“Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles”

By Mehdi Noroozi and Paolo Favaro
Presentor: Chenshan Yuan

Outline

- Introduction
  - Motivation
  - Goal
  - Related work
- Design and Architecture
  - CFN: Context Free Network
- Experiments and Results
  - Implication

Introduction

- Object classification and detection have been successfully approached through ***supervised learning***.
- BUT, what about ***unsupervised learning***?

Related Work : Self-Supervised Learning

“Unsupervised Visual Representation Learning by Context Prediction”,
By Doersch, C., Gupta, A., Efros, A.A.

- Relative spatial co-location of patches in image as label

“Unsupervised Learning of Visual Representations Using Videos”,
By Wang, X., Gupta, A.

- Object correspondence obtained through tracking in videos

“Learning to see by moving”,
By Agrawal, P., Carreira, J. Malik, J.

- Egomotion information used as supervisory signal for feature learning

Introduction

- Problem: Self-supervised learning CNN
- Solution:
  - Train CNN to solve Jigsaw puzzle as pretext task
  - Transfer learned features to solve object classification and detection

Overall Architecture

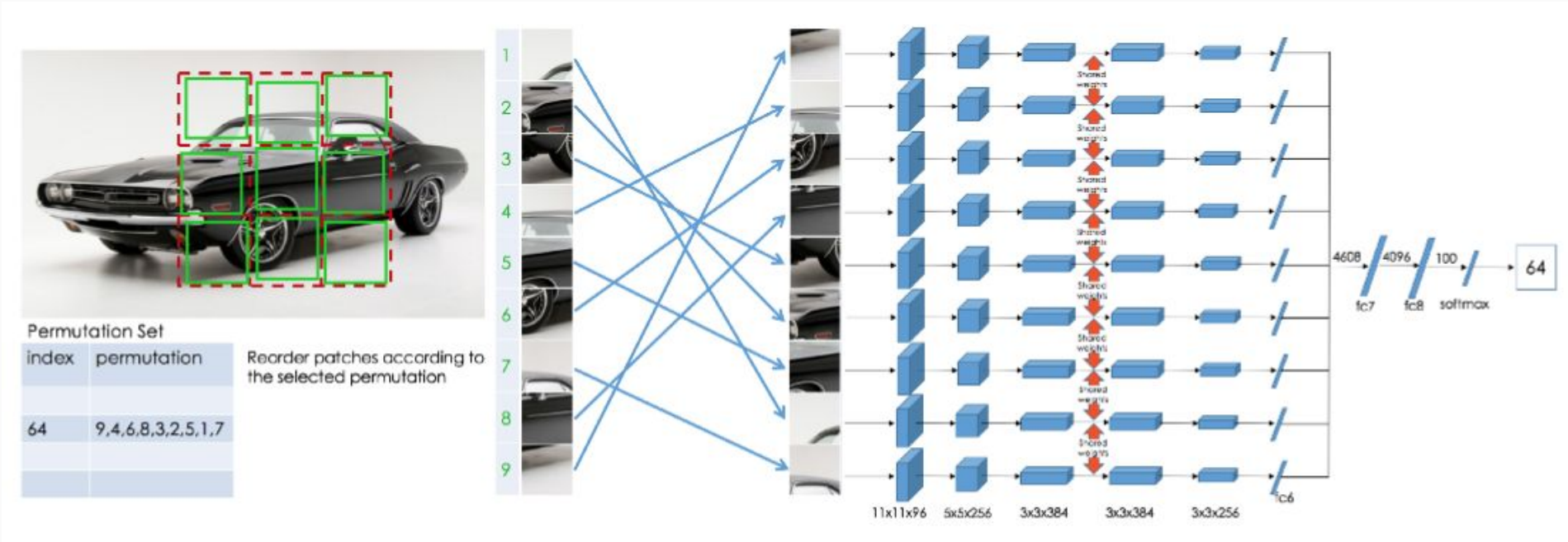


Figure 3, on Page 7, Context Free Network.

Overall Architecture

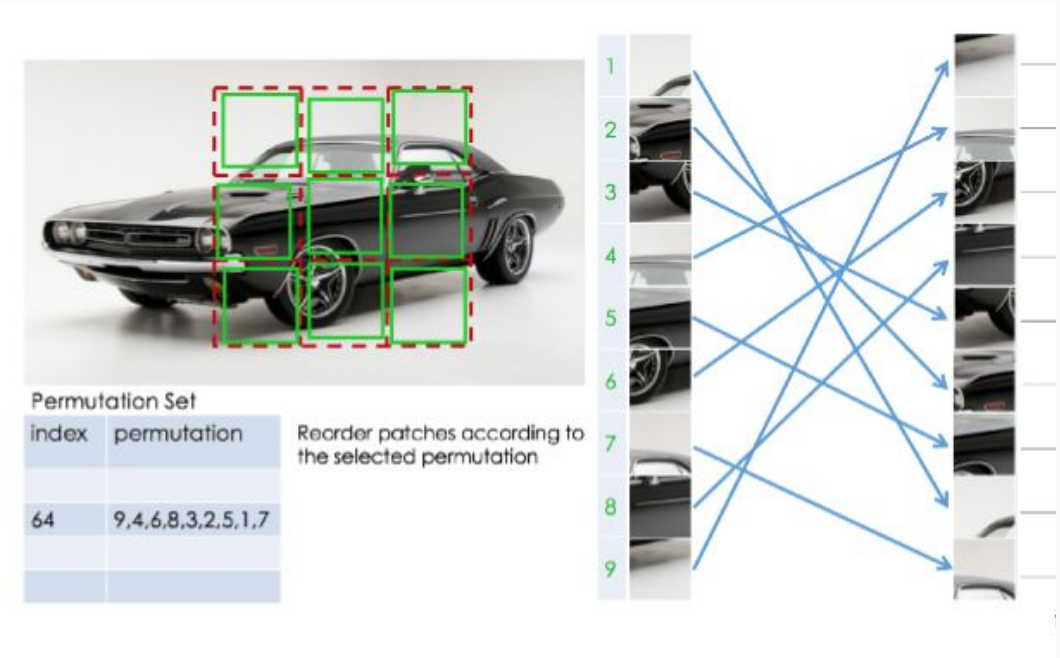


Figure 3, on Page 7, Context Free Network.

- An image:
  - Crop a 225 x 225 pixel window
  - Divide into 3x3 grid
  - Randomly pick 64 x 64 pixel tiles (from 75 x 75 pixel cell)

Overall Architecture: Context Free Network

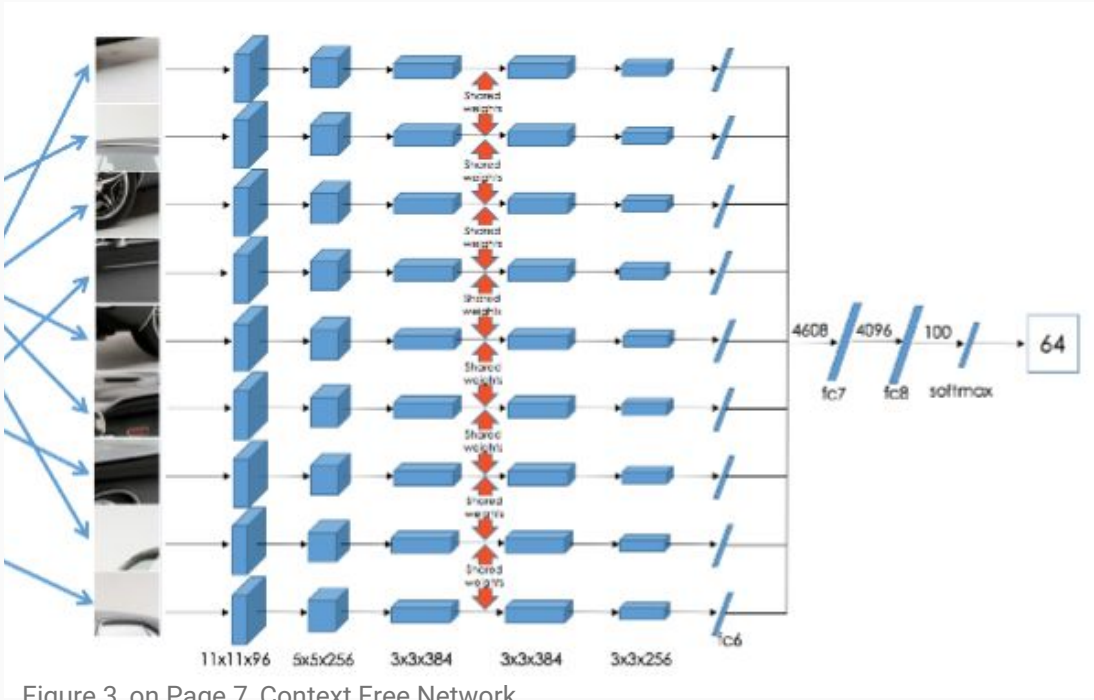


Figure 3, on Page 7, Context Free Network.

- Each row up to the first fully connected layer(fc6) uses AlexNet
- Context handled only in the last fully connected layers

Train CFN:Jigsaw Puzzle Task

- Jigsaw Puzzle Permutations
  - Example: S = (3,1,2,9,5,4,8,7,6)
  - Given 9 tiles, 9! = 362880 permutations in total
- Use only a subset of 100 permutations
  - Selected based on Hammington distance between permutations
  - H(S1, S2) = # of different tile locations / 9

Train CFN:Generating permutation set using MAX Hamming distance

Algorithm 1. Generation of the *maximal* Hamming distance permutation set

Output: P

$\bar{P} \leftarrow$ all permutations $[\bar{P}_1, \ldots, \bar{P}_{9!}]$ \\ $\bar{P}$ *is a* $9 \times 9!$ *matrix*

$P \leftarrow \emptyset$

$j \sim \mathcal{U}[1, 9!]$ \\ *uniform sample out of* 9! *permutations*

$i \leftarrow 1$

repeat

  $P \leftarrow [P\ \bar{P}_j]$ \\ *add permutation* $\bar{P}_j$ *to* P

  $\bar{P} \leftarrow [\bar{P}_1, \ldots, \bar{P}_{j-1}, \bar{P}_{j+1}, \ldots]$ \\ *remove* $\bar{P}_j$ *from* $\bar{P}$

  $D \leftarrow \text{Hamming}(P, P')$ \\ D *is an* $i \times (9! - i)$ *matrix*

  $\bar{D} \leftarrow \mathbf{1}^T D$ \\ $\bar{D}$ *is a* $1 \times (9! - i)$ *row vector*

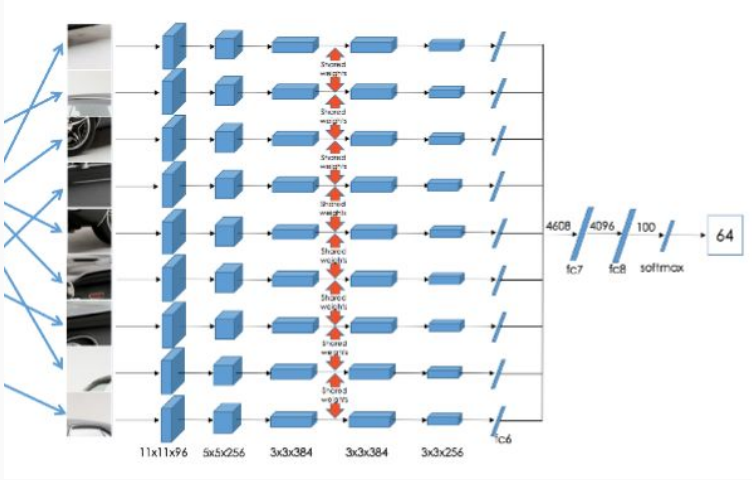
  $j \leftarrow \arg\max_k \bar{D}_k$ \\ $\bar{D}_k$ *denotes the* k*-th entry of* $\bar{D}$

  $i \leftarrow i + 1$

until $i \leq 100$

Train CFN

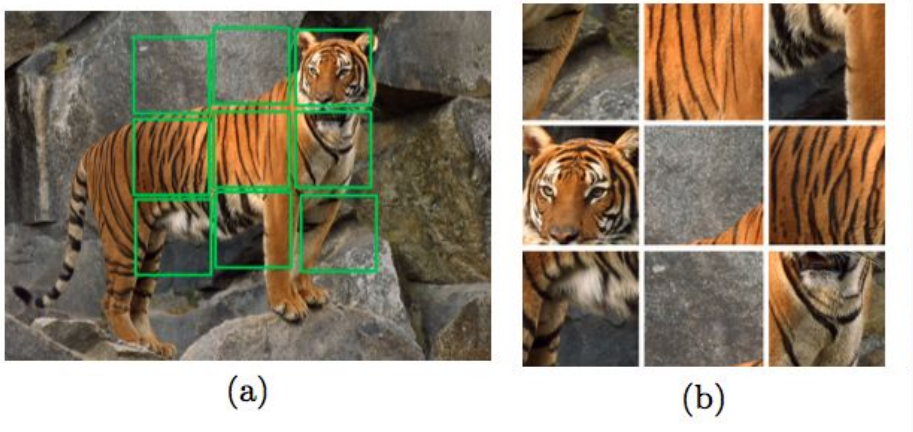
- Objective: train CFN so that feature have semantic attribution to relative position between parts.
- Output of CFN can be seen as conditional probability density function(pdf) of spatial arrangement of object parts



Train CFN

- If configuration stated as (1), CFN learns to associate each part to an absolute position (arbitrary 2D position)

$$p(S|A_1, A_2, \ldots, A_9) = p(S|F_1, F_2, \ldots, F_9) \prod_{i=1}^{9} p(F_i|A_i) \quad (1)$$



(a) (b)

Train CFN

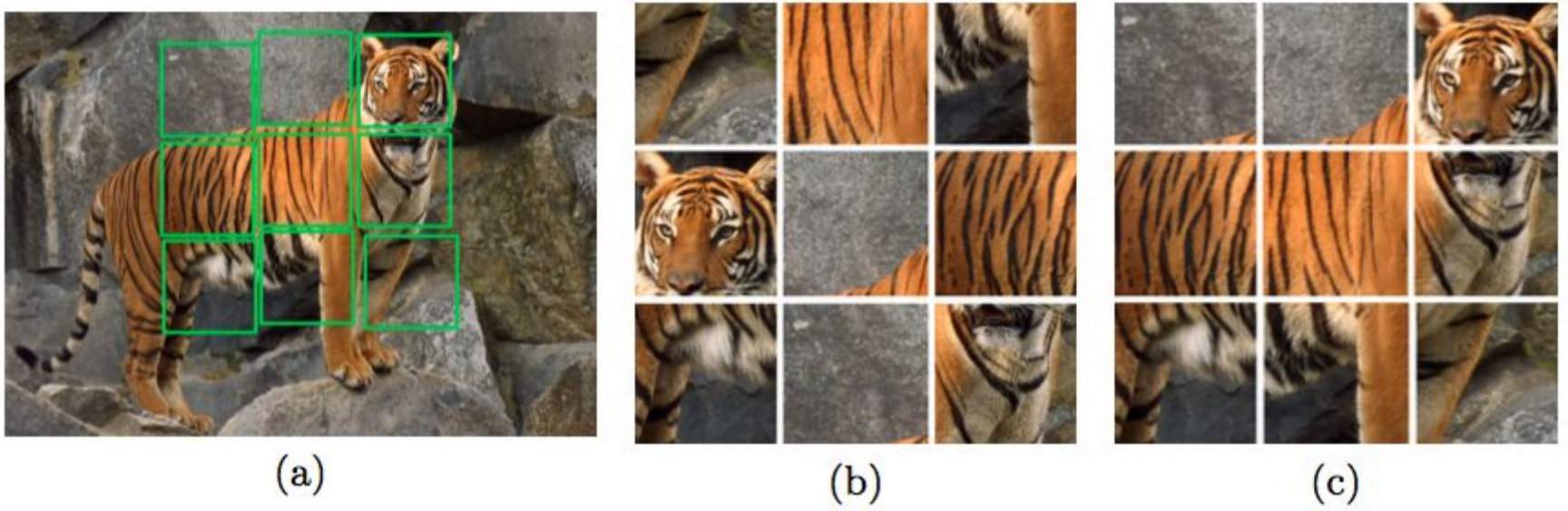
- If configuration represented as list of tile positions S = (L1, L2,...L9), then pdf p(S|F1,F2...F9) can factorize into independent terms.

$$p(L_1, \ldots, L_9 | F_1, F_2, \ldots, F_9) = \prod_{i=1}^{9} p(L_i | F_i) \qquad (2)$$

- Each tile location is determined by its feature but no correlation across tiles can be learn

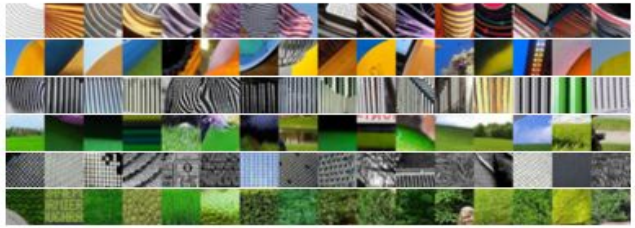
Train CFN

- Feed multiple Jigsaw puzzles for same image into CFN
- Tiles are shuffled such that each tile are assigned to different location in different configuration



Train CFN: Filter activation

(a) conv1 activations

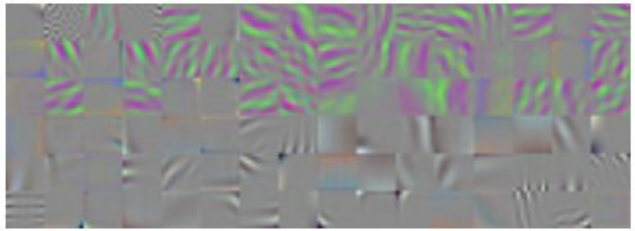


(b) conv2 activations

(c) conv3 activations

(d) conv4 activations

(e) conv5 activations



(f) conv1 filters

- CFN features corresponds to patterns that has similar shapes
- Good correspondence based on object parts

Experiments and Evaluation: Usefulness of semantic classification to solve Jigsaw

Table 1: Transfer learning of AlexNet from a classification task to the Jigsaw puzzle reassembly problem. The j-th column indicates that all layers from `conv1` to `conv-`j were locked and all subsequent layers were randomly initialized and retrained. Notice how the first 4 layers provide very good features for solving puzzles. This shows that object classification and the Jigsaw puzzle problems are related.

	🔒 conv1	🔒 conv2	🔒 conv3	🔒 conv4	🔒 conv5
AlexNet [25]	88	87	86	83	74

- Semantic information is helpful towards recognizing object parts.

Experiments and Evaluation: ImageNet Classification

Table 2: Comparison of classification results on ImageNet LSVRC 2012 [9].

	🔒 conv1	🔒 conv2	🔒 conv3	🔒 conv4	🔒 conv5
CFN	**57.1**	**56.0**	**52.4**	**48.3**	**38.1**
Doersch *et al.* [10]	53.1	47.6	48.7	45.6	30.4
Wang and Gupta [38]	51.8	46.9	42.8	38.8	29.8
Random	48.5	41.0	34.8	27.1	12.0

- Conv5 specialize in Jigsaw puzzle reassembly task

Experiments and Evaluation: Pascal VOC 2007 Classification and Detection

Table 3: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [29].

Method	Pretraining time	Supervision	Classification	Detection
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**
Wang and Gupta[38]	1 week	motion	58.4%	44.0%
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%
Pathak *et al.* [29]	14 hours	context	56.5%	44.5%
CFN-9(max)	2.5 days	context	**68.6%**	**51.8%**

CFN-4	CFN-9(min)	CFN-9(middle)	CFN-9(max)	CFN-sup
49.8%	51.0%	51.2%	**51.8%**	**56.3%**

Table 4.

Experiments and Evaluation: Image Retrieval

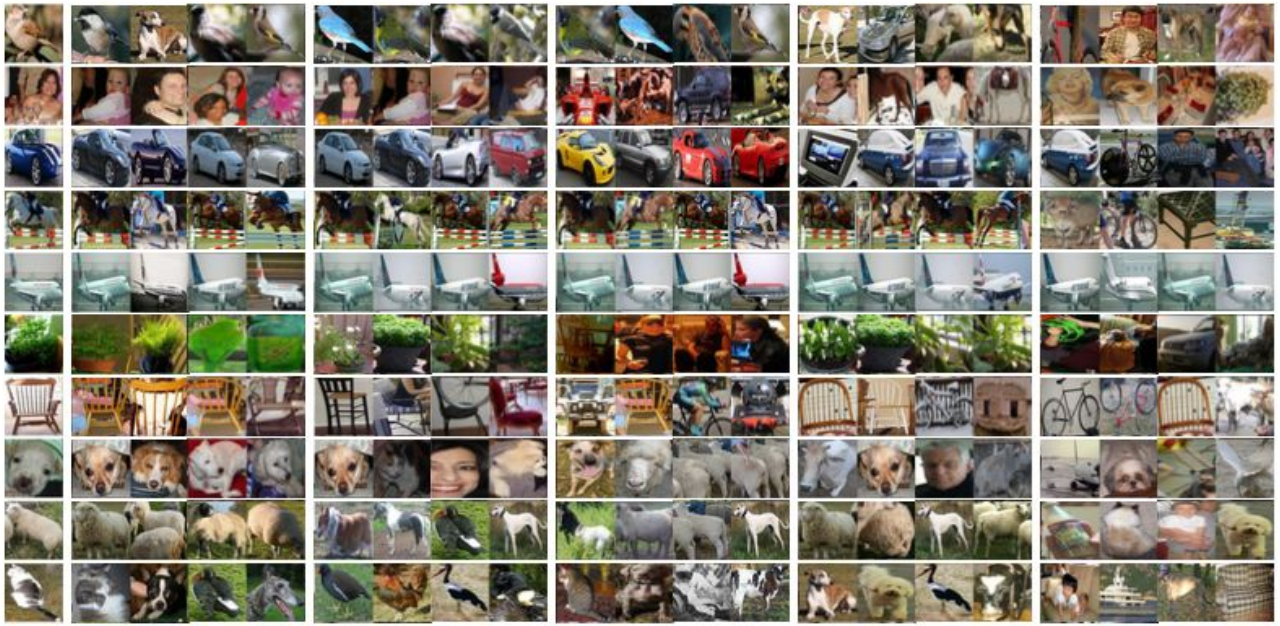


Fig. 5: Image retrieval (qualitative evaluation). (a) query images; (b) top-4 matches with AlexNet; (c) top-4 matches with the CFN; (d) top-4 matches with Doersch *et al.* [10]; (e) top-4 matches with Wang and Gupta [38]; (f) top-4 matches with AlexNet with random weights.

Experiments and Evaluation: Image Retrieval

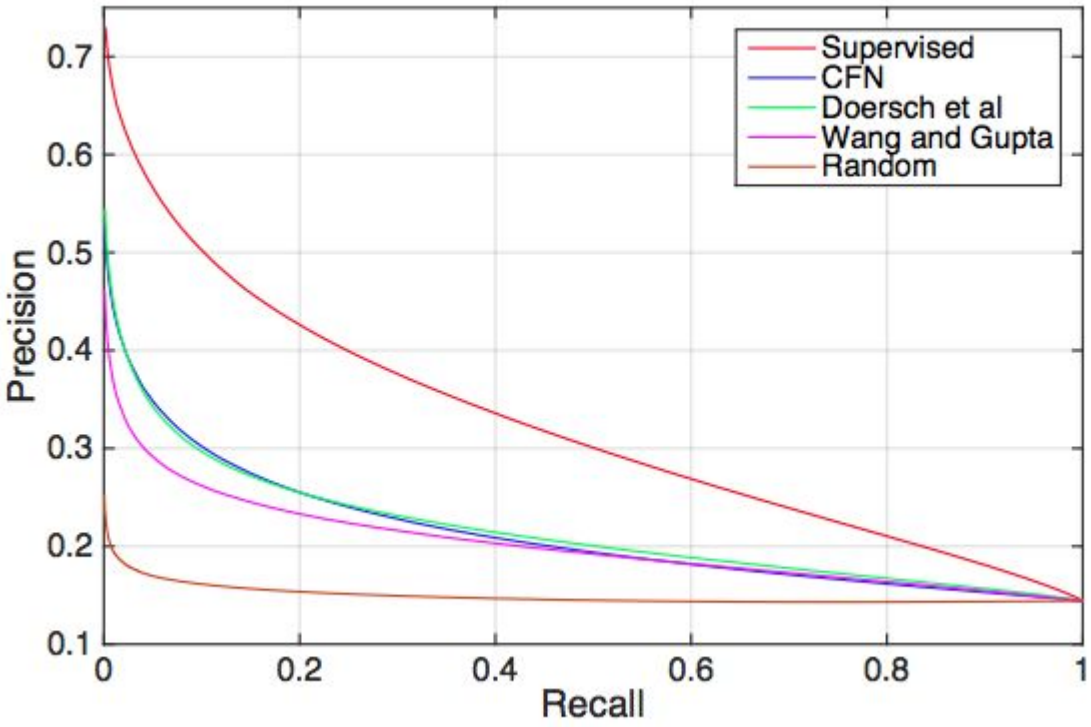


Figure 6, Images retrieval.

Conclusion

- CFN can use features learned from solving Jigsaw puzzle to solve visual challenges
- Explored potential in self-supervised learning and provide alternatives to human annotation in future