
Learning with Small Samples

Including zero-shot learning

Nour Karessli
DSR 2019

Nour Karessli

Data Scientist, Zalando SE
Computer vision engineer, EyeEm GmbH

Master in computer science, Saarland University - Max Planck Institute
Bachelor in software engineering, Damascus University

nour.karessli@gmail.com
[LinkedIn](#)

About this tutorial

- A basic understanding of zero-shot and low-shot learning
- Get to know state of the art approaches
- Hands-on experience with zero-shot image classification
- Hands-on experience with training image classifier with small set



Prerequisites

- Basic math e.g. matrix operations, derivatives,.., etc
- Basic understanding of ML concepts e.g. classifier, loss function,.., etc.
- Basic DL concepts e.g. MLP, CNN, LSTM, ..etc
- Python, Keras

Structure

- Introduction & motivation
- Zero-shot learning
 - Definition
 - Side information
 - Zero-shot learning models
 - Exercise
- Low-shot learning
 - Definition
 - Low-shot learning models
- Tips & tricks
- Exercises

Structure

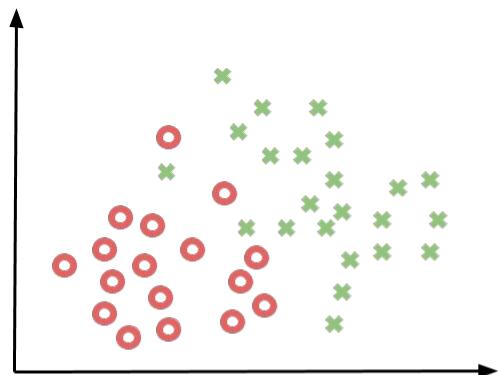
- Introduction & motivation
- Zero-shot learning
 - Definition
 - Side information
 - Zero-shot learning models
 - Exercise
- Low-shot learning
 - Definition
 - Low-shot learning models
- Tips & tricks
- Exercises



Introduction & Motivation

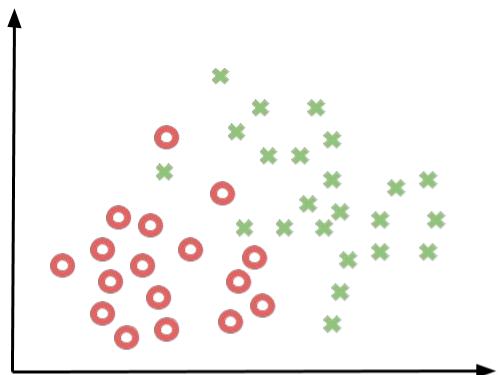
Learning

Supervised

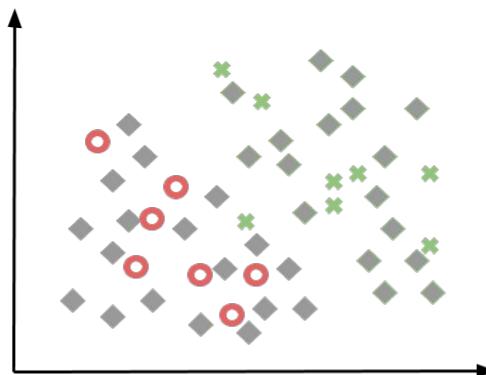


Learning

Supervised

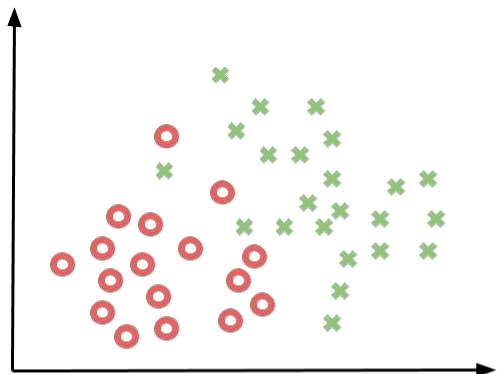


Semi-supervised

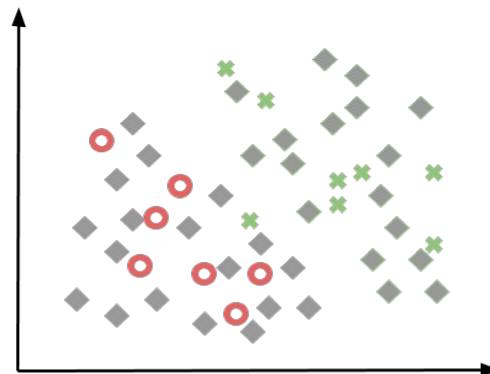


Learning

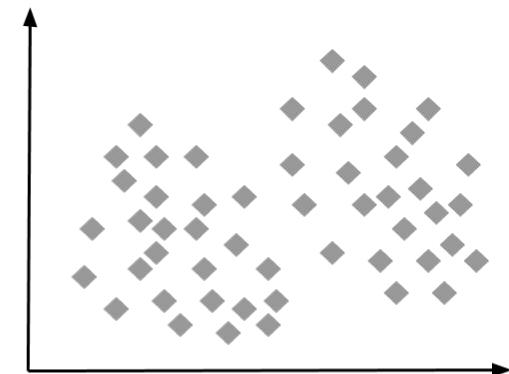
Supervised



Semi-supervised

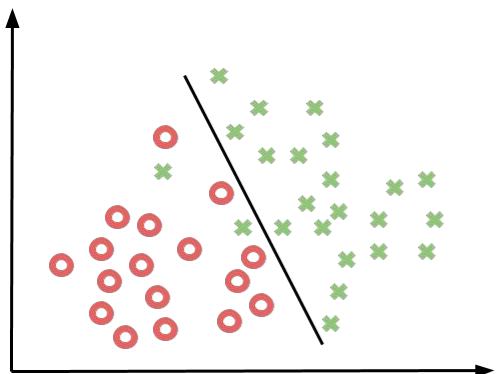


Unsupervised

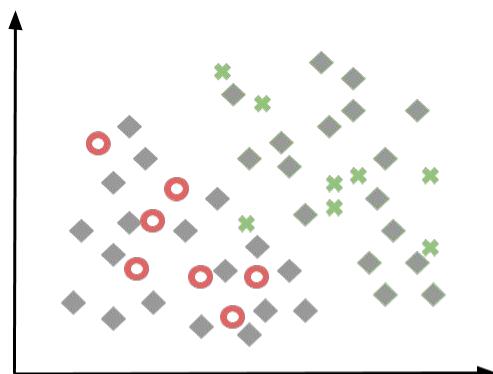


Learning

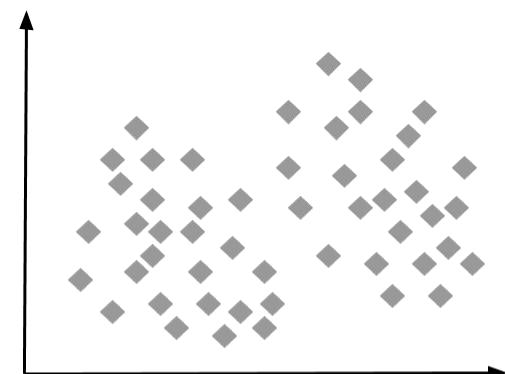
Supervised



Semi-supervised

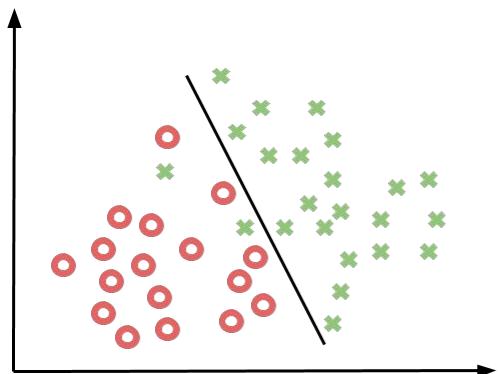


Unsupervised

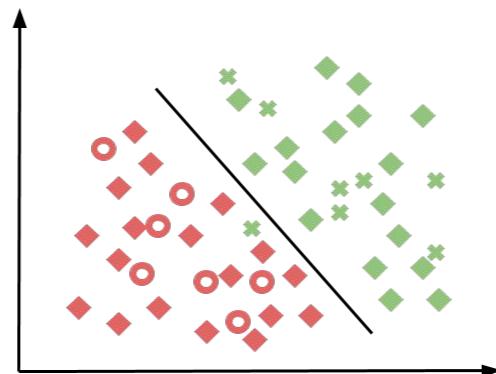


Learning

Supervised



Semi-supervised

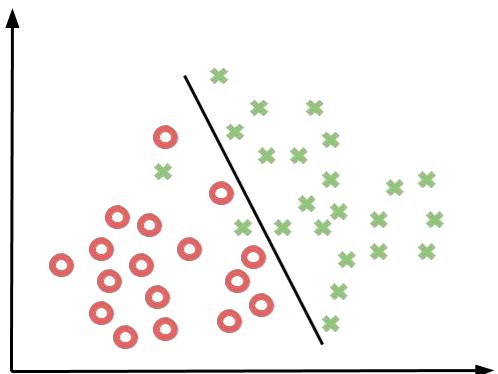


Unsupervised

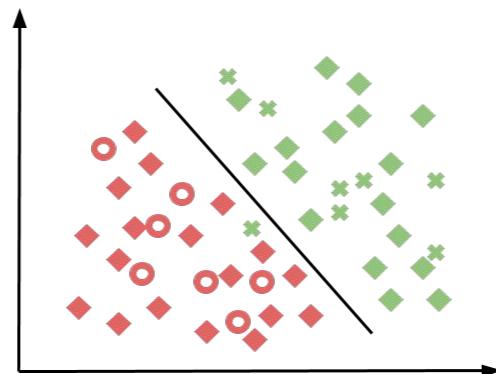


Learning

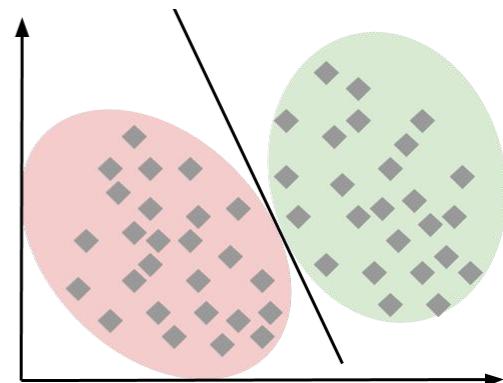
Supervised



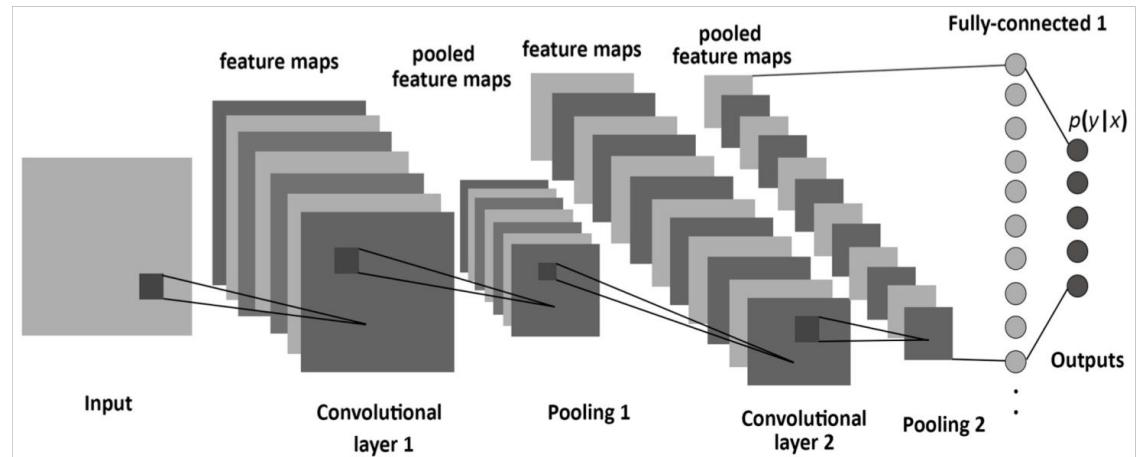
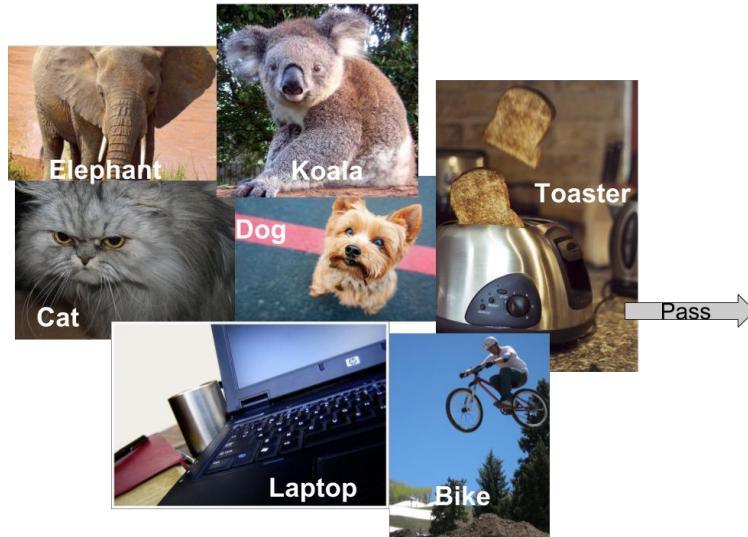
Semi-supervised



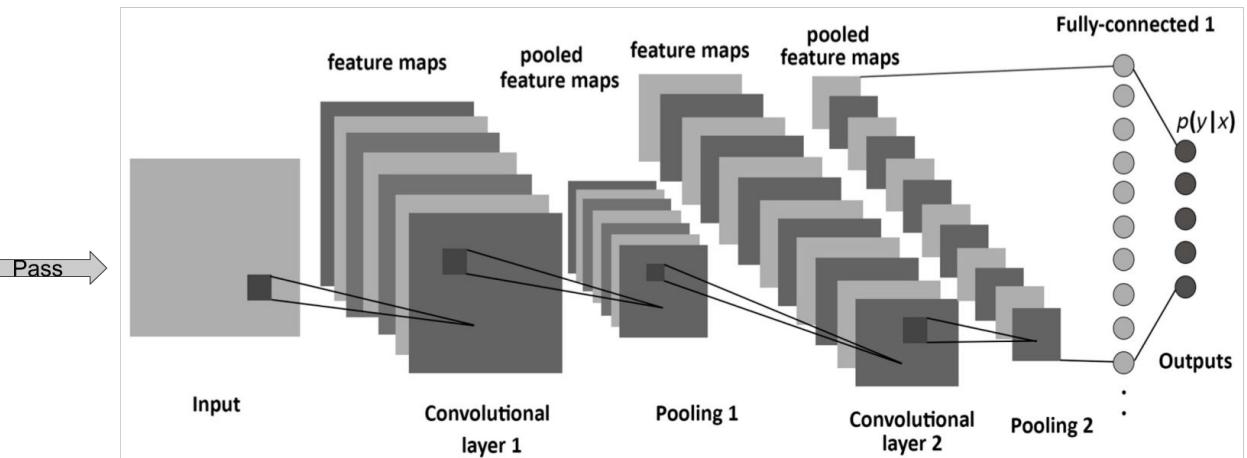
Unsupervised



Many-shots supervised learning



Many-shots supervised learning



Annotation Effort

IMAGENET



14M images, 21K categories

[Deng, et al CVPR2009]

Annotation Effort

- OpenImages
9M images, 6K categories
- COCO
330K images
- MIT Places
2.5M images



Complexity

Task complexity

- Classification

Single labeling is relatively easy task



Complexity ↓

Task complexity

- Classification
Single labeling is relatively easy
- Localization
Requires precise bounding boxes



Task complexity

- Classification

Single labeling is relatively easy

- Localization

Requires precise bounding boxes

- Captioning

Many other possibilities!

“Surprised capybara looking at the camera”

“Portrait photo of a capybara”



Task complexity vs. annotation

Complexity ↓

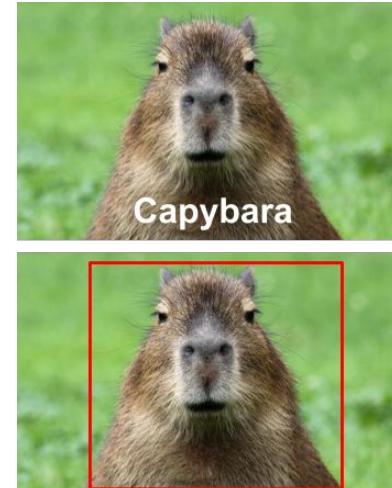
- Classification
ImageNet 14M labeled images



Complexity ↓

Task complexity vs. annotation

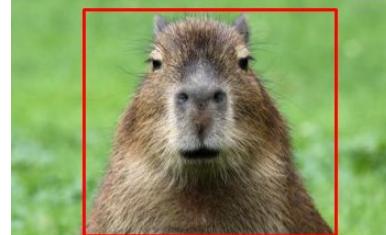
- Classification
ImageNet 14M labeled images
- Localization
openImages 3.6M bounding boxes



Complexity ↓

Task complexity vs. annotation

- Classification
ImageNet 14M labeled images
- Localization
openImages 3.6M bounding boxes
- Captioning
COCO 300K captioned images

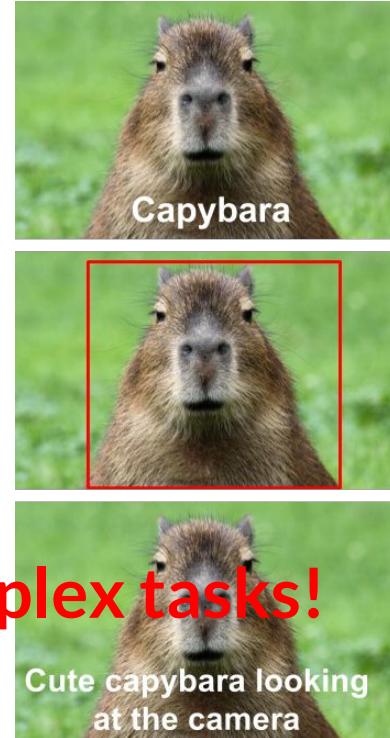


Complexity ↓

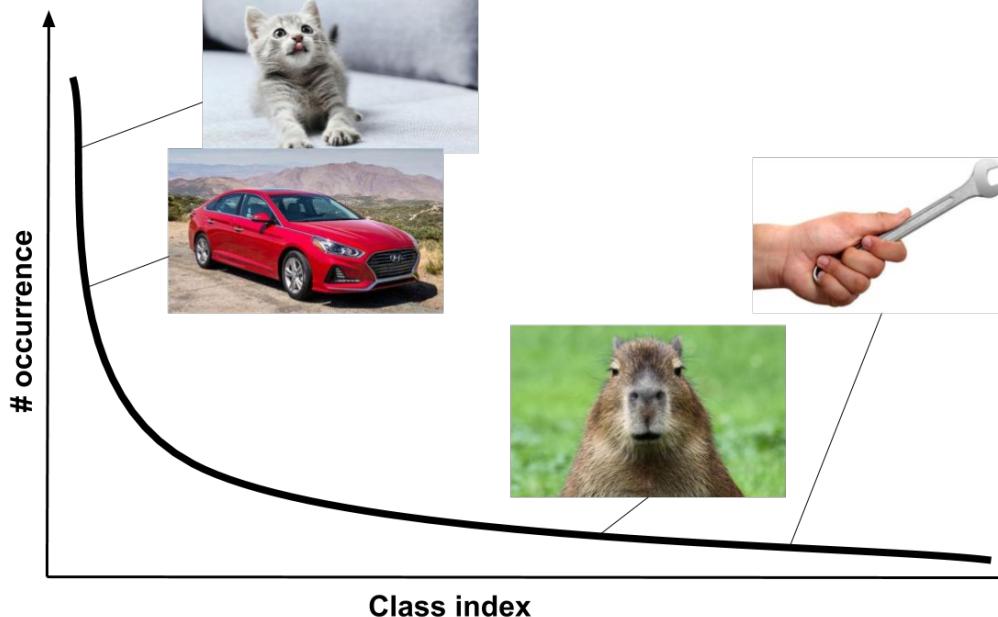
Task complexity vs. annotation

- Classification
ImageNet 14M labeled images
- Localization
openImages 3.6M bounding boxes
- Captioning
COCO 300K captioned images

Less annotations for more complex tasks!



Tail distribution



<https://www.cars.com/>

<http://www.foxnews.com/lifestyle/2017/11/09/how-to-keep-cat-from-scratching-your-sofa-to-shreds.html>

<https://www.livescience.com/55223-capybara-facts.html>

<https://www.indiamart.com/proddetail/hand-wrench-13045857897.html>

Fine-grained categories

Hard: subtle differences



Oxford Pet dataset

Fine-grained categories

Hard: subtle differences



Cars dataset

Fine-grained categories

Hard: subtle differences



Fine-grained categories

Hard: subtle differences



60 classes of Caltech Birds dataset

Fine-grained categories

Novice annotator



Fine-grained categories

Bird expert → expensive





Zero-shot Learning

Structure

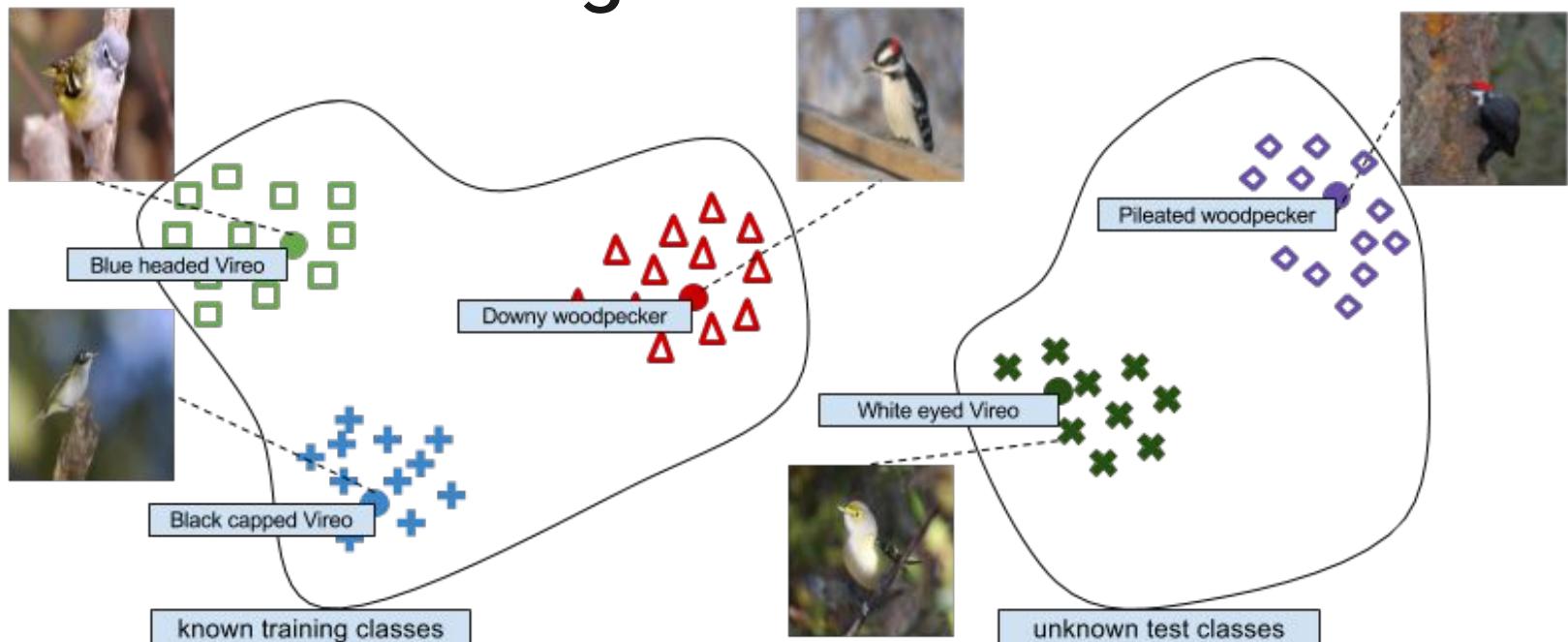
- Introduction & motivation
- Zero-shot learning
 - Definition
 - Side information
 - Zero-shot learning models
 - Exercise
- Low-shot learning
 - Definition
 - Low-shot learning models
- Tips & tricks
- Exercises

Zero-shot learning

- Extreme case of scarce training data for some categories
- Disjoint sets of training and test classes
- New class examples appear only after training stage (at test time)

Animals classifier: No panda images at training time → panda images in testset

Zero-shot learning





Zero-shot learning

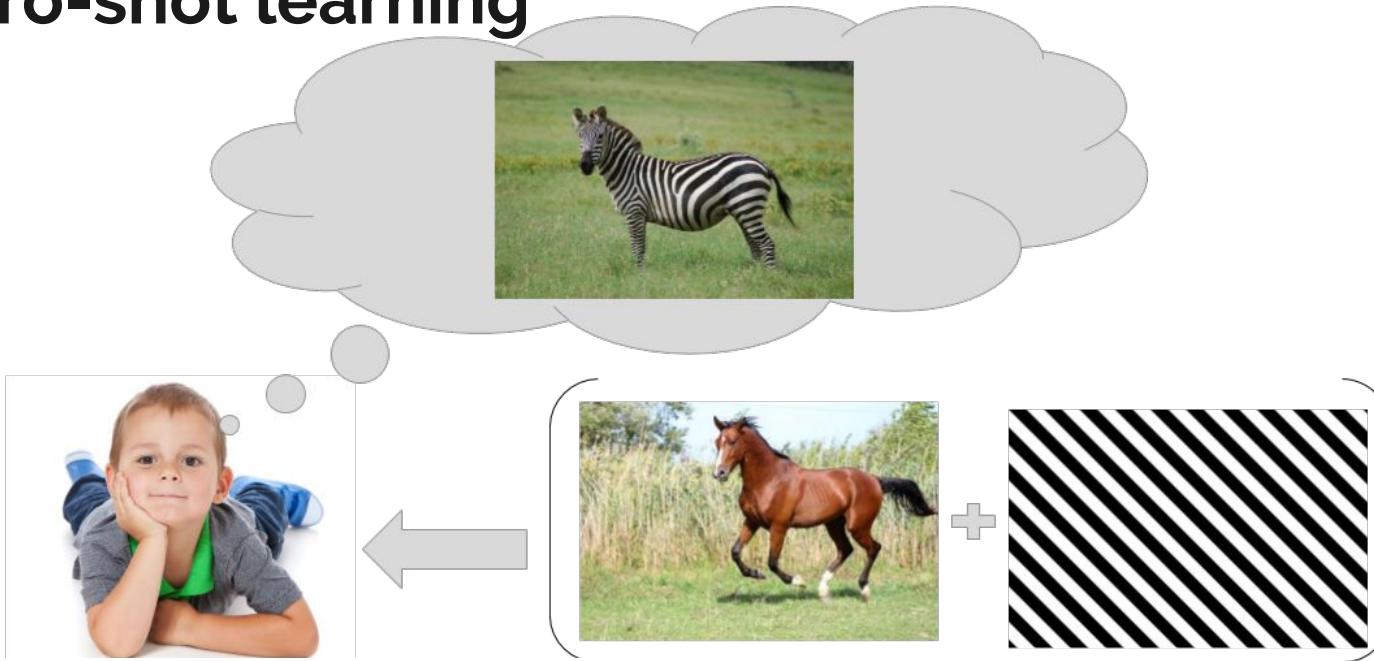
How?

Zero-shot learning



<http://www.youngparents.com.sg/development/10-steps-raising-self-confident-child/>
<http://www.thehorse.com/articles/33568/5-tips-for-packing-the-pounds-on-performance-horses>
<http://earlylearningtoys.org/stripes/>

Zero-shot learning



<http://www.youngparents.com.sg/development/10-steps-raising-self-confident-child/>

<http://www.thehorse.com/articles/33568/5-tips-for-packing-the-pounds-on-performance-horses>

<http://earlylearningtoys.org/stripes/> <https://pt.wikipedia.org/wiki/Zebra-de-grant>

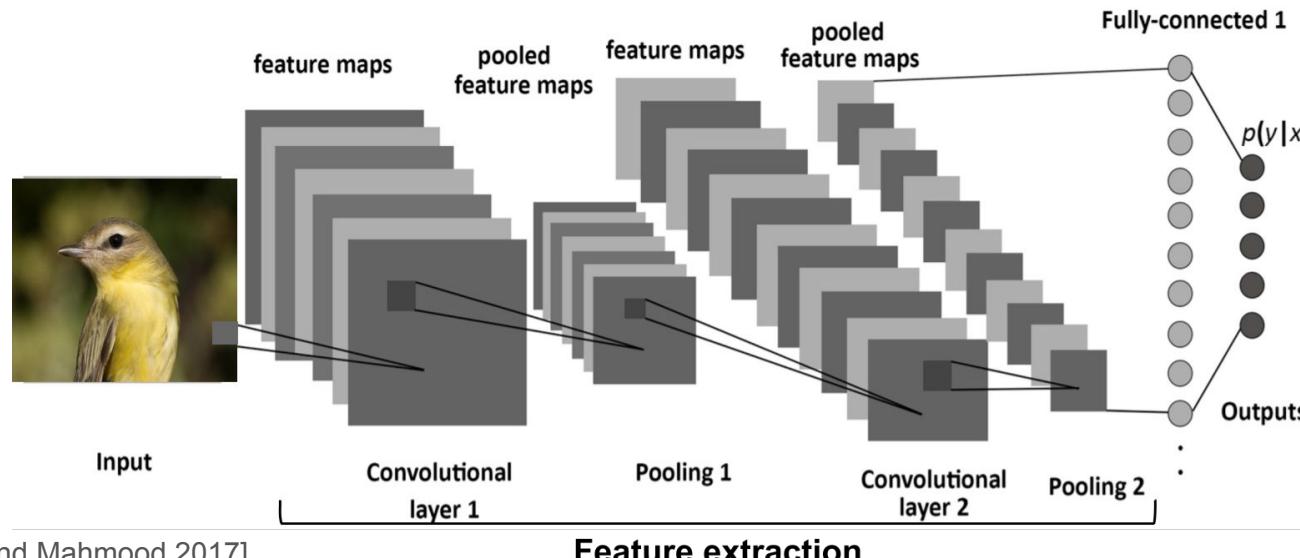


Zero-shot learning

Knowledge transfer & Side information

Knowledge transfer

Use bottleneck image features of pre-trained model





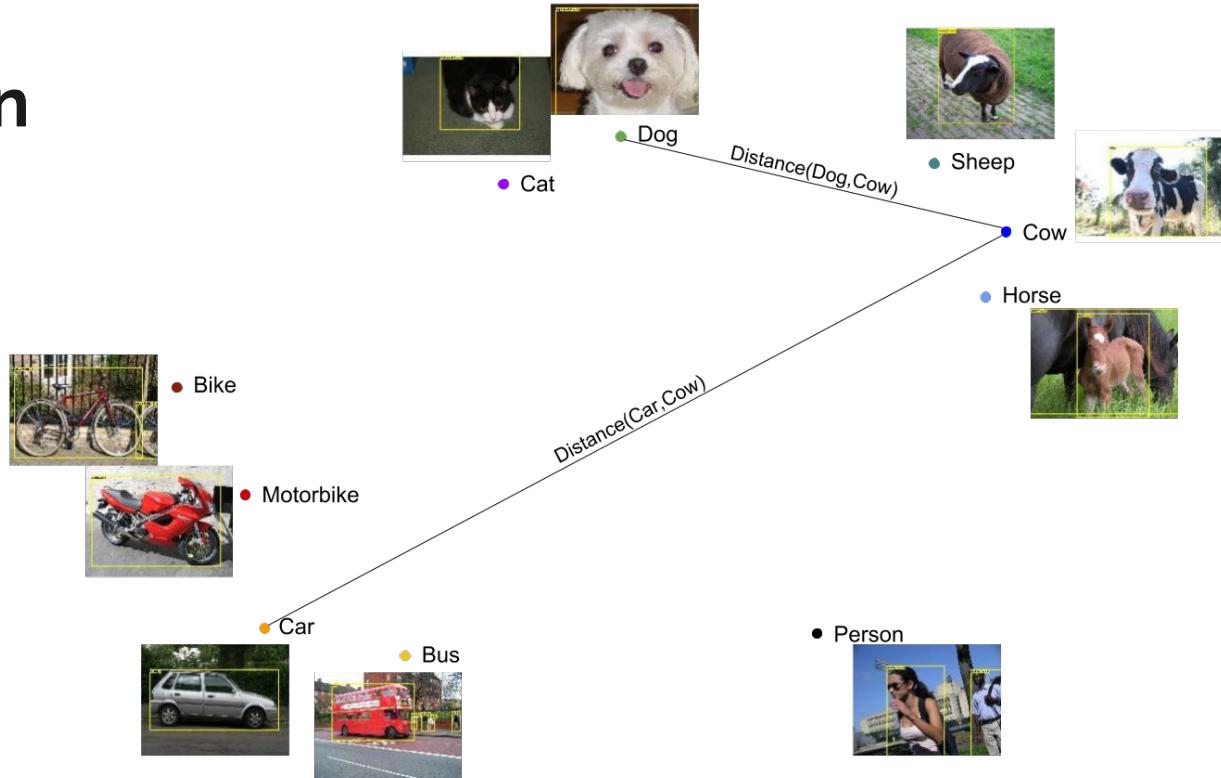
Structure

- Introduction & motivation
- Zero-shot learning
 - Definition
 - Side information
 - Zero-shot learning models
 - Exercise
- Low-shot learning
 - Definition
 - Low-shot learning models
- Tips & tricks
- Exercises

Side information

Use side information to get a better representation for classes (categories).

Labels → Vector representations



Side information - Attributes embedding



Polar bear

black: no
white: yes
brown: no
stripes: no
water: yes
eats fish: yes

[0 1 0 0 1 1]



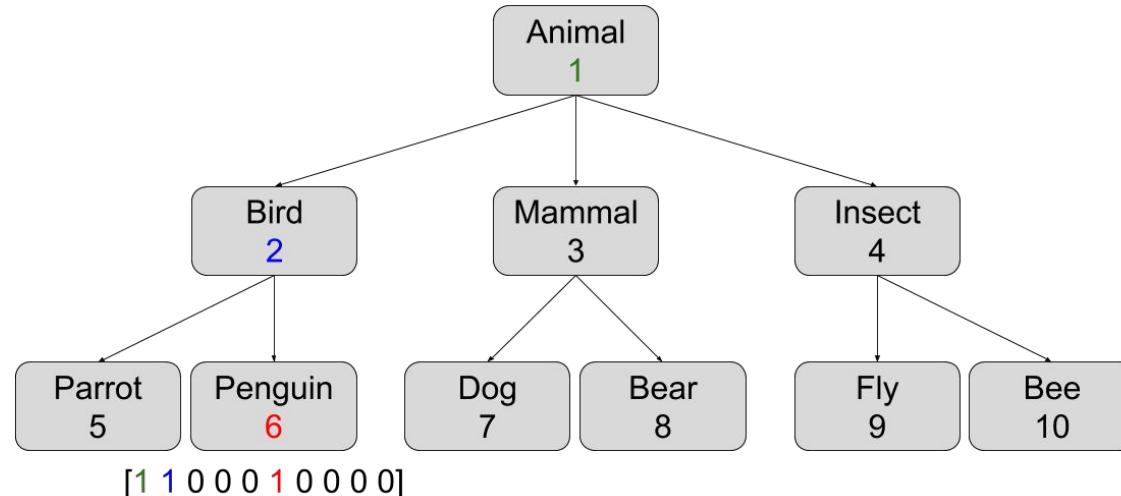
Otter

black: yes
white: no
brown: yes
stripes: no
water: yes
eats fish: yes

[1 0 1 0 1 1]

Side information - Hierarchical embedding

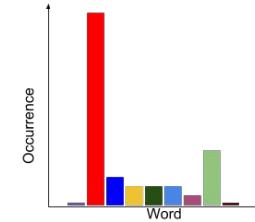
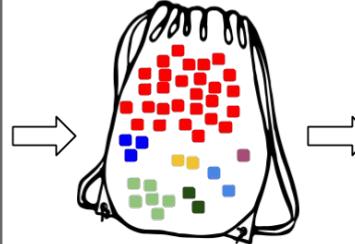
HLE Hierarchical Label Embedding extracted from Wordnet



Side information - Text embedding

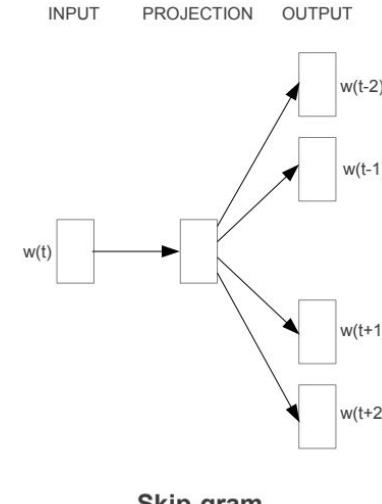
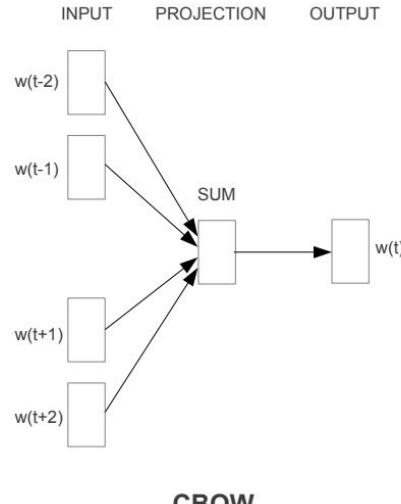
Sparse representations: BoW Bag of Words from Wikipedia articles

The screenshot shows the Wikipedia page for the Red-headed woodpecker (*Melanerpes erythrocephalus*). It includes sections such as 'Contents', 'Taxonomy' (with a photo of a bird perched on a branch), 'Description' (with a photo of a bird's head and body), 'Behavior' (with a photo of a bird on the ground), and 'Conservation'. The page also features a sidebar with a map of the bird's range.



Side information - Text embedding

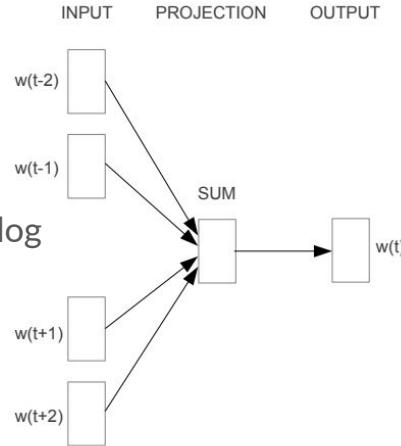
Dense representations: Word2vec



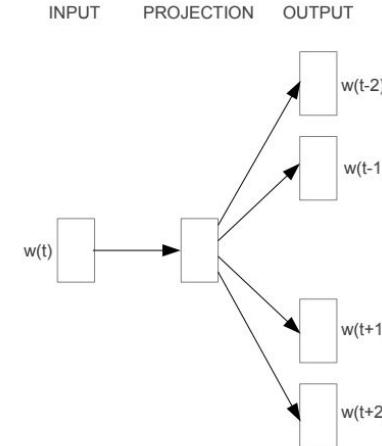
Side information - Text embedding

Dense representations: Word2vec

The fox __?__ over the lazy dog



CBOW



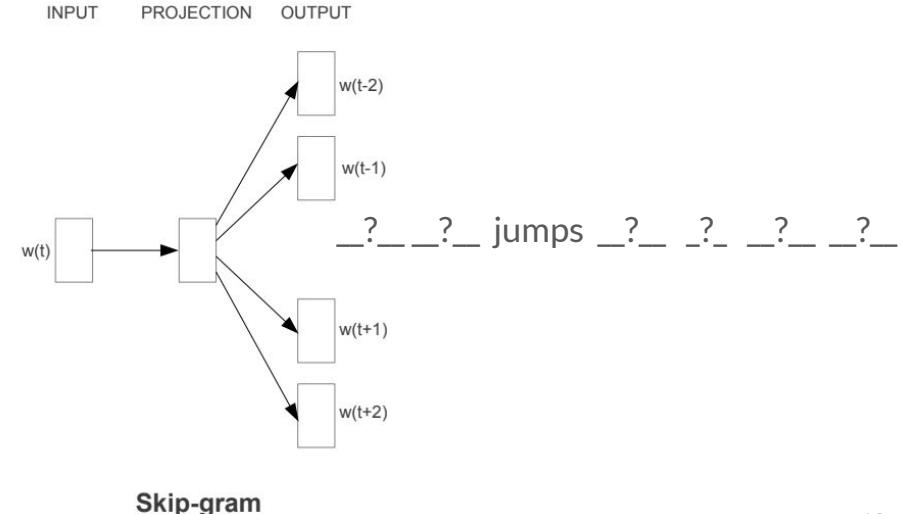
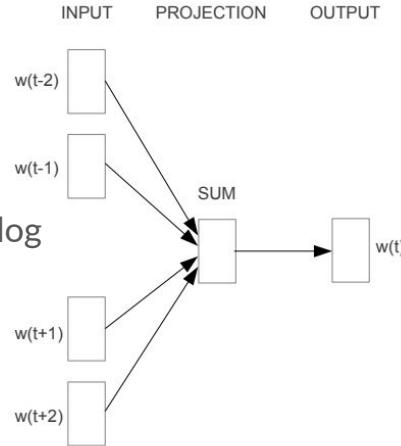
Skip-gram

[Mikolov, et al. NIPS2013]

Side information - Text embedding

Dense representations: Word2vec

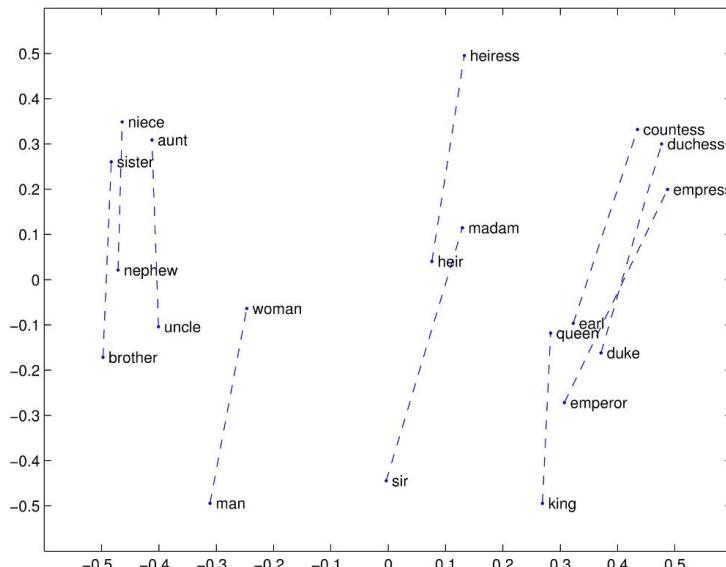
The fox _?_ over the lazy dog



[Mikolov, et al. NIPS2013]

Side information - Text embedding

Dense representations: Glove Global vectors for word representation



Side information - Visual descriptions



The bird has a white underbelly, black feathers in the wings, a large wingspan, and a white beak.



This flower has a central white blossom surrounded by large pointed red petals which are veined and leaflike.



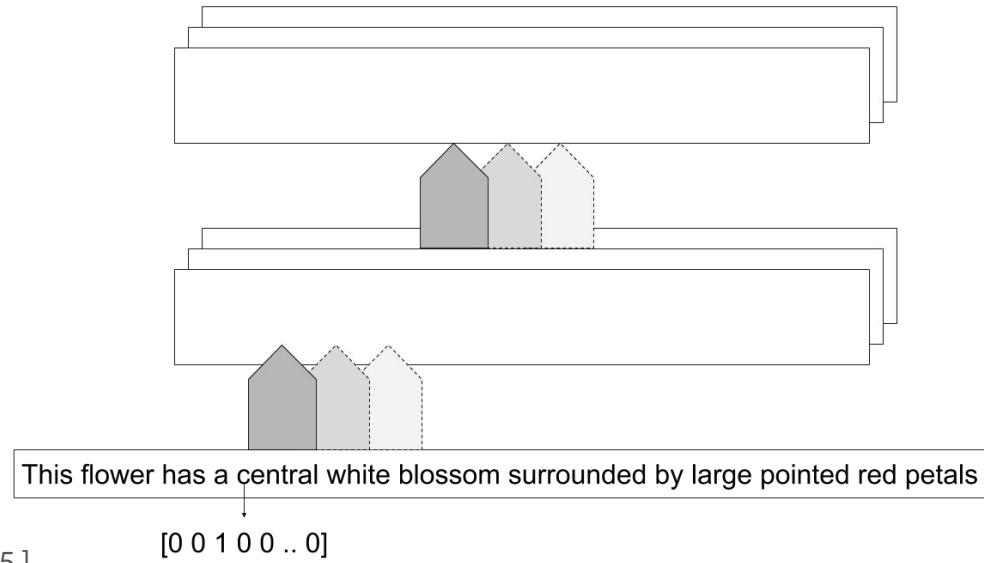
This bird has distinctive-looking brown and white stripes all over its body, and its brown tail sticks up.



Light purple petals with orange and black middle green leaves

Side information - Visual descriptions

- Character level CNN



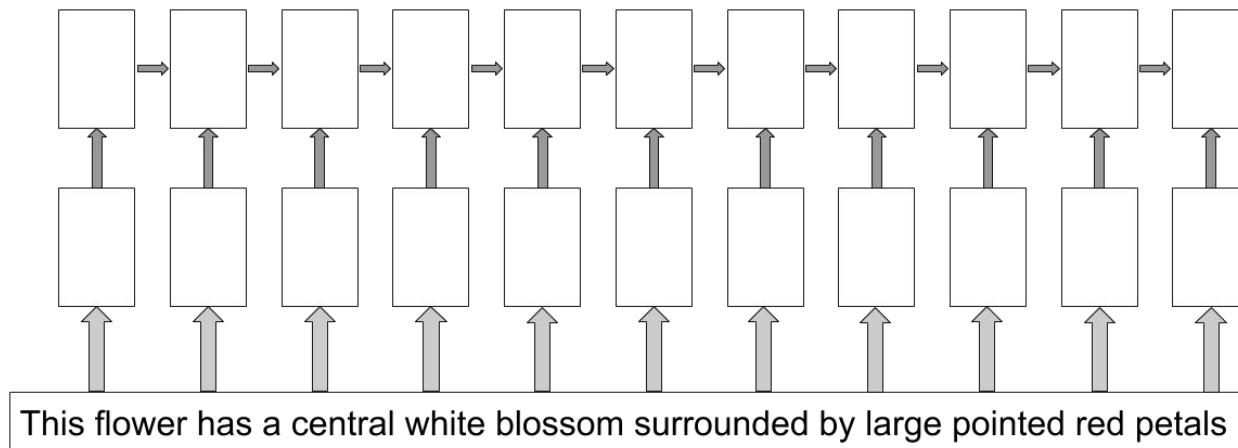
Side information - Visual descriptions

- Word level CNN



Side information - Visual descriptions

- LSTM



Side information - Gaze embedding

- Discrimination of objects by novice
- Data collection is fast
- Implicit annotation,
you don't need to name the object



Side information - Gaze embedding

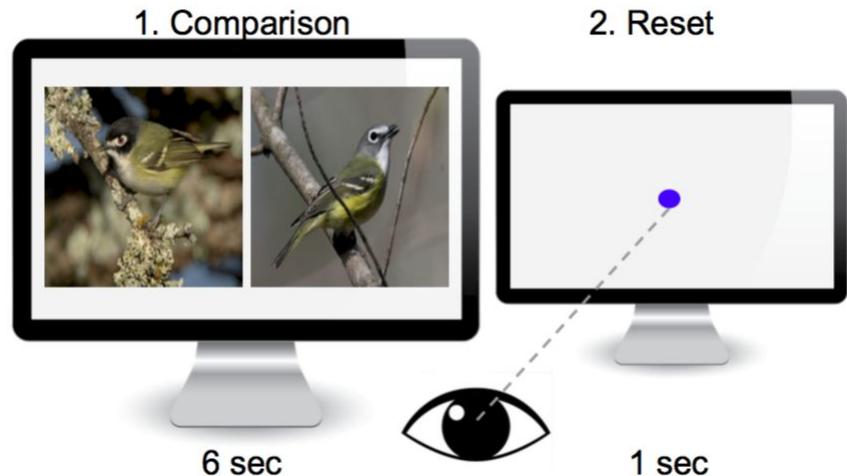
Experiment

1. Comparison



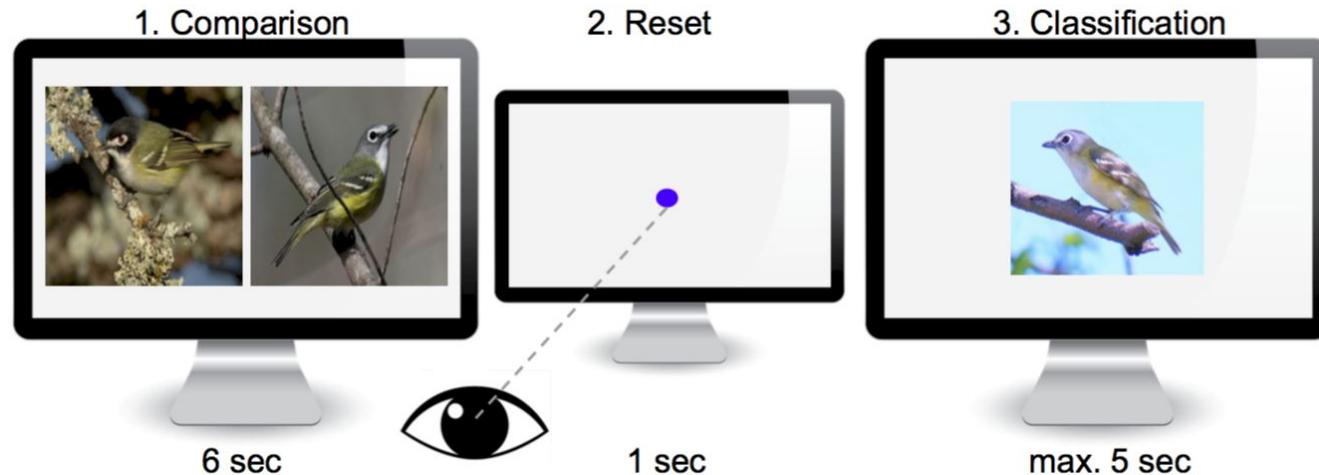
Side information - Gaze embedding

Experiment



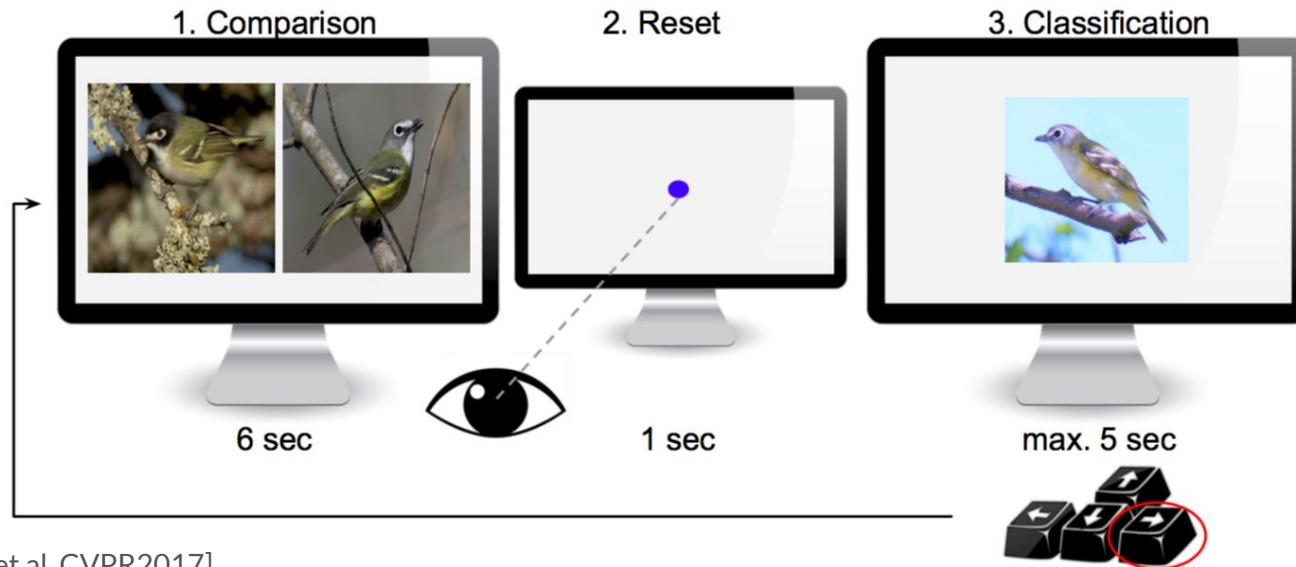
Side information - Gaze embedding

Experiment



Side information - Gaze embedding

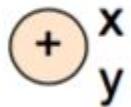
Experiment



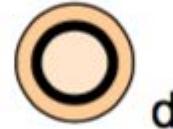
Side information - Gaze embedding

Features

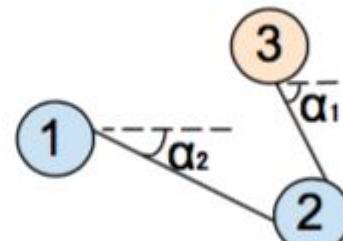
Location



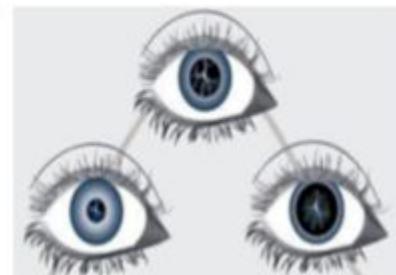
Duration



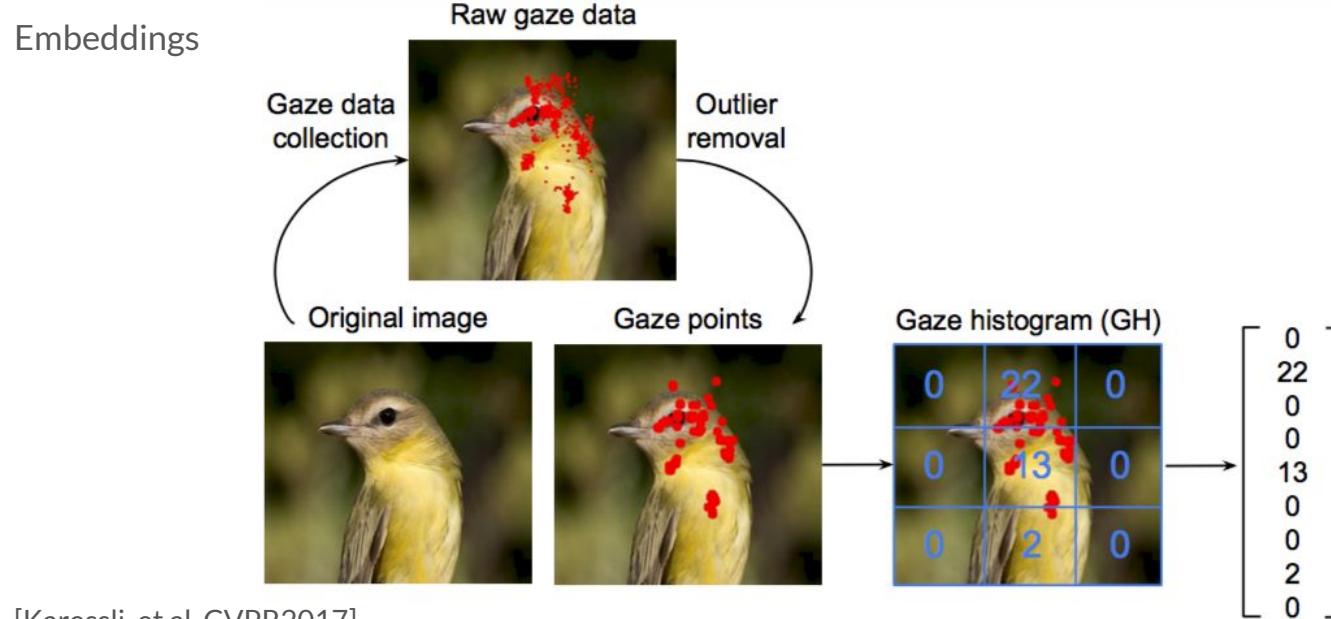
Sequence



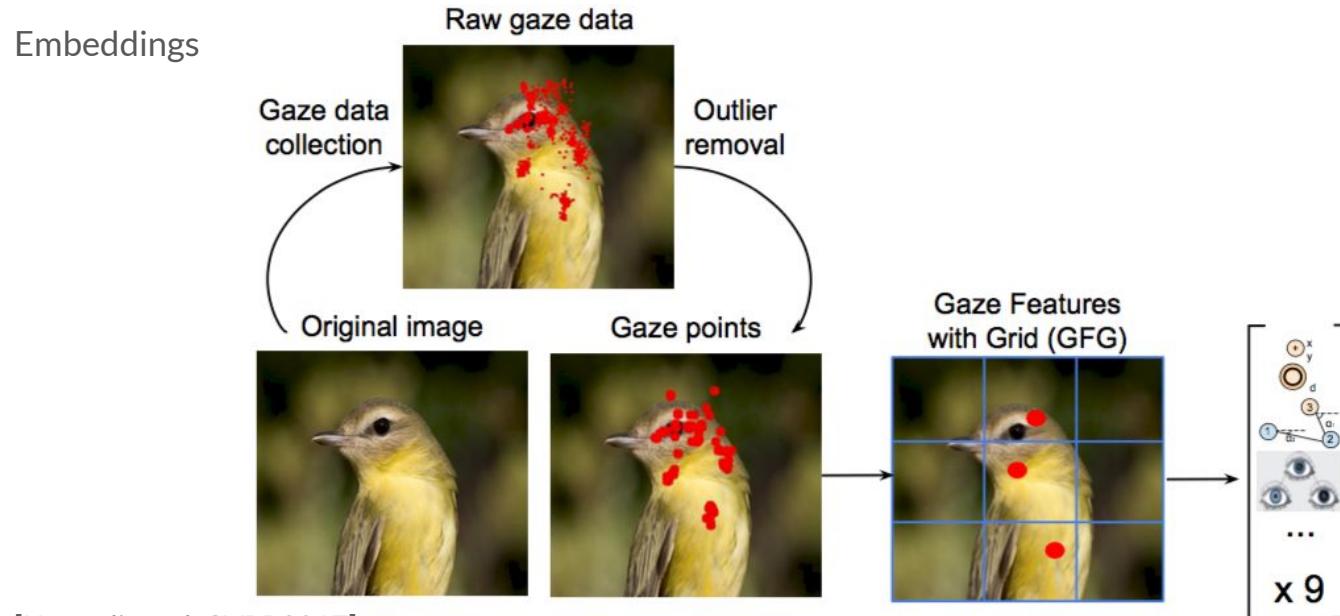
Pupil Diameter



Side information - Gaze embedding

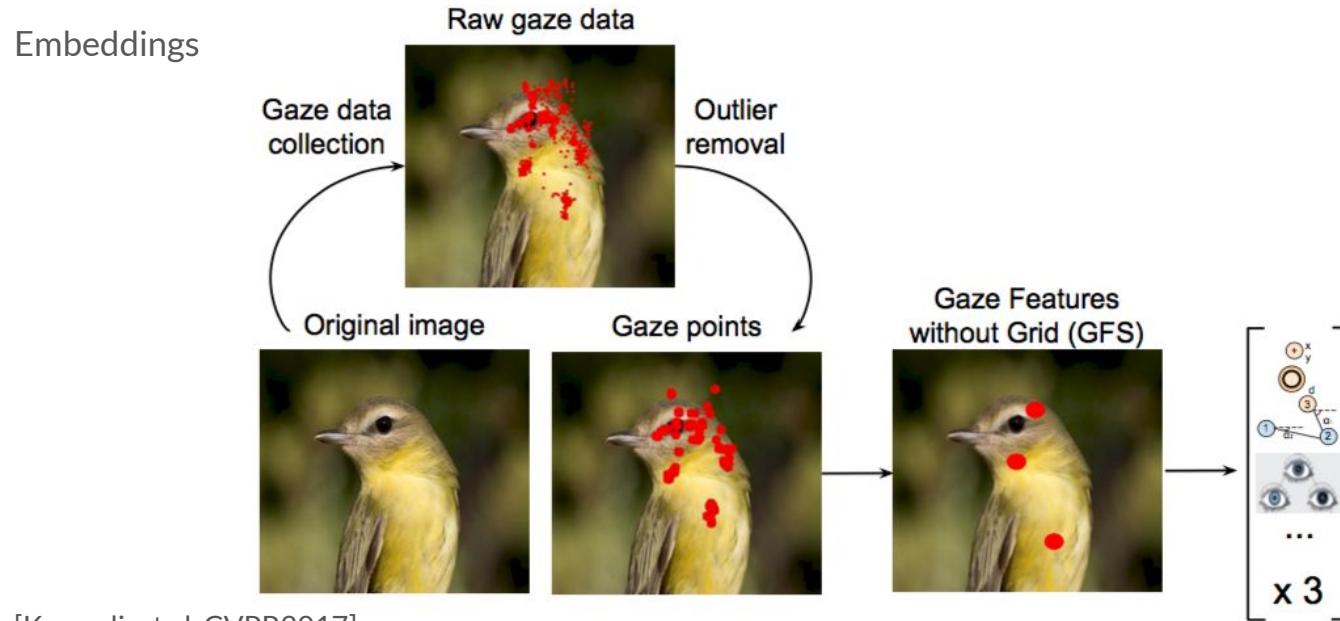


Side information - Gaze embedding

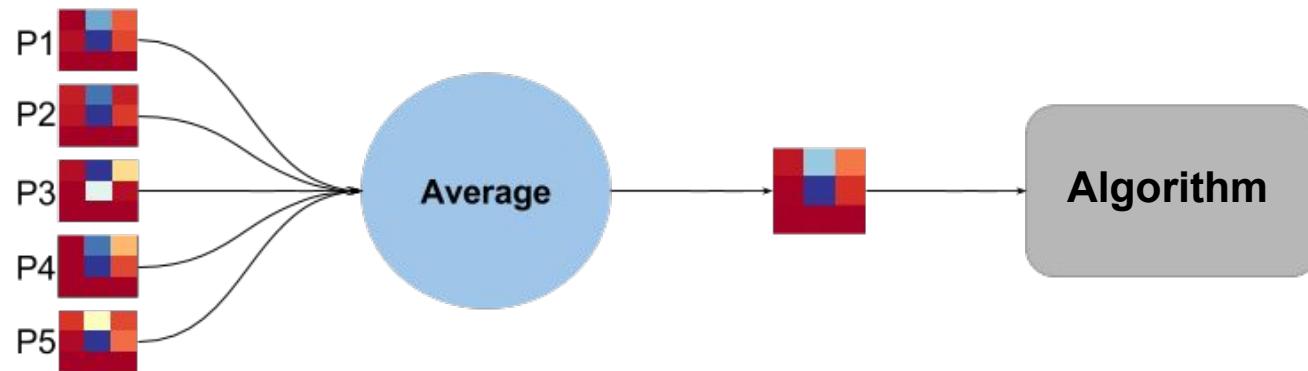


[Karelli, et al. CVPR2017]

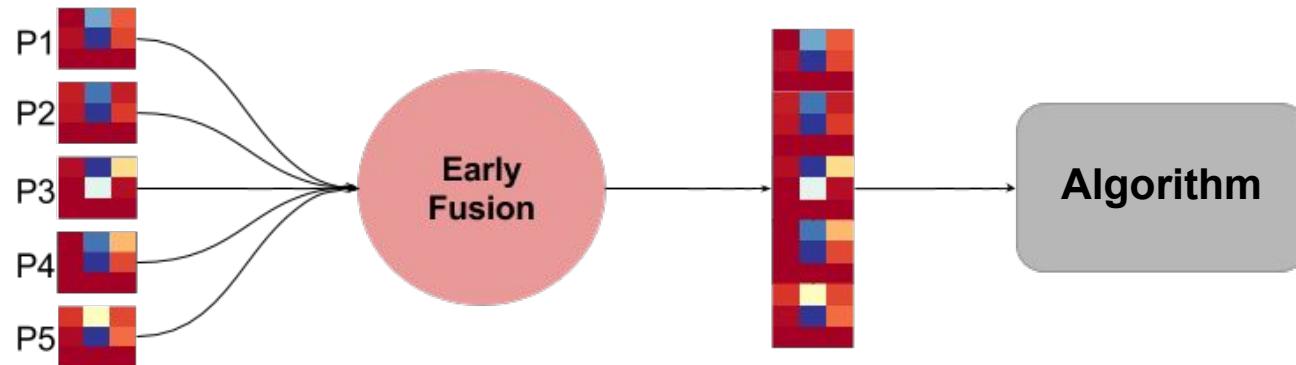
Side information - Gaze embedding



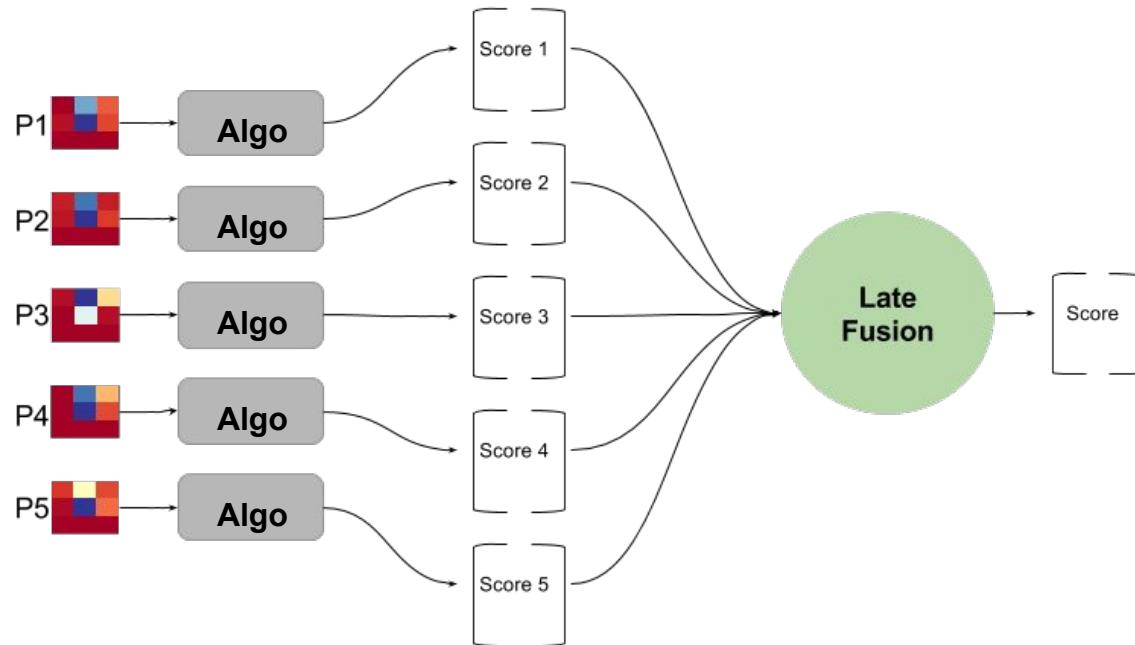
Side information - Gaze embedding



Side information - Gaze embedding



Side information - Gaze embedding



Side information summary

- Supervised
 - Expert
 - Class attributes
 - Novice
 - Detailed visual descriptions (deep representations of visual descriptions)
 - Human gaze
- Unsupervised
 - Hierarchical similarity
 - Text embeddings

CUB birds

- 11,788 images
- 200 bird species, 150 train+val set and 50 test classes



[Welinder, et al. 2010]

Results

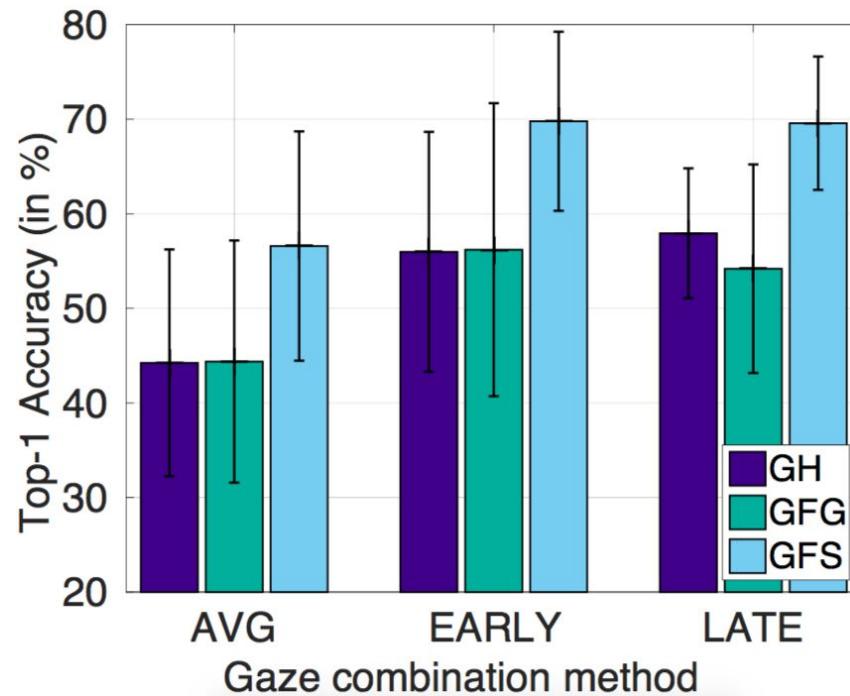
Source	Side information	Accuracy
Text	Hierarchical	20.6
	Bag of Words	22.1
	Word2vec	28.4
	Glove	24.2
Expert annotator	Attributes	50.1
Novice annotator	Detailed visual descriptions	56.8

Mini-CUB birds

- 464 images
- 14 bird species, 11 train+val set and 3 test classes



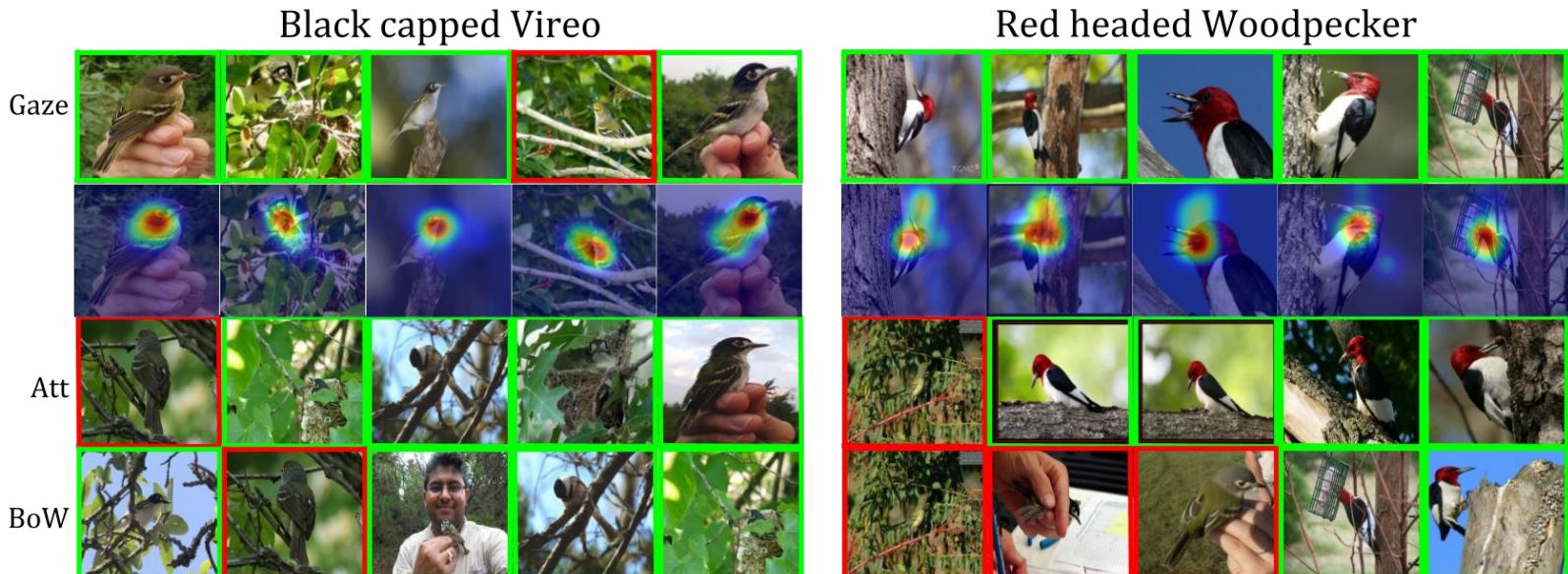
Results



Results

Method	Accuracy
Bag-of-Words from Wiki	55.2
Human annotated attributes	72.9
Gaze embeddings	73.9
Attributes + Gaze	78.2

Results

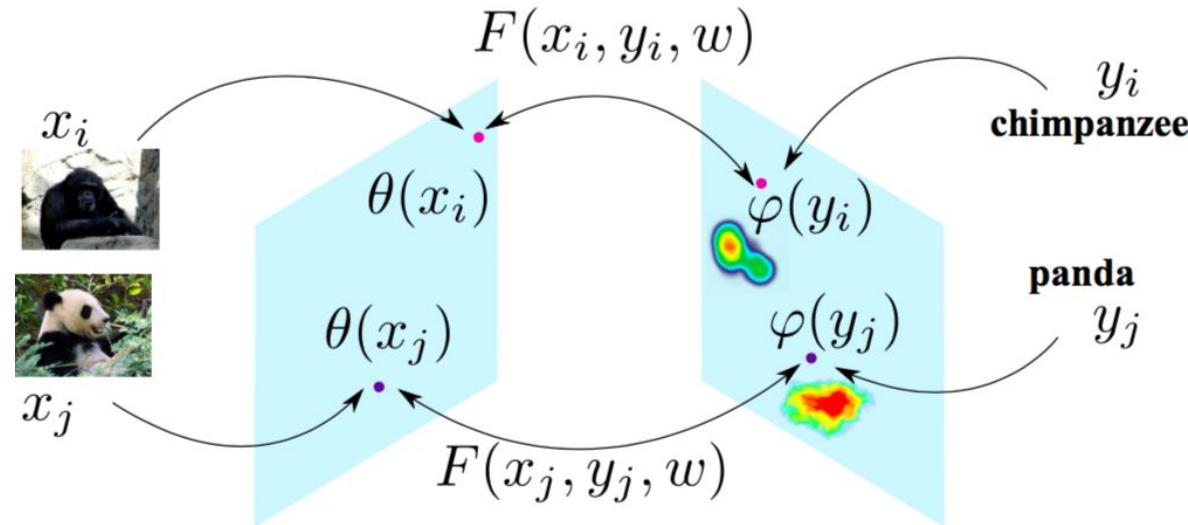




Structure

- Introduction & motivation
- Zero-shot learning
 - Definition
 - Side information
 - Zero-shot learning models
 - Exercise
- Low-shot learning
 - Definition
 - Low-shot learning models
- Tips & tricks
- Exercises

Zero-shot models



Task formulation

Training set $S = \{(x_n, y_n), n = 1..N\}, y_n \in Y^{train}$

We want to learn a function $f : X \rightarrow Y$ by minimize the empirical risk:

$$\frac{1}{N} \sum_{n=1}^N L(y_n, f(x_n; W)) + \Omega(W)$$

- Loss function
- Regularization term

Task formulation

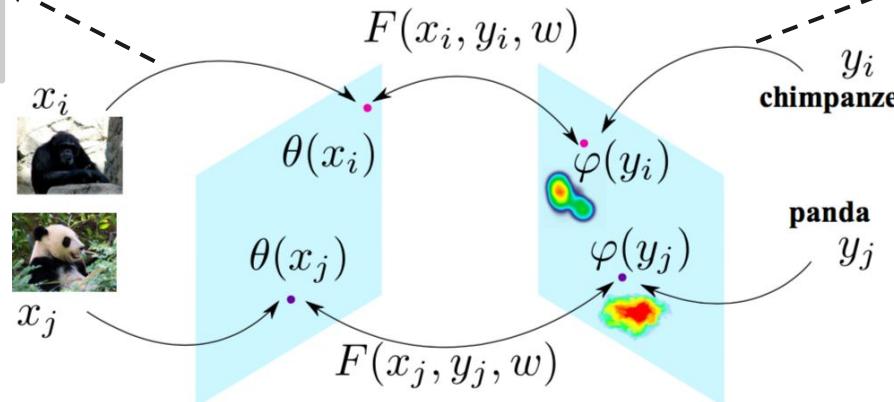
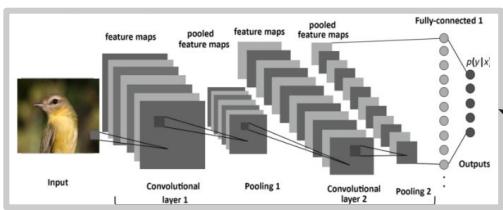
Training set $S = \{(x_n, y_n), n = 1..N\}, y_n \in Y^{train}$

Maximizing the compatibility (score):

$$f(x_n; W) = \operatorname{argmax}_{y \in Y} F(x, y; W)$$

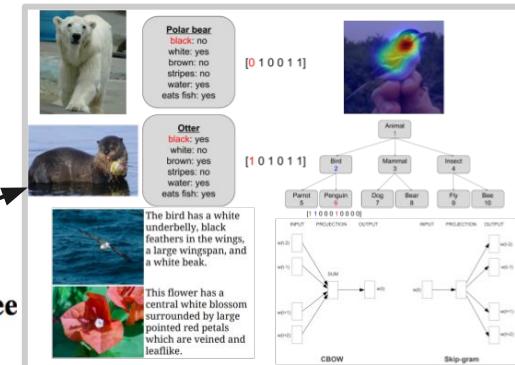
At test time we predict $y \in Y^{test}$ that gives the highest compatibility $Y^{test} \subset Y$

Linear Compatibility



$$F(x, y; W) = \theta(x)^T W \phi(y)$$

[Akata, et al. CVPR2015]



Recap - Gradient Descent

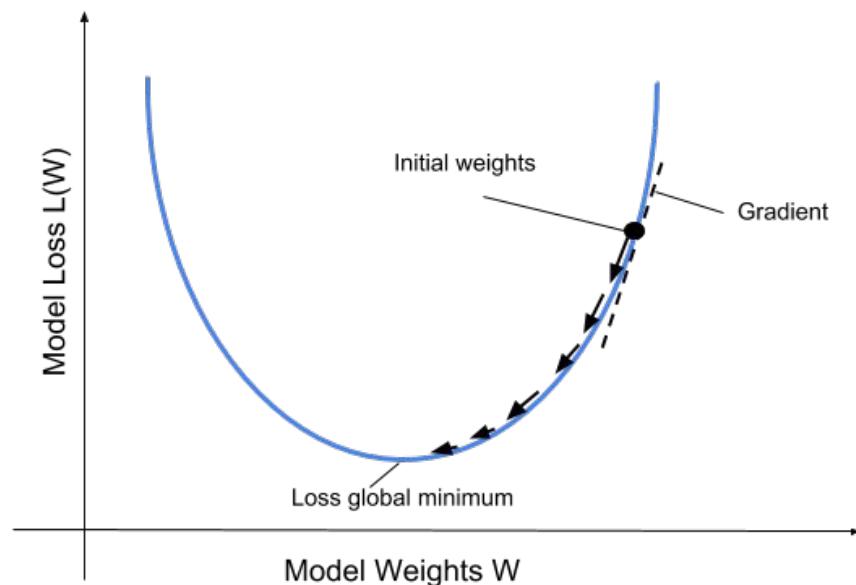
Minimize the loss function by updating the weights with respect to the gradient.

Gradient descent

updates weights only after each epoch

Stochastic gradient descent

updates weights on each sample



Linear Compatibility - DEVISE

Deep visual semantic embedding, pairwise ranking

$$\sum_{y \in \mathcal{Y}^{tr}} [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+$$

$$\Delta(y_n, y) = \begin{cases} 0, & \text{if } y_n = y. \\ 1, & \text{otherwise.} \end{cases}$$

Linear Compatibility - ALE

Attribute label embedding, weighted pairwise ranking

$$\sum_{y \in y^{tr}} l_k [\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+$$

$$l_k = \sum_{i=1}^k \alpha_i \quad \text{where} \quad \alpha_i = 1/i$$

Linear Compatibility - SJE

Structured joint embedding, multiclass objective

$$[\max_{y \in \mathcal{Y}^{tr}} (\Delta(y_n, y) + F(x_n, y; W)) - F(x_n, y_n; W)]_+$$



Structure

- Introduction & motivation
- Zero-shot learning
 - Definition
 - Side information
 - Zero-shot learning models
 - Exercise
- Low-shot learning
 - Definition
 - Low-shot learning models
- Tips & tricks
- Exercises



Zero-shot learning exercise



Recap Exercise - Linear Regression & Gradient Descent

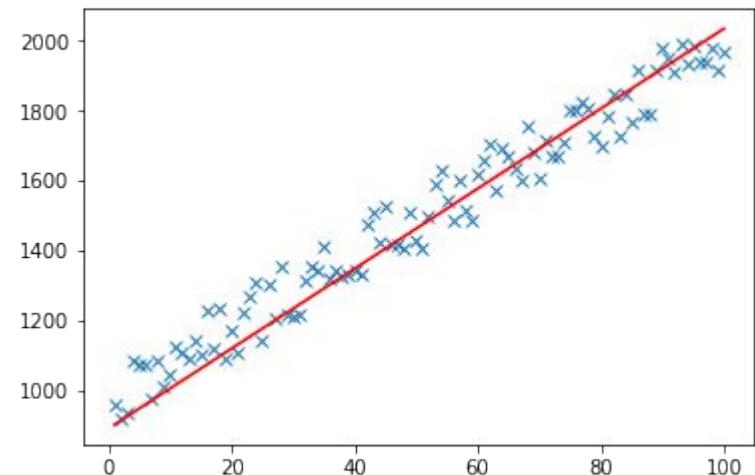
Linear regression - implementation

- Clone [repo](#)
- Generate data for linear regression
- Implement optimisation GD and SGD

Linear regression

$$y = w_0 x_0 + w_1 x_1 ;$$

$$x_0 = 1$$

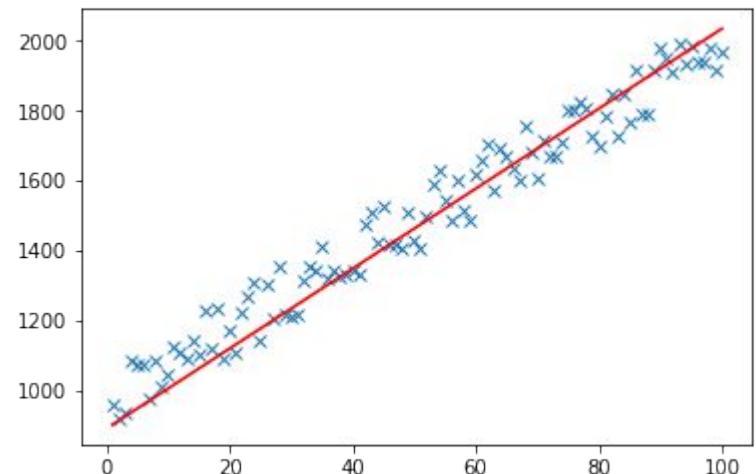


Linear regression with GD - implementation

$$\text{Loss} = \frac{1}{2N} (P-Y)^2$$

foreach epoch

1. Calculate predictions $P = X * W$
2. Calculate error $= (P - Y)^2$
3. Calculate gradient $= X.T * \text{error} / N$
4. Update weights $W = W - lr * \text{gradient}$

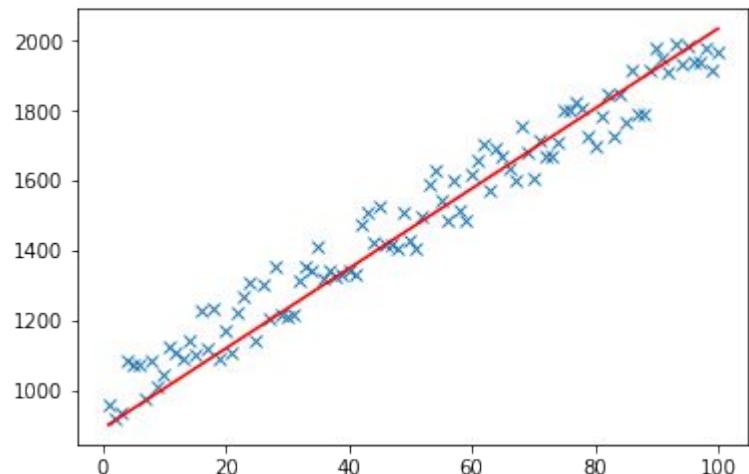


Linear regression with SGD - implementation

$$\text{Loss} = \frac{1}{2N} (P-Y)^2$$

foreach epoch

1. Shuffle training samples
2. **foreach** sample x_i, y_i
 1. Calculate sample prediction $p_i = x_i * W$
 2. Calculate error $_i = (p_i - y_i)^2$
 3. Calculate gradient $_i = x_i * \text{error}_i$
 4. Update weights $W = W - lr * \text{gradient}_i$





Structured Joint Embedding exercise

AWA dataset

Animal with attributes dataset

- 30K images
- 50 classes
- 85 attribute
- Standard split 40 train+val 10 test

SJE implementation

- Select dataset (AWA)
- Download [data](#) (image features & corresponding labels) & use class embeddings in [repo](#)
- Implement zero-shot algorithm SJE
- Evaluate Top-1 accuracy on test set

SJE implementation

Initialize W (DxE)

foreach epoch

 Shuffle training samples

foreach sample (x_i, y_{true})

 score_{true} = $\theta(x_i) * W^T * \phi(y_{true})$

 loss_{max} = -1, y_{max} = -1

foreach training class y_{tr}

 score = $\theta(x_i) * W^T * \phi(y_{tr})$

 loss = $\Delta(y_{tr}, y_{true}) + score - score_{true}$

if loss > loss_{max}

 update loss_{max} and y_{max}

if y_{max} ≠ y_{true}

 W = W - lr * ($\theta(x_i) \otimes (\phi(y_{max}) - \phi(y_{true}))$)

D: x dim, E: y dim

find max score

violation
update weights



More Zero-shot models

Linear Compatibility - ESZSL

Embarrassingly simple zero-shot learning, adds regularization terms

$$\gamma \| W\phi(y) \|^2 + \lambda \| \theta(x)^T W \|^2 + \beta \| W \|^2$$

γ, λ, β regularization parameters

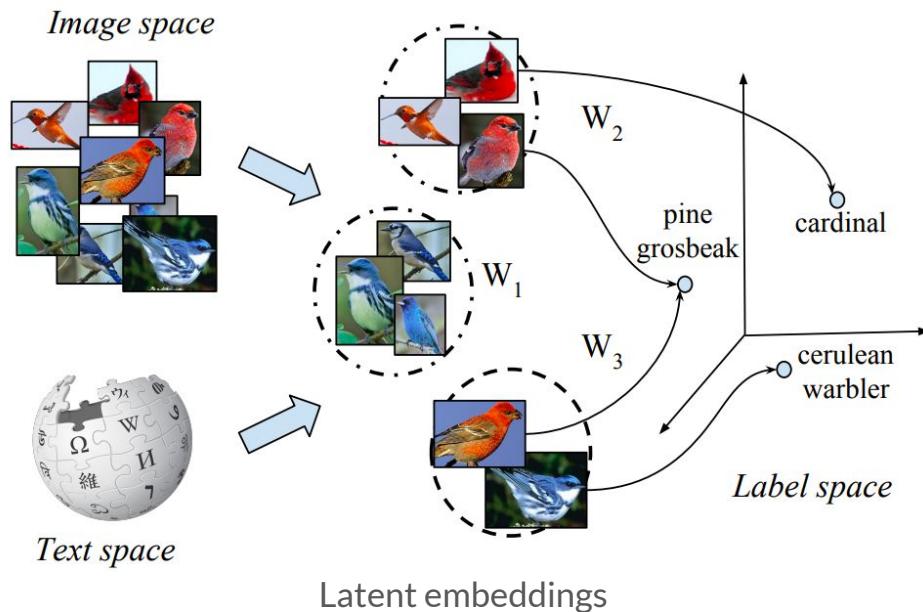
Linear Compatibility - SAE

Semantic autoencoder, linear objective

$$\min_W \| \theta(x) - W^T \phi(y) \|^2 + \lambda \| W\theta(x) - \phi(y) \|^2$$

- Autoencoder learns projection from image features to label embedding
- The autoencoder must reconstruct original image features

Nonlinear Compatibility - LATEM



Nonlinear Compatibility - LATEM

Latent embeddings, piecewise linear compatibility

$$F(x, y; W_i) = \max_{1 \leq i \leq K} \theta(x)^T W_i \phi(y)$$

- Support non-linearity
- Different W encodes different characteristics

Nonlinear Compatibility - CMT

Cross modal transfer, nonlinear compatibility using two layers NN

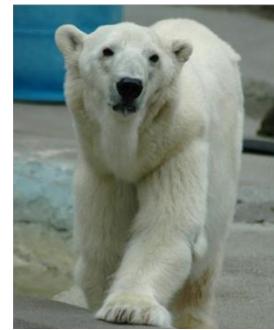
$$\sum_{y \in \mathcal{Y}^{tr}} \sum_{x \in \mathcal{X}_y} \|\phi(y) - W_1 \tanh(W_2 \cdot \theta(x))\|$$

Intermediate Classifier - DAP

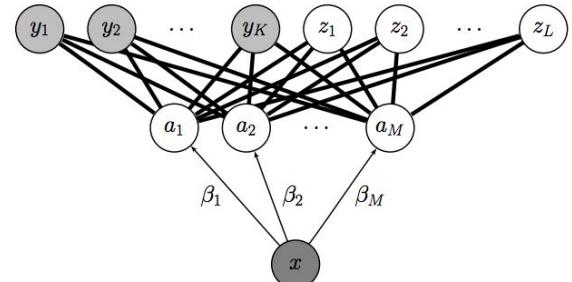
Direct attribute prediction

1. Learning probabilistic classifiers for each attribute
2. Combie scores

$$f(x) = \operatorname{argmax}_c \prod_{m=1}^M \frac{p(a_m^c | x)}{p(a_m^c)}$$



Polar bear
black: no
white: yes
brown: no
stripes: no
water: yes
eats fish: yes



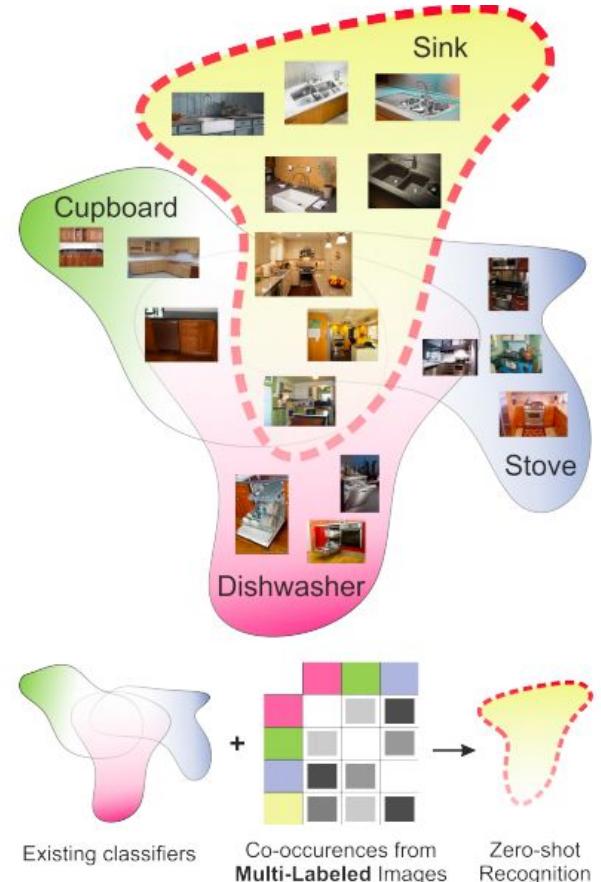
Intermediate Classifier - COSTA

Co-occurrence statistics of visual concepts
between seen and unseen classes

$$\hat{\mathbf{w}}_l = \sum_k \mathbf{w}_k s_{lk}$$

k classifier weights (svm)

Similarity between seen k and unseen l





Models summary

1. Linear compatibility

DEVISE [Frome, et al. NIPS2013], ALE [Akata, et al. CVPR2015], SJE [Akata, et al. CVPR2015],
ESZSL [Romera-Paredes and Torr ICML2015], SAE [Kodirov, et al. CVPR2017]

2. Nonlinear compatibility

LATEM [Xian, et al. CVPR2016], CMT [Socher, et al. NIPS2013]

3. Intermediate classifier

DAP [Lampert, et al. CVPR2009], COSTA [Mensink, et al. CVPR2014]



AWA dataset

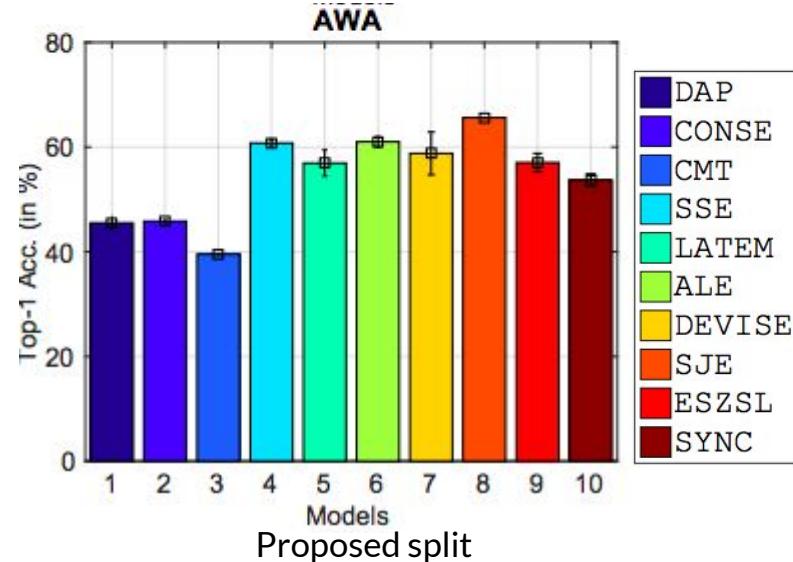
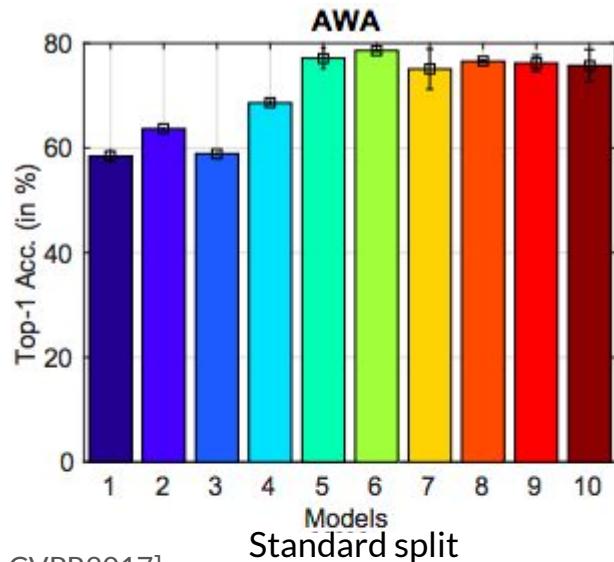
Animal with attributes dataset

- 30K images
- 50 classes
- 85 attribute
- Standard split 40 train+val 10 test
- Suggested split

Insures that none of the test classes is used to train the image features base model

Results

Evaluation on Animals with Attributes AWA





Q & A