Identification of Raphael's Paintings from the Forgeries

Group memeber: Ye Rougang, Tan Chunxi, Han Ruijian
December 11, 2017

1 INTRODUCTION

In this project, we select the topic of identifying Raphael's paintings, of which the dataset provided by Prof.Yang Wang consists of high resolution scans of 28 paintings. These painting scans have been scaled to a uniform density of 200 dots per painted inch and the picture size varies from 1192*748 to 6326*4457 pixels. Among the total 28 paintings, 12 paintings have been attributed to the label "Raphael", 9 have been known to be "Non-Raphael", and others are currently questioned by experts.
In this paper, we introduce how we cope with the given dataset and then conclude experiment results. In easier studying, we transform them into gray-scale scans. Competitively, we use two different methods to get the identification. The first approach is Converlutional Neural Network (CNN), the most-used image model. For the second way, we apply AutoEncoder in featuring capturing and make a classification.

2 CNN MODEL

2.1 METHOD

For our dataset, we crop original picture to $256 * 256$ smaller pictures. A picture after cropped cannot match the size with other pictures, so we remove this one in our dataset and then have 11 "Raphael" of 27 pictures in total. Under this processing, we have nearly 6000 sample which guarantees enough sample in training.
A convolutional neural network is feed-forward artificial networks that has a widely use in the image processing and pattern recognition. Inspired by tight frame method with moment

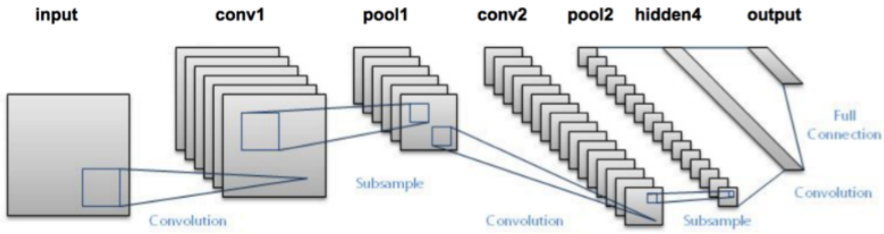

Figure 2.1: The architecture of CNN

statistics in the previous work with the dataset, using CNN may be another competitive approach.
In our model, we pass an input image to the first convolutional layer. The convoluted output is obtained as an activation map. The filters applied in the convolution layer extract relevant features from the input image to pass further. In this way, each filter shall give a different feature to aid the correct class prediction. In case we need to retain the size of the image, pooling layers are then added to further reduce the number of parameters.
In total, two convolution and pooling layers are added before the prediction is made. Convolutional layer help in extracting features. As we go deeper in the network more specific features are extracted as compared to a shallow network where the features extracted are more generic. The output layer in a CNN as mentioned previously is a fully connected layer, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network. Then, we use multilayer perceptron to make the classification.
In the learning process, cross-entropy loss function is defined in the fully connected output layer to compute the mean square loss of target and output. The gradient of error is then calculated with SGD.Adam optimizer with learning rate = 0.00005. Then, the error is back-propagated to update the filter(weights) and bias values.

2.2 Result and Analysis

In the evaluation, we apply leave-one-out cross-validation(CV) procedure to measure the accuracy of our models. All images have been cropped different amounts of smaller pictures with same size and the output of our model have assigned every small image a variable which is constraint to be 0 or 1, indicating the classification for this input is "Non-Raphael" or "Raphael" respectively. We are given a percentage of assigning to be "Raphael" for one image. That means the percent reaches up tp 100% if its all the small images have been classified into "Raphael".
In a word, we expect all known Raphael image have a percent over 50%, otherwise all "Non-Raphael" below 50%. Then we can guarantee the accuracy of our model. Two tables are given below, showing percentage results for known 11 of "Raphael", 9 of "Non-Raphael" paintings. From tables 2.1, 2.2, our model make 18 right classification out of 20. To this extend, we have a

Index of "Raphael" Paings	# Cropped Pictures	Percentage of being "Raphael"(%)
2	110	100
21	210	72
22	240	62
24	160	71
27	100	62
3	170	60
4	300	71
5	410	95
6	300	54
8	430	63
9	280	80

Table 2.1: Percentage of known "Raphael" Paintings to be "Raphael"

Index of "Non-Raphael" Paings	# Cropped Pictures	Percentage of being "Raphael"(%)
11	200	35
12	200	30
13	330	31
14	200	10
15	320	51
16	210	22
17	200	28
18	430	28
19	120	64

Table 2.2: Percentage of known "Non-Raphael" Paintings to be "Raphael"

Index of "Uncertain" Paings	# Cropped Pictures	Percentage of being "Raphael"(%)
1	127	67
10	285	46
20	420	67
23	12	58
25	48	54
26	121	46
7	299	51

Table 2.3: Percentage of "Uncertain" Paintings to be "Raphael"

relatively high probability to give a right classification for uncertain paintings. Table 2.3 is the result table for 7 uncertain paintings. From table 2.3, we find the picture 1, 20, 23, 25 and 7 has high probability (larger than 0.5) to be the painting of "Raphael", while the other picture may not be the painting of "Raphael".

3 Autoencoder and Logistic Regression

3.1 Feature Learning with Autoencoders

Besides CNN model, we choose to use the autoencoder to further capture features of paintings. Then we make a prediction by logistic regression based on features captured.
An autoencoder neural network is an unsupervised learning algorithm that applies back propagation, setting the target values to be equal to the inputs. There are two parts in the Autoencoder: encoding and decoding. Encoder constructs the network so that it maps input data to a lower dimensional, compressed feature representation, while decoder maps the feature representation back into the input data space. The aim of the autoencoder in our design is to learn a feature representation for a set of pixels of paintings.
Same as CNN method, we preprocessed data in the beginning. After transforming paintings to greyscale images, we split each painting into patches of $256*256$ pixels to enlarge sample size and unify the input size to make it convenient for the training of the autoencoder. In this way, we get the $2716*1*256*256$ training tensor for autoencoder of Raphael paintings. Also we get the $2213*1*256*256$ and $1313*1*256*256$ training tensors for the autoencoder of Non-Raphael paintings and Disputed Raphael paintings respectively.
Through tuning parameters and structure of the three autoencoders, we use the sigmoid activation function between the input layer and the first hidden layer and the ReLu activation function between the last hidden layer and the output layer for all the three autoencoders. We set the dimension of the middle hidden layer to be 10, which is also the input of decoder. That means we can take 10 features to carry on further logistic regression. In the training, we take MSE as the loss function and choose 10 batches once lasting for 3 epochs.

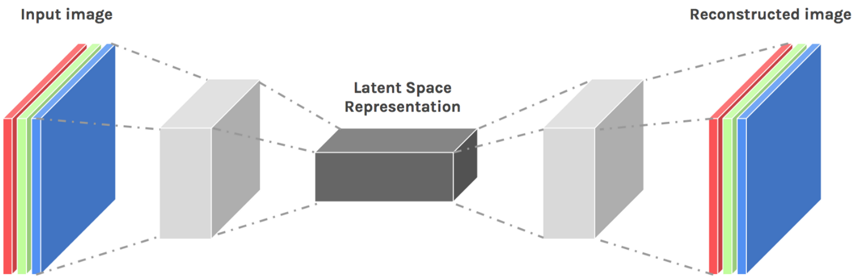

Figure 3.1: The architecture of Autoencoder

Index of "Uncertain" Paings	# Cropped Pictures	Percentage of being "Raphael"(%)
1	127	69
10	285	67
20	420	70
23	12	83
25	48	67
26	121	72
7	299	72

Table 3.1: Percentage of "Uncertain" Paintings to be "Raphael"

3.2 Logistic Regression

Then we used the logistic regression to predict the seven disputed paintings after extracting 10 features for all of patches of different images. Also we used leave-one-out cross validation with the logistic regression to validate the feature extraction, i.e. choose one painting as the testing data and the remaining 19 paintings as the training data to train the logistic regression. The classification accuracy for the leave-one-out cross validation is surprisingly 100% which means the logistic regression can predict correctly no matter for Raphael paintings or Non-Raphael paintings.Then based on these features, using features of Raphael and Non-Raphael to train the logistic regression, we get the predicted probabilities for the 7 disputed paintings shown in table 3.1. From table 3.1, we find all of disputed pictures tend to be (with probabilities larger than 0.5) paintings of "Raphael" by this method.

4 Summary

Given the 28 image dataset, we firstly preprocess data: transform to gray-scale and crop into same size. Then we take two methods to identify the uncertain paintings. On the one side, we apply CNN and MLP to do classification. On the other side, we take autoencoder as way to capture features and make logistic regression on uncertain paintings. Results of this two show above. In the CNN, we train both "Raphael" with label 1 and "non-Raphael" with label 0

together. Then, we apply the model on the "disputed-Raphael" data and give the conclusion. We find the conclusion is not as we expected, since picture 20 may be drew by data-builder but our results tell us the picture 20 is drew by "Raphael". This problem may be caused by the spilt of original figure. With the split by 256∗ 256 figure, we may loss some information.
In the autoencoder, we train 3 autoencoders with three datasets – "Raphael", "non-Raphael" and "disputed-Raphael". We choose 10 dimensional space as our feature space. With the feature space, we apply logistic regression to make the prediction on "disputed-Raphael". The cross-validation is perfect in this approach but results tells us that almost "disputed-Raphael" should be the paintings of "Raphael". This phenomenon may be caused by the use of logistic model to make classification. In the future, we will try more different models to make classification. If we have more enough time, we would like to try the BIGAN model to extract features, which is a powerful model combining GAN, autoencoder and CNN together.

5 ACKNOWLEDGMENT

Han Ruijian used the CNN model to transform the problem and get the classification by Neural Network.
Ye Rougang preprocessed the dataset and used the Autoencoder method to get features and make prediction by logistic regression.
Tan Chunxi processed data and summarized results of different methods.

Codes and results can be found in the attachments.