Deep Networks with Applications to Computer Vision and Robotics

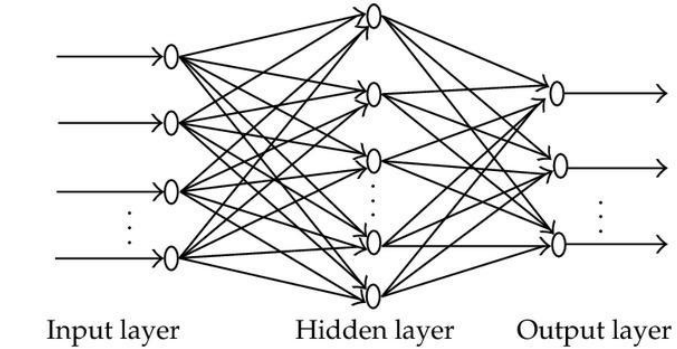
Anelia Angelova, Google

Overview

- Deep networks overview
- **Grasping**: Real-time grasp detection
- **Pedestrian detection**: Large-Field-of-View Deep Network

Deep Networks introduction

A simple neural network

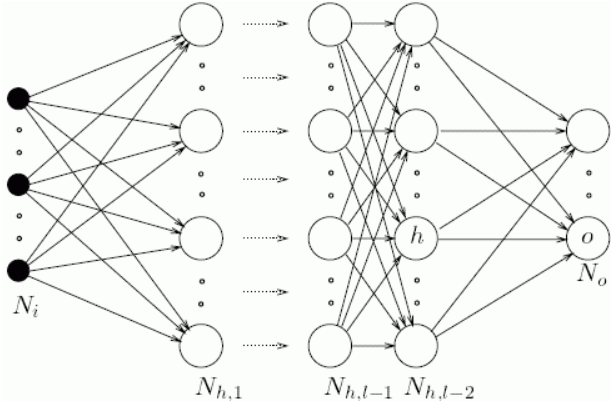


hidden: $h=\varphi_1 (W_1 x+b_1)$

output: $y=\varphi_2 (W_2 h+b_2)$

$y=\varphi_2(W_2\varphi_1 (W_1 x+b_1)+b_2)$

Network with more layers.



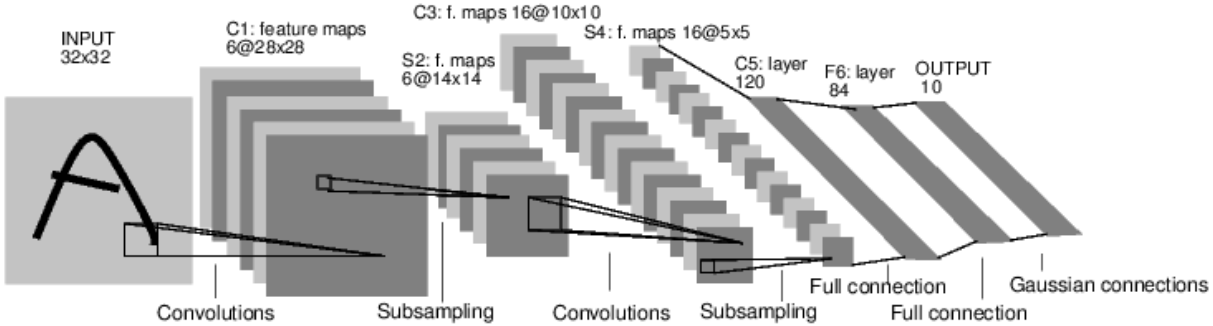
$y=\varphi_L(W_{L-1}\varphi_{L-1}(W_{L-2}\varphi_{L-2}(\ldots\varphi_1(W_1 x))))$

Typically 3-4 layers.
Local minima problems.
Useful for:

- moderate size learning problems
- small vision problems (digit recognition)

Deep Convolutional Networks and Deeper Networks

First convolutional neural network: LeCun'98



Contemporary deep convolutional networks (20+ layers): Szegedy'14, Simonyan'14

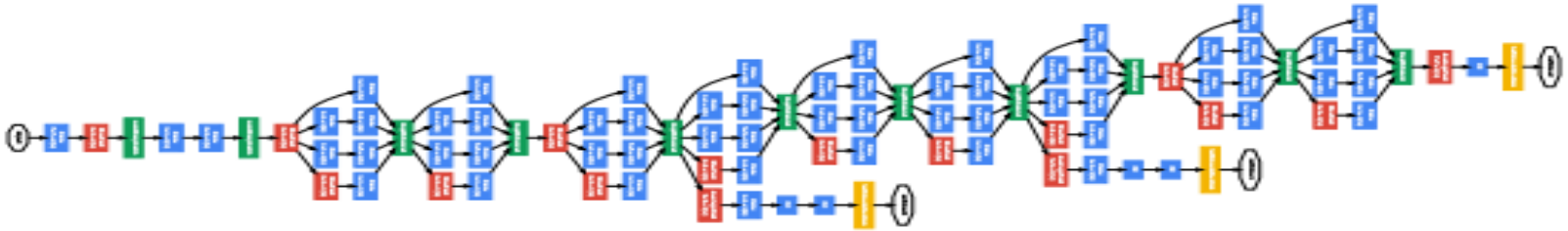


Image credit Szegedy et al'14

Deep Nets in computer vision

Breakthroughs in object recognition

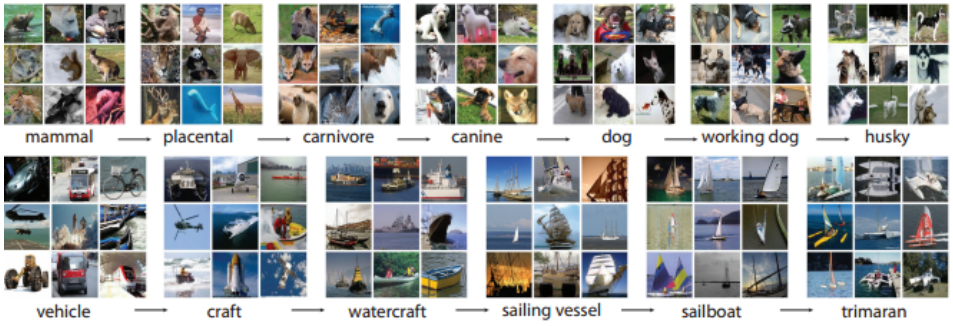


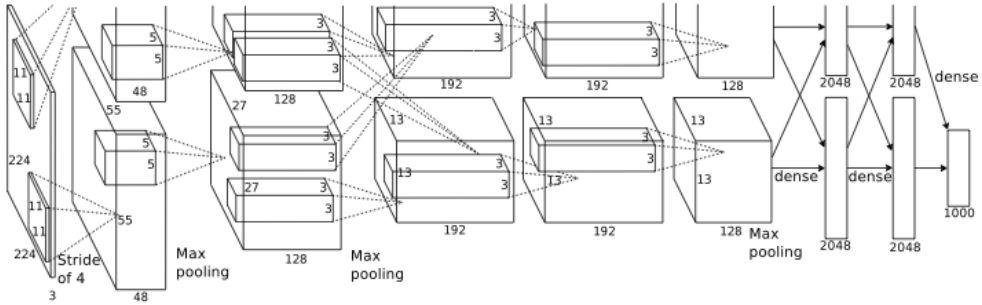
Image credit Deng09

Imagenet: 1000 classes, 1.2M images

Top 5 err, 2012	Top 5 err. 2011
1.Supervision 16.4% 2. ISI 26.2%	1. 25.8%

~10% better!

Krizhevsky et al. 2012



First **deep net** to win Imagenet

Single machine, 2 GPUs

Open source:

https://code.google.com/p/cuda-convnet2

Deep Nets in computer vision: Follow-up

Imagenet object recogniton competitions

Winning Team, Year	Top 5 error (%)	Number of Deep Net approaches / Total
Supervision Kizhevsky et al, 2012	16.4	1 / 6
Clarifai Zeiler & Fergus, 2013	11.7	17 / 24
GoogLe Net Szegedy et al, 2014	**6.66**	31 / 32

Recent results:

Team/Company	Top 5 err. (%)
VGG,Simonyan'14	6.8
Baidu, Wu'15	5.98
Microsoft, He'15	4.94
Google, Ioffe'15	**4.82**

Human: 5.1% (estim)

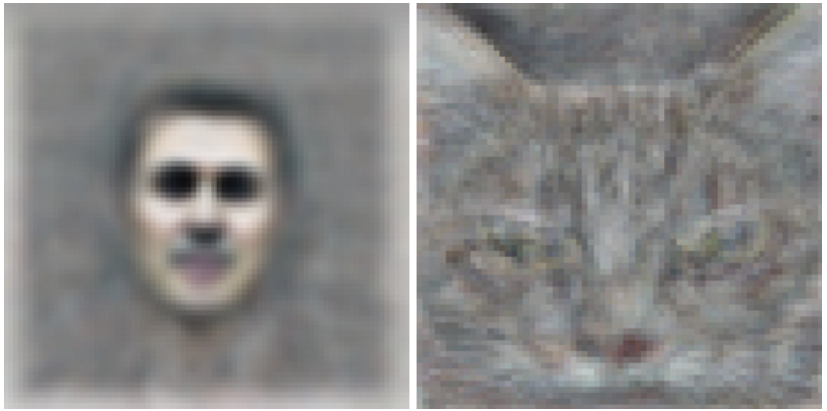
Note: Error was 25.8% in 2011!

All subsequent years : DNN solutions

Wide adoption in industry: Google, Microsoft, Baidu, Apple, Nuance, NVidia, etc integrate deep network solutions

Deep Nets in computer vision

Breakthroughs in object recognition



Le et al. 2012

Large scale deep networks

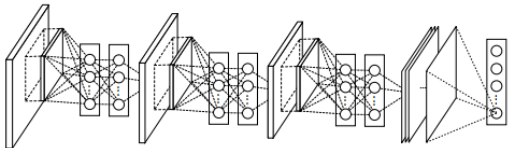
Unsupervised training & emergence of object specific “neurons”

Increase in accuracy of large-scale recognition tasks:

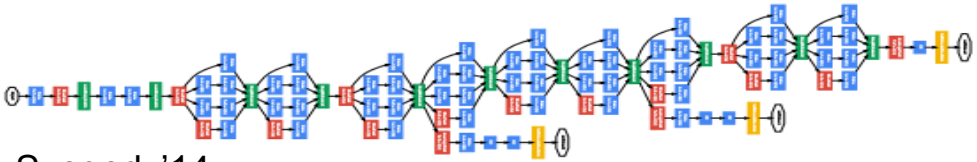
Large-scale Imagenet data	9M images, 10K classes	14M images 22K classes
State-of-the-art	16.7% (Sanchez& Prerronnin’11)	9.3% (Weston’10)
Le et al’12	**19.2%**	**15.8%**

Advances in Detection, Segmentation

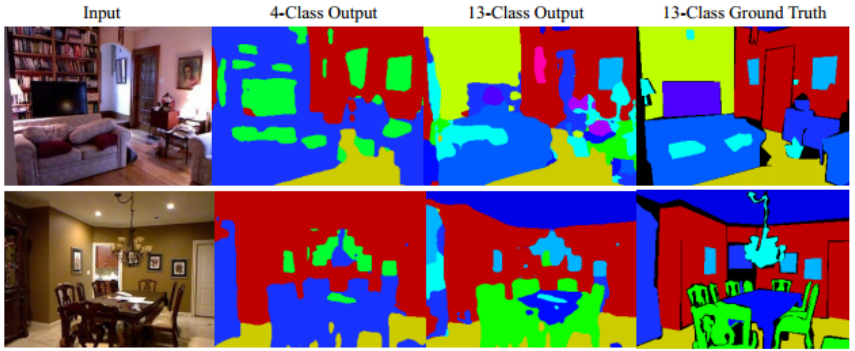
20+ layer nets win localization and detection competitions



Lin'13 NiN (Network in Network)



Szegedy'14



Eigen & Fergus'14, Segmentation

More improvements:

- **deeper nets**
- **more data**
- **ensembles**
- **context**

Others:

Depth map, surface normals: Eigen'14

Human pose: Toshev'13, Tompson'14

Image captioning: Vinyals'14, Fang'14

Kiros'14, etc.

‘Human Level’ Face Recognition

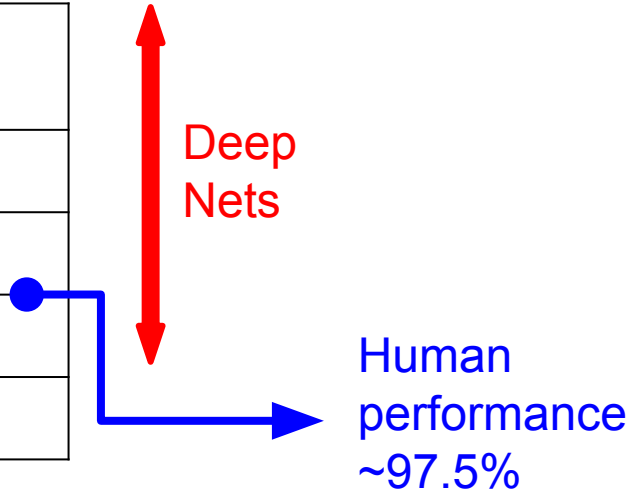
Image credit: Wang’09

Labeled Faces in the Wild

LFW benchmark

NUS-LV* Baidu*	99.70% 99.62%
Face++, Megvii	99.50%
DeepId2+, CUHK	99.47%
DeepFace, Facebook, 2014	97.35%
LBP/SVM	91.37%

*Newest results, no publications yet



Robotics / Autonomous driving applications

Stereo depth estimation:

- Memisevic & Conrad’13
- Zbontar & LeCunn’14

Traffic sign recognition: “Superhuman” Ciresan’12

Robotics:

- DNN-based grasping, Lenz et al’14

Pedestrian detection

Breakthroughs in other domains

Speech

- G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, Deep Neural Networks for Acoustic Modeling in Speech Recognition --- The shared views of four research groups, IEEE Signal Processing Magazine, 2012
- L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related application, 2013

Natural Language Processing

- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representation of Words and Phrases and their Compositionality, NIPS 2013
- I. Sutskever, O. Vinyals, Q. Le, Sequence to Sequence Learning with Neural Networks," NIPS, 2014
- J. Gao, X. He, W. Yih, and L. Deng, Learning Continuous Phrase Representations for Translation Modeling, ACL, 2014

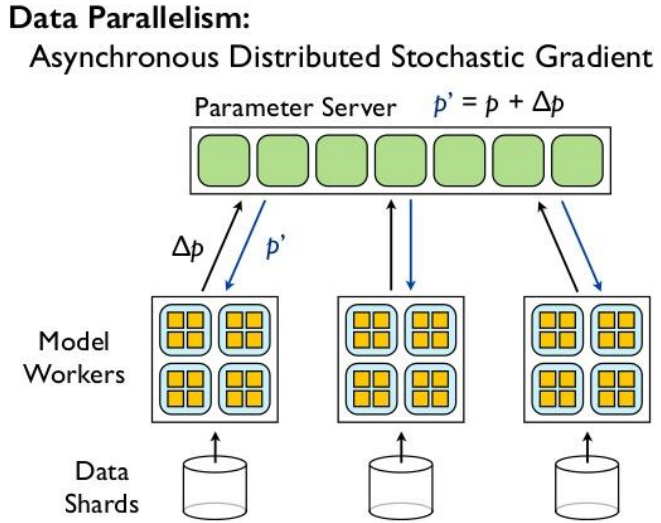
What makes deep networks successful?

→ More complex models

→ Large amounts of data & Computational resources

→ Large scale learning for deep networks

Dean et al'12: DistBelief Distributed DNN learning



Other recent systems:

- Baidu
- Microsoft: Adam
- NUS: Purine
- Yahoo

Why use deep networks?

We are still exploring, but so far DNNs:

- Improve significantly the state-of-the-art
- Shown to have capabilities that are not demonstrated before
- Work across domains

Deep Networks for Grasp Detection

Joseph Redmon, Anelia Angelova

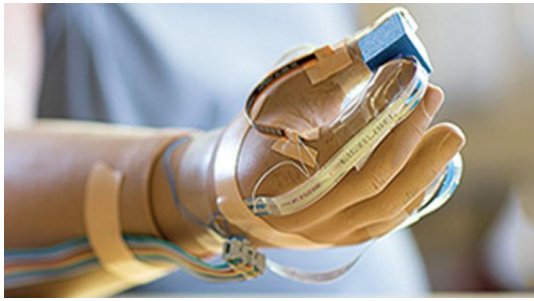
Why grasping?

Potential applications

Manipulation

Cooking

Robo-Chef,Motoman, Yaskawa Electric



Prosthetic applications, fescenter.org

Cleveland Veterans Affairs Medical Center and Case Western Reserve University

Baxter: Rethink Robotics

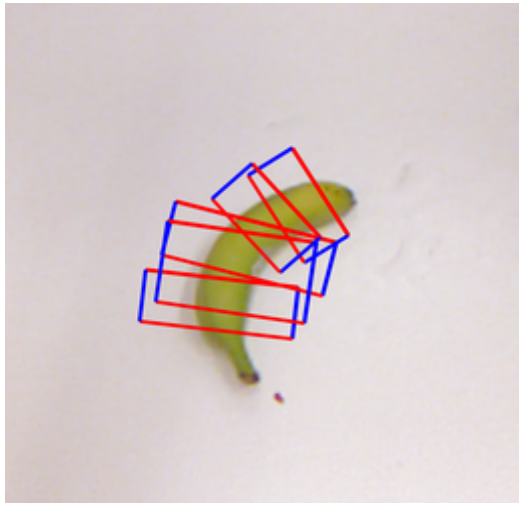
Autonomous checkout Klingbeil'11

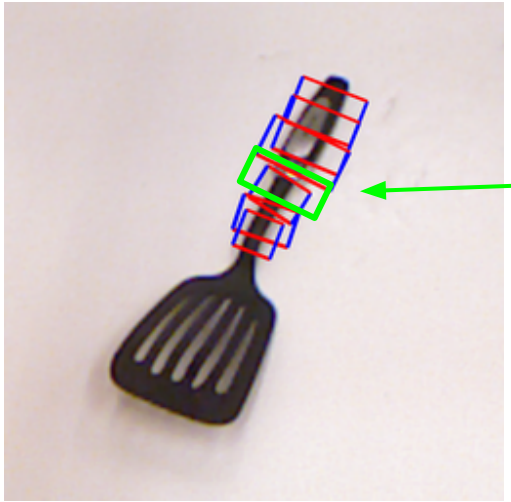
Challenges:

- Detection is still slow, 10+ seconds/image
- Need real-time

Grasp detection setup

Ground truth grasps





Note: Ground truth is not exhaustive. Some viable grasps are not in the data

Grasping rectangle to align robot gripper
Multiple viable grasps

Grasp detection target objects

Objects from Cornell Grasping Dataset

Tools
Household
Food items

===

Varying in:
Sizes
Shapes
Functions

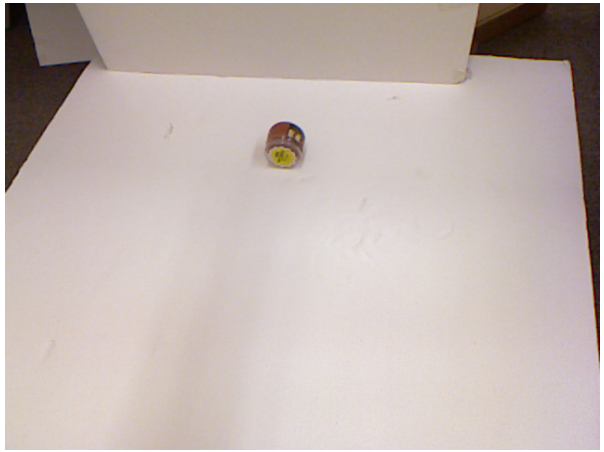
===

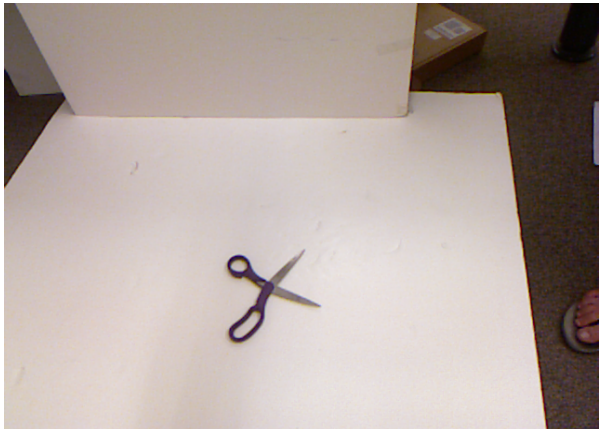
Different grasping strategies

Recognition?

Grasp detection input

RGB + Depth data; Note focus is on **predicting grasping coordinates**, so it is a simplified setup (no



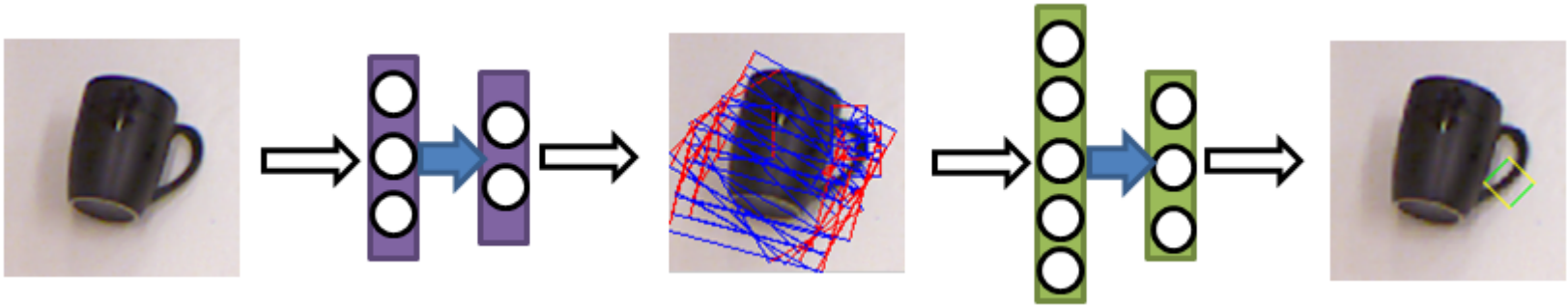
Current state-of-the-art

Sliding window

Cascade of two deep networks

Sample candidate rectangles

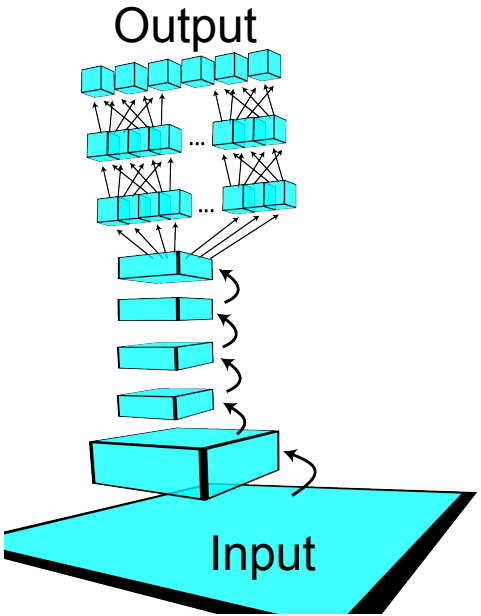
- Runtime: 13 seconds per image!



Lenz et al
2013, 2014

Our approach

Deep NNet
over the full image

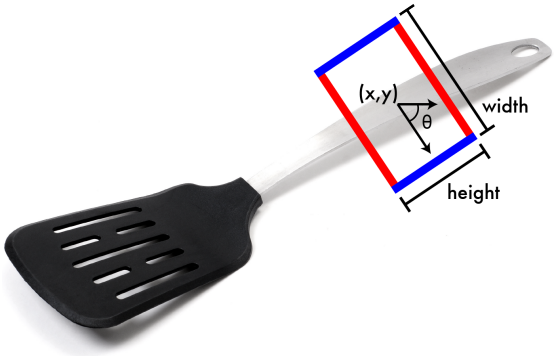


Input

Central square crop

Output

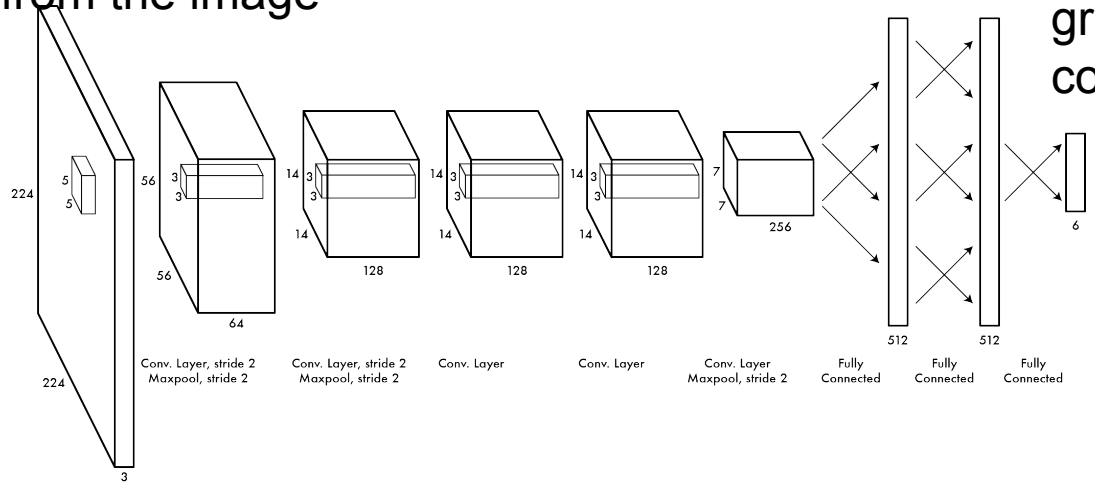
x, y, θ,
Width, Height



Learning / Architecture details

Random crop
from the image

Parameterized
grasp rectangle
coordinates



Architecture after Krizhevsky'12. Differences:

- Fully connected layers are smaller: 512
- We use an L2 squared error cost function

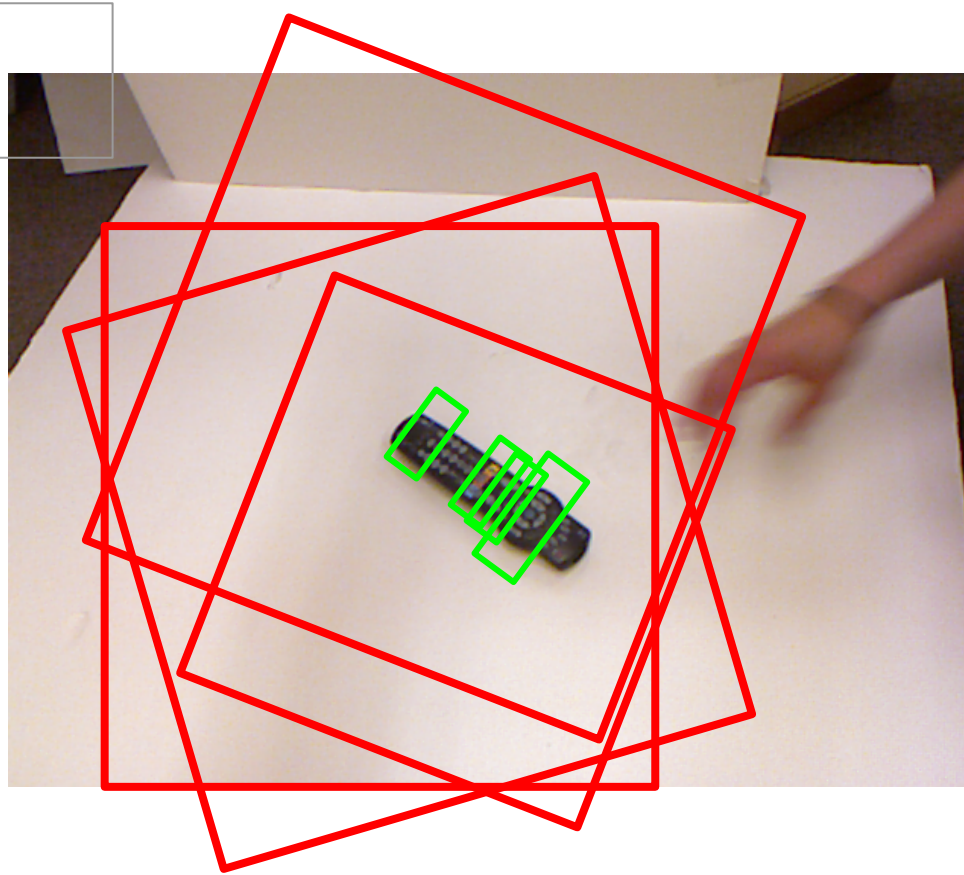
Key:

- **A single network per image**
- **No sliding window**

Training

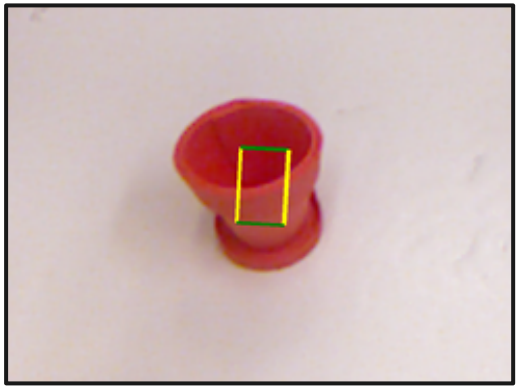
Data augmentation is key

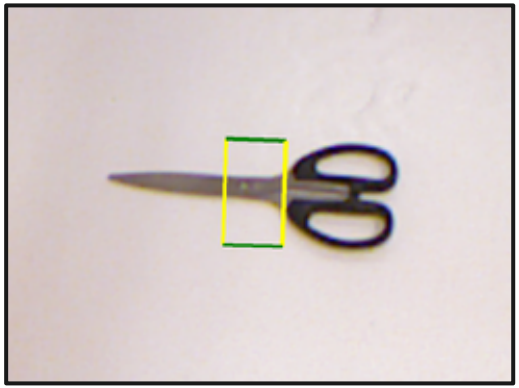
- Multiple crops
- Transformations
  - Rotations
  - Translations
  - Scaling
- Selects grasp at random

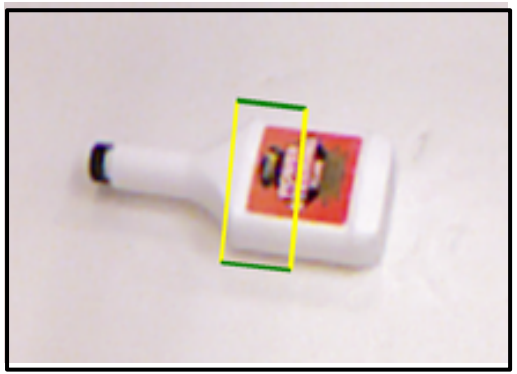


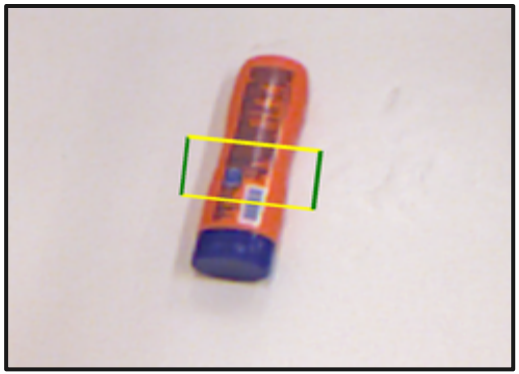
Grasp results

Good grasps







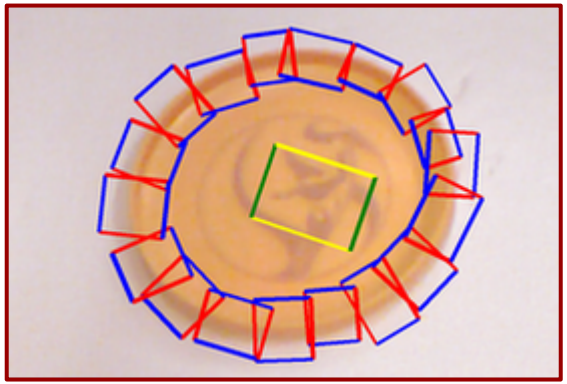


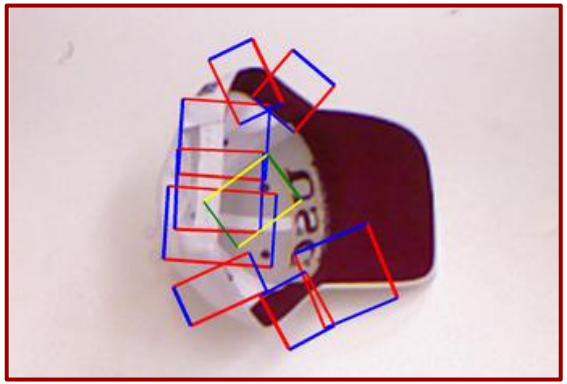
Identifies good grasp rectangle.

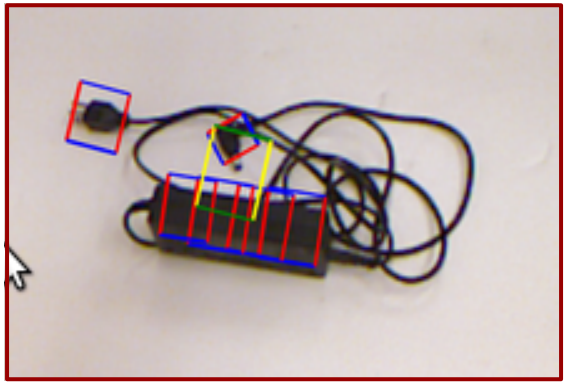
Note:
Has also implicitly detected the object!

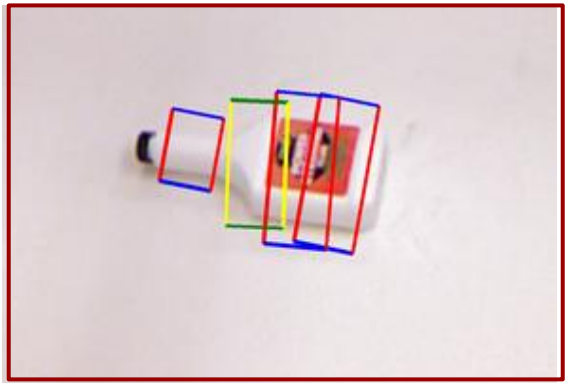
Grasp results

Poor grasps









"Averaging" effect

Training selects any viable grasp

Regresses to one grasp

Problem: Single grasp per image averages out good grasps

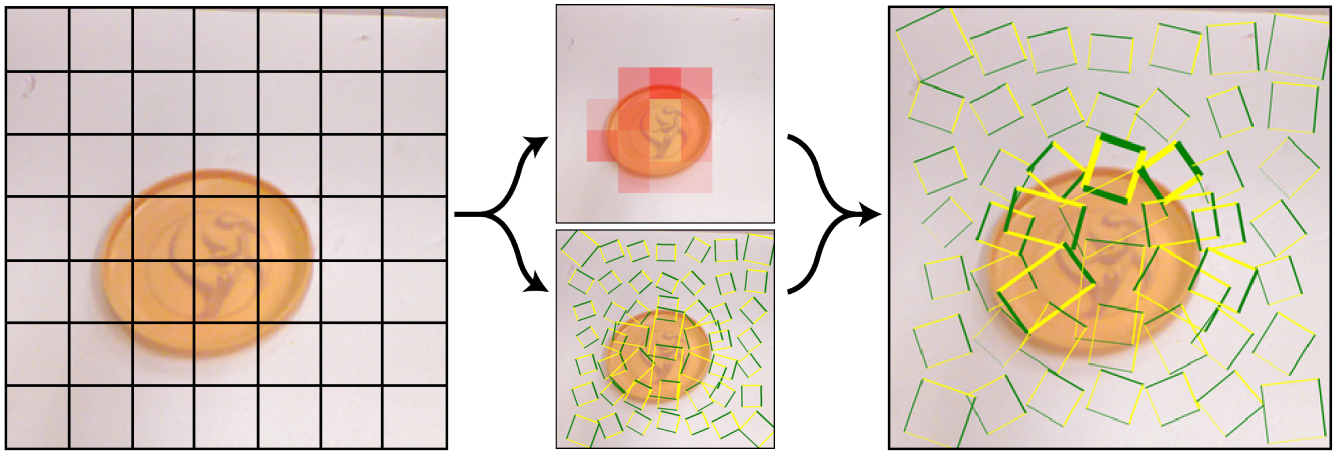
Solution: MultiGrasp

Detect multiple grasps

Define a grid

Each cell predicts a cand. grasp

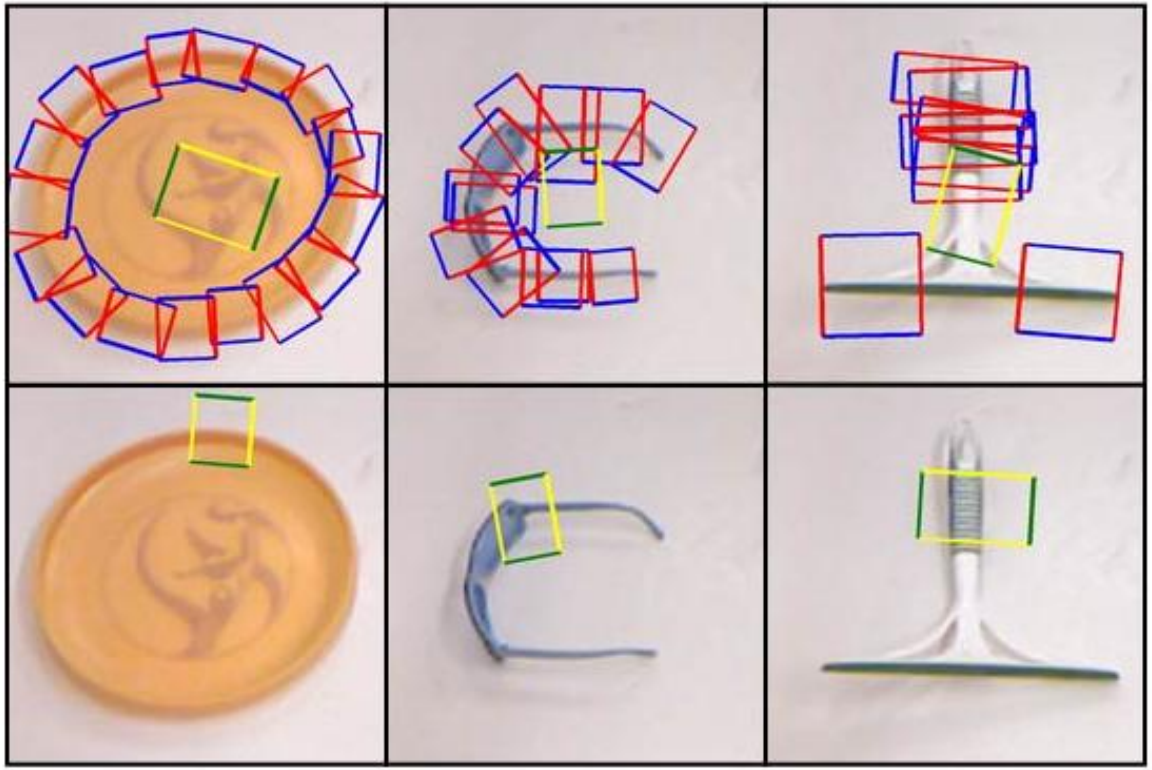
Select highest confidence cells.



Key is a **switching layer**: propagates gradients on max response.

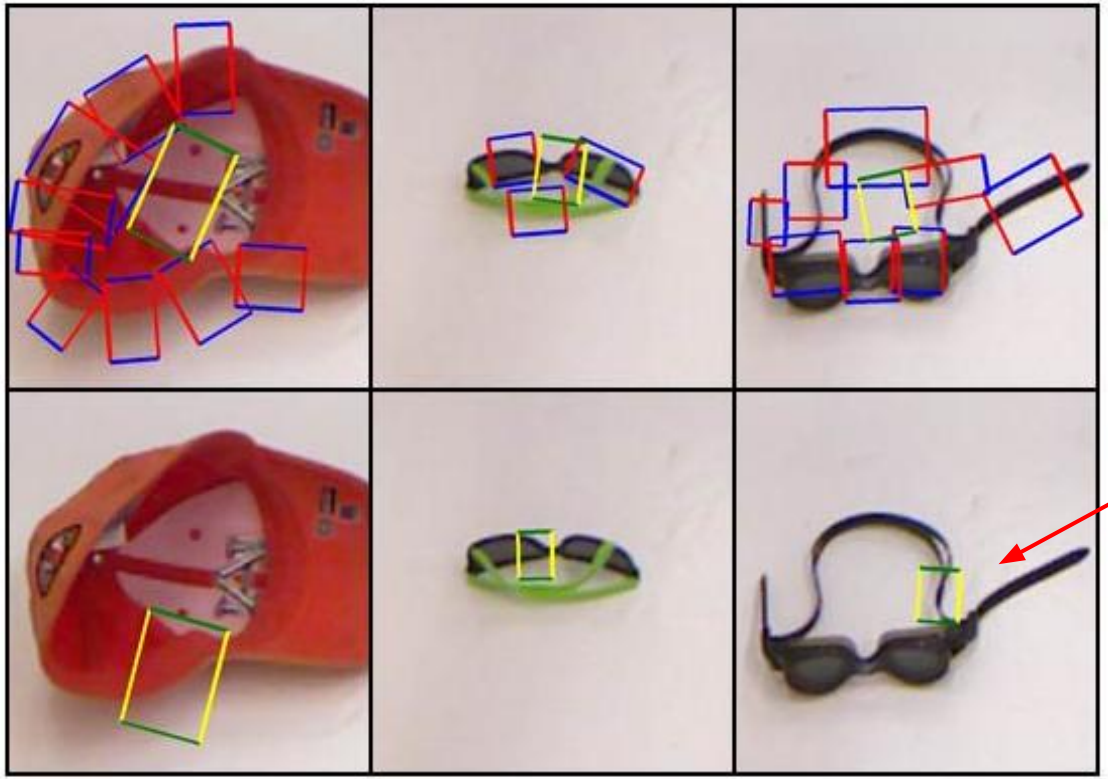
Proposes multiple viable grasp locations.
Done in the same timeframe!

MultiGrasp results



Top row: Results with regression (w. ground truth in red/blue)
Bottom row: Results with MultiGrasp

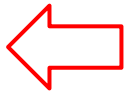
Failure cases for both Regression and MultiGrasp



Missing
ground truth?

Grasp detection results

Algorithm	Accuracy Image-wise split	Improvement
Best previous result Lenz et al'14 (DNN)	73.9%	-
Ours: DNN Regression	**84.4%**	**+10.5%**
Ours: DNN MultiGrasp	**88.0%**	**+14.1%**
W. object classification	**Object classif. accuracy 90%**	



Large improvements in grasp accuracy!

Simultaneous grasp detection and classification.

Grasp detection results: Timing

Algorithm	**Accuracy Image-wise split**	**Time per image**
Lenz et al: 2-stage DNN, sliding window	73.9%	13.5 sec
Deep CNN Regression	**84.4%**	**76 ms**
Deep CNN MultiGrasp	**88.2%**	

Real time 13 FPS / GPU
(1/13 seconds)
170X faster!!

Real-time 13 FPS solution!
Single net per image: can do additional 127 computations.

Summary

New capability:

- 88% grasp detection success
- Simultaneous classification (90%)
- Runtime: 13 FPS for 1/128 compute
- Handles multiple grasps

Previous work:

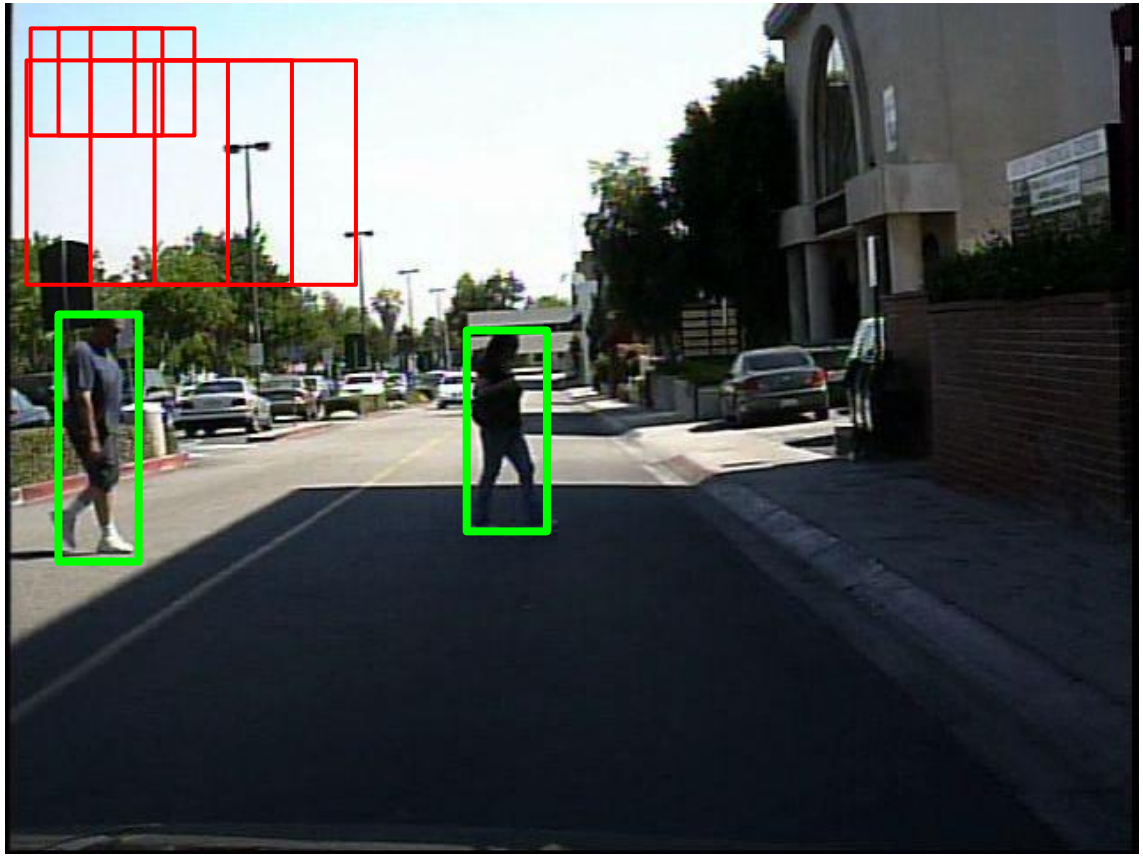
74% detection success
at 0.07 FPS

Key: Single DNN per image, multiple outputs

Large-Field-of-View Deep Network for Pedestrian Detection

Anelia Angelova
Alex Krizhevsky
Vincent Vanhoucke

Pedestrian detection: Sliding window



- Dense sampling
- 100K patches

- Deep nets?
- Slow

Our proposal

Instead of standard sliding window

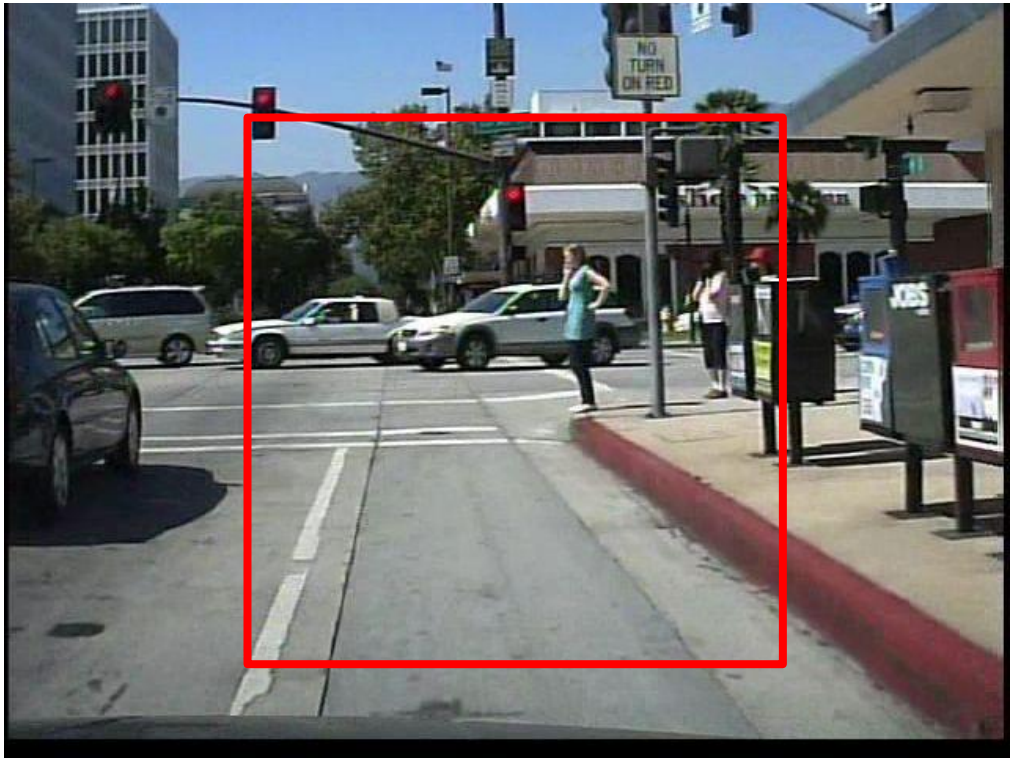
DNN input

DNN output

Pedestrian?

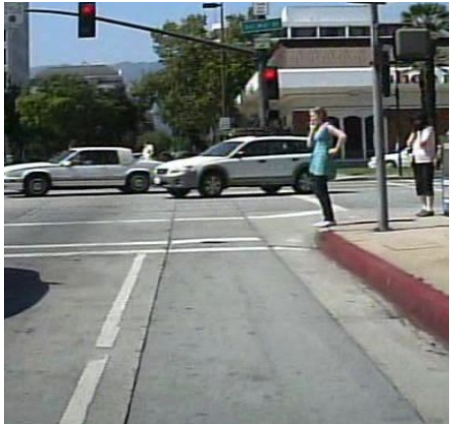
1

Our proposal



We propose:

DNN input



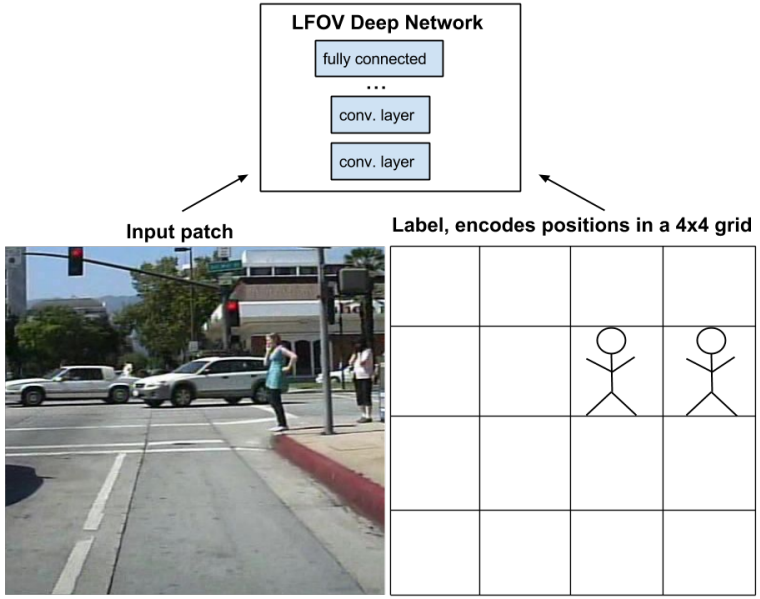
DNN output

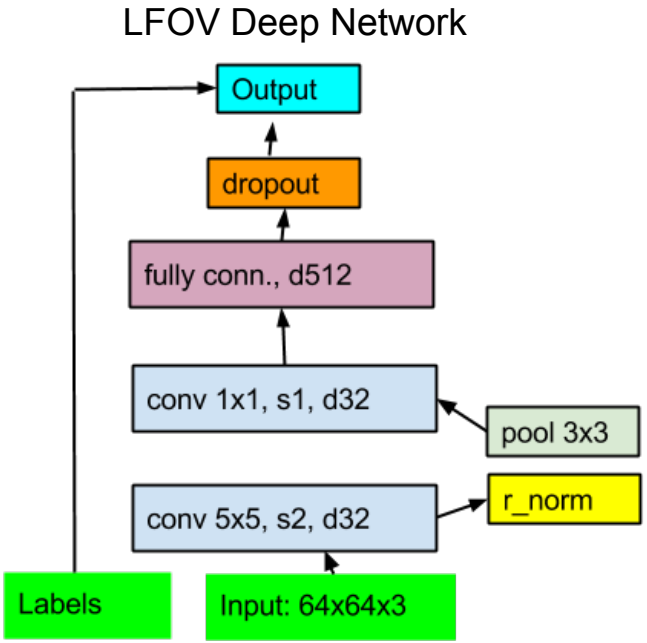
Pedestrian at 4x4

0	0	0	0
0	0	1	1
0	0	0	0
0	0	0	0

Looks at larger area!
Outputs more information

Large Field Of View Deep Network Design



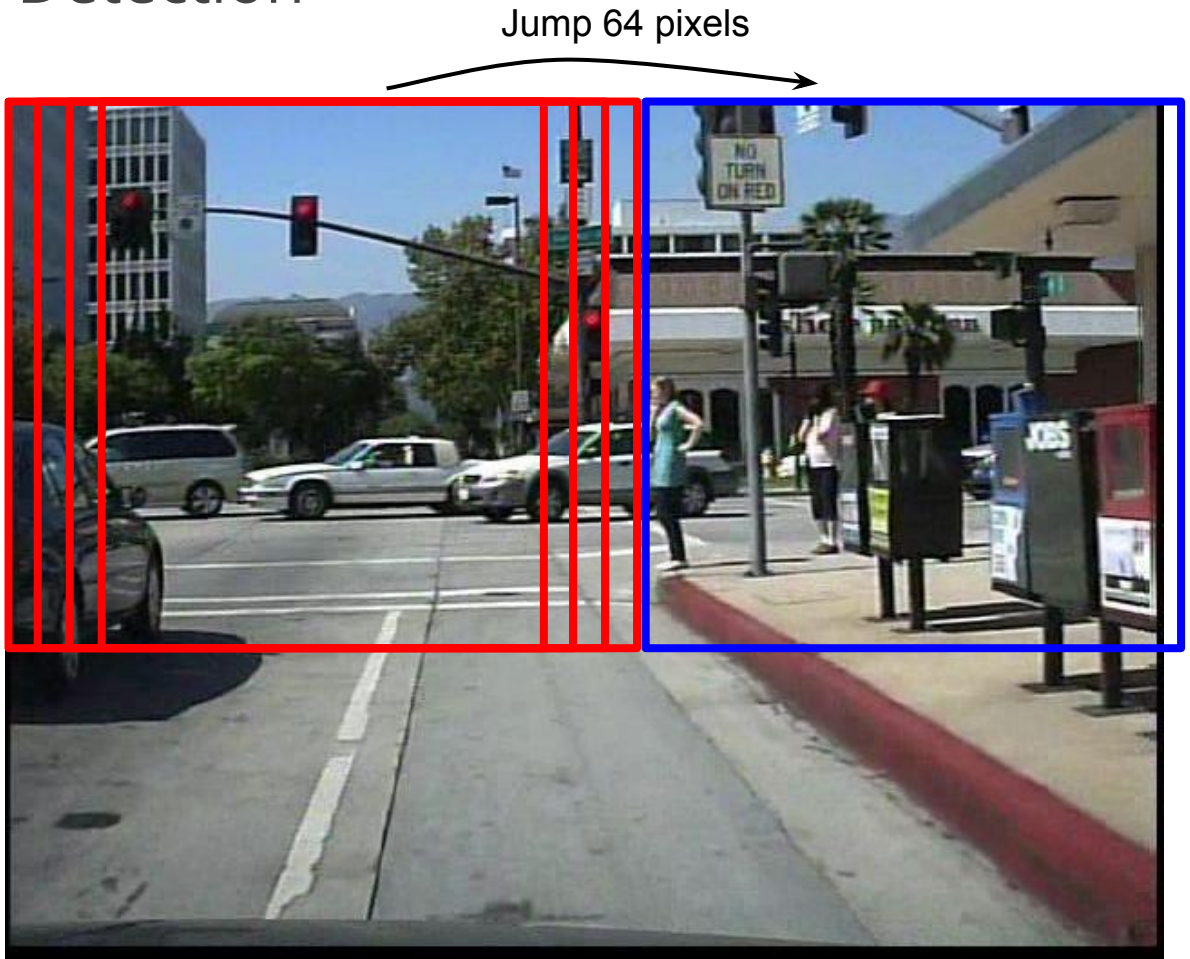


3 layers
Small depths
Input is resized to 64x64
Fast! 3 ms per 128 patches.

Capabilities:

Simultaneous detection
Reuse computations
3.5x faster than 16 same size DNNs

Detection



Still dense sampling
- e.g. every 4 pixels.

Speedup!

Does 1/16 of “sliding”

Data generation for LFOV training

Pedestrians are placed at random 4x4 positions in the grid.

Full pedestrian detection system

LFOV is used as initial proposal system

Three stage classifier:

- Stage 1) LFOV

  Generates proposals

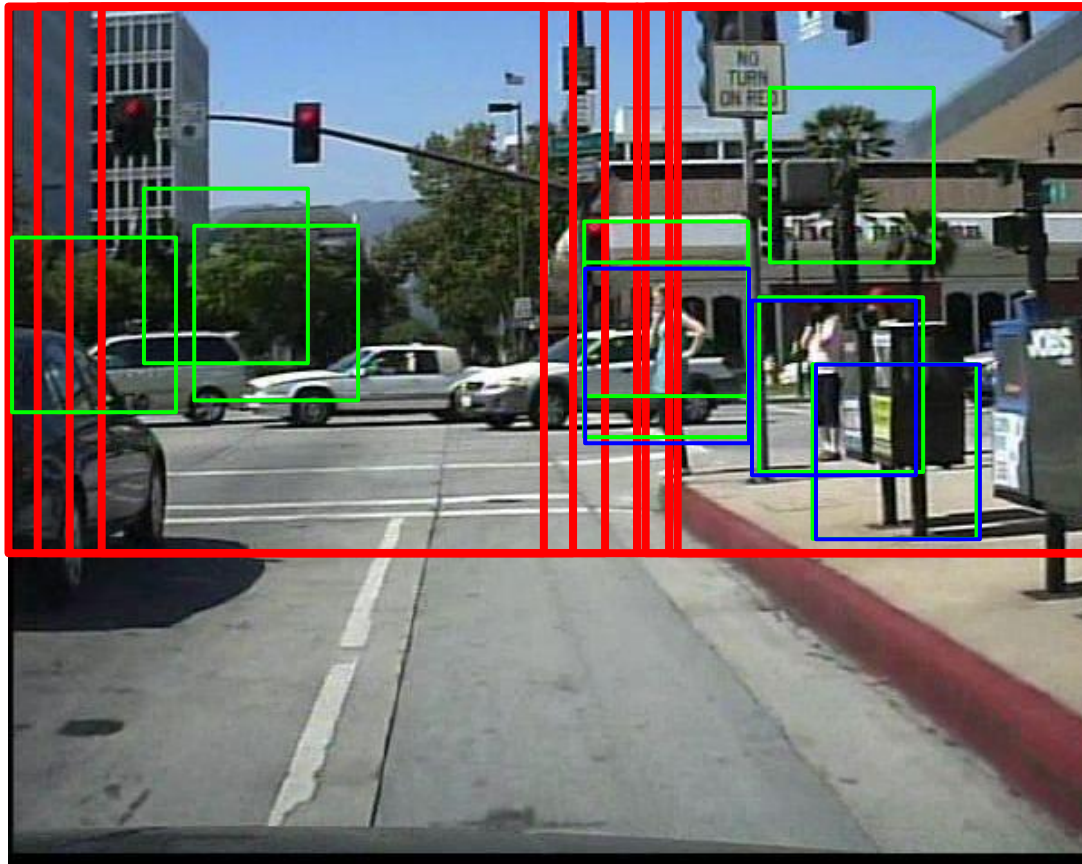
  ~100ms per image

- Stage 2) Individual

  16x16, 3 layer net

- Stage 3) 7 layer DNN

  Slow: 50ms



Results and Timing

Caltech Pedestrian Benchmark, Mean miss rate, Time per image (seconds)

Method	Average miss rate (%)	Timing (seconds per image)	Training dataset
MultiResC [28] (multires)	48.5		Caltech
DBN-Mut [41] (deep)	48.2		Caltech, Inria
Roerei [9]	46.13	1	Inria
MOCO[42]	45.5		Caltech
MultiSDP [11] (deep, w. context)	45.4		Caltech, Inria + Context
WordChannels [40] (multires)	42.3	0.06 (GPU)	Caltech, Inria
MT-DPM [29]	40.5	1	Caltech
JointDeep[10] (deep)	39.3		Caltech, Inria
SDN[12] (deep)	37.9	1-1.5 (GPU)	Caltech, Inria
MT-DPM+Context [29] (w. context)	37.64		Caltech + Context
ACF+SDt [43] (w. motion)	37.3		Caltech + Motion
InformedHaar [38]	34.6	1.6	Caltech, Inria
Ours (LFOV, deep)	35.85	0.28 (GPU)	Caltech, Pretr.
Ours (LFOV-2St, deep)	35.31	0.55 (GPU)	Caltech, Pretr.

*Further impovements: 0.29, 0.22 with motion Paisitkriangkrai'14

Speedup

	CPU	GPU
Speed (seconds)	122	**0.22-0.28**
Speedup	63	83-108

- Our pedestrian detection is state of the art, @220-280ms
- LFOV as a proposal generation system is ~100ms
- It is better and 3-5x faster than prior DNN approaches
- Pedestrian detection works 63-108x faster

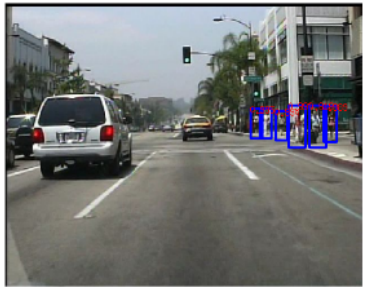
Summary

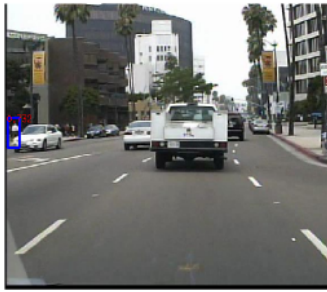
LFOV does multiple detections simultaneously

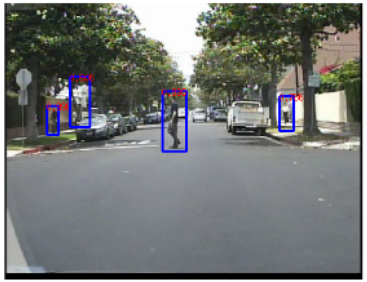
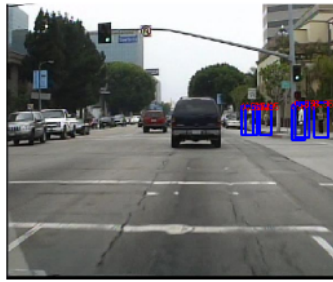
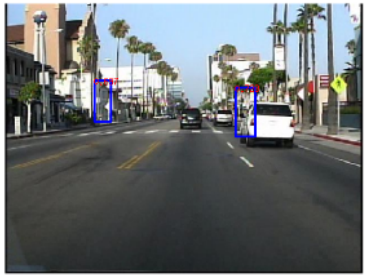
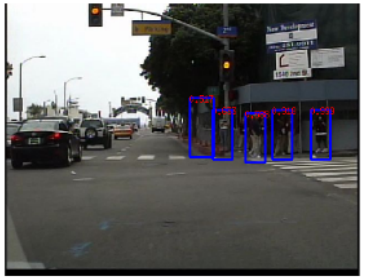
Works at:

- 100ms for proposals
- 280ms total 3.6FPS!
- 63-100x speedup

Obtains accurate detection






Summary: Here we showed

- Although DNN models are slow, they can be made fast!
- How? DNN are high capacity models can do/detect more things. We showcased:

  → Real-time grasp-detection solution

    - grasping pose + object location + classification
    - @13 FPS. x170 faster, +14% better; 88% detection, 90% classification

  → Large-field-of-view DNN

    - 60 x speedup in pedestrian detection, state-of-the-art accuracy.

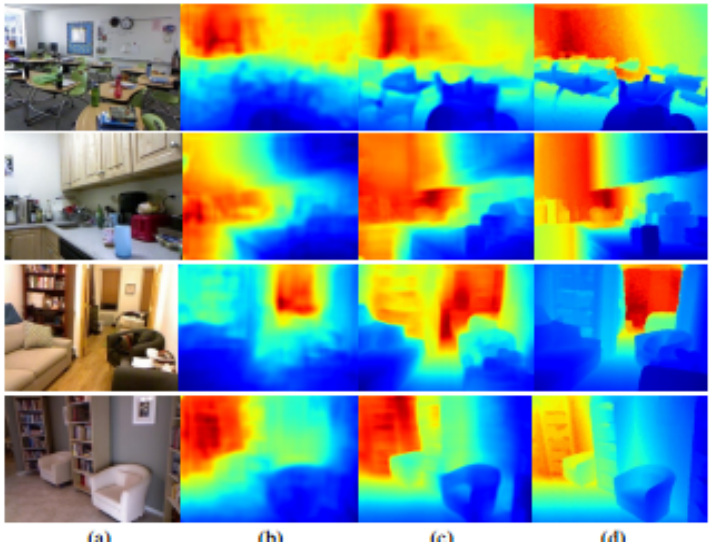
- High accuracy & speed are very important!

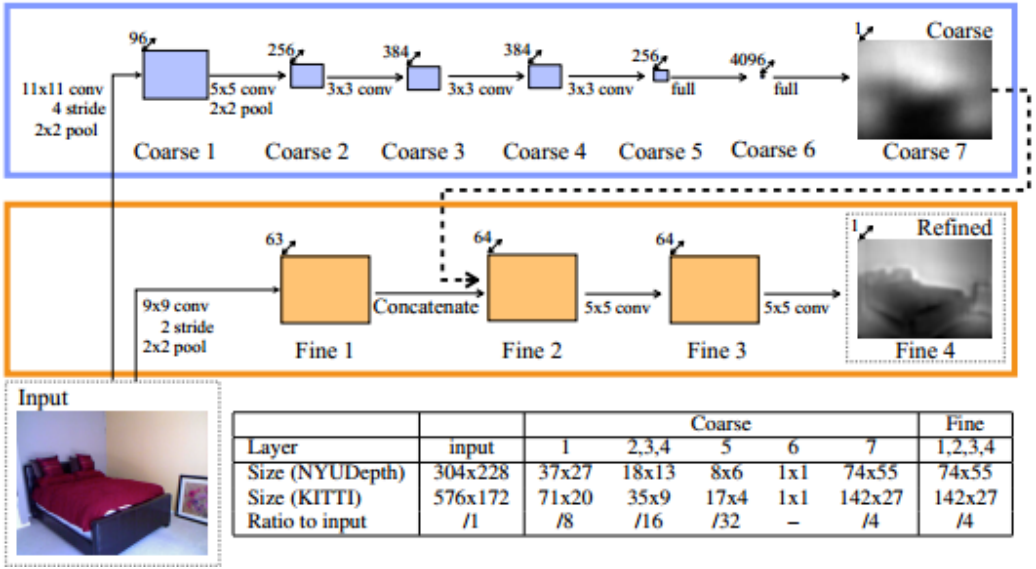
Thank you!

Supplementary materials

Depth map prediction

Depth map, normals and labels: Eigen & Fergus'14





Layer	input	Coarse 1	Coarse 2,3,4	Coarse 5	Coarse 6	Coarse 7	Fine 1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

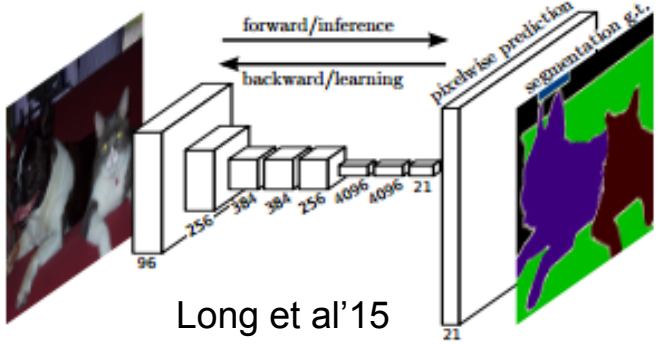
Others:

Human pose estimation: Toshev & Szegedy'13, Tompson et al'14

Image captioning: Vinyals'14, Fang et al'14, Kiros et al'14

Texture recognition: Byueon et al'14

Segmentation



Long et al'15

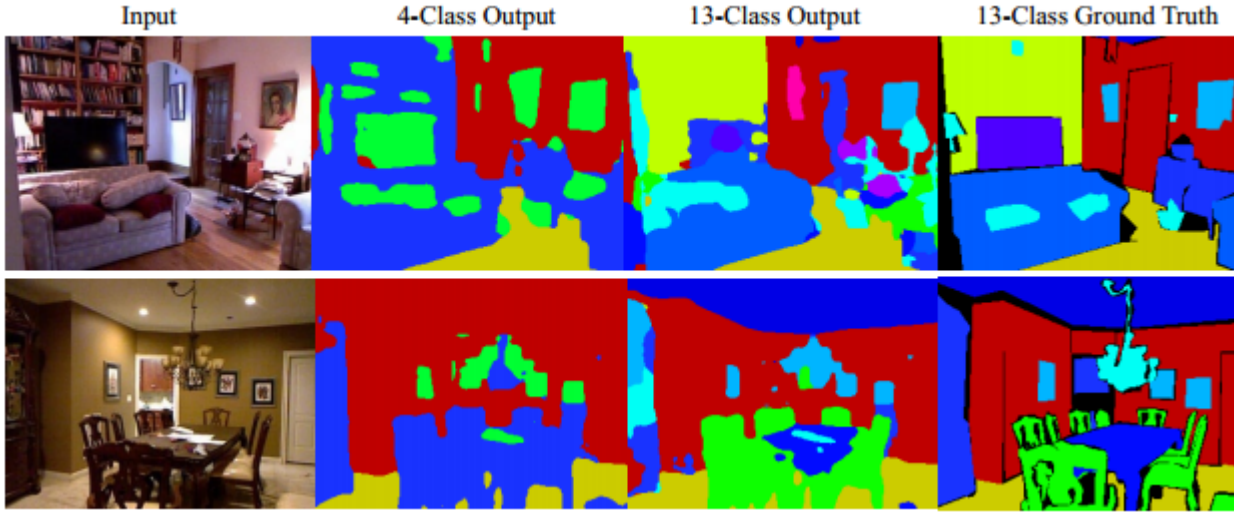


Image credit: Eigen et al'14

Learn regression at **pixel level!**
Handles multiple classes: >40.

Eigen et al'14 Multi-scale DNN

Gupta et al'14 Normalization

Long et al'15 Convolutional

Chen'15 With CRFs

Advances in Recognition and Detection

More Imagenet results:

Team/Approach	Task/Relevant metric (%)
NUS, 3NiN+Context	Detection: 37.21
GoogLeNet, Inception+Ensemble+Data	Detection+: 43.9
VGG, Simonyan'14, 20 layer net+Ensemble	Localization: 25.3

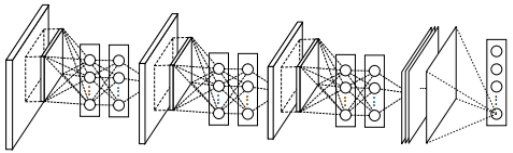
Post-competition: Detection **45%** DeepID (Ouyang'14)

Main takeaways:
deeper nets, more data, ensembles, context

Other works:

Girshick'14 R-CNN

Sermanet'13 Overfeat



Lin'13 NiN (Network in Network)

Szegedy'14, Inception