics of Style in Art. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, 156–163.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3): 453–465.

Lang, B. 1987. *The concept of style*. Cornell University Press.

Li, L.-J.; Su, H.; Lim, Y.; and Fei-Fei, L. 2010. Objects as attributes for scene classification. In *European Conference on Computer Vision (ECCV)*, 57–69. Springer.

Ma, S.; Sclaroff, S.; and Ikizler-Cinbis, N. 2012. Unsupervised learning of discriminative relative visual attributes. In *European Conference on Computer Vision (ECCV)*, 61–70. Springer.

Mahajan, D.; Sellamanickam, S.; and Nair, V. 2011. A joint learning framework for attribute models and object descriptions. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, 1227–1234. IEEE.

Mao, H.; Cheung, M.; and She, J. 2017. Deepart: Learning joint representations of visual arts. In *Proceedings of the 25th ACM international conference on Multimedia (ACMMM)*, 1183–1191. ACM.

Murphy, G. 2004. *The big book of concepts*. MIT press.

Ocvirk, O. G.; Stinson, R. E.; Wigg, P. R.; Bone, R. O.; and Cayton, D. L. 2002. *Art fundamentals: Theory and practice*. McGraw-Hill.

Parikh, D.; and Grauman, K. 2011. Relative attributes. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, 503–510. IEEE.

Patterson, G.; Xu, C.; Su, H.; and Hays, J. 2014. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2): 59–81.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Romera-Paredes, B.; and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*, 2152–2161.

Shankar, S.; Garg, V. K.; and Cipolla, R. 2015. Deep-carving: Discovering visual attributes by carving deep neural nets. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 3403–3412.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*.

Van Dyke, J. C. 1887. *Principles of Art: Pt. 1. Art in History; Pt. 2. Art in Theory*. Fords, Howard, & Hulbert.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, 5998–6008.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, X.; and Ji, Q. 2013. A unified probabilistic approach modeling relationships between attributes and objects. In *Proceedings of the 2013 International Conference on Computer Vision (ICCV)*, 2120–2127. IEEE.

WikiArt. 2010. WikiArt: Visual Art Encyclopedia. http://www.wikiart.org. Accessed:2019-12-01.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; and othersa. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2018. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 41(9): 2251–2265.

Yu, F. X.; Cao, L.; Merler, M.; Codella, N.; Chen, T.; Smith, J. R.; and Chang, S.-F. 2014. Modeling attributes from category-attribute proportions. In *Proceedings of the 22nd ACM international conference on Multimedia (ACMMM)*, 977–980. ACM.