MATH6380O Mini-Project 1
Image Classification with Extracted Feature

ZHANG Jianhui, ZHANG Hongming, ZHU Weizhi, FAN Min

Hong Kong University of Science and Technology

March 13, 2018
Outline

1. Preprocessing data

2. Feature Extraction
   - Scattering Net
   - ResNet-50

3. Classification

4. Discussion
Outline

1. Preprocessing data
2. Feature Extraction
   - Scattering Net
   - ResNet-50
3. Classification
4. Discussion
More Samples from Limited Paintings

Original data: 28 paintings totally, 12 genuine, 9 fake, 7 unknown. Crop more samples from one single painting, that is, we crop 200 samples with 224*224 size.

Random cropping does not work well.

Figure: Painting No.9
We randomly crop 5000 samples from each paintings and compute their variance, whereby we get empirical distribution of variances. Small variance may represent empty sample. We could set a proper threshold to distinguish meaningful samples from empty ones.
Variance Threshold

First, for every painting we set 90\textsuperscript{th} percentile as variance threshold to select samples cropped randomly. But it does not work well. Then we pre-crop paintings with edges like No. 18 and then crop them randomly with variance threshold.

Figure: Samples from No. 18

Figure: Samples from pre-cropped No. 18

Figure: Painting No. 18
Choose a Proper Variance Threshold

If we choose large variance threshold, we could only crop samples from small area. How to choose a proper variance threshold? In project, we simply choose 45\textsuperscript{th} percentile as variance threshold.

Figure: Samples with 45\textsuperscript{th} percentile from No. 18
Outline

1. Preprocessing data
2. Feature Extraction
   - Scattering Net
   - ResNet-50
3. Classification
4. Discussion
Parameters of Scattering Net

We use the package ScatNet 2.0 from ENS.

Parameter setting

- `filt_opt.J = 5`, the number of scale of wavelets (high pass filters)
- `filt_opt.L = 6`, the number of orientations
- `scat_opt.M = 3`, the maximum scattering order (layers of scatter net)

Samples are RGB small images. We implement scattering net on each channel and then concatenate transformed feature together as a single vector.
RestNet 50

- We use a pre-trained ResNet-50 model on Image Net trained by Tensor flow.
- Data pre-processing.
- Remove last layer and use the output as feature.
Outline

1. Preprocessing data

2. Feature Extraction
   - Scattering Net
   - ResNet-50

3. Classification

4. Discussion
Classification Methods

Use features extracted by CNN and Scatter Network

- Linear Regression
- SVM
- KNN

Fine tune ResNet 50
Results on extracted features

Leave one out scheme for testing

<table>
<thead>
<tr>
<th>Features</th>
<th>Classifier</th>
<th>Acc on sample</th>
<th>Acc on painting</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>SVM</td>
<td>66.7%</td>
<td>65.8%</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>71.4%</td>
<td>65.1%</td>
</tr>
<tr>
<td></td>
<td>Logistic Regression</td>
<td>81.0%</td>
<td>71.4%</td>
</tr>
<tr>
<td>Scatter Network</td>
<td>SVM</td>
<td>77.3%</td>
<td>80.9%</td>
</tr>
</tbody>
</table>
Data splitting for ResNet 50

- Pre-select samples of 7 paintings as test set.
- Pre-select 20% in training set as validation set
  - One concern is that features of samples from the same painting are similar.
- Trained on a 1080ti GPU.
Classification Results

- Accuracy on samples: 84.28%
- Accuracy on paintings: 85.71% (6/7)
- Voting result:

<table>
<thead>
<tr>
<th>Painting</th>
<th>#1 Not</th>
<th>#2 Not</th>
<th>#3 Yes</th>
<th>#4 Yes</th>
<th>#5 Yes</th>
<th>#6 Yes</th>
<th>#7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voting Acc</td>
<td>199/200</td>
<td>200/200</td>
<td>200/200</td>
<td>200/200</td>
<td>181/200</td>
<td>0/200</td>
<td>200/200</td>
</tr>
</tbody>
</table>
Outline

1 Preprocessing data

2 Feature Extraction
   • Scattering Net
   • ResNet-50

3 Classification

4 Discussion
During the first epoch, the training accuracy is about 100%.
Re-choose training and validation set

- The validation set are all non-Raphael paintings while the training set is consist of mostly Raphael paintings.
- Split the data again, and use most of the non-Raphael paintings as training data.
Result

- Accuracy on samples: 31.9%.
- Accuracy on paintings: 28.6%.
- Voting result:

<table>
<thead>
<tr>
<th>Painting</th>
<th>#1 Not</th>
<th>#2 Not</th>
<th>#3 Yes</th>
<th>#4 Yes</th>
<th>#5 Yes</th>
<th>#6 Yes</th>
<th>#7 Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voting Acc</td>
<td>200/200</td>
<td>200/200</td>
<td>0/200</td>
<td>1/200</td>
<td>0/200</td>
<td>0/200</td>
<td>46/200</td>
</tr>
</tbody>
</table>
Conclusion

- The dominant factor is which painting the sample belongs to.
- The network draw a boundary for Rapheals paintings out, and hopefully they are close to each others. So we have many true positive cases.
- But the boundary is not accurate, so it leads to many false negative cases.
Thank you for listening!

Q&A