Learning from Limited Data, Open-Set and Open-World learning

Saikat Dutta

PhD scholar

IITB-Monash Research Academy

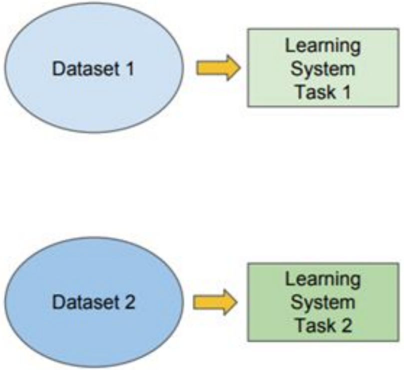
Table of contents

Transfer Learning

Traditional ML vs Transfer Learning

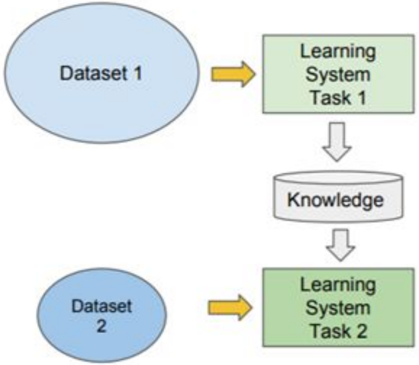
Traditional ML

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



Transfer Learning

- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data



Transfer Learning

- **Inductive Transfer Learning:**
  - Source and target domains are same
  - The specific tasks the model is working on are different
  - e.g. ImageNet -> Oxford-IIIT pets dataset

- **Transductive Transfer Learning:**
  - Domains of Source and target tasks are not exactly the same but interrelated
  - e.g. Real images of cats and dogs -> animated images

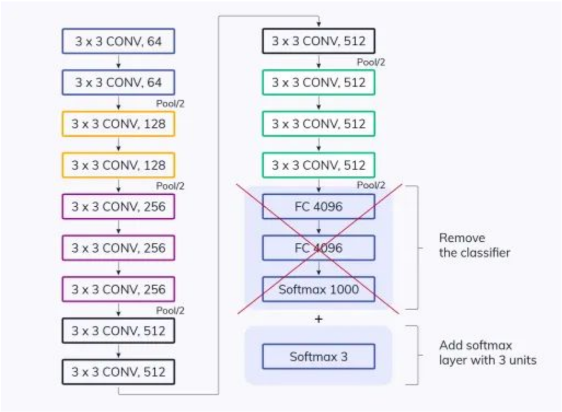
Transfer Learning

- **Unsupervised Transfer Learning**
  - Similar to Inductive Transfer learning.
  - Focuses on unsupervised tasks and involve unlabeled datasets both in the source and target tasks.

Transfer Learning

Obtain the pre-trained model → Create a base model → Freeze layers → Train the new layers on the dataset

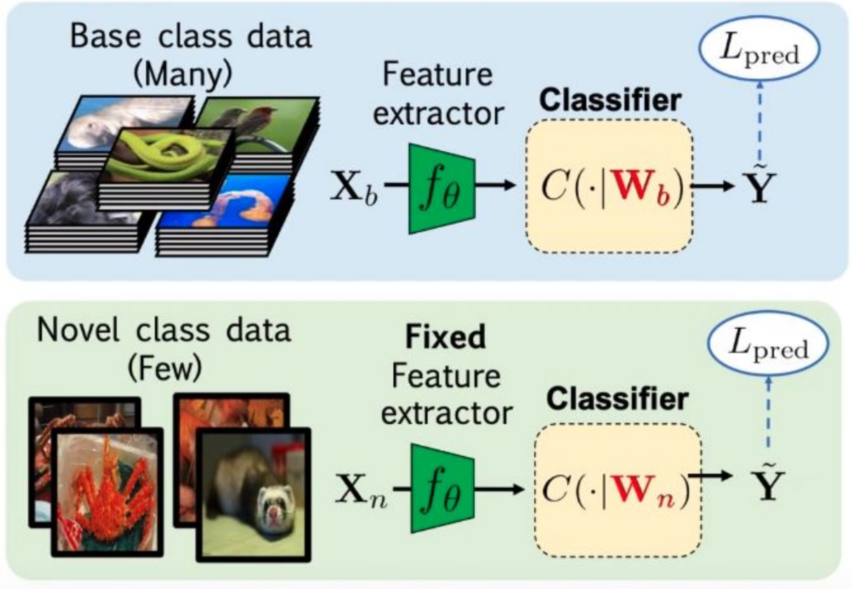
ImageNet pretrained
ResNet-50



Freeze initial layers
(optional)

Few-shot Learning

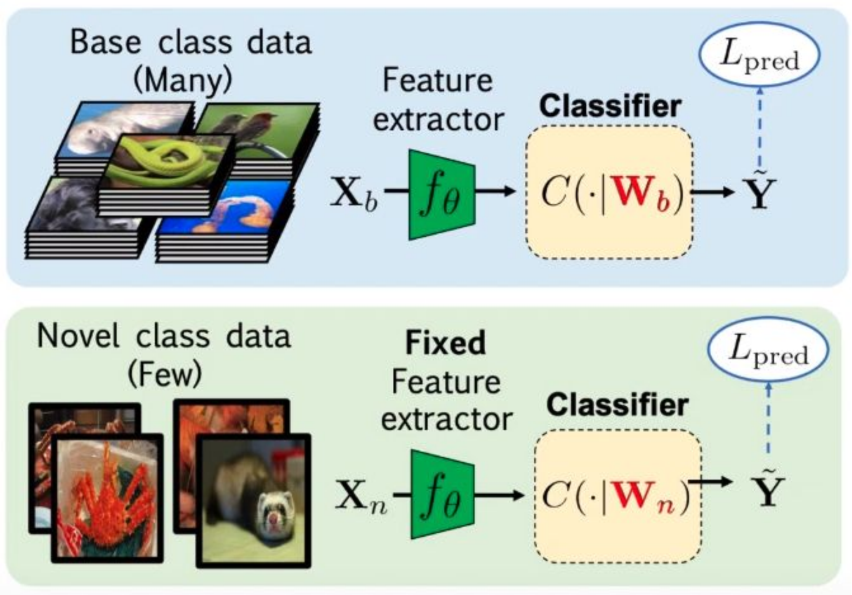
- N-way-K-shot classification approach: discriminate between N classes with K examples.
- Base class data available.
- Different variations possible:
  - One-shot
  - 5-shot etc.



Chen et al., ICLR 19

Few-shot Learning

- N-way-K-shot classification approach: discriminate between N classes with K examples.
- Base class data available.
- Different variations possible:
  - One-shot
  - 5-shot etc.



- *Q: Can we use transfer learning?*

Chen et al., ICLR 19

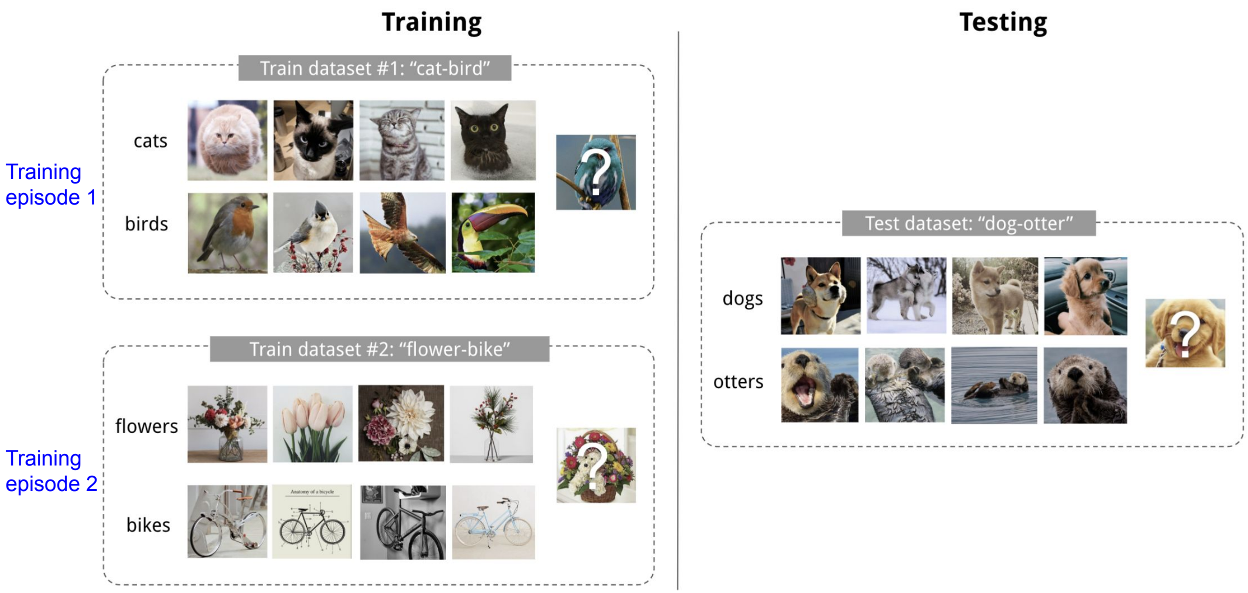
Zero-shot Learning

- Training data:
  - Labelled data and corresponding attribute feature
- Test data:
  - Labelled data and corresponding attribute feature
  - Test classes are not seen in training

- Attribute features: related to the object e.g.
  - Is the object an animal?
  - How many legs?
  - Property of tail etc.

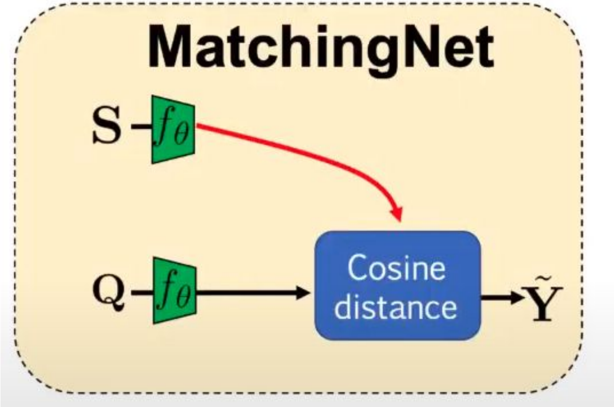
Meta-learning

- A common framework for solving few-shot and zero-shot learning.
- “Learning to learn”
- Training is done in multiple episodes on Base set
- Different set of classes are picked in each episode



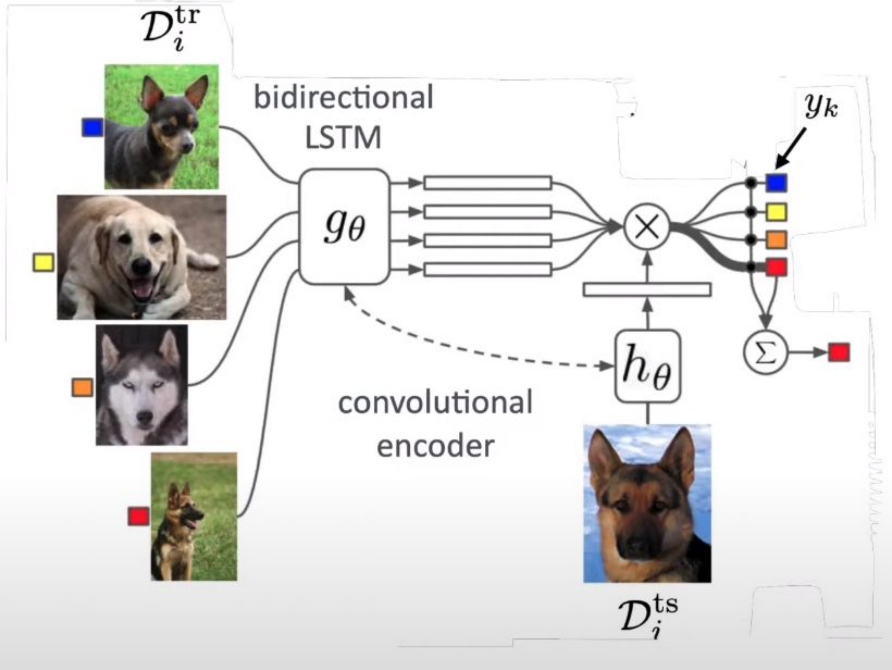
Matching Networks for One-shot learning

- **Training phase:** Learn cosine similarity based embedding models
- **Testing phase:** Use Nearest Neighbors in learned embedding space



Vinyals et al., NeurIPS 16

Matching Networks for One-shot learning



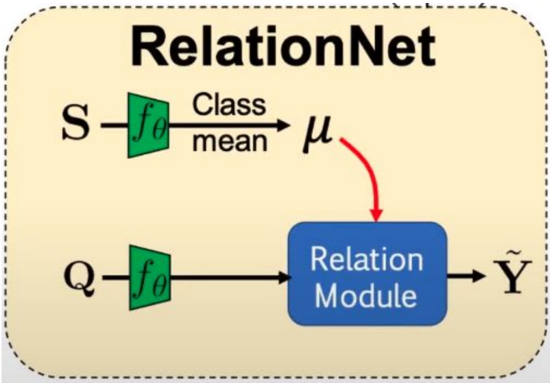
- $\hat{y}_{ts} = \sum_{i=1}^{k} a(\hat{x}, x_i) * y_i$
  where $a(\hat{x}, x_i)$ denotes the attention mechanism over examples

- Simplest form of attention mechanism $\implies$ softmax over cosine distances $c(.,.)$

$$a(\hat{x}, x_i) = \frac{e^{c(h(\hat{x}), g(x_i))}}{\sum_{j=1}^{k} e^{c(h(\hat{x}), g(x_j))}}$$

RelationNet for Few-shot and Zero-shot learning

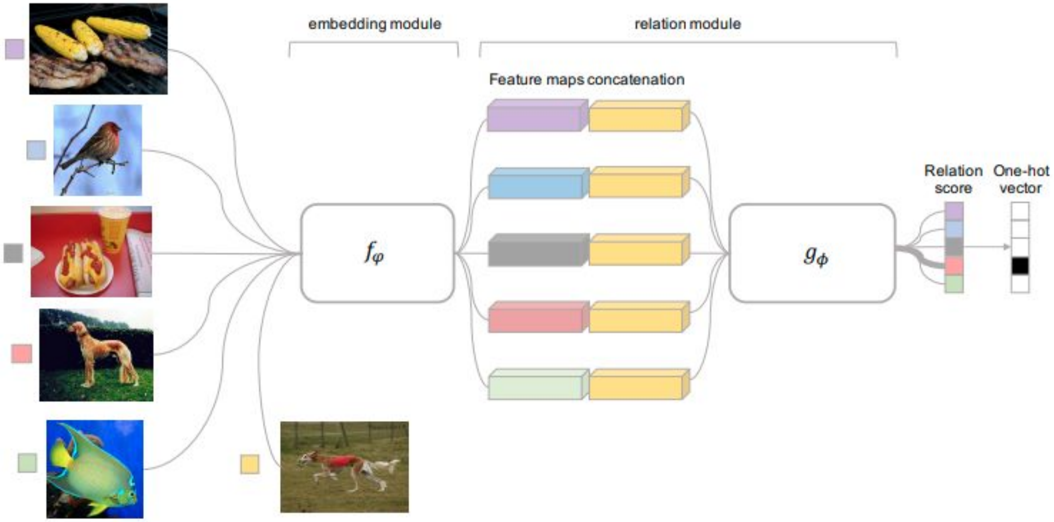
- Training phase:
  - Meta-learn both embedding module (feature representation)
  - **Relation module** (learnable transferable deep metric)
    - Instead of cosine
- Testing phase:
  - Use relation scores in embedding space to classify new samples



Sung et al., CVPR 18

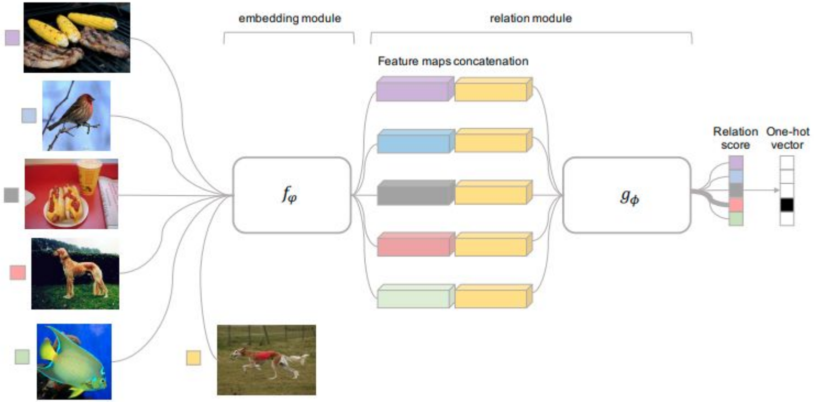
RelationNet for Few-shot and Zero-shot learning

- One-shot learning:



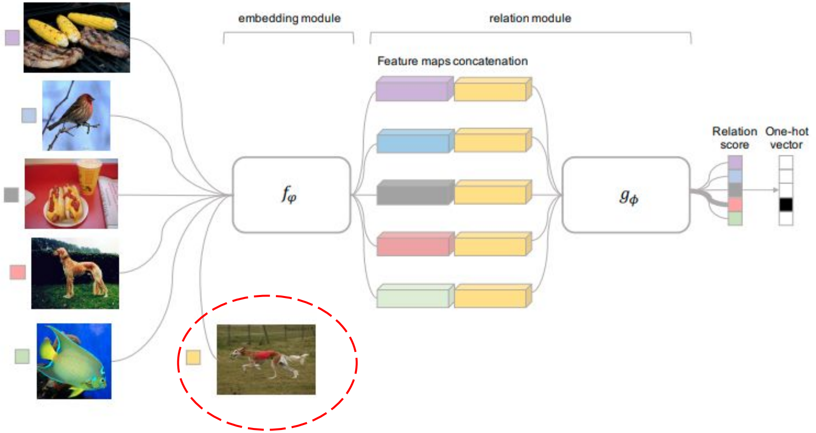
RelationNet for Few-shot and Zero-shot learning

- Few-shot learning:
  - Average over the embeddings for samples in each class of training set.
  - Rest of the framework stays same.



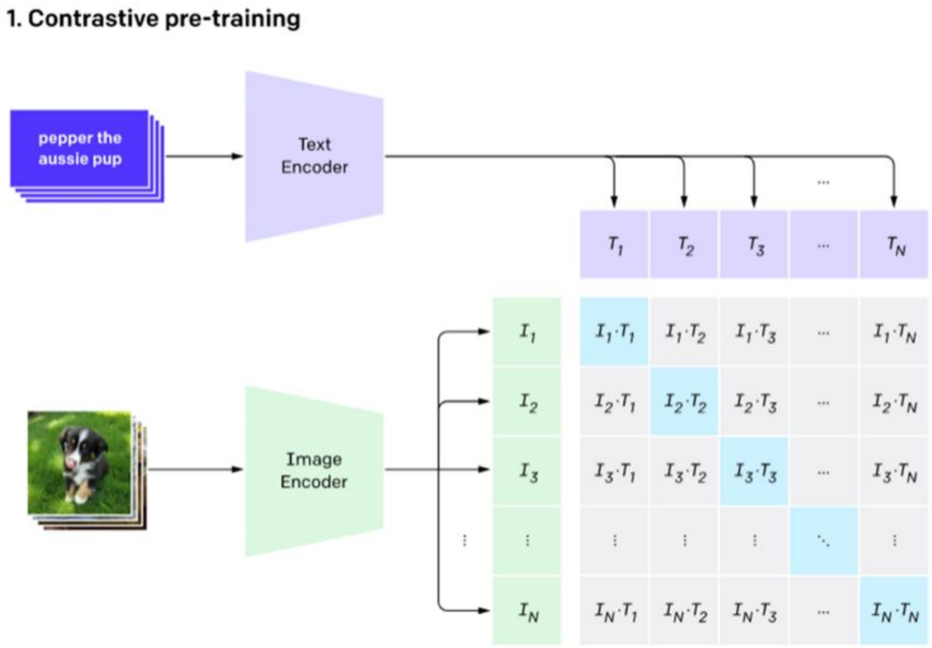
RelationNet for Few-shot and Zero-shot learning

- Zero-shot learning:
  - Use semantic class embedding for query
    - Attribute vector
    - Text embedding e.g. word2vec
  - Separate embedding modules are used for semantic and visual modalities

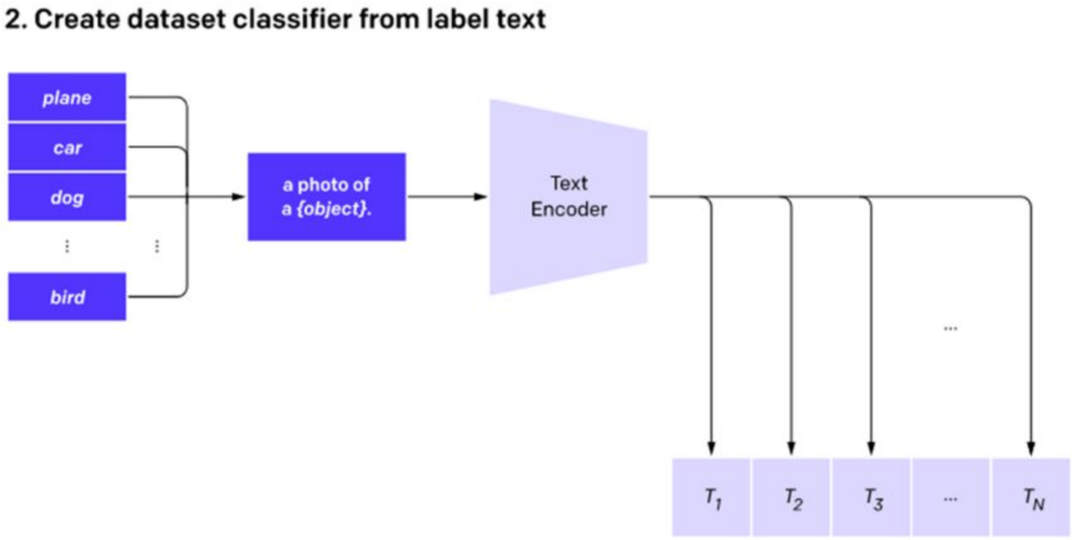


use semantic embedding

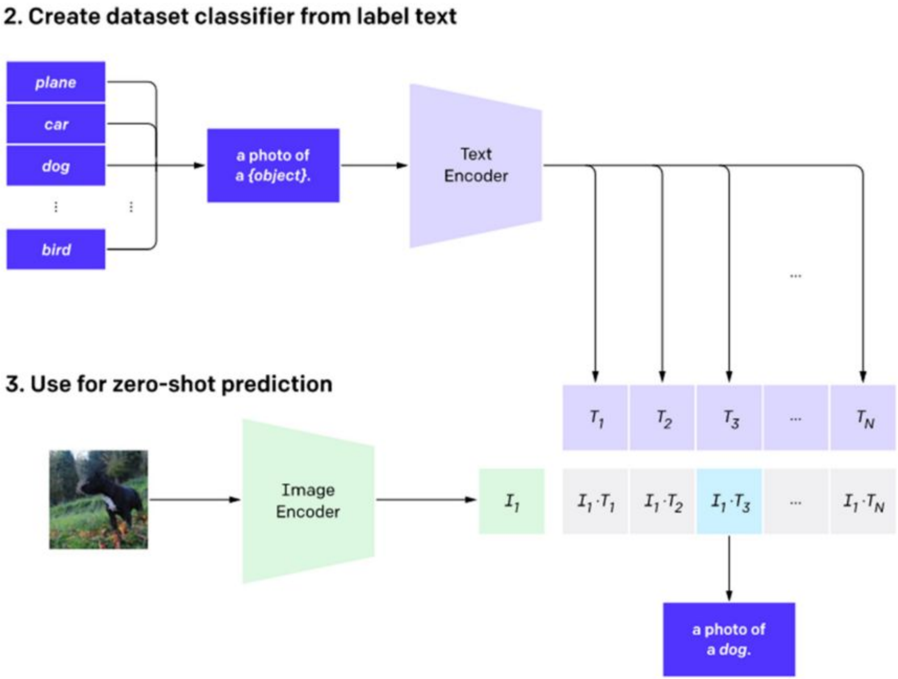
Zero-shot Learning with CLIP



Zero-shot Learning with CLIP



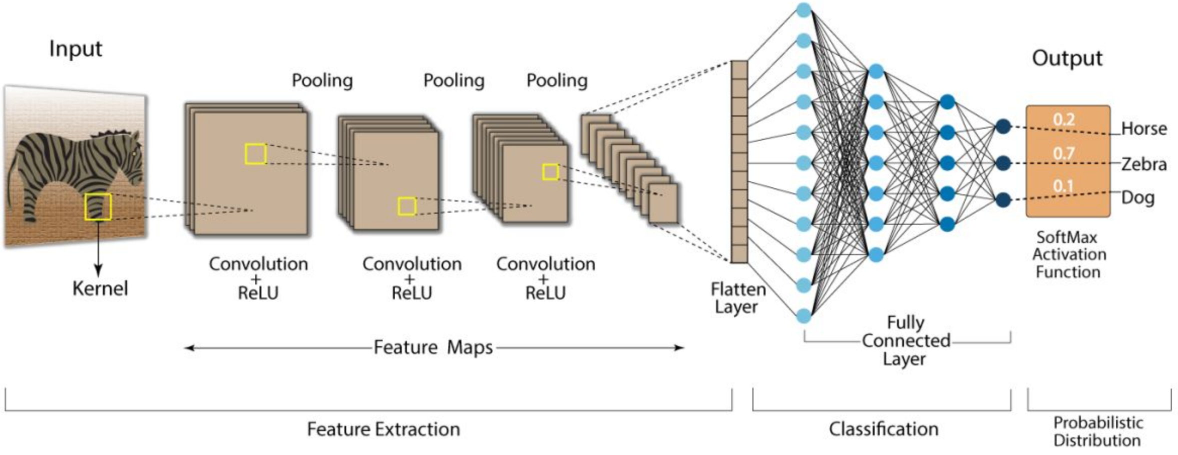
Zero-shot Learning with CLIP



Open-Set Recognition

- **Closed-set learning:** Testing samples always belong to training classes
- **Open-set learning:** There can be samples not belonging to train classes
    - Build a system that can classify known classes and reject unknown classes

Open-Set Recognition: Issue with softmax



- A network is trained to classify Horse, Zebra and Dog
- What happens if we pass a Cat image?

Open-Set Recognition: Issue with softmax

- **Naive Solutions:**
  1. If highest probability in softmax layer is low -> reject as unknown class
  2. Compute mean average vector for each class and apply a threshold to reject

Open-Set Recognition: OpenMax

- **Multi-class Meta-recognition:**
  - Compute mean average vectors for each class (correctly classified samples)
  - Fit highest distance values to a Weibull distribution.

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks, with per class Weibull fit to η largest distance to mean activation vector. Returns libMR models ρ_j which includes parameters τ_i for shifting the data as well as the Weibull shape and scale parameters:κ_i, λ_i.

Require: FitHigh function from libMR
Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \ldots v_N(x)$
Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.

for $j = 1 \ldots N$ **do**
  Compute mean AV, $\mu_j = mean_i(S_{i,j})$
  EVT Fit $\rho_j = (\tau_j, \kappa_j, \lambda_j) = \text{FitHigh}(\|\hat{S}_j - \mu_j\|, \eta)$
end for
Return means μ_j and libMR models ρ_j

Bendale and Boult, CVPR 16

Open-Set Recognition: OpenMax

- **OpenMax:**
  - “Opening up” of softmax
  - Revise activation values for top α classes.
  - 0-th position is for unknown class.
  - Intuition: Borrow activation values from top α classes to unknown class.

Algorithm 2 OpenMax probability estimation with rejection of unknown or uncertain inputs.

Require: Activation vector for $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$
Require: **means** μ_j and libMR models $\rho_j = (\tau_i, \lambda_i, \kappa_i)$
Require: α, the numer of “top” classes to revise

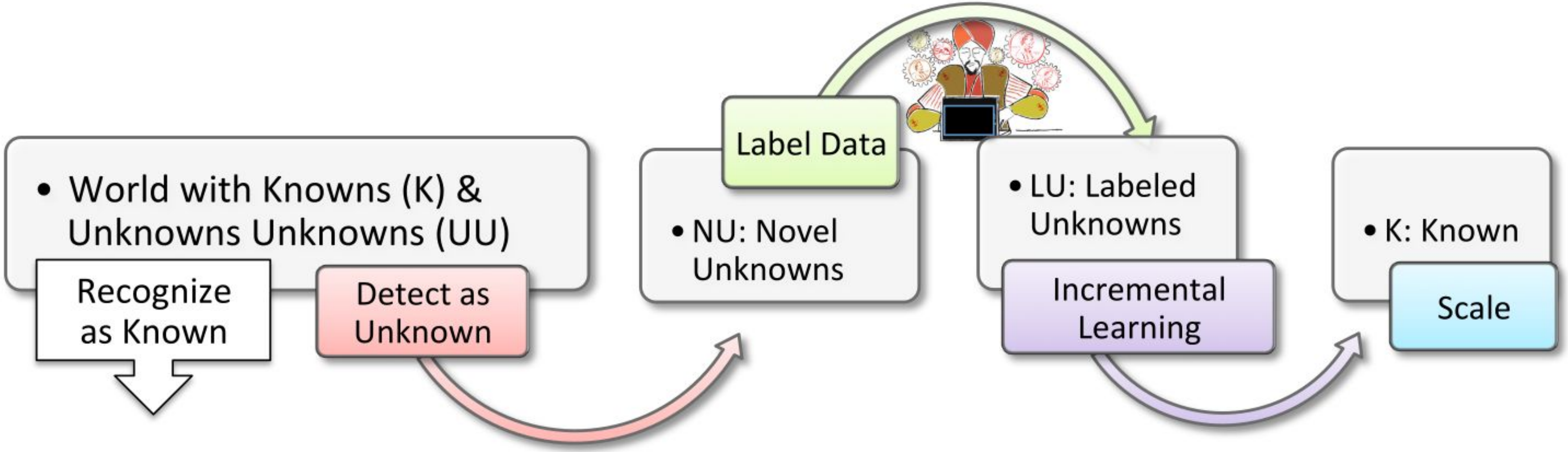
Let $s(i) = \text{argsort}(v_j(x))$; Let $\omega_j = 1$
for $i = 1, \ldots, \alpha$ **do**
 $\omega_{s(i)}(x) = 1 - \frac{\alpha - i}{\alpha} e^{-\left(\frac{\|x - \tau_{s(i)}\|}{\lambda_{s(i)}}\right)^{\kappa_{s(i)}}}$
end for
Revise activation vector $\hat{v}(x) = \mathbf{v}(\mathbf{x}) \circ \omega(\mathbf{x})$
Define $\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$.

$$\hat{P}(y = j|\mathbf{x}) = \frac{e^{\hat{\mathbf{v}}_\mathbf{j}(\mathbf{x})}}{\sum_{i=0}^{N} e^{\hat{\mathbf{v}}_\mathbf{i}(\mathbf{x})}} \qquad (2)$$

Let $y^* = \text{argmax}_j\, P(y = j|\mathbf{x})$
Reject input if $y^* == 0$ or $P(y = y^*|\mathbf{x}) < \epsilon$

Open-World Recognition

- **Open-set learning:** Rejects unknowns
- **Open-world learning:** Discover and/or learn new categories without labeled data



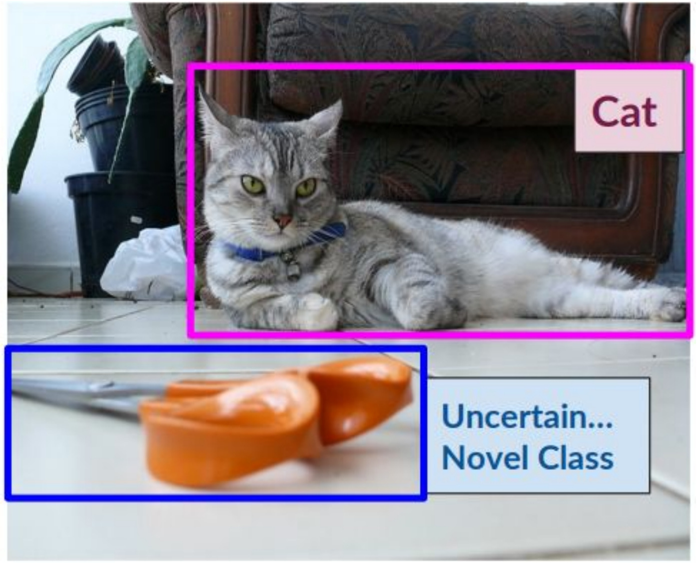
Bendale and Boult, CVPR 15

Open-World Recognition

There is a cat and a novel class object in the image.

The novel class is scissors.

Updating on scissors...



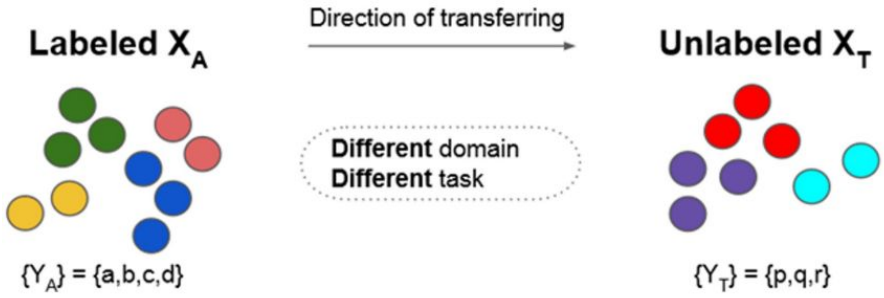
Open-World Recognition

- Spectrum of Open-World:

Open-World Recognition: Category discovery

- **Problem:**
    - How can we discover new things in the world with only unlabeled data and no names?

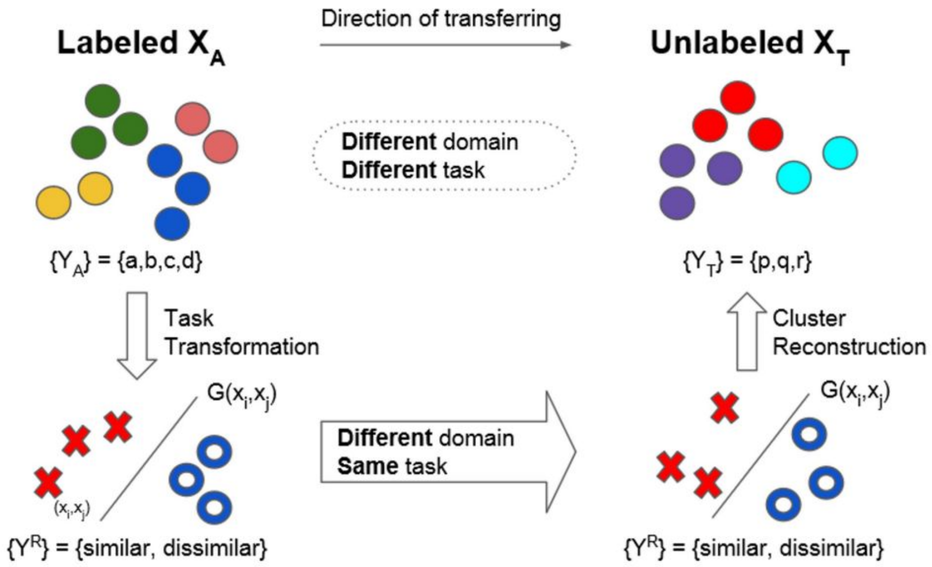
- **Given:**
    - Labeled dataset with known categories
    - Unlabeled dataset with unknown categories
    - Goal: Cluster unknown categories in unlabeled data, leveraging labeled set if possible



Hsu et al., ICLR 2018

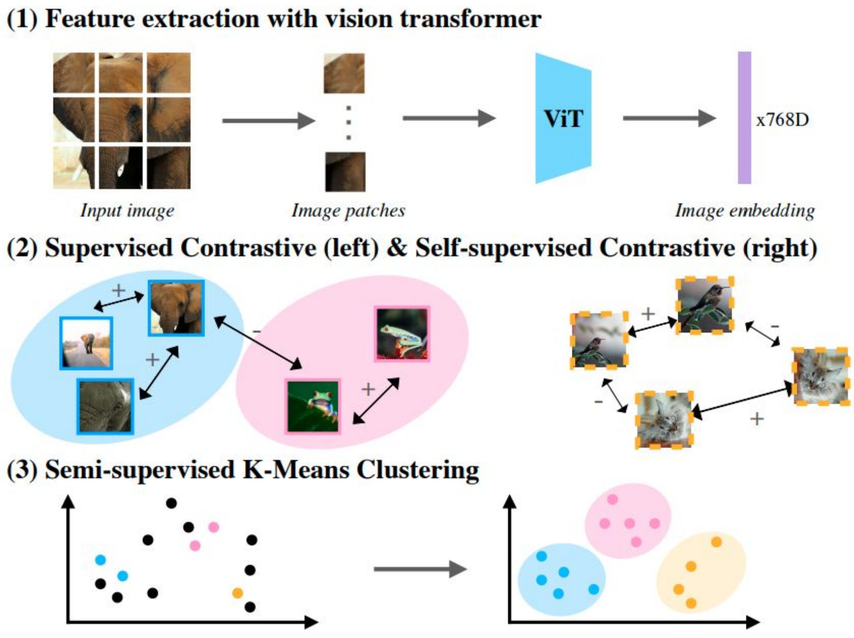
Open-World Recognition: Category discovery

- Approach:
  - Learn to compare things (similar/dissimilar)
  - Neural-network based clustering
  - Constraint-based clustering loss derived



Open-World Recognition: Generalized Class Discovery

- Problem: Generalized category discovery in unlabeled data
    - Some labeled known categories
    - Mix of known and unknown categories in unlabeled data
- Approach:
    - DINO self-supervised ViT backbone
        - strong NN classifier
    - Semi-supervised contrastive learning
    - Semi-supervised k-means clustering
        - Mean embeddings for each class act as centroid initialization
        - Other centroids are randomly initialized



Vaze et al., CVPR 22

References

1. Zsolt Kira, ICCV 2023 Tutorial: Open-World Learning: From Zero-Shot to Truly Open-World
2. Vineeth Balasubramanian, Deep Learning for Computer Vision course
3. A Newbie-Friendly Guide to Transfer Learning https://www.v7labs.com/blog/transfer-learning-guide#types-of-deep-transfer-learning
4. Convolutional Neural Networks https://developersbreach.com/convolution-neural-network-deep-learning/
5. Meta-Learning: Learning to Learn Fast https://lilianweng.github.io/posts/2018-11-30-meta-learning/

Thank you!