Project 3 Q&A

Jonathan Krause

Outline

- R-CNN Review
- Error metrics
- Code Overview
- Project 3 Report
- Project 3 Presentations

Outline

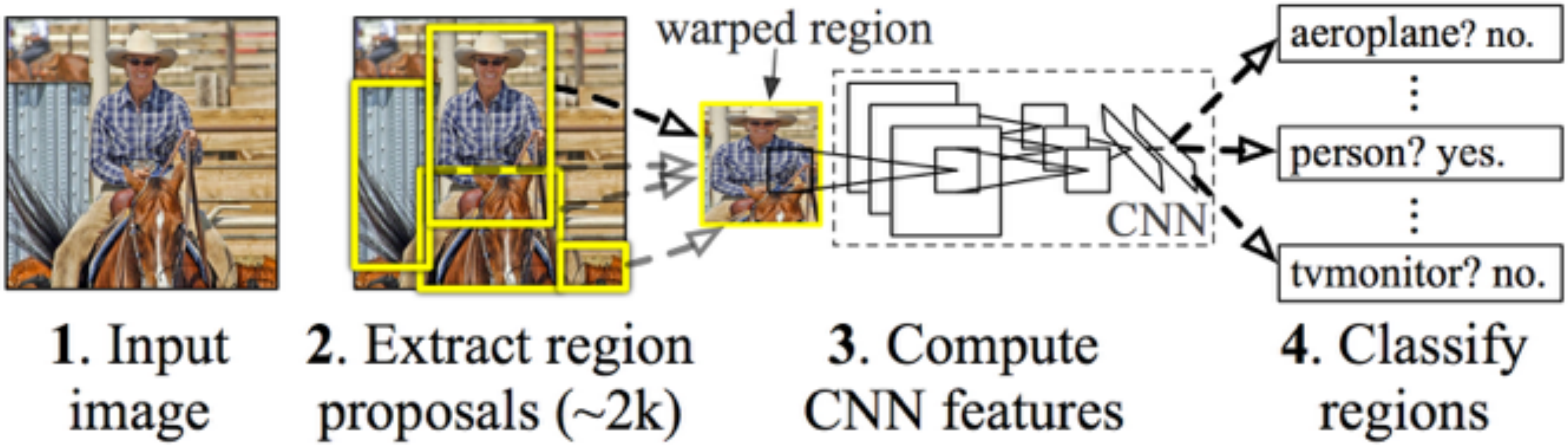
- R-CNN Review
- Error metrics
- Code Overview
- Project 3 Report
- Project 3 Presentations

R-CNN

- Selective Search + CNN
- Many design choices
- Train SVMs for detection
- Bounding box regression
- Non-max suppression

R-CNN

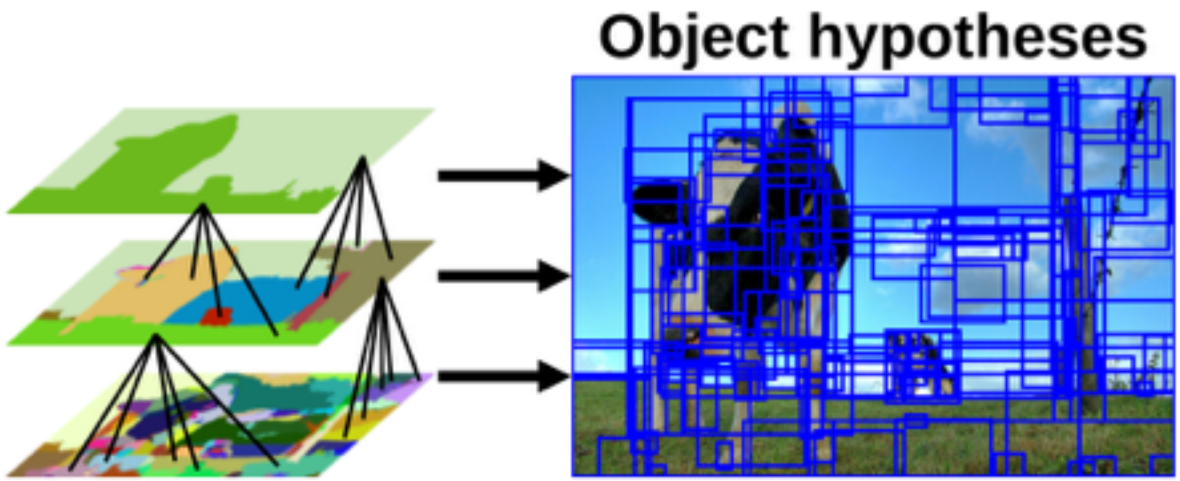
- Selective Search + CNN features



Girshick et al., 2014

Selective Search

- Generic object proposals
- Hierarchical grouping of superpixels based on color



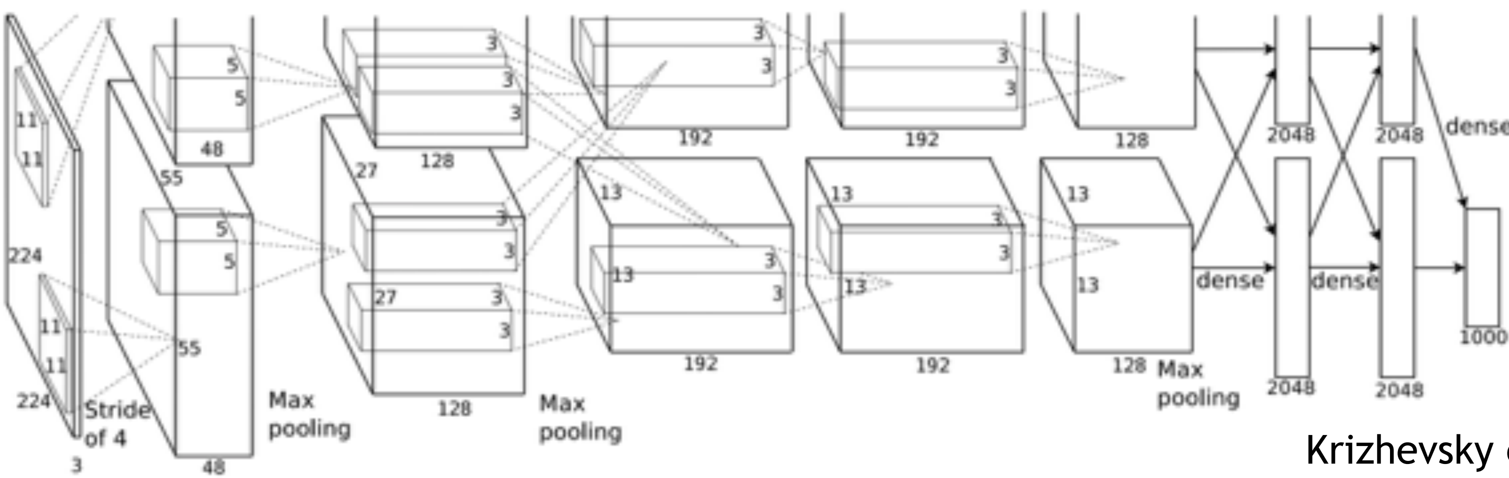
van de Sande et al., 2011

Selective Search

- A few sec/image (CPU)
- Depends on image resolution!
- 2,307 regions/image on average for our images
- Given to you in Project 3

CNN Features

- Typically pre-train on ImageNet
- Can fine-tune on detection data
- The better the CNN for classification, the better it will be for detection

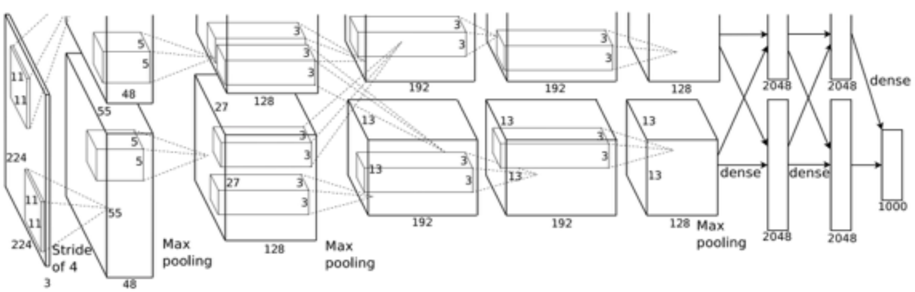


Krizhevsky et al. 2012

Network Choice

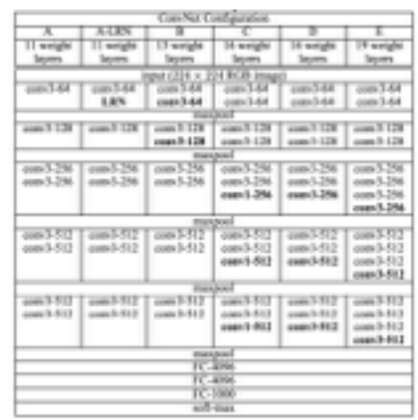
AlexNet

- Krizhevsky, Sutskever, Hinton
- NIPS 2012
- ILSRVC Top-5 Error: 18.2%
- R-CNN AP: 58.5



VGGNet

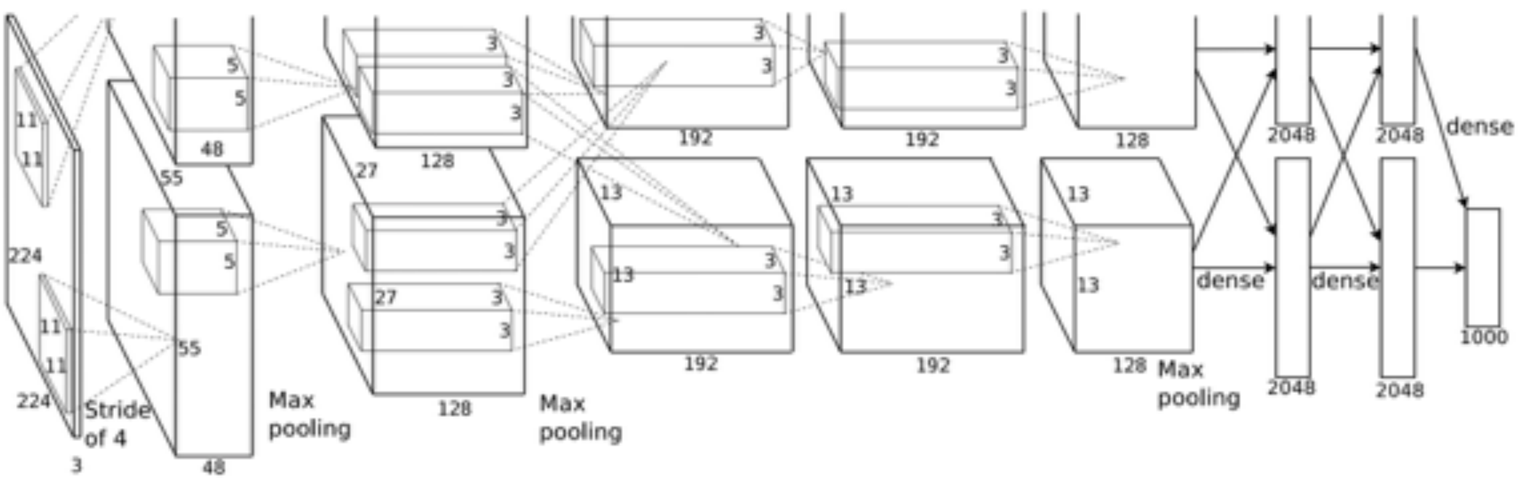
- Simonyan and Zisserman
- ICLR 2015
- ILSVRC Top-5 Error: 7.5%
- R-CNN AP: 66.0



Which Layer?

- Just try out a few high-level layers

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN pool$_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc$_6$	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc$_7$	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT pool$_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc$_6$	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc$_7$	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2

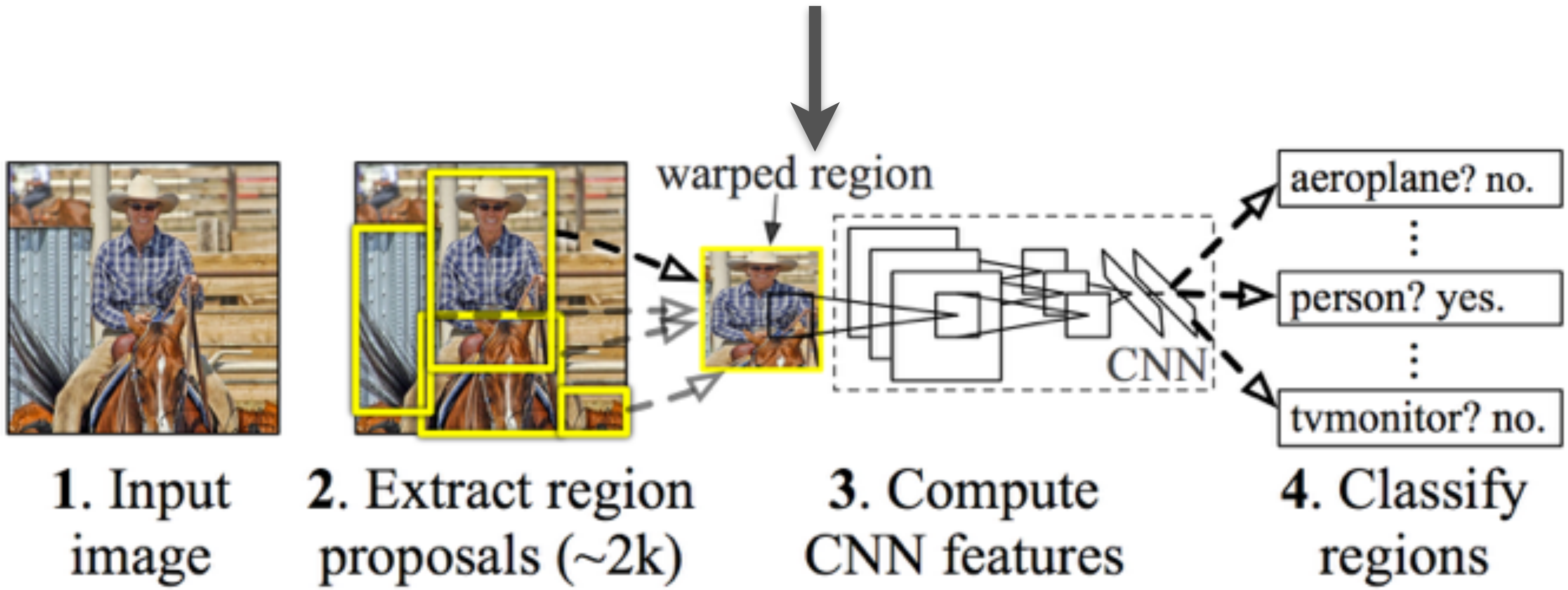


Our Network

- Took pre-trained AlexNet
- Replaced 4096-d FC layers with 512-d
  - Reduces size of extracted features with some performance loss
- Trained on ILSVRC (i.e. no fine-tuning)

R-CNN: Extracting Features

- Extract CNN features around a region
- But CNNs take a fixed-size input!



Girshick et al., 2014

Extracting Features

- Need region to fit input size of CNN
- Region warping method:

Girshick et al., 2014

Extracting Features

- Context around region
- 0 or 16 pixels (in CNN reference frame)

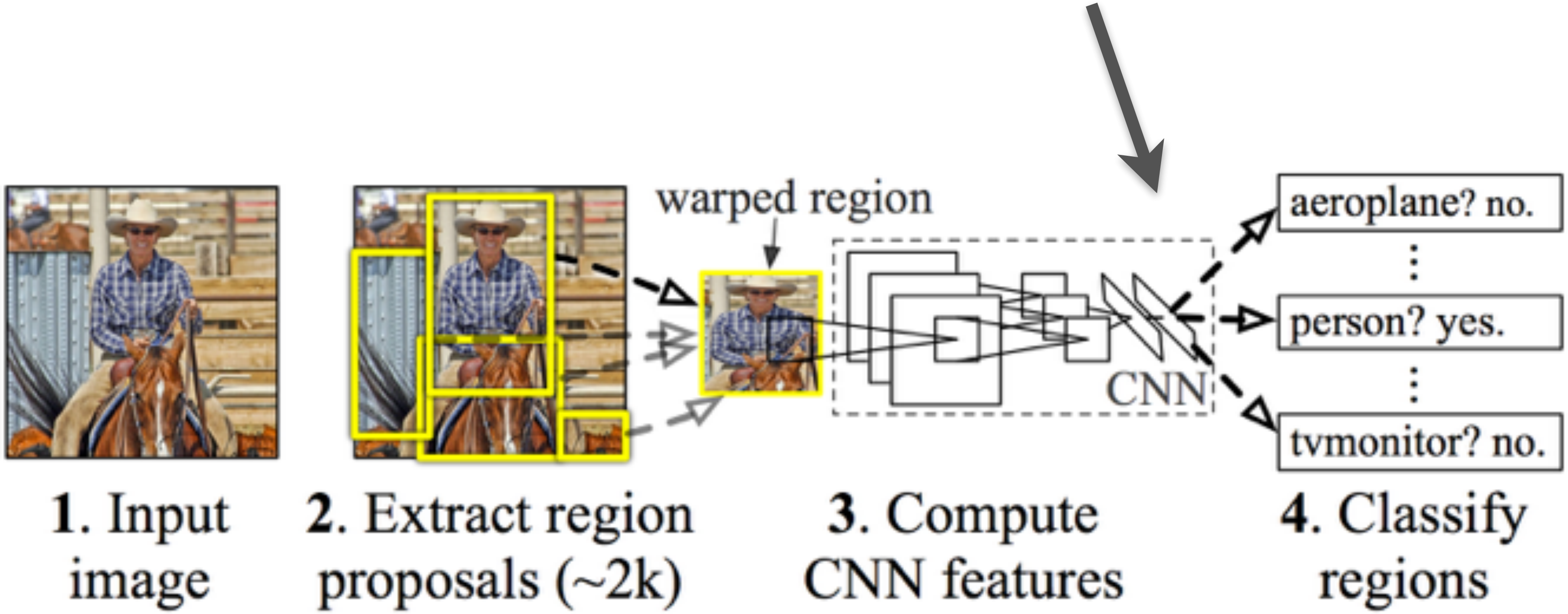
Girshick et al., 2014

Extracting Features

- Takes 15-20 sec/image with a good GPU
- Easily the slowest part for Project 3
- Do this part early!!

R-CNN Detector

- Binary SVM for each class on regions
- Lots of implementation details!



Girshick et al., 2014

SVM Training

- Which regions should be positive vs negative?
- Weights on positive/negative examples
- What type/strength of regularization should you do?
- Feature normalization?
- Use a bias?
- Memory constraints (the big one)

Positives/Negatives

- Positives: overlap $\geq$ threshold_1
- Negatives: overlap $\leq$ threshold_2
- Read the paper to get good choices of thresholds/experiment!

Positive/Negative Weights

- Typically have way more negatives than positives
  - Can lead to favoring negatives too much

- Solution: Weigh positives more in SVM training
  - Many solvers have an option for this

Regularization

- SVMs need regularization
- L_1 or L_2 regularization?
- What strength?
- Cross-validate this or subsample training to get validation set.
- Super important!

Feature Normalization

- Often necessary to get high-dimensional SVMs to work.
- Options
  - Zero norm, unit standard deviation
  - L_1/L_2-normalize
  - Make features have a certain norm *on average*
  - Make each dimension fit in range [a,b] (e.g. [-1,1])
- Most of these work fine.

Bias

- Add a bias to SVMs by augmenting features with a 1 (non-zero constant).
- Most SVM solvers (e.g. liblinear) have an option for this.
- Important when class imbalance
- Do this!

Memory Constraints

- Features take up a lot of space!
  - Typically hundreds of GB
  - For us, only 2-3 GB (smaller CNN, fewer images)

- Even if you have enough memory, training an SVM on that much data is slow

- Subsample negatives: *hard negative mining*

Hard Negatives

- Hard as in "difficult"
- Only keep negatives whose decision value is high enough
  - Specific to max-margin, but can be used with other classifiers
- Problem: Need classifier to get decision values in the first place!
- Solution: Iteratively train SVMs

Training SVMs

For each image:

1. Add as positives all regions with sufficient overlap
2. Add as negatives all regions with low enough overlap with large enough decision values according to current model
3. Retrain SVM if it's been too long (for some definition of "too long")

Repeat for some number of epochs

Implementation Notes

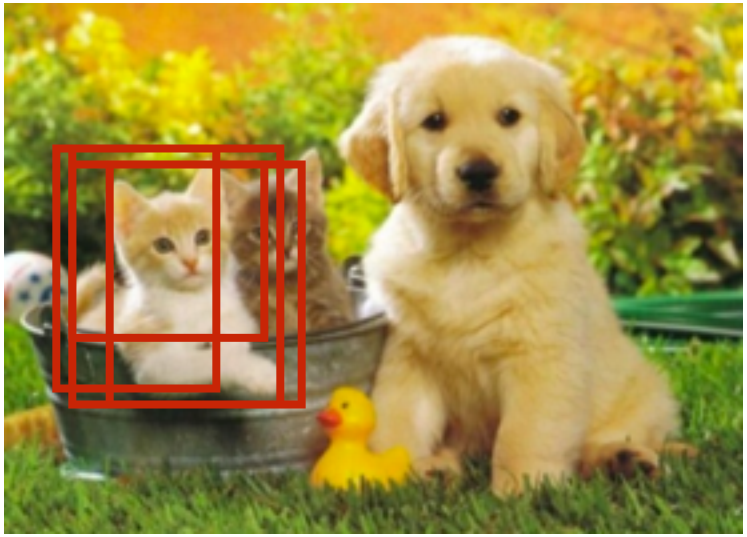
- Use an SVM solver that's memory efficient (i.e. uses single precision, doesn't copy all the data)
- Try training with SGD?
- Runtime performance largely determined by number of negatives

Bounding Box Regression

- Predict new detection window from region-level features
  - R-CNN uses $pool_5$ features, use those or the default fc_6 ones provided (probably $pool_5$ works better)
- Class-specific
- Ridge regression on bounding box offset (c_x, c_y, log(width), log(height))
- Regularization amount super important

Non-max suppression

- Turn multiple detections into one
- Approach: merge bounding boxes with ≥ threshold IoU, keep the higher scoring box.
- Threshold of 0.3 is decent



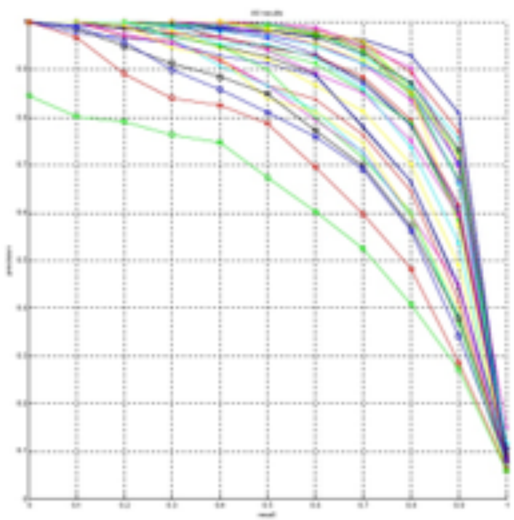
R-CNN Questions?

Outline

- R-CNN Review
- Error metrics
- Code Overview
- Project 3 Report
- Project 3 Presentations

Average Precision

- Detection is correct if IoU ≥ 0.5 with ground truth
  - Can't have multiple detections for one GT box
- Rank by detection score
- Get area under the curve (roughly)
- Mean AP (mAP) averages across classes



“Baseline” Performance

- Before bounding box regression:
  - Car: 30.72
  - Cat: 35.91
  - Person: 18.83
  - mAP: 28.49
- With bounding box regression:
  - Car: 32.97
  - Cat: 38.58
  - Person: 20.05
  - mAP: 30.53
- Try to get this without any major changes!

Outline

- R-CNN Review
- Error metrics
- Code Overview
- Project 3 Report
- Project 3 Presentations

What We Provide

- `readme.txt`: Contains more details about all of this. Read this in detail!
- `detection_images.zip`: The images. Download from course website (110 MB)
- `{train,test}_ims.mat`: Annotations for all images.
- `ssearch_{train,test}.mat`: Selective search regions (as bounding boxes)
- `extract_cnn_feat_demo.m`: Demo script extracting CNN features with caffe

What We Provide

- `Makefile.config.rye`: A Makefile you can use if you run on the `rye` farmshare machines. Change g++ version to 4.7 if on `rye02`.
- `ilsvrc_mean.mat`: Mean image for the CNN
- `cnn_deploy.prototxt`: CNN architecture for extracting features (fc_6).
- `cnn512.caffemodel`: Learned CNN weights

What We Provide

- `display_box.m`: Visualizes a bounding box
- `det_eval.m`: Evaluates precision, recall, AP for a single class
- `boxoverlap.m`: Calculates IoU for many bounding boxes at once (fast).

What We Provide

- Implement these:
  - `extract_region_feats.m`
  - `train_rcnn.m`
  - `train_bbox_reg.m`
  - `test_rcnn.m`

extract_region_feats.m

- Extract features around for each region in every image
- Also extract them around the ground truth bounding box (for training images)
- Save them for use later

- Note: This will take a long time to run. Do this early!

train_rcnn.m

- Train the classifier on top of CNN features
- Be careful about hard negative mining and all the other parameters!
- Might take a bit of iteration to get this right, but should run relatively fast (less than an hour with a relatively bad implementation)
- Debug with a single class first!

train_bbox_reg.m

- Train the bounding box regressor
- Independent of the classifier
- Be careful about bounding box and offset representation!
- Pay attention to regularization!

`test_rcnn.m`

- Run the trained R-CNN on test images
- Run the bounding box regressor
  - Should be able to turn this on and off
- Do non-maximum suppression
  - Code this up yourself
- Do evaluation
  - Code given for single-class evaluation

Code Subtleties

- It may take some time to get caffe working
  - Ask the TAs if it takes more than a couple hours to get the demo script running

- To extract features from multiple regions at once, need to change the first `input_dim` in `cnn_deploy.prototxt` before initializing caffe.

Results to Report

- AP for each class with and without bounding box regression
- At least one qualitative result per class
- Quantitative and qualitative results for any changes made

Outline

- R-CNN Review
- Error metrics
- Code Overview
- Project 3 Report
- Project 3 Presentations

Project 3 Report

- Write-up template provided on website (link)
- Use CVPR LaTeX template
- No more than 5 pages (additional figures ok)
- Rough sections:
  1. Overview of the field (i.e. detection)
  2. The algorithm (how R-CNN works)
  3. Any changes/extensions made
  4. Code README
  5. Results

Notes from Grading Project 2

- Much better than Project 1 reports :)
- If you tried out something and it worked worse, quantify it!
- When identifying a failure mode, qualitative results are good

Extensions

- Need at least a few extensions (depending on scope). The more (and the higher quality) the better.

Possible Extensions

- Feature representation
  - Compare CNN with other vision features
- Which layer of CNN to use
  - Maybe fc_6 is bad when only 512-d?
- Parameters used during training
  - Regularization, overlap thresholds…
  - Try to draw insight!

Possible Extensions

- Classifier
  - Something better than SVM? Random forest?
- Larger CNN
  - AlexNet or VGGNet?
- Fine-tuning
  - How much does it help in this case?

Possible Extensions

- Other detection methods
  - DPM? HOG? Other?
- Other region proposals?
  - Edge Boxes? Objectness? Your own?
- Segmentation
  - Combine project 1 and 3

Possible Extensions

- Fancier training?
  - Dropout?
- Classification via detection
  - What changes do you have to make?
- Joint classification + bbox reg training
  - Does it help?

Possible Extensions

- NMS
  - Something better than greedy picking?
- Multi-label prediction
  - Predict attributes of objects?

Possible Extensions

- Make it faster
  - Faster training? Filter out regions?
- Make it better
  - Add some other signal???
- Analyze it
  - What really makes R-CNN tick?

Outline

- R-CNN Review
- Error metrics
- Code Overview
- Project 3 Report
- Project 3 Presentations

Project 3 Presentations

- Every team should submit 4-5 slides to me (jkrause@cs) by 5 pm the day before (Tues June 2)
- You know the drill

Late Days

- Reminder: Total of 7 late days spread across the 3 assignments
- 20% off per late day afterward
- Most of you have already used up a lot of late days, check with TAs if you need to find out the exact number you have left.

Important Dates

- June 2 (5 pm): Send presentations to jkrause@cs
- June 3 (in class): Presentations
- June 4 (5 pm): Reports due

Questions?

You're almost done!