Brewing a Deeper Understanding of Images

Yangqing Jia, Google Research
About Me

- Ph.D. from UC Berkeley
- Research scientist @Google

- Wrote the open-source Caffe library
- (with collaborators) won ILSVRC 2014
GoogLeNet

Network in network

We need to go deeper
Deep Convolutional Networks

Revolutionizing computer vision since 1989
Deep Convolutional Networks

Revolutionizing computer vision since 1989
Why is the deep learning revolution arriving just now?

- Deep learning needs a lot of training data.
- Deep learning needs a lot of computational resources
Why is the deep learning revolution arriving just now?

- It used to be hard and cumbersome to train deep models due to **sigmoid** nonlinearities.
Why is the deep learning revolution arriving just now?

- It used to be hard and cumbersome to train deep models due to **sigmoid** nonlinearities.

- Deep neural networks are highly non-convex without any obvious optimality guarantees or nice **theory**.
Why is the deep learning revolution arriving just now?

- It used to be hard and cumbersome to train deep models due to sigmoid nonlinearities.

- Deep neural networks are highly non-convex without any optimality guarantees or nice theory.
UFLDL (2010) on Deep Learning

“While the theoretical benefits of deep networks in terms of their compactness and expressive power have been appreciated for many decades, until recently researchers had little success training deep architectures.”

… snip …

“How can we train a deep network? One method that has seen some success is the greedy layer-wise training method.”

… snip …

“Training can either be supervised (say, with classification error as the objective function on each step), but more frequently it is unsupervised “

Andrew Ng, UFLDL tutorial
GoogLeNet
GoogLeNet vs State of the art

GoogLeNet

Zeiler-Fergus Architecture (1 tower)

Convolution
Pooling
Softmax
Other
Problems with training deep architectures?

- Vanishing gradient?
- Exploding gradient?
- Tricky weight initialization?
Problems with training deep architectures?

Vanishing gradient?
Exploding gradient?
Tricky weight initialization?
Justified Questions

Why does it have so many layers???

Network in network
We need to go deeper
Hebbian Principle
Cluster according activation statistics
Cluster according correlation statistics
Cluster according correlation statistics

Layer 3

Layer 2

Layer 1

Input
In images, correlations tend to be local
Cover very local clusters by 1x1 convolutions
Less spread out correlations

number of filters

1x1
Cover more spread out clusters by 3x3 convolutions.
Cover more spread out clusters by 5x5 convolutions

number of filters

1x1

3x3
Cover more spread out clusters by 5x5 convolutions
A heterogeneous set of convolutions

number of filters

1x1
3x3
5x5
Schematic view (naive version)

1x1
3x3
5x5

number of filters
Naive idea (does not work!)

Diagram:
- Previous layer
- 1x1 convolutions
- 3x3 convolutions
- 5x5 convolutions
- 3x3 max pooling
- Filter concatenation
Inception module

- Filter concatenation
- 3x3 convolutions
- 5x5 convolutions
- 1x1 convolutions
- 1x1 convolutions
- 1x1 convolutions
- 3x3 max pooling
- Previous layer

1x1 convolutions
9 Inception modules

Network in a network in a network...
Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.
Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely
Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely

Number of parameters is reduced to 5 million
Inception

Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.

Can remove fully connected layers on top completely

Number of parameters is reduced to 5 million

Computational cost is increased by less than 2X compared to Krizhevsky’s network. (<1.5Bn operations/evaluation)
More on Crops

AlexDistortion

(and mirrored versions)
More on Crops

Our final choice

256x256 + WojCrop

320x320 + WojCrop

A total of 144 images.
(Hint - can use sliding window to speedup computation)
https://drive.google.com/a/google.com/folderview?id=0B2_Z149-oOCWbjNERTNVVVRDYjQ&usp=sharing
## Classification results on ImageNet 2012

<table>
<thead>
<tr>
<th>Number of Models</th>
<th>Number of Crops</th>
<th>Computational Cost</th>
<th>Top-5 Error</th>
<th>Compared to Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 (center crop)</td>
<td>1x</td>
<td>10.07%</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>10*</td>
<td>10x</td>
<td>9.15%</td>
<td>-0.92%</td>
</tr>
<tr>
<td>1</td>
<td>144 (Our approach)</td>
<td>144x</td>
<td>7.89%</td>
<td>-2.18%</td>
</tr>
<tr>
<td>7</td>
<td>1 (center crop)</td>
<td>7x</td>
<td>8.09%</td>
<td>-1.98%</td>
</tr>
<tr>
<td>7</td>
<td>10*</td>
<td>70x</td>
<td>7.62%</td>
<td>-2.45%</td>
</tr>
<tr>
<td>7</td>
<td>144 (Our approach)</td>
<td>1008x</td>
<td>6.67%</td>
<td>-3.41%</td>
</tr>
</tbody>
</table>

*Cropping by [Krizhevsky et al 2014]*
## Classification results on ImageNet 2012

<table>
<thead>
<tr>
<th>Team</th>
<th>Year</th>
<th>Place</th>
<th>Error (top-5)</th>
<th>Uses external data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuperVision</td>
<td>2012</td>
<td>-</td>
<td>16.4%</td>
<td>no</td>
</tr>
<tr>
<td>SuperVision</td>
<td>2012</td>
<td>1st</td>
<td>15.3%</td>
<td>ImageNet 22k</td>
</tr>
<tr>
<td>Clarifai</td>
<td>2013</td>
<td>-</td>
<td>11.7%</td>
<td>no</td>
</tr>
<tr>
<td>Clarifai</td>
<td>2013</td>
<td>1st</td>
<td>11.2%</td>
<td>ImageNet 22k</td>
</tr>
<tr>
<td>MSRA</td>
<td>2014</td>
<td>3rd</td>
<td>7.35%</td>
<td>no</td>
</tr>
<tr>
<td>VGG</td>
<td>2014</td>
<td>2nd</td>
<td>7.32%</td>
<td>no</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>2014</td>
<td>1st</td>
<td>6.67%</td>
<td>no</td>
</tr>
</tbody>
</table>
Inspecting predictions

Try Andrej's game of human vs GoogLeNet:

http://cs.stanford.edu/people/karpathy/ilsvrc/

Andrej: 5.1%*  GoogLeNet: 6.6%

* training time unknown
Classification failure cases

**Groundtruth:** ???
Classification failure cases

Groundtruth: Police car
Classification failure cases

Groundtruth: Police car

GoogLeNet:
- laptop
- hair drier
- binocular
- ATM machine
- seat belt
Classification failure cases

Groundtruth: ????
Classification failure cases

Groundtruth: coffee mug
Classification failure cases

**Groundtruth:** coffee mug

**GoogLeNet:**
- table lamp
- lamp shade
- printer
- projector
- desktop computer
Detection

## Final Detection Results

<table>
<thead>
<tr>
<th>Team</th>
<th>Year</th>
<th>Place</th>
<th>mAP</th>
<th>external data</th>
<th>ensemble</th>
<th>contextual model</th>
<th>approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>UvA-Euvision</td>
<td>2013</td>
<td>1st</td>
<td>22.6%</td>
<td>none</td>
<td>?</td>
<td>yes</td>
<td>Fisher vectors</td>
</tr>
<tr>
<td>Deep Insight</td>
<td>2014</td>
<td>3rd</td>
<td>40.5%</td>
<td>ILSVRC12 Classification + Localization</td>
<td>3 models</td>
<td>yes</td>
<td>ConvNet</td>
</tr>
<tr>
<td>CUHK DeepID-Net</td>
<td>2014</td>
<td>2nd</td>
<td>40.7%</td>
<td>ILSVRC12 Classification + Localization</td>
<td>?</td>
<td>no</td>
<td>ConvNet</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>2014</td>
<td>1st</td>
<td>43.9%</td>
<td>ILSVRC12 Classification</td>
<td>6 models</td>
<td>no</td>
<td>ConvNet</td>
</tr>
</tbody>
</table>
Brewing Deep Networks
With Caffe

Yangqing Jia, Google Research & UC Berkeley
Evan Shelhamer, Jeff Donahue, Sergey Karayev,
Jonathan Long, Ross Girshick, Sergio Guadarrama,
Trevor Darrell
... and more
xkcd: Tasks

“The Virtually Impossible”
PARK or BIRD

Want to know if your photo is from a U.S. national park? Want to know if it contains a bird? Just drag it into the box to the left, and we'll tell you. We'll use the GPS embedded in your photo (if it's there) to see whether it's from a park, and we'll use our super-cool computer vision skills to try to see whether it's a bird (which is a hard problem, but we do a pretty good job at it).

To try it out, just drag any photo from your desktop into the upload box, or try dragging any of our example images. We'll give you your answers below!

Want to know more about PARK or BIRD, including why the heck we did this? Just click here for more info →

EXAMPLE PHOTOS

PARK?  BIRD?

YES  YES

Ah yes, Everglades is truly beautiful.

Dude, that is such a bird.
But...
Rich feature hierarchies for accurate object detection and semantic segmentation

Ross Girshick  Jeff Donahue  Trevor Darrell  Jitendra Malik
UC Berkeley

Abstract

Object detection performance, as measured on the canonical PASCAL VOC dataset, has improved in the
last five years. The top-performing methods are complex ensembler sys-
tems that typically combine multiple low-level image features with
high-level context. In this paper, we propose a simple, end-to-end,
CNN-based approach to object detection that reaches a new state-of-the-
art, outperforming the best ensembler system by a margin of 5.5%. We
achieve this by using a single, large CNN and training it on the
PASCAL VOC 2012 training data, applying a single region-based
object detection algorithm to the CNN’s output. We call our method
CaffeNet and use it to build a system that outperforms the best
existing ensembler system on the PASCAL VOC 2012 test set.

1. Introduction

Features matter. The last decade of progress on various visual recognition tasks has been based
consistently on the use of SIFT [2] and HoG [3]. But if we look at perform-
ance on the canonical visual recognition task, PASCAL VOC object detection [4], it is
primarily acknowledged that the initial breakthroughs came from
features obtained by building ensemble systems and empiri-
cally tuning various parameters.

SIFT and HoG are extracting features, which
is then a very powerful tool for image
recognition. However, we are also
to see that recognition systems that
use these features are unable to
reach state-of-the-art performance
due to the limitations of the
features themselves.

Panoramic’s “matchpoint” [7], a
biologically-inspired hierarchical
and shift-invariant model for
eigen recognition, was an early
attempts at such a system. The
matchpoint, however, lacked a supervised training

Deep learning framework
developed by
Yangqing Jia / BVLC
In One Sip

• A fast implementation of deep networks
• Well unit-tested code
• Tools, demos, recipes, “model zoo”s
• Cross-platform (CPU, GPU, mobile…) code
• And Open Source!
A Network is...

- Defined with Google Protocol Buffer in raw text format; implementation independent.
- A DAG of layers.

```plaintext
define name: "dummy-net"
layers { name: "data" ... }
layers { name: "conv" ... }
layers { name: "pool" ... }
... more layers ...
layers { name: "loss" ... }
```
A Layer is...

name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
convolution_param {
  num_output: 20
  kernel_size: 5
  stride: 1
  weight_filler {
    type: "xavier"
  }
}
Caffe Supports…

- A complete set of core layers CNNs use
  - convolution
  - ReLU
  - Pooling
  - etc.

- Writing your own layer is easy!
We Hide CPU/GPU Details

SyncedMem allocation + communication
Brewing by the Numbers…

- Speed with Krizhevsky’s 2012 model:
  - 2ms/image with K40 GPU
  - 40 million images / day!
  - 20ms/image with 8-core CPU
- 9k lines of C++ code (20k with unit test)
Overall Work Flow

Custom Datasets (image, audio, etc) → Database (LevelDB, HDF5, etc) → Pretrained Models (platform-independent)

Model Definition (text protobufs) → Instantiated Solver (CPU/GPU)
Supporting Ongoing Research

- R-CNN for object detection
Model Zoo

• Open collection of deep models to share innovation
  • BVLC reference models
  • VGG models
  • Network-in-Network / CCCP models
• Help disseminate and reproduce research!
According to the Citations...

**Decaf**: A deep convolutional activation feature for generic visual recognition

... 3.2. Feature Generalization and Visualization We visualized the model features to gain insight into the semantic capacity of DeCAF and other features that have been typically employed in computer vision. In particular, we compare...

[C] **Caffe**: An open source convolutional architecture for fast feature embedding

Cited by 117
Cited by 56
How does it Solve My Problem?

- Transfer learned models to kick-start new models

Your Task
<table>
<thead>
<tr>
<th>Ethereal</th>
<th>HDR</th>
<th>Melancholy</th>
<th>Minimal</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Ethereal Image" /></td>
<td><img src="image2" alt="HDR Image" /></td>
<td><img src="image3" alt="Melancholy Image" /></td>
<td><img src="image4" alt="Minimal Image" /></td>
</tr>
</tbody>
</table>

Recognizing Styles
(http://vislab.berkeleyvision.org/)
Scene Understanding
(http://places.csail.mit.edu/)

Predictions:

- Type of environment: outdoor
- Semantic categories: skyscraper:0.69, tower:0.16, office_building:0.11,
- SUN scene attributes: man-made, vertical components, natural light, open area, nohorizon, glossy, metal, wire, clouds, far-away horizon
Dec 2013   May   Aug   Nov
(launched)   200   404   745
Community Effort

Fork 745

Visitors

90,356 Views
7,559 Unique visitors
Notable Recent Brews

• cuDNN: fast deep network routines from Nvidia

• Distributed Parallel Training from Flickr
“In open source, we feel strongly that to really do something well, you have to get a lot of people involved.”

– Linus Torvalds
Conclusional Remarks…

✓ Fast
✓ State-of-the-art
✓ Toolkits
✓ Ongoing research
✓ BSD
<table>
<thead>
<tr>
<th>Maximally accurate</th>
<th>Maximally specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>espresso</td>
<td>2.23192</td>
</tr>
<tr>
<td>coffee</td>
<td>2.19914</td>
</tr>
<tr>
<td>beverage</td>
<td>1.93214</td>
</tr>
<tr>
<td>liquid</td>
<td>1.89367</td>
</tr>
<tr>
<td>fluid</td>
<td>1.85519</td>
</tr>
</tbody>
</table>

http://caffe.berkeleyvision.org/
http://github.com/BVLC/caffe/
Brewing a Deeper Understanding of Images

Yangqing Jia