

---

# Anime Artist Analysis based on CNN

---

**Ziyi Chen**  
Computer Science  
UC Santa Cruz  
zchen139@ucsc.edu

**Chujiao Hou**  
Computer Science  
UC Santa Cruz  
chou8@ucsc.edu

**Xinyuan Ma**  
Computer Engineering  
UC Santa Cruz  
xma34@ucsc.edu

**Weicai Lin**  
Electrical Engineering  
UC Santa Cruz  
wlin42@ucsc.edu

## Abstract

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

## 12 1 Introduction

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

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

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

34 Also it is highly possible that simple Con2D net could not perform well and reach the required  
35 accuracy, we also consider building other CNN models like VGGNet and ResNet, which could  
36 build deeper and more efficient networks. By starting with basic Con2D net and only two author

37 identification, we do this project step by step to make the model more complicated and add more  
38 authors to be identified.

## 39 **2 Related work**

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

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

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

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

## 67 **3 Dataset**

### 68 **3.1 Crawling**

69 The image dataset is crawled from Pixiv (<https://www.pixiv.net/>). We first use a chrome  
70 extension called Pxer, which can be used under the webpage of a certain artist and crawls all image  
71 url of the artist. We randomly select 16 artrists who have uploaded more than 200 illustrations and  
72 use Pxer to get url. The we tried to use downloader like Thunder to batch download pictures, but  
73 it failed. Beacuse Pixiv forbids to access the image directly using the url without refer in the http  
74 head. Then we used Python and imitated the http head to crawl illustrations. We have more than 5000  
75 origin images from 16 different authors. Samples of illustrations is shown in Figure 1.

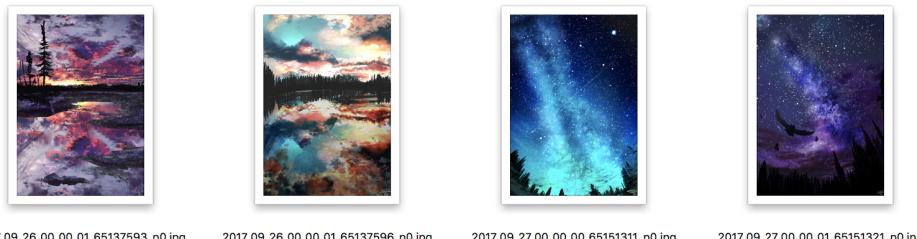


Figure 1: Samples of illustrations from Pixiv.

76 **3.2 Image Proprecessing**

77 Compressing and cropping images is an important step in image preprocessing. The size of images  
78 varies from 50 kilobytes to 3 megabytes. We first deleted the pictures without characteristics of  
79 painter (such as blank pages, advertisements, etc.). We used two different ways to proprecess the  
80 image.

81 The first is to rescale image. Through OpenCV, we compressed the picture to the size of  $500 \times 500$   
82 pixels. After that we manually selected the picture which is obviously distorted after being compressed  
83 and then deleted them. But it may lose the pixel information of images.

84 The second wat is to crop image from center as shown in 2. Since some iamge is quite large and the  
85 important information of image is more likely to be in the center of image. This way could use the  
86 information more efficiently. Also the image Pure white and pure black images are removed. We  
87 previous only have around 300 images. After cropping, we could have more than 3000 images.



Figure 2: Samples of cropping the image.

88 **4 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:

$$n = m + 2p - fs + 1$$

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

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

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

96 The first layer contains a convolutional layer of filter size equals  $4 \times 4 \times 3$ , and there are 8 such filters.  
97 Other hyper parameters are: stride equals 1 and padding equals 1. Then the result is passed to an  
98 activation function, which is ReLU. Thirdly comes the max pooling sublayer, which has filter size of  
99  $8 \times 8$  with number of stride equals 8 and padding equals 1. After the first layer, each output tensor is  
100 of size  $16 \times 16 \times 8$ .

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

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

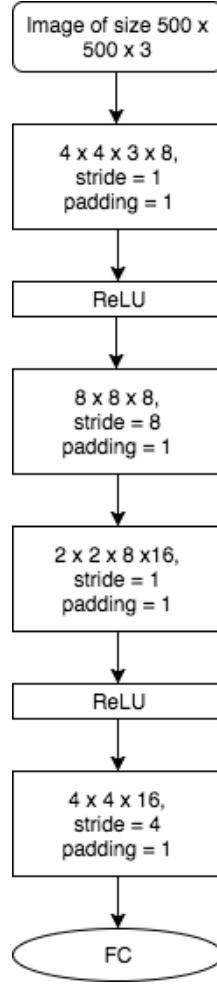


Figure 3: CNN structure.

- 107 Set the learning rate to 0.009. After running for 20 epochs, the cost converges. Training accuracy is  
 108 very high because the network remembers features of every image. But the testing accuracy is only  
 109 0.556. This is very bad. It means that when encounter a new image, the network performs just a little  
 110 bit better than simply ‘guess’ whose author this picture is. Also, the learning process is really slow.  
 111 For Macbook Air with 1.6 GHz Intel Core i5 and 8 GB RAM, each epoch takes nearly 2 minutes.
- 112 The reason for testing accuracy being small is that the network overfits the training set. And since the  
 113 whole process took a long time, so we decided to make the input images smaller and then tune the  
 114 hyper parameters so that the model could fit the images better. Below is the Conv2D net flow chart  
 115 for our second attempt and its cost.
- 116 The cost converges after 30 epochs and the training process is much faster. Gladly we obtained the  
 117 training accuracy of up to 0.63. From this we can tell that Conv2D network can more or less extract  
 118 unique features of each author, though the feature seems to be vague and implicit.
- 119 Then we tried to make the network deeper. We added a new layer to the second layer in the third  
 120 attempt, but the testing accuracy result is 0.58. To our surprise, adding layer to make the network  
 121 deeper did not benefit the result. Later we reached accuracy of up to 0.71 for these two authors  
 122 identification, and Con2D net met its bottleneck.

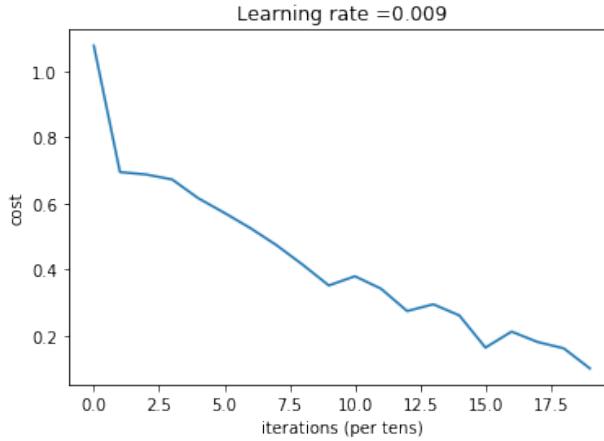


Figure 4: Result 1

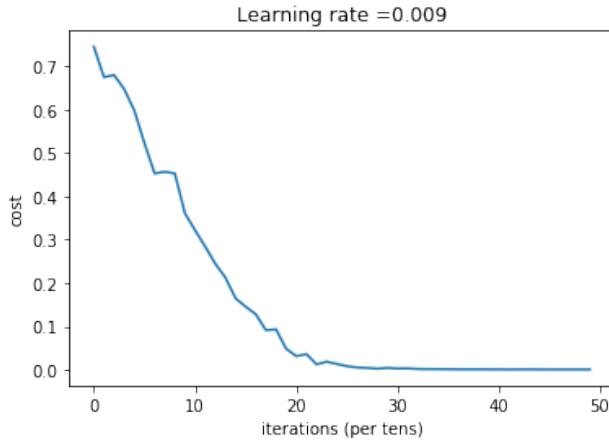


Figure 5: Result 2

## 123 5 ResNet

### 124 5.1 Implement

125 Deeper neural networks are more difficult to train. ResNet present a residual learning framework  
 126 to ease the training of networks that are substantially deeper than those used previously. ResNet  
 127 explicitly reformulate the layers as learning residual functions with reference to the layer inputs,  
 128 instead of learning unreference functions. ResNets use residual blocks to ensure that upstream  
 129 gradients are propagated to lower network layers, aiding in optimization convergence in deep networks  
 130 [7].

131 Our second network is the ResNet-14 network with fullyconnected layer. The model is trained from  
 132 scratch to learn features solely from our dataset. The network architecture is shown in Figure 6. We  
 133 used the 14-layer version of ResNet for faster training and decreasing the memory usage due to the  
 134 limit of our computer.

### 135 5.2 Experiment

136 We first use  $250 \times 250 \times 3$  size image from 9 different artists (each artist contributes 700 images) to  
 137 train the model. The result is shown in 7. With the increasing of training steps, the training accuracy  
 138 increases until 1. But the test accuracy is around 28% finally. The overfit might due to the lack of  
 139 images or the complexity of the model.

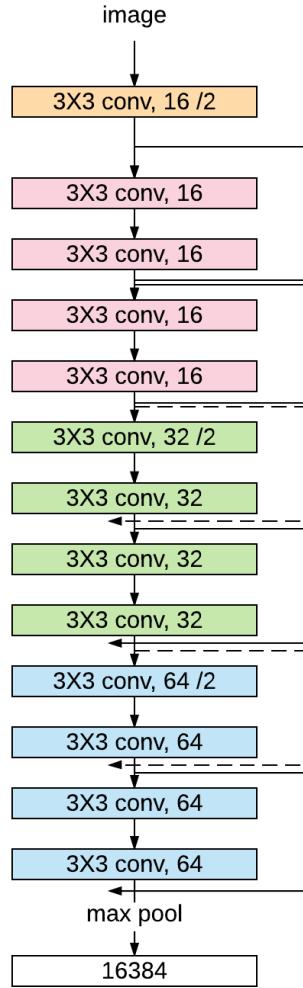


Figure 6: Result 2

140 Then we try  $100 \times 100 \times 3$  size image from 9 different artists (each artist contributes 4000 images)  
 141 and delete 4 layers before max pool. But the result is not better. It might because we didn't remove  
 142 the images that only contains same color and no other information. Since most of the image is drawn  
 143 directly in the computer, there are many pure color cropped images.

144 **6 Conclusion and Future Work**

145 **6.1 Tables**

- 146 All tables must be centered, neat, clean and legible. The table number and title always appear before  
 147 the table. See Table 1.
- 148 Place one line space before the table title, one line space after the table title, and one line space after  
 149 the table. The table title must be lower case (except for first word and proper nouns); tables are  
 150 numbered consecutively.
- 151 Note that publication-quality tables *do not contain vertical rules*. We strongly suggest the use of the  
 152 booktabs package, which allows for typesetting high-quality, professional tables:

153 <https://www.ctan.org/pkg/booktabs>

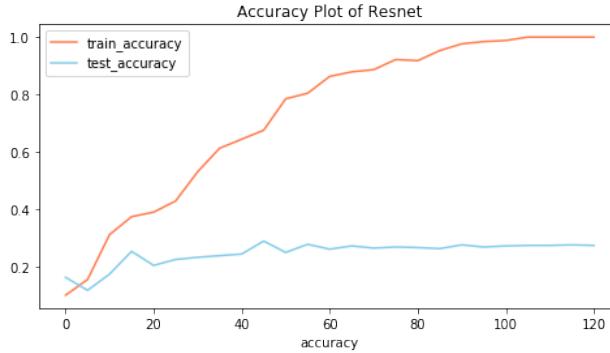


Figure 7: ResNet Accuracy

Table 1: Sample table title

Part		
Name	Description	Size ( $\mu\text{m}$ )
Dendrite	Input terminal	$\sim 100$
Axon	Output terminal	$\sim 10$
Soma	Cell body	up to $10^6$

154 This package was used to typeset Table 1.

## 155 7 Conclusion and Future Work

### 156 7.1 Conclusion

157 Based on our implementation and the attempt of hyper-parameters and different network, for Convolution  
 158 Neural Network, the accuracy for two classes' identification is 0.56. But for Resnet, for the same  
 159 classes' identification is also 0.559 which is only a little bit higher than the normal CNN network.  
 160 But we test for the different classes' and found that the range of the accuracy for different classes'  
 161 identification differs a lot, range from 0.559 - 0.957. The above results means that the accuracy  
 162 depends very much on the chosen of authors. There are several reasons that the performance of the  
 163 system is not good at all, one of the reason is that the dataset is quite small, for every authors we  
 164 only have 200 pictures, this limited size of the dataset makes the work harder. So we divide every  
 165 picture into four parts, trying to increase the number of the dataset, this helps a little to the system,  
 166 the accuracy for two classes' identification raises from 0.56 to 0.625. Another reason is most figures  
 167 under Japanese artist looks the very similar, big eye and small faces lolita with no obvious character's  
 168 distinct features, let alone with that the images losses the brushes and strokes' characteristic due to  
 169 that some authors prefer draw electronically. So Frankly speaking, the Anime Artist Analysis is a  
 170 hard topic.

### 171 7.2 Future Work

172 Based on the low accuracy of the results from both traditional Convolution Neural Network and  
 173 Residential Network, there are several ways that we think up to improve the performance of our  
 174 identification and classification system.

175 So first, the most straightforward way is to test other hyper-parameters. And then we'd like to try to  
 176 train the system by VGG network. VGG net always set the filter size as  $3 \times 3$ , and used  $2 \times 2$  pooling  
 177 size, and doubling the the number of filters after each pooling through out the whole network. VGG  
 178 is a network that indicates that the deeper the better, so it will be slow to train the data. Actually this  
 179 is already an on-going process, we started training our images by VGG net but it seems like VGG  
 180 network has a high requirement on the hardware, so based on our laptop it seems really difficult to  
 181 train to images and finished the process.

182 Another things that we could do is to increase our dataset size, crawling more images so that letting  
183 the system learn more.

184 **Acknowledgments**

185 This research was creative due to Machine Learning course taught by Manfred K. Warmuth, Professor  
186 in Computer Science Department at University of California, Santa Cruz. I want to thank the  
187 professor for giving us an opportunity to implement the project, as well as imparting so many  
188 interesting knowledge. And I also want to thanks the teaching assistant of the course, Ehsan Amid  
189 and Tianyi Luo, who give us a lot of useful advices.

190 **References**

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

- 195 [1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In  
196 G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp.  
197 609–616. Cambridge, MA: MIT Press.
- 198 [2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the*  
199 *GEneral NEural SImulation System*. New York: TELOS/Springer-Verlag.
- 200 [3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent  
201 synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.
- 202 [3] Z. A. M. X. Bosch, A. Image classification using rois and multiple kernel learning, 2008. [16]S. Lazebnik,  
203 C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene  
204 categories, 2006.
- 205 [7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint,  
206 abs/1512.03385, 2015.
- 207 [19]T. E. Lombardi. The classification of style in fine-art painting. ETD Collection for Pace University, 2005.
- 208 [20]D. G. Lowe. Distinctive image features from scale-invariant keypoints., 2004.
- 209 [12] J. Jou and S. Agrawal. Artist identification for renaissance paintings.