Convolutional Neural Network for Forgery Painting Identification

Guangju Wang
*Department of Industrial Engineering and Logistics Management
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
gwangal@connect.ust.hk

Abstract

This report presents an experiment of forgery identification with help of Convolutional Neural Network. Convolutional Neural Network is known to be powerful in field of image recognition. However, to conquer forgery identification is another challenge, as forgeries may be very similar to the authentic ones. In our problem, we have 11 authentic and 9 forgeries that can be used for training. To get a Neural Network that serves our purpose, I first extracted features from labeled paintings. And then used those features to train a Convolutional Neural Network. Two labeled paintings are reserved for validation purpose. Finally, we use our fitted model to determine whether the unlabeled paintings are authentic or forgery.

1 Background and Overview

1.1 Data description

Dataset "Raphael's Painting" contains 28 paintings in total, of which 12 are known to be authentic, 9 known to be forgeries, and 7 other paintings are disputed. Painting number 28 is too large for be read by python and is thus left out. Two paintings, one authentic and one forgery are randomly selected to be validation data. Our ultimate purpose is to get a math model that can evaluate to what degree a known painting is authentic of Raphael.

For a better description of data, please refer to section 3 of:

`http://math.stanford.edu/~yuany/course/2017.spring/project3.pdf`

1.2 Overview of the Report

Section 2 details how the picture data, which came in different format, was manipulated and how do we extract features from limited labeled data. Section 3 describes how the models was built and compares Multi-Layer Perceptron to Convolutional Neural Network. The training and validation methods are also detailed in Section 3. At the end of the section, I used the built model to get a prediction on the unlabeled paintings. Section 4 concludes the report.

1.3 Tools used

All analysis are done in Python 2, with the help of package 'scipy' and the well known Machine Learning package 'tensorflow'.(https://www.tensorflow.org).

*The dataset is provided by Prof.Yang WANG and is used by Prof. Yuan YAO for education purpose. This report is only for grading purpose of class MATH 6380, held by Prof.Yuan Yao, at the Hong Kong University of Science and Technology.

2 Data Clean up

2.1 Resolution, color code, and file type

There are several challenges about feature extraction. Firstly, the resolution of different pictures might be different. With a limited dataset, most likely we are going to extract features by cutting big pictures into small ones. Strictly speaking, we want all small parts be equal in size. Secondly, the files are provided in different color code. Without pretreatment, we cannot feed them into the same neural network. It is even more impossible to train multiple networks for different data format, as most authentic is provided in .tif format and all others in .jpg. Before we proceed to feature extraction, we have to solve the three problems.

All paintings are provided in digital copy. That being said, resolution of different paintings may be different. However, there is not much we can do to recover the true resolution. So that throughout the report, we are going to neglect the effect of different resolution.

The problem of file type and color code is easily solved by the highly integrated function 'scipy.misc.imread' and noticing the fact that fourth band of all color codes are 255.

To sum up, I read the file data by using 'scipy.misc.imread' to read all the painting files, and straight cut off the fourth band of all four bands color code. As a result, we have all pictures read in numpy arrays that have dimensions like 'height by width by 3'. But still they are in different size and some pictures contains the frame of the painting or write borders.

2.2 Image cutting and sampling

As said, for training purpose, we have to cut any parts that are not part of the paintings. This is done by manually look through all paintings and find right cutting points by trial-and-error. Below is an example.

the original picture(#11) looks like:

Through trial and error, we find the proper cutting points [50: 4680, 70: 2900] and the resulting picture looks like:

With all necessary cleaning work done, I tried to define what might be a good feature for this problem. In my opinion, the architecture of the whole paintings will be more than difficult to be analyzed by compute. In contrast, small parts of the painting may be a good representation of the author's drawing habit and his use of drawing materials. I further suppose that any part of a painting will be a good representation. If these assumptions holds, we can fix the size of our sample, and let the location of the sample to be random. For example, if we target to get some 50 by 50 small pictures from a 1000 by 1200, we can choose []0:50, 0:50] and [3:53, 1000:1050]. In such we, we can get almost as much samples as we want.

Note that in the above method, there might be overlap between different samples. One may argue that the samples may be correlated with each other. In fact, if we are trying to get multiple samples some one picture, the samples are always going to be correlated in some way. In some sense, making the location of the small parts random kind of make the samples less correlated.

3 Prediction models

3.1 Multi-Layer Perceptron – Why is it not good in our case

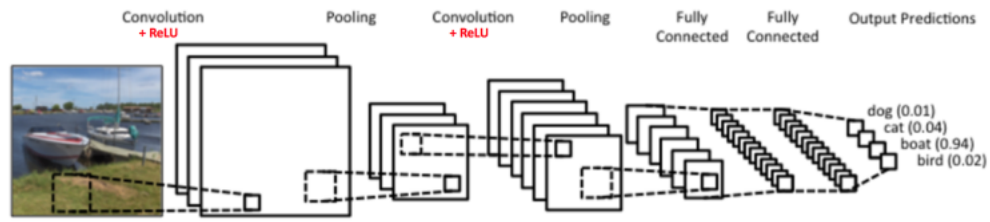
To warm up, I tried to use Multi-Layer Perceptron as the model and treat the problem like a pure regression problem. However, the performance of MLP differs little from that of random guessing. With a little bit more research, I will try to explain why Convolutional Neural Network may outperform MLP in this section.

In MLP, the structure of the data is not considered. To feed a sample to the network, I had to flat the sample into a 1-d array like:

$$(x_1, y_1, z_1, x_2, y_2, z_2, ...)$$

Then everything goes to the network in parallel.

However, in image recognition, the relationship between nearby pixels are sometimes more important than the pixel itself. Convolutional Neural Network, in contrast, read small parts of the picture as a whole and thus is more capable of discovering the relationship and pattern of the picture. Below is an illustration of Convolutional Neural Network.



3.2 Convolutional Neural Network

The training of Convolutional Neural Network is computationally intense. To do the job, most of the time we are going to need the help of GPUs. Fortunately, my supervisor provided me with a GPU server that can do the job.

With little experience of tuning Neural Networks, most of my tuning was done by manually choose the value, check the result and then adjust the value. The final model takes any 100*100 3 band colored pictures as input and return a float number that tells us how likely is it authentic or forgery, with 1 indicating authentic.

The Final model contains two convolutional layers, each take a 10*10 picture. The first layers turn each pixel from 3 channel to 8, and the second layer extend it further to 32 channels. The convolutional layers are then followed by a fully-connected layer and a final output layer. The fully-connected layer produces 128 features for each input, which are finally turned into output.

Compared to Convolutional Neural Network in real use, my model has far less layers and parameters. However, shortly we will see that the result of such model still outperform other models.

I will not attempt to explain the model further due to my limited knowledge of Neural Networks. Interested parties can find my code in the attachment.

3.3 Training and Validation

Training samples are generated by randomly selecting 100 by 100 pictures from the labeled paintings and then attached with proper labels. Models are then validated on the reserved validating sample. When the validating error stays below some threshold, the model is thought to be converged and the algorithm terminates.

3.4 Performance

As the training data is randomly drawn from labeled paintings, there is no guarantee that the model performs good on the training set. So in this problem, the performance of the model is measured by its performance on the training set and the validation set.

The result shows that our model identifies most forgeries correctly. However, The authentic works are not identified with high confidence and accuracy.

Of all the disputed paintings, three are marked as authentic. As the results indicate, the algorithm seldom classify forgeries as authentic. Then with high confidence, we can conclude that three disputed paintings , which are labeled 1, 20 and 26,are likely to be authentic. Paintings #7 and #25 are also likely to be authentic. However, Paintings #10 and #23 are not very likely to be authentic.

```python
for i in range (9):
    validate_disputed('Rapheal', i)
```

```
Picture 0 Probably FAKE
Picture 1 Probably Genuine
Picture 2 Genuine
Picture 3 Probably Genuine
Picture 4 Probably FAKE
Picture 5 Genuine
Picture 6 Probably FAKE
Picture 7 Probably Genuine
Picture 8 Genuine
```

```python
for i in range(7):
    validate_disputed('notRapheal', i)
```

```
Picture 0 FAKE
Picture 1 Probably FAKE
Picture 2 FAKE
Picture 3 FAKE
Picture 4 FAKE
Picture 5 Probably FAKE
Picture 6 Probably FAKE
```

```python
for i in range (7):
    validate_disputed('Disputed', i)
```

```
Picture 0 Genuine
Picture 1 Probably Genuine
Picture 2 Probably FAKE
Picture 3 Genuine
Picture 4 Probably FAKE
Picture 5 Probably Genuine
Picture 6 Genuine
```

4 Conclusion and Future Work

In this report, I built Convolutional Neural Network model to identify whether a painting is Raphael's authentic or forgery. I used fix-sized random sampling to extract large set of training data. Then I trained a 4 layer Convolutional Neural Network based on the data extracted. With limited experience of Neural Network, my model is lack of parameter tuning and thus the model presented may not be the best model that could be built. However, with some efforts, the model did return descent result.

The findings of this report can be summarized as:

1. Consistence in data format is critical to Machine Learning.
2. Convolutional Neural Networks considers the correlation between nearby pixels and thus is more suitable for image recognition task.
3. Parameter tuning is crucial to the performance of model.

Acknowledgments

The author is thankful to Prof. Yuan Yao for the nicely prepared lectures and also for providing the datasets and hosting Kaggle competition. Without his help, all the experiments and discussions above would not be possible. This report should not be used for publication purpose or distributed outside of MATH 6380 class. Any enquiries, please get in touch with the author(gwangal@connect.ust.hk).