Depth Sensing and Deep Learning: Grasping and Segmenting 3D Objects from Real Depth Images using Synthetic Data

Mike Danielczuk, Jeffrey Mahler, Matthew Matl, Saurabh Gupta, Andrew Lee, Andrew Li, Vishal Satish, Bill DeRose, Stephen McKinley, Ken Goldberg
Classification
Example: Spam Filter

- **Input:** an email
- **Output:** spam/ham

**Setup:**
- Get a large collection of example emails, each labeled "spam" or "ham"
- Note: someone has to hand label all this data!
- Want to learn to predict labels of new, future emails

**Features: The attributes used to make the ham/spam decision**
- Words: FREE!
- Text Patterns: $dd, CAPS
- Non-text: SenderInContacts
- ...

---

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.
Example: Digit Recognition

- Input: images / pixel grids
- Output: a digit 0-9

Setup:
- Get a large collection of example images, each labeled with a digit
- Note: someone has to hand label all this data!
- Want to learn to predict labels of new, future digit images

Features: The attributes used to make the digit decision
- Pixels: (6,8)=ON
- Shape Patterns: NumComponents, AspectRatio, NumLoops
- ...
Other Classification Tasks

- Classification: given inputs $x$, predict labels (c)

- Examples:
  - Spam detection (input: document, classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Automatic essay grading (input: document, classes: grades)
  - Fraud detection (input: account activity, classes: fraud / no fraud)
  - Customer service email routing
  - ... many more

- Classification is an important commercial technology!
Black Grouse
Imagenet: 14M labeled images, 20K categories
Classical Computer Vision Pipeline.

CV experts

1. Select / develop features: SURF, HoG, SIFT, RIFT, ...
2. Add Machine Learning for multi-class recognition and train classifier

Classical CV feature definition is domain-specific and time-consuming
Imagenet Classification 2012

- Krizhevsky et al. -- 16.4% error (top-5)
- Next best (non-convnet) -- 26.2% error
# Imagenet 2012 Leaderboard


<table>
<thead>
<tr>
<th>N</th>
<th>Error-5</th>
<th>Algorithm</th>
<th>Team</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.153</td>
<td>Deep Conv. Neural Network</td>
<td>Univ. of Toronto</td>
<td>Krizhevsky et al</td>
</tr>
<tr>
<td>2</td>
<td>0.262</td>
<td>Features + Fisher Vectors + Linear classifier</td>
<td>ISI</td>
<td>Gunji et al</td>
</tr>
<tr>
<td>3</td>
<td>0.270</td>
<td>Features + FV + SVM</td>
<td>OXFORD_VGG</td>
<td>Simonyan et al</td>
</tr>
<tr>
<td>4</td>
<td>0.271</td>
<td>SIFT + FV + PQ + SVM</td>
<td>XRCE/INRIA</td>
<td>Perronin et al</td>
</tr>
<tr>
<td>5</td>
<td>0.300</td>
<td>Color desc. + SVM</td>
<td>Univ. of Amsterdam</td>
<td>van de Sande et al</td>
</tr>
</tbody>
</table>
### Imagenet 2013 Leaderboard


<table>
<thead>
<tr>
<th>N</th>
<th>Error-5</th>
<th>Algorithm</th>
<th>Team</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.117</td>
<td>Deep Convolutional Neural Network</td>
<td>Clarifi</td>
<td>Zeiler</td>
</tr>
<tr>
<td>2</td>
<td>0.129</td>
<td>Deep Convolutional Neural Networks</td>
<td>Nat.Univ Singapore</td>
<td>Min LIN</td>
</tr>
<tr>
<td>3</td>
<td>0.135</td>
<td>Deep Convolutional Neural Networks</td>
<td>NYU</td>
<td>Zeiler Fergus</td>
</tr>
<tr>
<td>4</td>
<td>0.135</td>
<td>Deep Convolutional Neural Networks</td>
<td></td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>5</td>
<td>0.137</td>
<td>Deep Convolutional Neural Networks</td>
<td>Overfeat NYU</td>
<td>Pierre Sermanet et al</td>
</tr>
</tbody>
</table>
Imagenet Classification 2013


- Pre-2012: 26.2% error → 2012: 16.5% error → 2013: 11.2% error
Today’s Lecture

• (Brief) Intro to Convolutional Neural Networks (CNNs)
• Learning Instance Specific Grasping
  • Learning Grasp Quality CNNs
  • Learning Instance Segmentation CNNs
Today’s Lecture

- (Brief) Intro to Convolutional Neural Networks (CNNs)
- Learning Instance Specific Grasping
  - Learning Grasp Quality CNNs
  - Learning Instance Segmentation CNNs
Convolution Layer

32x32x3 image

32 height

32 width

3 depth

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
Convolution Layer

32x32x3 image

5x5x3 filter

Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
Convolution Layer

32x32x3 image
5x5x3 filter $w$

1 number: the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

$w^T x + b$

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
Convolution Layer

32x32x3 image
5x5x3 filter

convolve (slide) over all spatial locations

activation map

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
A closer look at spatial dimensions:

7x7 input (spatially)
assume 3x3 filter

=> 5x5 output
For example, if we had 6 5x5 filters, we’ll get 6 separate activation maps:

We stack these up to get a “new image” of size 28x28x6!
Why is Convolution good?

• Shared Parameters
  • Only store filter parameters: $5 \times 5 \times 3$ (for kernel) + 1 (for bias) for example

• Local Connectivity
  • Each neuron only corresponds to a local patch of the image (not the whole thing)
Pooling

Effect = invariance to small translations of the input
Pooling

- makes the representations smaller and more manageable
- operates over each activation map independently

slide from: Fei-Fei Li & Andrej Karpathy & Justin Johnson
Idea: Learn Features
Today’s Lecture

• (Brief) Intro to Convolutional Neural Networks (CNNs)

• Learning Instance-Specific Grasping
  • Learning Grasp Quality CNNs
  • Learning Instance Segmentation CNNs
Grand Challenge

The ability to “grasp millions of different sized and shaped objects... would have significant impact on deployment of robots in factories, in warehouses, and in homes.” - Rod Brooks
Universal Picking:
diversely shaped and sized objects
Motivation: Instance-Specific Grasping

Desired Object: Dasani Drops
Grasping under Uncertainty
First Wave: Analytic Methods

$R(x, u) \in \{0, 1\}$

$u^* = \pi(x) = \arg\max R(x, u)$
Uncertainty in object identity: aggregate pre-computed grasp information from multiple objects in database for robustness to recognition errors.

Computations performed over database of ~200 household objects.

Image-Based Grasp Planning

Input Image → Proposals → Filter Bank → Regressor → Grasp

Color Images
[Saxena et al., 2008]
[Stark et al., 2008]
[Bohg & Kragic, 2010]
[Le et al., 2010]

Point Clouds
[Saxena et al., 2008]
[Detry et al., 2009]
[Hubner & Kragic, 2010]
[Boularis et al., 2011]

Image sources: Saxena et al., 2008, Pinto & Gutpa, 2016
Deep Grasp Planning

Convolutional Neural Network

Input Image

[Image of a color image]

Grasp

[Image of a grasp]

Color Images
[Lenz et al., 2015]
[Redmon & Angelova, 2016]
[Pinto et al., 2016]
[Levine et al., 2017]

Point Clouds
[Kappler et al., 2015]
[Gualtieri et al., 2016]
[Johns et al., 2016]
[Viereck et al., 2017]

Image source: Pinto & Gutpa, 2016
Data Sources

Human labeling

1035 images: 80% success


Self-supervision

~1 robot year: 80-90% success

S. Levine et al.. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.” IJRR 2017.
Synthetic Point Clouds
Dex-Net
6.7 million examples

\[ D = \{(y_i, u_i, R_i)\}_{i=1}^{I} \]
Grasp Quality CNN
GQ-CNN for Bins

\[ \theta^* = \min_{\theta \in \Theta} \sum_{i=1}^{I} \mathcal{L}(R_i, Q_{\theta}(y_i, u_i)) \]

\[ u^* = \pi(y) \in \arg\max_{u \in U_y} Q_{\theta^*}(y, u) \]

Iter 0  Iter 1  Plan
Motivation: Instance-Specific Grasping

Desired Object: Dasani Drops from
One Approach

**Segment** all objects in the scene, **Classify** all segments, **Grasp** segment that matches the target!
Visual Perception Problems

Object Detection ✔

Semantic Segmentation ✔

Instance Segmentation ❎

Slide from: Kaiming He
Mask R-CNN
Mask R-CNN

- State-of-the-art instance segmentation network
- Requires massive hand-labeled datasets for training
- Does not generalize to unseen classes

Automated Dataset Generation

- **WISDOM-Sim**: 50,000 images with 320,000 object labels in just 3.5 hours
  - 80/20 train/test split for both images and objects
  - 1600 unique objects, 10,000 unique heaps
Sampling Distributions

1) We sample a state $\mathbf{x}_k \sim p(\mathbf{x})$ using a dataset of 3D CAD models, dynamic simulation, and domain randomization over object and camera states.

2) We sample a synthetic depth image $\mathbf{y}_k \sim p(\mathbf{y}_k | \mathbf{x}_k)$ using rendering.

3) We compute visible object masks $M_i$ determining the set of pixels with a corresponding 3D point on the surface of object $O_i$. 
Data Augmentation
Modal Segmentation Masks
Amodal Segmentation Masks
SD Mask R-CNN

- State-of-the-art performance for **object instance segmentation** on real depth images
- No hand-labeling required, trained solely on **synthetic depth images**
- Generalizes to unseen objects
- Outperforms point cloud clustering baselines by 15% in average precision and 20% in average recall
COCO Metrics

IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}
COCO Metrics

**Example**
Threshold: 0.5

True positive
- Ground truth
- Prediction
- IoU = 0.8

False negative
- Ground truth
- Prediction
- IoU = 0.1

**Precision**: \( \frac{TP}{TP + FP} \)
**Recall**: \( \frac{TP}{TP + FN} \)
<table>
<thead>
<tr>
<th>Method</th>
<th>High-Res</th>
<th></th>
<th>Low-Res</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP</td>
<td>AR</td>
<td>AP</td>
<td>AR</td>
</tr>
<tr>
<td>Euclidean Clustering</td>
<td>0.324</td>
<td>0.467</td>
<td>0.183</td>
<td>0.317</td>
</tr>
<tr>
<td>Region Growing</td>
<td>0.349</td>
<td>0.574</td>
<td>0.180</td>
<td>0.346</td>
</tr>
<tr>
<td>FT Mask R-CNN (Depth)</td>
<td>0.370</td>
<td>0.616</td>
<td>0.331</td>
<td>0.546</td>
</tr>
<tr>
<td>FT Mask R-CNN (Color)</td>
<td>0.384</td>
<td>0.608</td>
<td><strong>0.385</strong></td>
<td><strong>0.613</strong></td>
</tr>
<tr>
<td>SD Mask R-CNN</td>
<td><strong>0.516</strong></td>
<td><strong>0.647</strong></td>
<td>0.356</td>
<td>0.465</td>
</tr>
</tbody>
</table>
Application: Instance-Specific Grasping

Target: Dasani Drops
Thank You!

Project Website, Supplementary Material, and Datasets:  

Code:  
https://github.com/BerkeleyAutomation/sd-maskrcnn