

DeepPainter: Painter Classification Using Deep Convolutional Autoencoders

Omid E. David¹ and Nathan S. Netanyahu^{1,2}

¹ Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel
mail@omidavid.com, nathan@cs.biu.ac.il

² Center for Automation Research, University of Maryland, College Park, MD, USA
nathan@cfar.umd.edu

Abstract. In this paper we describe the problem of painter classification, and propose a novel approach based on deep convolutional autoencoder neural networks. While previous approaches relied on image processing and manual feature extraction from paintings, our approach operates on the raw pixel level, without any preprocessing or manual feature extraction. We first train a deep convolutional autoencoder on a dataset of paintings, and subsequently use it to initialize a supervised convolutional neural network for the classification phase.

The proposed approach substantially outperforms previous methods, improving the previous state-of-the-art for the 3-painter classification problem from 90.44% accuracy (previous state-of-the-art) to 96.52% accuracy, i.e., a 63% reduction in error rate.

1 Introduction

Art forgery, which dates back more than two thousand years, has played a key role in the development of painting authentication. This task has been usually performed manually by art experts who have dedicated their lives to this profession. Their expertise amounted to using various characteristics other than what the human eye can see, including chemical analysis, spectrometry, and infrared or X-ray imaging. The infamous Vermeer forgery [12] attests, perhaps, most vividly to the challenges presented by painting authentication. Han van Meegeren used historical canvasses and managed to deceive art experts into believing that his painting was an authentic Vermeer. Only after being charged with treason and sentenced to death for selling another (forged) Vermeer, did he confess and was forced to create another painting to prove himself innocent of treason. A more recent case of painting authenticity involves the Pollock paintings found a decade ago in a storage locker in Wainscott, NY. The authenticity of these paintings was compromised on the basis of computer analysis of the paintings' fractal dimension [14]. This claim was subsequently disputed by analyzing childlike drawings that supposedly have the same fractal dimension as the Pollock paintings [3].

In this paper we address the closely related problem of painting classification, i.e., the task of assigning a specific artist to a given painting (from a dataset of paintings by several artists). Note that the image authentication problem can be viewed as a binary image classification problem (i.e., determine whether or not a given painting was painted by a certain artist). Recent developments for both problem types have focused on preprocessing techniques of reducing the high dimensionality of visual

data to low-dimensional representations which can be manipulated towards image understanding.

Levy *et al.* [9, 10] applied feature extraction to paintings using generic image processing (IP) functions (e.g., fractal dimension, Fourier spectra coefficients, texture coefficients, etc.), and restricted Boltzmann machines (RBM), followed by genetic algorithms (GA)-based learning of the weights of a weighted nearest neighbor (WNN) classifier [13]. Their approach achieved 90.44% classification accuracy for the 3-painter classification problem.

In this paper we present the problem of painter classification and briefly survey recent research that has been conducted in the field. We then present our novel approach, which uses convolutional autoencoders (CAE) instead of image processing based feature extraction. We subsequently use the trained CAE to initialize a convolutional neural network (CNN) for supervised training on specific painters. The results demonstrate a substantial improvement over previous methods, improving the accuracy to 96.52%. This sets a new state-of-the-art for the painter classification problem,

2 Background

Image authentication is the task of determining whether or not a given painting was painted by a specific artist. The related task addressed by us, though, is image classification, i.e., the task of determining the artist of a given painting (from a certain group of artists). The input to our problem consists of painting images of the group of artists (several paintings of each artist), and our objective is to automatically classify a given painting. One of the difficulties in solving this problem is that we cannot define a certain set of rules that the painting has to conform to in order to classify it to the subgroup corresponding to the correct artist. For this reason, computer vision techniques which are capable of identifying shapes and objects in an image are not sufficiently effective for solving the problem.

Formerly there have been attempts to harness the strength of image analysis tools to classify historical art paintings into categories of artists or genres. Levy *et al.* [9] used GA-based WNN with a set of 78 prevalent image features for classifying paintings by Rembrandt, Renoir, and van Gogh, obtaining 80% classification accuracy. In their later work [10], they augmented their approach by also adding 20 features using restricted Boltzmann machines (RBM)[2], improving the classification accuracy to 90.44%.

Herik and Postma [15] surveyed image features relevant to the historic art domain and concluded that neural network techniques combined with domain knowledge were most suitable to the task of automatic image classification. Under-drawing strokes in infrared reflectograms were analyzed by Kammerer *et al.* [4] in order to classify how and by what tools paintings are painted. Natural language processing techniques using a naive-Bayes classifier and the coefficients of a discrete cosine transform (DCT) were used by Keren [5] in order to classify local features in an image. Kroner *et al.* [7] classified drawings by using image histograms and pattern recognition methods.

The above past research focused on specific image processing features tailored for specific datasets (such as ink paintings, infrared reflectograms, or black and white sketches). This domain-specific knowledge facilitates the exploitation of various characteristics of the painting-specific domain.

In the next section we present our convolutional autoencoder based approach, which does not incorporate any domain-specific knowledge, and in fact is operating solely on the raw pixel level.

3 Feature Extraction Using Convolutional Autoencoders

3.1 Convolutional Neural Networks

In recent years convolutional neural networks (CNN)[1, 8, 6] have outperformed conventional image processing methods in all computer vision related tasks they have been applied to. The architecture of a CNN typically includes several components which are stacked on top of each other: the convolutional layer, the max-pooling layer, which subsamples the data (e.g., for each 2×2 region selects only the maximum value, thus resulting in four times reduction in size), and finally a classification layer (and usually several fully connected layers before the classification layer). Figure 1 shows a typical CNN,

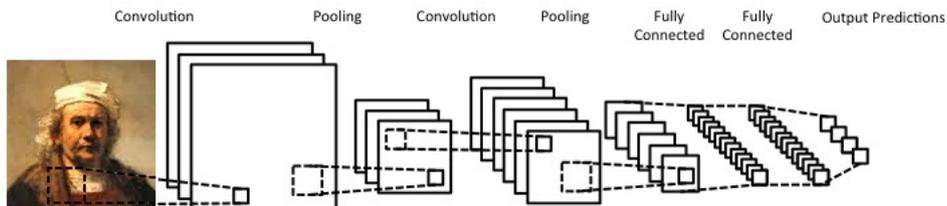


Fig. 1. Typical architecture of a convolutional neural network.

Standards CNNs are usually used in a supervised framework, where a large training dataset (typically including at least many thousands of images per class) is available. Thus, using CNNs for end-to-end painter classification is problematic, due to a smaller number of training samples available per painter (usually from a few tens of paintings to at most a few hundred paintings for more prolific painters).

3.2 Autoencoders

Where small number of training samples are available, unsupervised pretraining has proven highly effective [2, 16]. Unsupervised training methods using neural networks involve either the use of restricted Boltzmann machines (RBM)[2] which are trained using contrastive divergence, or autoencoders[16] which are training using standard backpropagation.

The basic principle for all methods involves receiving an input \mathbf{x} and mapping it to a latent representation \mathbf{h} , using a function $\mathbf{h} = \sigma(Wx + b)$, where σ is a nonlinear activation function, W is a matrix of weights between the two layers, and b is bias. The autoencoder then tries to reconstruct the original input by $\mathbf{y} = \sigma(W'h + b')$. Thus, each training sample x_i is first mapped to a hidden layer h_i and then reconstructed to y_i . The autoencoder is trained using backpropagation to reduce this reconstruction error.

3.3 Convolutional Autoencoders

The principles behind convolutional neural networks and autoencoders can be combined to produce convolutional autoencoders (CAE). Several approaches involving the combination of these methods have been explored in the past, and here we use a CAE architecture along the lines presented in [11, 18, 17].

In order to use CNNs as autoencoders, for each convolutional layer, a corresponding deconvolutional layer should be constructed. Additionally, max-pooling layers result in loss of information, and so an unpooling layer should try to approximately restore the original values. Note that the subsampling due to max-pooling in fact operates as a strong regularizer.

Deconvolution layers can either be equal but transposed to the original convolution layers, or learned from scratch. Often both approaches work equally well in practice. This is similar to standard autoencoders where the weights of the decoder layer W' can either be learned from scratch, or set to the transpose of the encoder layer ($W' = W^T$), this is referred to as *tied weights*.

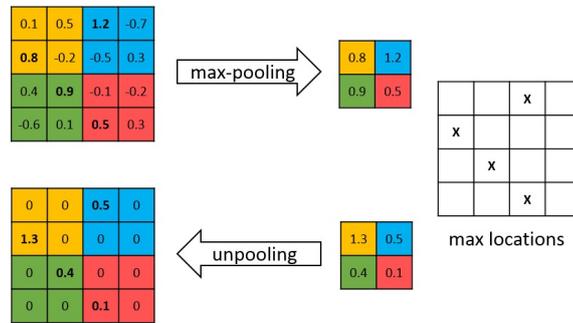


Fig. 2. Pooling and unpooling layers. For each pooling layer, the max locations are stored. These locations are then used in unpooling layer.

Several methods have been applied in the past for the unpooling operation [11, 18, 17]. Here we employ the method used in [18], where during pooling, the location of maximum value is stored, such that during unpooling the value is restored in that location, and the other locations are set to zero. Figure 2 illustrates unpooling, and Figure 3 shows a complete convolutional autoencoder structure.

After training a CAE, we can remove the unpooling and deconvolution components, and use the convolution and pooling components to initialize a supervised CNN, by adding a fully connected layer followed by a classification layer.

4 CAE and CNN for Painter Classification

For unsupervised training of CAE, we use a randomly selected set of 5,000 paintings from the Webmuseum (webmuseum.meulie.net/wm). The images have 24-bit color depth with varying resolutions averaged approximately at 1000×1000 pixels, and compressed as JPEG formatted files. We have resampled the images and normalized them to 256×256 pixels. The goal here is to train the CAE to find features that are

specifically useful for paintings, which have a more specific color and composition range in comparison to real-world images.

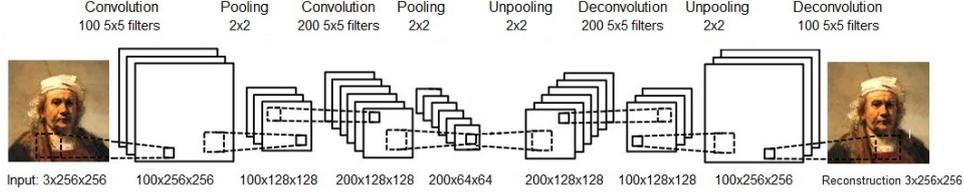


Fig. 3. Illustration of convolutional autoencoder. In this example the CAE comprises two convolution layers and their two corresponding deconvolution layers, and two max-pooling layers and their corresponding unpooling layers.

Our CAE contains the following layers (see Figure 3). The convolution filter sizes are always of size 5×5 .

1. the input layer consists of the raw image (resampled to 256×256 pixels) in three channels (R, G, and B)
2. convolutional layer of size $100 \times 256 \times 256$
3. max-pooling layer of size 2×2
4. convolutional layer of size $200 \times 128 \times 128$
5. max-pooling layer of size 2×2
6. unpooling layer of size 2×2
7. deconvolutional layer of size $200 \times 128 \times 128$
8. unpooling layer of size 2×2
9. deconvolutional layer of size $100 \times 256 \times 256$

The learning rate starts from 0.01 and is multiplied by 0.98 after each epoch. In order to further encourage the CAE to find meaningful features, we randomly remove 20% of the pixels for the images per epoch. The concept here is similar to that of denoising autoencoders [16] which outperform traditional autoencoders.

The supervised classification benchmark is identical to that used by Levy *et al.* [9, 10] in their experiments. It consists of ($3 \times 40 =$) 120 digital reproductions of paintings by Rembrandt, Renoir, and van Gough, downloaded from the Webmuseum. The Appendix contains the painting titles of the images used in our experiments.

Having trained a CAE, we can now remove the decoder components (items 6 to 9 in the above list) and use the CAE for initializing a supervised CNN. On top of these components due to CNN, we add two fully connected layers of size 400 and 200, followed by a softmax output unit of size three (since there are three painters in the benchmark). The cross entropy loss is used.

The full CNN contains the following layers (see Figure 1):

1. the input layer consists of the raw image (resampled to 256×256 pixels) in three channels (R, G, and B)
2. convolutional layer with 100 5×5 filters per input channel
3. max-pooling layer of size 2×2

4. convolutional layer with 200 5×5 filters per map
5. max-pooling layer of size 2×2
6. fully connected layer of size 400
7. fully connected layer of size 200
8. output softmax layer of size 3

To make our results directly comparable to those of Levy *et al.* [9, 10], we conducted 10-fold cross validation, where in each of 10 runs 90% of the data is used for training, and 10% for validation.

After performing 10 such training and validation runs, the average accuracy obtained for our CNN over the validation set is 96.52%. This represents a 63% reduction in error rate in comparison to the previous state-of-the-art on this benchmark, which stood at 90.44%.

Table 1 provides a summary of the classification accuracies obtained by previous methods and our method.

Feature extraction method	Supervised learning method	Accuracy
Image Processing	Nearest Neighbor	65.71%
Image Processing	SVM	68.33%
Image Processing	Genetic Algorithm	78.33%
RBM	Nearest Neighbor	64.41%
RBM	SVM	77.50%
RBM	Genetic Algorithm	73.92%
Image Processing + RBM	Nearest Neighbor	68.71%
Image Processing + RBM	SVM	71.66%
Image Processing + RBM	Genetic Algorithm	90.44%
Convolutional Autoencoder	CNN	96.52%

Table 1. Classification accuracy for several previous methods and our CAE based method. The results are the average over 10-fold cross validation.

5 Conclusion

Automatic painter classification has gained much attention over the past decades, and much progress has been made with regards to both relevant preprocessing techniques and classification algorithms. Still, the problem of painter classification remains a complex task that requires more sophisticated techniques.

The results presented in this paper show that deep learning methods can be effectively employed for painter classification. Specifically, our results show that convolutional autoencoders are capable of extracting meaningful information from paintings, and combined with supervised convolutional networks, we managed to substantially improve the previous state-of-the-art, from 90.44% accuracy (previous state-of-the-art) to 96.52% accuracy, i.e., a 63% reduction in error rate.

References

1. S. Behnke. Hierarchical neural networks for image interpretation. *LNCS*, 2766:1–13, 2003.
2. G.E Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
3. K. Jones-Smith and H. Mathur. Fractal analysis: revisiting Pollock’s drip paintings. *Nature*, 444(7119):E9–E10, 2006.
4. P. Kammerer, M. Lettner, E. Zolda, and R. Sablatnig. Identification of drawing tools by classification of textural and boundary features of strokes. *Pattern Recognition Letters*, 28(6):710–718, 2007.
5. D. Keren. Painter identification using local features and naive Bayes. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 2, pages 474–477, 2002.
6. Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.
7. S. Kroner and A. Lattner. Authentication of free hand drawings by pattern recognition methods. In *Proceedings of the IEEE 14th International Conference on Pattern Recognition*, volume 1, pages 462–464, 1998.
8. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11).
9. E. Levy, O.E. David, and N.S. Netanyahu. Painter classification using genetic algorithms. In *IEEE Congress on Evolutionary Computation*, pages 3027–3034, 2013.
10. E. Levy, O.E. David, and N.S. Netanyahu. Genetic algorithms and deep learning for automatic painter classification. In *ACM Genetic and Evolutionary Computation Conference*, pages 1143–1150, 2014.
11. Jonathan Masci, Ueli Meier, Dan Ciresan, and Jurgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks, ICANN*, pages 52–59, 2011.
12. D. Phillips. How do forgers deceive art critics? *The Artful Eye*. R. Gregory, J. Harris, P. Heard, and D. Rose, Eds., Oxford University Press, pages 372–388, 1995.
13. W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.
14. R.P. Taylor, R. Guzman, T.P. Martin, G.D.R. Hall, A.P. Micolich, D. Jonas, B.C. Scannell, M.S. Fairbanks, and C.A. Marlow. Authenticating Pollock paintings using fractal geometry. *Pattern Recognition Letters*, 28(6):695–702, 2007.
15. H.J. van den Herik and E.O. Postma. Discovering the visual signature of painters. *Future Directions for Intelligent Systems and Information Sciences*. N. Kasabov, Ed., Physica-Verlag, Heidelberg, pages 129–147, 2000.
16. P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *25th international Conference on Machine learning, ICML*, pages 1096–1103, 2008.
17. M.D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision, ECCV*, pages 818–833, 2014.
18. M.D. Zeiler, G.W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *International Conference on Computer Vision, ICCV*, pages 2018–2025, 2011.

This appendix lists the $(40 \times 3) = 120$ titles of the paintings experimented with by van Gogh, Rembrandt, and Renoir.

#	van Gogh	Rembrandt	Renoir
1	bandaged-ear	abraham	apres-bain
2	berceuse	anslo	baigneuses
3	cordeville	aristotle-homer	bathers-1887
4	corridor-asylum	artemis	bathers-1918
5	cypress-star	bathing-river	bougival
6	cypresses	bathsheba	canoeist
7	flower-beds-holland	belshazzar	chocquet
8	green-vineyard	children	city
9	green-wheat-field	danae	country
10	house-ploughman	david	dancer
11	mme-trabuc	descent	durieux
12	mr-trabuc	emmaus	flowers
13	old-mill	hendrickje	gabrielle
14	old-vineyard	holy-family	girl-seated
15	olive-almilles	jan-six	jugglers
16	olive-trees	magn-glass	lady-piano
17	orchard-bloom-poplars	meditation	laundress
18	orchard-plum-trees	mill	loge
19	poppies	music-party	lucie-berard
20	red-vineyard	nicolaes-tulp	near-lake
21	reminiscences	old-man	fournaise
22	road-menders	ostrich	horsewoman
23	roulin	potiphar	meadow
24	self-1	prodigal-son	moulin-galette
25	self-2	raising-lazarus	nini
26	self-easel	.1640	parapluiers
27	self-gauguin	.1661	premiere-sortie
28	self-orsay	.1669	promenade
29	self-whitney	.night-watch	ride
30	skull-cigarette	return-prodigal-son	romain-lacaux
31	sun-cloud	ruts	sisley-wife
32	threatening-skies	samson	women
33	trees-asylum	scholar	seashore
34	trees-ivy-asylum	self-1629	seated-bather
35	village-stairs	self-1634	sewing
36	wheat-field	self-1660	sisley
37	wheat-rising-sun	slaughtered-ox	swing
38	willows	staalmeesters	terrace
39	peasant	stofells	watercan
40	woman-arles	tobias	woman-veil