Team G-RMI: Google Research & Machine Intelligence

Alireza Fathi (alirezafathi@google.com)
Nori Kanazawa, Kai Yang, George Papandreou, Tyler Zhu, Jonathan Huang, Vivek Rathod, Chen Sun, Kevin Murphy, et al.
Google: Mobile First to AI First
Beyond image classification...
What we need: boxes, segments, human pose

Based on a figure from Jia Deng
And also: attributes, relations, 3-d, ...
This Talk

- Semantic Stuff Segmentation
- Object Instance Segmentation
- Human Pose Estimation
Team Members

Alireza Fathi
Kai Yang
Kevin Murphy
Masternet (DenseLab)
A Model based on DeepLab for Semantic Segmentation

Rgb Image

Feature Extractor

[240 x 320 x 3]

(repeated for multiple scales)

[30 x 40 x 2048]

Atrous Conv (24)

Atrous Conv (12)

Atrous Conv (6)

Atrous Conv (3)

[240 x 320 x 10]

Semantic Labels

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille. ICLR'15, CVPR'16, PAMI'17.
Tricks for improving results

- **Feature extractor**
  - Inception resnet gives around 3% improvement in comparison to Resnet101.

- **Initial checkpoint (Not used for the competition)**
  - Starting from a COCO semantic segmentation checkpoint gives a 2-3% boost in comparison to starting from an ImageNet classification checkpoint.

- **Ensembling**
  - We get around 3-4% improvement in performance by ensembling 5 models.

- **Batch size**
  - Larger batch size and smaller crop size seems to be better than smaller batch size but larger crop size. We use a batch size of 72.

- **Balancing classes**
  - We get around 1% improvement by balancing the loss among different classes.

- **Moving average? Focal loss? New network architectures?**
Example results on ADE20K

Image

Groundtruth

Prediction
Example results on ADE20K

Image  

Groundtruth  

Prediction
Example results on ADE20K

Image

Groundtruth

Prediction
Example results on Coco Stuff

Image

Groundtruth

Prediction
Example results on Coco Stuff

Image  
Groundtruth  
Prediction
Example results on Coco Stuff

Image

Groundtruth

Prediction
Boundary Detection
Depth/Surface Normal Prediction
Depth/Surface Normal Prediction

Groundtruth

Prediction
Depth/Surface Normal Prediction

Groundtruth

Prediction
This Talk

- Semantic Stuff Segmentation
- **Object Instance Segmentation**
- Human Pose Estimation
Team Members

Alireza Fathi  Nori Kanazawa  Kevin Murphy
Available open source: pre-trained models or train your own on GCloud

Tensorflow Object Detection API

Creating accurate machine learning models capable of localizing and identifying multiple objects in a single image remains a core challenge in computer vision. The Tensorflow Object Detection API is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. At Google we’ve certainly found this codebase to be useful for our computer vision needs, and we hope that you will as well.

Contributions to the codebase are welcome and we would love to hear back from you if you find this API useful. Finally if you use the Tensorflow Object Detection API for a research publication, please consider citing:

TensorFlow Object Detection API

<table>
<thead>
<tr>
<th>Meta-architecture</th>
<th>SSD, Faster R-CNN, R-FCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Extractor</td>
<td>Mobilenet, VGG, Inception V2, Inception V3, Resnet-50, Resnet-101, Resnet-152, Inception Resnet v2</td>
</tr>
<tr>
<td>Learning schedule</td>
<td>Manually Stepped, Exponential Decay, etc</td>
</tr>
<tr>
<td>Location Loss function</td>
<td>L2, L1, Huber, IOU</td>
</tr>
<tr>
<td>Classification Loss function</td>
<td>SigmoidCrossEntropy, SoftmaxCrossEntropy</td>
</tr>
</tbody>
</table>
Rule of thumb: SSD (diamonds) faster than R-FCN (squares), which is faster than Faster R-CNN (circles).

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. Speed/accuracy trade-offs for modern convolutional object detectors. CVPR 2017
SSD with MobileNet (and low resolution images) is fastest

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. Speed/accuracy trade-offs for modern convolutional object detectors. CVPR 2017
Mask R-CNN

Image → Features → Region Proposals → Cropped Region Features →

- Classification
- Box Refinement
- Segmentation
- Boundary Detection

Resnet
Inception v3
MobileNet
Inception Resnet etc.
Mask R-CNN: Segmentation Head
Example results from ADE20K
Example results from MS COCO
Example results from MS COCO
Example results from MS COCO
Object Instance Segmentation
Instance Segmentation API

What is Instance Segmentation?

Updated 2017-08-14

Instance segmentation is the problem of identifying individual instances of objects and their categories (such as person and car) in an image. It differs from object detection in that the output is a mask representing the shape of each object, rather than just a bounding box. It differs from semantic segmentation in that our goal is not just to classify each pixel with a label (or as background), but also to distinguish individual instances of the same class.
Tricks for improving results

- **Input image size**
  - 600 to 800 gives around 1.5-2% improvement on MSCOCO

- **Network stride**
  - Changing inception resnet from stride 16 to stride 8 gives around 1% improvement

- **Using intermediate layers**
  - Intermediate layers for mask prediction part of the network gives around 0.8% improvement

- **Weight of the mask prediction loss**
  - I found the best balance given the current architecture is to give weight 4.0 to mask loss

- **Mask size**
  - Changing mask prediction size from 16 to 32 gives around 0.15% improvement

- **Number of convolutions in mask prediction head**
  - Increasing number of conv2d layers from 1 to 3 improves performance by 0.3%
Boundary and mask prediction

- Gives around 0.5% improvement
Boundary and mask prediction
Boundary and mask prediction
Issues in Box-based Mask Prediction 1/3

- Sometimes there are multiple instances of the same object in a box.
Issues in Box-based Mask Prediction 1/3

- Sometimes there are multiple instances of the same object in a box.
Some objects are thin and appear diagonal in image. Barely occupy 10% of the given box!
Issues in Box-based Mask Prediction 2/3

- Some objects are thin and appear diagonal in image. Barely occupy 10% of the given box!
Issues in Box-based Mask Prediction 3/3

- Some things are objects, and some are stuff such as grass, sky, etc. Not meaningful to put a box around stuff.
- Sometimes objects are occluded.
WIP: Bottom-up (Box-Free) Instance Segmentation

Fathi, Wojna, Rathod, Wang, Song, Guadarrama, Murphy, Semantic Instance Segmentation via Deep Metric Learning, arXiv 1703.10277
Projection of Embedding Vectors to RGB Space
Qualitative Results
This Talk

- Semantic Stuff Segmentation
- Object Instance Segmentation
- Human Pose Estimation
Team Members

George Papandreou

Tyler Zhu

Nori Kanazawa

Alex Toshev

Jonathan Tompsoon

Chris Bregler

Hartwig Adam

Kevin Murphy
Top-down (2-stage) pipeline

"Towards Accurate Multi-person Pose Estimation in the Wild",
George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, Kevin Murphy. CVPR'17.
Person Detection

1. Uses our open source object detection system*, which won COCO-Det 2016.
2. Faster-RCNN, with ResNet-101 feature extractor, trained on person vs non-person (COCO data).
4. Output stride=8 via atrous convolution.
5. Image resized to 800 min side or 1200 max side.

Box AP for person (testdev): 0.487

*Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. Speed/accuracy trade-offs for modern convolutional object detectors. CVPR 2017
Heatmap Output

- Heatmap field for each keypoint
  - 17 channels (1 within a disk around each keypoint, 0 outside)
  - Sigmoid cross entropy loss
- Intermediate and final CNN layers

Crop | Target | Net-Layer52 | Net-Layer101

photo credit: Andrew Taylor
Offset Output

- Offset field towards the center of the disk
  - 34 channels for x- and y- offsets
  - Huber loss, only active within disks
- Intermediate and final CNN layers
Fusing Heatmaps and Offsets via Hough Voting

**Algo: Offset-guided Hough voting**

For each point in the heatmap:

1. Transfer its mass by the corresponding offset.
2. Accumulate into one 2-D Hough array per part.

\[
P_i(x_i) = \sum_x G_i(x) \cdot K(x_i - x - F_i(x))
\]
Final Pose Prediction: Keypoint Position and Score

CNN + Hough Voting

Fused activation maps

Keypoint Position \((x_i, y_i)\)

Keypoint score \(s_i\)

Instance score \(s\)

Mean
Our Progress on COCO-Keypoints Benchmark

- Submission to COCO-2016 keypoints
  - AP: 0.605 (COCO+internal*, ranked #2)
- Improvements\(^1\) for CVPR 2017 paper
  - AP: 0.649 (COCO)
  - AP: 0.685 (COCO+internal)
- Improvements\(^2\) for COCO-2017 competition:
  - AP: 0.669 (COCO)
  - AP: 0.710 (COCO+internal, ranked #4)

*Internal dataset (400k people, 130k images).

\(^1\)Exponential moving average of parameters, better model and system tuning and hyperparameter settings.

\(^2\)Intermediate supervision for all heatmap+offsets+displacements, resnet-152, better feature alignment, longer training without decreasing learning rate.

All AP numbers use testdev
## COCO Keypoints Results (testdev)

<table>
<thead>
<tr>
<th></th>
<th>AP</th>
<th>AP@.5</th>
<th>AP@.75</th>
<th>AP (M)</th>
<th>AP (L)</th>
<th>AR</th>
<th>AR@.5</th>
<th>AR@.75</th>
<th>AR (M)</th>
<th>AR (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours COCO-2016 #2 (COCO+internal)</td>
<td>0.605</td>
<td>0.822</td>
<td>0.662</td>
<td>0.576</td>
<td>0.666</td>
<td>0.662</td>
<td>0.866</td>
<td>0.714</td>
<td>0.619</td>
<td>0.722</td>
</tr>
<tr>
<td>CMU-Pose COCO-2016 #1</td>
<td>0.618</td>
<td>0.849</td>
<td>0.675</td>
<td>0.571</td>
<td>0.662</td>
<td>0.665</td>
<td>0.872</td>
<td>0.718</td>
<td>0.606</td>
<td>0.746</td>
</tr>
<tr>
<td>paper</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mask-RCNN paper</td>
<td>0.631</td>
<td>0.873</td>
<td>0.687</td>
<td>0.578</td>
<td>0.714</td>
<td>0.697</td>
<td>0.916</td>
<td>0.749</td>
<td>0.637</td>
<td>0.778</td>
</tr>
<tr>
<td>Associative Embedding paper</td>
<td>0.655</td>
<td>0.868</td>
<td>0.723</td>
<td>0.606</td>
<td>0.726</td>
<td>0.702</td>
<td>0.895</td>
<td>0.760</td>
<td>0.646</td>
<td>0.781</td>
</tr>
<tr>
<td>Ours (COCO-only)</td>
<td>0.669</td>
<td>0.864</td>
<td>0.736</td>
<td>0.640</td>
<td>0.720</td>
<td>0.716</td>
<td>0.892</td>
<td>0.776</td>
<td>0.661</td>
<td>0.791</td>
</tr>
<tr>
<td>Ours (COCO+internal)</td>
<td>0.696</td>
<td>0.872</td>
<td>0.766</td>
<td>0.670</td>
<td>0.742</td>
<td>0.742</td>
<td>0.903</td>
<td>0.804</td>
<td>0.692</td>
<td>0.811</td>
</tr>
<tr>
<td>Ours (COCO+internal) ResNet-152</td>
<td>0.710</td>
<td>0.879</td>
<td>0.777</td>
<td>0.690</td>
<td>0.752</td>
<td>0.758</td>
<td>0.912</td>
<td>0.819</td>
<td>0.714</td>
<td>0.820</td>
</tr>
</tbody>
</table>
WIP*: Bottom-up part detection + grouping

3 fully convolutional outputs

K binary heatmaps
K 2d offsets
K^2 pairwise offsets

*G. Papandreou et al, CVPR'18

Each part predicts offset to other parts. Greedy decoding.
## Top-down vs bottom-up

<table>
<thead>
<tr>
<th>Feature</th>
<th>Top-Down (box based)</th>
<th>Bottom-Up (grouping based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Slower ((time \propto #\text{people}))</td>
<td>Faster ((\text{time independent of } #\text{people}))</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Higher ((\text{zoom in on image}))</td>
<td>Lower on small instances</td>
</tr>
<tr>
<td>Flexibility/ modularity</td>
<td>Higher ((\text{cf MaskRCNN}))</td>
<td>Lower</td>
</tr>
<tr>
<td>Model size/ complexity</td>
<td>Larger ((\text{more code, params}))</td>
<td>Smaller</td>
</tr>
</tbody>
</table>
Multi-Person Pose Estimation Demo
Thanks!

SPEAKER
Alireza Fathi

Special thanks to
Nori Kanazawa, Kai Yang, Kevin Murphy, Derek Chow, Ian Fischer, Sergio Guadarrama, Jonathan Huang, Anoop Korattikara, Kevin Murphy, Vivek Rathod, Yang Song, Chen Sun, Matt Tang, Zbigniew Wojna, Menglong Zhu, George Papandreou, Rahul Sukthankar.
Example results from MS COCO
Example results from MS COCO
Example results from MS COCO
Towards Box-Free (Bottom-up) Instance Segmentation
Embedding Loss

- K Samples of N-pair loss.
Embedding Loss

- K Samples of N-pair loss.
Embedding Loss

- K Samples of N-pair loss.
Embedding Loss

- K Samples of N-pair loss.
Embedding Loss

- K Samples of N-pair loss.
Embedding Loss

- K Samples of N-pair loss.

- Randomly Pick an embedding vector from class $c_i$. Let's call it $e_1$.
- Randomly pick one embedding vector from each class $c_i, c_j, \ldots$.
- Compute the similarity between $e_1$ and the other $n$ embedding vectors.
- We like the similarity between $e_1$ and the vector belonging to class $c_i$ to be high, and the similarity with embedding vectors belonging to other classes to be low.
Embedding Loss

- **K Samples of N-pair loss.**

- Randomly pick an embedding vector from class $c_i$. Let's call it $e_1$.
- Randomly pick one embedding vector from each class $c_i, c_j, \ldots$.
- Compute the similarity between $e_1$ and the other $n$ embedding vectors.
- We like the similarity between $e_1$ and the vector belonging to class $c_i$ to be high, and the similarity with embedding vectors belonging to other classes to be low.
Embedding Loss

- K Samples of N-pair loss.

Sigmoid Cross Entropy Loss
Projection of Embedding Vectors to RGB Space
Qualitative Results