Convolutional Neural Networks

Deep Learning Lecture 5

Samuel Cheng

School of ECE
University of Oklahoma

Spring, 2017

Table of Contents

Presentation starting next week!

Date	Student	Package
2/24	Aakash	Tensorflow
	Amed	Tensorflow
3/3	Soubhi	Tensorflow
3/10	Ahmad A	Theano
	Tamer	Theano
3/24	Ahmad M	Keras
	Obada	Keras
4/3	Muhanad	Caffe
	Siraj	Caffe
4/10	Dong	Torch
	Varun	Lasagne
4/17	Naim	MatConvNet

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- Rate your classmates' presentation according to
  - How much did you learn from the presentation
  - How much effort does the speaker put into
- A simple 1-5 rating, 5 is the best. For example,
  - If you think you have learned a lot (*assuming that you know nothing at first but only materials from previous presentations*) and you think the speaker has put lots of effort, then give a 5
  - If you think it is just average to you but you feel the speaker has put lots of effort on that, give a 4
  - If you think the presentation is quite useless but you still think the speaker put some (but not a lot) effort on that, give a 2

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- Rate your classmates' presentation according to
  - How much did you learn from the presentation
  - How much effort does the speaker put into
- A simple 1-5 rating, 5 is the best. For example,
  - If you think you have learned a lot (*assuming that you know nothing at first but only materials from previous presentations*) and you think the speaker has put lots of effort, then give a 5
  - If you think it is just average to you but you feel the speaker has put lots of effort on that, give a 4
  - If you think the presentation is quite useless but you still think the speaker put some (but not a lot) effort on that, give a 2

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- Rate your classmates' presentation according to
  - How much did you learn from the presentation
  - How much effort does the speaker put into
- A simple 1-5 rating, 5 is the best. For example,
  - If you think you have learned a lot (*assuming that you know nothing at first but only materials from previous presentations*) and you think the speaker has put lots of effort, then give a 5
  - If you think it is just average to you but you feel the speaker has put lots of effort on that, give a 4
  - If you think the presentation is quite useless but you still think the speaker put some (but not a lot) effort on that, give a 2

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- You will be asked to give the rating right after the presentation (don't worry though, the speaker won't see your vote)
- Vote seriously, "outlier" vote will be counted
  - Your vote is considered outlier if it is farthest from the mean (you are okay if at least three other votes are the same as you tho)
  - Absent vote will be counted as outlier as well
  - For every 4 outlier votes, your rating will be deducted by 0.5
- Your final score will be curved. With 75% as the mean and 20% as standard deviation. (max 100% and min 0% tho)
- If you don't show up for presentation, you will score nothing. If you absolutely cannot make it, please be ready with a very good excuse

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- You will be asked to give the rating right after the presentation (don't worry though, the speaker won't see your vote)
- Vote seriously, "outlier" vote will be counted
  - Your vote is considered outlier if it is farthest from the mean (you are okay if at least three other votes are the same as you tho)
  - Absent vote will be counted as outlier as well
  - For every 4 outlier votes, your rating will be deducted by 0.5
- Your final score will be curved. With 75% as the mean and 20% as standard deviation. (max 100% and min 0% tho)
- If you don't show up for presentation, you will score nothing. If you absolutely cannot make it, please be ready with a very good excuse

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- You will be asked to give the rating right after the presentation (don't worry though, the speaker won't see your vote)
- Vote seriously, "outlier" vote will be counted
  - Your vote is considered outlier if it is farthest from the mean (you are okay if at least three other votes are the same as you tho)
  - Absent vote will be counted as outlier as well
  - For every 4 outlier votes, your rating will be deducted by 0.5
- Your final score will be curved. With 75% as the mean and 20% as standard deviation. (max 100% and min 0% tho)
- If you don't show up for presentation, you will score nothing. If you absolutely cannot make it, please be ready with a very good excuse

Presentation starting next week!

	student	packages
0	aakash	tensorflow
1	amed	tensorflow
2	soubhi	tensorflow
3	ahmad_a	theano
4	tamer	theano
5	ahmad_m	keras
6	obada	keras
7	muhanad	caffe
8	siraj	caffe
9	dong	torch
10	varun	lasagne
11	naim	matconvnet

- You will be asked to give the rating right after the presentation (don't worry though, the speaker won't see your vote)
- Vote seriously, "outlier" vote will be counted
  - Your vote is considered outlier if it is farthest from the mean (you are okay if at least three other votes are the same as you tho)
  - Absent vote will be counted as outlier as well
  - For every 4 outlier votes, your rating will be deducted by 0.5
- Your final score will be curved. With 75% as the mean and 20% as standard deviation. (max 100% and min 0% tho)
- If you don't show up for presentation, you will score nothing. If you absolutely cannot make it, please be ready with a very good excuse

Presentation bonuses

- Instructor vote counts double tentatively
- Auditor votes (Niki and Nishaal)?
- First prize: 5% of the whole course
- Second prize: 3% of the whole course

Logistics

- Quiz 1 is due today
  - 5% per day penalty (of Quiz 1) starting tomorrow. Assignment won't be accepted after next Friday
- HW 2 was posted and will be due in two weeks
  - 3% bonus for the first correct submitter
  - As the winner of HW 1, Naim is out for this round
- More about grading

Overall percentage	Grade
> 80	A
$60 - 80$	B
$40 - 60$	C
< 40	D

Very unlikely to get below B provided that you finish all assignments (reasonably) on time

Logistics

- Quiz 1 is due today
  - 5% per day penalty (of Quiz 1) starting tomorrow. Assignment won't be accepted after next Friday
- HW 2 was posted and will be due in two weeks
  - 3% bonus for the first correct submitter
  - As the winner of HW 1, Naim is out for this round
- More about grading

Overall percentage	Grade
> 80	A
$60-80$	B
$40-60$	C
< 40	D

Very unlikely to get below B provided that you finish all assignments (reasonably) on time

Logistics

- Quiz 1 is due today
  - 5% per day penalty (of Quiz 1) starting tomorrow. Assignment won't be accepted after next Friday
- HW 2 was posted and will be due in two weeks
  - 3% bonus for the first correct submitter
  - As the winner of HW 1, Naim is out for this round
- More about grading

Overall percentage	Grade
> 80	A
$60 - 80$	B
$40 - 60$	C
< 40	D

Very unlikely to get below B provided that you finish all assignments (reasonably) on time

Logistics

- Tensorflow 1.0 is now available in the OU supercomputer schooner
- Request account at `http://www.ou.edu/content/oscer/support/accounts/new_account.html`
- Use the group name **ouecedeeplrn**
- Try “module load TensorFlow” to access it
- Presenters: please try it out :)

Overview

- We talked about the basics of CNNs last week
- We will look into several applications of CNNs besides image recognition
  - Object localization
  - Object detection
- How to visualize a CNN
- CNNs and arts
- Fooling a CNN

Overview

- We talked about the basics of CNNs last week
- We will look into several applications of CNNs besides image recognition
  - Object localization
  - Object detection
- How to visualize a CNN
- CNNs and arts
- Fooling a CNN

Overview

- We talked about the basics of CNNs last week
- We will look into several applications of CNNs besides image recognition
  - Object localization
  - Object detection
- How to visualize a CNN
- CNNs and arts
- Fooling a CNN

Overview

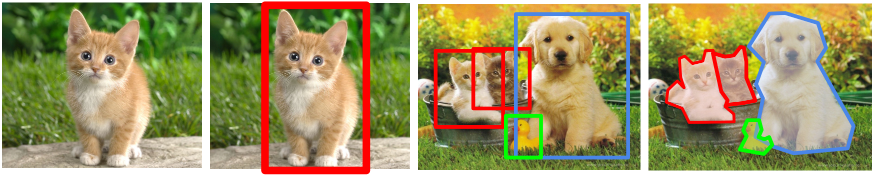
- We talked about the basics of CNNs last week
- We will look into several applications of CNNs besides image recognition
  - Object localization
  - Object detection
- How to visualize a CNN
- CNNs and arts
- Fooling a CNN

Overview

- We talked about the basics of CNNs last week
- We will look into several applications of CNNs besides image recognition
  - Object localization
  - Object detection
- How to visualize a CNN
- CNNs and arts
- Fooling a CNN

Computer Vision Tasks

Classification	Classification + Localization	Object Detection	Instance Segmentation



CAT	CAT	CAT, DOG, DUCK	CAT, DOG, DUCK

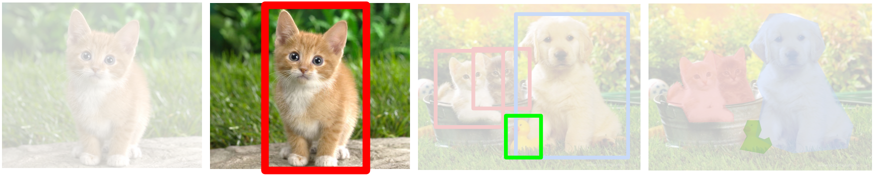
Single object

Multiple objects

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 8 1 Feb 2016

Computer Vision Tasks

Classification | Classification + Localization | Object Detection | Instance Segmentation



Classification + Localization: Task

Classification: C classes
Input: Image
Output: Class label
Evaluation metric: Accuracy

→ CAT

Localization:
Input: Image
Output: Box in the image (x, y, w, h)
Evaluation metric: Intersection over Union

→ (x, y, w, h)

Classification + Localization: Do both

ImageNet localization challenge

Classification + Localization: ImageNet

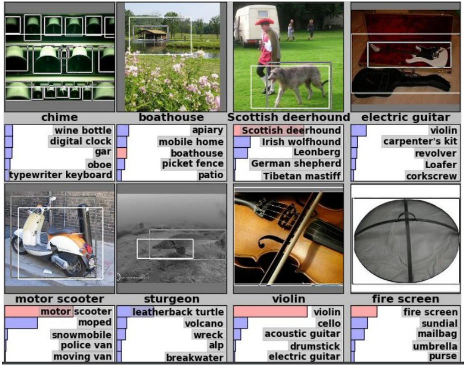
1000 classes (same as classification)

Each image has 1 class, at least one bounding box

~800 training images per class

Algorithm produces 5 (class, box) guesses

Example is correct if at least one one guess has correct class AND bounding box at least 0.5 intersection over union (IoU)



Krizhevsky et. al. 2012

IoU explain

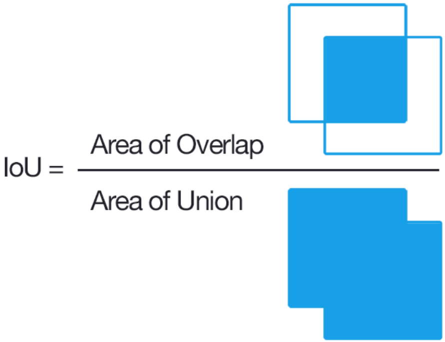


Image from http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Localization as regression

Idea #1: Localization as Regression

Input: image

Neural Net →

Output:
Box coordinates
(4 numbers)

Correct output:
box coordinates
(4 numbers)

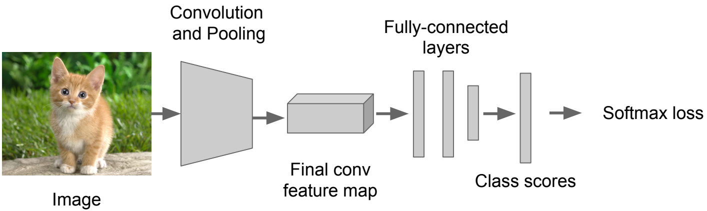
Loss:
L2 distance

Only one object,
simpler than detection

Localization as regression

Simple Recipe for Classification + Localization

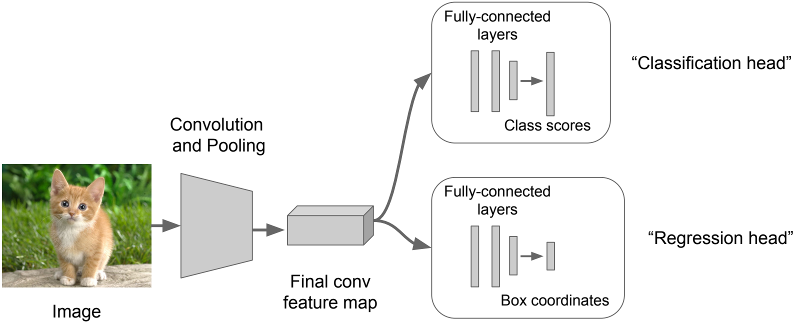
Step 1: Train (or download) a classification model (AlexNet, VGG, GoogLeNet)



Localization as regression

Simple Recipe for Classification + Localization

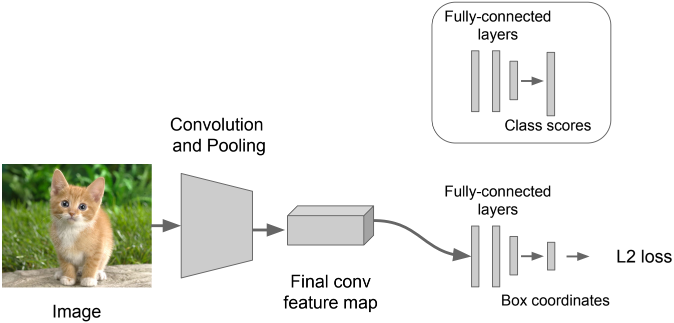
Step 2: Attach new fully-connected "regression head" to the network



Localization as regression

Simple Recipe for Classification + Localization

Step 3: Train the regression head only with SGD and L2 loss

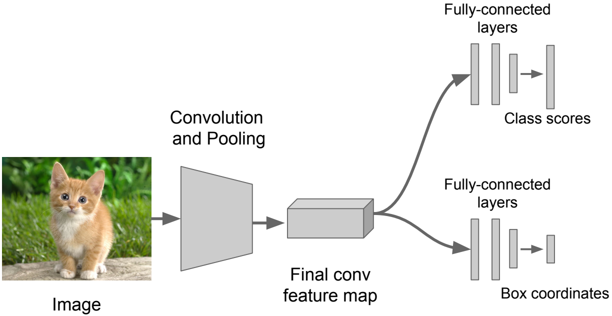


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 15 1 Feb 2016

Localization as regression

Simple Recipe for Classification + Localization

Step 4: At test time use both heads

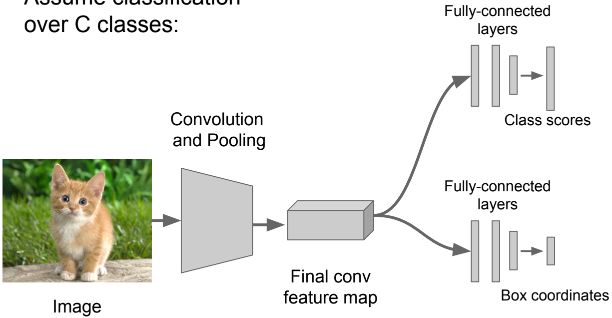


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 16 1 Feb 2016

Localization as regression

Per-class vs class agnostic regression

Assume classification
over C classes:



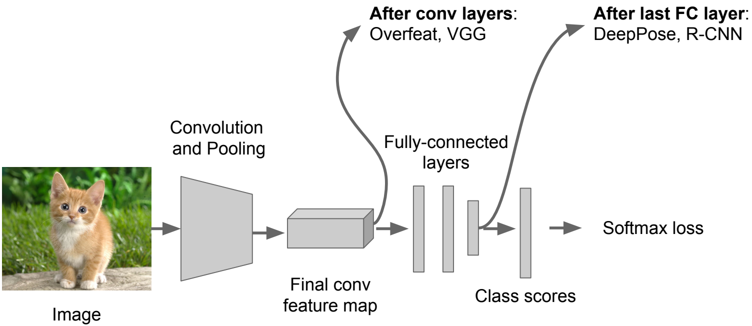
Classification head:
C numbers
(one per class)

Class agnostic:
4 numbers
(one box)

Class specific:
C x 4 numbers
(one box per class)

Localization as regression

Where to attach the regression head?



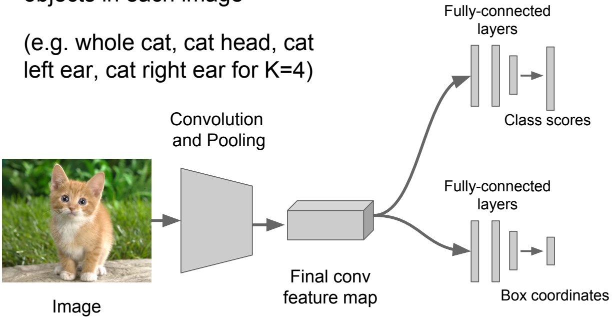
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 18 1 Feb 2016

Localization as regression

Aside: Localizing multiple objects

Want to localize **exactly** K objects in each image

(e.g. whole cat, cat head, cat left ear, cat right ear for K=4)



K x 4 numbers
(one box per object)

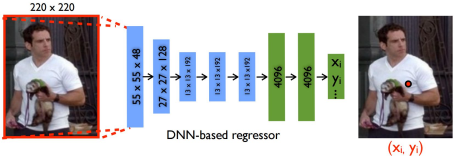
Localization as regression

Aside: Human Pose Estimation

Represent a person by K joints

Regress (x, y) for each joint from last fully-connected layer of AlexNet

(Details: Normalized coordinates, iterative refinement)



Toshev and Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks”, CVPR 2014

Localization as regression

Localization as Regression

Very simple

Think if you can use this for projects

Sliding windows

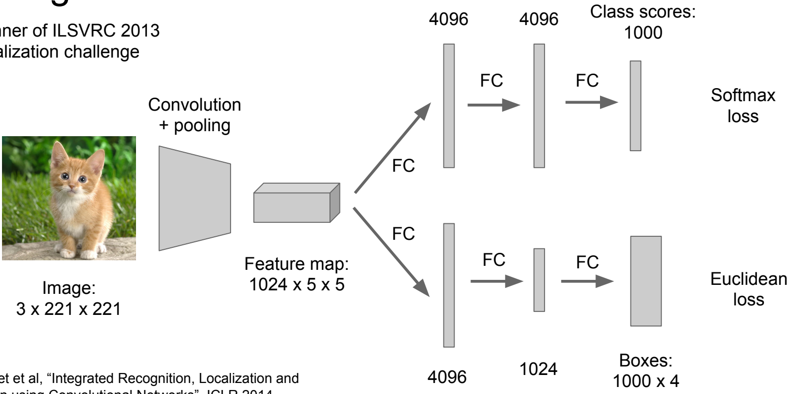
Idea #2: Sliding Window

- Run classification + regression network at multiple locations on a high-resolution image
- Convert fully-connected layers into convolutional layers for efficient computation
- Combine classifier and regressor predictions across all scales for final prediction

Sliding windows

Sliding Window: Overfeat

Winner of ILSVRC 2013
localization challenge



Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Sliding windows

Sliding Window: Overfeat

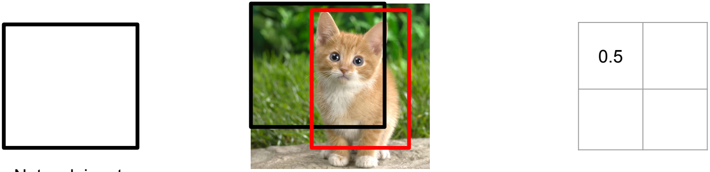
Network input:
3 x 221 x 221

Larger image:
3 x 257 x 257

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 8 - 24   1 Feb 2016

Sliding windows

Sliding Window: Overfeat



Network input:
3 x 221 x 221

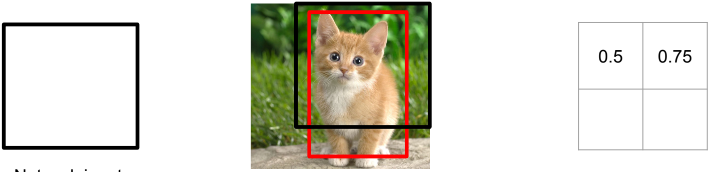
Larger image:
3 x 257 x 257

Classification scores:
P(cat)

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 25 1 Feb 2016

Sliding windows

Sliding Window: Overfeat



Network input:
3 x 221 x 221

Larger image:
3 x 257 x 257

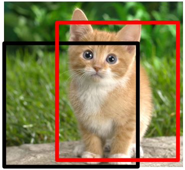
Classification scores:
P(cat)

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 26 1 Feb 2016

Sliding windows

Sliding Window: Overfeat

Network input:
3 x 221 x 221



Larger image:
3 x 257 x 257

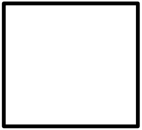
0.5	0.75
0.6	

Classification scores:
P(cat)

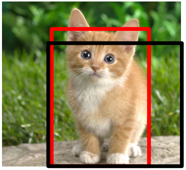
Fei-Fei Li & Andrej Karpathy & Justin Johnson | Lecture 8 - 27 | 1 Feb 2016

Sliding windows

Sliding Window: Overfeat



Network input:
3 x 221 x 221



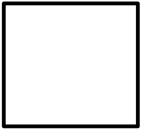
Larger image:
3 x 257 x 257

0.5	0.75
0.6	0.8

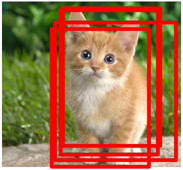
Classification scores:
P(cat)

Sliding windows

Sliding Window: Overfeat



Network input:
3 x 221 x 221



Larger image:
3 x 257 x 257

0.5	0.75
0.6	0.8

Classification scores:
P(cat)

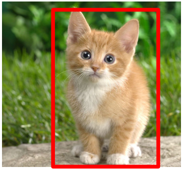
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 29 1 Feb 2016

Sliding windows

Sliding Window: Overfeat

Greedily merge boxes and scores (details in paper)

Network input:
3 x 221 x 221



Larger image:
3 x 257 x 257

0.8

Classification score: P (cat)

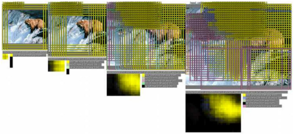
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 30 1 Feb 2016

Sliding windows

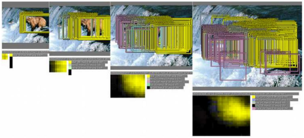
Sliding Window: Overfeat

In practice use many sliding window locations and multiple scales

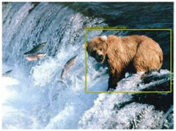
Window positions + score maps



Box regression outputs



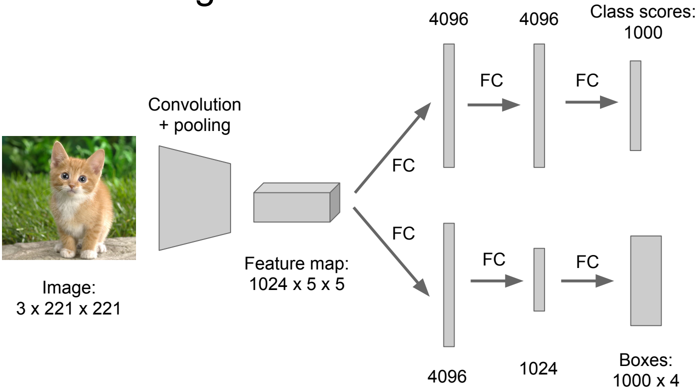
Final Predictions



Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Sliding windows

Efficient Sliding Window: Overfeat

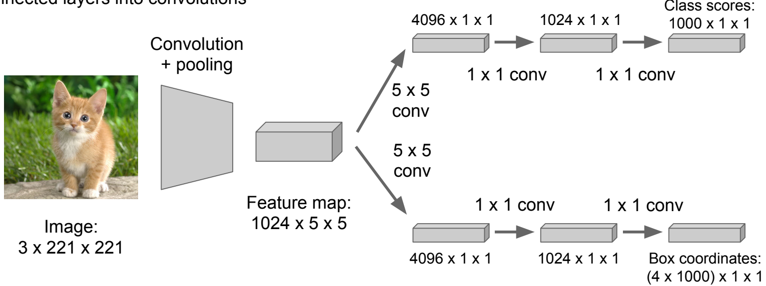


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 32 1 Feb 2016

Sliding windows

Efficient Sliding Window: Overfeat

Efficient sliding window by converting fully-connected layers into convolutions



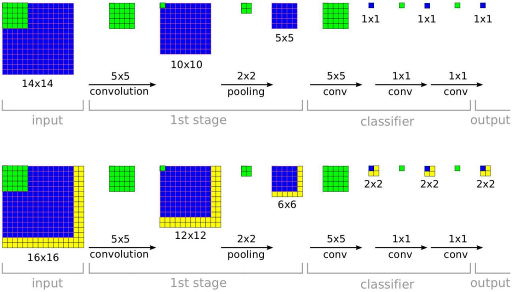
Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 8 - 33   1 Feb 2016

Sliding windows

Efficient Sliding Window: Overfeat

Training time: Small image, 1 x 1 classifier output

Test time: Larger image, 2 x 2 classifier output, only extra compute at yellow regions

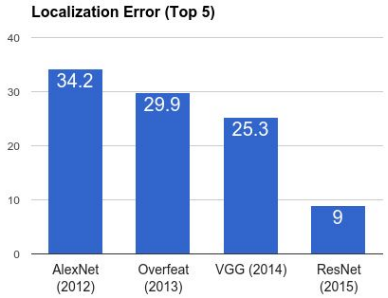


Sermanet et al, "Integrated Recognition, Localization and Detection using Convolutional Networks", ICLR 2014

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 34 1 Feb 2016

Sliding windows

ImageNet Classification + Localization



AlexNet: Localization method not published

Overfeat: Multiscale convolutional regression with box merging

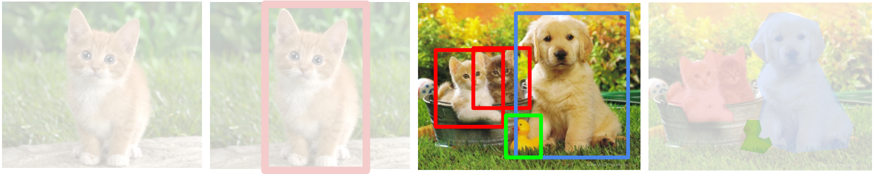
VGG: Same as Overfeat, but fewer scales and locations; simpler method, gains all due to deeper features

ResNet: Different localization method (RPN) and much deeper features

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 35 1 Feb 2016

Computer Vision Tasks

Classification | Classification + Localization | **Object Detection** | Instance Segmentation



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 37 1 Feb 2016

Detection as regression?

Detection as Regression?

DOG, (x, y, w, h)
CAT, (x, y, w, h)
CAT, (x, y, w, h)
DUCK (x, y, w, h)

= 16 numbers

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 38 1 Feb 2016

Detection as regression?

Detection as Regression?

DOG, (x, y, w, h)
CAT, (x, y, w, h)

= 8 numbers

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 39 1 Feb 2016

Detection as regression?

Detection as Regression?

CAT, (x, y, w, h)
CAT, (x, y, w, h)
....
CAT (x, y, w, h)

= many numbers

Need variable sized outputs

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 40 1 Feb 2016

Detection as classification

Detection as Classification

CAT? NO

DOG? NO

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 41 1 Feb 2016

Detection as classification

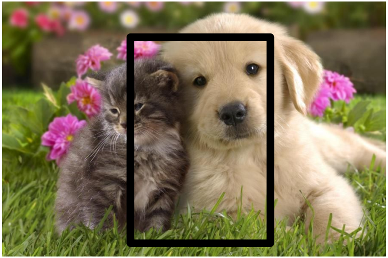
Detection as Classification

CAT? YES!

DOG? NO

Detection as classification

Detection as Classification



CAT? NO

DOG? NO

Detection as classification

Detection as Classification

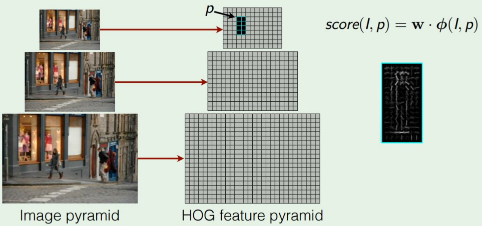
Problem: Need to test many positions and scales

Solution: If your classifier is fast enough, just do it

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 44 1 Feb 2016

Detection as classification

Histogram of Oriented Gradients

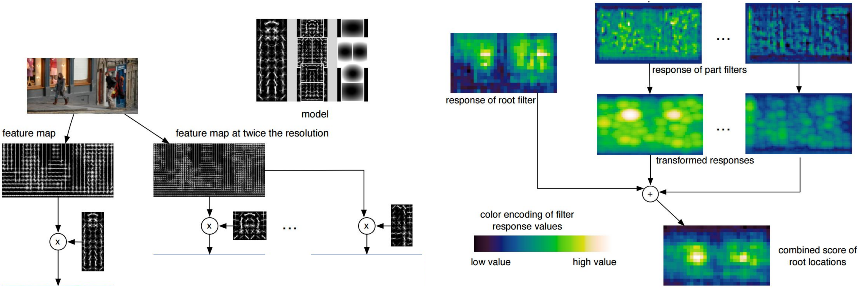


- Compute HOG of the whole image at multiple resolutions
- Score every subwindow of the feature pyramid
- Apply non-maxima suppression

Dalal and Triggs, “Histograms of Oriented Gradients for Human Detection”, CVPR 2005
Slide credit: Ross Girshick

Detection as classification

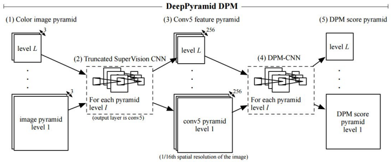
Deformable Parts Model (DPM)

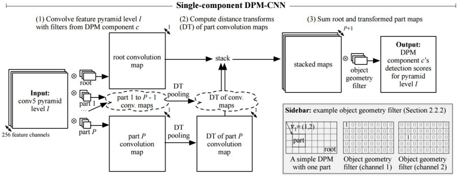


Felzenszwalb et al, “Object Detection with Discriminatively Trained Part Based Models”, PAMI 2010

Detection as classification

Aside: Deformable Parts Models are CNNs?





Girschick et al, "Deformable Part Models are Convolutional Neural Networks", CVPR 2015

Detection as classification

Detection as Classification

Problem: Need to test many positions and scales, and use a computationally demanding classifier (CNN)

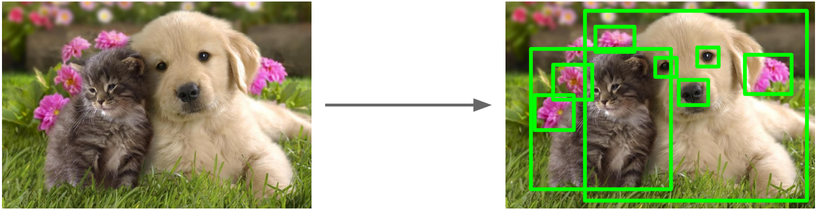
Solution: Only look at a tiny subset of possible positions

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 48 1 Feb 2016

Region proposal

Region Proposals

- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector
- Look for “blob-like” regions



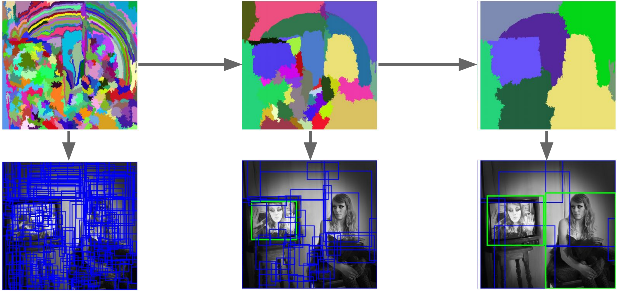
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 49 1 Feb 2016

Region proposal

Region Proposals: Selective Search

Bottom-up segmentation, merging regions at multiple scales

Convert regions to boxes



Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Region proposal

Region Proposals: Many other choices

Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea-tability	Recall Results	Detection Results
Bing [18]	Window scoring		✓	✓	0.2	⋆⋆⋆	⋆	·
CPMC [19]	Grouping	✓	✓	✓	250	-	⋆⋆	⋆
EdgeBoxes [20]	Window scoring		✓	✓	0.3	⋆⋆	⋆⋆⋆	⋆⋆⋆
Endres [21]	Grouping	✓	✓	✓	100	-	⋆⋆⋆	⋆⋆
Geodesic [22]	Grouping	✓		✓	1	⋆	⋆⋆⋆	⋆⋆
MCG [23]	Grouping	✓	✓	✓	30	⋆	⋆⋆⋆	⋆⋆⋆
Objectness [24]	Window scoring		✓	✓	3	·	⋆	·
Rahtu [25]	Window scoring		✓	✓	3	·	·	⋆
RandomizedPrim's [26]	Grouping	✓		✓	1	⋆	⋆	⋆⋆
Rantalankila [27]	Grouping	✓		✓	10	⋆⋆	·	⋆⋆
Rigor [28]	Grouping	✓		✓	10	⋆	⋆⋆	⋆⋆
SelectiveSearch [29]	Grouping	✓	✓	✓	10	⋆⋆	⋆⋆⋆	⋆⋆⋆
Gaussian				✓	0	·	·	⋆
SlidingWindow				✓	0	⋆⋆⋆	·	·
Superpixels		✓			1	⋆	·	·
Uniform				✓	0	·	·	·

Hosang et al, "What makes for effective detection proposals?", PAMI 2015

Region proposal

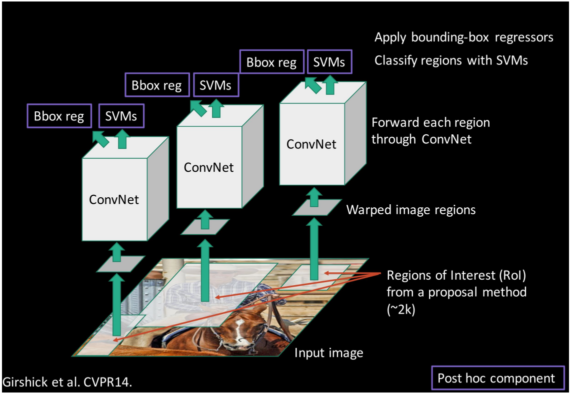
Region Proposals: Many other choices

Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea-tability	Recall Results	Detection Results
Bing [18]	Window scoring		✓	✓	0.2	⋆⋆⋆	⋆	·
CPMC [19]	Grouping	✓	✓	✓	250	-	⋆⋆	⋆
EdgeBoxes [20]	Window scoring		✓	✓	0.3	⋆⋆	⋆⋆⋆	⋆⋆⋆
Endres [21]	Grouping	✓	✓	✓	100	-	⋆⋆⋆	⋆⋆
Geodesic [22]	Grouping	✓		✓	1	⋆	⋆⋆⋆	⋆⋆
MCG [23]	Grouping	✓	✓	✓	30	⋆	⋆⋆⋆	⋆⋆⋆
Objectness [24]	Window scoring		✓	✓	3	·	⋆	·
Rahtu [25]	Window scoring		✓	✓	3	·	·	⋆
RandomizedPrim's [26]	Grouping	✓		✓	1	⋆	⋆	⋆⋆
Rantalankila [27]	Grouping	✓		✓	10	⋆⋆	·	⋆⋆
Rigor [28]	Grouping	✓		✓	10	⋆	⋆⋆	⋆⋆
SelectiveSearch [29]	Grouping	✓	✓	✓	10	⋆⋆	⋆⋆⋆	⋆⋆⋆
Gaussian				✓	0	·	·	⋆
SlidingWindow				✓	0	⋆⋆⋆	·	·
Superpixels		✓			1	⋆	·	·
Uniform				✓	0	·	·	·

Hosang et al, "What makes for effective detection proposals?", PAMI 2015

R-CNN

Putting it together: R-CNN



Girschick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014

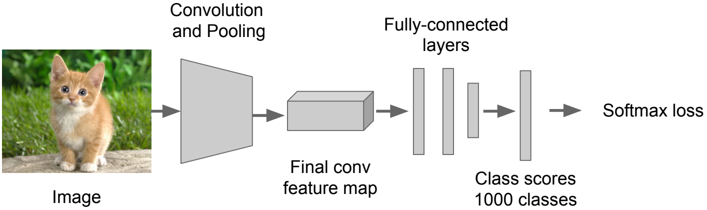
Slide credit: Ross Girschick

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 53 1 Feb 2016

R-CNN

R-CNN Training

Step 1: Train (or download) a classification model for ImageNet (AlexNet)



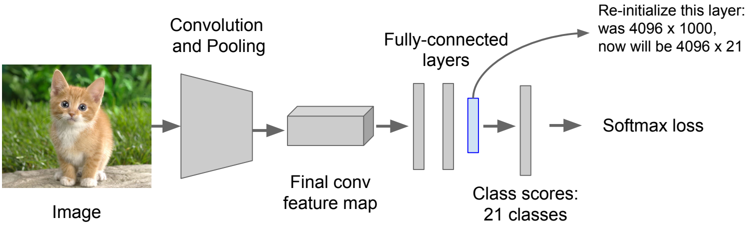
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 54 1 Feb 2016

R-CNN

R-CNN Training

Step 2: Fine-tune model for detection

- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize from scratch
- Keep training model using positive / negative regions from detection images



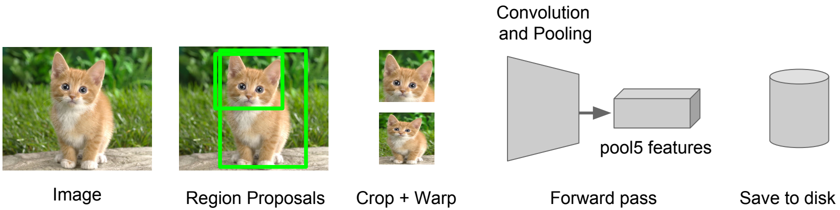
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 55 1 Feb 2016

R-CNN

R-CNN Training

Step 3: Extract features

- Extract region proposals for all images
- For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk
- Have a big hard drive: features are ~200GB for PASCAL dataset!

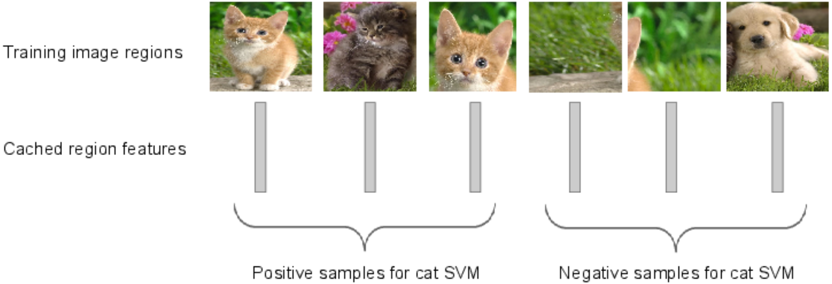


Fei-Fei Li & Andrej Karpathy & Justin Johnson | Lecture 8 - 56 | 1 Feb 2016

R-CNN

R-CNN Training

Step 4: Train one binary SVM per class to classify region features

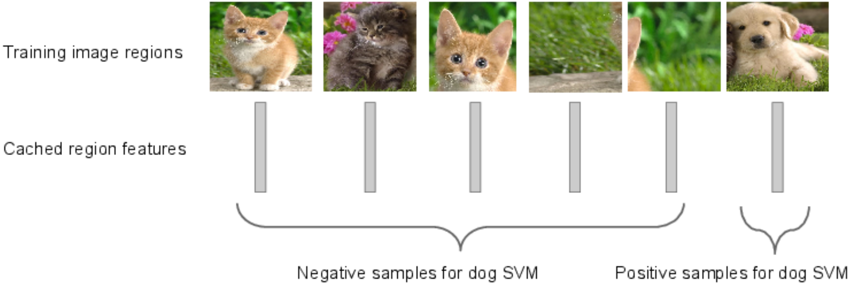


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 57 1 Feb 2016

R-CNN

R-CNN Training

Step 4: Train one binary SVM per class to classify region features



Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 8 - 58   1 Feb 2016

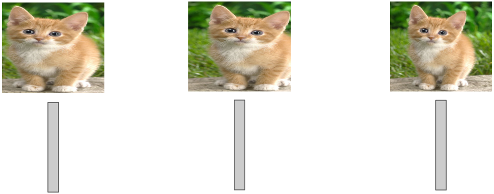
R-CNN

R-CNN Training

Step 5 (bbox regression): For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for “slightly wrong” proposals

Training image regions

Cached region features



Regression targets
(dx, dy, dw, dh)
Normalized coordinates

(0, 0, 0, 0)
Proposal is good

(.25, 0, 0, 0)
Proposal too far to left

(0, 0, -0.125, 0)
Proposal too wide

R-CNN

Object Detection: Datasets

	PASCAL VOC (2010)	ImageNet Detection (ILSVRC 2014)	MS-COCO (2014)
Number of classes	20	**200**	80
Number of images (train + val)	~20k	**~470k**	~120k
Mean objects per image	2.4	1.1	**7.2**

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 60 1 Feb 2016

R-CNN

Object Detection: Evaluation

We use a metric called “mean average precision” (mAP)

Compute average precision (AP) separately for each class, then average over classes

A detection is a true positive if it has IoU with a ground-truth box greater than some threshold (usually 0.5) (mAP@0.5)

Combine all detections from all test images to draw a precision / recall curve for each class; AP is area under the curve

TL;DR mAP is a number from 0 to 100; high is good

More on AP

- *AP* is computed as the *average precision* of the precision-recall curve $p(r)$. That is
$$AP = \int_{r=0}^{1} p(r)dr,$$
which essentially is also the area under $p(r)$
- Assume that there is n matches and $P(k)$ is the precision of the first k matches, we can write *AP* as
$$AP = \sum_{k=1}^{n} P(k)\Delta r(k) = \frac{\sum_{k=1}^{n} P(k) rel(k)}{\#relevant\ matches},$$
where $\Delta r(k)$ is the change of recall after considering the k-th match, and $rel(k)$ is 1 if k-th match is relevant or 0 otherwise

More on AP

- It is common to reduce the ”wiggles” of the precision-recall curve by using interpolation and approximate AP as below instead

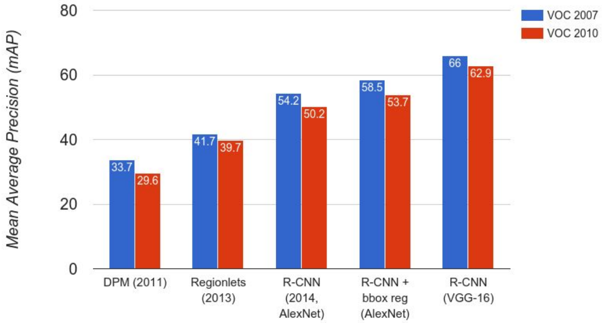
$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, 0.2, \cdots, 1\}} p_{interp}(r),$$

where $p_{interp}(r) = max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r})$

See https://en.wikipedia.org/wiki/Information_retrieval#Average_precision and http://homepages.inf.ed.ac.uk/ckiw/postscript/ijcv_voc09.pdf

R-CNN

R-CNN Results



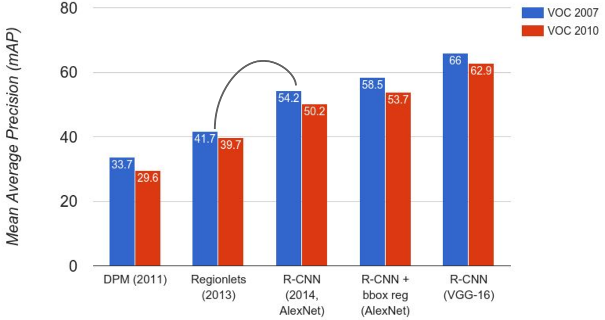
Wang et al, "Regionlets for Generic Object Detection", ICCV 2013

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 62 1 Feb 2016

R-CNN

R-CNN Results

Big improvement compared to pre-CNN methods

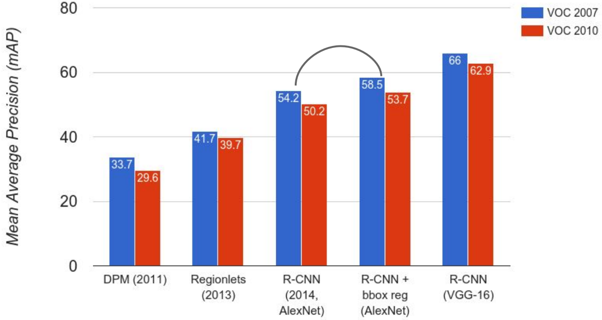


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 63 1 Feb 2016

R-CNN

R-CNN Results

Bounding box regression helps a bit

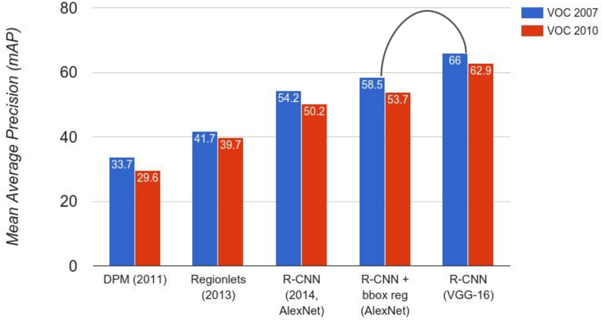


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 64 1 Feb 2016

R-CNN

R-CNN Results

Features from a deeper network help a lot



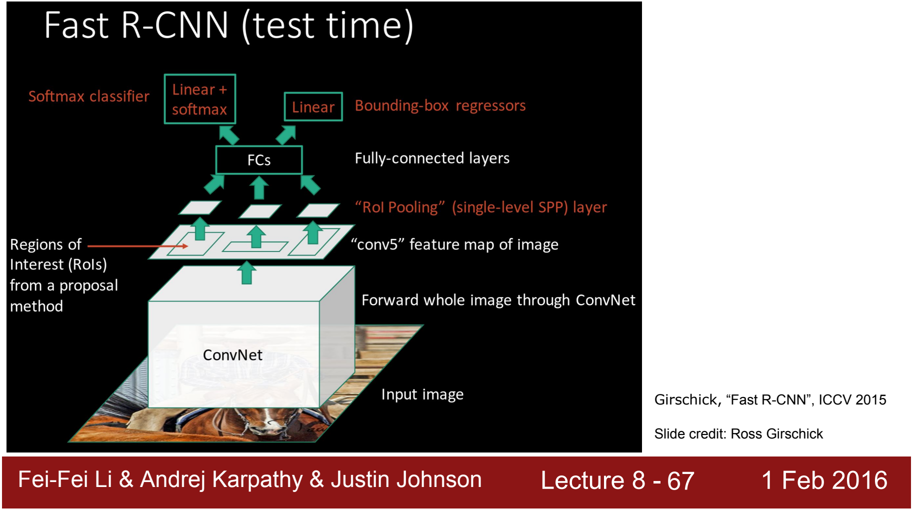
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 65 1 Feb 2016

R-CNN

R-CNN Problems

1. Slow at test-time: need to run full forward pass of CNN for each region proposal
2. SVMs and regressors are post-hoc: CNN features not updated in response to SVMs and regressors
3. Complex multistage training pipeline

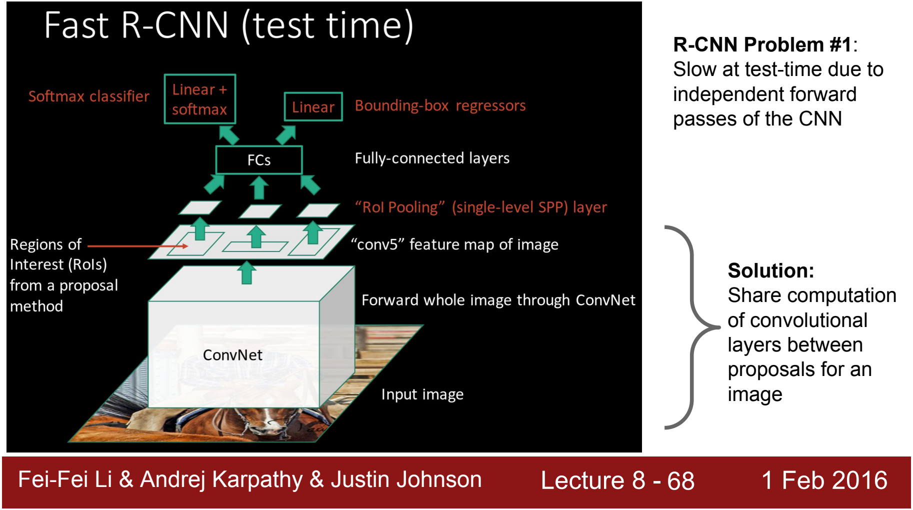
Fast R-CNN



Girschick, "Fast R-CNN", ICCV 2015

Slide credit: Ross Girschick

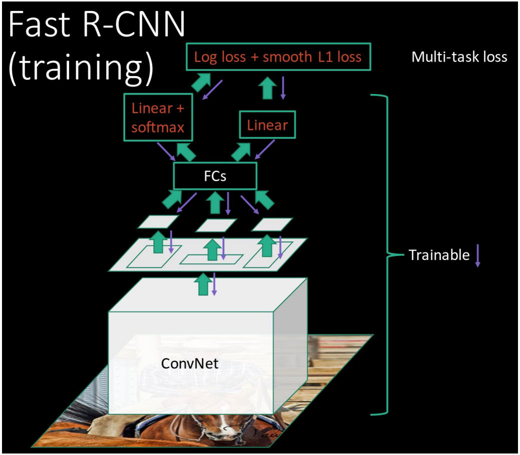
Fast R-CNN



R-CNN Problem #1: Slow at test-time due to independent forward passes of the CNN

Solution: Share computation of convolutional layers between proposals for an image

Fast R-CNN



R-CNN Problem #2:
Post-hoc training: CNN not updated in response to final classifiers and regressors

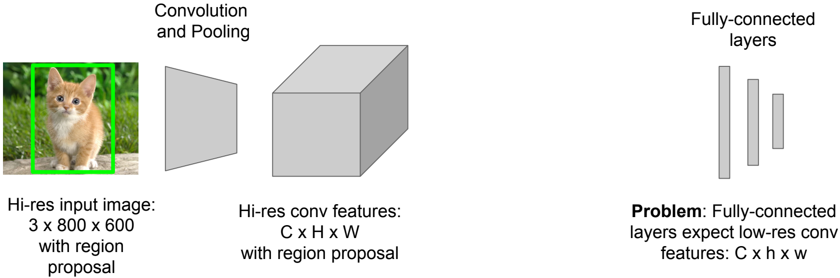
R-CNN Problem #3:
Complex training pipeline

Solution:
Just train the whole system end-to-end all at once!

Slide credit: Ross Girschick

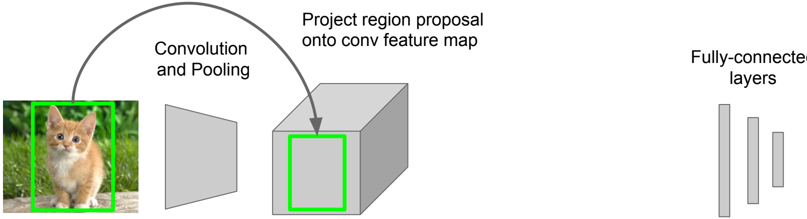
Fast R-CNN

Fast R-CNN: Region of Interest Pooling



Fast R-CNN

Fast R-CNN: Region of Interest Pooling



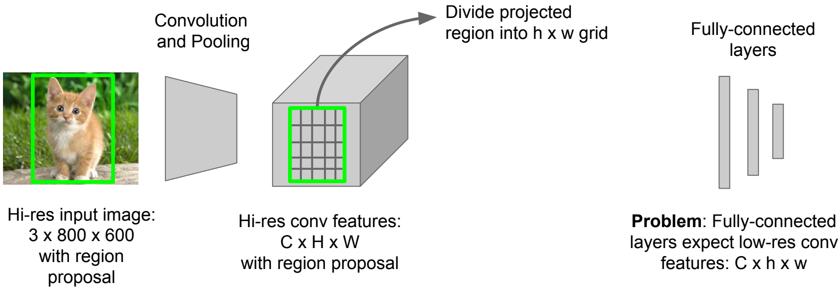
Hi-res input image:
3 x 800 x 600
with region
proposal

Hi-res conv features:
C x H x W
with region proposal

Problem: Fully-connected layers expect low-res conv features: C x h x w

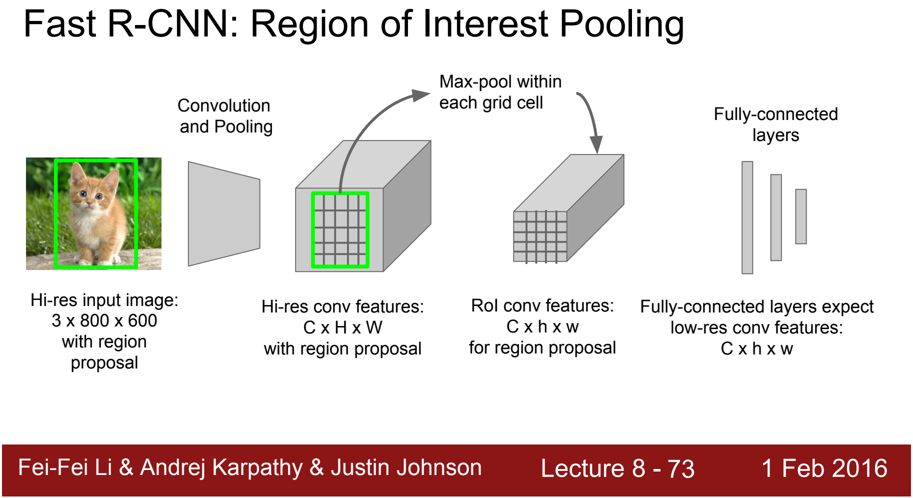
Fast R-CNN

Fast R-CNN: Region of Interest Pooling



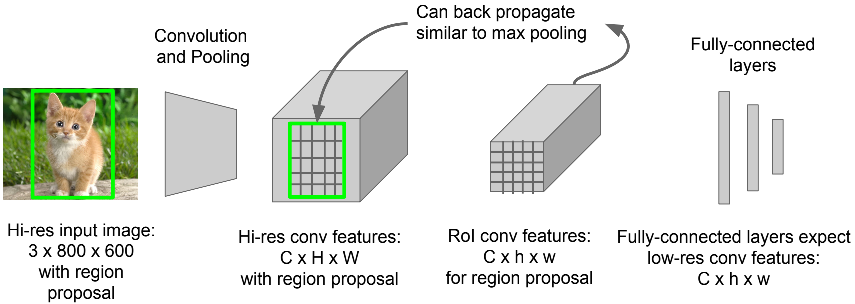
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 72 1 Feb 2016

Fast R-CNN


Fast R-CNN: Region of Interest Pooling
Convolution and Pooling
Max-pool within each grid cell
Fully-connected layers
Hi-res input image: 3 x 800 x 600 with region proposal
Hi-res conv features: C x H x W with region proposal
RoI conv features: C x h x w for region proposal
Fully-connected layers expect low-res conv features: C x h x w
Fei-Fei Li & Andrej Karpathy & Justin Johnson
Lecture 8 - 73
1 Feb 2016

Fast R-CNN

Fast R-CNN: Region of Interest Pooling



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 74 1 Feb 2016

Fast R-CNN

Fast R-CNN Results

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**

Using VGG-16 CNN on Pascal VOC 2007 dataset

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 75 1 Feb 2016

Fast R-CNN

Fast R-CNN Results

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**
FASTER!	Test time per image	47 seconds	**0.32 seconds**
	(Speedup)	1x	**146x**

Using VGG-16 CNN on Pascal VOC 2007 dataset

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 76 1 Feb 2016

Fast R-CNN

Fast R-CNN Results

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**
FASTER!	Test time per image	47 seconds	**0.32 seconds**
	(Speedup)	1x	**146x**
Better!	mAP (VOC 2007)	66.0	**66.9**

Using VGG-16 CNN on Pascal VOC 2007 dataset

Fast R-CNN

Fast R-CNN Problem:

Test-time speeds don't include region proposals

	R-CNN	**Fast R-CNN**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 78 1 Feb 2016

Fast R-CNN

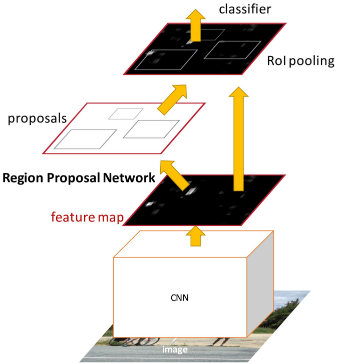
Fast R-CNN ~~Problem~~ Solution:

Test-time speeds don't include region proposals
Just make the CNN do region proposals too!

	R-CNN	**Fast R-CNN**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

Faster R-CNN

Faster R-CNN:



Insert a **Region Proposal Network (RPN)** after the last convolutional layer

RPN trained to produce region proposals directly; no need for external region proposals!

After RPN, use RoI Pooling and an upstream classifier and bbox regressor just like Fast R-CNN

Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Slide credit: Ross Girschick

Faster R-CNN

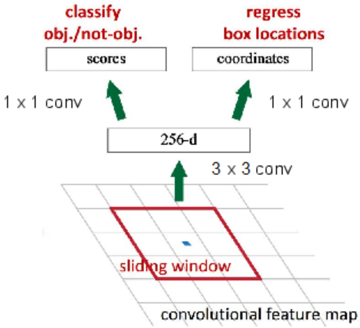
Faster R-CNN: Region Proposal Network

Slide a small window on the feature map

Build a small network for:
- classifying object or not-object, and
- regressing bbox locations

Position of the sliding window provides localization information with reference to the image

Box regression provides finer localization information with reference to this sliding window



Slide credit: Kaiming He

Faster R-CNN

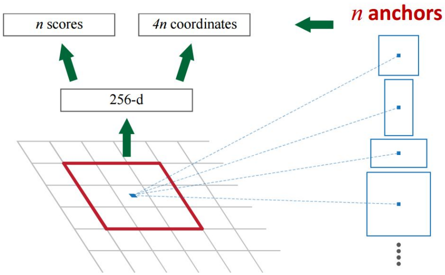
Faster R-CNN: Region Proposal Network

Use **N anchor boxes** at each location

Anchors are **translation invariant**: use the same ones at every location

Regression gives offsets from anchor boxes

Classification gives the probability that each (regressed) anchor shows an object



Faster R-CNN

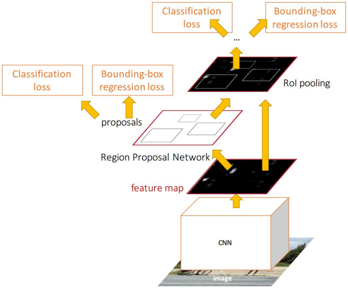
Faster R-CNN: Training

In the paper: Ugly pipeline

- Use alternating optimization to train RPN, then Fast R-CNN with RPN proposals, etc.
- More complex than it has to be

Since publication: Joint training!
One network, four losses

- RPN classification (anchor good / bad)
- RPN regression (anchor -> proposal)
- Fast R-CNN classification (over classes)
- Fast R-CNN regression (proposal -> box)



Slide credit: Ross Girschick

Faster R-CNN

Faster R-CNN: Results

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 84 1 Feb 2016

Faster R-CNN

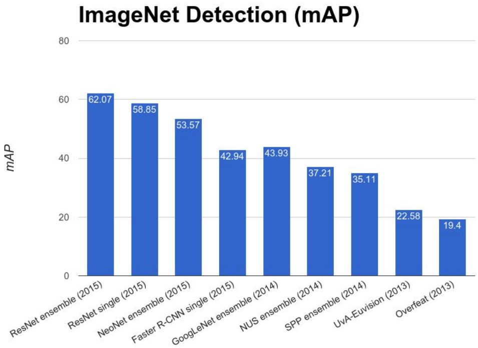
Object Detection State-of-the-art: ResNet 101 + Faster R-CNN + some extras

training data	COCO train		COCO trainval	
test data	COCO val		COCO test-dev	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16)	41.5	21.2		
baseline Faster R-CNN (ResNet-101)	48.4	27.2		
+box refinement	49.9	29.9		
+context	51.1	30.0	53.3	32.2
+multi-scale testing	53.8	32.5	**55.7**	**34.9**
ensemble			**59.0**	**37.4**

He et. al, "Deep Residual Learning for Image Recognition", arXiv 2015

Faster R-CNN

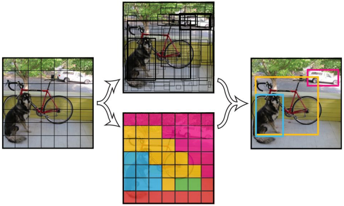
ImageNet Detection 2013 - 2015



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 86 1 Feb 2016

YOLO

YOLO: You Only Look Once Detection as Regression

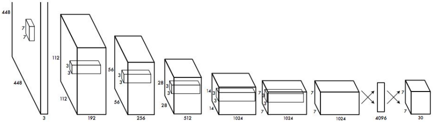


Divide image into S x S grid

Within each grid cell predict:
- B Boxes: 4 coordinates + confidence
- Class scores: C numbers

Regression from image to
7 x 7 x (5 * B + C) tensor

Direct prediction using a CNN



Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", arXiv 2015

YOLO

YOLO: You Only Look Once Detection as Regression

Faster than Faster R-CNN, but not as good

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18

Redmon et al, “You Only Look Once: Unified, Real-Time Object Detection”, arXiv 2015

Summary

Object Detection code links:

R-CNN
(Cafffe + MATLAB): https://github.com/rbgirshick/rcnn
Probably don't use this; too slow

Fast R-CNN
(Caffe + MATLAB): https://github.com/rbgirshick/fast-rcnn

Faster R-CNN
(Caffe + MATLAB): https://github.com/ShaoqingRen/faster_rcnn
(Caffe + Python): https://github.com/rbgirshick/py-faster-rcnn

YOLO
http://pjreddie.com/darknet/yolo/
Maybe try this for projects?

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 89 1 Feb 2016

Recap

Localization:
- Find a fixed number of objects (one or many)
- L2 regression from CNN features to box coordinates
- Much simpler than detection; consider it for your projects!
- Overfeat: Regression + efficient sliding window with FC -> conv conversion
- Deeper networks do better

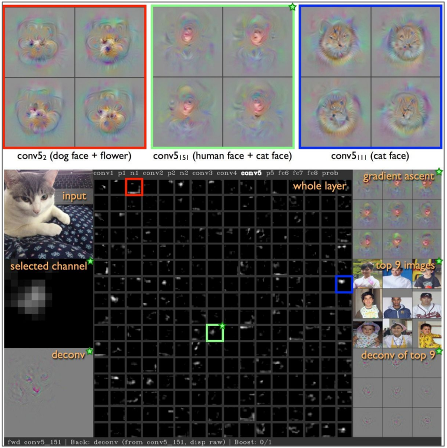
Object Detection:
- Find a variable number of objects by classifying image regions
- Before CNNs: dense multiscale sliding window (HoG, DPM)
- Avoid dense sliding window with region proposals
- R-CNN: Selective Search + CNN classification / regression
- Fast R-CNN: Swap order of convolutions and region extraction
- Faster R-CNN: Compute region proposals within the network
- Deeper networks do better

Visualizing and understanding conv-nets

- Study weights directly
- Occlusion experiment
- Visualizing representation
  - t-SNE
  - through deconvolution
  - through optimization

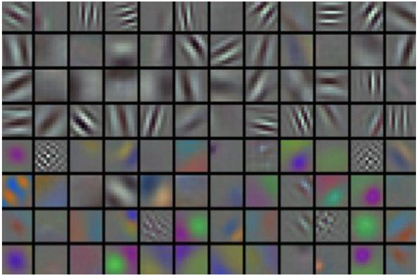
Visualizing Activations

http://yosinski.com/deepvis

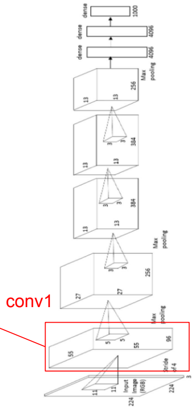


YouTube video
https://www.youtube.com/watch?v=AgkflQ4IGaM
(4min)

Visualize the filters/kernels (raw weights)



one-stream AlexNet



only interpretable on the first layer :(

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 8 3 Feb 2016

Visualize the filters/kernels (raw weights)

you can still do it for higher layers, it's just not that interesting

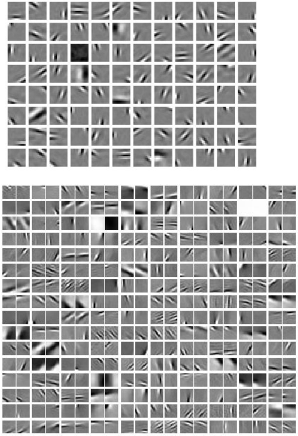
(these are taken from ConvNetJS CIFAR-10 demo)

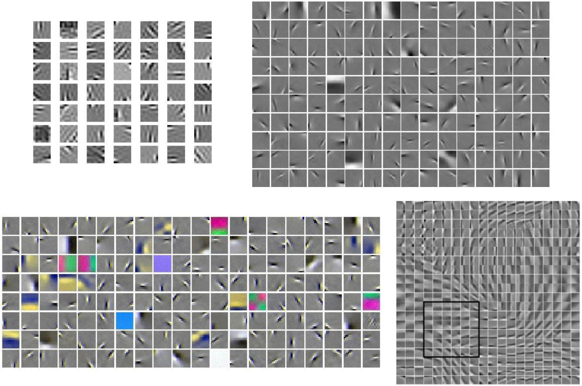
layer 1 weights

layer 2 weights

layer 3 weights

The gabor-like filters fatigue

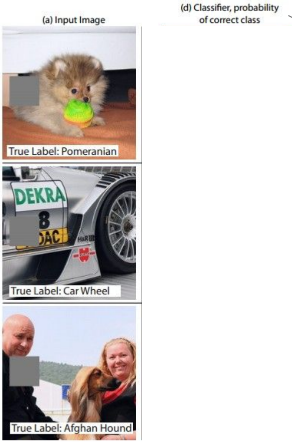




Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 10   3 Feb 2016

Occlusion experiments

[Zeiler & Fergus 2013]

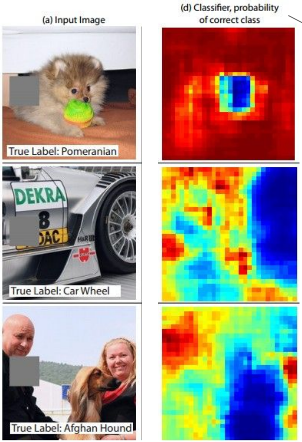


(as a function of the position of the square of zeros in the original image)

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 14 3 Feb 2016

Occlusion experiments

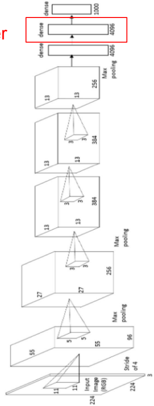
[Zeiler & Fergus 2013]



(as a function of the position of the square of zeros in the original image)

Visualizing the representation

fc7 layer



4096-dimensional "code" for an image
(layer immediately before the classifier)

can collect the code for many images

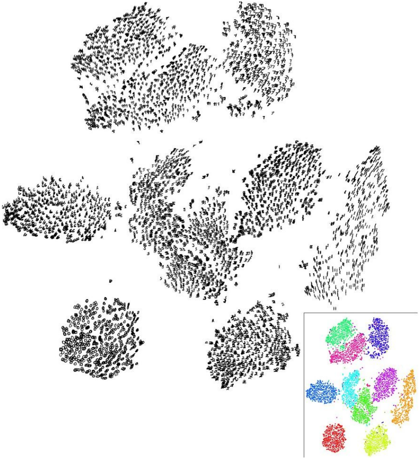
Visualizing the representation

t-SNE visualization
[van der Maaten & Hinton]

Embed high-dimensional points so that locally, pairwise distances are conserved

i.e. similar things end up in similar places. dissimilar things end up wherever

Right: Example embedding of MNIST digits (0-9) in 2D

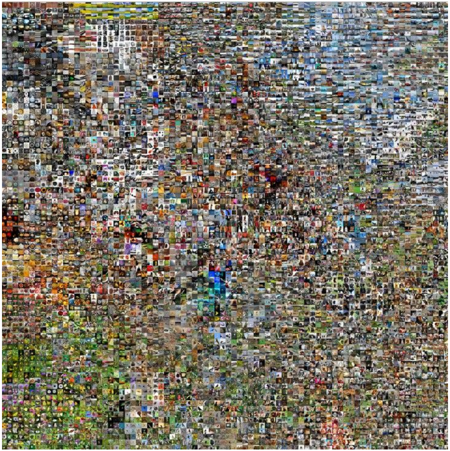


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 12 3 Feb 2016

t-SNE visualization:

two images are placed nearby if their CNN codes are close. See more:

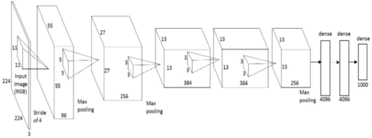
http://cs.stanford.edu/people/karpathy/cnnembed/



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 13 3 Feb 2016

Deconv approaches

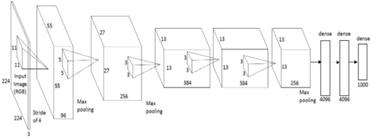
1. Feed image into net

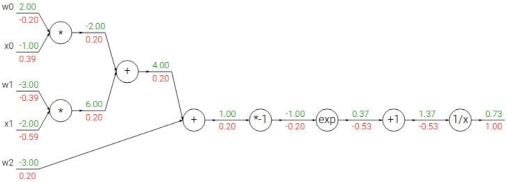


Q: how can we compute the gradient of any arbitrary neuron in the network w.r.t. the image?

Deconv approaches

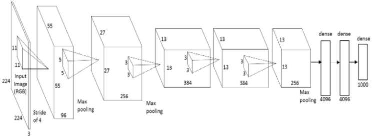
1. Feed image into net



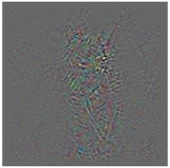


Deconv approaches

1. Feed image into net



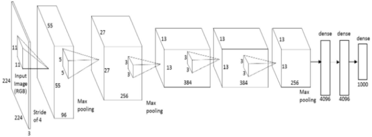
2. Pick a layer, set the gradient there to be all zero except for one 1 for some neuron of interest
3. Backprop to image:



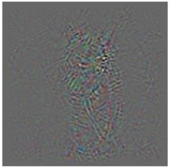
Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 19   3 Feb 2016

Deconv approaches

1. Feed image into net



2. Pick a layer, set the gradient there to be all zero except for one 1 for some neuron of interest
3. Backprop to image:



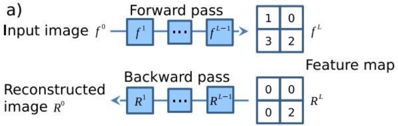
“Guided backpropagation:” instead

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 20   3 Feb 2016

Guided backprop

Deconv approaches

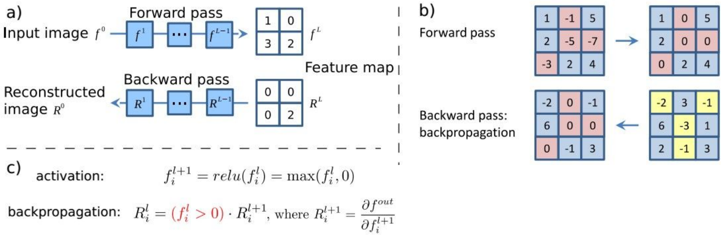
[Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]
[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, Simonyan et al., 2014]
[Striving for Simplicity: The all convolutional net, Springenberg, Dosovitskiy, et al., 2015]



Guided backprop

Deconv approaches

[Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]
[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, Simonyan et al., 2014]
[Striving for Simplicity: The all convolutional net, Springenberg, Dosovitskiy, et al., 2015]

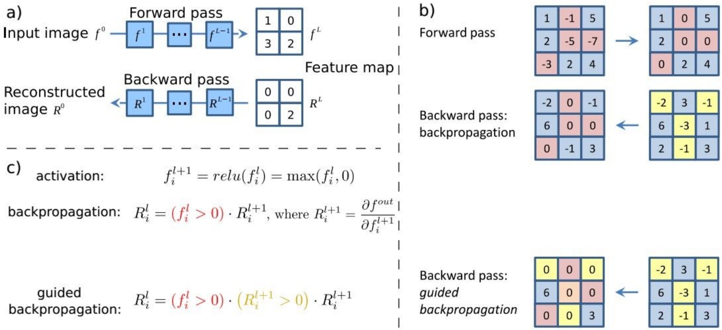


Backward pass for a ReLU (will be changed in Guided Backprop)

Guided backprop

Deconv approaches

[Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]
[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, Simonyan et al., 2014]
[Striving for Simplicity: The all convolutional net, Springenberg, Dosovitskiy, et al., 2015]



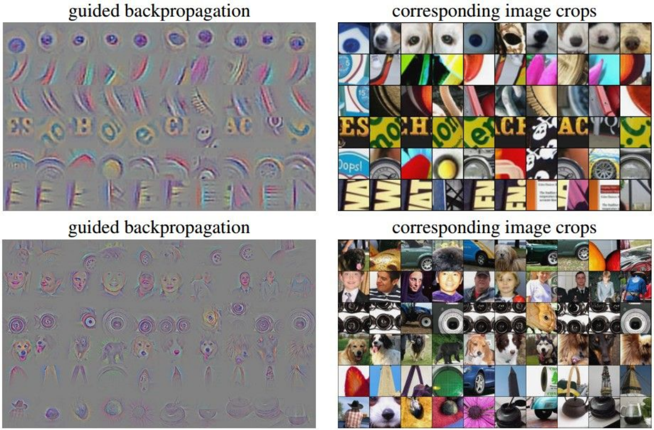
Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 23   3 Feb 2016

Guided backprop

Visualization of patterns learned by the layer **conv6** (top) and layer **conv9** (bottom) of the network trained on ImageNet.

Each row corresponds to one filter.

The visualization using "guided backpropagation" is based on the top 10 image patches activating this filter taken from the ImageNet dataset.

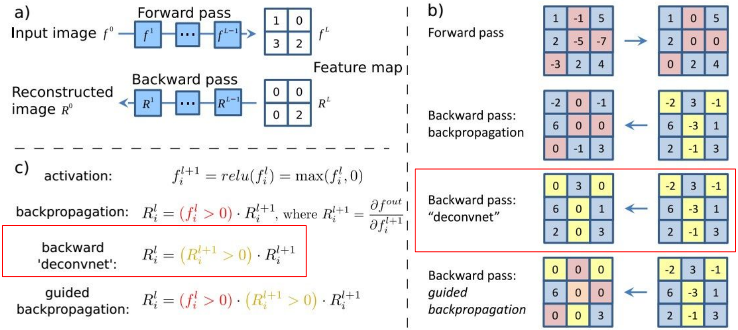


[Striving for Simplicity: The all convolutional net, Springenberg, Dosovitskiy, et al., 2015]

Backward deconvolution

Deconv approaches

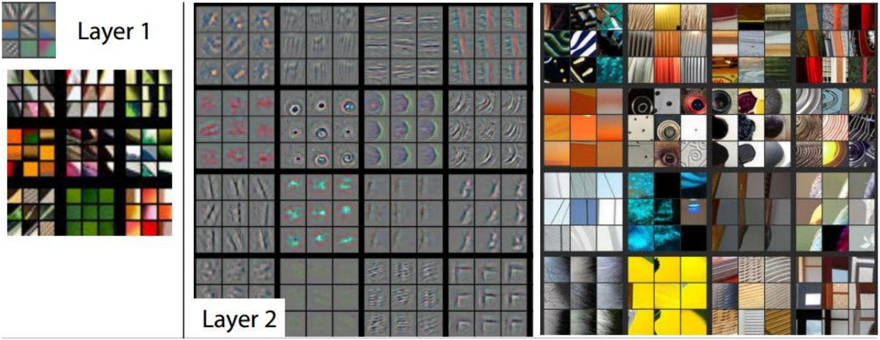
[Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]
[Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, Simonyan et al., 2014]
[Striving for Simplicity: The all convolutional net, Springenberg, Dosovitskiy, et al., 2015]



bit weird

Visualizing and Understanding Convolutional Networks
Zeiler & Fergus, 2013

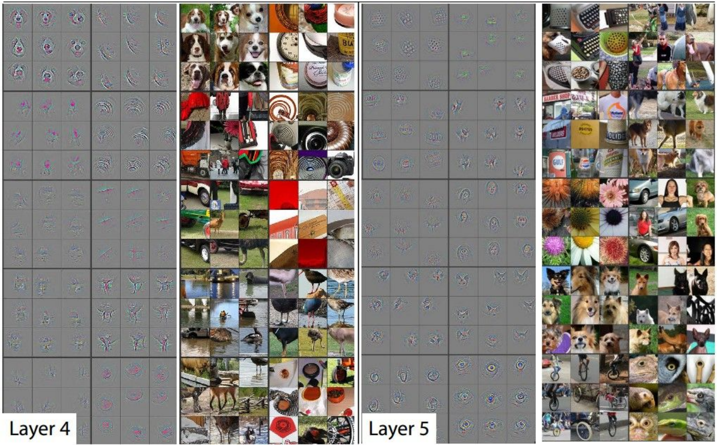
Visualizing arbitrary neurons along the way to the top...



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 26 3 Feb 2016

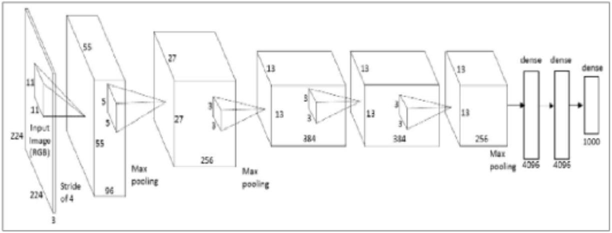
Visualizing arbitrary neurons along the way to the top...

Visualizing arbitrary neurons along the way to the top...



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 28 3 Feb 2016

Finding salient map of an object



Repeat:

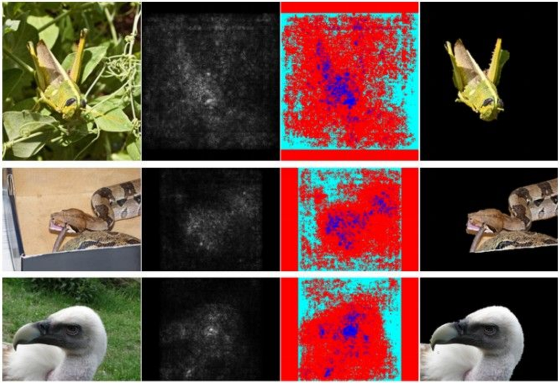
1. Forward an image
2. Set activations in layer of interest to all zero, except for a 1.0 for a neuron of interest
3. Backprop to image
4. Do an "image update"

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 38 3 Feb 2016

Finding salient map of an object

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, 2014

- Use **grabcut** for segmentation



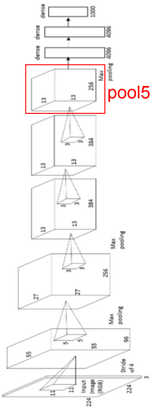
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 37 3 Feb 2016

Patches maximally activate a neuron

Visualize patches that maximally activate neurons

Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

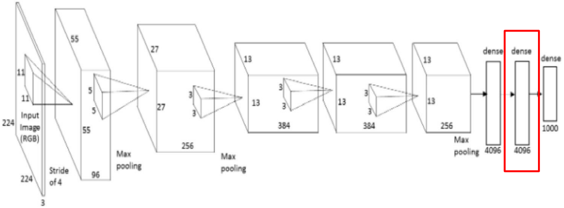
one-stream AlexNet



Rich feature hierarchies for accurate object detection and semantic segmentation [Girshick, Donahue, Darrell, Malik]

Recovering original image

Question: Given a CNN code, is it possible to reconstruct the original image?



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 44 3 Feb 2016

Recovering original image

Find an image such that:

- Its code is similar to a given code
- It "looks natural" (image prior regularization)

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^{H\times W\times C}}{\operatorname{argmin}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda\mathcal{R}(\mathbf{x})$$

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 45 3 Feb 2016

Recovering original image

Understanding Deep Image Representations by Inverting Them [Mahendran and Vedaldi, 2014]

original image

reconstructions from the 1000 log probabilities for ImageNet (ILSVRC) classes

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 46   3 Feb 2016

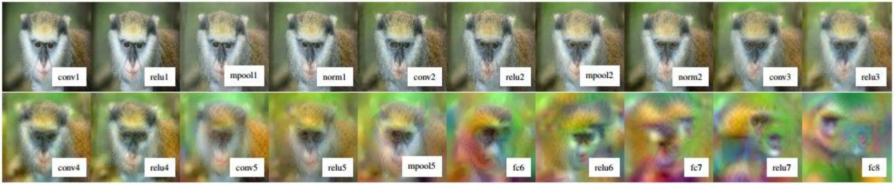
Recovering original image

Reconstructions from the representation after last last pooling layer (immediately before the first Fully Connected layer)

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 47 3 Feb 2016

Recovering original image

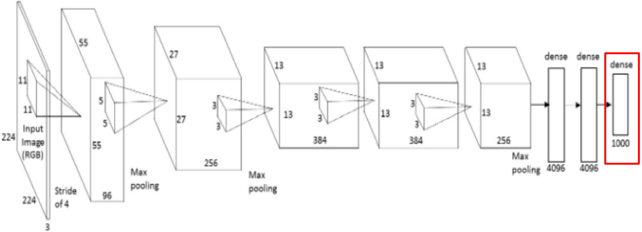
Reconstructions from intermediate layers



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 48 3 Feb 2016

Class model visualization

Optimization to Image



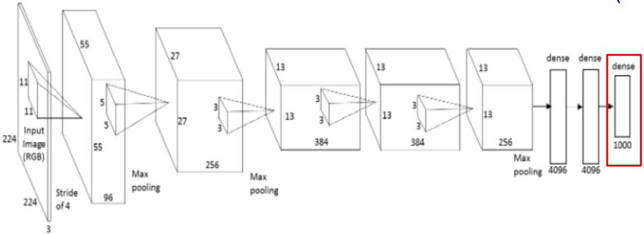
Q: can we find an image that maximizes some class score?

Class model visualization

Optimization to Image

$$\arg\max_{I} \boxed{S_c(I)} - \lambda \|I\|_2^2$$

score for class c (before Softmax)



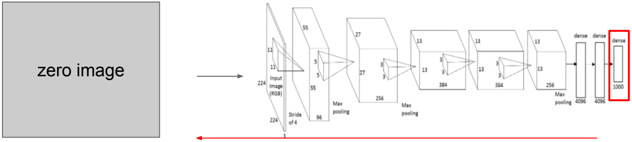
Q: can we find an image that maximizes some class score?

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 30   3 Feb 2016

Class model visualization

Optimization to Image

1. feed in zeros.

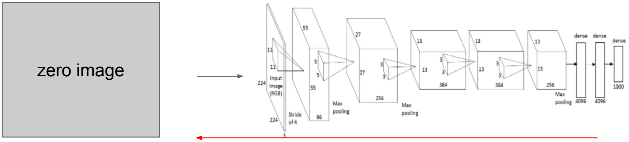


2. set the gradient of the scores vector to be [0,0,....1,....,0], then backprop to image

Class model visualization

Optimization to Image

1. feed in zeros.



2. set the gradient of the scores vector to be [0,0,....1,....,0], then backprop to image
3. do a small “image update”
4. forward the image through the network.
5. go back to 2.

$$\arg\max_{I} \boxed{S_c(I)} - \lambda \|I\|_2^2$$

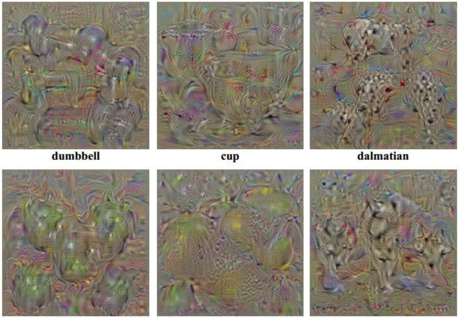
score for class c (before Softmax)

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 32   3 Feb 2016

Class model visualization

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, 2014

1. Find images that maximize some class score:



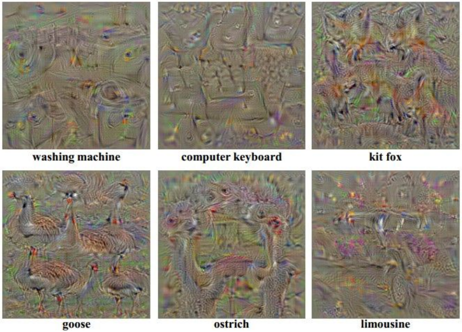
dumbbell cup dalmatian
bell pepper lemon husky

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 33 3 Feb 2016

Class model visualization

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, 2014

1. Find images that maximize some class score:



Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 34   3 Feb 2016

Class model visualization

[Understanding Neural Networks Through Deep Visualization, Yosinski et al. , 2015]

Proposed a different form of regularizing the image

$$\arg\max_I S_c(I) - \boxed{\lambda \|I\|_2^2}$$

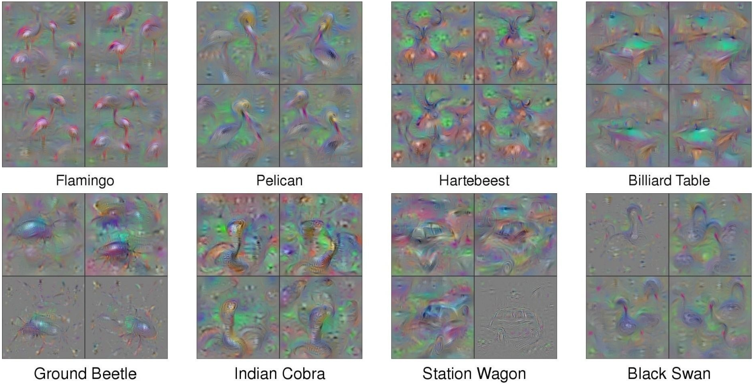
More explicit scheme:

Repeat:

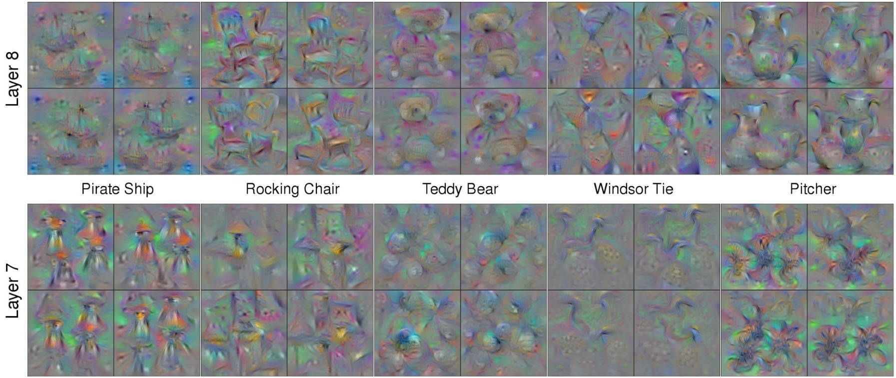
- Update the image **x** with gradient from some unit of interest
- Blur x a bit
- Take any pixel with small norm to zero (to encourage sparsity)

Class model visualization

[Understanding Neural Networks Through Deep Visualization, Yosinski et al. , 2015]
http://yosinski.com/deepvis

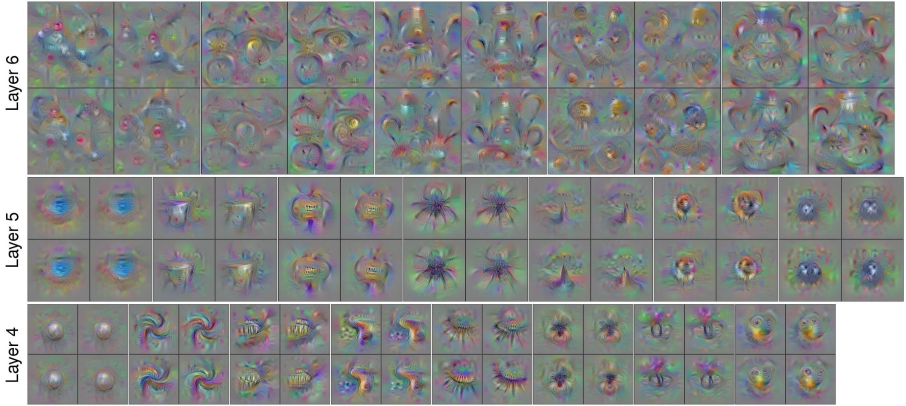


Class model visualization



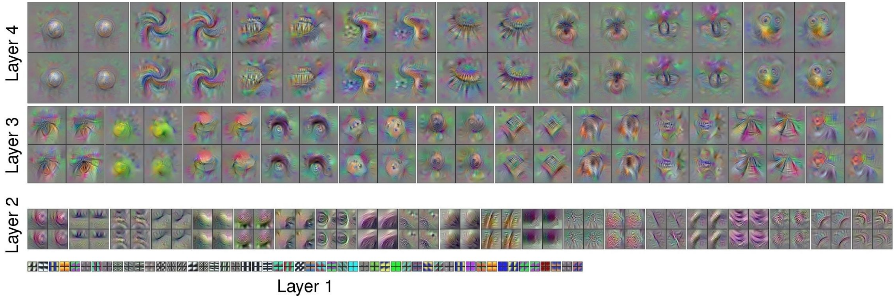
Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 41 3 Feb 2016

Class model visualization



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 42 3 Feb 2016

Class model visualization



Fei-Fei Li & Andrej Karpathy & Justin Johnson | Lecture 9 - 43 | 3 Feb 2016

DeepDream https://github.com/google/deepdream

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 50 3 Feb 2016

```python
def objective_L2(dst):
    dst.diff[:] = dst.data

def make_step(net, step_size=1.5, end='inception_4c/output',
              jitter=32, clip=True, objective=objective_L2):
    '''Basic gradient ascent step.'''

    src = net.blobs['data'] # input image is stored in Net's 'data' blob
    dst = net.blobs[end]

    ox, oy = np.random.randint(-jitter, jitter+1, 2)
    src.data[0] = np.roll(np.roll(src.data[0], ox, -1), oy, -2) # apply jitter shift

    net.forward(end=end)
    objective(dst)  # specify the optimization objective
    net.backward(start=end)
    g = src.diff[0]
    # apply normalized ascent step to the input image
    src.data[:] += step_size/np.abs(g).mean() * g

    src.data[0] = np.roll(np.roll(src.data[0], -ox, -1), -oy, -2) # unshift image

    if clip:
        bias = net.transformer.mean['data']
        src.data[:] = np.clip(src.data, -bias, 255-bias)
```

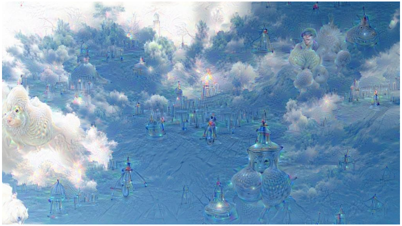
```python
def objective_L2(dst):
    dst.diff[:] = dst.data

def make_step(net, step_size=1.5, end='inception_4c/output',
              jitter=32, clip=True, objective=objective_L2):
    '''Basic gradient ascent step.'''

    src = net.blobs['data'] # input image is stored in Net's 'data' blob
    dst = net.blobs[end]

    ox, oy = np.random.randint(-jitter, jitter+1, 2)
    src.data[0] = np.roll(np.roll(src.data[0], ox, -1), oy, -2) # apply jitter shift

    net.forward(end=end)
    objective(dst)  # specify the optimization objective
    net.backward(start=end)
    g = src.diff[0]
    # apply normalized ascent step to the input image
    src.data[:] += step_size/np.abs(g).mean() * g

    src.data[0] = np.roll(np.roll(src.data[0], -ox, -1), -oy, -2) # unshift image

    if clip:
        bias = net.transformer.mean['data']
        src.data[:] = np.clip(src.data, -bias, 255-bias)
```

DeepDream: set dx = x :)

jitter regularizer

"image update"

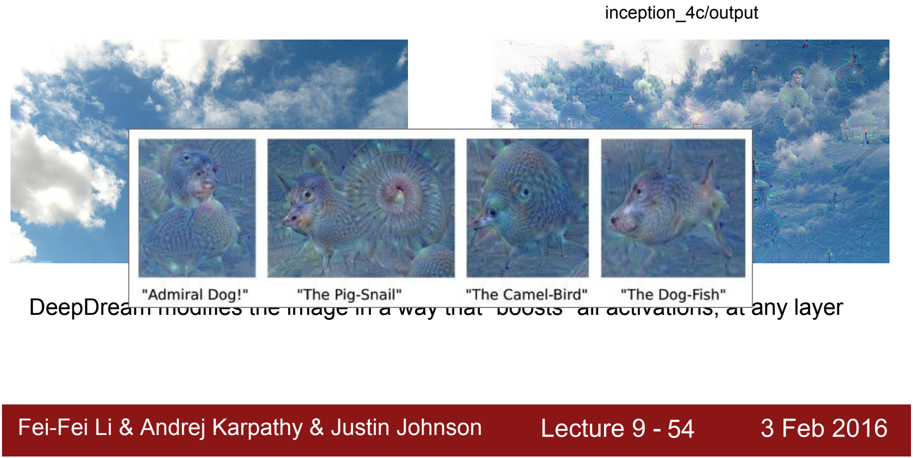
inception_4c/output

 → 

DeepDream modifies the image in a way that “boosts” all activations, at any layer

this creates a feedback loop: e.g. any slightly detected dog face will be made more and more dog like over time

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 53 3 Feb 2016


inception_4c/output
"Admiral Dog!"
"The Pig-Snail"
"The Camel-Bird"
"The Dog-Fish"
DeepDrea
any layer
Fei-Fei Li & Andrej Karpathy & Justin Johnson
Lecture 9 - 54
3 Feb 2016

inception_3b/5x5_reduce

DeepDream modifies the image in a way that “boosts” all activations, at any layer

Bonus videos

Deep Dream Grocery Trip
https://www.youtube.com/watch?v=DgPaCWJL7XI

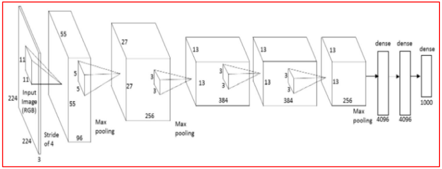
Deep Dreaming Fear & Loathing in Las Vegas: the Great San Francisco Acid Wave
https://www.youtube.com/watch?v=oyxSerkkP4o


NeuralStyle
[A Neural Algorithm of Artistic Style by Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge, 2015]
good implementation by Justin in Torch:
https://github.com/jcjohnson/neural-style
Fei-Fei Li & Andrej Karpathy & Justin Johnson
Lecture 9 - 57
3 Feb 2016

make your own easily on deepart.io

Fei-Fei Li & Andrej Karpathy & Justin Johnson | Lecture 9 - 58 | 3 Feb 2016

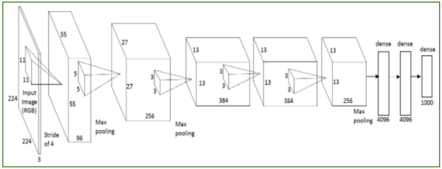
Step 1: Extract **content targets** (ConvNet activations of all layers for the given content image)



content activations

e.g.
at CONV5_1 layer we would have a [14x14x512] array of target activations

Step 2: Extract **style targets** (Gram matrices of ConvNet activations of all layers for the given style image)



style gram matrices

$$G = V^{\mathrm{T}}V$$

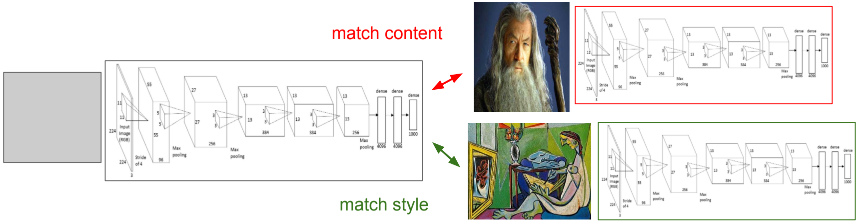
e.g.
at CONV1 layer (with [224x224x64] activations) would give a [64x64] Gram matrix of all pairwise activation covariances (summed across spatial locations)

Step 3: Optimize over image to have:

- The **content** of the content image (activations match content)
- The **style** of the style image (Gram matrices of activations match style)

$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x})$$

(+Total Variation regularization (maybe))



Fei-Fei Li & Andrej Karpathy & Justin Johnson　Lecture 9 - 61　3 Feb 2016

We can pose an optimization over the input image to maximize any class score.
That seems useful.

Question: Can we use this to “fool” ConvNets?

spoiler alert: yeah

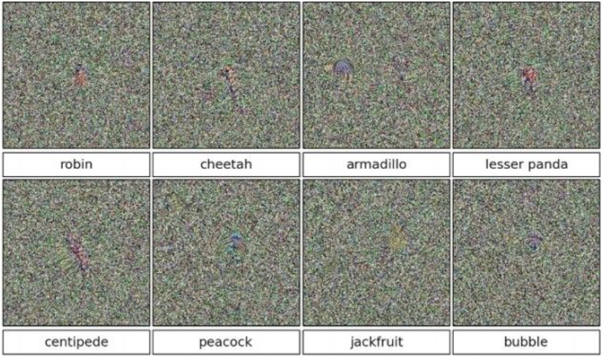
[Intriguing properties of neural networks, Szegedy et al., 2013]

correct +distort ostrich correct +distort ostrich

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 63 3 Feb 2016

[Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images Nguyen, Yosinski, Clune, 2014]

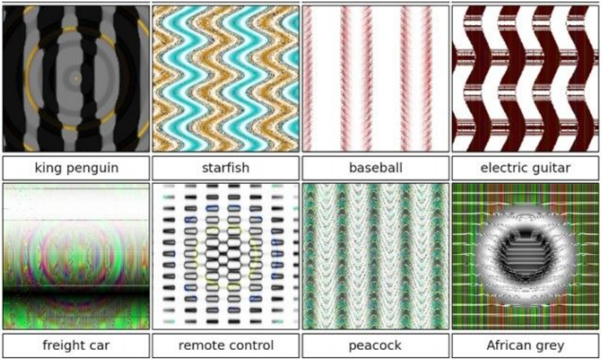
>99.6% confidences



Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 64   3 Feb 2016

[Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images Nguyen, Yosinski, Clune, 2014]

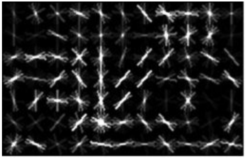
>99.6% confidences



Fei-Fei Li & Andrej Karpathy & Justin Johnson | Lecture 9 - 65 | 3 Feb 2016

These kinds of results were around even before ConvNets…

[Exploring the Representation Capabilities of the HOG Descriptor, Tatu et al., 2011]

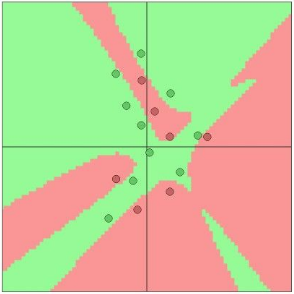


Identical HOG represention

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 66 3 Feb 2016

EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES
[Goodfellow, Shlens & Szegedy, 2014]

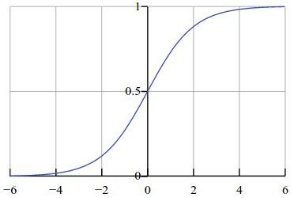
“primary cause of neural networks’ vulnerability to adversarial perturbation is their **linear nature**“



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 67 3 Feb 2016

Lets fool a binary linear classifier:
(logistic regression)

$$P(y=1 \mid x; w, b) = \frac{1}{1+e^{-(w^T x+b)}} = \sigma(w^T x + b)$$



Since the probabilities of class 1 and 0 sum to one, the probability for class 0 is $P(y=0 \mid x; w, b) = 1 - P(y=1 \mid x; w, b)$. Hence, an example is classified as a positive example (y = 1) if $\sigma(w^T x + b) > 0.5$, or equivalently if the score $w^T x + b > 0$.

Lets fool a binary linear classifier:

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights

$$P(y=1 \mid x; w, b) = \frac{1}{1+e^{-(w^T x + b)}} = \sigma(w^T x + b)$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 69   3 Feb 2016

Lets fool a binary linear classifier:

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights

class 1 score = dot product:
= -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3
=> probability of class 1 is 1/(1+e^(-(-3))) = 0.0474
i.e. the classifier is **95%** certain that this is class 0 example.

$$P(y=1 \mid x; w, b) = \frac{1}{1+e^{-(w^T x+b)}} = \sigma(w^T x + b)$$

Lets fool a binary linear classifier:

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights
adversarial x	?	?	?	?	?	?	?	?	?	?	

class 1 score = dot product:
= -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3
=> probability of class 1 is 1/(1+e^(-(-3))) = 0.0474
i.e. the classifier is **95%** certain that this is class 0 example.

$$P(y=1 \mid x; w, b) = \frac{1}{1+e^{-(w^T x+b)}} = \sigma(w^T x + b)$$

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 71   3 Feb 2016

Lets fool a binary linear classifier:

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights
adversarial x	1.5	-1.5	3.5	-2.5	2.5	1.5	1.5	-3.5	4.5	1.5	

class 1 score before:

-2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3

=> probability of class 1 is 1/(1+e^(-(-3))) = 0.0474

$$P(y=1 \mid x; w, b) = \frac{1}{1+e^{-(w^T x + b)}} = \sigma(w^T x + b)$$

-1.5+1.5+3.5+2.5+2.5-1.5+1.5-3.5-4.5+1.5 = 2

=> probability of class 1 is now 1/(1+e^(-(2))) = 0.88

i.e. we improved the class 1 probability from 5% to 88%

Lets fool a binary linear classifier:

x	2	-1	3	-2	2	2	1	-4	5	1	← input example
w	-1	-1	1	-1	1	-1	1	1	-1	1	← weights
adversarial x	1.5	-1.5	3.5	-2.5	2.5	1.5	1.5	-3.5	4.5	1.5	

class 1 score before:
-2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3
=> probability of class 1 is 1/(1+e^(-(-3))) = 0.0474

-1.5+1.5+3.5+2.5+2.5-1.5+1.5-3.5-4.5+1.5 = 2
=> probability of class 1 is now 1/(1+e^(-(2))) = 0.88
i.e. we improved the class 1 probability from 5% to 88%

This was only with 10 input dimensions. A 224x224 input image has 150,528.

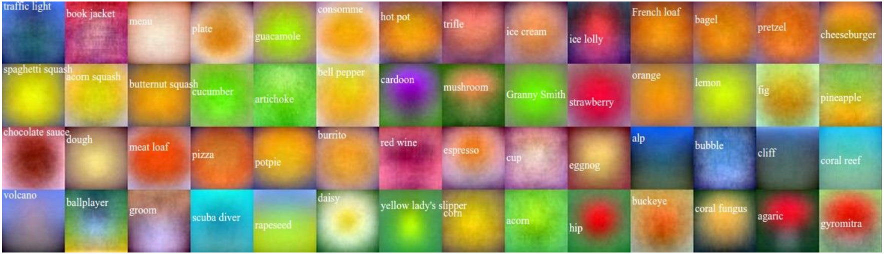
(It's significantly easier with more numbers, need smaller nudge for each)

Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 73 3 Feb 2016

Blog post: Breaking Linear Classifiers on ImageNet

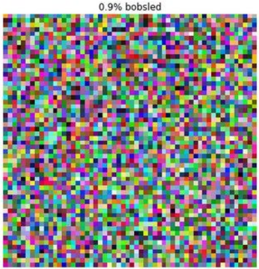
Recall CIFAR-10 linear classifiers:

ImageNet classifiers:

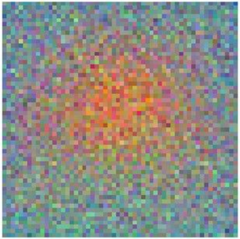


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 74 3 Feb 2016

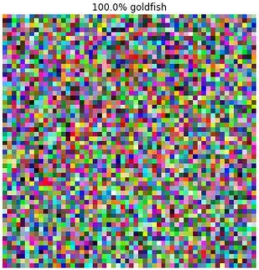
mix in a tiny bit of
Goldfish classifier weights



+

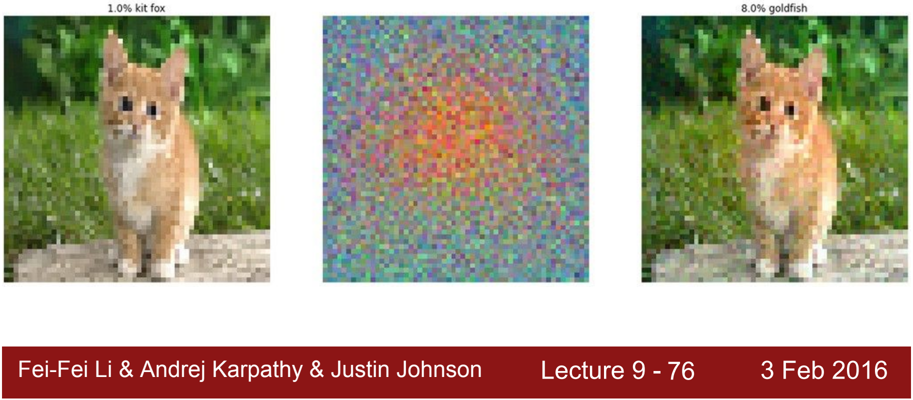


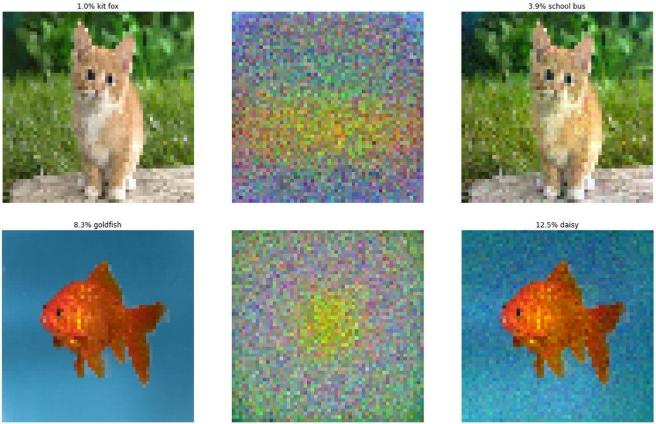
=



100% Goldfish

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 75   3 Feb 2016


1.0% kit fox
8.0% goldfish
Fei-Fei Li & Andrej Karpathy & Justin Johnson
Lecture 9 - 76
3 Feb 2016

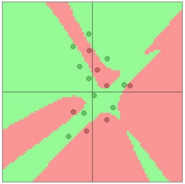


Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 9 - 77 3 Feb 2016

Conclusions

EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES [Goodfellow, Shlens & Szegedy, 2014]

“primary cause of neural networks’ vulnerability to adversarial perturbation is their **linear nature**“
(and very high-dimensional, sparsely-populated input spaces)



In particular, this is not a problem with Deep Learning, and has little to do with ConvNets specifically. Same issue would come up with Neural Nets in any other modalities.

Fei-Fei Li & Andrej Karpathy & Justin Johnson   Lecture 9 - 78   3 Feb 2016

Conclusions

- Regression and classification can be combined with CNN to achieve object localization and detection
  - Localization: CNN + regression (e.g., overfeat)
  - Detection: CNN + classification + regression (e.g., R-CNN)
- Some common tricks to speed things up
  - Use 1x1 convolution instead of FC layers
  - Rearrange order of conv layers. Do everything (finding region proposal) with convolution
- Can use optimization and backprop (deconv) to visualize weight
  - Can be used to find salient map as well
  - Probably many other uses for this trick as well. Be imaginative!
- CNN for arts (how about not visual data, how about music?)
- Unfortunately, like any other “linear” based classifier, conv-net with softmax layer at the end can be easily fooled

Conclusions

- Regression and classification can be combined with CNN to achieve object localization and detection
  - Localization: CNN + regression (e.g., overfeat)
  - Detection: CNN + classification + regression (e.g., R-CNN)
- Some common tricks to speed things up
  - Use 1x1 convolution instead of FC layers
  - Rearrange order of conv layers. Do everything (finding region proposal) with convolution
- Can use optimization and backprop (deconv) to visualize weight
  - Can be used to find salient map as well
  - Probably many other uses for this trick as well. Be imaginative!
- CNN for arts (how about not visual data, how about music?)
- Unfortunately, like any other "linear" based classifier, conv-net with softmax layer at the end can be easily fooled

Conclusions

- Regression and classification can be combined with CNN to achieve object localization and detection
  - Localization: CNN + regression (e.g., overfeat)
  - Detection: CNN + classification + regression (e.g., R-CNN)
- Some common tricks to speed things up
  - Use 1x1 convolution instead of FC layers
  - Rearrange order of conv layers. Do everything (finding region proposal) with convolution
- Can use optimization and backprop (deconv) to visualize weight
  - Can be used to find salient map as well
  - Probably many other uses for this trick as well. Be imaginative!
- CNN for arts (how about not visual data, how about music?)
- Unfortunately, like any other “linear” based classifier, conv-net with softmax layer at the end can be easily fooled

Conclusions

- Regression and classification can be combined with CNN to achieve object localization and detection
  - Localization: CNN + regression (e.g., overfeat)
  - Detection: CNN + classification + regression (e.g., R-CNN)
- Some common tricks to speed things up
  - Use 1x1 convolution instead of FC layers
  - Rearrange order of conv layers. Do everything (finding region proposal) with convolution
- Can use optimization and backprop (deconv) to visualize weight
  - Can be used to find salient map as well
  - Probably many other uses for this trick as well. Be imaginative!
- CNN for arts (how about not visual data, how about music?)
- Unfortunately, like any other "linear" based classifier, conv-net with softmax layer at the end can be easily fooled

Conclusions

- Regression and classification can be combined with CNN to achieve object localization and detection
  - Localization: CNN + regression (e.g., overfeat)
  - Detection: CNN + classification + regression (e.g., R-CNN)
- Some common tricks to speed things up
  - Use 1x1 convolution instead of FC layers
  - Rearrange order of conv layers. Do everything (finding region proposal) with convolution
- Can use optimization and backprop (deconv) to visualize weight
  - Can be used to find salient map as well
  - Probably many other uses for this trick as well. Be imaginative!
- CNN for arts (how about not visual data, how about music?)
- Unfortunately, like any other "linear" based classifier, conv-net with softmax layer at the end can be easily fooled