Image Search with Deep Learning

Sung-Eui Yoon
(윤성의)

Tentative Schedule

- **10/25: mid-term exam**

10/27: Student pre. I
11/1
11/3
11/8
11/10: mid-pre
11/15: mid-pre
11/17: no class
11/22: Student pre. II
11/24
11/29
12/1
12/6
12/8: reserved
12/13: final pre.
12/15: final pre.
D: 10/4 (student schedule

Deadlines

- **Declare project team members**
  - **By 10/3 at Noah**
- **Confirm schedules of paper talks and project talks at 10/4**
- **Declare two papers for student presentations**
  - **by 10/17 at Noah**
  - **Discuss them at the class of 10/18**

Deep Learning for Image Search

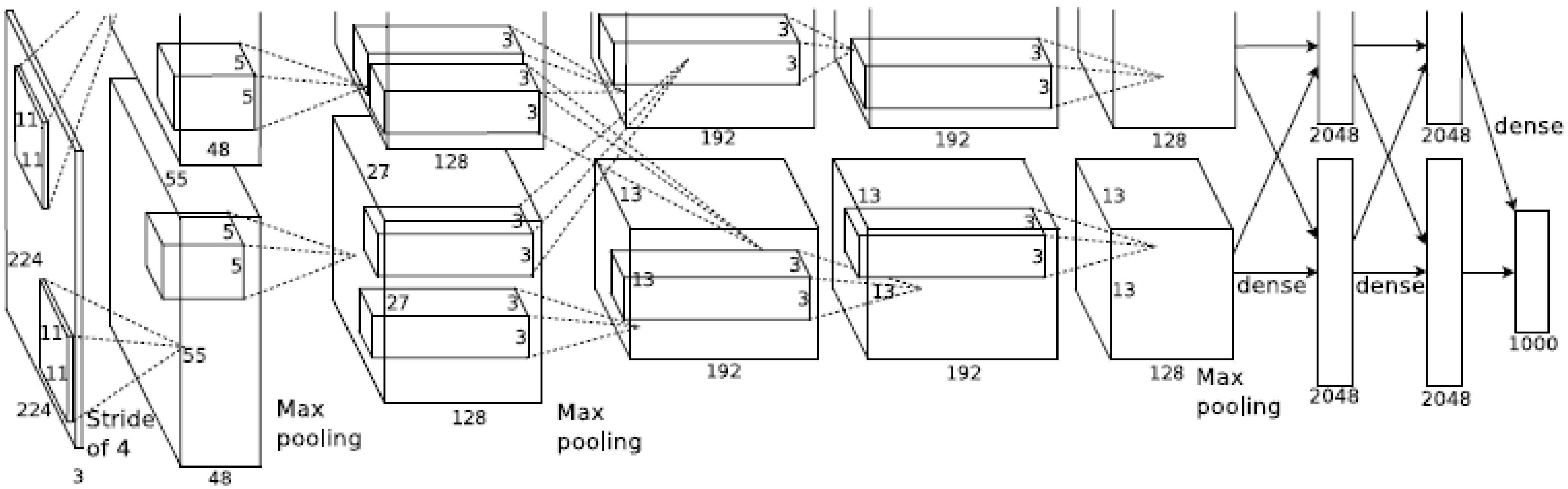
- **Not well studied yet**
- **Designed by integrating some machine learning and computer vision techniques within deep learning**

Outline

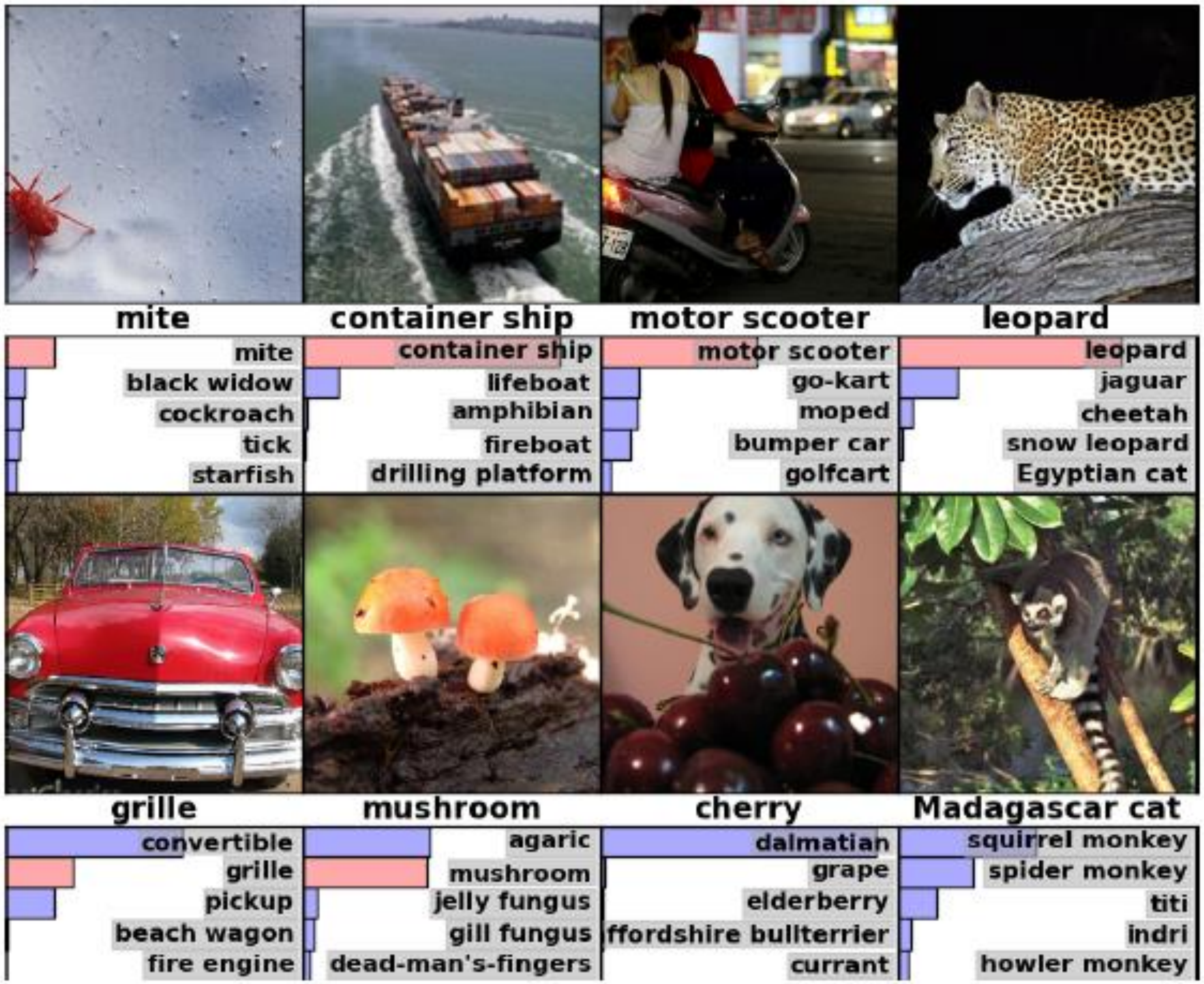
- Representations
  - Fine-tuning
  - Dimension reduction
- Localization and detections
- Not that much on:
  - Post-processing
  - Matching

ImageNet Classification with Deep Convolutional Neural Networks [NIPS 12]

- **Rekindled interest on CNNs**
  - **Use a large training images of 1.2 M labelled images**
  - **Use GPU w/ rectifying non-linearities and dropout regularization**

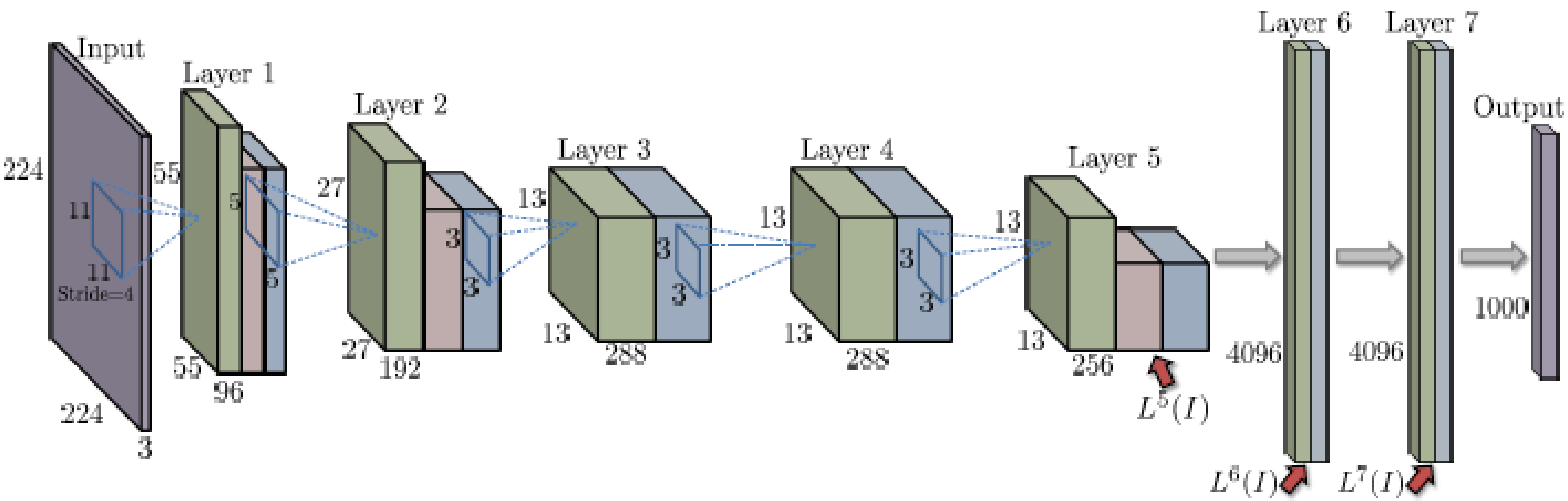


Tested on ILSVRC-2010



Neural Codes for Image Retrieval [ECCV 14]

- **Uses top layers of CNNs as high-level global descriptors (Neural Codes) for image search**
- **Shows higher accuracy with re-training**



Sum Pooling and Centering Priors

- **Inspired by many prior aggregated features (e.g., BoW)**
  - **Use convolution layers as local features as dense SIFTs**
- **Aggregation**
  - **Simply sums those local features or**
  - **Considers centering priors w/ varying weights**

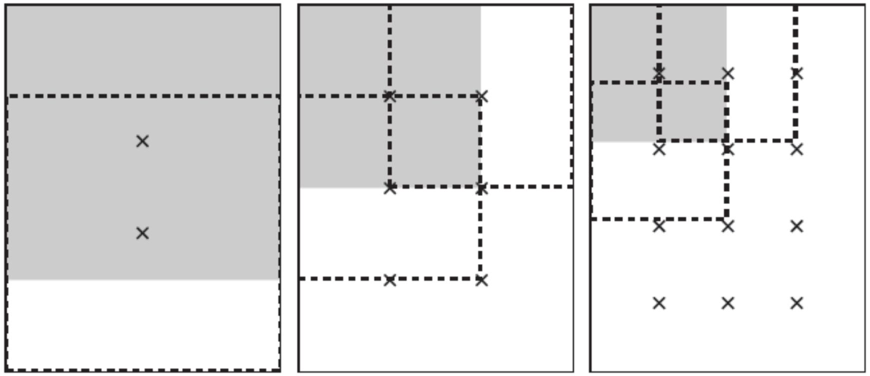
$$\psi_1(I) = \sum_{y=1}^{H} \sum_{x=1}^{W} f_{(x,y)}$$

Method	Holidays	Oxford5K (full)	Oxford105K (full)	UKB
Fisher vector, k=16	0.704	0.490	—	—
Fisher vector, k=256	0.672	0.466	—	—
Triangulation embedding, k=1	0.775	0.539	—	—
Triangulation embedding, k=16	0.732	0.486	—	—
Max pooling	0.711	0.524	0.522	3.57
Sum pooling (SPoC w/o center prior)	0.802	0.589	0.578	3.65
SPoC (with center prior)	0.784	0.657	0.642	3.66

Ack.: Aggregating Deep Convolutional Features for Image Retrieval

R-MAC: Regional Maximum Activation of Convolutions

- **Use maximum activation of convolutions for translation invariance**
- **Consider uniformly generated regions with different scales, and take the maximum per each feature channel**
  - **Aggregation can be considered as a simple method for cross matching among all possible regions**



Ack.: PARTICULAR OBJECT RETRIEVAL WITH INTEGRAL MAX-POOLING

Approximate Integral Max-Pooling

- **Approximate the maximum with L_p norm**
  - $\alpha = 10$

$$\tilde{f}_{\mathcal{R},i} = \left(\sum_{p \in \mathcal{R}} \mathcal{X}_i(p)^{\alpha} \right)^{\frac{1}{\alpha}} \approx \max_{p \in \mathcal{R}} \mathcal{X}_i(p) = f_{\mathcal{R},i},$$

- **Need to sum values of many different regions**
  - **Use integral images, summed-area table, of features**
  - **Do not need to extract features again from regions**

Post-Processing

- Once a shortlist is identified, various post-processing can be adopted
- Localization
  - Exhaustive search takes too much time
  - Refine box coordinates from initial responses
- Reranking and query expansion can be performed

Fine-Tuning for Search

- **Use CNN features that were trained with ImageNet**
- **Retraining with a task-specific dataset achieve higher accuracy**
  - **Can lower accuracy when using dissimilar datasets**

Fine-Tuning for Search

Descriptor	Dims	Oxford	Oxford 105K	Holidays	UKB
Fisher+color[7]	4096	—	—	**0.774**	3.19
VLAD+adapt+innorm[2]	32768	0.555	—	0.646	—
Sparse-coded features[6]	11024	—	—	0.767	**3.76**
Triangulation embedding[9]	8064	**0.676**	**0.611**	0.771	3.53
Neural codes trained on ILSVRC					
Layer 5	9216	0.389	—	0.690*	3.09
Layer 6	4096	0.435	0.392	0.749*	3.43
Layer 7	4096	0.430	—	0.736*	3.39
After retraining on the Landmarks dataset					
Layer 5	9216	0.387	—	0.674*	2.99
Layer 6	4096	0.545	0.512	**0.793***	3.29
Layer 7	4096	0.538	—	0.764*	3.19
After retraining on turntable views (Multi-view RGB-D)					
Layer 5	9216	0.348	—	0.682*	3.13
Layer 6	4096	0.393	0.351	0.754*	3.56
Layer 7	4096	0.362	—	0.730*	3.53

Landmark dataset has similar images to Oxford

Ack.: Neural Codes for Image Retrieval

KAIST

Results before & after retraining

Dimension Reduction

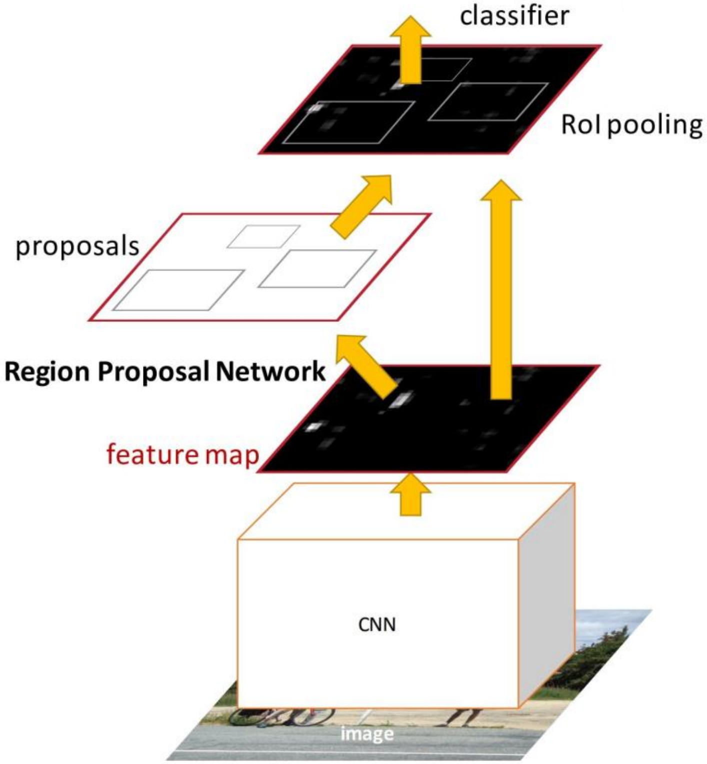
- **CNN features (4096D) are robust to PCA compression**
  - **Maintain accuracy by 256 D**

Dimensions	16	32	64	128	256	512
Oxford						
Layer 6	0.328	0.390	0.421	0.433	0.435	0.435
Layer 6 + landmark retraining	0.418	0.515	0.548	0.557	0.557	0.557
Layer 6 + turntable retraining	0.289	0.349	0.377	0.391	0.392	0.393

Outline

- Representations
  - Fine-tuning
  - Dimension reduction
- Localization and detections
- Not that much on:
  - Post-processing
  - Matching

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

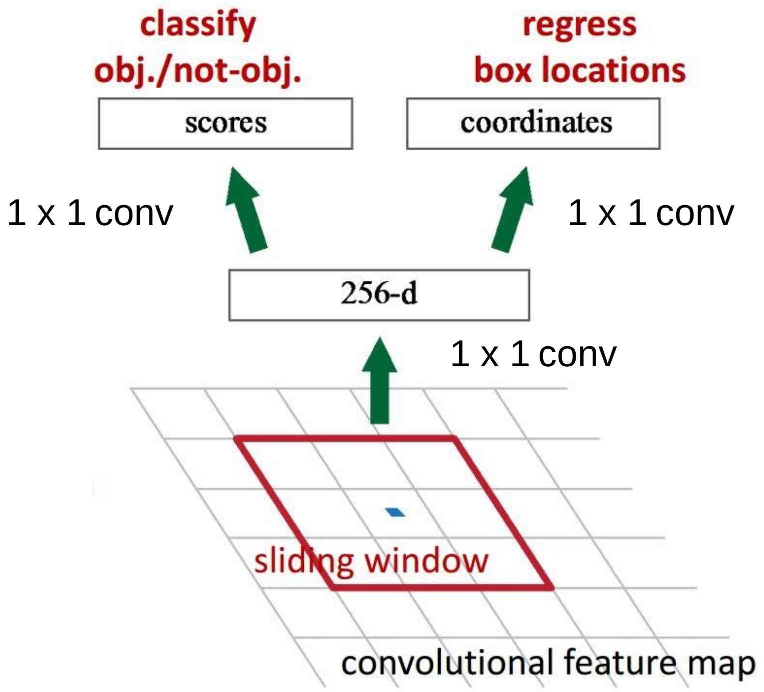
Faster R-CNN: Region Proposal Network

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

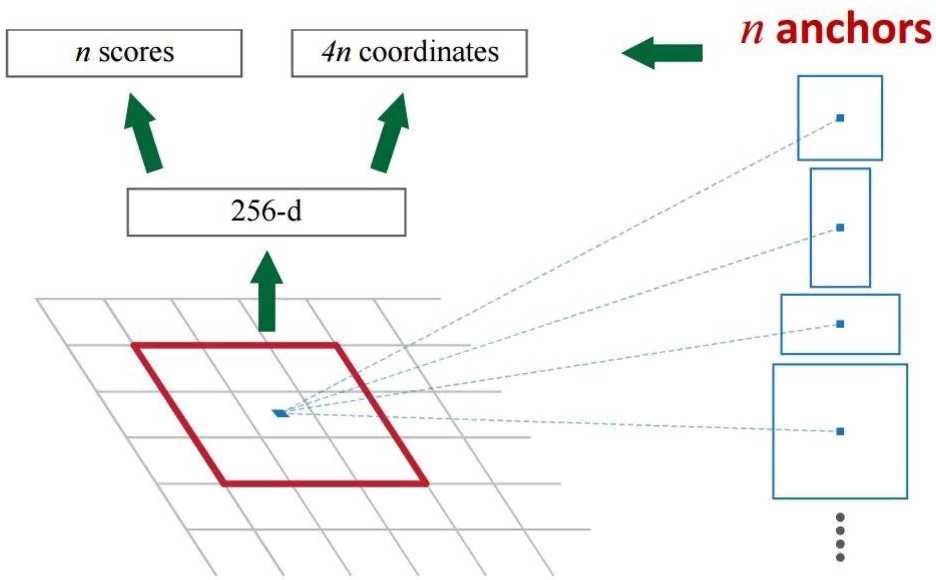
Faster R-CNN: Region Proposal Network

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: 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**

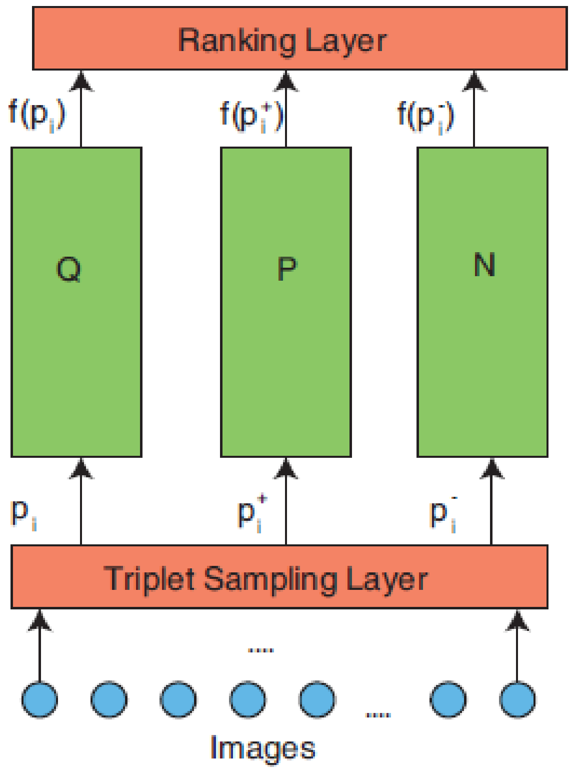
Object Detection State-of-the-art: ResNet 101 + Faster R-CNN + some extras

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

Instance-Level or Fine-Grained Image Search

- **Uses a ranking loss w/ triplet of data**
  - **Used commonly for metric learning**

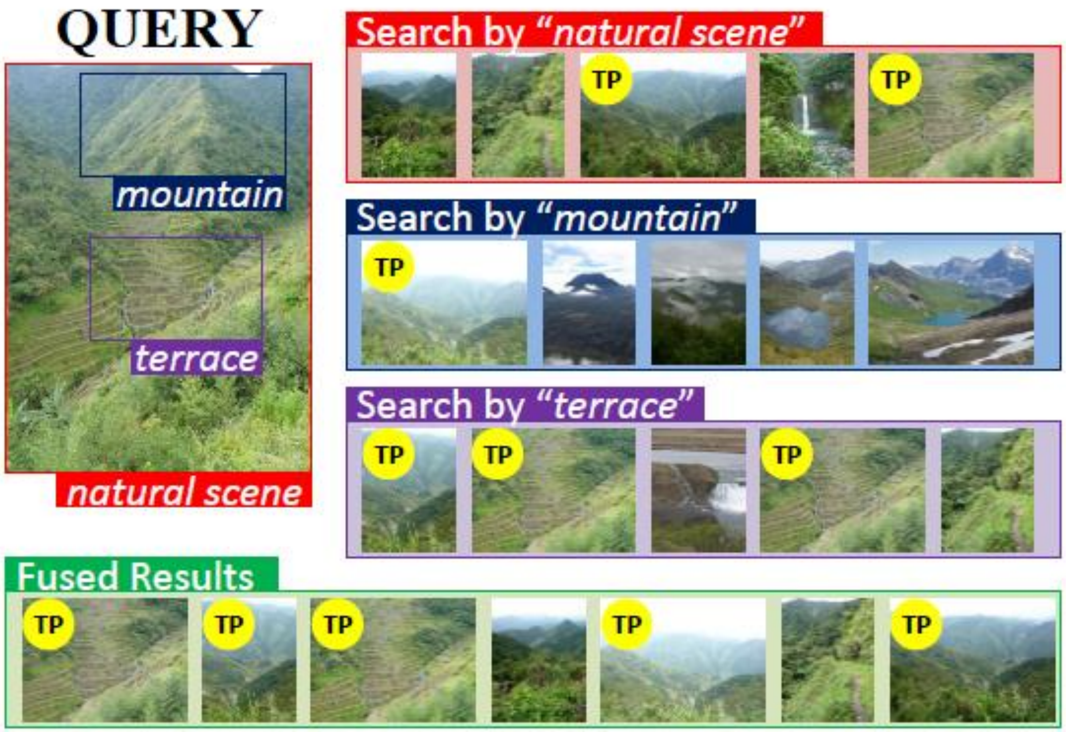


- **Ranking model based on the Siamese network**
- **Given a image p_i, p_i^+ and p_i^- are similar and dissimilar images**
- **The Siamese network may share CNN features**

Ack.: Deep Image retrieval &
Learning Fine-grained Image Similarity with Deep Ranking

Image Classification and Retrieval are ONE [ICMR 15]

- **Handle the classification and search in a unified framework**
  - **Uses region proposals**
  - **Uses nearest neighbor search for both problems**



Outline

- Representations
  - Fine-tuning
  - Dimension reduction
- Localization and detections
- Not that much on:
  - Post-processing
  - Matching