Google AI ML Winter Camp

I Fashion
Pick up your favourite one!

Burn My TPU Team
Xia Li, Yang Hu, Chaopeng Zhang
I ❤️ Fashion
Pick up your favourite one!

I saw a cool T-shirt!

But I can’t find it in the Internet. So sad..

iFashion
http://34.80.200.205

Wow, so cool!
I ❤️ Fashion
Pick up your favourite one!

I love sailor suits!

Emmmm, I wanna...

Career Dresses!

How to recommend??

Emmmm, I wanna...

Yes! Sports Clothes.
I ❤️ Fashion

Pick up your favourite one!

What we do?

How to construct it?
Data Insight
Dive into iMaterialist Dataset

● 1 million+ images
  ○ 1,014,544 train, 9,897 validation, 39,706 test (~100:1:4)

● 8 kinds of labels:
  ○ category, color, gender, material, neckline, pattern, sleeve, style

● 228 labels in total.
Data Insight
Dive into iMaterialist Dataset

- The labels are very imbalanced!

The distribution of label occurrences for each task

The distribution of label occurrences from a macro view
Data Insight
Dive into iMaterialist Dataset

The distribution of label number/image

- 0 label
- 1 label
- more labels

The distribution of label number/image for each task
Data Insight
Dive into iMaterialist Dataset

Some non-related images like...
Naive Pipeline

Offline
- Image Database
- Feature Extraction
- Feature Pool

Online
- Query
- Feature Extraction
- Similarity Computation
- TopK ranking
- Results
Feature Extractor

**Origin** choice: ImageNet pretrained model

<table>
<thead>
<tr>
<th></th>
<th>ResNet50</th>
<th>MobileNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>top1 error</td>
<td>22.96</td>
<td>26.61</td>
</tr>
<tr>
<td>top5 error</td>
<td>6.85</td>
<td>8.95</td>
</tr>
<tr>
<td>Params</td>
<td>25,549K</td>
<td>4,232K</td>
</tr>
<tr>
<td>FLOPs</td>
<td>3,875M</td>
<td>580M</td>
</tr>
</tbody>
</table>
Feature Extractor
Sophisticated choice: finetune on iMaterialist Dataset

- Classification evaluation metric for iMaterialist
  - micro F1 (average on images)

- **Naive** method:
  - multi-label learning for all labels
  - use Binary Cross Entropy Loss (BCE Loss)
  - ResNet50 + 112*112 + 1 epoch: 45.27

- **Fancy** method: learning
  - learning on each separate task
  - share the same backbone
  - BCE Loss or CE Loss
  - ResNet50 + 112*112 + 1 epoch: 48.48
  - MobileNet + 112*112 + 1 epoch: 47.71
  - MobileNet + 224*224 + 1 epoch: 49.47
Similarity Computation

- Similarity metric of features: **Cosine Distance**

- Brutally search:
  - Compare with each image in gallery

- Levelly search:
  - Clustering features using K-means, etc.
  - Compare with cluster center
  - then compare with images in one cluster

- Category search: (used in Alibaba)
  - Clustering images with their attributes
  - ...

![Diagram of Similarity Computation](Image)
Evaluation Method

Experiment settings:
  gallery: 100000 images
  query: 1000 images

Similarity Metric for images:
  \( IOU(A, B) = \frac{|L(A) \cap L(B)|}{|L(A) \cup L(B)|} \)

Makeup the groundtruth
  if \( IOU(A, B) \geq 0.5 \):
    pair(A, B)

Evaluation metric for search:
  precision, recall, F1 for topk results
Feature Extractor
Evaluation upon precision
Feature Extractor
Evaluation upon recall
Feature Extractor

Evaluation upon F1
Feature Extractor

Visual example: mobilenet pretrained (origin)
Feature Extractor

Visual example: mobilenet fine-tuned (fancy)
Sophisticated Pipeline

Offline

Image Database → Fashion Detection → Feature Extraction → Feature Pool

Online

Query → Fashion Detection → Feature Extraction → Similarity Computation → TopK ranking → Results
Fashion Detection

Mask rcnn-benchmark
https://github.com/facebookresearch/maskrcnn-benchmark

Result on Fashion Dataset

Thinking
1: Can’t detect fashion
2: Mask is not needed
Fashion Detection

**Yolov3**

https://github.com/TencentYoutuResearch/ObjectDetection-OneStageDet

1: In one stage

2: Faster

3: Good performance

4: The work of Our Youtu Lab

But … can’t detect fashion too

<table>
<thead>
<tr>
<th></th>
<th>544x544</th>
<th>VOC2007 Test(mAP)</th>
<th>Time per forward (batch size = 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yolov2</td>
<td></td>
<td>77.6%</td>
<td>11.5ms</td>
</tr>
<tr>
<td>Yolov3</td>
<td></td>
<td>79.6%</td>
<td>23.1ms</td>
</tr>
</tbody>
</table>
Fashion Detection

Training

- **Dataset**
  DeepFashion: Fashion Landmark Detection (By CUHK MMLab)

- **VOC Format**

- **Time**
  ~ 50000 steps
  ~ 2 days
Fashion Detection

**Deploy**
- Modify the code support the format from voc to single raw image
- Integrated with Classification

**Result**

![Good cases](image1.png)
![Bad cases](image2.png)
Deployment

- **Back end**
  - **Framework**
    - Django
  - **Inference**
    - Load Gallery Features, net and weights to GPU when start up

- **Front end**
  - **Framework**
    - AngularJS
  - **Interaction**
    - Send users’ data and fetch results from back end
Demo
http://34.80.200.205

Pick your favorite or Upload your image

NOW, TRY IT YOURSELF!
Team member
BurnMyTPU

李夏  张潮鹏  胡杨
Thank You!