Every Coin Matters: Reweighting Everything for Large-scale Noisy Image Classification

Shumin Han
### Task Description

- **Describe the task**

  5,000 classes

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train dataset</td>
<td>8,366,429</td>
</tr>
<tr>
<td>Google</td>
<td></td>
</tr>
<tr>
<td>Flickr</td>
<td>7,710,236</td>
</tr>
<tr>
<td>Validation dataset</td>
<td>294,099</td>
</tr>
<tr>
<td>Test dataset</td>
<td>294,099</td>
</tr>
</tbody>
</table>
Challenge Analysis

- Class imbalance: unbalance data in different classes.
  - Motivates: reweighting Class
Challenge Analysis

• Cluster Diversity: High inter-class similarity, low intra-class similarity.
  – Motivates: reweighting Cluster

Class: 973
Query words: icecream, ice+cream

Class: 3440
Query words: ice-cream cone
Challenge Analysis

• Lable Ambiguity: different classes, similar labels
  – Motivates: reweighting/smoothing label

Class: 218    Label: revolver pistol

Horse. It may be class 331, 1044,....

Motorcycle. It may be class 2334

Jeep. It may be class 268, 563,....
Challenge Analysis

• Noisy Instance: Positive instances are overwhelmed by massive noisy instances.
  – Motivates: reweighting Instance selection probability

Class: 1048  Label: *ben*

- Noise 99.4%
- Correct 0.6%
Challenge Analysis

• Instance Saliency: Not all positive instance contribute equally
  – Motivates: reweighting positive instances inside a bag (attention).

Class: 0  Label: *kit fox, vulpes macrotis*

Do they make the *same* contribution?
What if…

All Class are equally considered

All Labels are considered to be precise and correct

- No noisy
- No MIL

Baseline performance

<table>
<thead>
<tr>
<th>model</th>
<th>Top1 accuracy</th>
<th>Top5 accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnext101</td>
<td>47.2%</td>
<td>71.7%</td>
</tr>
</tbody>
</table>
Motivation

unbalance data in different classes: reweighting class

High inter-class similarity, low intra-class similarity: reweight cluster

Label Ambiguity: reweighting label

Noisy Instance: reweighting Instance selection probability

Not all positive instance contribute equally: reweighting positive instances inside a bag
Reweighting Class

• Assigning loss weight of each class

\[
\text{ratio}[i] = \frac{\text{const}}{\text{count[\text{train}[i]]}}
\]

\[
\text{weight}[i] = (1 - \vartheta) + \vartheta \times \frac{\text{ration}[i]}{\sum_{j=1}^{\text{total_class_num}} \text{ration}[j]}, i \in [1, \text{total_class_num}]
\]

• Performance
Reweighting Cluster

- Unsupervised learning: assigning weight by cluster density

**Step 1:** calculate top 5 confusion classes

- revolver pistol, ...
- automatic pistol, ...
- gat, ...
- assault + rifle, ...
- ammunition arms, ...

**Step 2:** merge all images together and cluster by k-means.

**Step 3:** assign each cluster a sampling weight according to density

\[ w_1 > w_2 > w_3 > w_4 > w_5 \]
Reweighting Cluster

- Unsupervised learning: assigning weight by cluster density

- Performance
Reweighting Instance

• Text-image correlated model structure
Reweighting Instance

• How to train Text-image correlated model

Step 1: Pick Gallery image collection of high quality google images from each class, rank < 30
word2vec: fastText 300
CNN: Pretrained Resnext101

Step 2: Train a text-image correlated model
Training loss:

\[ \text{Loss} = \max (\cos(u_p, v_p) - \cos(u_p, v_n) + m), 0) \]

- \( u_p \): image feature
- \( v_p, v_n \): text feature

Step 3: Score (reweight) the whole training dataset
Reweighting Instance

- Score results

Score in [0.9, 1.0]

Score in [0.7, 0.8]

Score in [0.0, 0.6]

- Performance

![Graph showing performance metrics](image_url)
Reweighting Bag-Specific Instance Saliency

- Bag-instance learning structure

\[
\text{Loss} = \text{Loss}_{\text{ins}} + \text{Loss}_{\text{bag}}
\]
Reweighting Bag-Specific Instance Saliency

• Performance

<table>
<thead>
<tr>
<th>12</th>
<th>15</th>
<th>18</th>
<th>21</th>
<th>24</th>
<th>26</th>
</tr>
</thead>
<tbody>
<tr>
<td>47.2</td>
<td>48.9</td>
<td>52.1</td>
<td>52.9</td>
<td>53.1</td>
<td></td>
</tr>
<tr>
<td>71.7</td>
<td>72.8</td>
<td>75.3</td>
<td>76</td>
<td>76.3</td>
<td></td>
</tr>
</tbody>
</table>

top1 accu  top5 accu
Reweighting Label

• bootstrapping

\[
\text{Loss}(q, t) = \sum_{k=1}^{\text{total class num}} [Bt_k + (1 - B)z_k] \log(q_k)
\]

\[
z_k = 1(k = \text{argmax} q_i, i = 1, \ldots, \text{total class num})
\]

Where \( z_k \) is predicted label, \( t_k \) is ground-truth label, \( B=0.8 \).

• Performance

![Graph showing performance metrics]
Solution Summary

Unsupervised clustering weight sampling

Supervised text-image correlated weight sampling

Multi-instance learning

Class weight loss

Bootstrapping
Solution Summary

Train Dataset

Class A

Class B

Reweighting Class

Train Dataset

Class A

Class B

Weight: 0.6
Solution Summary

Class B

Reweighting Cluster

Class B

Cluster 1

Cluster 2

Cluster 3

Weight: 0.6 * 0.5
Solution Summary

Cluster 3

Reweighting Instance

Cluster 3

Weight: 0.6 * 0.5 * 0.3
Solution Summary

Cluster 3

Reweighting Label

Weight: 0.6 * 0.5 * 0.75 * 0.7
### Training and testing tricks

<table>
<thead>
<tr>
<th>Tricks</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove noise out of top 15</td>
<td>+0.5% top5</td>
</tr>
<tr>
<td>Training different models (eg. Resnet, DPN, etc) for ensembling</td>
<td>+1.3% top5</td>
</tr>
<tr>
<td>Multi-crop testing</td>
<td>+1.0% top5</td>
</tr>
<tr>
<td>Multi-scale testing</td>
<td>+0.5% top5</td>
</tr>
</tbody>
</table>
The result

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team name</th>
<th>Top-5 Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vibranium</td>
<td>79.25</td>
</tr>
<tr>
<td>2</td>
<td>Overfit</td>
<td>75.30</td>
</tr>
<tr>
<td>3</td>
<td>ACRV_ANU</td>
<td>69.56</td>
</tr>
</tbody>
</table>
Future Work

1. Data cleaning
2. Optimize training models
3. Iterative cleaning of data and optimization models
Our Team

Jianfeng Zhu  Lele Cheng  Ying Su  Pengcheng Yuan

Lin Ye  Shumin Han  Jia Li

Contact: hanshumin@baidu.com