Fast R-CNN

Author: Ross Girshick
Speaker: Charlie Liu
Date: Oct, 13\textsuperscript{th}


ECS 289G 001 Paper Presentation, Prof. Lee
Girshick. Fast R-CNN
Girshick et. Al. Rich feature hierarchies for accurate object detection and semantic segmentation

1. Dataset: PASCAL VOC 2012
Result

Accuracy

Training Speed-up

Girshick. Fast R-CNN
Girshick et. Al. Rich feature hierarchies for accurate object detection and semantic segmentation

1. Dataset: PASCAL VOC 2012
Result

Accuracy

Training Speed-up

Testing Speed-up

Girshick. Fast R-CNN
Girshick et. Al. Rich feature hierarchies for accurate object detection and semantic segmentation

1. Dataset: PASCAL VOC 2012
Object Detection after R-CNN

- **R-CNN**
  - 66.0% mAP
  - 1x Test Speed

- **SPPnet**
  - 63.1% mAP
  - 24x than R-CNN

- **Fast R-CNN**
  - 66.6% mAP
  - 10x than SPPnet

---

1. mAP based on PASCAL VOC 2007, results from Girshick

Girshick. Fast R-CNN

He et. al. Spatial pyramid pooling in deep convolutional networks for visual recognition

Girshick et. al. Rich feature hierarchies for accurate object detection and semantic segmentation
R-CNN

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

R-CNN: Regions with CNN features

Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation
R-CNN Limitations

• Too slow
  • 13s/image on a GPU
  • 53s/image on a CPU
  • VGG-Net 7x slower
R-CNN Limitations

- Too slow
- Proposals need to be warped to a **fixed size**
  - Potential loss of accuracy

---

**R-CNN: Regions with CNN features**

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Girshick et. al. Rich feature hierarchies for accurate object detection and semantic segmentation
R-CNN Limitations

• Cropping may loss the object’s information
• Warping may change the object’s appearance

He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
SPPnet: Motivation

- SPP: Spatial Pyramid Pooling
SPPnet: Motivation

- **SPP: Spatial Pyramid Pooling**
  - Fixed size input

- Arbitrary size output

---

He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
SPPnet: Motivation

- SPP: Spatial Pyramid Pooling
SPPnet: Motivation

R-CNN: 2000 nets on 1 image

SPPnet: 1 net on 1 image

---

He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
SPPnet: Motivation

- SPP Detection

He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
## SPPnet: Result

<table>
<thead>
<tr>
<th></th>
<th>R-CNN</th>
<th>SPPnet 1-scale</th>
<th>SPPnet 5-scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>58.5</td>
<td>58.0</td>
<td>59.2</td>
</tr>
<tr>
<td>GPU time / img</td>
<td>9s</td>
<td>0.14s</td>
<td>0.38s</td>
</tr>
<tr>
<td>Speed-up²</td>
<td>1x</td>
<td>64x</td>
<td>24x</td>
</tr>
</tbody>
</table>

VOC 2007¹

---

¹ mAP based on PASCAL VOC 2007, results from He et al.
² Speed stands for testing speed
SPPnet: Limitations

- High memory consumption
SPPnet: Limitations

• High memory consumption
• Training is inefficient
  • Inefficient back-propagation through SPP
SPPnet: Limitations

- High memory consumption
- Training is inefficient
  - Inefficient back-propagation through SPP
- Multi-stage pipeline
Fast R-CNN: What’s New

R-CNN: Regions with CNN features

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation
Girshick. Fast R-CNN
Fast R-CNN: What’s New

R-CNN: 

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Girshick et. al. Rich feature hierarchies for accurate object detection and semantic segmentation
Girshick. Fast R-CNN
Fast R-CNN: What’s New

- RoI Pooling Layer
  - ≈ one scale SPP layer
Fast R-CNN: What’s New

- Sibling Classifier & Regressor Layers
Fast R-CNN: What’s New
Fast R-CNN: What’s New

• Joint Training – one stage framework
  • Joint them together: feature extractor, classifier, regressor

Girshick. Fast R-CNN
Liliang Zhang, Detection: From R-CNN to Fast R-CNN
Fast R-CNN: Advantages

• One fine-tuning stage
  • Optimizes the classifier and bbox regressor
  • Loss function:
  • \( L = L_{\text{classifier}} + \lambda L_{\text{regressor}} \)
Fast R-CNN: Advantages

• One fine-tuning stage
• Fast training
  • Using mini-batch stochastic gradient descent
    • E.g. 2 images * 64 RoIs each
    • 64x faster than 128 images * 1 RoI each (strategies of R-CNN, SPPnet)
Fast R-CNN: Advantages

- One fine-tuning stage
- Fast training
- Efficient back-propagation

Girshick, Fast R-CNN
Liliang Zhang, Detection: From R-CNN to Fast R-CNN
Fast R-CNN: Advantages

• One fine-tuning stage
• Fast training
• Efficient back-propagation
• Scale invariance
  • In practice, single scale is good enough
  • Single scale: faster x10 than SPP-Net

Girshick. Fast R-CNN
Liliang Zhang, Detection: From R-CNN to Fast R-CNN
Fast R-CNN: Advantages

• One fine-tuning stage
• Fast training
• Efficient back-propagation
• Scale invariance
• Fast detecting
  • Truncated SVD\(^1\) \(W \approx U\Sigma_t V^T\)
  • Single FC layer -> two FC layers
  • Reduce parameters\(^2\) from \(uv\) to \(t(u + v)\)

---

1. \(U: \text{size}(u \times t); \Sigma: \text{size}(t \times t); V: \text{size}(v \times t)\)
2. In practice, \(t \ll \min(u, v)\)
Fast R-CNN: Advantages

- One fine-tuning stage
- Fast training
- Efficient back-propagation
- Scale invariance
- Fast detecting
  - Truncated SVD\(^1\) \(W \approx U\Sigma_t V^T\)
  - Single FC layer -> two FC layers
  - Reduce parameters\(^2\) from \(uv\) to \(t(u + v)\)

---

1. \(U: \text{size}(u * t); \Sigma: \text{size}(t * t); V: \text{size}(v * t);\)
2. In practice, \(t \ll \text{min}(u, v)\)
Fast R-CNN: Result

Results using VGG16

<table>
<thead>
<tr>
<th></th>
<th>R-CNN</th>
<th>SPPnet</th>
<th>Fast R-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train Time (h)</td>
<td>84</td>
<td>25</td>
<td>9.5</td>
</tr>
<tr>
<td>Train Speedup</td>
<td>1x</td>
<td>3.4x</td>
<td>8.8x</td>
</tr>
<tr>
<td>Test Rate (s/im)</td>
<td>47.0</td>
<td>2.3</td>
<td>0.22</td>
</tr>
<tr>
<td>Test Speedup</td>
<td>1x</td>
<td>20x</td>
<td>213x</td>
</tr>
<tr>
<td>VOC07 mAP</td>
<td>66.0%</td>
<td>63.1%</td>
<td>66.6%</td>
</tr>
</tbody>
</table>

1. Results from Girshick
Fast R-CNN: Result

Results using VGG16

<table>
<thead>
<tr>
<th></th>
<th>R-CNN</th>
<th>SPPnet</th>
<th>Fast R-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train Time (h)</td>
<td>84</td>
<td>25</td>
<td>9.5</td>
</tr>
<tr>
<td>Train Speedup</td>
<td>1x</td>
<td>3.4x</td>
<td>8.8x</td>
</tr>
<tr>
<td>Test Rate (s/im)</td>
<td>47.0</td>
<td>2.3</td>
<td>0.22</td>
</tr>
<tr>
<td>Test Speedup</td>
<td>1x</td>
<td>20x</td>
<td>213x</td>
</tr>
<tr>
<td>VOC07 mAP</td>
<td>66.0%</td>
<td>63.1%</td>
<td>66.6%</td>
</tr>
</tbody>
</table>

1. Results from Girshick
Fast R-CNN: Results

Results from Girshick

R-CNN | SPPnet | Fast R-CNN
--- | --- | ---
Train Time (h) | 84 | 25 | 9.5
Train Speedup | 1x | 3.4x | 8.8x
Test Rate (s/im) | 47.0 | 2.3 | 0.22
Test Speedup | 1x | 20x | 213x
VOC07 mAP | 66.0% | 63.1% | 66.6%

1. Results from Girshick
Fast R-CNN: Discussions

• Which layers to fine-tune?
  • Fine-tuning $\text{conv}_1$: over-runs GPU memory.
  • Fine-tuning $\text{conv}_2$: mAP +0.3%, 1.3x slower.
  • Fine-tuning $\text{conv}_3$ and up.
Fast R-CNN: Discussions

• Which layers to fine-tune? conv$_3$ and up.
• Does multi-task training help?
  • Multi-task: potential to improve results
  • Tasks influence each other through a shared representation
  • mAP +0.8% to 1.1%
Fast R-CNN: Discussions

- Which layers to fine-tune? $\text{conv}_3$ and up.
- Does multi-task training help? Yes, mAP $+0.8\%$ to $1.1\%$.
- Scale invariance: single scale or multi scale?
  - Single scale: faster
  - Multi scale: accurate
  - Best trade-off: single scale

Girshick. Fast R-CNN
Fast R-CNN: Discussions

• Which layers to fine-tune? conv$_3$ and up.
• Does multi-task training help? Yes, mAP +0.8% to 1.1%.
• Scale invariance: single scale or multi scale? Single.
• Do we need more training data?
  • VOC07: 66.9% -> 70% with augment of VOC12 dataset.
Fast R-CNN: Discussions

- Which layers to fine-tune? conv\textsubscript{3} and up.
- Does multi-task training help? Yes, mAP +0.8\% to 1.1\%.
- Scale invariance: single scale or multi scale? Single.
- Do we need more training data? Yes.
- Do SVMs outperform softmax?
  - SVM: Yes/No; Softmax: 1 vs. all.
  - Softmax: mAP +0.1\% to 0.8\%, competition between classes.
Fast R-CNN: Discussions

- Which layers to fine-tune? conv$_3$ and up.
- Does multi-task training help? Yes, mAP +0.8% to 1.1%.
- Scale invariance: single scale or multi scale? Single.
- Do we need more training data? Yes.
- Do SVMs outperform softmax? Softmax, mAP +0.1% to 0.8%.
- Are more proposals always better?
Fast R-CNN: Discussions

![Graph showing mAP vs. Number of object proposals with different scenarios: Sel. Search (SS), SS (2k) + Rand Dense, SS replace Dense, 45k Dense Softmax, 45k Dense SVM. The graph highlights the performance differences between these methods.](Girshick_Fast_R-CNN)
Fast R-CNN: Discussions

• Which layers to fine-tune? conv$_3$ and up.
• Does multi-task training help? Yes, mAP +0.8% to 1.1%.
• Scale invariance: single scale or multi scale? Single.
• Do we need more training data? Yes.
• Do SVMs outperform softmax? Softmax, mAP +0.1% to 0.8%.
• Are more proposals always better? No.
Questions?

Thanks!