Artistic Image Recovering from Principal Components

Jorge Isaac Chang Ortega[1*] and Edgar Roman-Rangel[2]
[1] Brown University
[2] Instituto Tecnológico Autónomo de México (ITAM)
Email: jorge_chang@brown.edu, edgar.roman@itam.mx

Abstract

This work presents a comparison of different deep learning models for the reconstruction of artistic images from compact representations generated using Principal Component Analysis. The reconstruction models correspond to different types of Convolutional Neural Networks. Our results show that the statistics captured by the principal components transformation are enough to obtain good approximations in the reconstruction process, especially in terms of color and object visual features, even when using compact representations whose length is only about 1% of the original image space's total number of features.

1 Introduction

Principal Component Analysis (PCA) [1] has been used extensively in a variety of Machine Learning tasks. Specifically, since PCA generates representations in a new space of decorrelated features, it has been proven to work as a powerful tool in several computer vision tasks, including dimensionality reduction, face recognition, and image classification [2]. However, it presents limitations given its methodology. Namely, when used as a dimensionality reduction tool, it results in the loss of some information that, while of low variability, could still be relevant for reconstruction purposes. Moreover, PCA reduces its performance with datasets in which the correlation of features tends to zero [1].

On the other hand, the promise of neural networks [3], and in particular, of Convolutional Neural Networks (CNN's) [4], has permeated through virtually all areas of Artificial Intelligence, since they have presented significant advances and better alternatives to various challenges [5]. Hence, assuming that PCA has been used to compute *latent representations* (compact decorrelated vectors) of images, while inducing some reduction of their

*Corresponding Author

dimensionality and therefore some loss of information, it makes sense to investigate the potential of CNN's in the task of recovering the original images from their respective latent representations.

The **goal of this paper** is to evaluate the potential of CNN's in the task of recovering such lost information and retrieving original images.

Namely, we use Convolutional Neural Networks (CNN's) [4] and Convolutional AutoEncoders (CAE's) [6], combined with Inception modules [7], to generate reconstruction of images from PCA-generated compact representations.

Training on a paintings dataset extracted from Wikiart[1] [10], we show that the Convolutional AutoEncoders performed better than CNN's in terms of both mean square error and higher visual appreciation, with almost the same number of parameters, and that the use of Inception modules further improves performances. Figure 1 shows examples of images from the Wikiart used in this work.

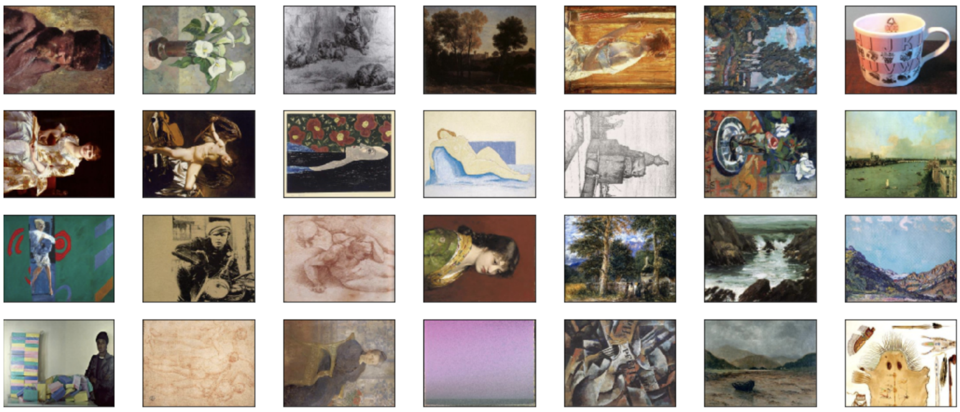

Figure 1: Examples of images from the Wikiart used in this work.

The present research is conformed as follows. In section 2, the design of the proposed solution is explained. Then, in section 3, the implementation and how the experiments were carried out are described. Section 4 contains the description of the results obtained and their analysis. It concludes in section 5.

2 Proposed Method

This section presents the method we use for dimensionality reduction of images into Principal Componentes (PC's), and the general architecture of the reconstruction module, which is based on CNN's. Figure 2 shows a diagram of the general proposed architecture.

[1]https://www.wikiart.org/

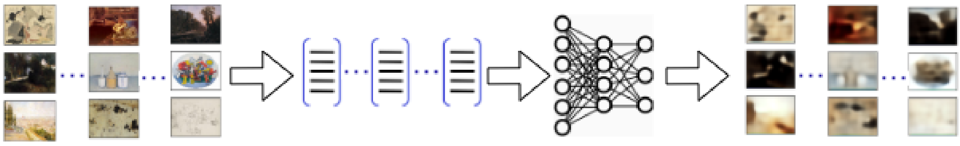

Figure 2: Proposed architecture: input images are mapped to compact representations of Principal Components, which are fed to CNN-based networks for image reconstruction.

2.1 Vectorization

The first part of the general system puts images into a standard format and size, so that they can be used to compute PC's.

First, we rescale images to $(112, 128)$ pixels. Considering a set of n RGB images (3 channels), this step produces a high-dimensional array of size $(n, 112, 128, 3)$. Later, we reshape each image into a 1-dimensional vector, this is, each image is put into a vector of 43008 elements, which again for a set of n images, corresponds to a matrix of size $(n, 43008)$.

2.2 Dimensionality reduction

Once images are put into vector format, we can proceed to compute their PC representations. For this end, a dimensionality reduction system with Principal Component Analysis was developed similarly to the eigenfaces' methodology [8] in order to define the paintings' PC space.

We selected the first 500 principal components to represent our images, which capture 95% of the dataset variance. Therefore, images are now vectors of 500 elements, which represents only 1% of their initial length.

Figure 3 shows the cumulative explained variance of the first 500 principal components after PCA transformation.

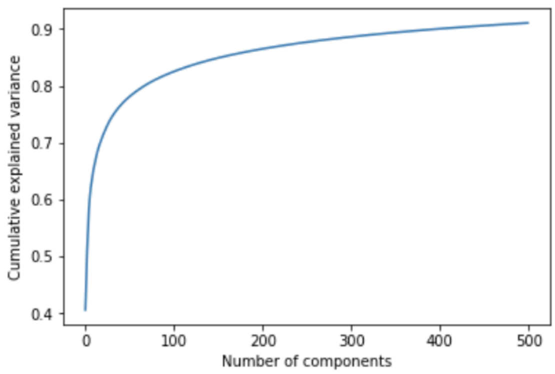

Figure 3: Cumulative explained variance of the first 500 principal components after PCA transformation.

2.3 Reconstruction architectures

Three different architectures were tested for image reconstruction: a regular CNN, a CNN using inception modules, and convolutional autoencoder.

All architectures start with fully-connected layers of 500 perceptrons, whose outputs are fed to subsequent fully-connected layers which are used to increase progressively the length of the input vector. The combinations of fully-connected layers of each architecture are illustrated in Figure 4.

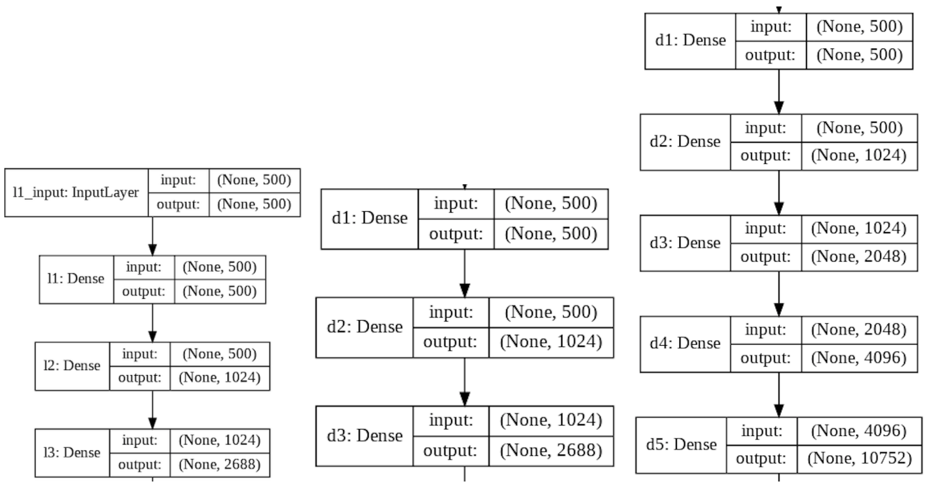

Figure 4: Fully-connected layers in each architecture. From left to right: simple CNN, CNN with inception module, and convolutional autoencoder.

Subsequently, the output of the last fully-connected layers is reshaped to obtain a three-channel matrix of size 28×32 for the CNN and CNN with inception, and of size 56×64, for the Convolutional-Autoencoder.

The first architecture consists of several modules, each of which contains a convolutional layer, followed by batch normalization and ReLU activation. The first convolutional layers have kernels of size 7×7, which decrease up to 3×3 in the latter layers. In between, there are upsampling layers to gradually reach the original image size $(112, 128)$.

The second architecture contains several modules like the one detailed above, plus an Inception layer at the end of the convolutional operations, which help improve the reconstruction by automatically selecting the more useful features. Figure 5 depicts the sections of this layer.

The third architecture extends the second one and adds a Convolutional-based implementation of the Auto-encoders with Symmetric Skip Connections [6]. This model uses bridge connections between the convolutional layers of the same level in order to speed up convergence and improve output results. It also uses separable convolutional layers which operate channel-wise as opposed to regular convolutions that integrate samples across all channels. This can be considered a highly simplified and shallow model compared to the 30 layer architecture used in [6].

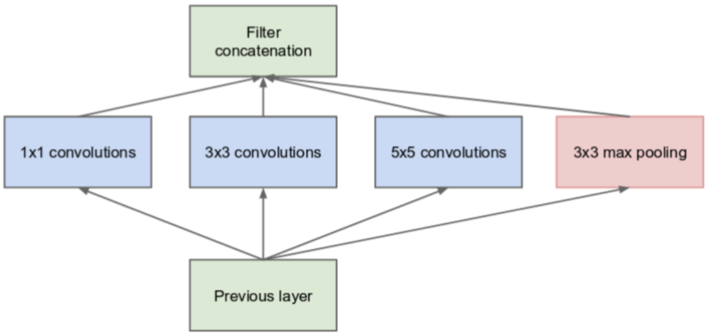

Figure 5: Inception module, naive version. Image from [7].

As a summary, for the general proposed method, a set of RGB images enter the module; these images are transformed into their principal components, with a previously PCA model fitted to the images dataset. The principal components enter one of three different neural network architectures, which transforms principal components representation back into the original image space.

3 Experiments

This section presents our experiments. First, it explains information about the data we used. It also details the architectures we evaluated, and provides implementation details.

3.1 Data

We used a dataset of paintings from a great variety of artists. With this significant variability of styles, we ensure that we have a data set in which the correlation of characteristics tends to zero.

The dataset used contains RGB images from the Wikiart repository [10]. Figure 1 shows examples of the images in our dataset. Since these images are of different sizes, we standardized them by rescaling them to fit 112×128 RGB pixels. This is, each image is a high dimensional vector of $(112, 128, 3)$ elements.

From the dataset, 16,671 paintings were randomly chosen as traning set, and 7,146 paintings were kept as test set.

3.2 Implementation details

PCA: The PCA model from sklearn was trained to generate compact representations of 500 principal components. The rescaling of the images and the number of components chosen enabled the use of the K80 GPU without exceeding its memory limit.

CNNs: The neural networks were implemented in Tensorflow and Keras. The size of the convolutional kernels was 7×7, 5×5, and 3×3 pixels; within the architecture of skip connections[6], some kernels were of size 1×1. This type of layers is often used both for dimensionality reduction and for feature selection.

The size of the kernels of the first convolutional layers was explored, we observed that 9×9 or 7×7 kernels generated a more fuzzy reconstruction compared to those of size 5×5 and 3×3.

The number of epochs in training varied between 80 to 100 epochs, with a batch size of 32. We used the Adam optimizer [9] and the mean squared error (MSE) loss. We also monitored the accuracy and mean average error (MAE) for potential additional information.

4 Results

This section presents our results. For all three architectures, loss, mean absolute error (MAE), and visual reconstruction were compared.

Sequential CNN. Of the three models, the sequential CNN was the one that obtained the worst performing loss and MAE, as we observe in Figure 6. Overall, the network manages to reconstruct the most general shapes and map back color tones close to those of the original images. However, it loses the finer details as seen when comparing Figure 7 and Figure 8.

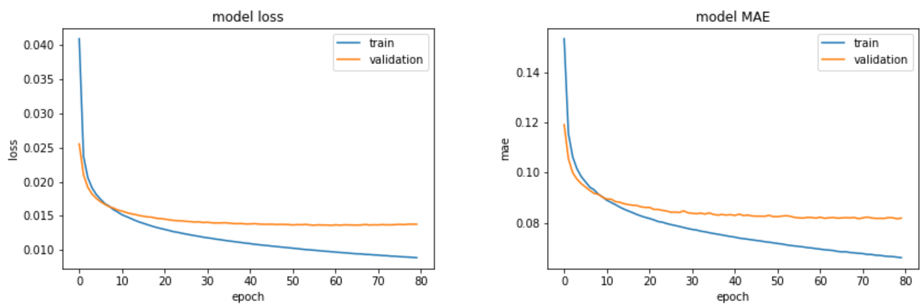

Figure 6: Loss and MAE for the sequential CNN architecture.

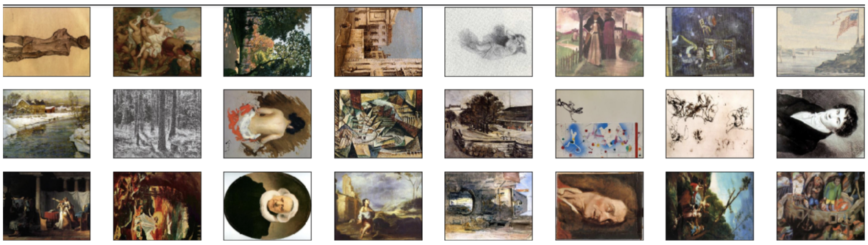

Figure 7: A random set of images fed to the sequential CNN.

Figure 8: Images reconstructed by the sequential CNN.

CNN with Inception. As shown in Figure 9, the Inception model helped obtain MSE loss and MAE below 0.007 and 0.06, respectively. Also, it is observed that this change eliminates the problem of overfitting, in comparison the the previous model (sequential CNN), whose difference between training and validation performance is about 0.005 in terms of MSE. Moreover, the CNN with inception module architecture achieves better visual results since small features of objects and textures are captured, as seen when comparing Figure 10 with Figure 11.

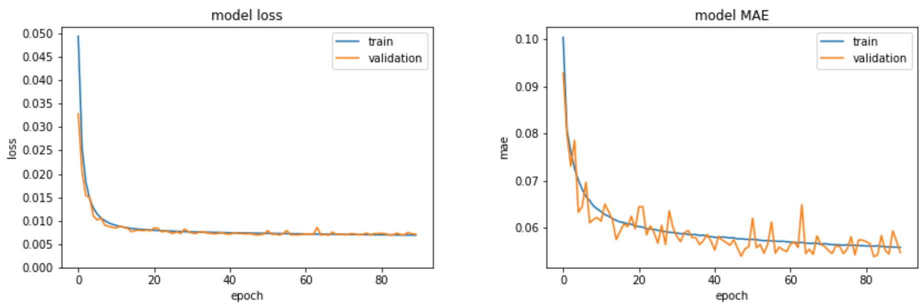

Figure 9: Loss and MAE for the CNN architecture with inception module.

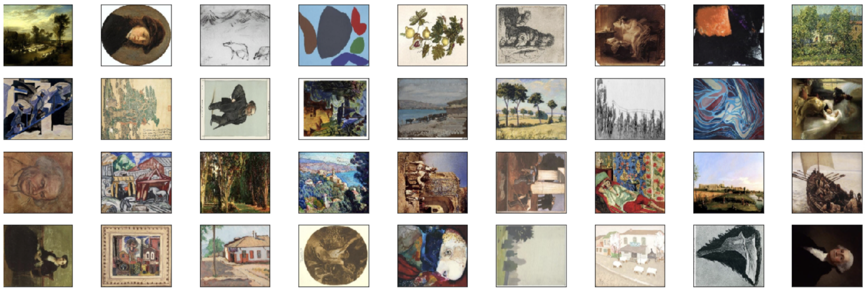

Figure 10: A random set of images fed to the CNN with inception module.

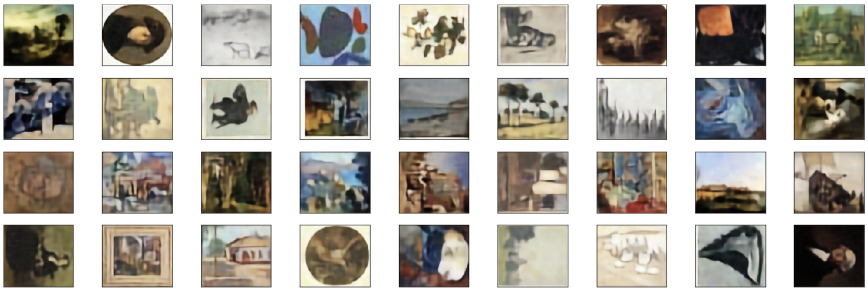

Figure 11: Images reconstructed by the CNN with inception module.

Convolutional Auto-encoder with Symmetric Skip Connections. This convolutional autoencoder (CAE) was the one that gave the least loss and MAE in the training set, as shown in Figure 12. However, we noticed that the overfitting is higher compared to the previous models. Comparing Figure 13 with Figure 14, we can see that in general, the images' details are reconstructed with colors closer to the originals. Although the previous architecture and the auto-encoder differ in terms of the performance metrics, visually, it is difficult to grasp the difference between the reconstructions.

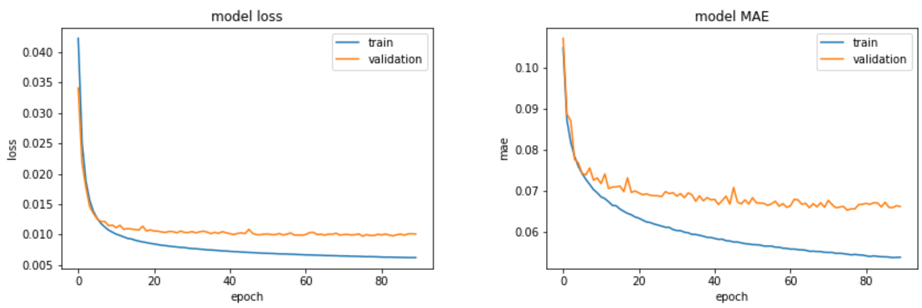

Figure 12: Loss and MAE for the Convolutional AutoEncoder architecture.

Figure 13: A random set of images fed to the CAE architecture.

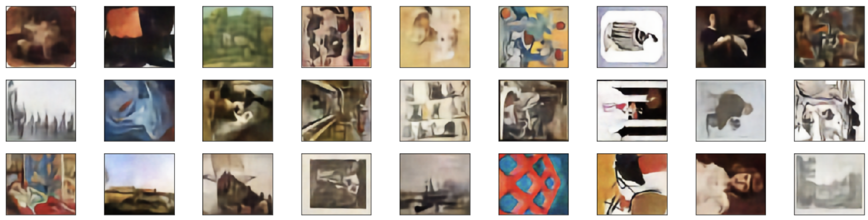

Figure 14: Images reconstructed by the sequential CAE architecture.

The inception module and the skip connections generated more faithful reconstructions to the original ones and reduced overfitting. However, there is a point where the loss, no matter how low, does not benefit the visual results.

Overall, CNN's are capable of depicting different forms and objects but fall short when recovering textures and fine details of complex artworks.

5 Conclusions

In this paper we investigate the potential of CNN models for the problem of reconstructing artistic images from compact vector representations, which are generated by principal component analysis.

We fixed the number of principal components to 500, which correspond to approximately only 1% of the initial number of features in the input images. This compact representations is able to capture the relevant statistics of the contents of the images, and CNN's are adequate models for image reconstructions from principal components representations.

We evaluated the reconstruction performance of three different CNN models: a simple sequential CNN, a CNN equipped with inception modules, and a convolutional autoencoder. From these models, we observed that the CNN with the inception module obtains the best results in terms of MSE and visual appreciation. Concretely, it achieves only 0.007 MSE, and provides fairly good reconstruction in terms of color and objects shape. Overall, CNN's are capable of depicting different forms and objects but fall short when recovering textures and fine details of complex artworks.

As future work, it is convenient to consider other loss functions focused on the image similarity and transposed convolutional layers instead of an upsampling to reconstruct minute details and texture.

Acknowledgments

We thank the support of the Asociación Mexicana de Cultura A.C., in the realization of this research work.

References

[1] Jolliffe, I. T. (2002). "Principal Component Analysis". *Springer Series in Statistics. New York: Springer-Verlag.*

[2] I. S. Bajwa and S. I. Hyder. (2005). "PCA based image classification of single-layered cloud types". *Proc. IEEE Symposium on Emerging Technologies.*

[3] Ian Goodfellow and Yoshua Bengio and Aaron Courville. (2016). "Deep Learning". *MIT Press.*

[4] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. "Backpropagation applied to handwritten zip code recognition". *Neural computation*, 1(4):541–551. 1989.

[5] Khan, A., Sohail, A., Zahoora, U., and Qureshi, A.S. (2020). "A survey of the recent architectures of deep convolutional neural networks". Artificial Intelligence Review. 53:5455–5516.

[6] Mao, X., Shen, C., Yang, Y. (2016). "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections". *Advances in Neural Information Processing Systems (NIPS).*

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. (2015). "Going deeper with convolutions". *IEEE Conf. Computer Vision and Pattern Recognition (CVPR).*

[8] M. A. Turk and A. P. Pentland. (1991). "Face recognition using eigenfaces". *IEEE Conf. Computer Vision and Pattern Recognition (CVPR).*

[9] Kingma, D. P. Ba, J. (2015). "Adam: A Method for Stochastic Optimization". *Int. Conf. Learn. Represen. (ICLR).*

[10] Kiri Nichol. (2016). "Painter by Numbers". Retrieved from https://www.kaggle.com/c/painter-by-numbers/overview.