RICH FEATURE HIERARCHIES FOR ACCURATE OBJECT DETECTION AND SEMANTIC SEGMENTATION

By: Ross Girshick, Jeff Donahue,
Trevor Darrell, Jitendra Malik.

Presentation on R-CNN

Hands-on R-CNN full knowledge

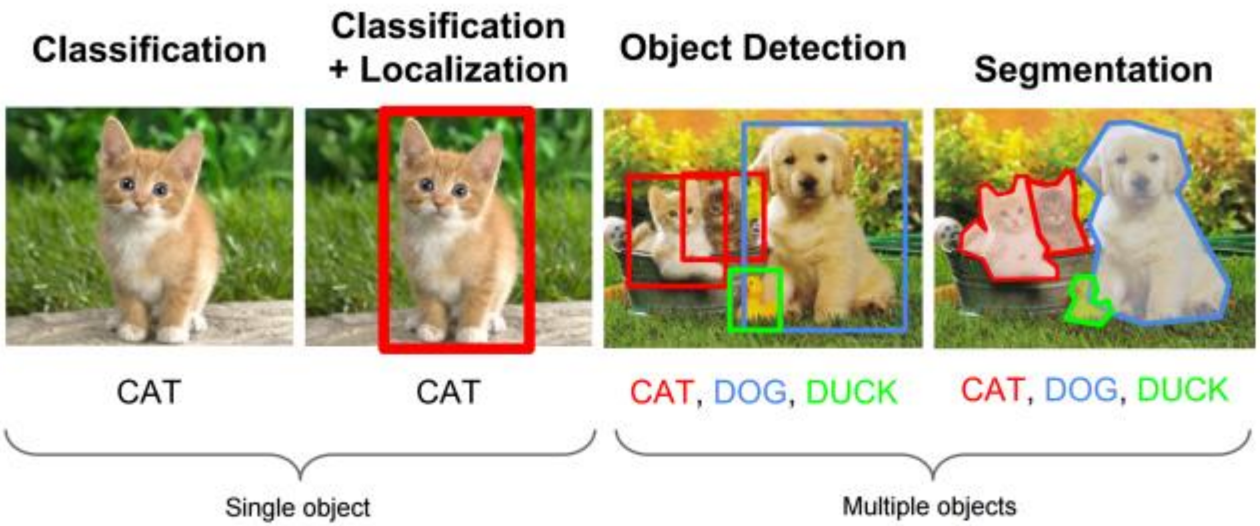
Presented by: Kamal Zakieldin
University of Innsbruck, Austria.

AGENDA

- **Introduction**
  - Problem Overview
- **Terminologies**
- **Paper's Discussion**
  - Intro. Section.
  - Object detection Section.
  - Visualization, results and Ablation studies Section.
  - Datasets
  - Semantic Segmentation
- **Further Work**
  - Comparisons
  - Results
- **Further Questions**
- **References**

Introduction

- Problem Overview
  - It's important to notice that classification has huge previous contribution.
  - Good contributions can be found in object detection.
  - But Segmentation has contributions less than the other problems.



  - The paper is working on object detection and segmentation in single and multiple objects in the same image.

Terminologies

PASCAL VOC

PASCAL2

Pattern Analysis, Statistical Modelling and Computational Learning

PASCAL Visual Object Classes Challenge

- To evaluate algorithms for **object detection, classification and segmentation**.
- Last Competition held in 2012, but evaluation server still running for evaluating algorithms performance.
- 20 classes, ~20K images, ~25K labeled objects.

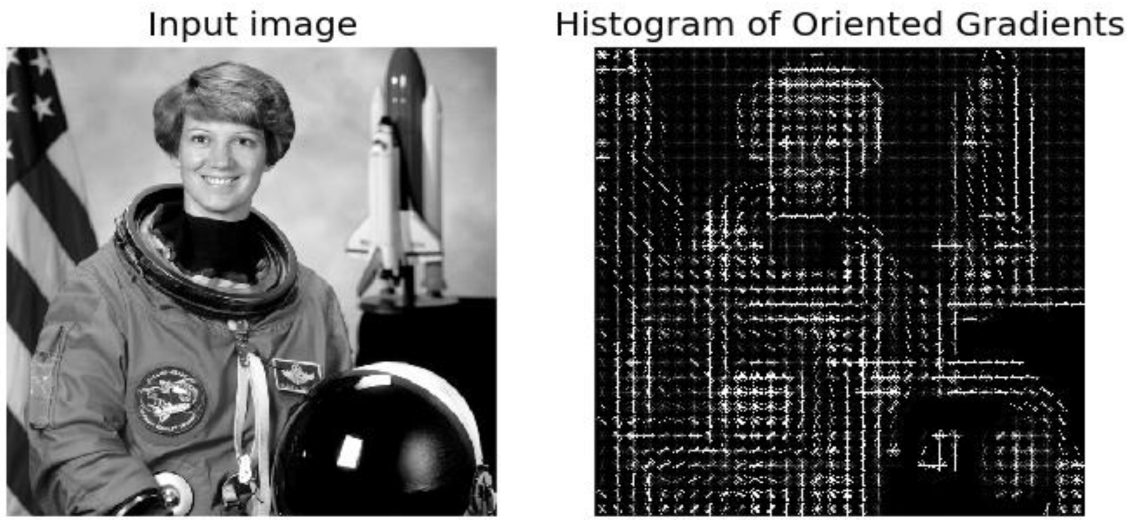
ILSVRC

ImageNet Large Scale Visual Recognition Challenge

- To evaluate algorithms for **object detection** and **image classification** at large scale.

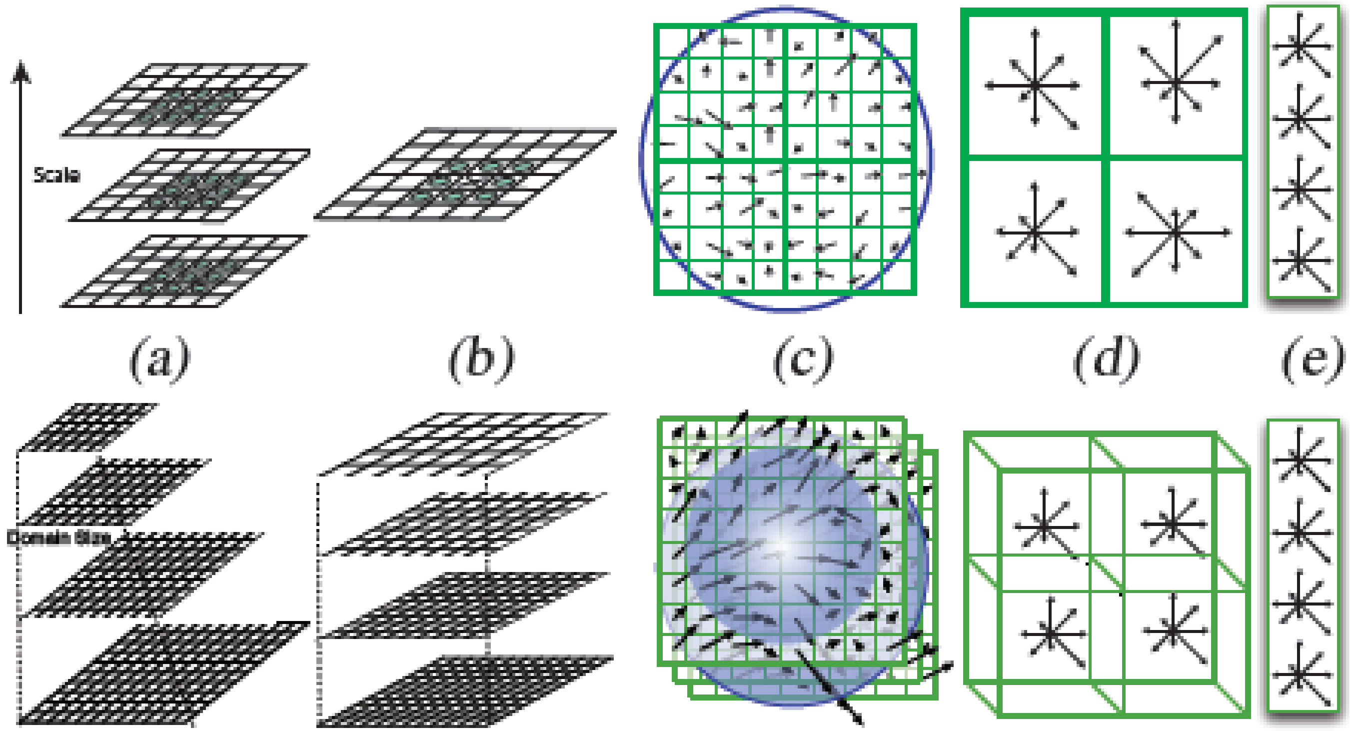
- Over 14 Million labeled images.
- Object Localization for 1000 Classes (Categories).
- Object Detection for 200 fully labeled Classes.

(Histogram of oriented gradient) HOG



- Using **feature representation** and **orientation**
- Compute the **histogram** for all oriented features.

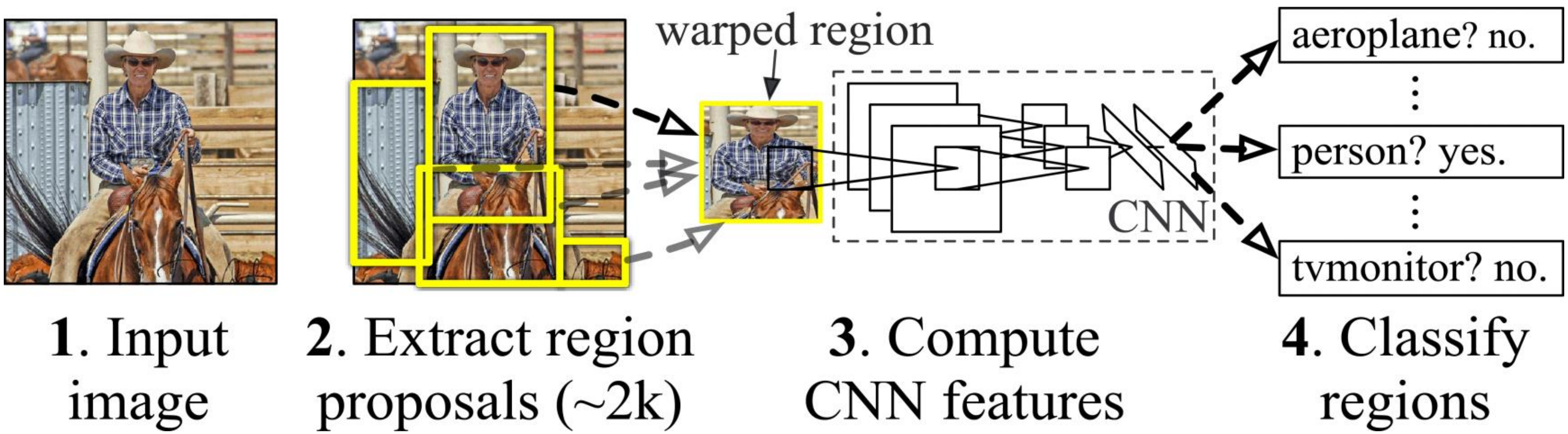
(Scale Invariant Feature Transform) SIFT



- Transfer Image content into **local features** by using the **Difference of Gaussian (DoG)**.
- **Sensitive** to any changes in pixels (rotation, scale, illumination , …).

Paper's Work

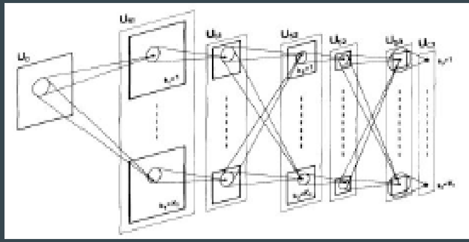
R-CNN: Region-based Convolutional Network



- apply high-capacity convolutional neural networks (**CNNs**) to bottom-up ~ 2K **region proposals** in order to detect, localize and segment objects.
- Solve the rare of datasets problem by using **transfer learning**; supervised pre-training, followed by **fine tuning**.

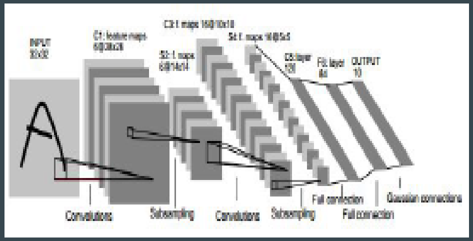
- Apply **SVM** to classify all regions, and **BBR** for localization.
- Improve mean average precision (**mAP**) by achieving a mAP of **66%** on **VOC 2007**, a mAP of **53.3%** on **VOC 2012** and a mAP of **31.4%** on **ILSVRC 2013.**

Convolutional Neural Network



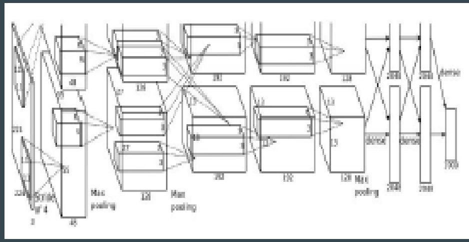
Fukushima 1980

Neocognitron



LeCun et al. 1998

SGD for document recognition



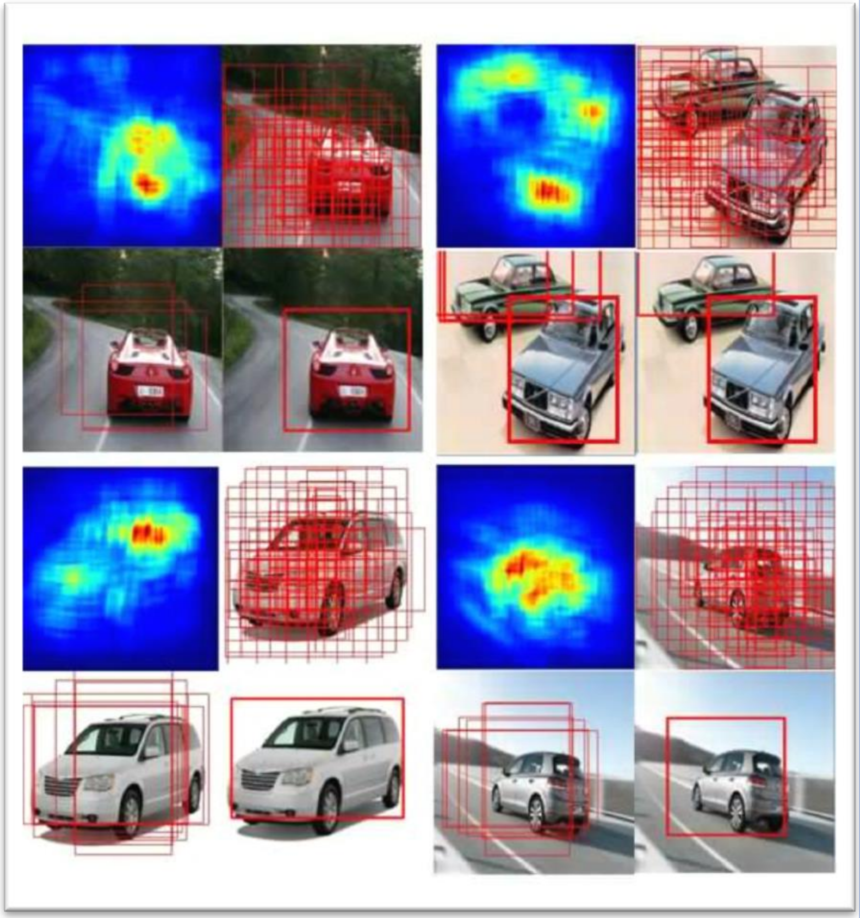
Krizhevsky et al. 2012

ImageNet classification (AlexNet)

"R-CNN" presentation by (Pandian Raju and Jialin Wu).

Region Proposals

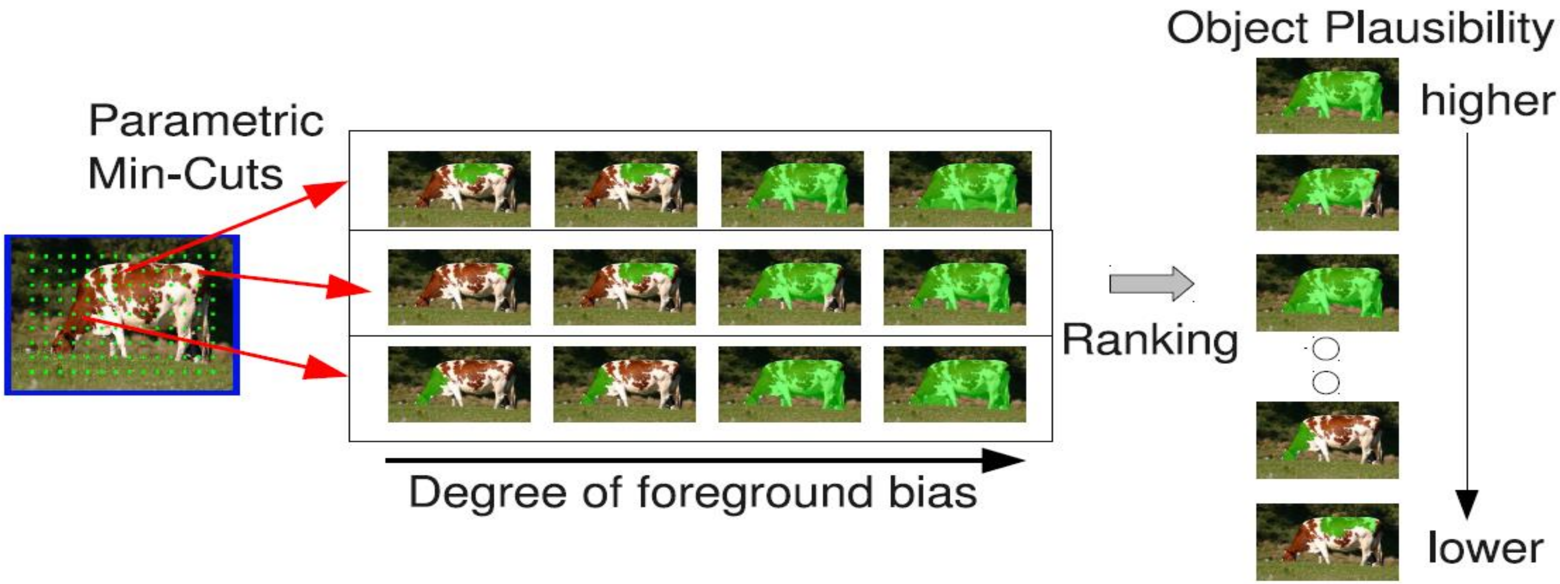
Region proposals methods are dealing with the image as a huge number of **regions**, assuming that any blobby region is containing object.



Ex:

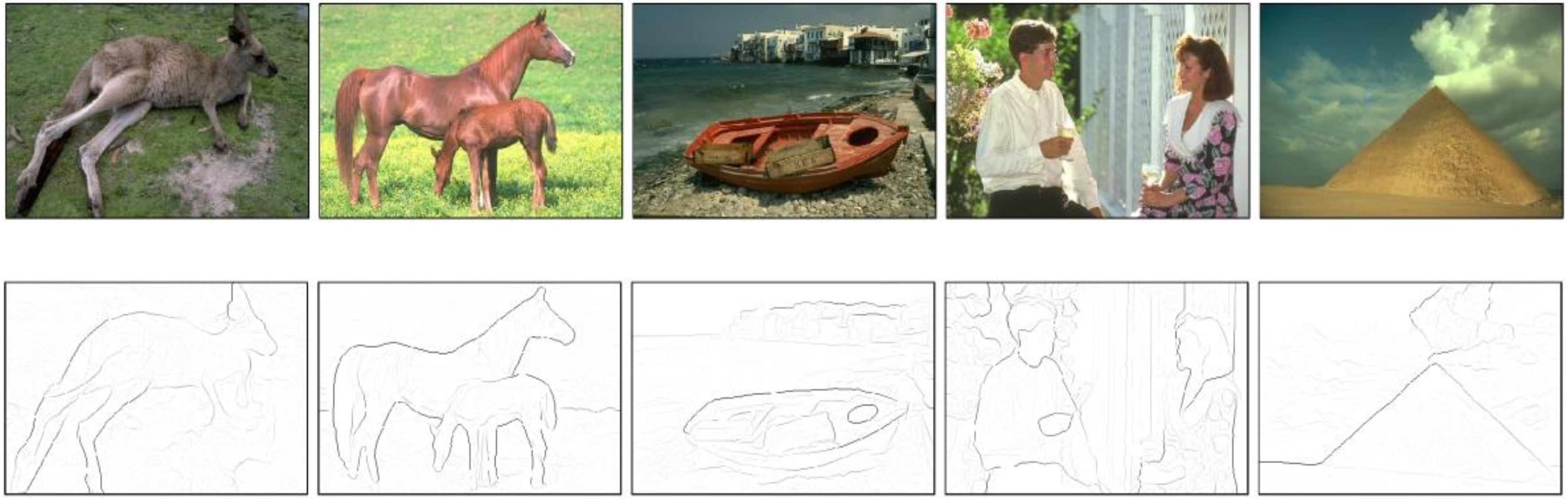
- **Selective search**
- **Edge Boxes.**
- **CPMC.**

CPMC (Constrained Parametric Min-Cut)



- **Foreground (FG)** consist of small square pixels that are regularly placed over the image.
- **Background (BG)** has four different hypothesis:
- 1) covering the **full** image boundary,
- 2) just the **vertical** image boundaries,
- 3) just the **horizontal** image boundaries and
- 4) all image boundaries but the **bottom** one.

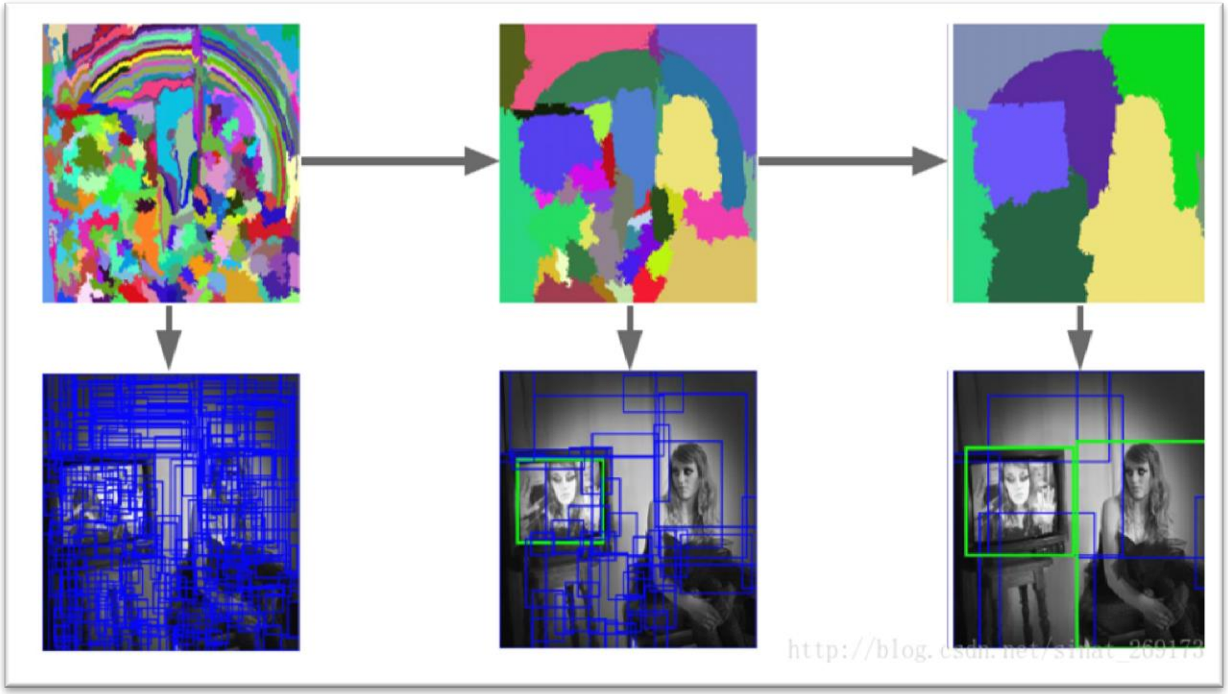
CPMC



- Then filter and rank the regions according to the most acceptable object hypotheses.
- Using **edge detection** techniques to get the most acceptable object hypotheses.
- Ranking involves first removing duplicates, then diversifying the segment overlap scores.

Region Proposals (Selective Search)

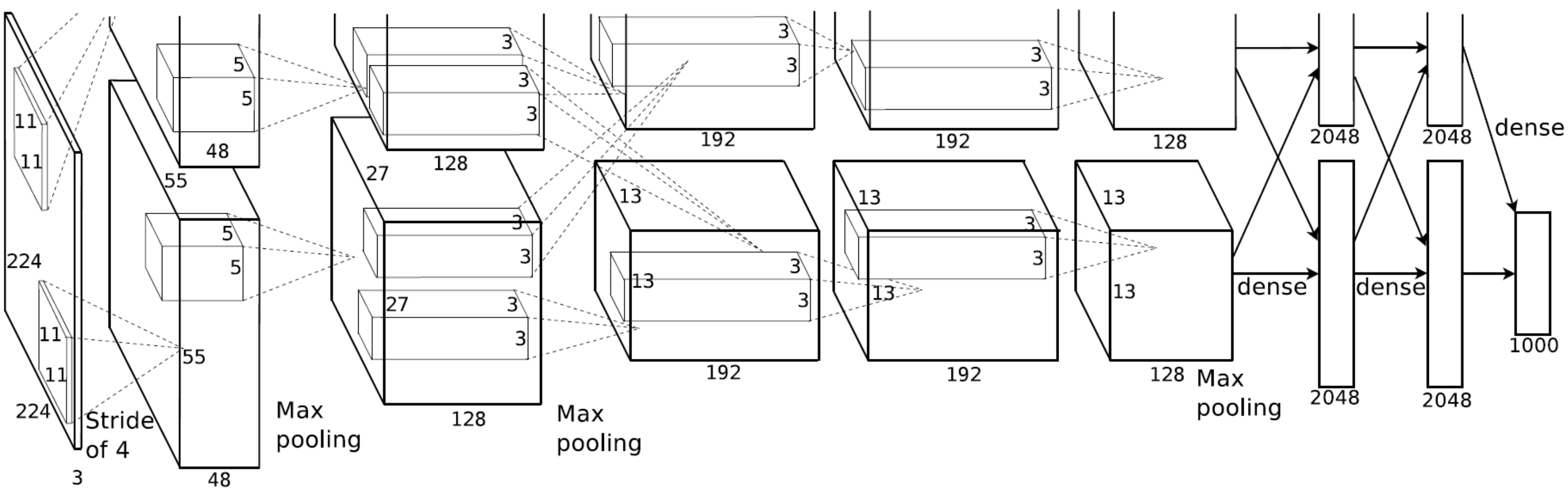
- start from every pixel, search **similarity** around it like (same color, same texture, same histogram, ..).
- Generate these **regions** in multiple blobby scales.
- Then, convert these regions to **boxes**.



Selective Search UVA => (Universiteit van Amsterdam)

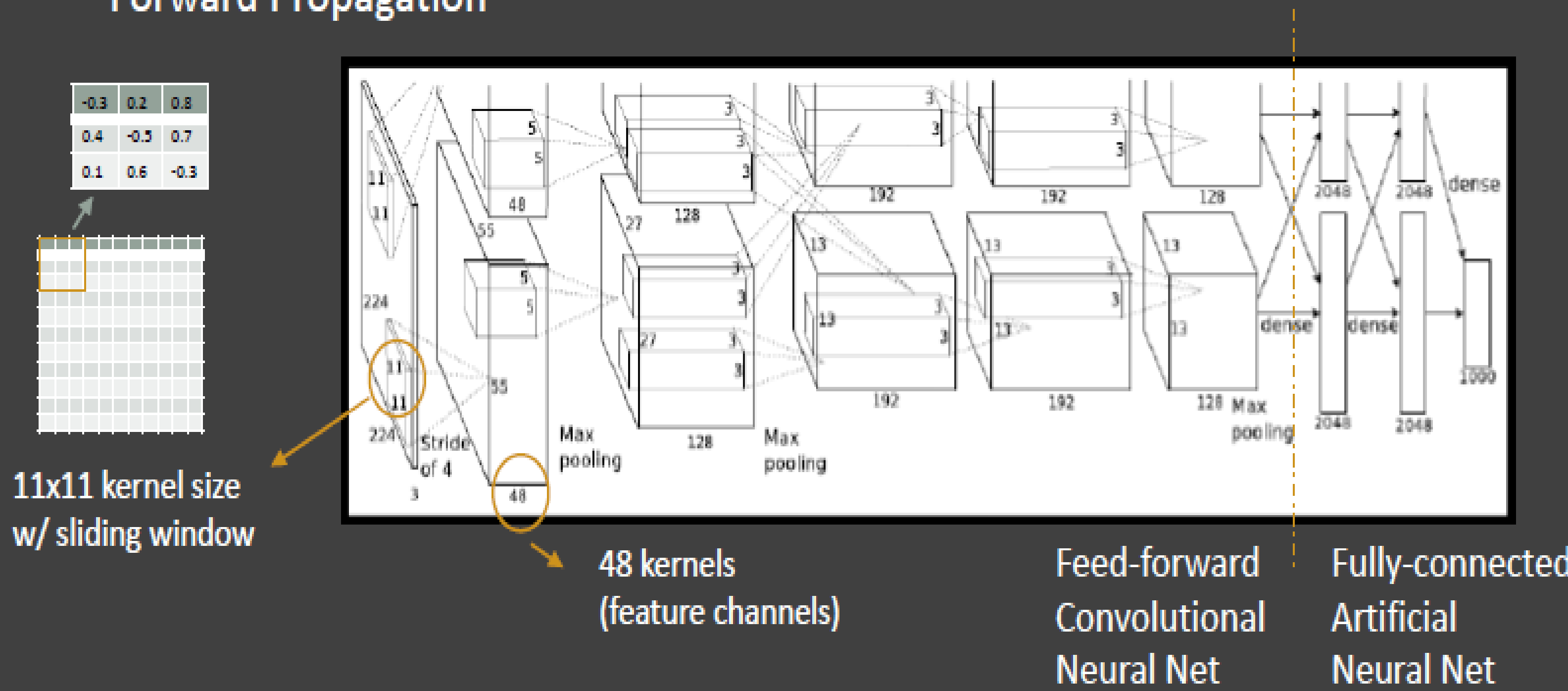
CONVOLUTIONAL NEURAL NETWORKS (CNNS)

- 5 Pool CNN layers.
- 2 Full connected layers.
- Krizhevsky et al., “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012.
- T-Net (Toronto) - AlexNet



- The **first** conv. layer **filters** the 224×224×3 input image with **96 kernels** of size 11×11×3 with a stride of 4 pixels (this is the distance between the receptive field centers of neighboring neurons in a kernel map).
- The **second** conv. layer takes as input the (response-normalized and pooled) output of the first conv. layer and **filters** it with **256 kernels** of size 5 × 5 × 48.
- The **third** conv. layer has **384 kernels** of size 3 × 3 × 256 connected to the (normalized, pooled) outputs of the second conv. layer.
- The **fourth** conv. layer has **384 kernels** of size 3 × 3 × 192.
- The **fifth** conv. layer has **256 kernels** of size 3 × 3 × 192.
- The **fully-connected layers** have **4096** neurons each.

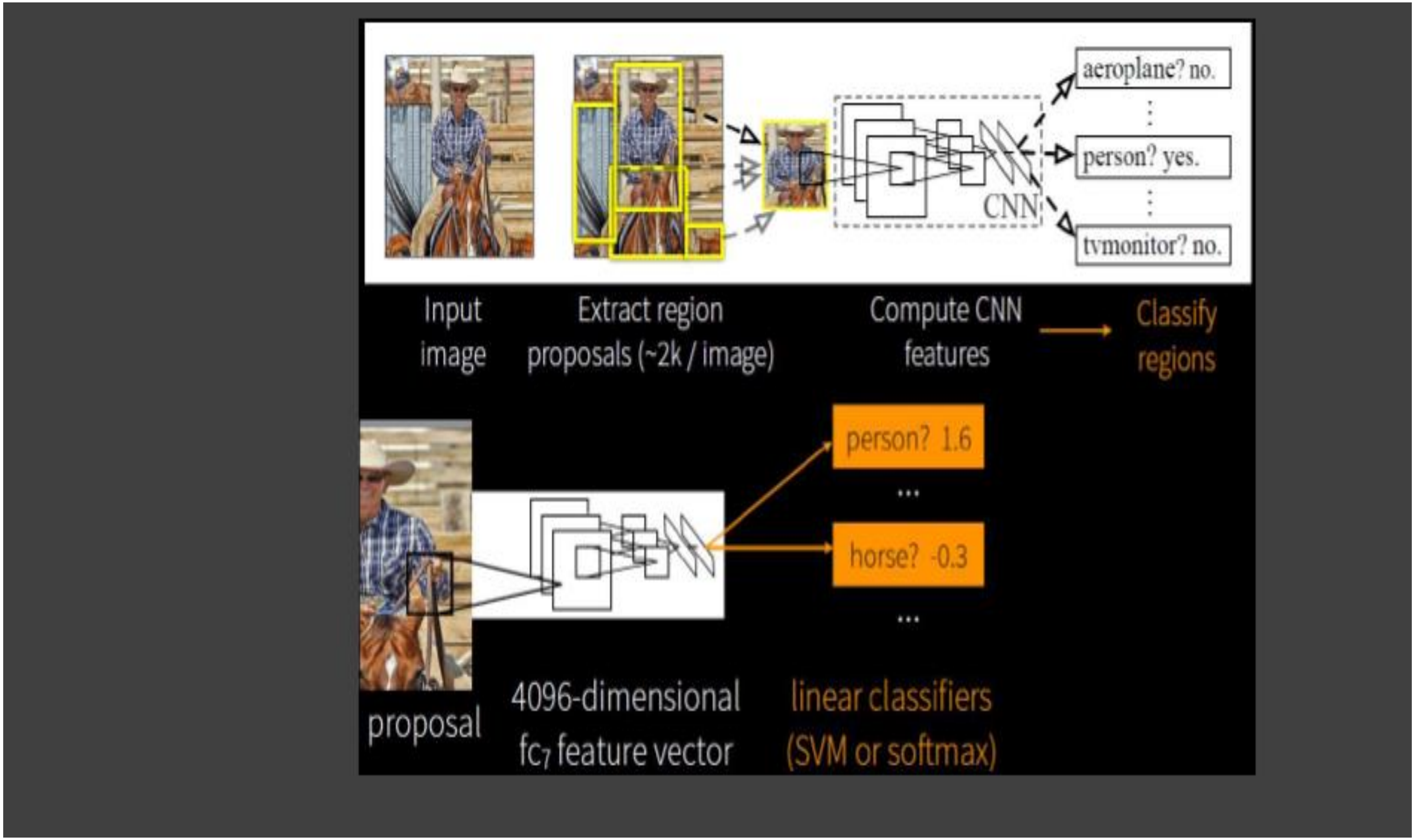
Forward Propagation



At each stage

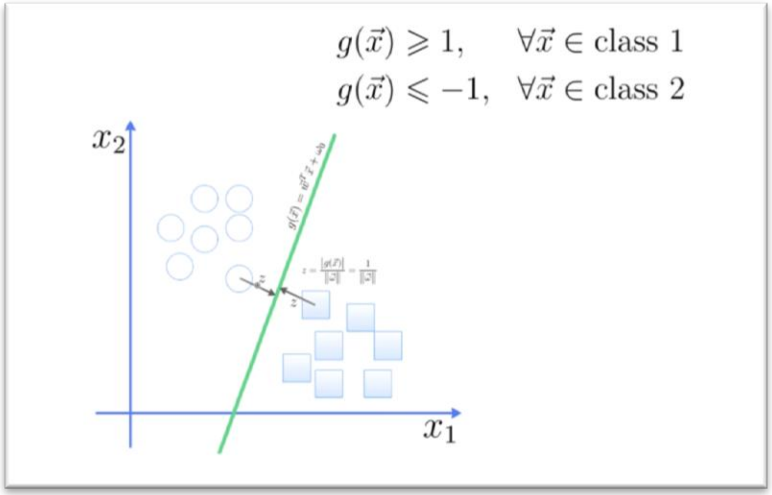
- Higher level features, from convolution alone!
- Max pooling keeps best features
- Convolution kernels learned from training

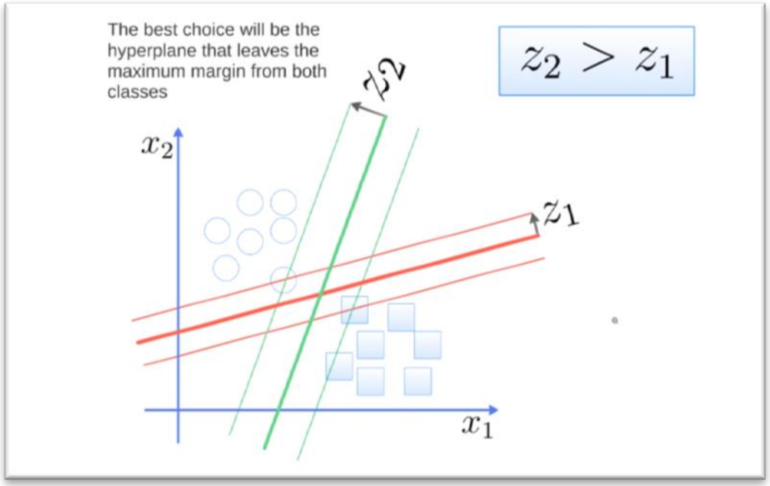
"R-CNN" presentation by (COLLIN MCCARTHY).



"R-CNN" presentation by (COLLIN MCCARTHY).

SUPPORT VECTOR MACHINE (SVM)

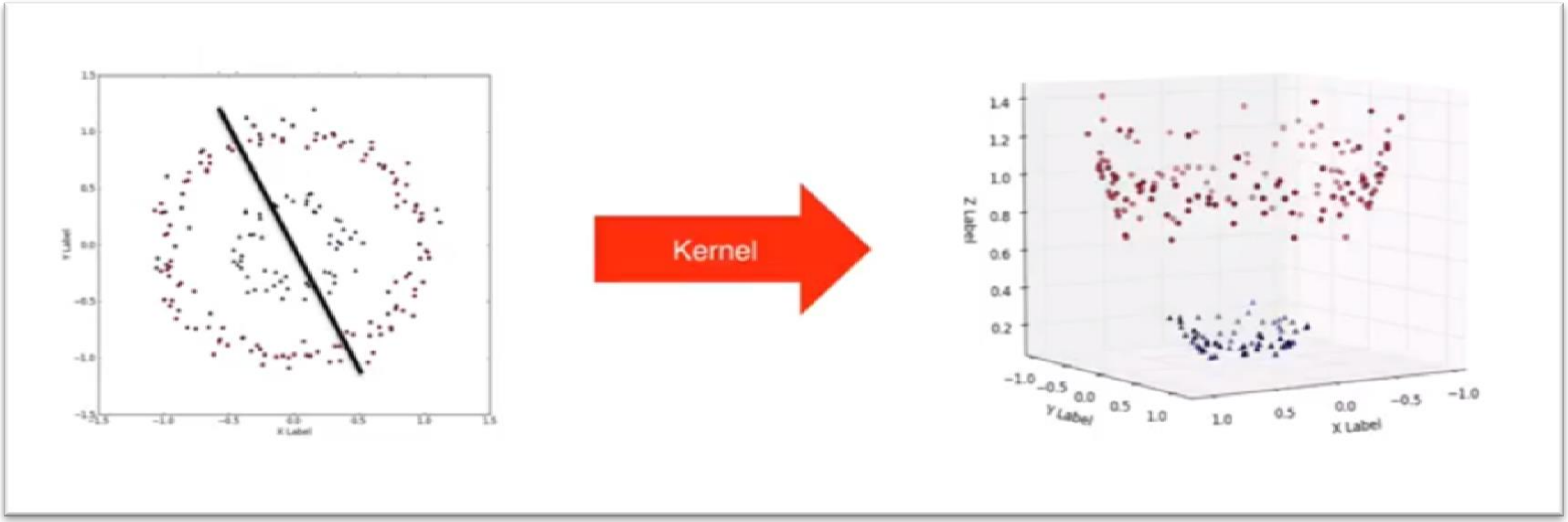




- SVM apply **hyperplanes**, check the **margin** between each plane with all classes, then choose the **best** hyperplane that **leaves the maximum margin** from all classes.

SVM

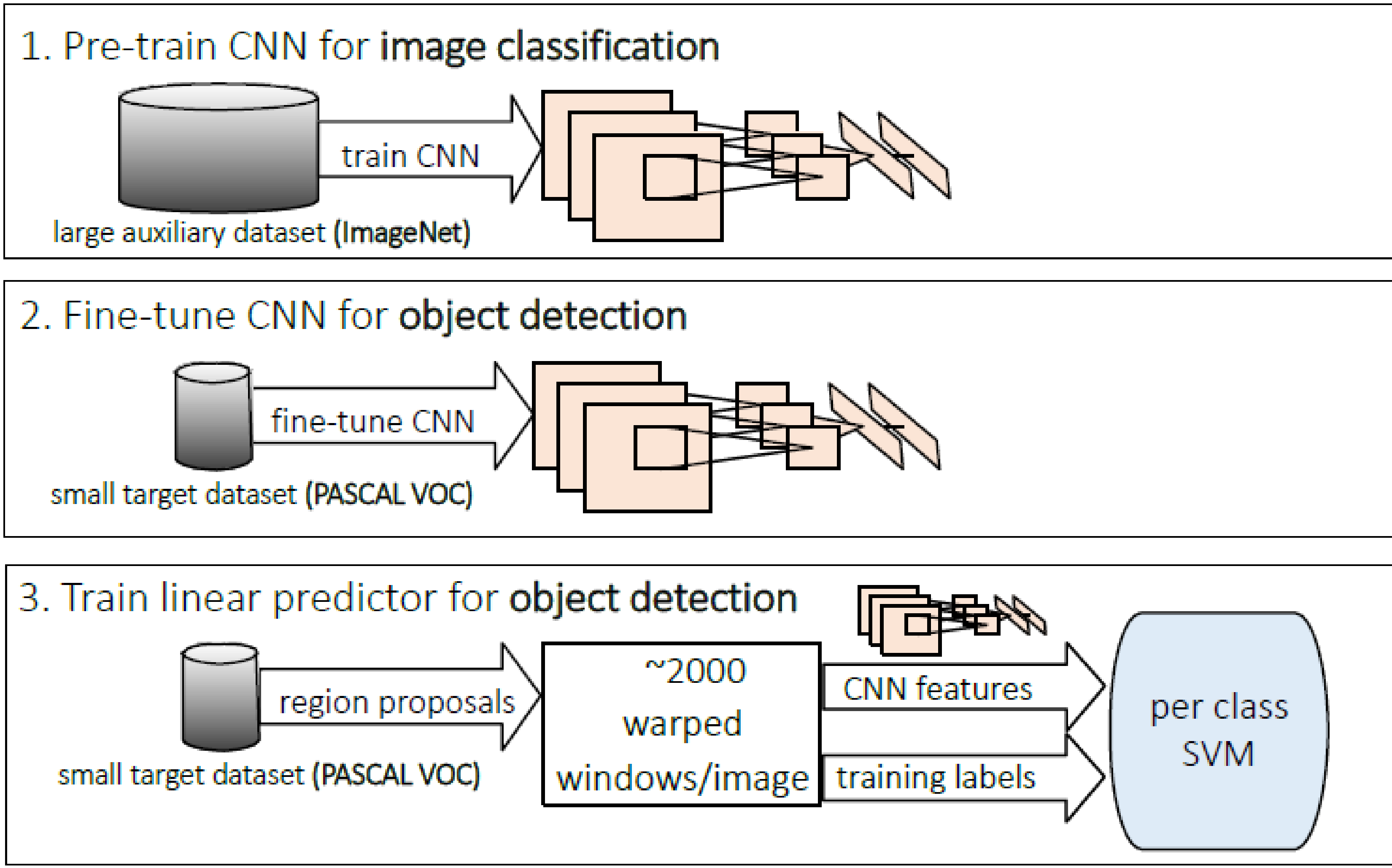
- To solve non-linearity the **kernel** functions transform the data into a **higher dimensional** feature space to make it possible to perform the linear separation.

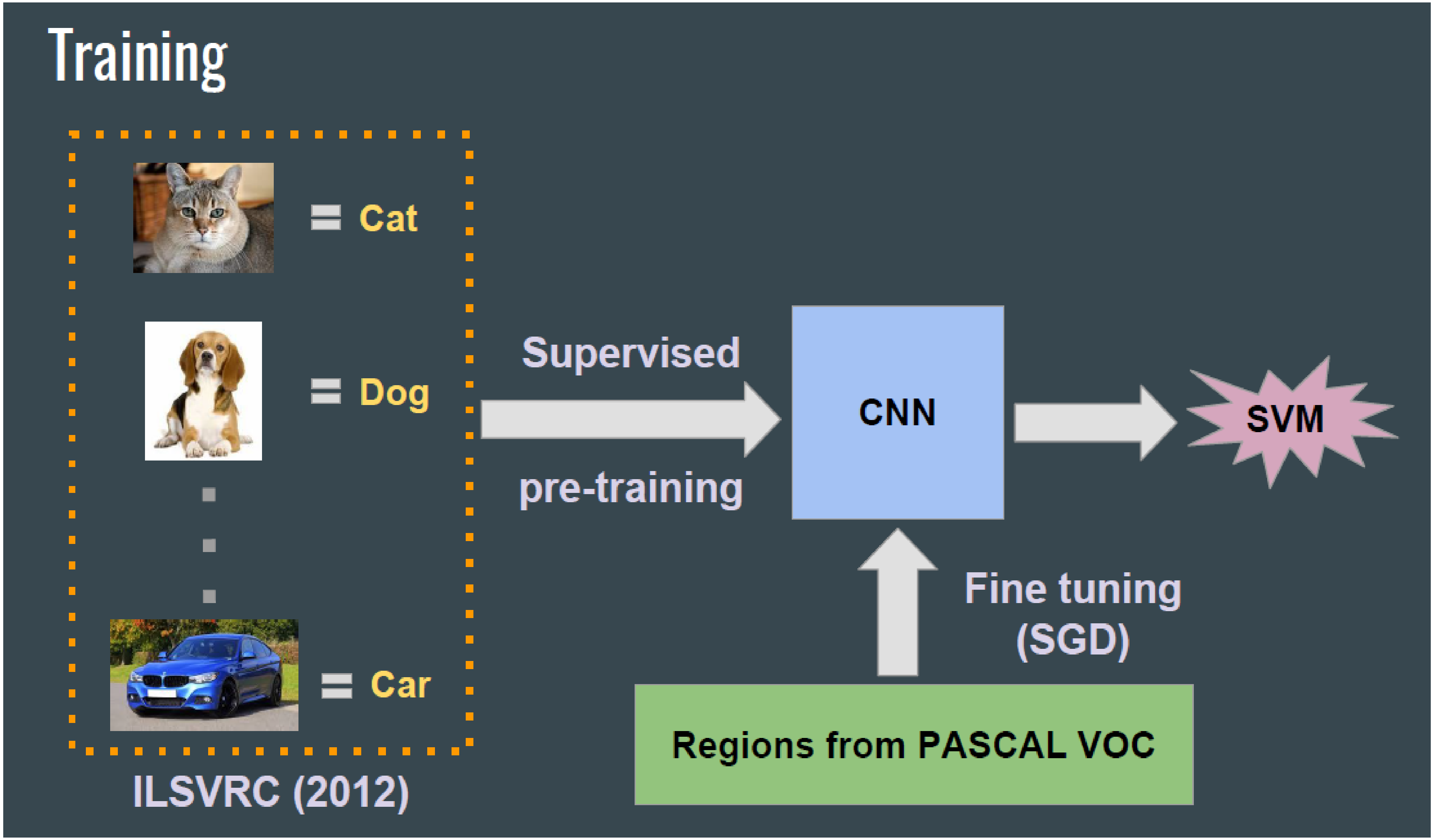


Greedy Non-maximum Suppression

- Greedy search for the next highest score and go to it, and never get back to lower results.
- Greedy non-maximum suppression is used for each class, to reject a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold.

R-CNN: Training



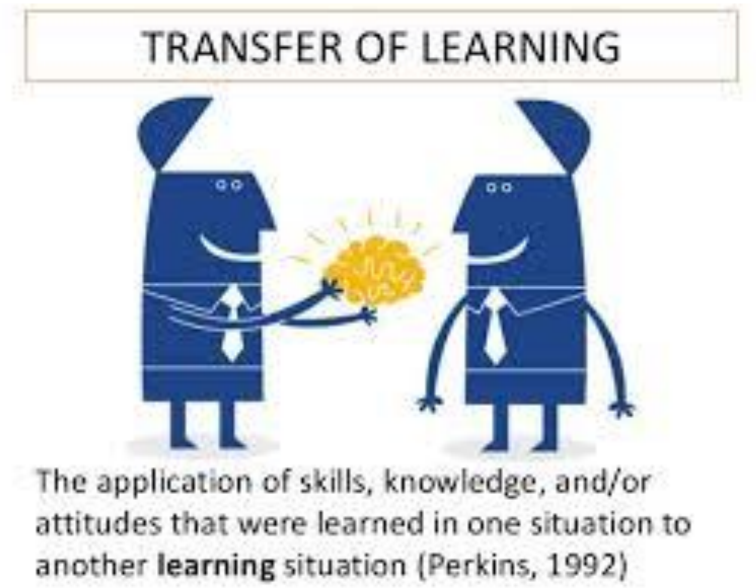


"R-CNN" presentation by (Pandian Raju and Jialin Wu).

Transfer Learning (Fine Tuning)

- Instead of building the Model from scratch, use a pre-trained model as a starting point.

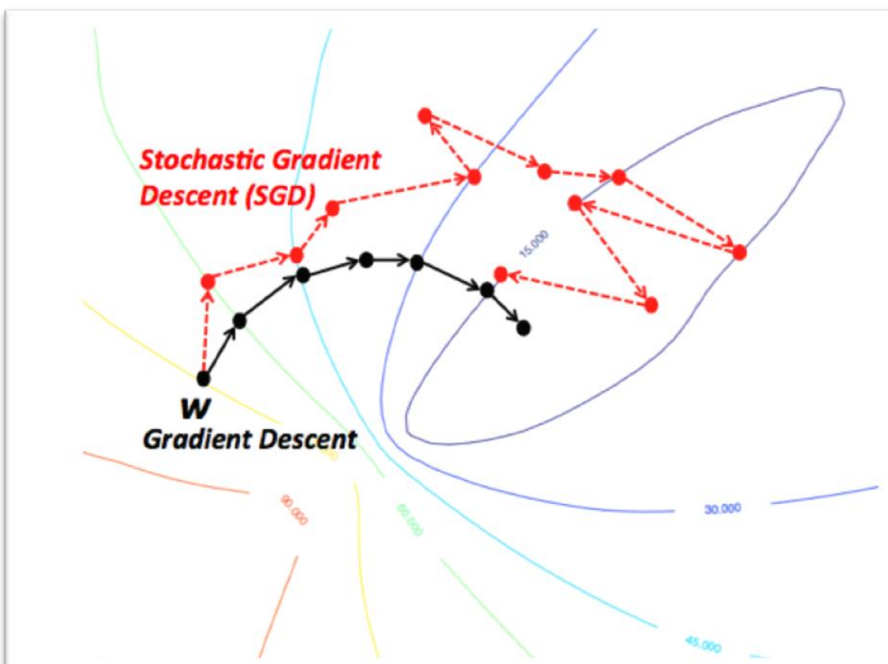
- Then do, **Fine Tuning ;** By train the pre-trained model with your algorithm.

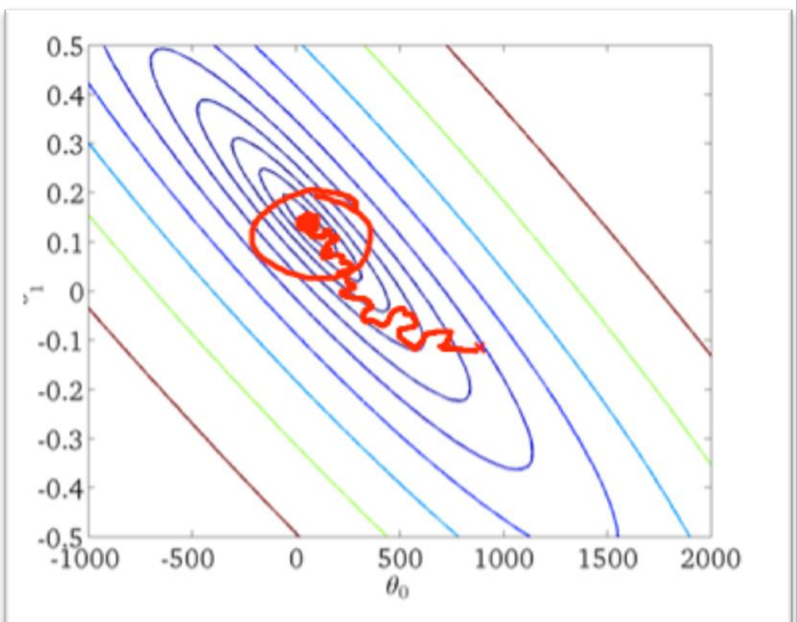


SGD (Stochastic gradient descent)

- Pros:
  - Fast to find the lowest area
  - Will not stuck in local minima.
- Cons:
  - slower in convergence

- SGD is used to adapt the pre-trained output of 2000 class to the fine tuned new challenge.
- 21 classes for VOC or 201 classes for ILSVRC.





Ground truth bounding box

- The **expected object** surrounded with a **bounding box**, which you will compare your algorithm output to the ground truth would be the **ideal output** you would hope your algorithm can produce.
- It is also the **standard** you are defining, by which you evaluate an algorithm.
- The **closer** your algorithm is to **ground truth** the **better.**

INTERSECTION OVER UNION OVERLAP (IOU)

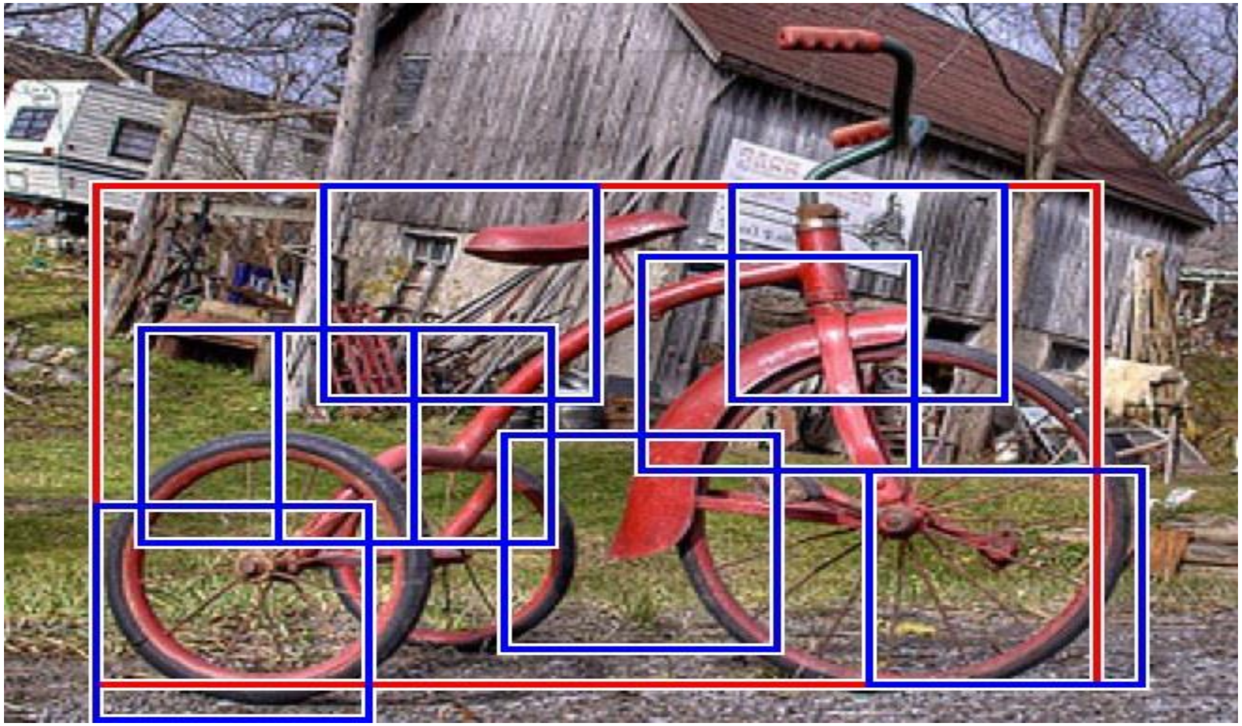
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Hard Negative Mining Method

- Select some random images (windows), check if they are +ve or –ve.
- If they appears truely –ve, we use them to train our data increase the trained examples.

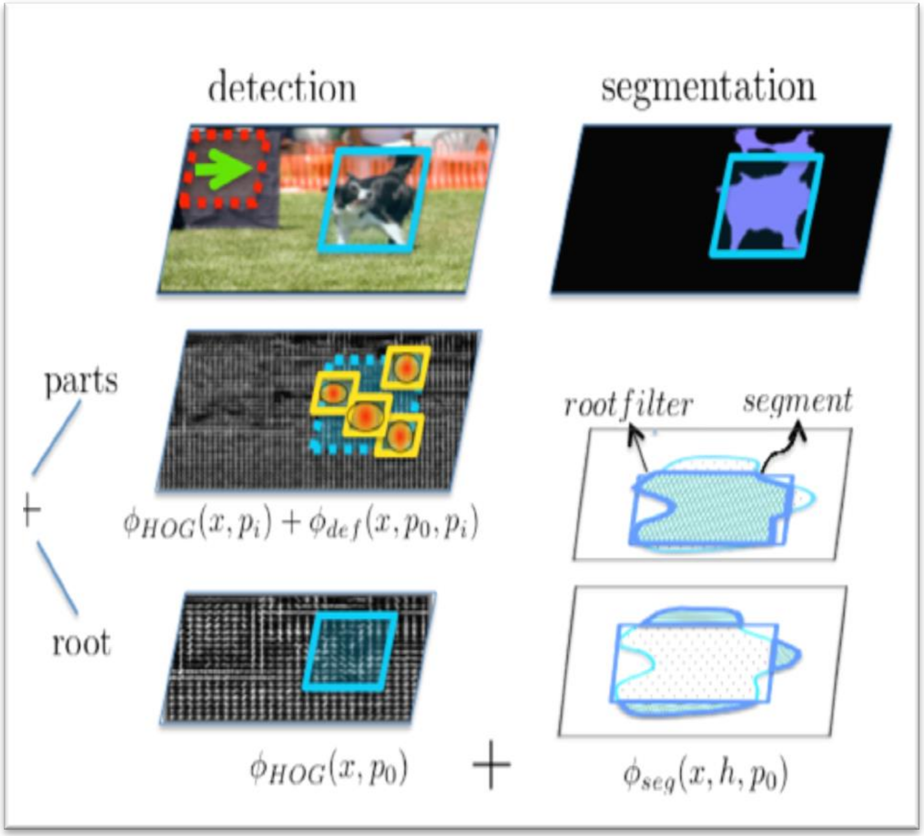
DPM (Deformable Parts Model)

- DPM assumes an **object** is constructed by its **parts**. The detector will first found a match of its whole, and then using its part models to fine-tune the result.



SegDPM (Segmentation Deformable Parts Model)

- use **segmentation algorithms** that compute candidate object **regions**.
- allows every detection hypothesis to select a segment, and scores each box in the image using both the traditional **HOG** filters as well as a set of novel segmentation features.

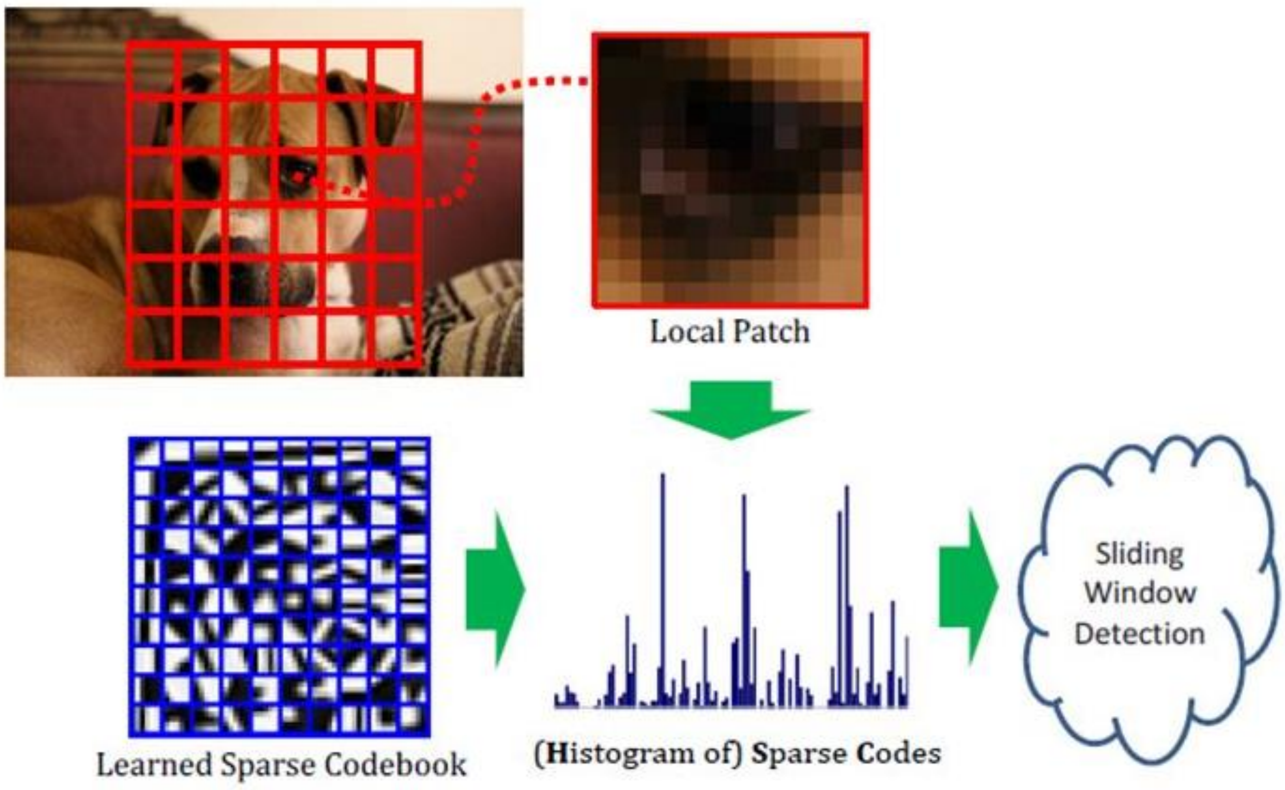


Activation Function

- Function can decide whether that input belongs to a specific class or not.
- Used To decide is that feature describes that class.
- Activation function such as:
  - ReLU: $y = \max(0, x)$
  - Sigmoid: $y = 1 / (1 + \exp(x))$.
  - Tanh: $y = \text{Tanh}^{-1}(x)$.

Histograms of Sparse Codes for Object Detection

- Key idea: Build a HOG-like descriptor on top of K-SVD learned patch dictionary instead of gradients, then DPM



Normalized Average Precision

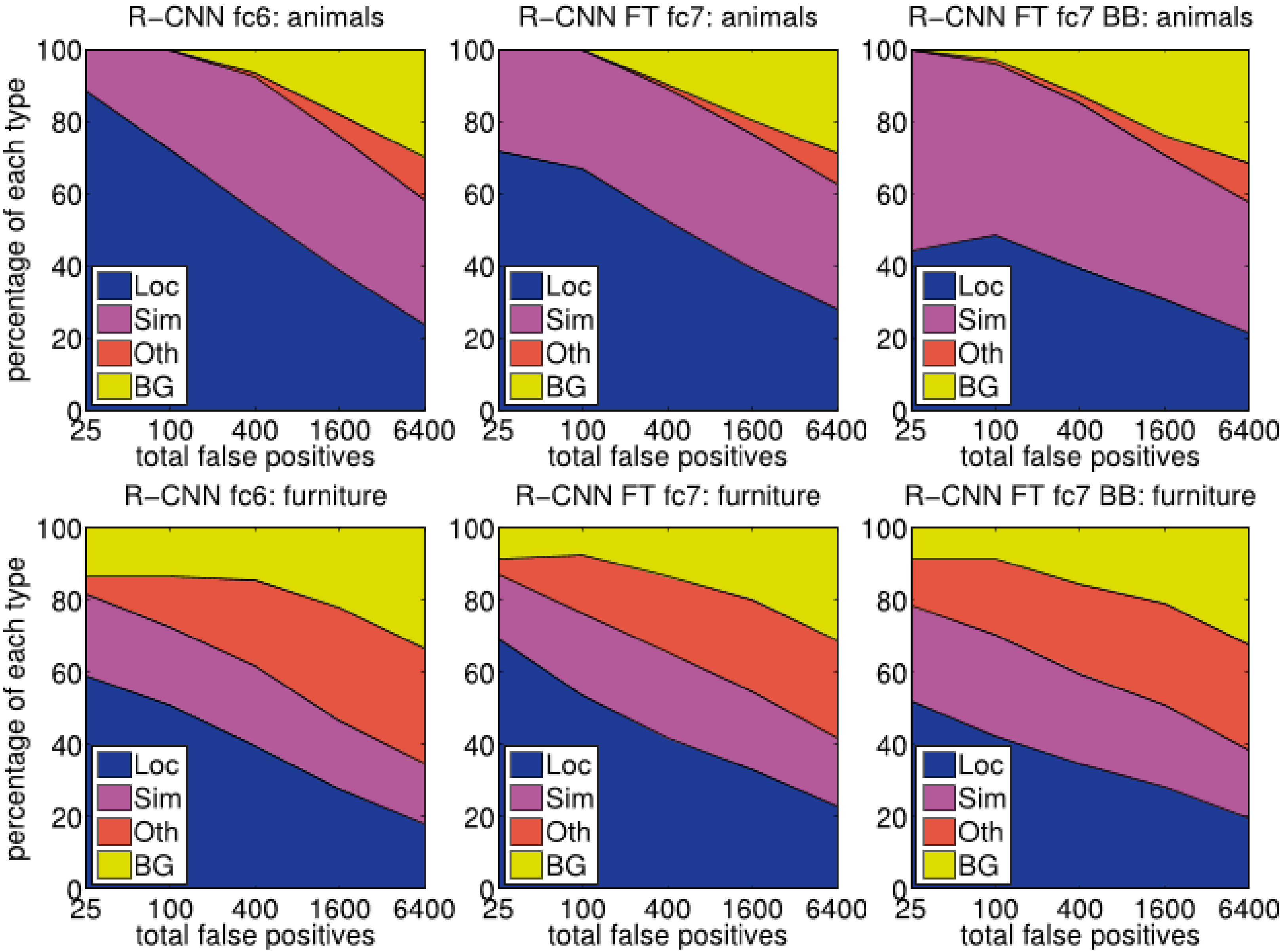
- Average precision is **sensitive** to number of positive examples

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$TruePositive = Recall * Nj$$

Number of object examples in subset j

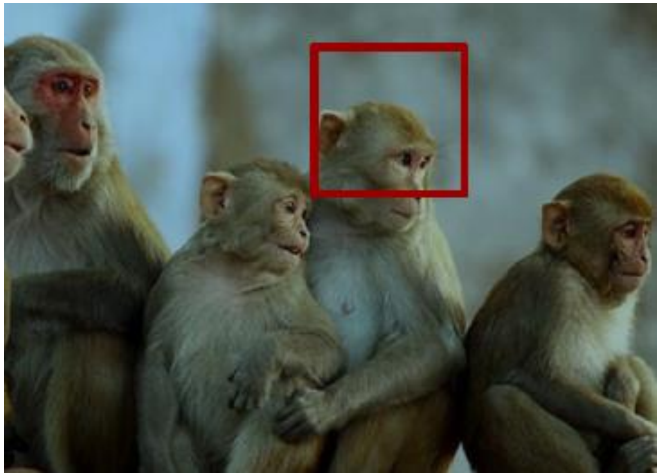
- **Normalized** average precision:
  - replace variable N_j with **fixed** N

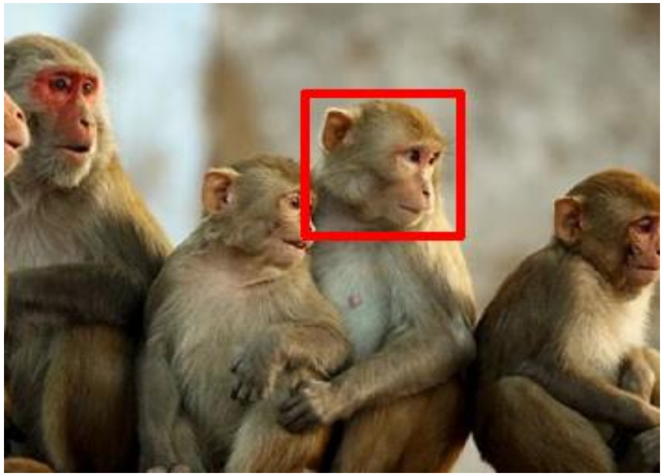


- Loc: poor localization
- Sim: Confusion with similar category
- Oth: Confusion with dissimilar object category
- BG: Confusion with Background

BBR (Bounding box regression)

- **Linear** regression to segment the object, as most errors in segmentation are **mislocalization.**
- By learning a transformation that **maps** a **proposed box P** to a **ground-truth box G**.
- The input set is N training pairs (**P,G**).
- where **P** = (P_x, P_y, P_w, P_h) specifies the pixel coordinates and P's width and height in pixels.
- where **G** = (G_x, G_y, G_w, G_h), G is ground-truth box.





$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

- Functions $d_\star(P)$ is modeled as a linear function of the pool5 features. ($\star$ is one of x, y, w, h)
- $d_\star(P) = W_\star^{T} Ø_5(P)$
  - $W_\star^{T}$ is a vector of learnable model.
  - $Ø_5(P)$ is the pool5 features.
- Learn $W_\star$ by optimizing the regularized squares objective.

Regularization to prevent overfitting.

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \phi_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2 .$$

(λ =1000 is the regularization parameter)

$$t_w = \log(G_w / P_w)$$
$$t_h = \log(G_h / P_h).$$
$$t_x = (G_x - P_x)/P_w$$
$$t_y = (G_y - P_y)/P_h$$

Training Stages

Training	Validation	Testing

They split the validation set to 2 sets. **Why?**

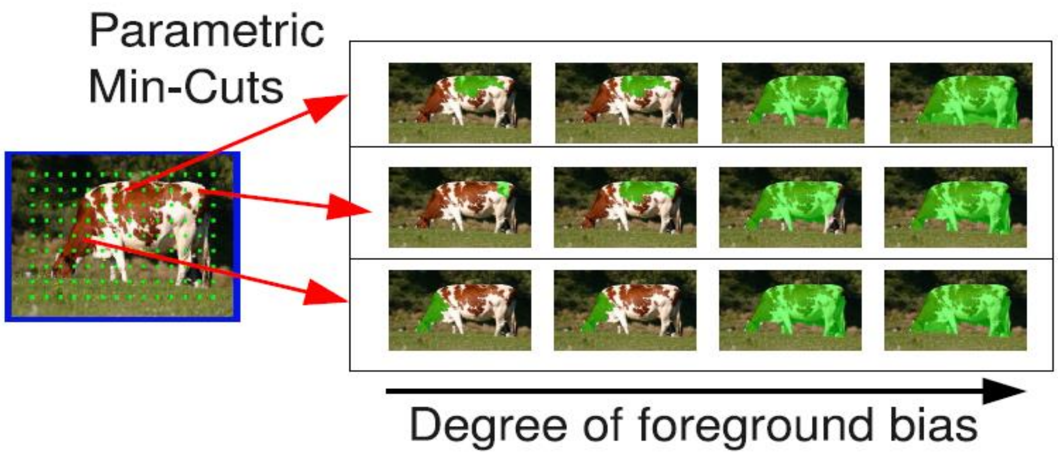
Training	**Val1**	**Val2**	**Testing**

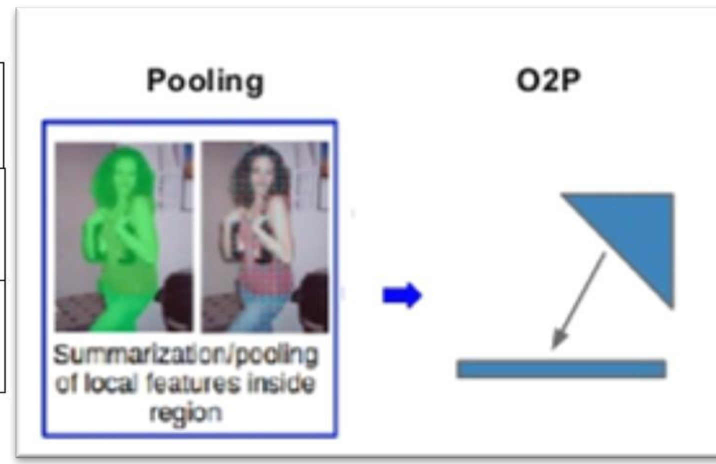
Because **Validation** set and **test** set are **labeled** with a **bounding box** around the objects, but training set images **have not** a bounding box around the objects.

So, Val1 used to train the Bounding Box Regression.

O2P (Second order Pooling)

- Compute the second-order statistics of local descriptors for a region by introduce **average and max pooling** that together solve non-linearity.
- By enriching **local descriptors** with additional information from **CPMC** and **LBP** leads to large performance gains.





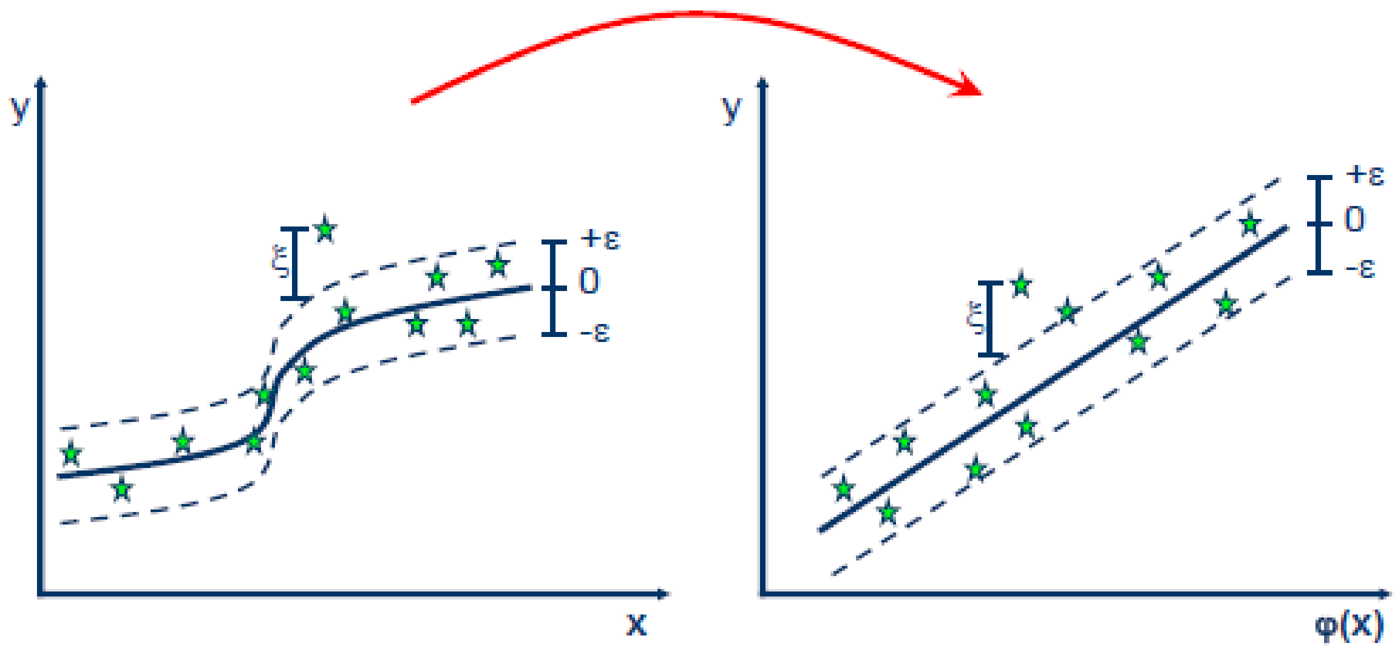
Average Pooling

$$\mathbf{G}_{avg}(R_j) = \frac{1}{|F_{R_j}|} \sum_{i:(\mathbf{f}_i \in R_j)} \mathbf{x}_i \cdot \mathbf{x}_i^\top$$

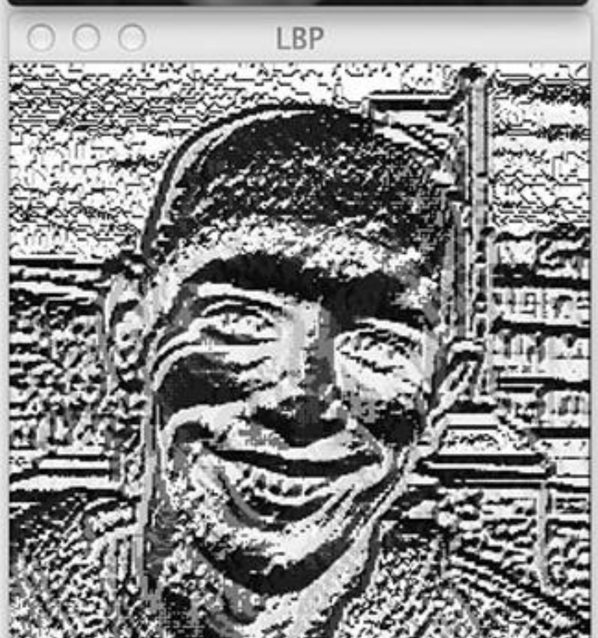
Max Pooling

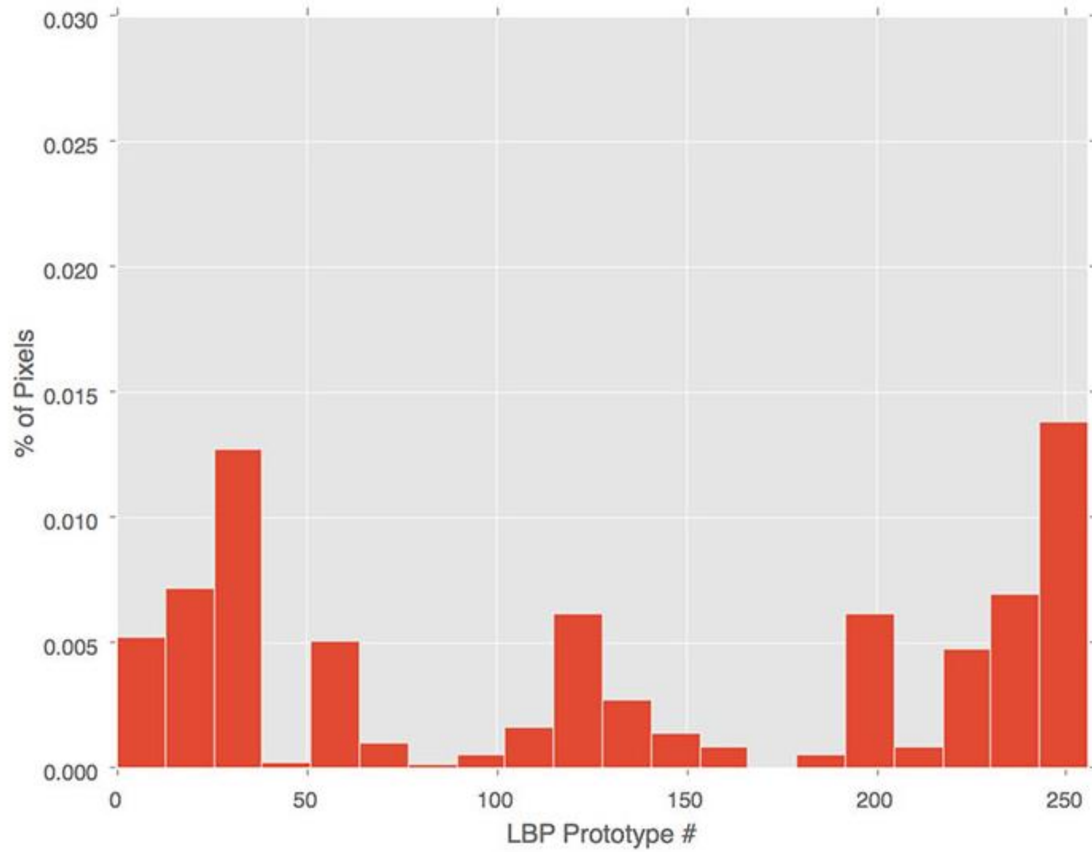
$$\mathbf{G}_{max}(R_j) = \max_{i:(\mathbf{f}_i \in R_j)} \mathbf{x}_i \cdot \mathbf{x}_i^\top$$

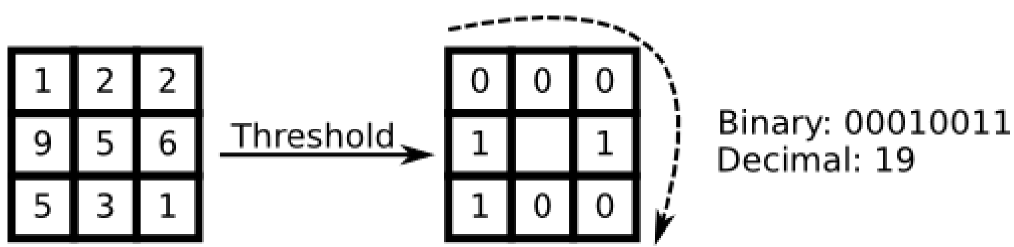
Support Vector Regression (SVR)



LBP (Local Binary Patterns)







Softmax Classifier

Softmax Classifier (Multinomial Logistic Regression)

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

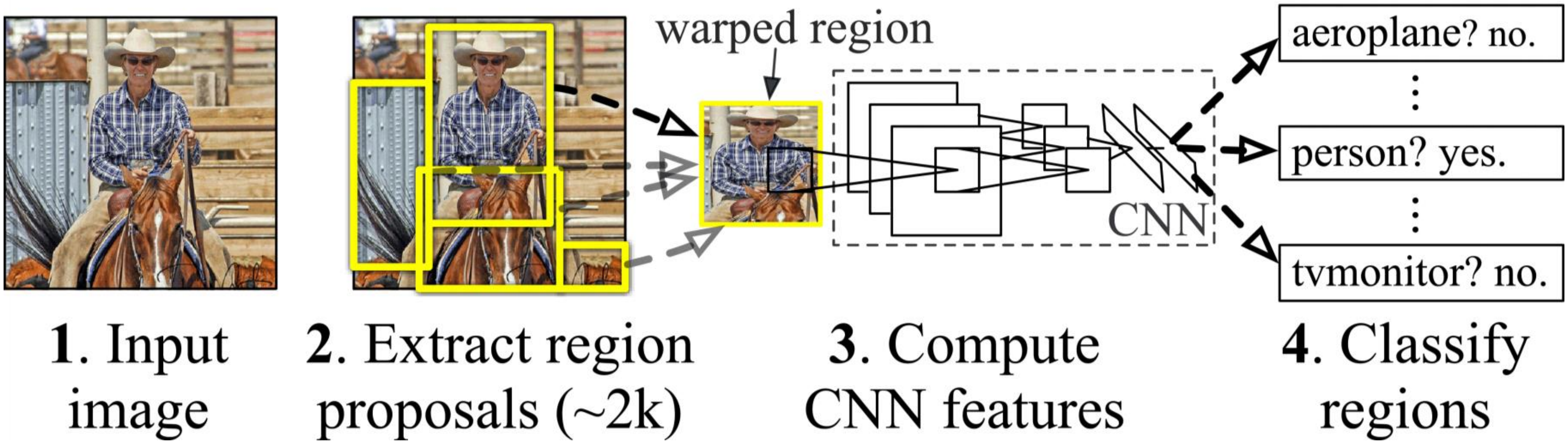
unnormalized probabilities

	unnormalized log probabilities	exp →	unnormalized probabilities	normalize →	probabilities	
cat	**3.2**		**24.5**		**0.13**	→ L_i = -log(0.13) = **0.89**
car	5.1		164.0		0.87	
frog	-1.7		0.18		0.00	

unnormalized log probabilities

probabilities

Further Work



R-CNN Problems

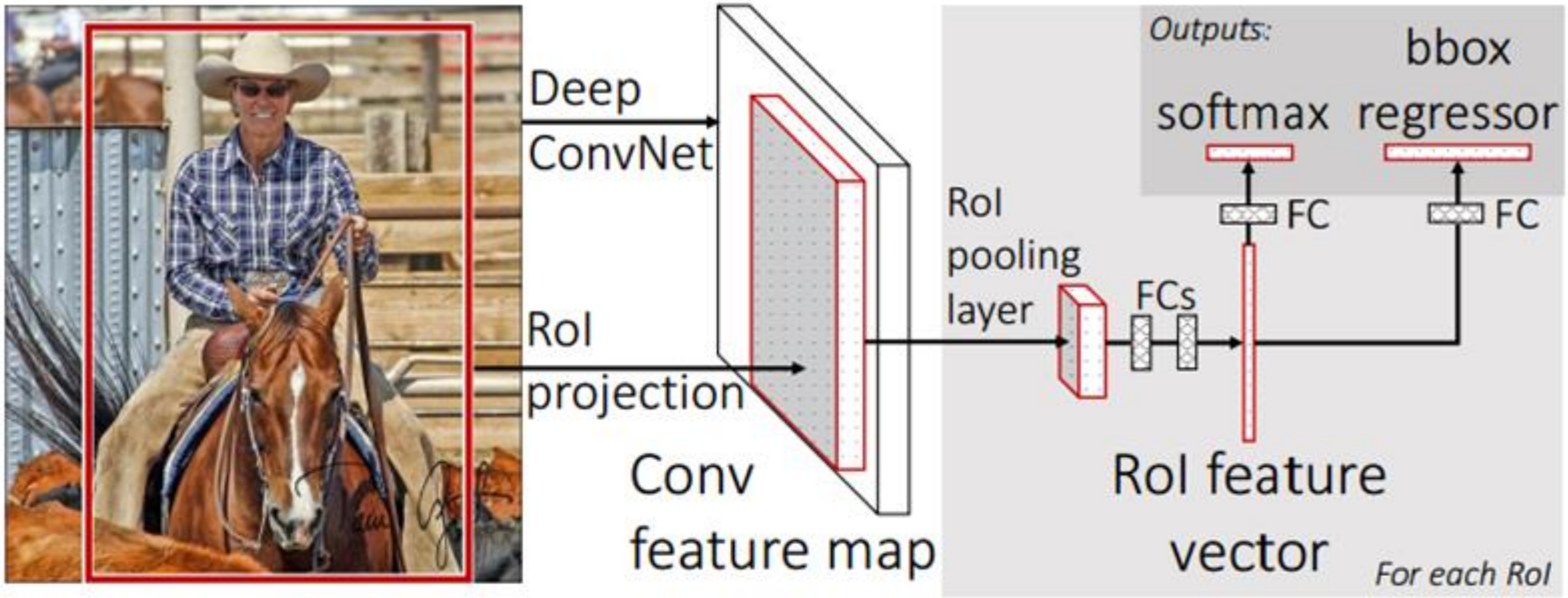
- Slow at test time.
- SVM and BBR are Post-Hoc; can't update the features in runtime.
- Complex multistage in the training pipeline and need a huge memory.

Solution

Fast R-CNN
Faster R-CNN
Mask R-CNN

Fast R-CNN

- **Swap** the order of extracting the **region proposals** and running the **CNN** first.
- Run the region proposals on region on interests (**ROI**) only.
- **ROI** can make a back-propagation for the regions.
- Use **Softmax** as a classifier.

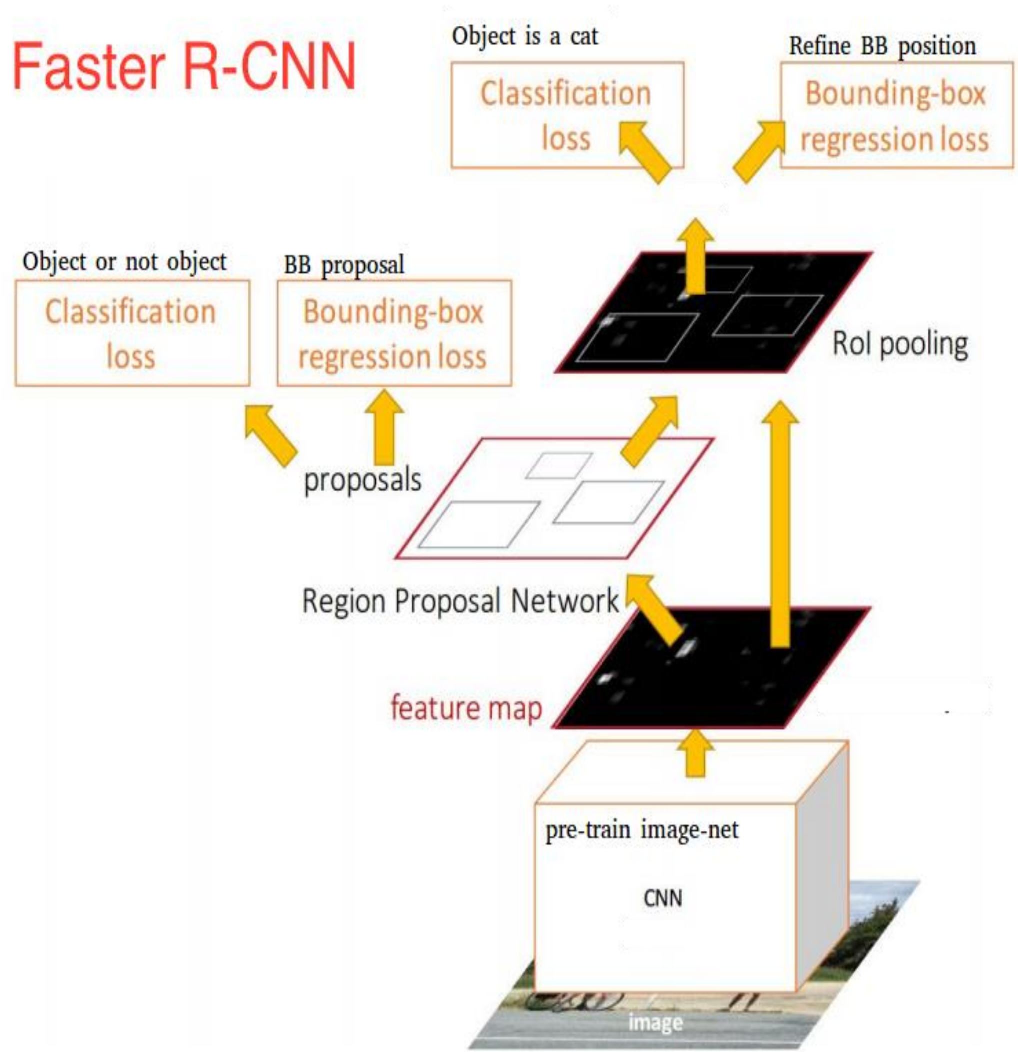


Faster R-CNN

Instead of region proposal selective search method, they use region proposal network (**RPN**).

Also, use **CNN** as a classifier and regression instead of **SVM** and **BBR**.

Faster R-CNN run **backward** from the **feature map** to the **image**.

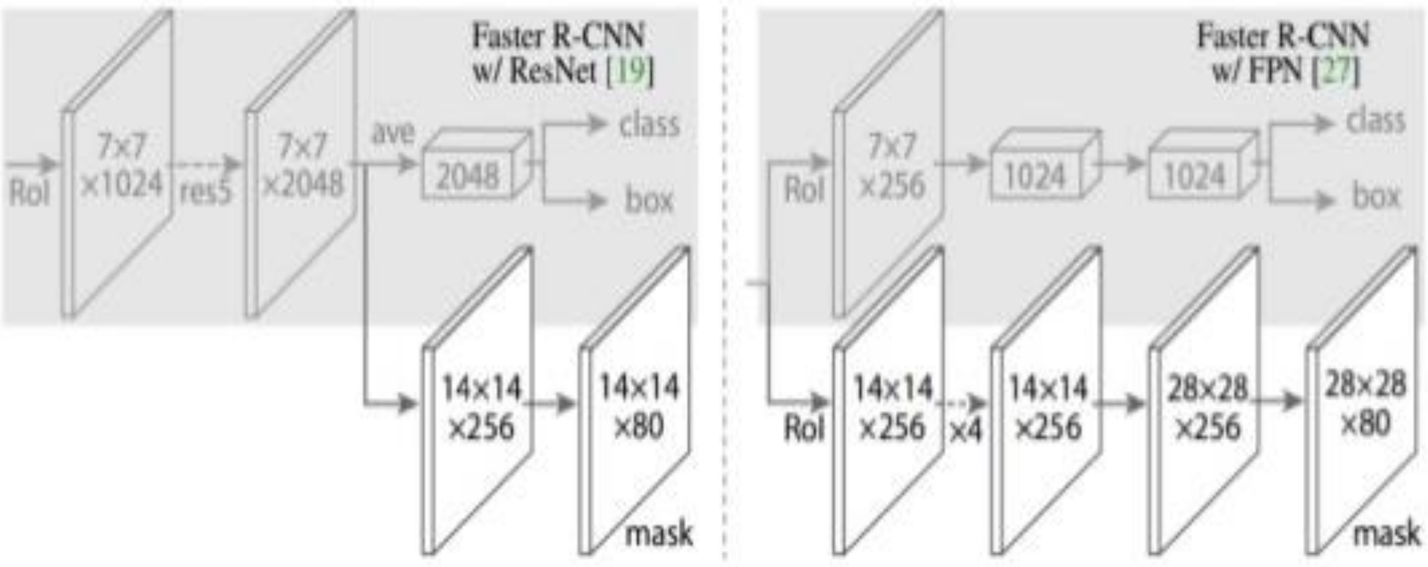


RPN (Region Proposal Network)

- Slide a small anchor (window) on the feature map.
- Build a network for:
  - Classifying (object - not object).
  - Regressing bounding box locations.
  - Use N anchor boxes at each location.
  - The anchor will project the feature map to find the corresponding point in the original image.

Mask R-CNN

- Mask R-CNN extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression



EVALUATION

	R-CNN	Fast R-CNN	Faster R-CNN	Mask R-CNN
Test time per image	50 Sec.	2 Sec.	0.2 Sec.	0.19 Sec.
Training time	84 Hrs.	9.5 Hrs.	-	32 Hrs.
Speed-Up	1x	25x	250x	-
mAP (VOC 2007)	66.0%	66.9%	66.9% 73.2% *2012	-

Papers' Discussion

References

- GitHub repository for R-CNN paper.
  - https://github.com/rbgirshick/rcnn
- Mask R-CNN vs Faster R-CNN vs Fast R-CNN vs R-CNN
  - https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4
- CNN_Course (CS 231n Stanford Uni.).
  - http://cs231n.stanford.edu/
- Ric Poirson presentation on CNN_Course (CS 231n Stanford Uni.).
  - http://slideplayer.com/slide/10395667/35/
- PASCAL VOC Evaluating Server
  - http://host.robots.ox.ac.uk:8080/
- ILSVRC Competition Server.
  - https://www.kaggle.com/c/imagenet-object-localization-challenge
- SVR: Support Vector Regression
  - http://www.saedsayad.com/support_vector_machine_reg.htm
- CPMC: constrained parametric min-cut
  - http://bitsearch.blogspot.co.at/2014/01/object-candidates-with-constrained.html
- Selective Search
  - https://www.learnopencv.com/selective-search-for-object-detection-cpp-python/

- Transfer learning and fine tuning
  - https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/
- Stochastic Gradient Descent
  - http://curtis.ml.cmu.edu/w/courses/index.php/Stochastic_Gradient_Descent
- “Diagnosing error in object detection” presentation by (Yuduo Wu).
- “R-CNN” presentation by (Pandian Raju and Jialin Wu).
- “R-CNN” presentation by (COLLIN MCCARTHY).
- [2] Semantic segmentation using regions and parts. In CVPR, 2012.
- [4] Semantic segmentation with second-order pooling. In ECCV, 2012.
- [5] CPMC: Automatic object segmentation using constrained parametric min-cuts. TPAMI, 2012.
- [23] Diagnosing error in object detectors. In ECCV. 2012.
- 25] classification with deep convolutional neural networks. In NIPS,2012.
- [30] Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV, 2001.
- [39] Selective search for object recognition. IJCV, 2013.
- Fine-tuning Deep Neural Networks in Continuous Learning Scenarios,ACCV,2016.
- Machine Learning Course by Andrew NG, Stanford Uni.
  - https://www.coursera.org/learn/machine-learning

Thank You !