Classification of Artistic Styles using Binarized Features Derived from a Deep Neural Network

Yaniv Bar, Noga Levy, Lior Wolf

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract. With the vast expansion of digital contemporary painting collections, automatic theme stylization has grown in demand in both academic and commercial fields. The recent interest in deep neural networks has provided powerful visual features that achieve state-of-the-art results in various visual classification tasks. In this work, we examine the perceptiveness of these features in identifying artistic styles in paintings, and suggest a compact binary representation of the paintings. Combined with the PiCodes descriptors, these features show excellent classification results on a large scale collection of paintings.

1 Introduction

As digital acquisition of artistic images has advanced, vast digital libraries have been assembled over the Internet and in museums. With the development of recent automatic image analysis and machine vision techniques, the mission of artistic resource discovery is no longer left to the human expert. Automatic art identification and classification support the expert's mission of painting analysis, assist in organizing large collections of paintings and can be used for art recommendation systems.

Artistic visual styles such as impressionism, baroque and cubism have a set of distinctive properties which permits the grouping of artworks into related art movements. Therefore, every artwork has a visual style idiosyncratic "signature" which relates it to other works.

Style divisions are often identified and later defined by art experts and historians. This division is not a strict one; in many cases a style can span across many different painters where a single painter might span across several styles. Pablo Picasso, for example, painted in both surrealism and cubism styles. Some of these styles may be easily recognized by human art enthusiasts or experts while others are more subtle [2].

Visual style is not rigorously defined, but can be deduced from visual motifs present in a painting such as the choice of color palette, composition, scene, lighting, contours and brush strokes [21]. Yet, there has been little research in computer vision that explored style classification in recent years.

In this work, we investigate several known visual descriptors that attempt to extract these subtle artistic properties. These techniques serve as a benchmark for our approach and include gradient histograms, color histograms, statistical

methods [32], LBP [22], dictionary-based methods ([7], [26]) and pyramid based methods ([25], [19], [23], [29] [32]).

The "Picture Codes" (PiCoDes) [5] learns a compact binary code representation of an image optimized on a subset of ImageNet dataset [9], and is one of the leading methods tested.

Deep learning refers to generative machine learning models composed of multiple levels of non-linear operations, such as neural networks consisting of many layers. The deep architecture enables the generation of representations of mid-level and high-level abstractions obtained from raw data such as images.

Convolutional neural networks (CNNs) are feed-forward networks that can be learned efficiently, and recent results indicate that the generic descriptors extracted from CNN are very powerful and provide a breakthrough for recognition [28],[24].

In this work, we try to recognize the style of paintings using features extracted from a deep network, as well as PiCoDes and low-level descriptors. Combining features extracted from a deep network with PiCoDes yields a strong descriptor that outperforms all other tested descriptors while remaining compact.

The work that is most closely related to our work was done by Karayev et al. [16] on a variant of the artistic dataset we have used in this paper and using similar classes. However, in [16] small classes (less then 1000 images) are omitted, while we keep a more varying distribution. We use a low-dimensional binary descriptor, and analyze performance by average precision, average recall, F_1-score and classification accuracy. We combine different descriptors by concatenation or re-ranking based on the Borda count [25] and compare them to a broad set of methods.

An overview of the previous work is given in Section 2 , Section 3 describes the convolutional neural networks and our approach is explained in Section 4. Experimental setup is given in Section 5 followed by discussion on results in Section 6. Final conclusions are drawn in Section 7 .

2 Related Work

Readily available digitized art paintings data has been available for a few years, gaining more and more attention from researchers.

The related problem of artist identification within a specific style (Renaissance) was explored in [15] using the HOG descriptor. In [14], idiosyncratic characteristics of Van Gogh paintings were explored based on the artist's brush strokes. Stylization recognition is explored in [17] and [12] on smaller datasets with fewer classes compared to the dataset we use here. In [17], the dataset explored contains visually distinguishable artworks painted by a handful of painters. Each painter is associated with a different school or style. The dataset tested in [12] contains seven styles with less than a hundred images per style.

Comparative study of different classification methodologies for the task of fine-art style classification are covered in [32] and [1]. Several standard classifiers and features extraction techniques are explored in [32], inspired by visual cues

in painting such as gradient and statistical measures. In [1], discriminative and generative classification models are covered using intermediate level features (Bag of Words) and Semantic-level features. The datasets tested in [32] and [1] covers only a handful of styles with less than a hundred images per style.

In [16], deep features from the CNN in [18] are applied on large scale datasets. The conclusion arising from the experiments is that features extracted from deep architecture networks produce excellent classification performance on several large-scale datasets, including a dataset of paintings.

2.1 Low-Level Descriptors

Edge texture information [32]. The relative frequency of edges within a painting can be very informative. While impressionism is characterized by blurry and subtle edges, in analytical cubism and pop art the edges are very pronounced. The edge descriptor of an image is computed as the number of pixels that are labelled as an edge relative to the total number of pixels, extracted by the Canny edge detector for different sensitivity thresholds (0.2, 0.3, 0.4 and 0.6). Consequently, strong edges would be present in the image for any threshold level, while the subtle transitions would only show up for lower thresholds.

Texture information descriptor based on Steerable Filter Decomposition (SPD) [32],[8]. Steerable Filter Decomposition approximates a matching set of Gabor filters with different frequencies and orientations. The descriptor is 28-dimensional, consisting of the mean and variance of a low pass filter, a high pass filter, and 12 sub-band filters from three scales and four orientation decompositions. The mean and variance roughly correspond to the sub-band energy and characterize the artist brush strokes. The code is available in `http://live.ece.utexas.edu/research/quality`.

Color histogram [32]. The color histogram is described by a concatenation of three normalized 8-bin histograms, one per each HSV channel.

Statistical measures [32]. A concatenation of the mean, variance, skewness and kurtosis for each HSV channel.

Local Binary Patterns (LBP) [22]. LBP texture features are known for effective face recognition that can assist in scene and image texture classification. LBP texture features also perform well for face detection and are helpful in distinguishing portrait and non-portrait images [13]. The descriptor is derived by (a) dividing the image into cells (16×16 pixels for each cell), (b) for each pixel in a cell, the 3 x 3 neighborhood surrounding the pixel is thresholded with the central pixel intensity value, treating the subsequent pattern of 8 bits as a binary number, (c) a histogram is computed over the values in each cell and (d) the concatenation of the histograms of all cells forms a descriptor.

Dictionary-based descriptors [7],[26]. The bag-of-words model is applied to image classification by treating image features as words. Initially, a dictionary of visual "words" based on local feature descriptors is constructed using k-means and later compared against visual "words" (local features) of an image. The statistics of visual words occurrences are summarized in a sparse histogram.

Typically, local features are obtained using SIFT [20]. Fisher vectors serve a similar purpose of summarizing statistics of visual words, with two distinctions – the dictionary is obtained by Gaussian Mixture Models (GMMs), and rather than storing only visual word occurrences, the difference between dictionary words against the pooled image visual words is stored.

GIST [23],[11]. GIST is known to perform well for retrieving images that are visually similar at a low resolution scale, and consequently can represent the composition of an image to some extent. The descriptor is derived by resizing an image to 128 x 128 and iterating over the different scales where for each scale the image is divided into 8×8 cells. For each cell, orientation (every 45 degrees), color and intensity histograms are extracted, and the descriptor is a concatenation of all histograms, for all scales and cells. The code is available at `http://people.csail.mit.edu/torralba/code/spatialenvelope`

PHOG [19],[6]. Histogram of Oriented Gradients (HOG) is used for the purpose of object detection by counting occurrences of gradient orientation in localized portions of an image. The image is first divided into cells and for each cell a histogram of gradient directions or edge orientations is extracted. Concatenating all histograms forms a HOG descriptor. When this process is done on different scales, it is called Pyramid Histogram of Oriented Gradients (PHOG) and is known to perform well for scene categorization, which is required for some of the styles considered in this work.

The code is available at `http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html`.

SSIM [29]. A method of extracting a "local self-similarity" (SSIM) descriptor is depicted in [29]. While many methods measure the similarity among images by using common underlying visual properties, a SSIM descriptor captures internal geometric, color, edges, repetitive patterns and complex textures in a single unified way while accounting for small local affine deformations. That is, the descriptor captures spatial similarities between regions of different texture and color. The code is available at `http://www.robots.ox.ac.uk/~vgg/software/SelfSimilarity`.

Color Quad Trees (CQT) [25]. A quad tree is a tree data structure. The root represents the entire image, the second level contains four nodes each representing one quadrant of the image and so on, until the final level of the tree. For each node, the mean is computed over the pixels intensity values connected to the node. Concatenating all measurements level by level forms the CQT descriptor. This descriptor can be used to represent the composition of an image such as an outdoor landscape or a portrait to some extent.

2.2 Binary Compact Image Representations

PiCoDes binary features [5],[31]. A compact binary vector that is optimized to yield good categorization accuracy. As a preliminary step, an offline transformation matrix is learned as a non-linear combination of classifiers over features such as Bag of SIFTs, GIST, PHOG, and SSIM. The PiCoDes binary image

descriptor is computed by transforming the image data using the offline matrix such that the binary entries in the descriptor are thresholded projections of low-level visual features extracted from the image.

MC-Bit binary features [4],[31]. A descriptor learned as a non-linear combination of classifiers over the same bank of features is described for PiCoDes. While the classifiers of PiCoDes are jointly computed via an expensive iterative optimization that limits the actual number of classifiers (tested on a maximum of 2048 classifiers), the MC-bit classifiers are efficiently computed via recursive parallel optimization. Thus, the descriptor size can be scaled up.

VLG extractor was used for extracting both PiCoDes and MC-bit descriptors (`http://vlg.cs.dartmouth.edu/picodes/PiCoDes/Home.html`)

3 Deep Architecture Network

Deep neural networks have recently gained considerable interest due to the development of convolutional neural networks (CNN) that can be solved efficiently on large-scale datasets using limited computation capacity. The strength of deep networks is in learning multiple layers of concept representation, corresponding to different levels of abstraction. For visual datasets, the low levels of abstraction might describe edges in the image, while high layers in the network refer to object parts and even the category of the object viewed.

CNNs constitute a feed-forward family of deep networks, where intermediate layers receive as input the features generated by the former layer, and pass their outputs to the next layer.

Two popular choices are CNNs suggested by [18] and [27] for the Large Scale Visual Recognition Challenge of Imagenet [9], a large scale image database consisting of more than one million images categorized into 1000 classes. The features extracted from intermediate layers of these networks produce highly discriminative features that achieve state-of-the-art performance in visual classification tasks [28],[24].

The DeepFace architecture developed in [30] is another example of a successful CNN that achieves human accuracy in face recognition.

These CNNs are constructed of a few layers that learn convolutions, interleaved with non-linear and pooling operations, followed by locally or fully connected layers.

4 Our Approach

Our baseline descriptors are extracted from Decaf implementation [10] of a CNN trained on Imagenet, following the CNN in [18]. We use the notation of [10] to denote the activations of the n^{th} hidden layer of the network as $Decaf_n$, and use the $Decaf_5$ and $Decaf_6$ features as well as the final output of 1000 predictions. $Decaf_5$ contains 9216 activations of the last convolutional layer, and $Decaf_6$ contains 4096 activations of the first fully-connected layer.

4.1 Encoding Scheme

Following PiCoDes ([5], [31]), we suggest a compact binary encoding over the baseline descriptors that is designed to distinguish among different categories. Given a dataset classified into k categories and represented by a d-dimensional descriptor, our algorithm learns a d'-dimensional representation, where $d >> d'$ and d' is a multiplicative of k, that is $d' = tk$.
The encoding algorithm works as follows:

First, a subset of the training set is generated by randomly choosing an equal number of examples per class (25 examples per class in our experiments).

This subset is used to learn k linear SVMs in a One-vs-All manner – the i^{th} classifier is trained on m positive examples from class i and m negative examples randomly selected from all other classes ($m = 15$ in our experiments).

We learn t One-vs-All SVMs, that are each trained on different examples and generates k new binary classifiers. That is, we learn a total of d' binary classifiers with t classifiers per class.

For a new example x, define $p_i(x)$ as the binary decision of classifier i, $p_i(x) = \mathbb{1}\{w_i x - b_i\}$. The d'-dimensional encoding of x is $(p_1(x), \ldots, p_{d'}(x))$, the concatenation of the binary decisions of all classifiers on x.

Our dataset has 27 classes and we empirically set t to 15. Hence, the length of $Decaf_5$, for example, is reduced from 9216 dimensions to only 405 dimensions.

5 Experiments

Dataset. We use a subset of the WikiArt dataset which was collected from the visual art encyclopedia `www.wikiart.org` , a complete and well-structured online repository of fine art [3]. The collection describes 40,724 unique digitized paintings with variable resolution. Each painting is labelled with a subset of the following metadata, specifying the artist name and nationality, art movement (style), year of creation, material, technique, painting dimensions and the gallery it is presented at. Style label, however, is present in all paintings.

The collection covers over a thousand different artists and is categorized to 27 art styles. Figure 1 shows our dataset styles distribution. A small subset of less than 2% of the dataset is used to train the classifiers for the binary encoding, all other examples are used for multiclass classification.

Multiclass classification A three-folded cross validation was used. We applied popular machine learning classifiers – SVM, Adaboost, Nave Bayes and kNN and chose empirically the 5-NN classifier since it produced the best results among descriptors that were tested.
Accuracy Metrics We measure the success of the multiclass classification by the following measures:

- Classification accuracy – the rate of correctly classified examples out of all examples.

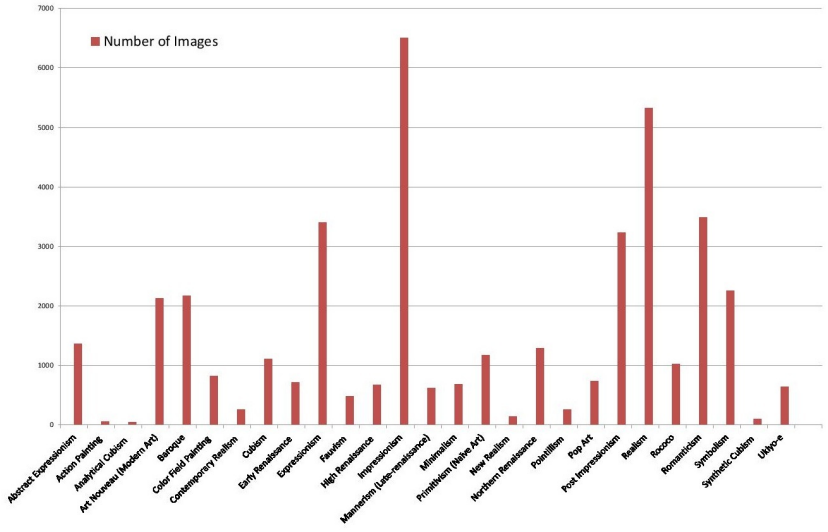

Fig. 1. Distribution of the image styles

- Precision and Recall – the precision of a class c is the proportion of images that are indeed in c out of the images predicted as c by the classifier, and recall is the proportion of images in c that are classified correctly. We report the average precision and recall over all classes. While accuracy is biased to larger classes, the average precision gives equal weight to all classes regardless of their size.
- F_1-score combines recall and precision, computed as $\frac{2precision \times recall}{precision + recall}$.

Descriptors Fusion We use two approaches to combine different descriptors. The early fusion method combines the different descriptors before applying classification. The late fusion refers to assembling classification results based on the different descriptors.

For early fusion, we simply concatenate the descriptors. Since the range of values of raw data varies widely, the values of each feature are normalized to zero mean and standardized to unit variance.

In the late fusion approach, the features of each descriptor are learned separately and their outputs are then merged. We adapt a re-ranking method based on a Borda count [25]. The Borda count is a single-winner election method in which voters rank options or candidates in order of preference. Once all votes are counted, the candidate that gained the highest number of points is the winner. In this work, the voters are the classifiers trained on different descriptors and the candidates are styles. For a given test image, kNN finds for each descriptor, the five closest images. We score the style results of those images in a descending order (5 to 1) and perform a maximum points scored vote based on the scores

Table 1. The performance of multiclass classification on the paintings dataset when using low-level descriptors

Descriptor	**Dim.**	**Average Precision**	**Average Recall**	**F_1-Score**	**Accuracy**
Edge texture information [32]	4	0.09	0.07	0.08	0.14
SPD [32],[8]	28	0.17	0.14	0.15	0.22
Color histogram [32]	24	0.14	0.11	0.12	0.20
Statistical measures [32]	12	0.09	0.07	0.08	0.14
LBP [22]	59	**0.18**	**0.15**	**0.16**	**0.23**
Bag Of Sifts [7]	500	0.17	0.13	0.15	0.22
Fisher Vectors [26]	1024	0.11	0.10	0.10	0.15
GIST [23],[11]	512	0.16	0.13	0.15	0.21
PHOG [19],[6]	336	0.13	0.10	0.11	0.19
SSIM [29]	1024	0.13	0.09	0.11	0.13
CQT [25]	1023	0.12	0.09	0.10	0.13

given by all descriptors.

Code. The entire code was implemented in Matlab using VLFEAT framework `www.vlfeat.org`. The Decaf feature extraction part is obtained from Decaf CNN [10].

6 Results

Table 1 shows the performance of low-level descriptors. LBP, SPD and SIFT descriptors achieve the best performance in terms of accuracy. Empirically, descriptors that achieve the highest accuracy also achieve high scores in the other metrics. Therefore, we analyze the results based on accuracy and the other metrics are reported in the supporting tables. Table 2 shows PiCoDes with various dimensions and MC-bit descriptor. Not surprisingly, both descriptors outperform all low-level features since they are learned in an optimized way over a combination of powerful low-level descriptors.

In Table 3, the results of the Decaf CNN descriptors with and without binary encoding are presented. Similarly to the results in Table 2, the deep features outperform all low-level descriptors tested.

In Table 4, several forms of features fusion are examined, incorporating combinations of low-level descriptors, PiCoDes descriptors and Decaf CNN descriptors. Examining both early fusion (EF) and late fusion (LF) approaches, we reported only LF results as both of them gave similar results.

The best features fusion descriptor incorporates PiCoDes (2048-dimensionality) and $Decaf_6$ (4096-dimensionality), and has a 43% accuracy, a 6% increase over the best result of single descriptors.

Table 5 shows several forms of binary features fusion results that incorporates combinations of PiCoDes descriptors and encoded Decaf CNN descriptors.

Table 2. The performance of multiclass classification on the paintings dataset when using binary compact image representations

Descriptor	**Dim.**	**Sparsity Ratio**	**Average Precision**	**Average Recall**	**F_1-Score**	**Accuracy**
PiCoDes [5]	128	0.44	0.21	0.16	0.18	0.24
PiCoDes [5]	1024	0.53	0.37	0.26	0.31	0.35
PiCoDes [5]	2048	0.45	**0.38**	**0.29**	**0.33**	**0.37**
MC-bit [4]	15232	0.70	0.31	0.25	0.28	0.32

Table 3. The performance of multiclass classification on the paintings dataset when using Decaf CNN and encoded Decaf CNN descriptors. The sparsity ratio is the average number of zero elements within the descriptor

Descriptor	**Dim.**	**Type**	**Sparsity Ratio**	**Average Precision**	**Average Recall**	**F_1-Score**	**Accuracy**
Decaf predictions [10]	1000	Real	0.26	0.19	0.15	0.16	0.21
$Decaf_6$ [10]	4096	Real	0.00	**0.38**	**0.31**	**0.34**	**0.37**
$Decaf_5$ [10]	9216	Real	0.72	0.42	0.26	0.32	0.35
Encoded $Decaf_6$	405	Binary	0.65	0.30	0.27	0.28	0.34
Encoded $Decaf_5$	405	Binary	0.54	0.27	0.23	0.25	0.31

Table 4. Multiclass classification performance results when using features fusion on the paintings dataset. All results for Late-Fusion approach. We use the following abbreviation: PD - PiCoDes

Descriptor	**Dim.**	**Type**	**AP.**	**AR.**	**F_1.**	**Accuracy**
HSV + LBP + SPD	111	Real	0.21	0.14	0.17	0.25
GIST + PHOG + SSIM + BOW	2372	Real	0.32	0.15	0.21	0.26
Decaf6 + LBP + SPD	4183	Real	0.42	0.25	0.31	0.35
Decaf6 + GIST	4608	Real	0.41	0.29	0.34	0.37
PD-2048 + LBP +SPD	2135	Real	0.43	0.24	0.31	0.35
PD-2048 + $Decaf_5$ + $Decaf_6$	15360	Real	**0.56**	0.32	**0.41**	0.39
PD-2048 + $Decaf_6$	6144	Real	0.48	**0.36**	**0.41**	**0.43**
PD-1024 + PD-2048 + $Decaf_6$	7168	Real	0.44	0.29	0.35	0.37
PD-1024 + PD-2048 + $Decaf_5$ + $Decaf_6$	16384	Real	0.50	0.34	**0.41**	0.42

Each combination is reported for both early fusion and late fusion. All combinations show excellent results in terms of accuracy and descriptor compactness, matching or surpassing non-encoded features fusion results. The best features fusion descriptor incorporates PiCoDes (1024-dimensionality), PiCoDes (2048-dimensionality), encoded $Decaf_5$ (405-dimensionality) and encoded $Decaf_6$ (405-dimensionality) and matches the best (non-encoded) features fusion result using a binary descriptor with 63% compression.

Table 5. Multiclass classification performance results when using binary features fusion on the paintings dataset. We use the following abbreviations: EF - early fusion, LF - late fusion, PD - PiCoDes and Enc - encoded

Descriptor	**Dim.**	**Fusion**	**Average Precision**	**Average Recall**	**F_1-Score**	**Accuracy**
PD-2048 + Enc. $Decaf_6$	2453	EF	0.43	0.34	0.38	0.41
PD-2048 + Enc. $Decaf_6$	2453	LF	0.43	0.34	0.38	0.41
PD-2048 + Enc. $Decaf_5$ + Enc. $Decaf_6$	2858	EF	0.43	**0.36**	0.39	0.42
PD-2048 + Enc. $Decaf_5$ + Enc. $Decaf_6$	2858	LF	0.43	0.33	0.38	0.41
PD-1024 + Enc. PD-2048 + Enc. $Decaf_5$ + Enc. $Decaf_6$	2239	EF	0.42	**0.36**	0.39	0.42
PD-1024 + Enc. PD-2048 + Enc. $Decaf_5$ + Enc. $Decaf_6$	2239	LF	0.46	0.34	0.39	0.42
PD-1024 + PD-2048 + Enc. $Decaf_5$ + Enc. $Decaf_6$	3882	EF	0.42	**0.36**	0.39	0.42
PD-1024 + PD-2048 + Enc. $Decaf_5$ + Enc. $Decaf_6$	3882	LF	**0.47**	0.34	**0.40**	**0.43**

A greater breakdown is illustrated in Figure 2 by showing the confusion matrix of our best features fusion.

Different painting styles share similarities of color, composition and texture as well as sharing the object of the painting (e.g. still life, landscape, portraits etc.). Thus, misclassification between closely related styles occur quite often. A closer look in Figure 2 reveals the quality of classification. Confusion between relatively unrelated styles might occur. For example, Fauvism and Cubism are visually distinct as fauvism uses strong colors while cubism uses bland colors. In cubism objects are often reduced to their geometric form in a non-realistic way while fauvism is a more realistic simplified style. Cubism is precisely rendered while fauvism is loose and minimal. However, misclassification errors tend to occur significantly more frequently among closely related groups of styles, reflecting subtleties within these styles. Several examples are Renaissance styles (such as Early Renaissance, Late Renaissance, High Renaissance and Northern Renaissance) and Cubism related styles. Styles that are influenced by other styles in terms of visual motifs (continuation, branching out or reaction movement) also tend to get confused, such as Abstract Expressionism and Color Field Painting, Minimalism and Color Field Painting or Impressionism and Realism. For example, Color Field painting is referred as an extension of Abstract Expressionism paintings; both are abstract and express a dramatic use of colors. Figure 3 qualitatively demonstrates these relations through the confusion matrix, by rearranging the order of the styles so that groups of related styles are clustered together.

Experimentally, our best features fusion descriptor has the ability to find similarities within the style or even within the genre without over-tuning to a specific set of visual cues. For example, Portrait and landscape images appear in about half of the styles, yet performing a landscape/portrait image search of a specific style works well and landscape/portrait images of the same style are retrieved.

7 Conclusions

In this work, we suggest a representation of paintings that is both simple and efficient. We applied Decaf, a novel deep network trained on a comprehensive real-life ImageNet dataset to a problem of a different nature, known as art stylization. Style recognition differs from the task of object recognition, since two paintings can describe the same scene (e.g. a landscape painting of a marina) using very different artistic techniques. For example, a realistic painting as opposed to pointillism, a technique of painting in which small, distinct dots of pure color are applied to form a painting. For efficiency, we suggest an approach for encoding the features obtained by the Decaf convolutional network into a lower dimensional binary representation. Experiments lead us to conclude that deep features as well as their encoded version can distinguish styles better than hand-crafted low level descriptors.

When combining Decaf encoding with PiCoDes, a method of assembling low level features in an optimized way, we receive state-of-the-art performance results. A closer inspection of the misclassified examples indicate that this fused representation captures the subtle characteristics of styles and tends to mix up closely related styles, a reasonable confusion for the non-expert human observer.

There is still considerable room for improvement. One future direction is using a hierarchical style-based classification. A different direction is incorporating other deep neural network features known for good classification and detection results (e.g. OverFeat[28]). Examining the classification performance of our descriptor on other classification tasks – either art related (genre classification) or real life images, is also worth trying.

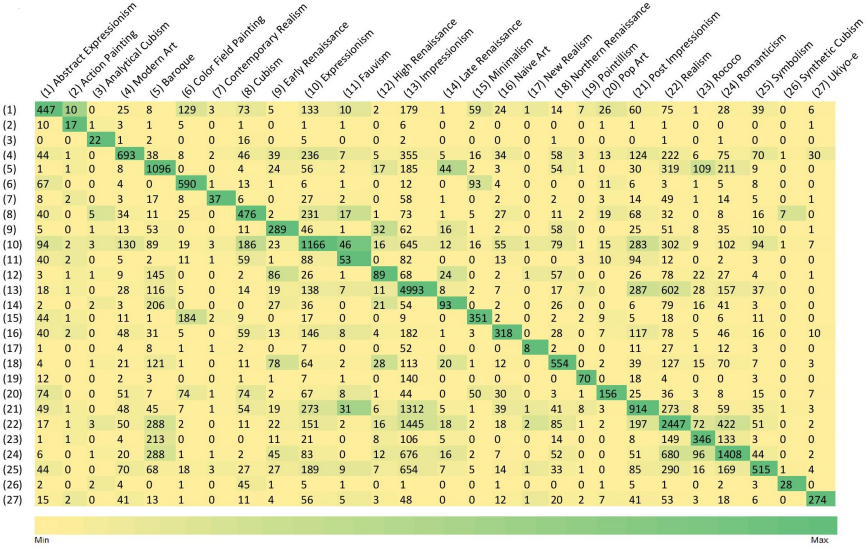

Fig. 2. The confusion matrix of our best features fusion representation: PiCoDes (1024-dim.), PiCoDes (2048-dim.), encoded $Decaf_5$ (405-dim.) and encoded $Decaf_6$ (405-dim.). Columns are color scaled from a minimum value of 0 (yellow) to a per-style maximum value (green). The main diagonal cells of the matrix, representing correct style classification is marked in green colors while off-diagonal cells, representing misclassification are marked in yellowish-light green colors. As described in Section 5, most of the off-diagonal cells that are marked in light-green colors represent misclassification between correlated styles

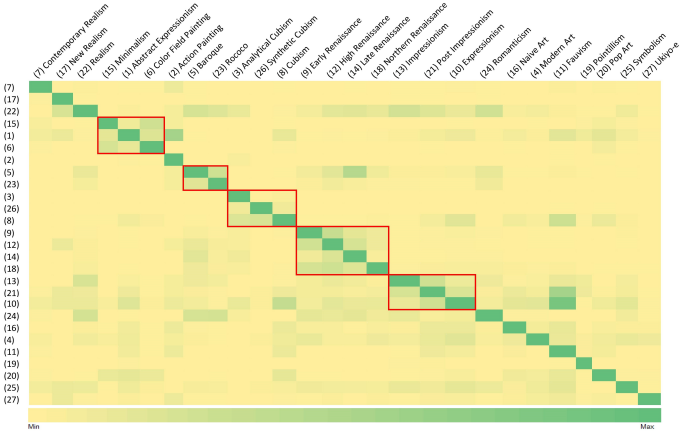

Fig. 3. The confusion matrix of our best features fusion representation: PiCoDes (1024-dim.), PiCoDes (2048-dim.), encoded $Decaf_5$ (405-dim.) and encoded $Decaf_6$ (405-dim.), rearranged by related styles. Each red box indicates the confusion within a family of styles

References

1. Arora, R.S.: Towards automated classification of fine-art painting style: A comparative study. Ph.D. thesis, Rutgers University-Graduate School-New Brunswick (2012)
2. Beckett, W., Wright, P.: The story of painting. Dorling Kindersley London (1994)
3. Ben-Shalom, I., Levy, N., Wolf, L., Dershowitz, N., Ben-Shalom, A., Shweka, R., Choueka, Y., Hazan, T., Bar, Y.: Congruency-based reranking. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014 (2014)
4. Bergamo, A., Torresani, L.: Meta-class features for large-scale object categorization on a budget. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 3085–3092. IEEE (2012)
5. Bergamo, A., Torresani, L., Fitzgibbon, A.W.: Picodes: Learning a compact code for novel-category recognition. In: Advances in Neural Information Processing Systems. pp. 2088–2096 (2011)
6. Bosch, A., Zisserman, A., Munoz, X.: Image classification using rois and multiple kernel learning. International journal of computer vision 2008, 1–25 (2008)
7. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV. vol. 1, pp. 1–2 (2004)
8. Deac, A.I., van der Lubbe, J., Backer, E.: Feature selection for paintings classification by optimal tree pruning. In: Multimedia Content Representation, Classification and Security, pp. 354–361. Springer (2006)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
10. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)
11. Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., Schmid, C.: Evaluation of gist descriptors for web-scale image search. In: Proceedings of the ACM International Conference on Image and Video Retrieval. p. 19. ACM (2009)
12. Ivanova, K., Stanchev, P., Velikova, E., Vanhoof, K., Depaire, B., Kannan, R., Mitov, I., Markov, K.: Features for art painting classification based on vector quantization of mpeg-7 descriptors. In: Data Engineering and Management, pp. 146–153. Springer (2012)
13. Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved lbp under bayesian framework. In: Multi-Agent Security and Survivability, 2004 IEEE First Symposium on. pp. 306–309. IEEE (2004)
14. Johnson, C.R., Hendriks, E., Berezhnoy, I.J., Brevdo, E., Hughes, S.M., Daubechies, I., Li, J., Postma, E., Wang, J.Z.: Image processing for artist identification. Signal Processing Magazine, IEEE 25(4), 37–48 (2008)
15. Jou, J., Agrawal, S.: Artist identification for renaissance paintings
16. Karayev, S., Hertzmann, A., Winnemoeller, H., Agarwala, A., Darrell, T.: Recognizing image style. arXiv preprint arXiv:1311.3715 (2013)
17. Keren, D.: Painter identification using local features and naive bayes. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. vol. 2, pp. 474–477. IEEE (2002)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

19. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 2, pp. 2169–2178. IEEE (2006)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)
21. Mishory, A.: Art history: an introduction. Open University of Israel (2000)
22. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern recognition 29(1), 51–59 (1996)
23. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision 42(3), 145–175 (2001)
24. Oquab, M., Bottou, L., Laptev, I., Sivic, J., et al.: Learning and transferring mid-level image representations using convolutional neural networks (2013)
25. Parker, J.R.: Algorithms for image processing and computer vision. John Wiley & Sons (2010)
26. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
27. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
28. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. arXiv preprint arXiv:1403.6382 (2014)
29. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. pp. 1–8. IEEE (2007)
30. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: IEEE CVPR (2014)
31. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: Computer Vision–ECCV 2010, pp. 776–789. Springer (2010)
32. Zujovic, J., Gandy, L., Friedman, S., Pardo, B., Pappas, T.N.: Classifying paintings by artistic genre: An analysis of features & classifiers. In: Multimedia Signal Processing, 2009. MMSP'09. IEEE International Workshop on. pp. 1–5. IEEE (2009)