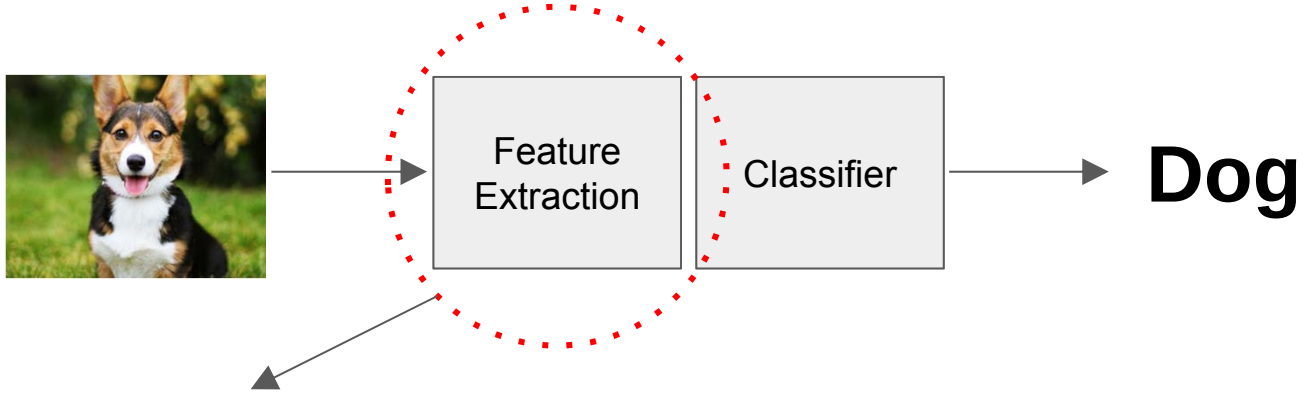
Representation Learning by Learning to Count

Presented by: Muhammad Tayyab

Background

★ Supervised representation learning

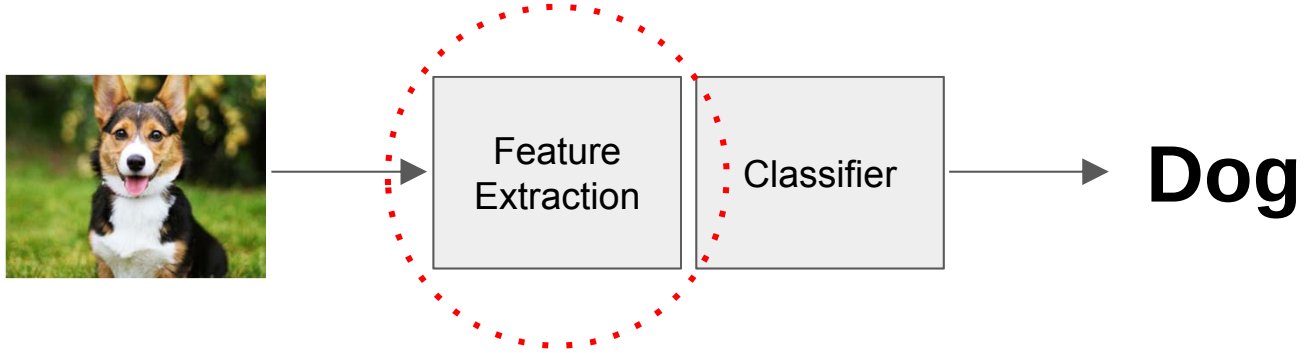


Object classification
Object detection
Semantic segmentation

https://www.youtube.com/watch?v=ZmaXDb9akEI

Background

★ Supervised representation learning

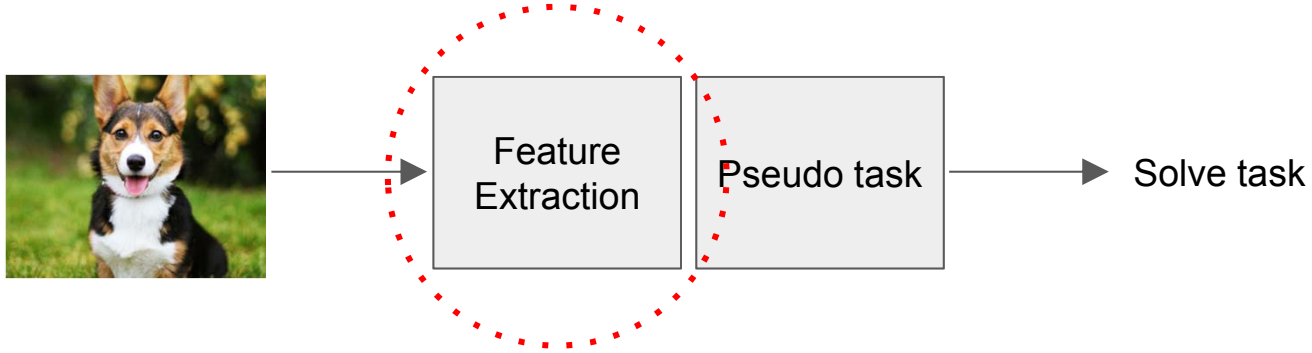


Require human annotation

- Costly
- Error prone
- Time consuming

Background

★ Self-supervised representation learning

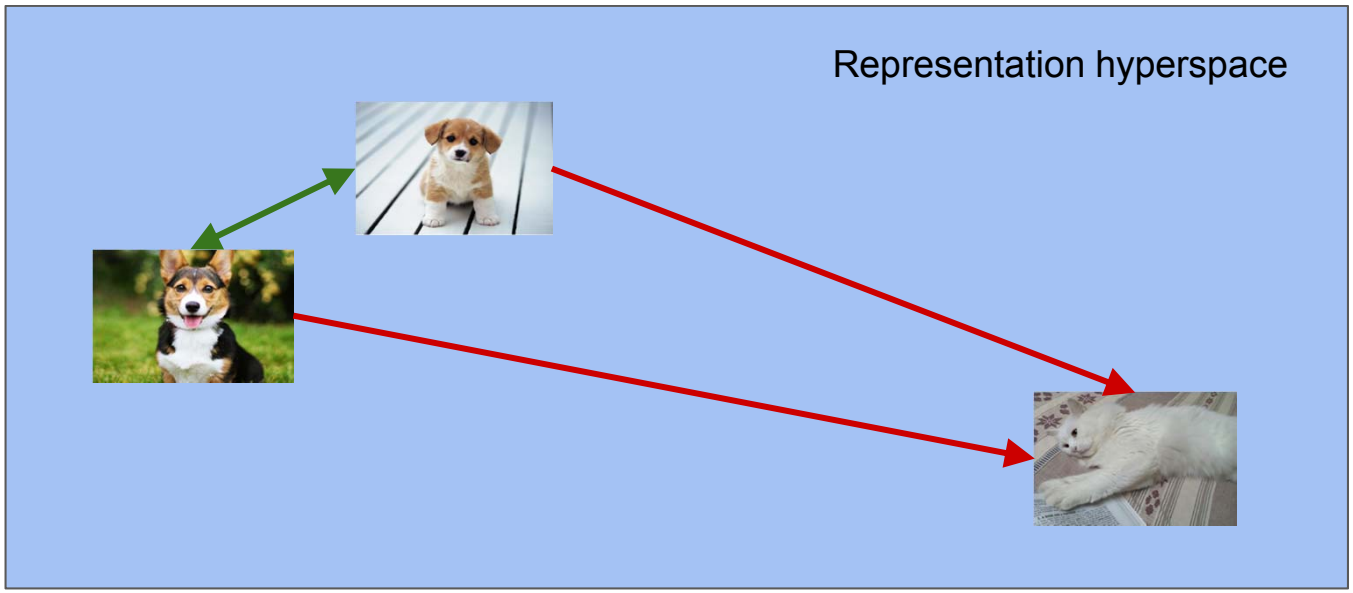


Doesn’t require human annotation

- Cheap
- Scalable

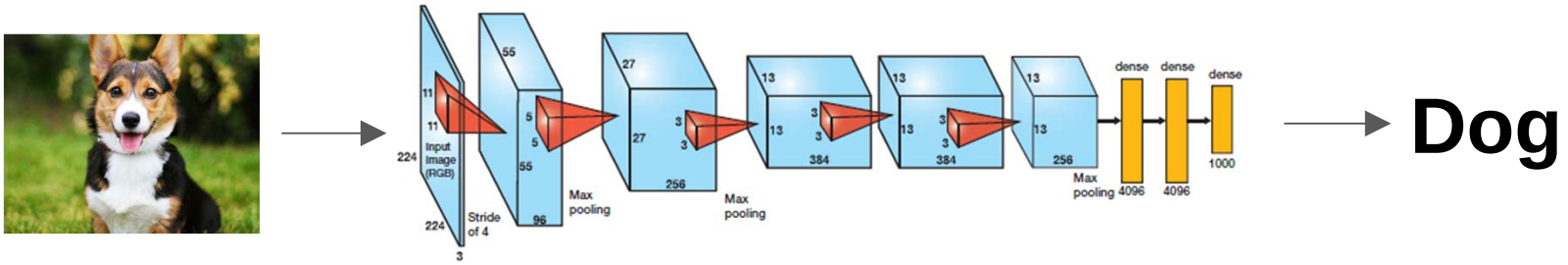
Background

★ Good representations



Background

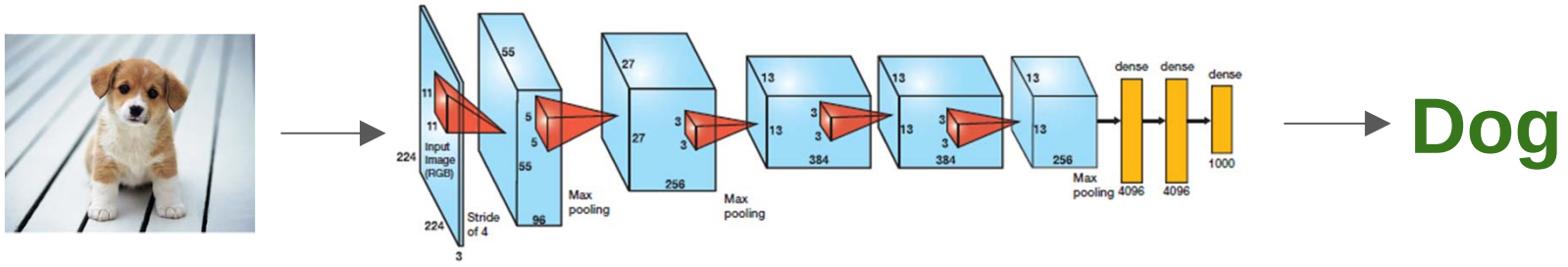
★ Representation learning



http://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

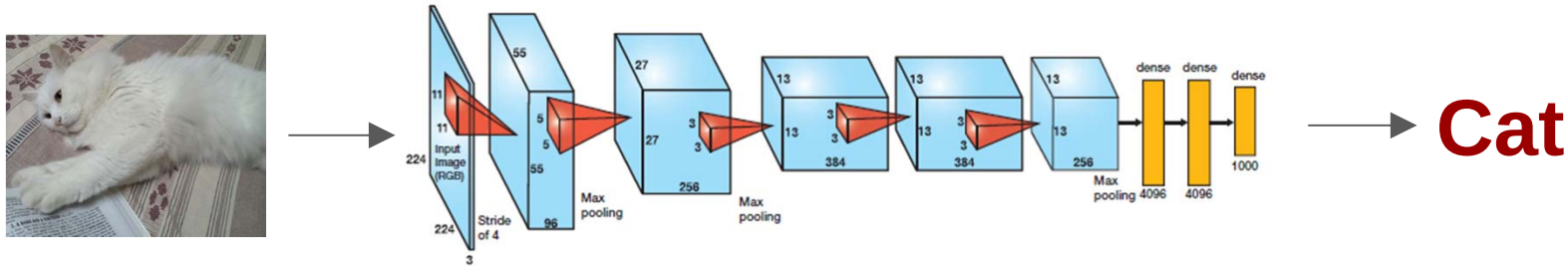
Background

★ Representation learning



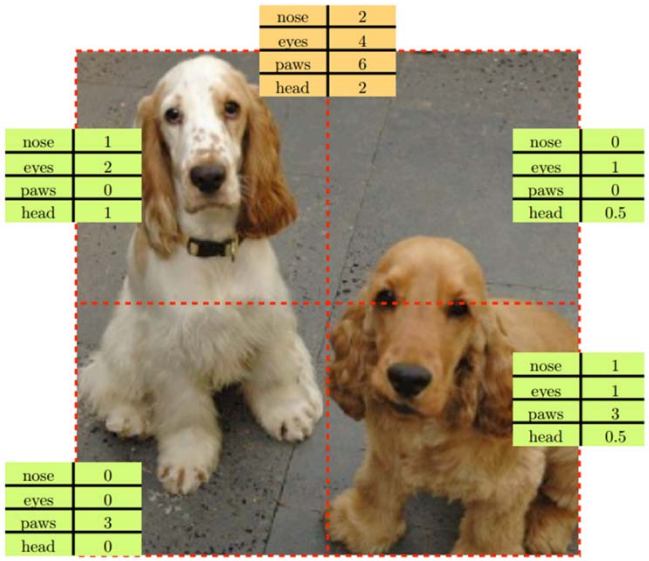
Background

★ Representation learning



Similarity based on counting

★ Idea: Sum of predicted patchwise *feature* count should be similar to the image *feature* count



Learning to Count

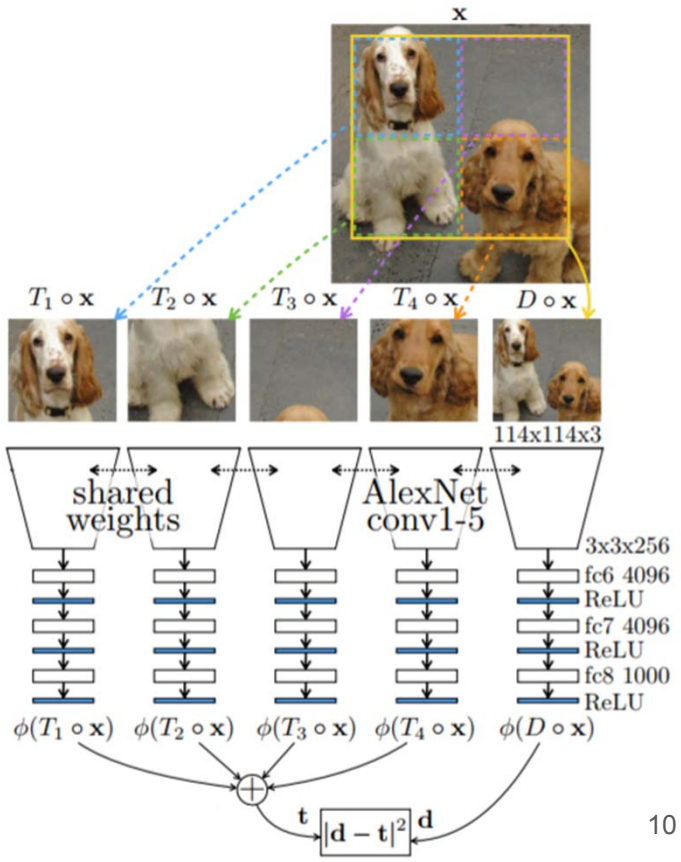
★ Formulation:

- Downsampling operator **D**.
- Tiling operator $\mathbf{T_i}$.

$$\phi(D \circ \mathbf{x}) = \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x})$$

★ Loss

$$\ell(\mathbf{x}) = \left| \phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x}) \right|^2$$



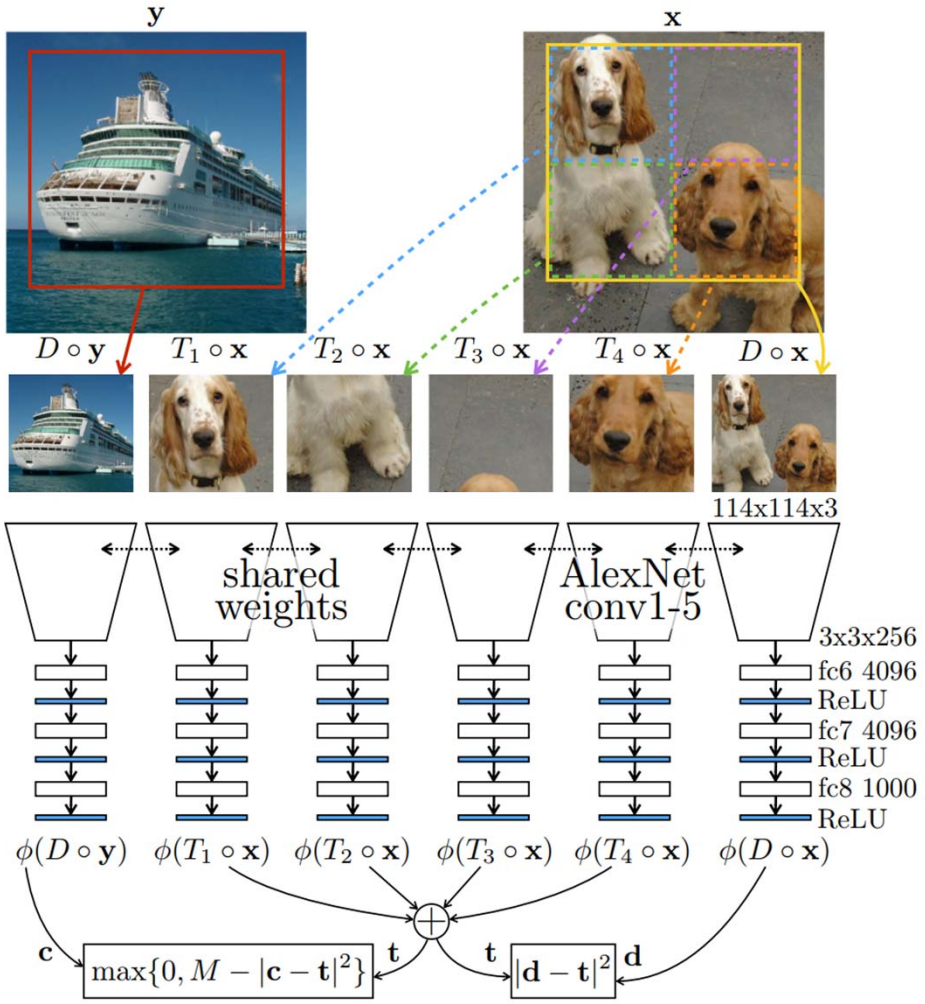
Learning to Count

- ★ Least effort bias
  - Easy to satisfy loss if network always outputs zero.
- ★ Solution?
  - Contrastive loss
  - Learn features useful for discrimination.

Learning to Count

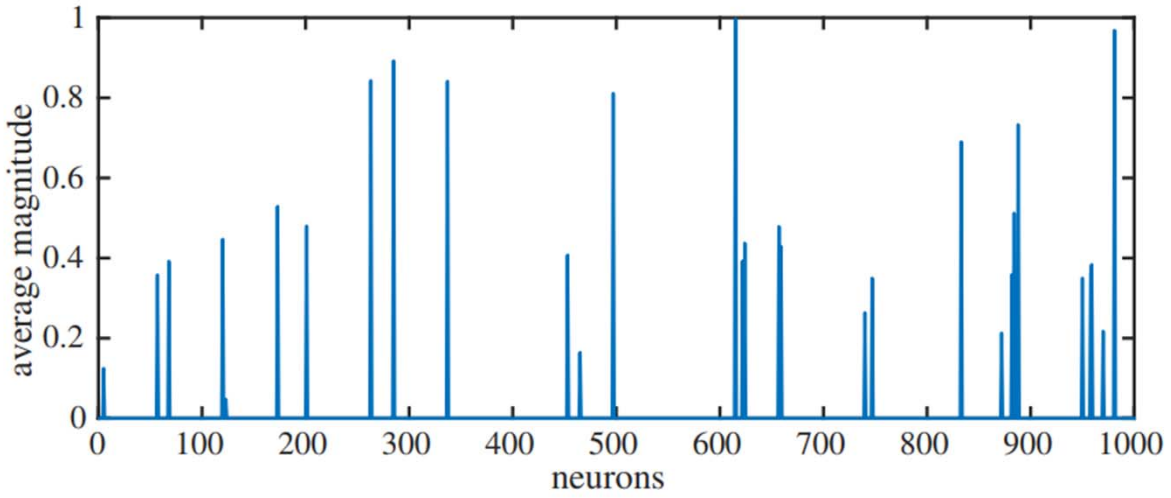
Contrastive loss:

$$\ell_{\text{con}}(\mathbf{x}, \mathbf{y}) = \left| \phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x}) \right|^2 + \max \left\{ 0, M - \left| \phi(D \circ \mathbf{y}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x}) \right|^2 \right\}$$



Experiments

★ Observation



Average response of trained network on the ImageNet validation set

Experiments

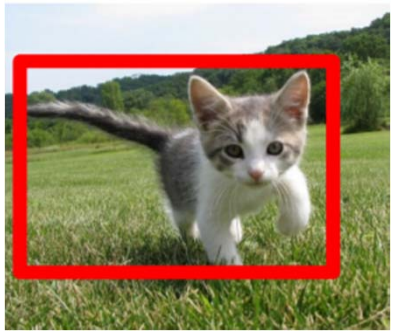
★ Transfer Learning

Cat

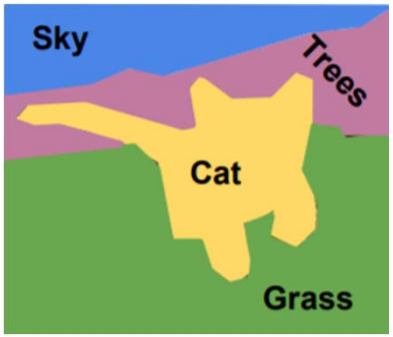


Classification

Cat



Detection



Segmentation

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Experiments

★ Transfer Learning

Method	Ref	Class.	Det.	Segm.
Supervised [20]	[43]	79.9	56.8	48.0
Random	[33]	53.3	43.4	19.8
Context [9]	[19]	55.3	46.6	-
Context [9]*	[19]	65.3	51.1	-
Jigsaw [30]	[30]	67.6	**53.2**	37.6
ego-motion [1]	[1]	52.9	41.8	-
ego-motion [1]*	[1]	54.2	43.9	-
Adversarial [10]*	[10]	58.6	46.2	34.9
ContextEncoder [33]	[33]	56.5	44.5	29.7
Sound [31]	[44]	54.4	44.0	-
Sound [31]*	[44]	61.3	-	-
Video [41]	[19]	62.8	47.4	-
Video [41]*	[19]	63.1	47.2	-
Colorization [43]*	[43]	65.9	46.9	35.6
Split-Brain [44]*	[44]	67.1	46.7	36.0
ColorProxy [22]	[22]	65.9	-	**38.0**
WatchingObjectsMove [32]	[32]	61.0	52.2	-
Counting		**67.7**	51.4	36.6

Evaluation of transfer learning on PASCAL

Experiments

★ Transfer Learning

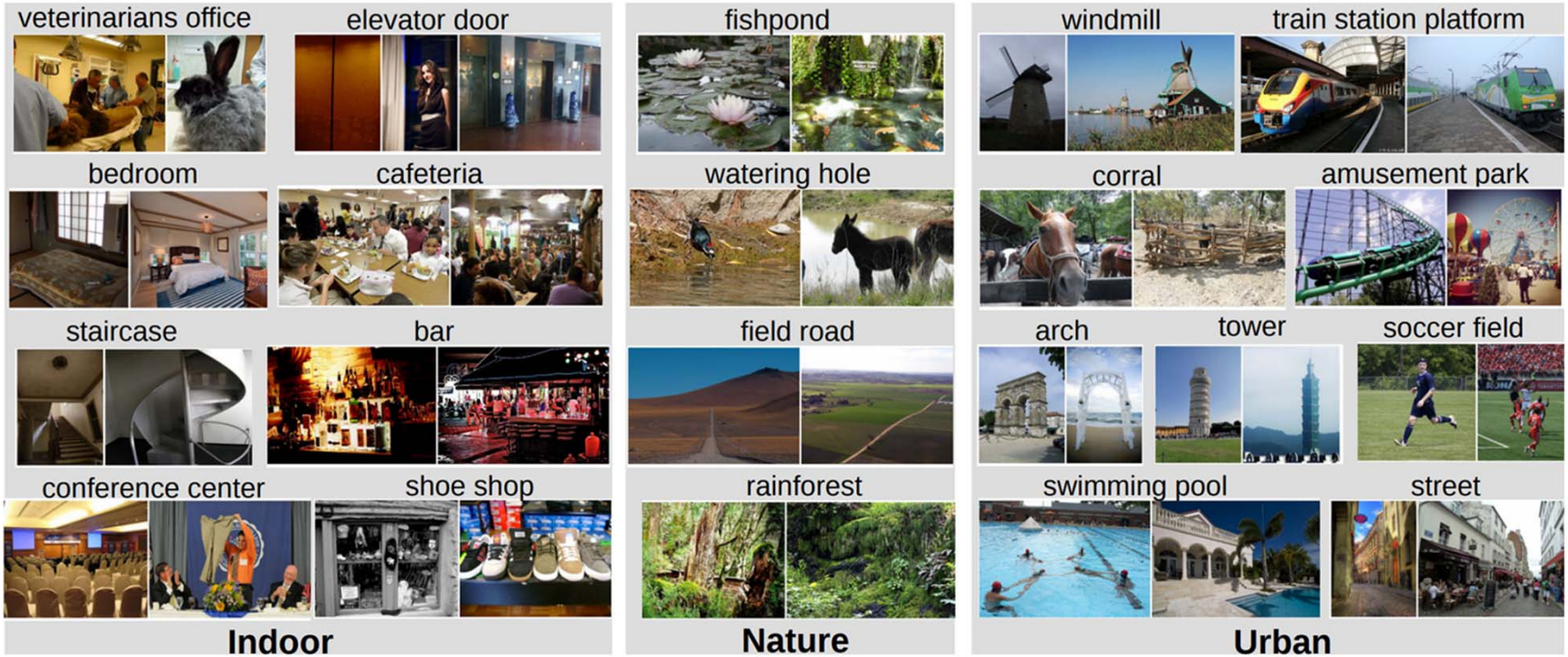
Method	conv1	conv2	conv3	conv4	conv5
Supervised [20]	19.3	36.3	44.2	48.3	50.5
Random	11.6	17.1	16.9	16.3	14.1
Context [9]	16.2	23.3	30.2	31.7	29.6
Jigsaw [30]	**18.2**	28.8	34.0	33.9	27.1
ContextEncoder [33]	14.1	20.7	21.0	19.8	15.5
Adversarial [10]	17.7	24.5	31.0	29.9	28.0
Colorization [43]	12.5	24.5	30.4	31.5	30.3
Split-Brain [44]	17.7	29.3	**35.4**	**35.2**	**32.8**
Counting	18.0	**30.6**	34.3	32.5	25.7

ImageNet classification with a linear classifier

Experiments

- ★ Places data set
  - By MIT
  - 10,624,928 Images
  - 434 Classes

Experiments



Experiments

★ Transfer Learning

Method	conv1	conv2	conv3	conv4	conv5
Places labels [45]	22.1	35.1	40.2	43.3	44.6
ImageNet labels [20]	22.7	34.8	38.4	39.4	38.7
Random	15.7	20.3	19.8	19.1	17.5
Context [9]	19.7	26.7	31.9	32.7	30.9
Jigsaw [30]	23.0	31.9	35.0	34.2	29.3
Context encoder [33]	18.2	23.2	23.4	21.9	18.4
Sound [31]	19.9	29.3	32.1	28.8	29.8
Adversarial [10]	22.0	28.7	31.8	31.3	29.7
Colorization [43]	16.0	25.7	29.6	30.3	29.7
Split-Brain [44]	21.3	30.7	34.0	34.1	**32.5**
Counting	**23.3**	**33.9**	**36.3**	**34.7**	29.6

Places classification with a linear classifier

Experiments

★ Ablation studies

Interpolation method	Training size	Color space	Counting dimension	Detection performance
Mixed	1.3M	RGB/Gray	20	50.9
Mixed	128K	Gray	1000	44.9
Mixed	512K	Gray	1000	49.1
Mixed	1.3M	RGB	1000	48.2
Mixed	1.3M	Gray	1000	50.4
Linear	1.3M	RGB/Gray	1000	48.4
Cubic	1.3M	RGB/Gray	1000	48.9
Area	1.3M	RGB/Gray	1000	49.2
Lanczos	1.3M	RGB/Gray	1000	47.3
Mixed	1.3M	RGB/Gray	1000	**51.4**

Experiments

★ Ablation studies

As an error metric, we use the first term in the loss function normalized by the average of the norm of the feature vector. More precisely, the error when the network is trained with the i-th downsampling style and tested on the j-th one is

$$\mathbf{e}_{ij} = \frac{\sum_{\mathbf{x}} \left| \sum_{p=1}^{4} \phi^i \left(T_p \circ \mathbf{x} \right) - \phi^i (D^j \circ \mathbf{x}) \right|^2}{\sum_{\mathbf{x}} \left| \phi^i \left(D^i \circ \mathbf{x} \right) \right|^2}$$

Experiments

★ Ablation studies

train\test	Linear	Cubic	Area	Lanczos	Mixed	std
Linear	**0.33**	0.63	0.33	0.65	0.48	0.18
Cubic	0.79	**0.25**	0.78	0.22	0.52	0.32
Area	0.32	0.85	**0.31**	0.95	0.50	0.34
Lanczos	1.023	0.31	1.02	**0.19**	0.58	0.45
Mixed	0.36	**0.29**	0.37	0.30	0.34	**0.04**

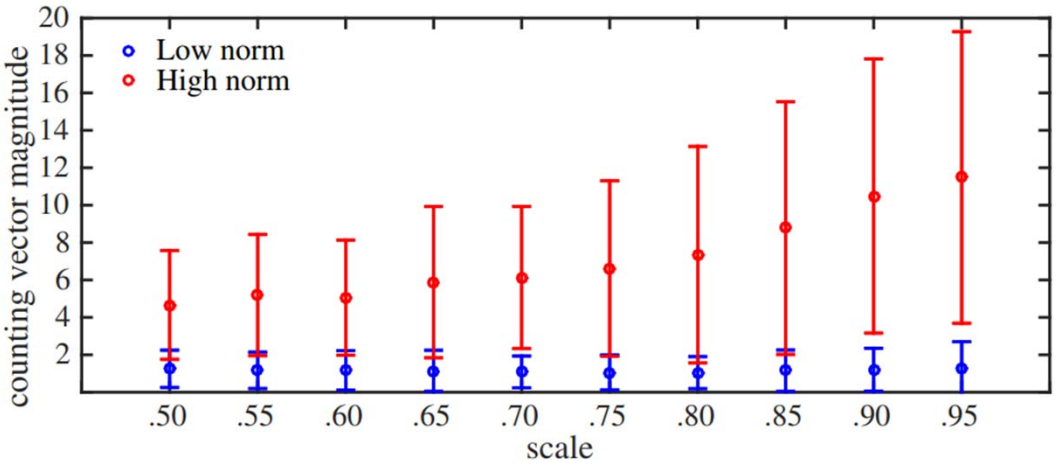
Learning the downsampling style

Experiments

★ Counting

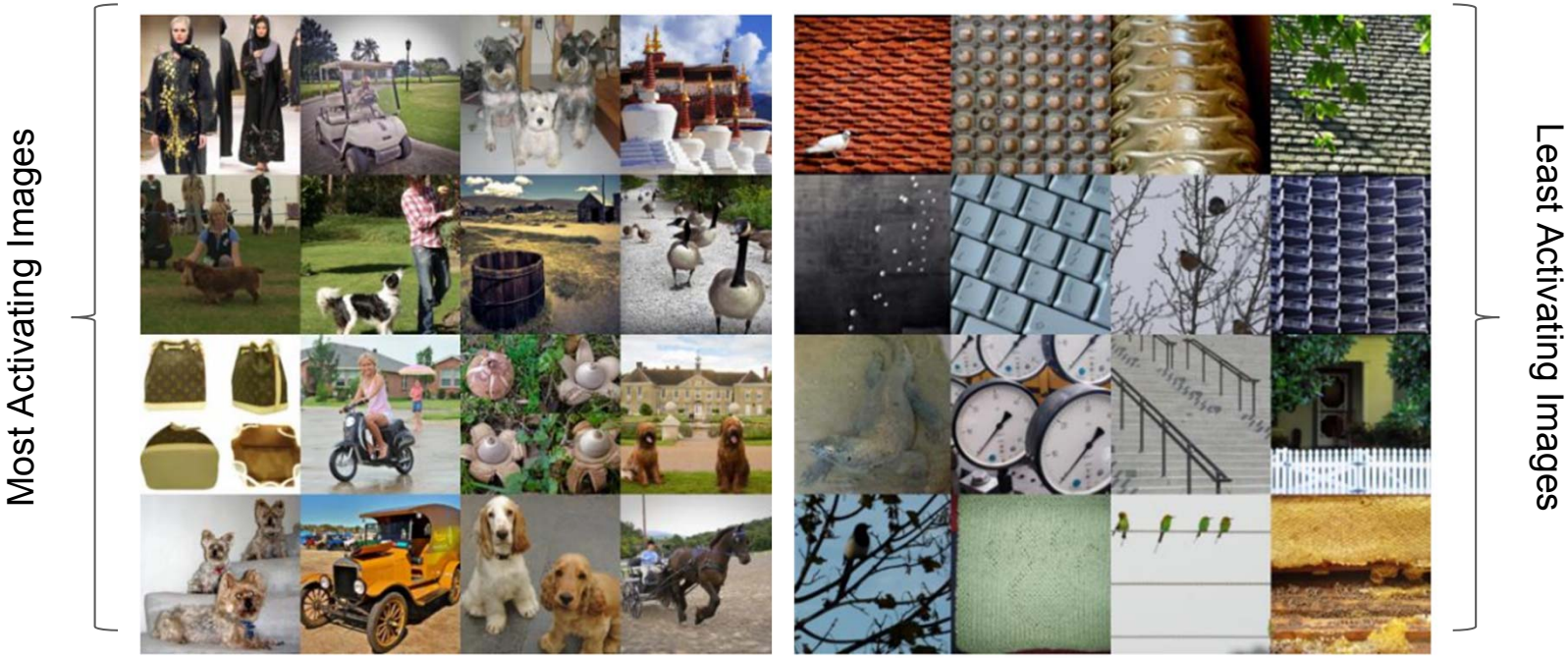


Image croppings of increasing size. The number of visual primitives should increase going from left to right



Counting evaluation on ImageNet

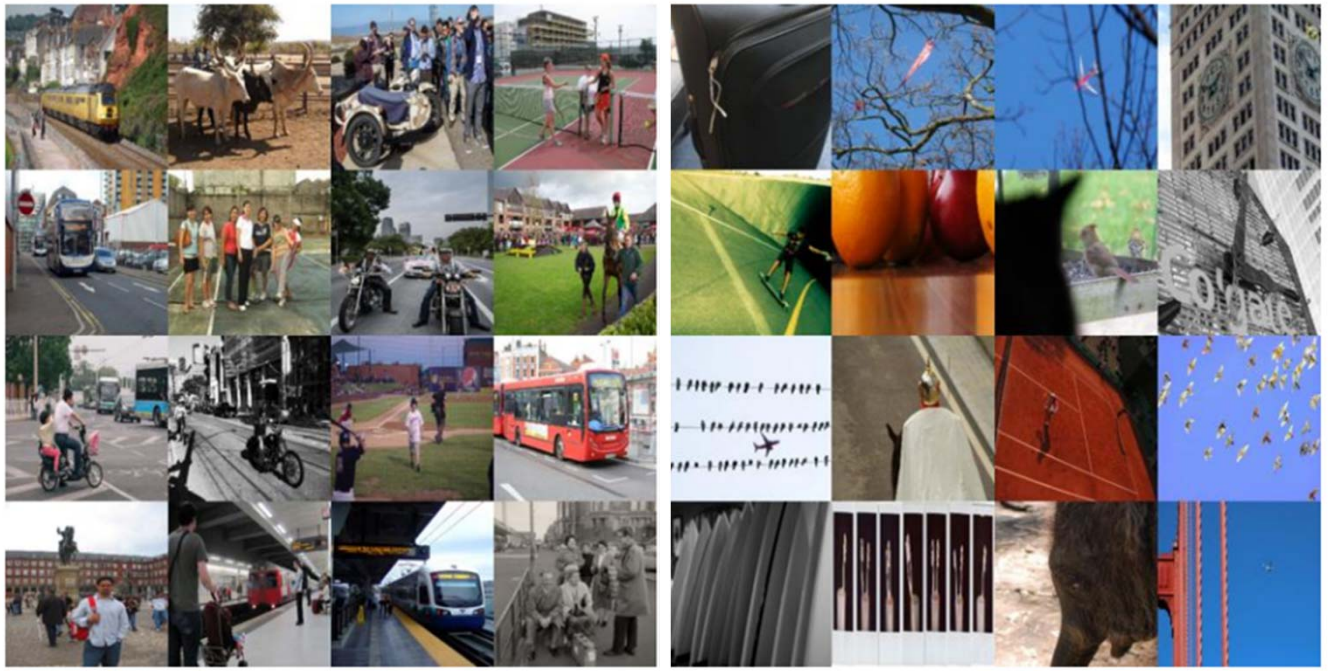
Experiments



Examples of activating/ignored images for **ImageNet** test set

Experiments

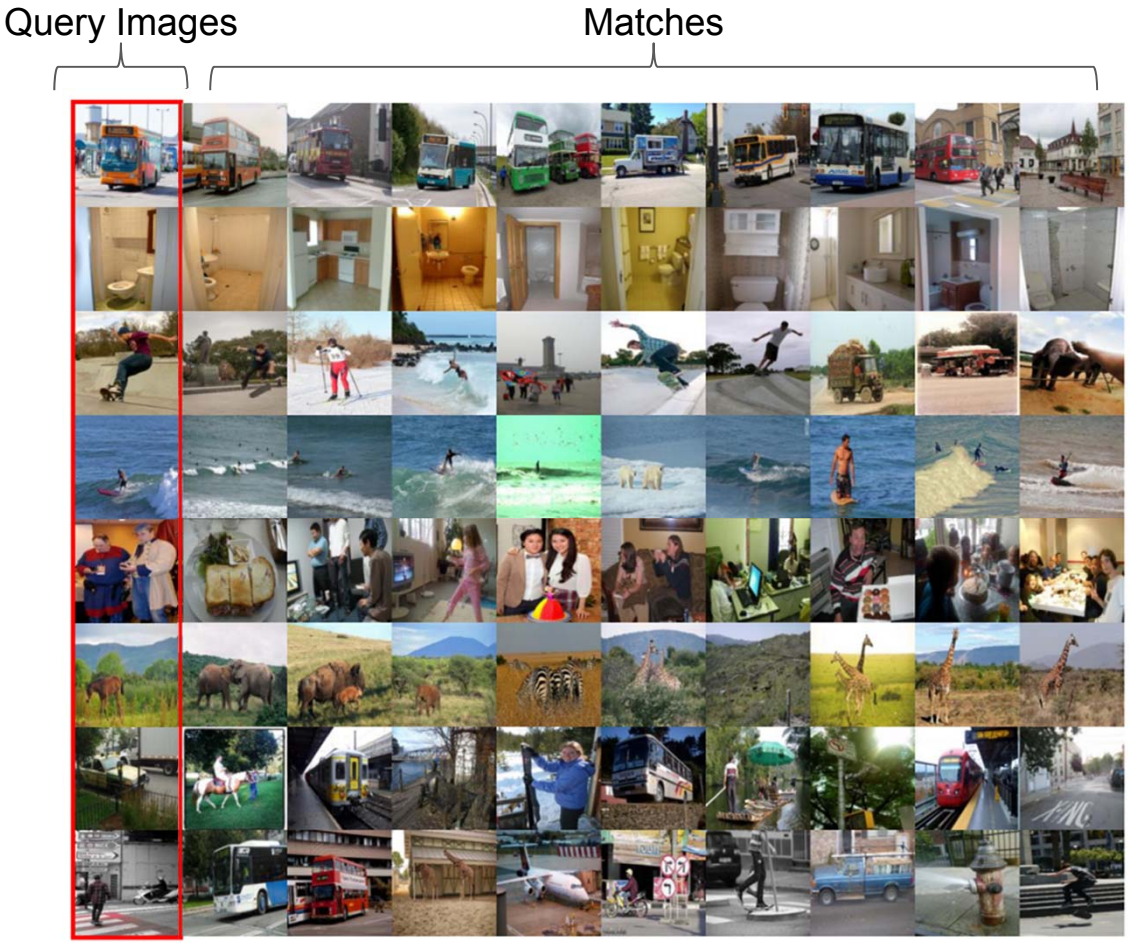
Most Activating Images



Least Activating Images

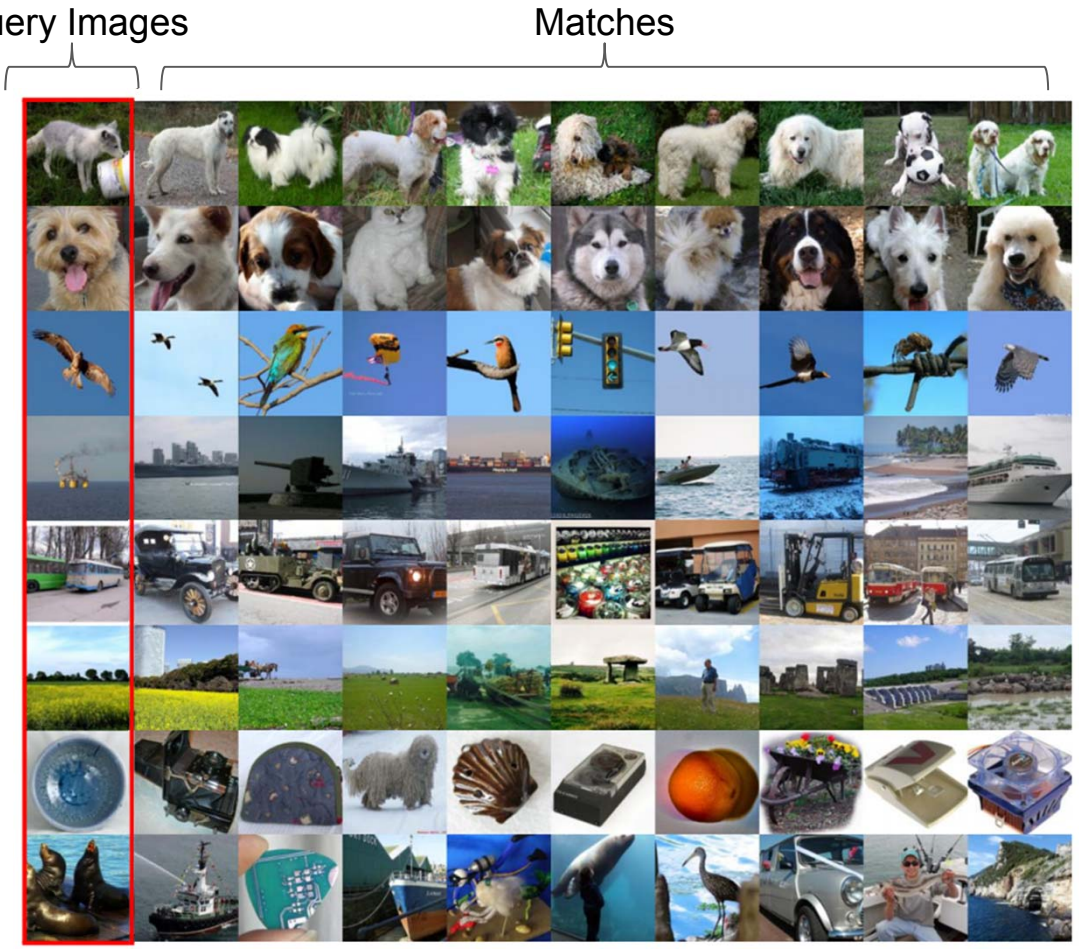
Examples of activating/ignored images for **COCO** test set

Experiments



Nearest neighbor retrievals for **ImageNet**

Experiments



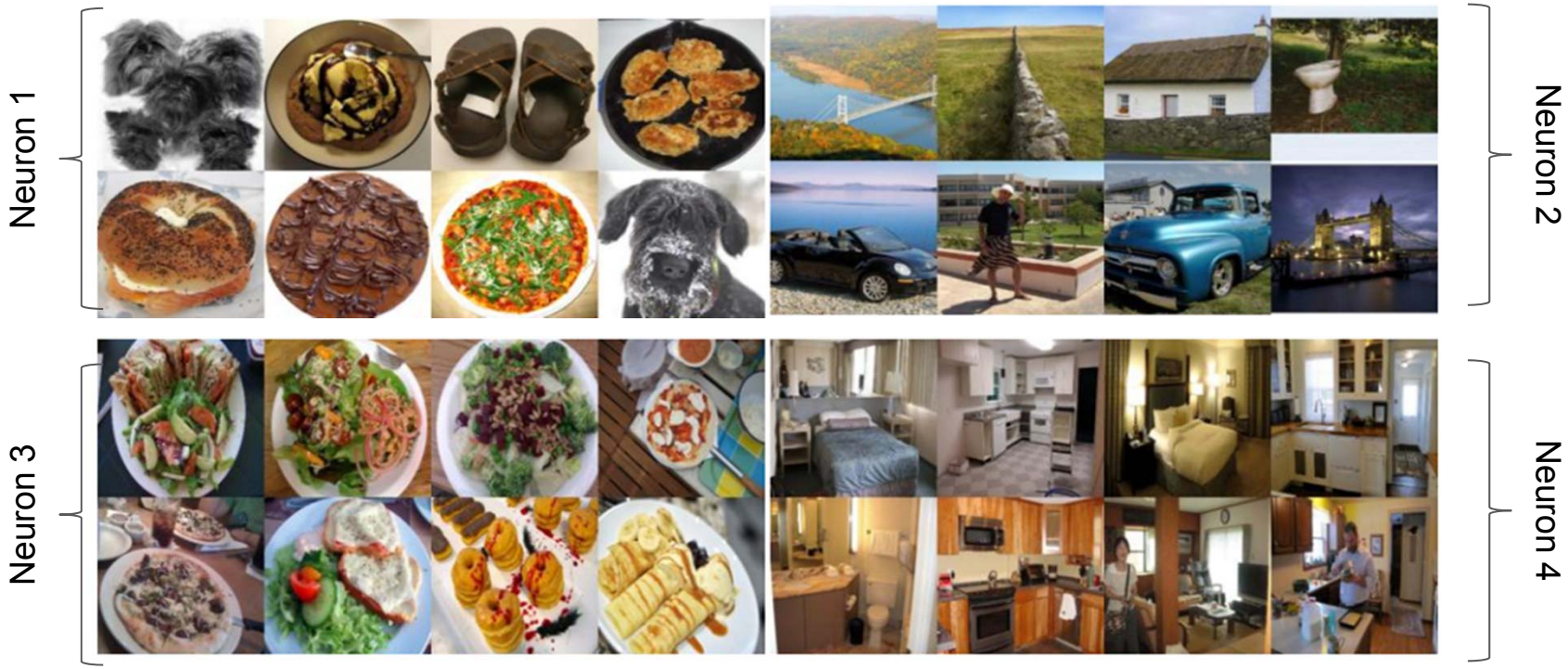
Nearest neighbor retrievals for **COCO**

Experiments



Blocks of the 8 most activating images for 4 neurons for **ImageNet**

Experiments



Blocks of the 8 most activating images for 4 neurons for **COCO**

Thank you!