Anime Artist Analysis based on CNN

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Artist recognition is not easy for human beings if the artworks are general and share many similarities, but modern machine learning or deep learning models may help people identify the artwork authors. In this project, we trained Convolutional Neural Networks (CNN) identifying Japanese anime artists. Our dataset was crawled from Pixiv, a well-known Japanese website containing lots of fine anime artworks. Specifically, the dataset has 9 authors and each one has 200 paintings. We trained different kinds of networks, from very basic Conv2D net to more advanced networks, in order to carry out the identification. It turned out that traditional neural networks can in some way identify different authors, but the accuracy could be improved to a better level by tuning hyper parameters or using more advanced networks such as ResNet and VGGNet.

1 Introduction

Pixiv is a Japanese online community for artists, launched on September 10, 2007, by Takahiro Kamitani. Pixiv aims to provide a place for artists to exhibit their illustrations and get feedback via a rating system and user comments. As of September 2016, the site consists of over 43 million illustration submissions.

Artist identification has very different difficulties. Some classic artworks by very famous artists like Van Gogh, Monet, or Pablo Picasso, have very distinguished artists' personal styles and painting structures. These artworks' authors may be easier for both human beings and neural networks to identify. But the problem lies in many modern arts like Japanese animes. There are great works and unique styles but many more artworks are highly similar. Even well-known comics artists draw people, scenery or still lives alike. To make things worse, many artists no longer draw in traditional ways such as pen and brush. Many artists especially young ones like sharing paintings across the internet. In order to conveniently spreading their work, they are more likely to directly draw pictures in the computer with equipments like digital panels or handwritting pads.

We want to concentrate on illustrations of Japanese anime, which have a similar style as a whole. But there are still differences between every artist, like color preference, brushwork and painting structures. Previous work on artist recognition were done by specialists in artworks, or computer scientists who had good comprehension of art, and they typically selected feature representations manually and used these features to build supervised learning models. But in our project, based on the hypothesis that every artist has unique styles and characteristics, even if differences are very latent, we trained CNNs for this problem. It is likely that CNNs can identify best representations of features as they are trained.

Also it is highly possible that simple Con2D net could not perform well and reach the required accuracy, we also consider building other CNN models like VGGNet and ResNet, which could build deeper and more efficient networks. By starting with basic Con2D net and only two author identification, we do this project step by step to make the model more complicated and add more authors to be identified.

2 Related work

Previously many work on artists identification was done by humans. The first several attempts to make machine do this task were based on manually selected features such as brush strokes, although this kind of learning algorithms aims at style identification rather than artists identification. Many image features have been used, including scale-invariant feature transforms (SIFT), histograms of oriented gradients (HOG), and more [3, 16, 19, 20]. Prior work typically used supervised learning algorithms such as SVMs to identify artist and style given these features. Other classification methods such as k-nearest neighbors and hierarchical clustering have been used as well.

Maximum Likelihood Image Identification and Restoration Based on the expectation maximum (EM) Algorithm, proposed by A.K. Katsaggelos, showed that expectation maximum algorithm is a powerful iterative procedure for computing machine learning estimates of unknown parameters involved in the likelihood function of the observed data[1]. Deep Residual Learning for Image Recognition[2], presented a residual learning framework to ease the training of networks that are substantially deeper than those used previously; this paper explicitly reformulated the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions, which provide a new idea for us.

And for image clustering, Spatial Models for Fuzzy Clustering proposed by Dzung L. Pham, proposing a novel approach to fuzzy clustering for image segmentation where a new objective function is proposed for incorporating spatial context into fuzzy C-means algorithm[3]. Image Clustering using Local Discriminant Models and Global Integration, by Yi Yang , Dong Xu, Feiping Nie, Shuicheng Yan, YueTing Zhuang, they constructing a local clique comprising the data point and its neighboring data points, and using a local discriminant model for each local clique to evaluate the clustering performance of samples within the local clique[4].

Recently, CNN has been proved to be quite efficient and accurate on many image recognition tasks. After trained under properly selected parameters, CNNs can successfully decompose artworks especially paintings into style and content components. They can even transfer style from one painting to another and there are already many fancy applications such as Prisma. This implies that CNNs can capture artwork styles.

3 Implement and Experiment

3.1 Conv2D networks

Hyper parameters include number of layers, filter size, stride and padding number, learning rate and so forth. The input tensor is fed into the model and after each layer (or sublayer) the size of output tensor could be calculated by the following equation:

$$n = m + 2p - fs + 1$$
$$(n = \frac{m + 2p - f}{s} + 1)$$

where m is the size of input tensor, n is the size of output tensor, p is the number of padding number, s is the number of stride.

In the first attempt the Conv2D network only contains 2 layers. Each layer contains 3 sublayers, which are convolutional layer, rectified linear unit layer (ReLU) and max pooling layer. Also the input training set contains 360 images as is mentioned above, with each image of size 500 x 500 x 3. The input images were first transformed into matrix representation, and then stored in a Numpy ndarray as a whole. This works well with input tensors in TensorFlow.

The first layer contains a convolutional layer of filter size equals 4 x 4 x 3, and there are 8 such filters. Other hyper parameters are: stride equals 1 and padding equals 1. Then the result is passed to an activation function, which is ReLU. Thirdly comes the max pooling sublayer, which has filter size of 8 x 8 with number of stride equals 8 and padding equals 1. After the first layer, each output tensor is of size 16 x 16 x 8.

The second layer contains the filter size equals 2 x 2 x 8. There are 16 filters each is applied to the tensor with 1 stride and 1 padding. The filter in the max pooling sublayer is of size 4 x 4, with stride equals 4 and padding equals 1. After this layer, each output tensor is of size 16 x 16 x 16.

Then each output tensor is flattened and combined in a vector. This vector is the input vector of a fully connected (FC) layer. Corresponding weights are assigned to each entry automatically in TensorFlow. The output of this FC layer is the output of the whole Con2D net.

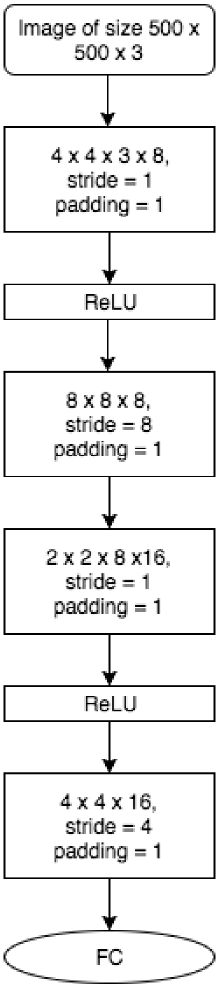
Image of size 500 x 500 x 3
4 x 4 x 3 x 8, stride = 1 padding = 1
ReLU
8 x 8 x 8, stride = 8 padding = 1
2 x 2 x 8 x16, stride = 1 padding = 1
ReLU
4 x 4 x 16, stride = 4 padding = 1
FC

Figure 1: CNN structrue.

Set the learning rate to 0.009. After running for 20 epochs, the cost converges. Training accuracy is very high because the network remembers features of every image. But the testing accuracy is only 0.556. This is very bad. It means that when encounter a new image, the network performs just a little bit better than simply ‘guess’ whose author this picture is. Also, the learning process is really slow. For Macbook Air with 1.6 GHz Intel Core i5 and 8 GB RAM, each epoch takes nearly 2 minutes.

The reason for testing accuracy being small is that the network overfits the training set. And since the whole process took a long time, so we decided to make the input images smaller and then tune the hyper parameters so that the model could fit the images better. Below is the Conv2D net flow chart for our second attempt and its cost.

The cost converges after 30 epochs and the training process is much faster. Gladly we obtained the training accuracy of up to 0.63. From this we can tell that Conv2D network can more or less extract unique features of each author, though the feature seems to be vague and implicit.

Then we tried to make the network deeper. We added a new layer to the second layer in the third attempt, but the testing accuracy result is 0.58. To our surprise, adding layer to make the network

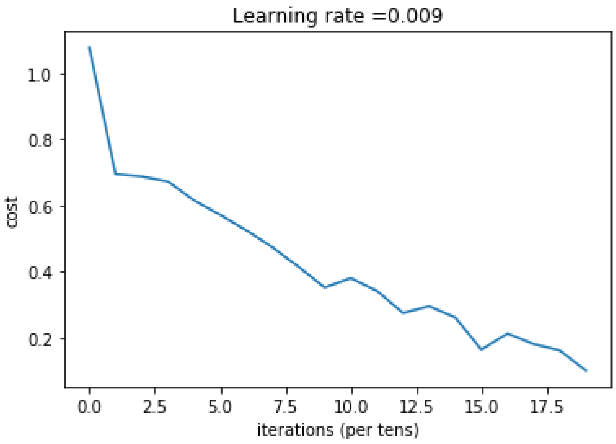

Figure 2: Result 1

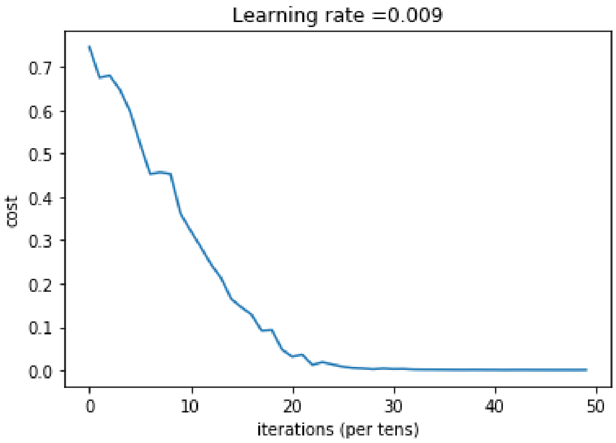

Figure 3: Result 2

deeper did not benefit the result. Later we reached accuracy of up to 0.71 for these two authors identification, and Con2D net met its bottleneck.

3.2 ResNet

4 Conclution and Future Work

4.1 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 1.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can go over 8 pages as long as the subsequent ones contain *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

[3]Z. A. M. X. Bosch, A. Image classification using rois and multiple kernel learning, 2008. [16]S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, 2006.

[19]T. E. Lombardi. The classification of style in fine-art painting. ETD Collection for Pace University, 2005.

[20]D. G. Lowe. Distinctive image features from scale-invariant keypoints., 2004.

[12] J. Jou and S. Agrawal. Artist identification for renaissance paintings.