*Rich feature hierarchies for accurate **object detection** and semantic segmentation*

BY; ROSS GIRSHICK, JEFF DONAHUE, TREVOR DARRELL AND JITENDRA MALIK

PRESENTER; **MUHAMMAD OSAMA**

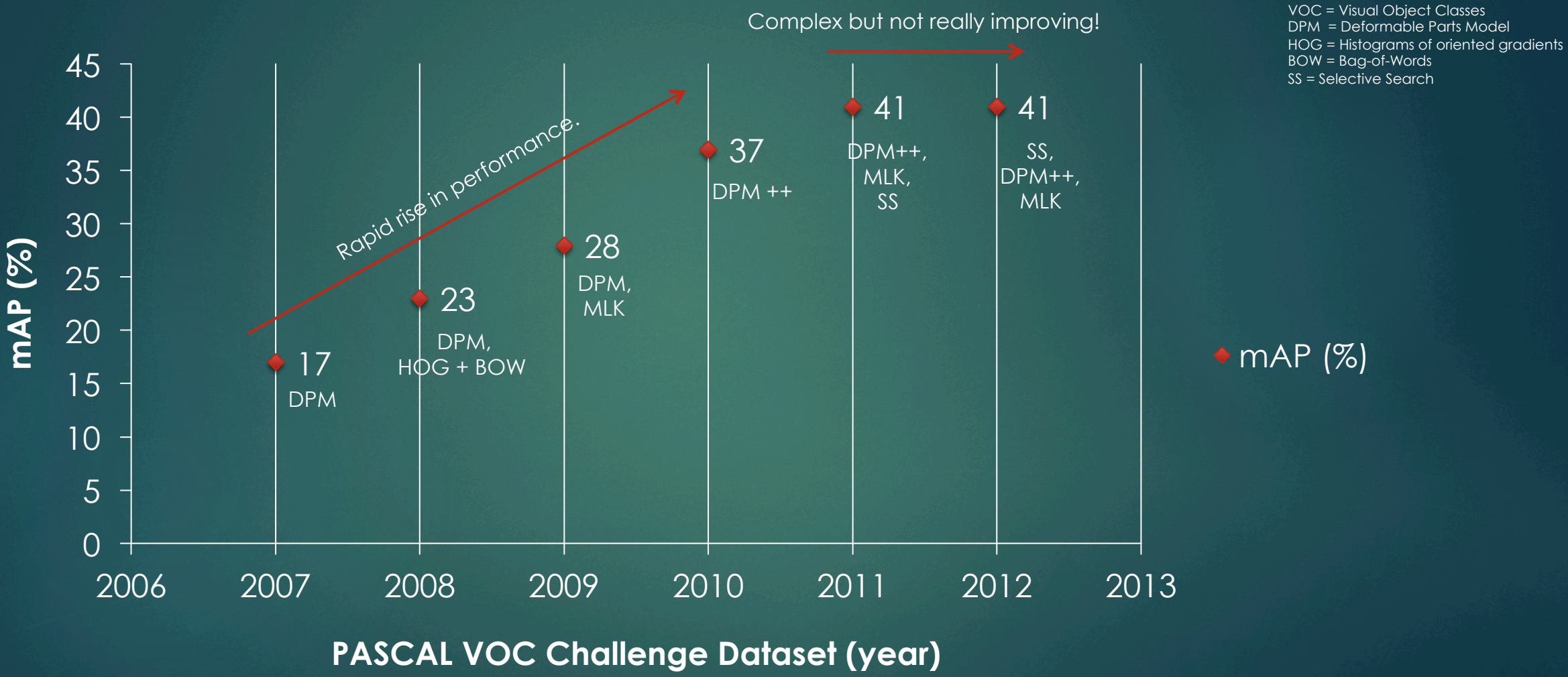
Object detection vs. classification

- **Detection:** Process of identifying the object (yes or no). Sometimes difficult because the focus is just on the object, you have to localize the object in the image (context is often ignored).

- **Classification:** Process of categorizing the image based on previously described properties (training).

Source: http://stackoverflow.com/questions/31750076/

Problem with object detection?



VOC = Visual Object Classes
DPM = Deformable Parts Model
HOG = Histograms of oriented gradients
BOW = Bag-of-Words
SS = Selective Search

Source: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc20{07,08,09,10,11,12}/results/index.html

“

Features *matter!*

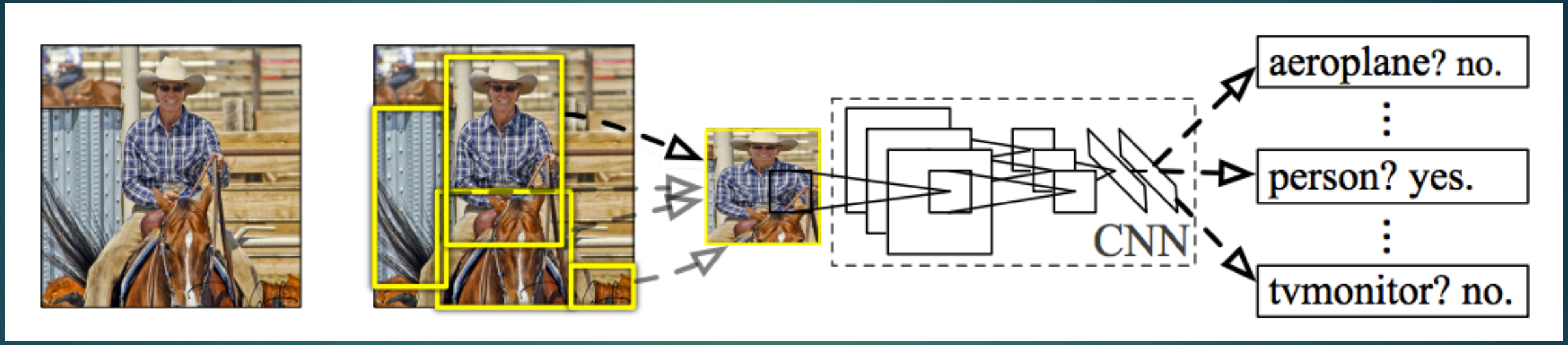
”

Proposed solution; Break through the PASCAL plateau with **feature learning**!

Core ideas

- Understand if the CNN can be used as an object detector.
- Evaluate results after different layers of the process.

R-CNN: Regions with CNN Features



1. Input Image

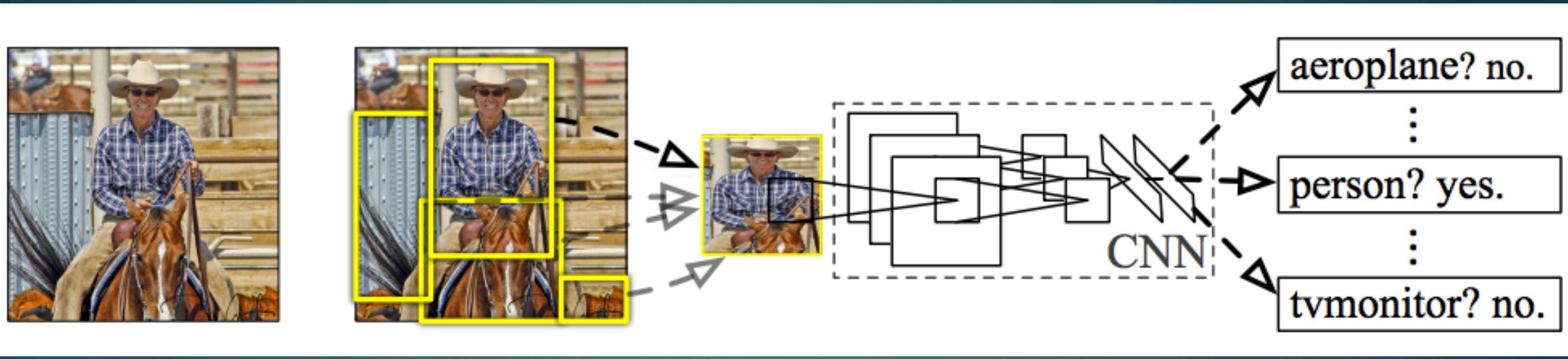
2. Extract region Proposals (~2k/image)

3. Compute CNN features

4. Classify regions (linear SVM)

Image Source: From the paper.

R-CNN: Extract Region Proposals

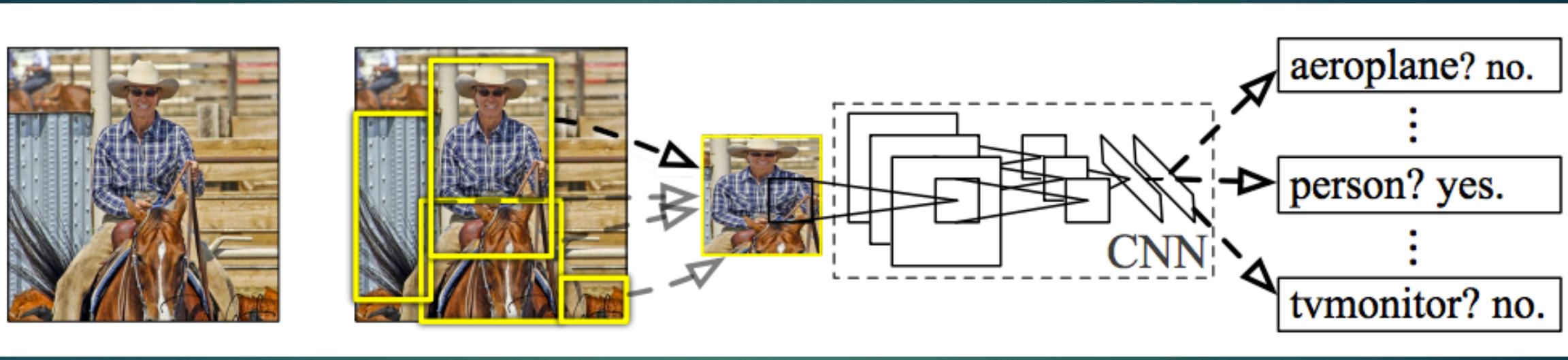


1. Input Image

2. Extract region Proposals (~2k/image)

- Proposal method:
  - Selective Search [van de Sande, Uijlings et al.]
    - Over segmentation.
    - Bottom-up grouping.
    - Different region proposals.
    - Reduce of false positives.

R-CNN: Compute CNN Features

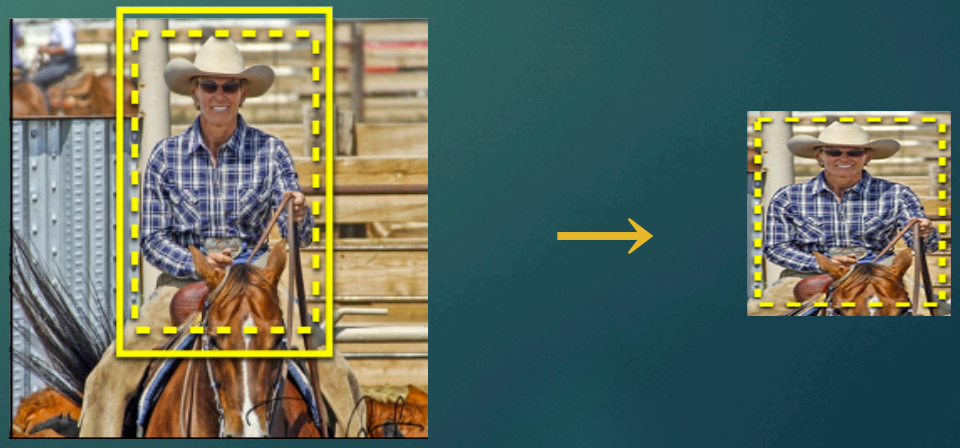


1. Input Image

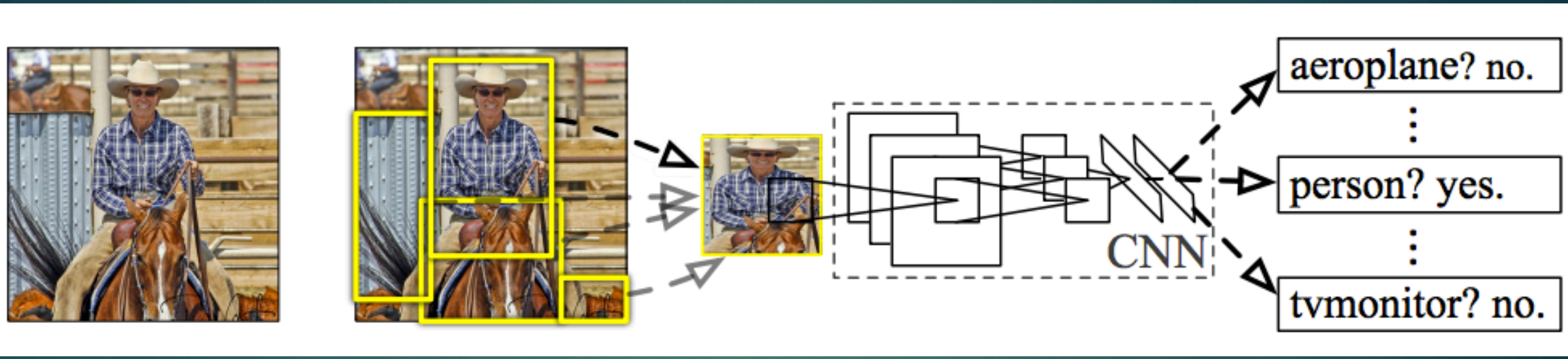
2. Extract region Proposals (~2k/image)

3. Compute CNN features

- Dilate proposal
- Crop
- Scale (227x227px)
- Overlapping Max Pooling
- 5 convolution layers
- 2 fully connected layers
- Forward propagate Output: "fc7" features



R-CNN: Linear Classifier



1. Input Image

2. Extract region Proposals (~2k/image)

3. Compute CNN features

4. Classify regions (linear SVM)

- Linear Classifier (SVM or Softmax) to classify the object.
- **Negatives;** Overlap threshold < 0.3

R-CNN: Refining object proposal

Original Proposal

Linear regression

On CNN features

Predicted Object Bounding Box

Source: http://www.cs.berkeley.edu/~rbg/slides/rcnn-cvpr14-slides.pdf

R-CNN: Results on PASCAL VOC

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [18]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [34]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [36]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [16]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table Source: From the paper.

- **Timing;**
  - Training SVM for all classes takes 1.5 hours (single core).
  - Extracting feature takes 5 ms (GPU).
  - Matrix multiplication for 100,000 classes takes 10 seconds.
- **Improvements;**
  - mAP of 53.7%!
  - *Google Dean et al. paper (CVPR best paper): 16% mAP in 5 minutes. Here 53% in about 1 minute!*
- **R-CNN BB;** R-CNN with Bounding Box regression.

R-CNN: Top bicycle FPs (AP=72.8%)

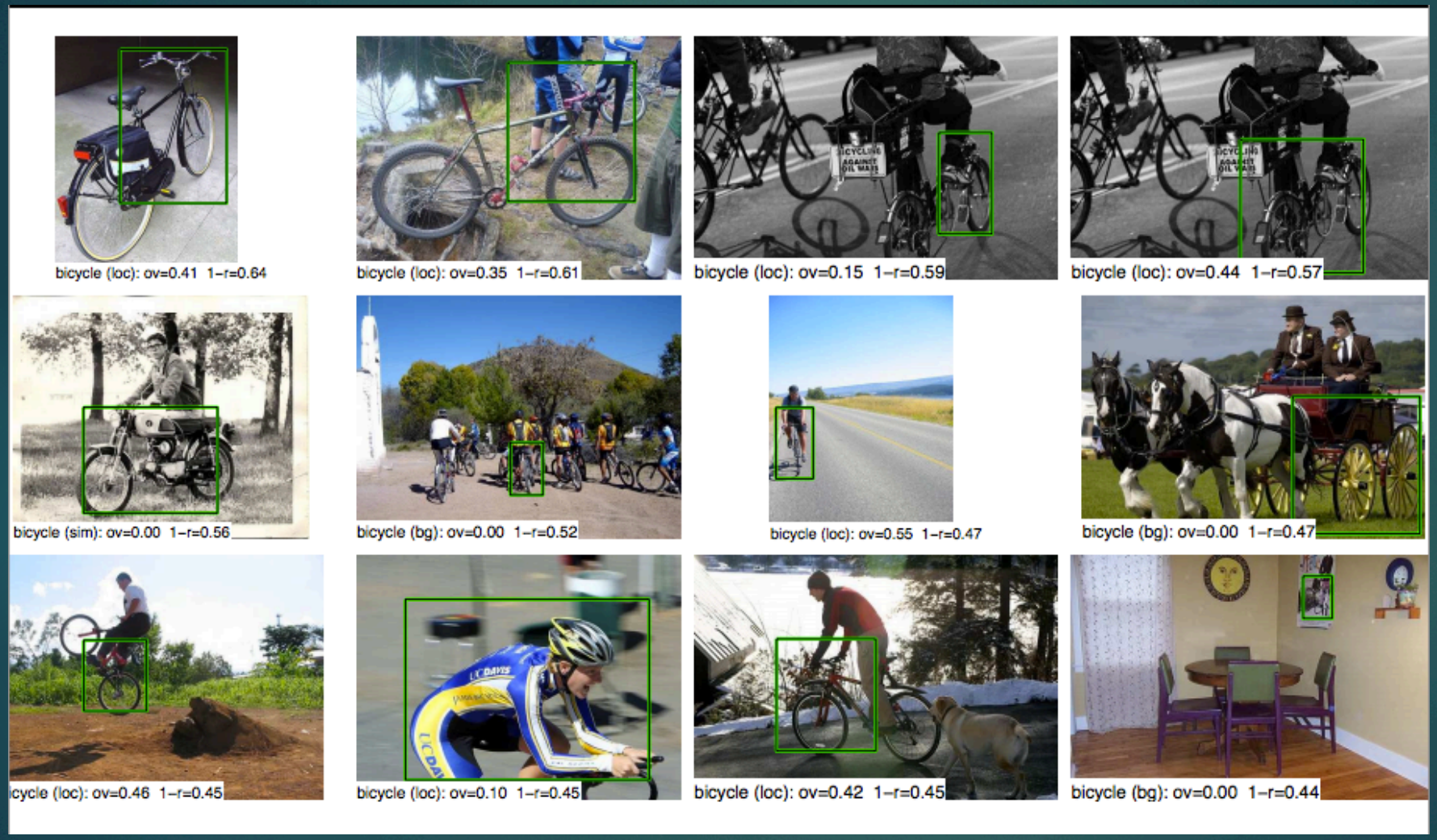
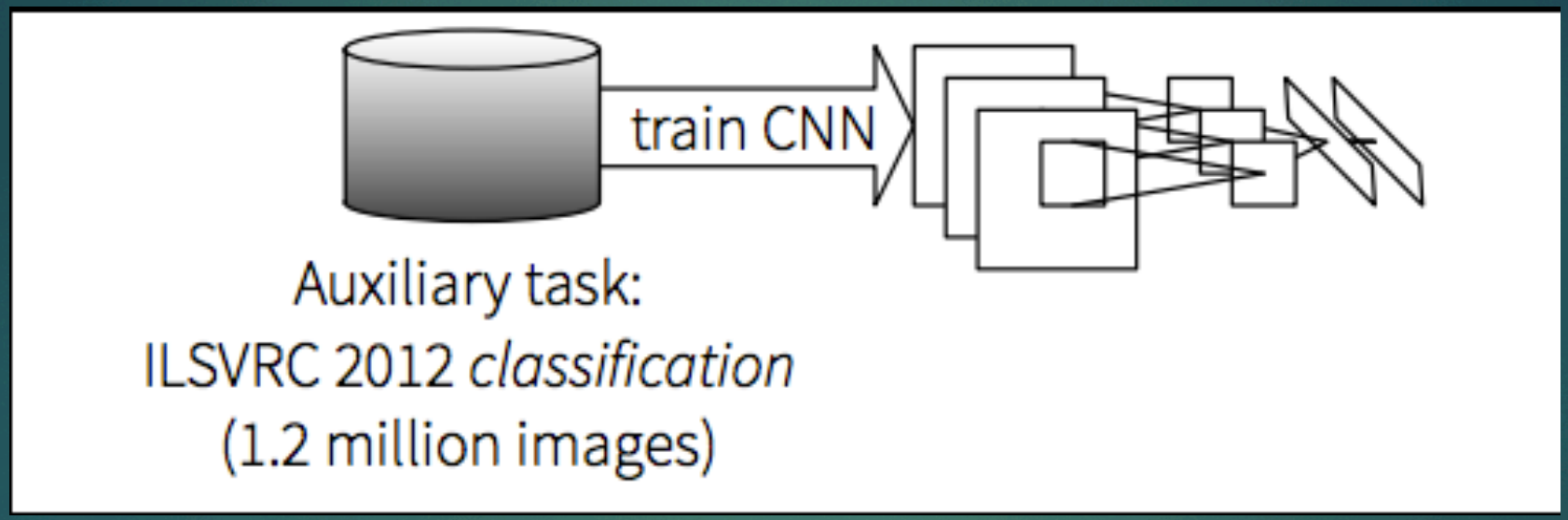


Image Source: http://www.cs.berkeley.edu/~rbg/slides/rcnn-cvpr14-slides.pdf

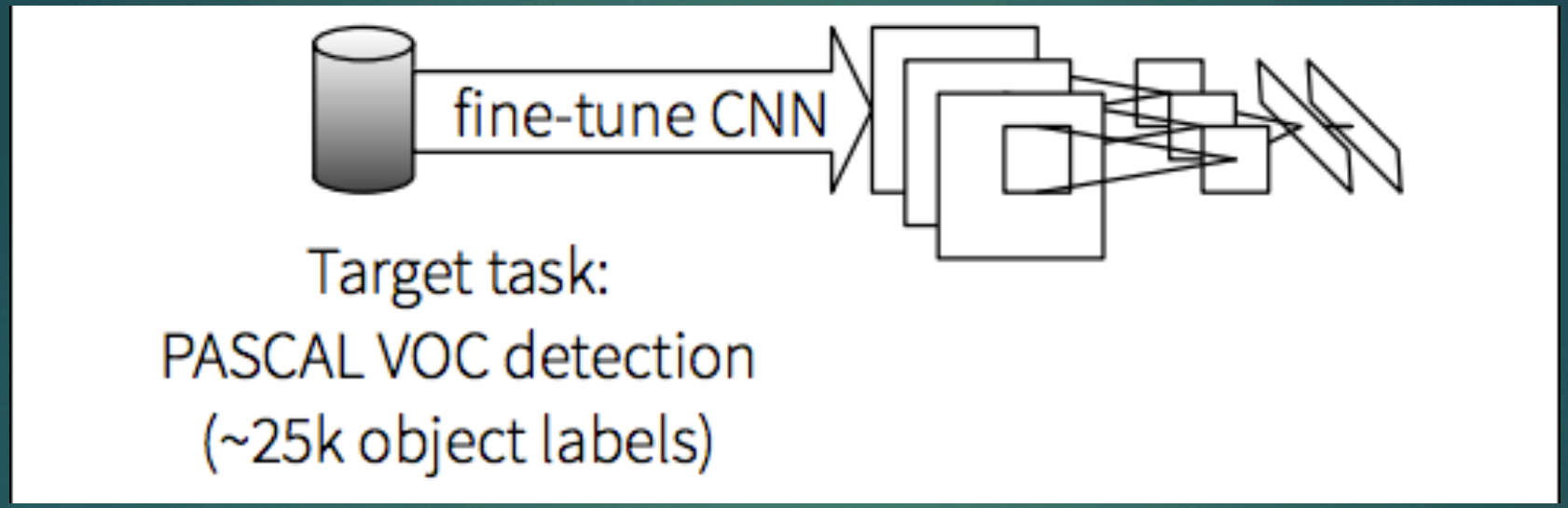
Training R-CNN: Supervised pre-training



- Train SuperVision CNN for 1000-way ILSVRC image classification task.

Source: http://www.cs.berkeley.edu/~rbg/slides/rcnn-cvpr14-slides.pdf

Training R-CNN: Fine-tune the CNN for detection



- Transfer the representation learned for ILSVRC classification to PASCAL (or ImageNet detection).

Source: http://www.cs.berkeley.edu/~rbg/slides/rcnn-cvpr14-slides.pdf

Training R-CNN: Train detection SVMs

- With the sofmax classifier from fine-tuning mAP decreases from 54% to 51%.

Source: http://www.cs.berkeley.edu/~rbg/slides/rcnn-cvpr14-slides.pdf

Visualization

- Single out a particular unit (feature).
- Use it as a object detector in its own right.
  - Compute unit's activation on 10 million held-out regions.
  - Sort from highest to lowest activation.
  - Display top-scoring regions.

Visualization: Top regions for six $pool_5$ units.

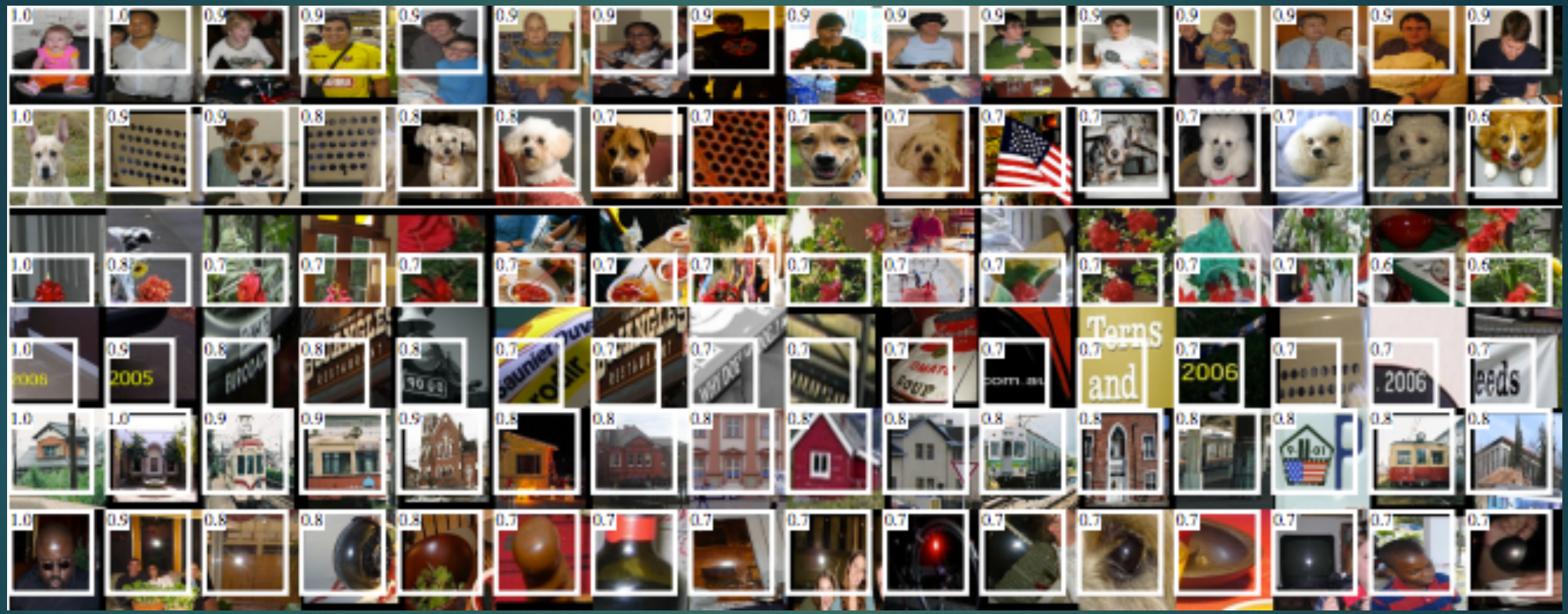


Image Source: From the paper.

- Row 1 and 4: Aligned by concepts (humans and texts).
- Row 2 and 6: Capture texture and material properties (dot arrays and reflections).

Ablation studies

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
R-CNN FT fc_7 BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**
DPM v5 [18]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [26]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [28]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

Table Source: From the paper.

- **FT;** Fine tuned layers.
- Not much difference between mAP of $pool_5$ vs. fc_6 and fc_7 given;
- R-CNN $pool_5$ only uses 6% of all parameters (out of ~60 million parameters).
- **Finding;** Much of the CNN's representational power comes from its convolutional layers, rather than from the much larger densely connected layers.

Semantic Segmentation

- **Strategy 1:** (full) ignores the region's shape and computes CNN features directly on the warped window.
  - It ignores the non-rectangular shape of the feature.
- **Strategy 2:** (fg) computes CNN features only on a region's foreground mask.
  - Replace background with mean input to zero them out.
- **Strategy 3:** (full + fg) concatenates full and foreground strategies.

Semantic Segmentation: Results

	full R-CNN		*fg* R-CNN		*full+fg* R-CNN	
O_2P [4]	fc_6	fc_7	fc_6	fc_7	fc_6	fc_7
46.4	43.0	42.5	43.7	42.1	**47.9**	45.8

Table Source: From the paper.

- **O_2P;** Second-order pooling, current leading semantic segmentation system.
- Training the 20 SVRs on our full+fg features takes an hour on a single core, compared to 10+ hours for training on O2P features.
- Improved result of 47.9% accuracy, outperforming O_2P's 46.4%.

Take away

- Improved PASCAL VOC mean average precision.
- R-CNN outperforms previous hand-crafted features based methods.
- Detection speed is manageable (~11s/image).
- Scales well with number of categories (30ms for 20 -> 200 classes).
- Simple implementation and open source.

Evolution of **R-CNN** (Source: Gource Visualization)

https://github.com/rbgirshick/rcnn

QUESTIONS?