UNIVERSITAT DE BARCELONA

CVC Centre de Visió per Computador

Transfer Learning

Petia Radeva,

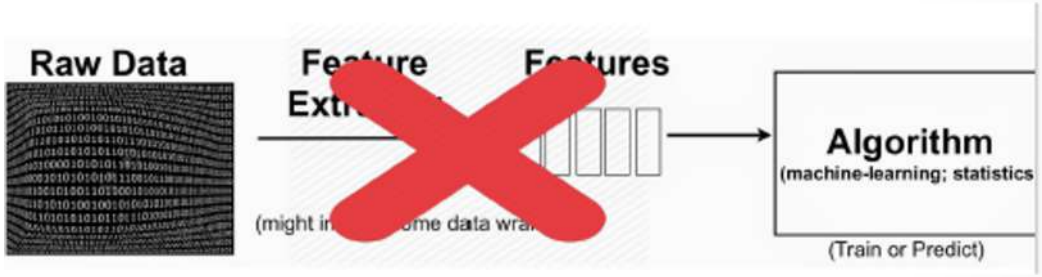
University of Barcelona &
Computer Vision Center

radevap@gmail.com

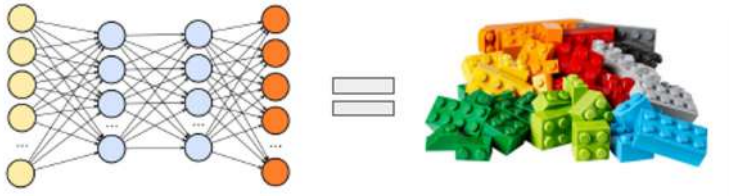
What makes DNN so popular?

It has the three advantages:

- 1. Self-learned high-level features representations

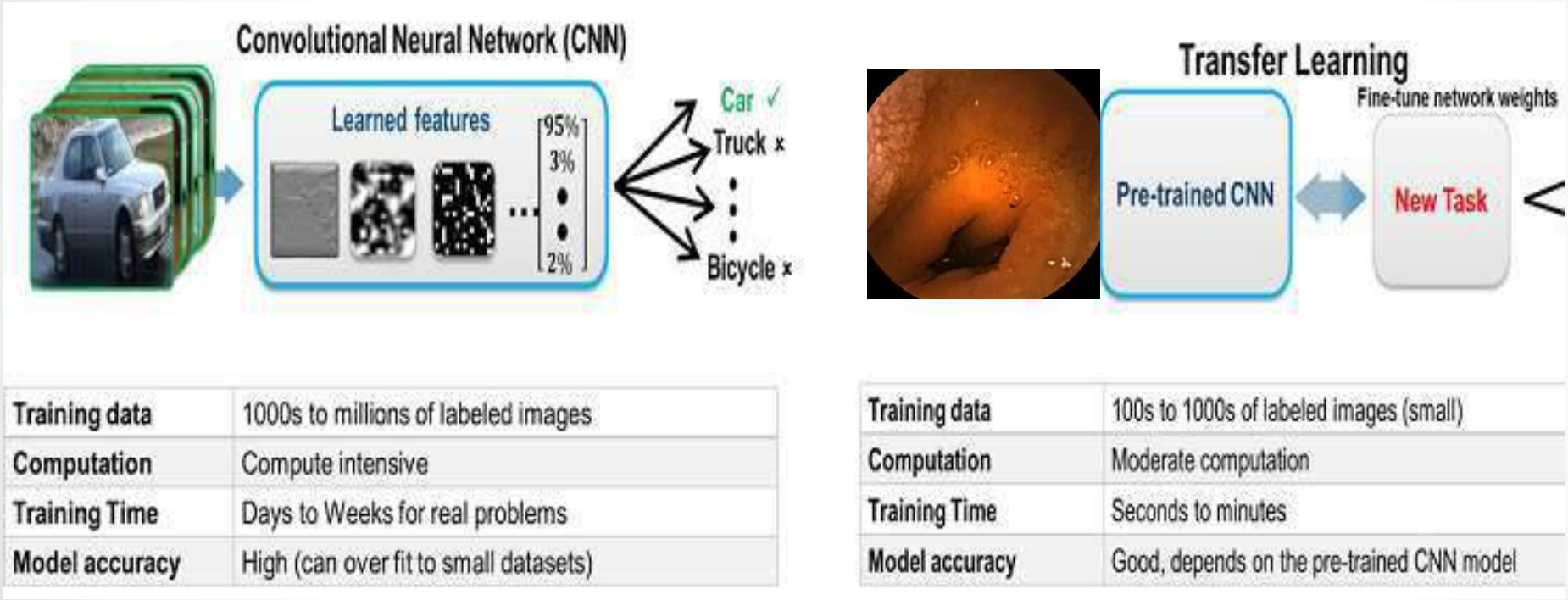


- 2. Modularity



- 3. Transfer Learning

Transfer Learning and Fine-tuning



Training data	1000s to millions of labeled images
Computation	Compute intensive
Training Time	Days to Weeks for real problems
Model accuracy	High (can over fit to small datasets)

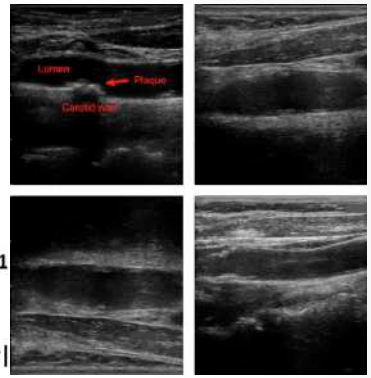
Training data	100s to 1000s of labeled images (small)
Computation	Moderate computation
Training Time	Seconds to minutes
Model accuracy	Good, depends on the pre-trained CNN model

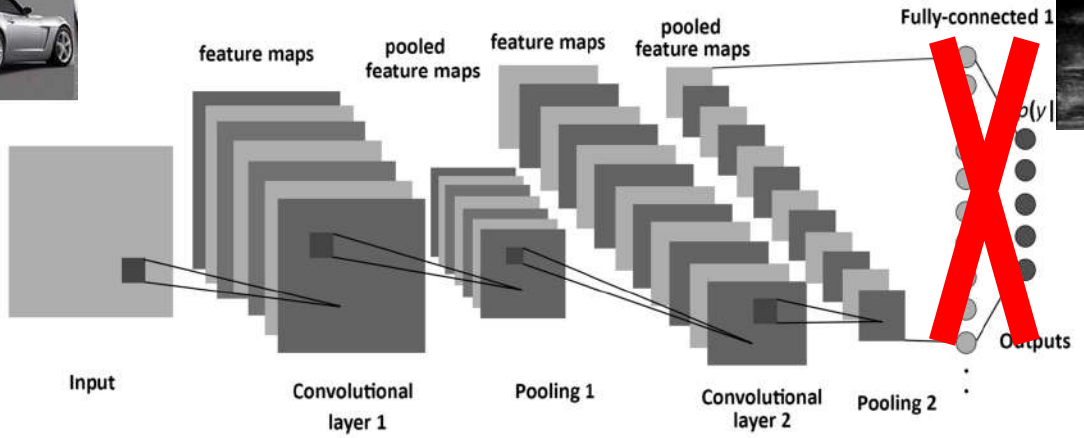
Example of Fine-tuning

Full (Re)Training or Fine-Tuning?

- Non-medical and medical images share characteristics

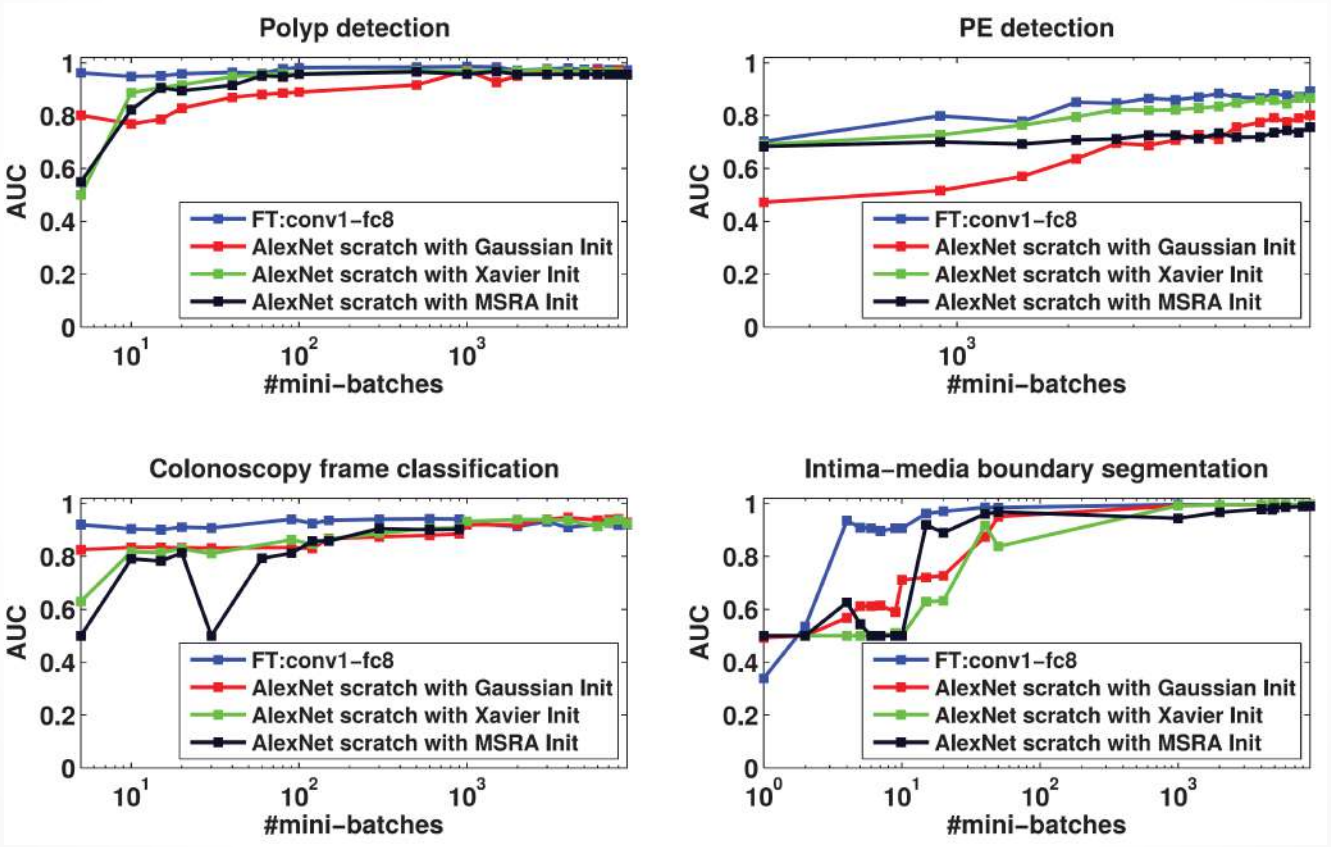
Update the w_i parameters





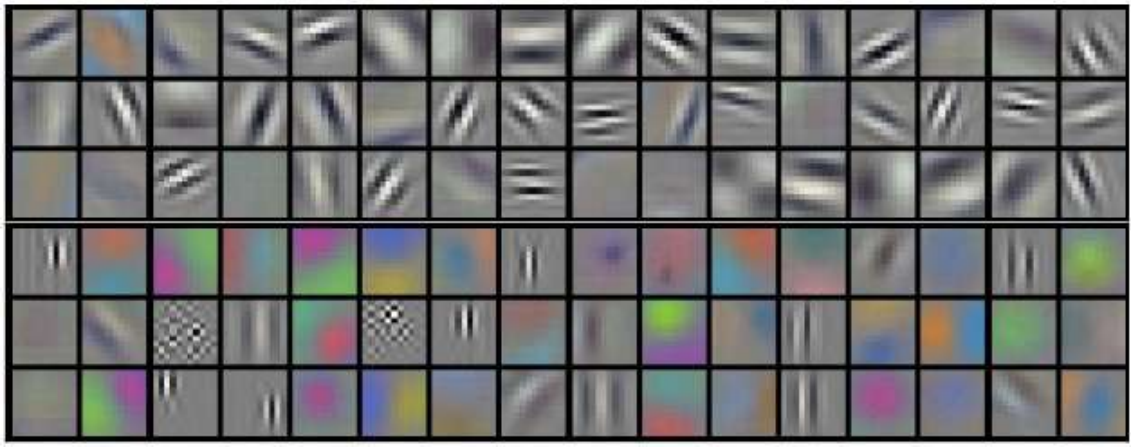
- Tajbakhsh, N. et al. 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE Transactions on Medical imaging, 35(5), pp.1299-1312.
- It has been shown that it is better to fine tune pre-trained models (incl. those trained with non-medical images)
- The training converges faster

Full (Re)Training or Fine-Tuning?



Transfer Learning Methods: Using pre-trained CNN features

- Understanding CNNs:



- Lower convolutional layers capture low-level image features
- Classification is done by FC layers that explain how edges and shapes are combined.

For a new task, we can thus simply use the off-the-shelf features of a state-of-the-art CNN pre-trained on ImageNet and train a new model on these extracted features.

- In practice, we tune them with a small learning rate in order to ensure that we do not unlearn the previously acquired knowledge.

Transfer learning

- What we do every day…

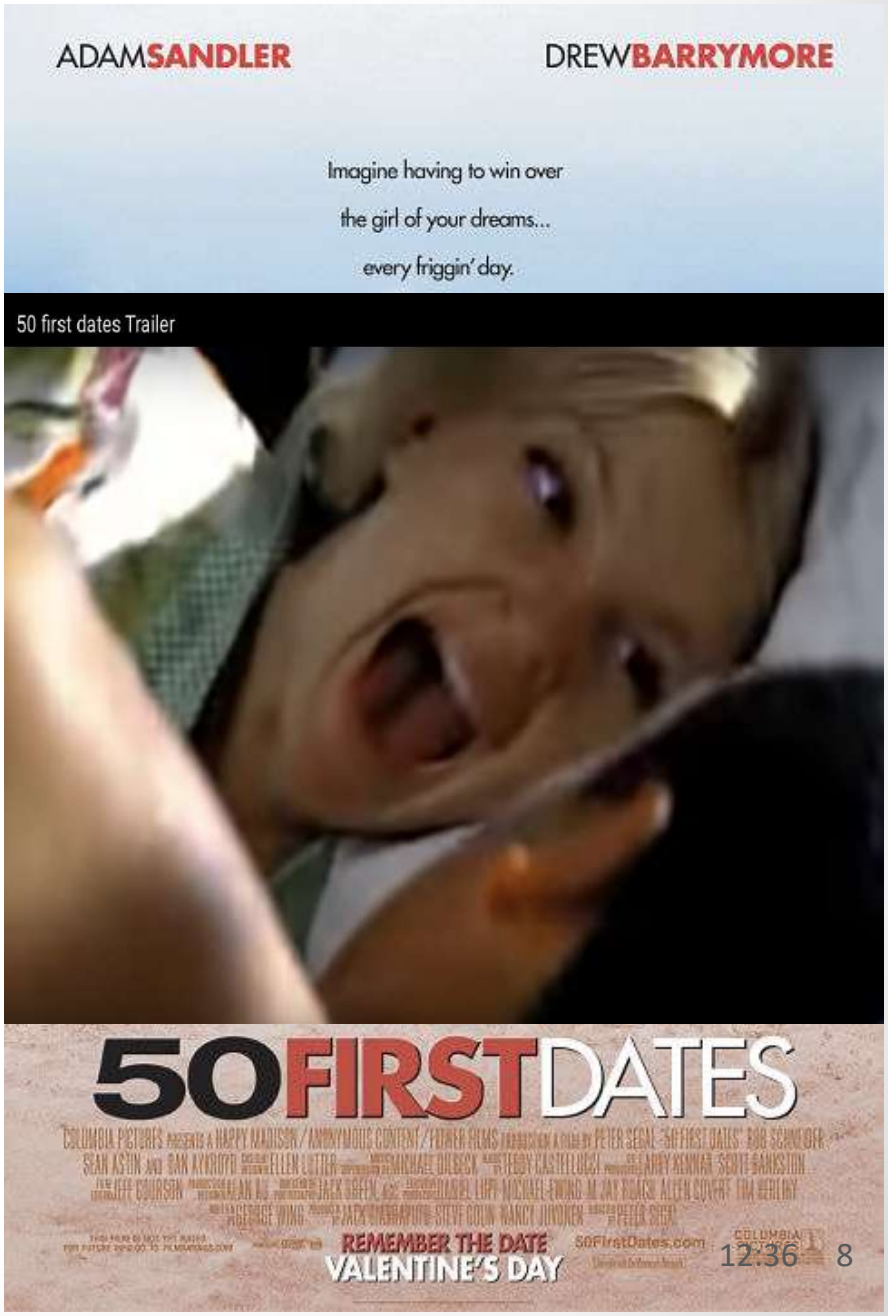
Let’s imagine that…

Henry Roth is a man afraid of commitment up until he meets the beautiful Lucy.

They hit it off and Henry think he's finally found the girl of his dreams,

until he discovers:

she has short-term memory loss and forgets him every next day.



Index

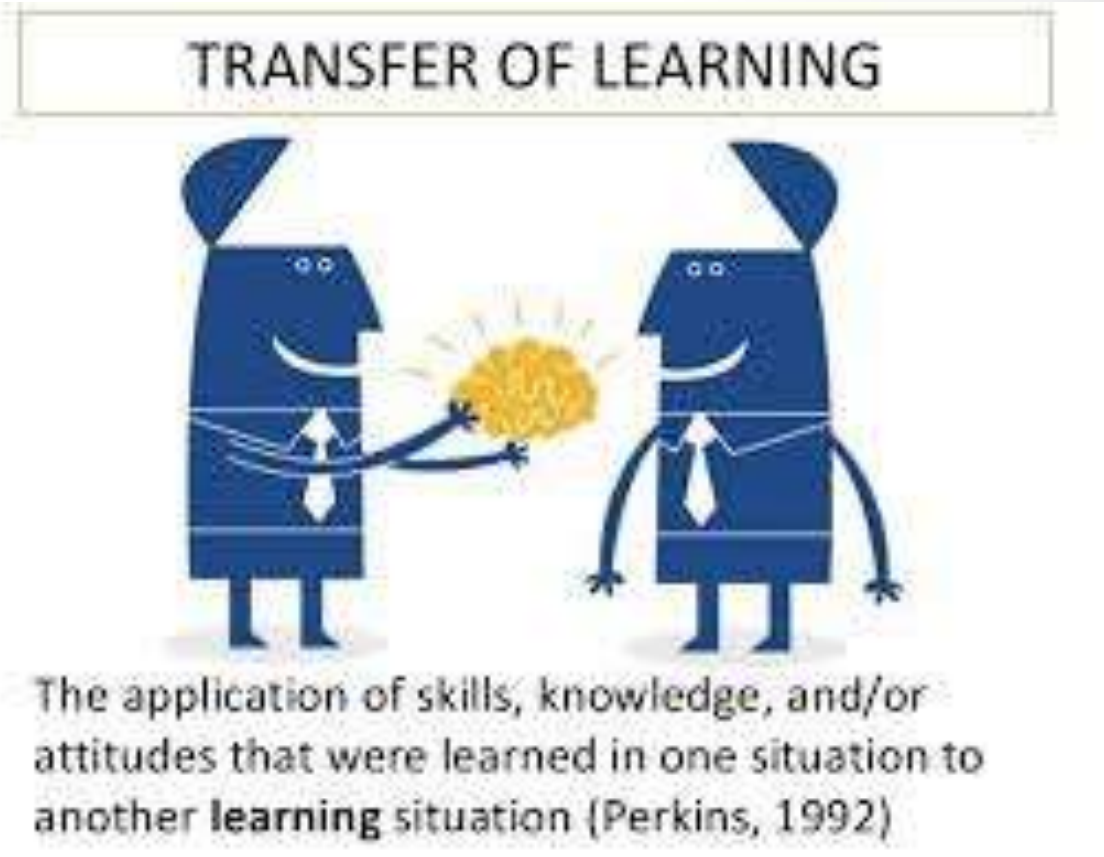
1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Transfer learning

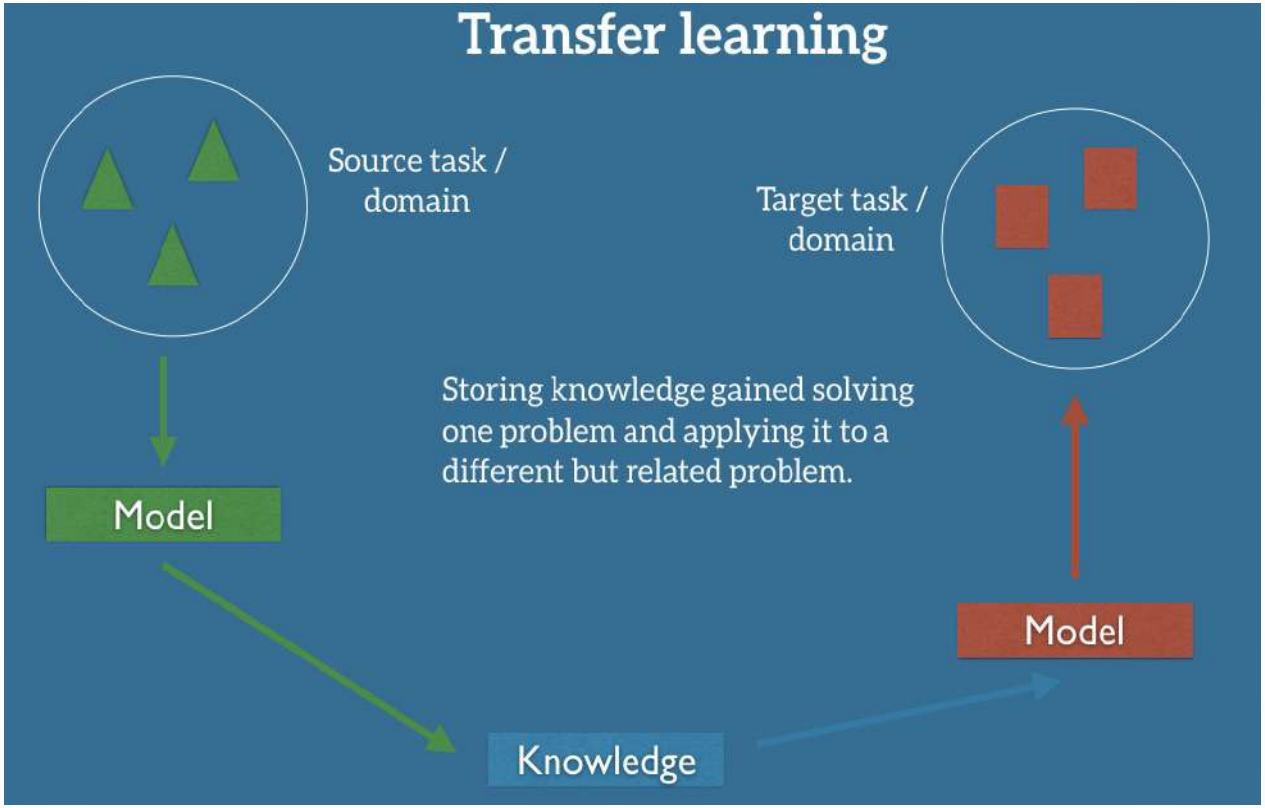
- What we do every day…

The origin of Transfer Learning:

how individuals transfer in one context to another context that share similar characteristics.



What is Transfer learning?

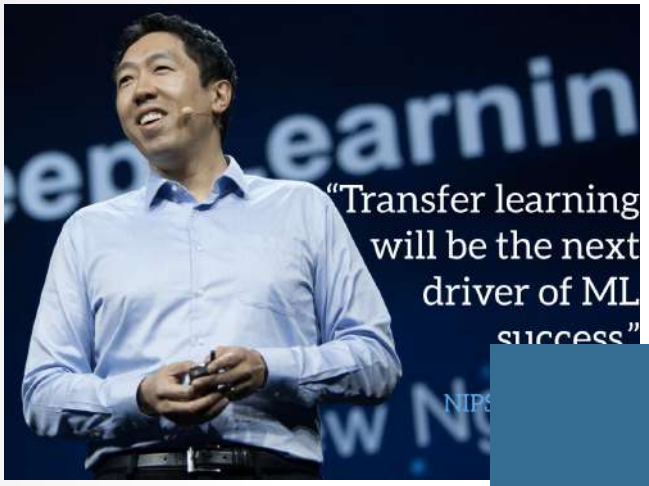


Transfer Learning (TL):

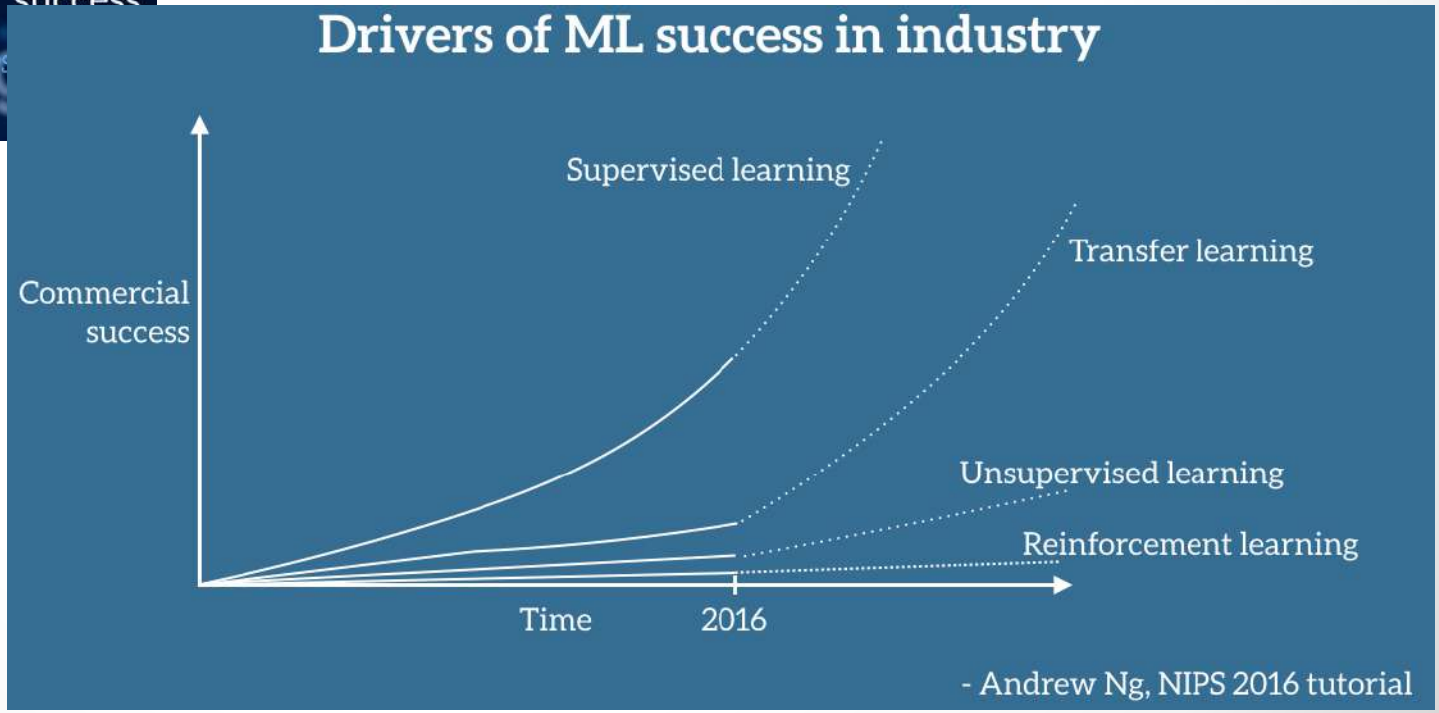
- The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks (in new domains).

https://ruder.io/transfer-learning/index.html#whatistransferlearning

Why Transfer Learning Now?



Andrew Ng, chief scientist at Baidu and professor at Stanford



https://ruder.io/transfer-learning/index.html#fn44

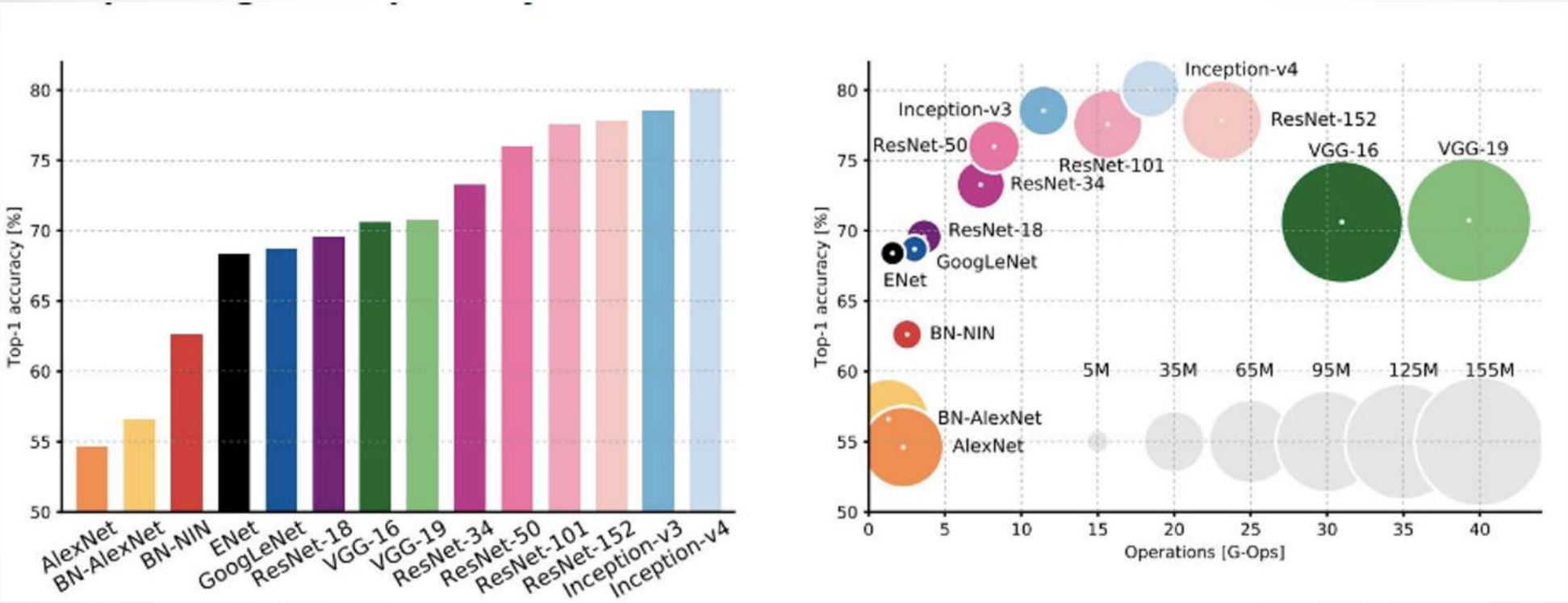
Why do we need Transfer Learning?

- Successful models are immensely **data-hungry** relying on **huge amounts of labeled data** to achieve their performance.
- A model is **asked to behave in a new situation** on tasks that not exactly it was trained for.

Why now is the time for Transfer Learning?

- Availability of well working models
  - ResNet achieveing superhuman performance on recognizing objects
  - Google's Smart Reply [2] automatically handles 10% of all mobile responses;
  - speech recognition error has consistently dropped and is more accurate than typing [3];
  - we can automatically identify skin cancer as well as dermatologists;
  - Google's NMT system [4] is used in production for more than 10 language pairs; Baidu

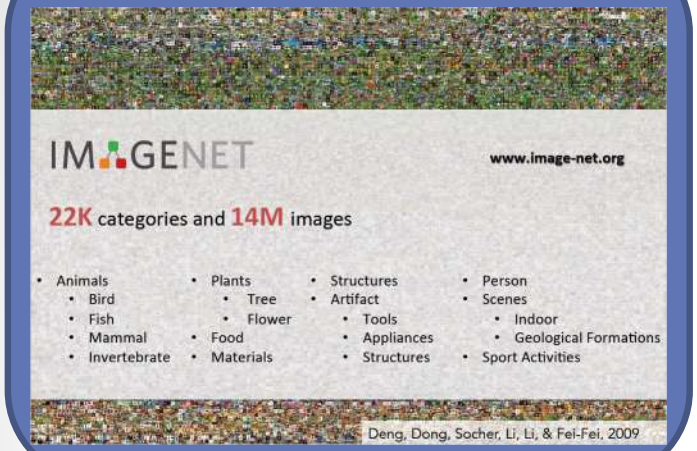
Analysis of CNNs



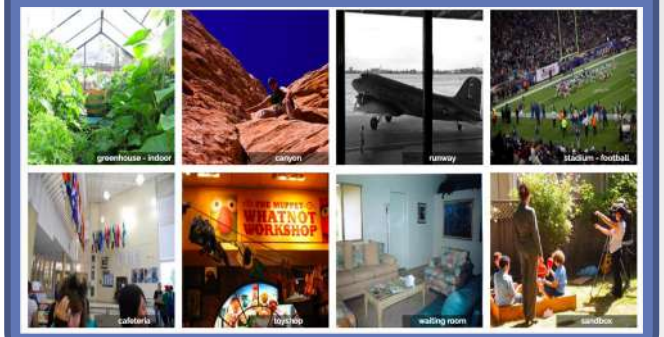
- **Millions of parameters!!!**

The process of training a CNN consists of training all hyperparameters: convolutional matrices and weights of the fully connected layers.

Deep Learning Datasets

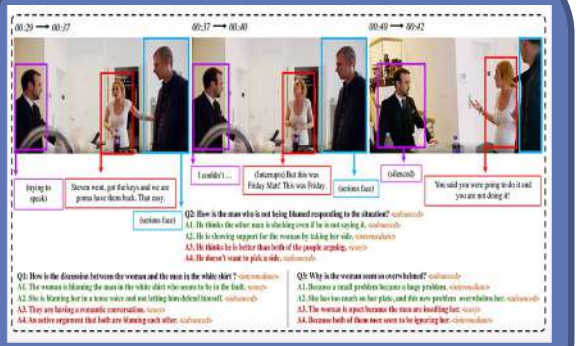


LVIS Challenge: 2.2M masks, 16K images



Places2: 10M images

Lyft Level 5



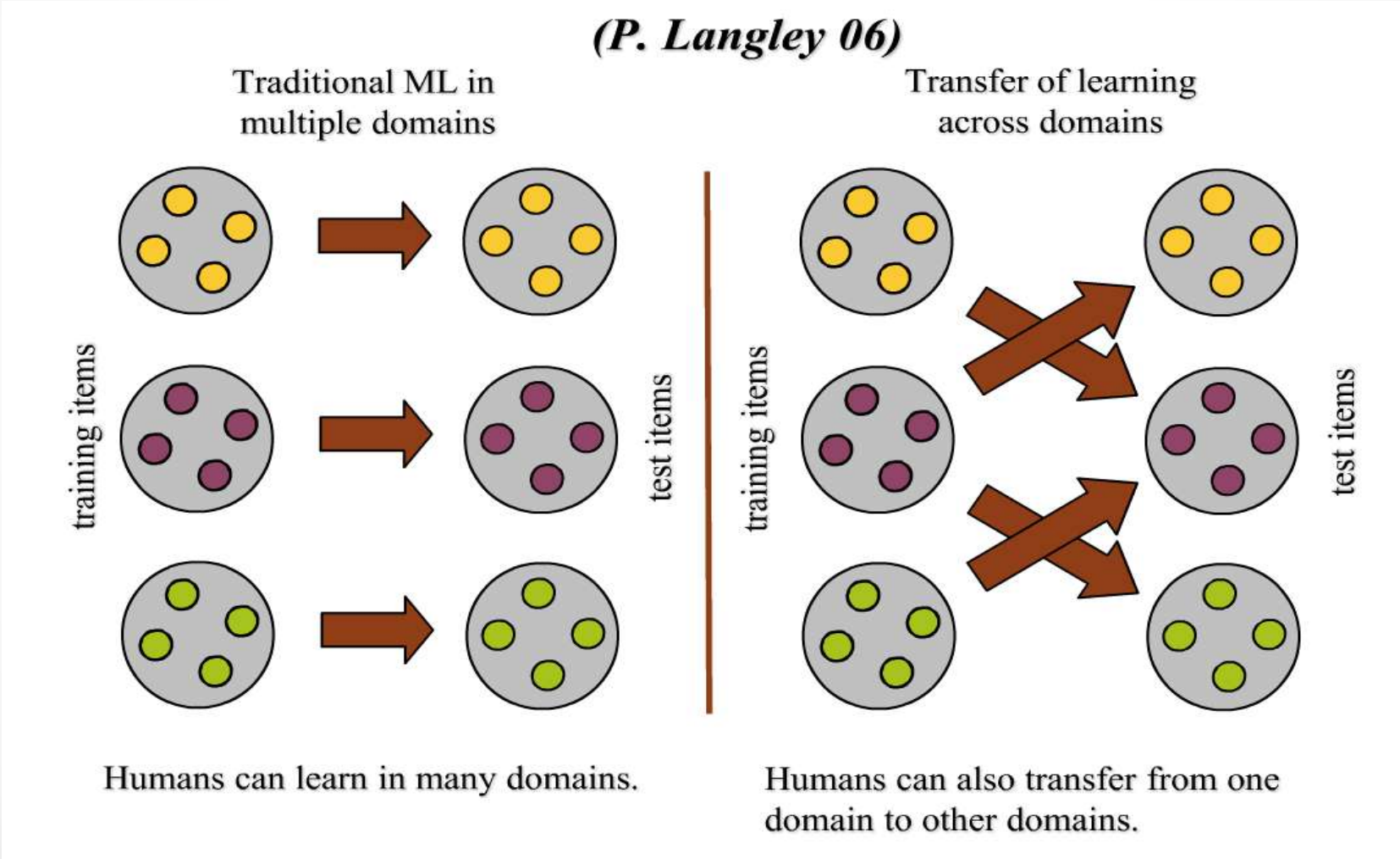
SocialIQ

TACO: Waste in the wild

FastMRI

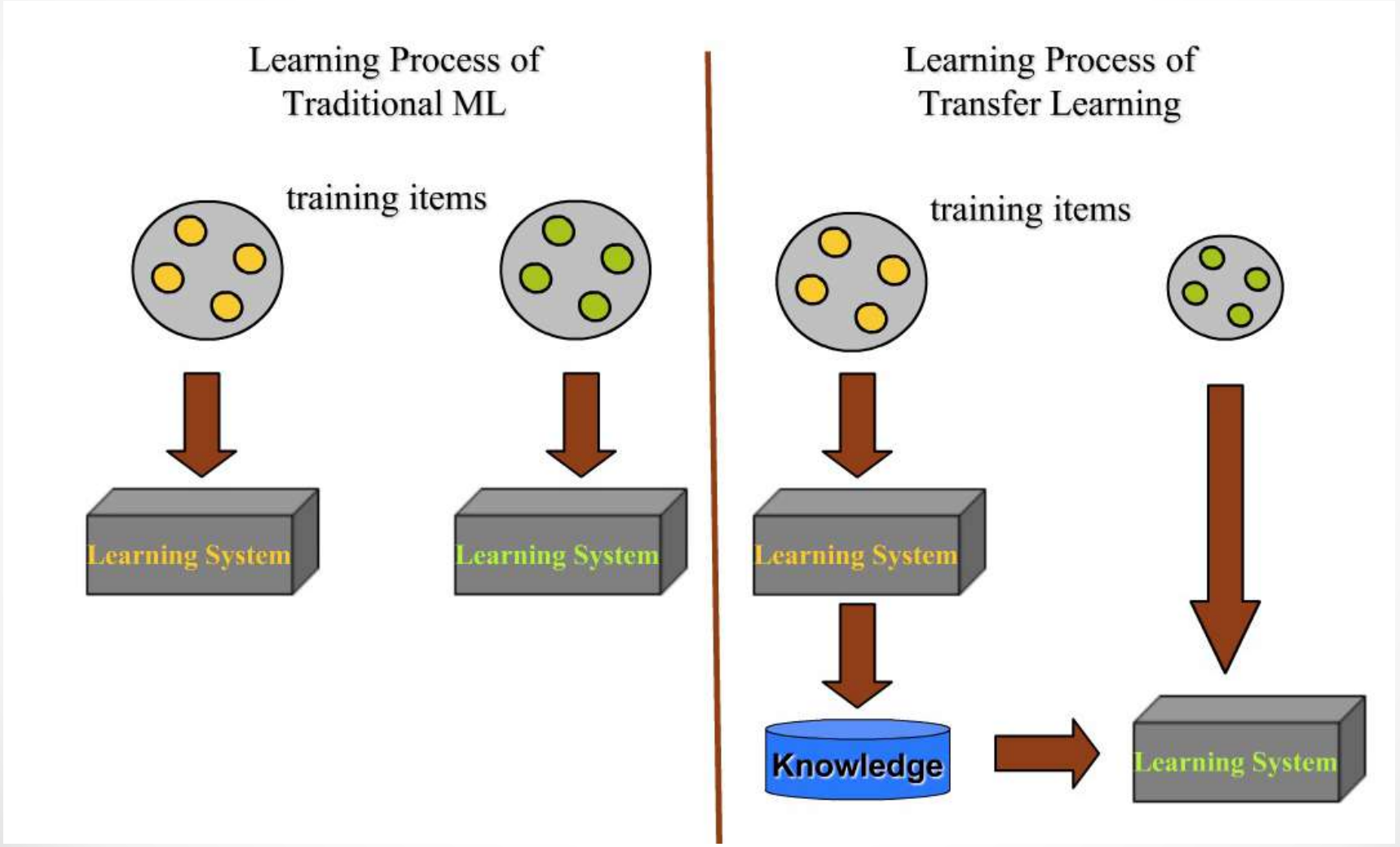
- https://www.datasetlist.com

Traditional ML vs. TL



- Pan, S. J., & Yang, Q. (2009). A Survey on Transfer Learning. IEEE Trans. KDE, 22(10), 1345-1359

Traditional ML vs. TL



- Pan, S. J., & Yang, Q. (2009). A Survey on Transfer Learning. IEEE Trans. KDE, 22(10), 1345-1359

Notation

Transfer learning is based on the dual concept of **domain** and **task**:

Domain:

- It consists of two components: A feature space X, a marginal probability distribution P(X), where $X=\{x_1,x_2,\ldots x_n\}$
  - In general, if two domains are different, then they may have different feature spaces or different marginal distributions.

Task:

- Given a specific domain D and label space Y, for each x in the domain D, to predict its corresponding label y
  - In general, if two tasks are different, then they may have different label spaces or different conditional distributions.

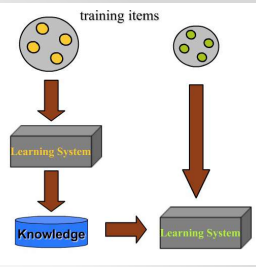
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Why Transfer Learning?

- In some domains:
- **labeled data are in short supply,**
- **the calibration effort is very expensive,**
- **the learning process is time consuming.**

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Why Transfer Learning?

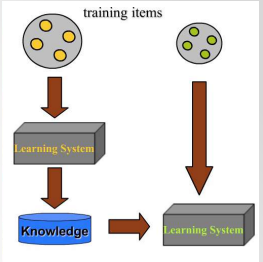


- How to extract knowledge learnt from related domains **to help learning** in a target domain with a **few labeled** data?
- How to extract knowledge learnt from related domains **to speed up learning** in a target domain?

Transfer learning techniques may help!

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359.

A Definition of TL



Given a source domain D_S, a corresponding source task T_S, as well as a target domain D_T and a target T_T.

the **objective** of transfer learning now is to enable us to learn
the **target conditional probability distribution** $P(Y_T|X_T)$ in D_T
with the information gained from D_S and T_S where $D_S \neq D_T$ or $T_S \neq T_T$.

In most cases, a limited number of labeled target examples, which is exponentially smaller than the number of labeled source examples are assumed to be available.

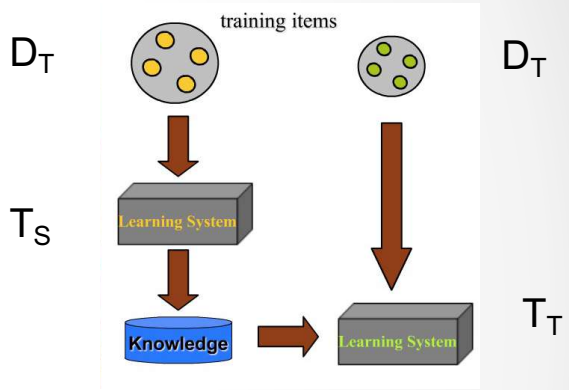
As both the domain D_i and the task T_i are defined as tuples, these inequalities give rise to four transfer learning scenarios.

Index

1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Transfer Learning Scenario (1)

Given source and target domains D_S and D_T where D={X,P(X)} and source and target tasks T_S and T_T where T={Y,P(Y|X)} source and target conditions can vary:

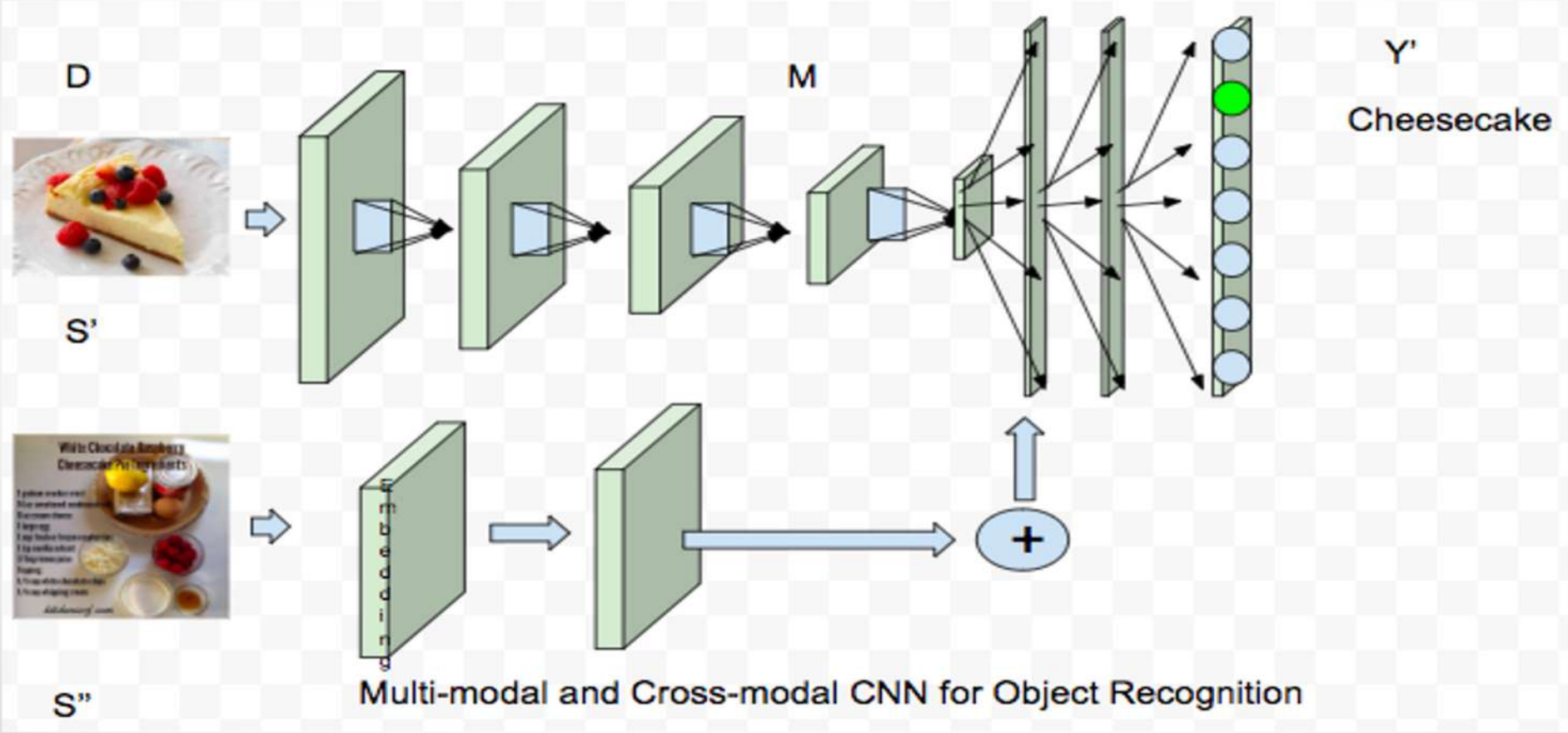


1. $X_S <> X_T$:

The feature spaces of the source and target domain are different, e.g. the documents are written in two different languages.

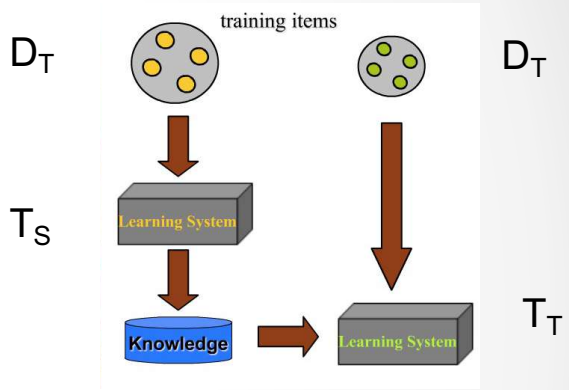
In the context of natural language processing, this is generally referred to as **cross-lingual adaptation**.

Transfer Learning Scenario 1



Transfer Learning Scenario (2)

Given source and target domains D_S and D_T where $D=\{X,P(X)\}$ and source and target tasks T_S and T_T where $T=\{Y,P(Y|X)\}$ source and target conditions can vary:

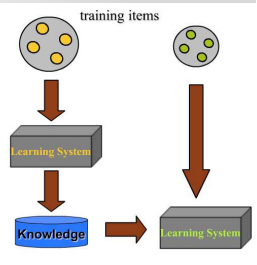


2. $P(X_S) \neq P(X_T)$.

The marginal probability distributions of source and target domain are different, e.g. the documents discuss different topics.

This scenario is generally known as **domain adaptation.**

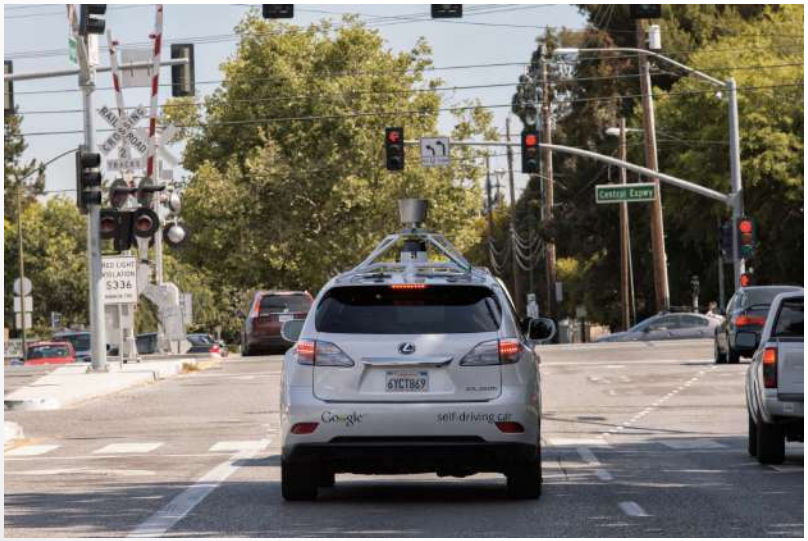
Learning from simulations



For many applications, gathering data and training a model in the real world is either expensive, time-consuming, or simply too dangerous.

Learning from a simulation and applying the acquired knowledge to the real world is an instance of **transfer learning scenario 2:**

- the conditional probability distributions between simulation and real wold might be different as the simulation is not able to fully replicate all reactions in the real world.



- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

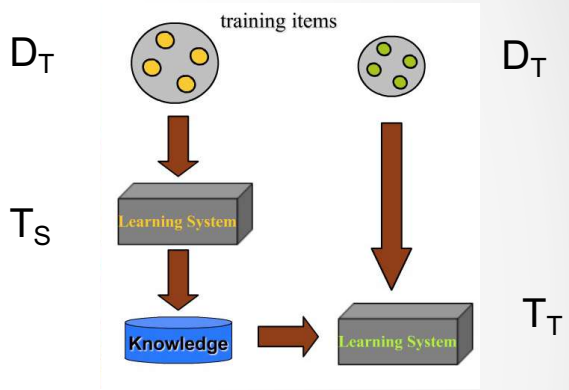
Domain adaptation

- Note!:

- Even of the training and the test data look the same, the training data could contain a bias imperceptible to humans but which the model will exploit to overfit in the training data.

Transfer Learning Scenarios (3)

Given source and target domains D_S and D_T where D={X,P(X)} and source and target tasks T_S and T_T where T={Y,P(Y|X)} source and target conditions can vary:



3. $L_S <> L_T$

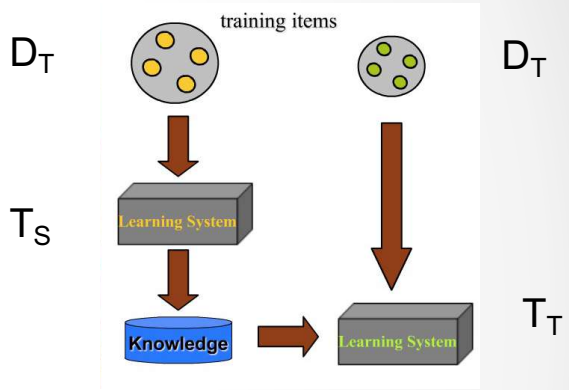
The label spaces between the two tasks are different, e.g. documents need to be assigned different labels in the target task.

- In practice, this scenario usually occurs with scenario 4, as it is extremely rare for two different tasks to have different label spaces, but exactly the same conditional probability distributions.

Transfer Learning Scenario (4)

Given source and target domains D_S and D_T where $D=\{X,P(X)\}$ and source and target tasks T_S and T_T

- where $T=\{Y,P(Y|X)\}$ source and target conditions can vary:

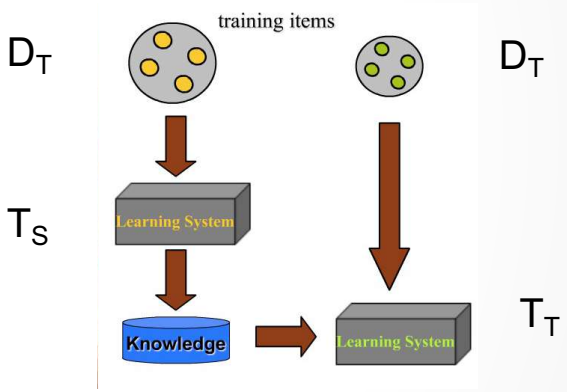


4. $P(Y_S|X_S)<>P(Y_T|X_T)$

The conditional probability distributions of the source and target tasks are different, e.g. source and target documents are unbalanced with regard to their classes.

Compare

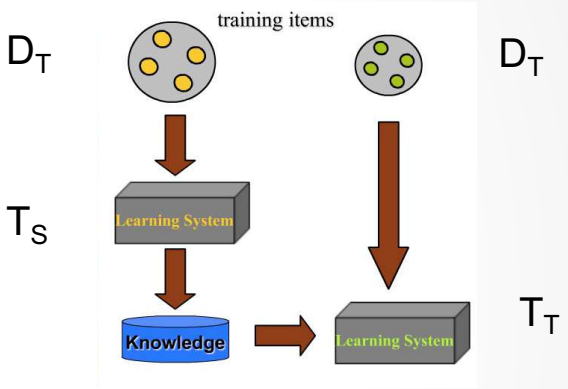
- Transfer learning
- Domain adaptation
- Fine-tunning



Transfer Learning

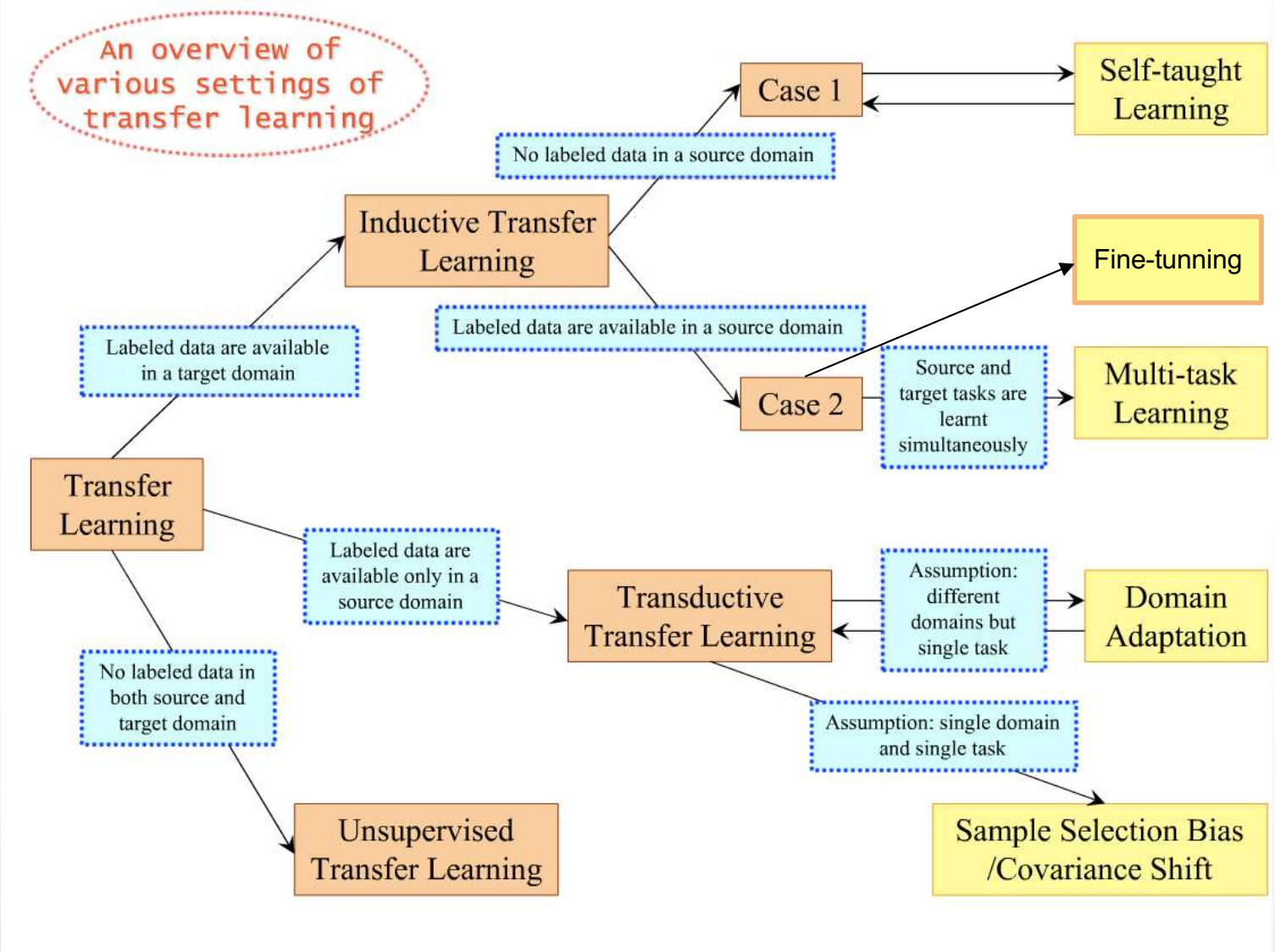
- Until now we considered 4 scenarios:

- Scenario 1: $X_S <> X_T$
- Scenario 2: $P(X_S) <> P(X_T)$
- Scenario 3: $L_T <> L_S$
- Scenario 4: $P(Y_S|X_S) <> P(Y_T|X_T)$

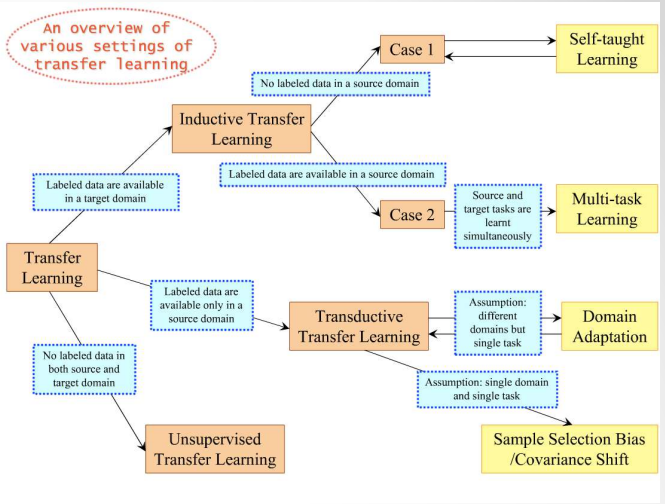


- But what if we do not have Labelled data?

Taxonomy according to available labels



Index



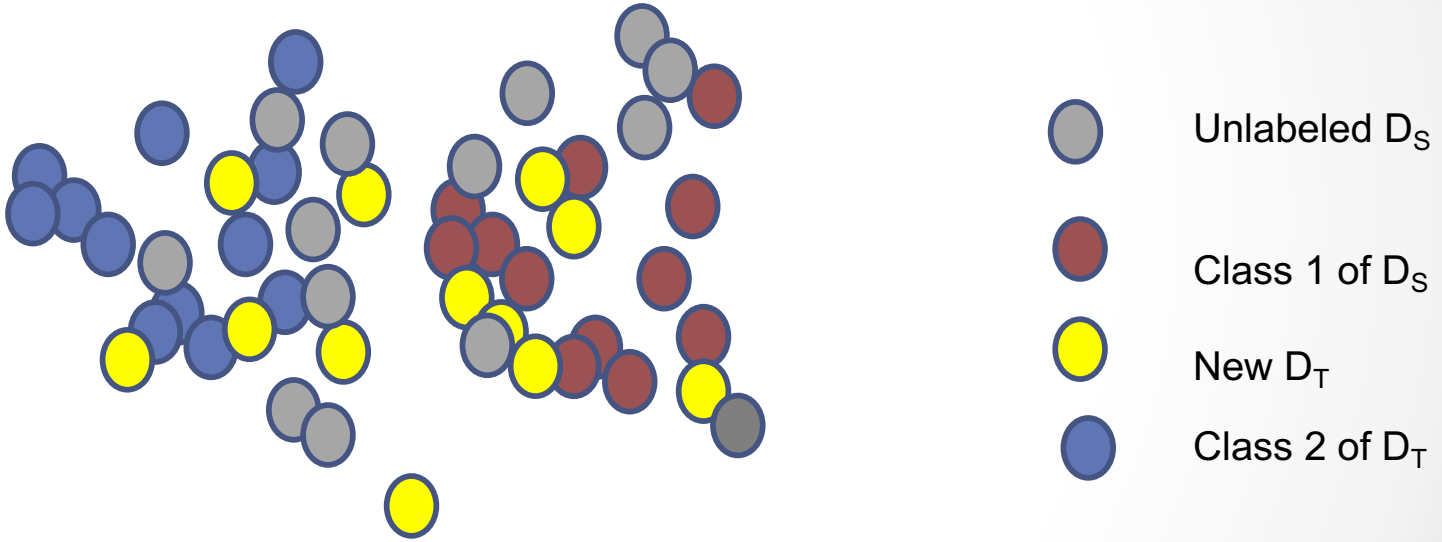
1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Self-taught Clustering (STC)

- Input: A lot of unlabeled data in a source domain and a few labeled data in a target domain.
- Goal: Clustering/classification the target domain data.
- Assumption: The source domain and target domain data share some common features, which can help clustering and classification in the target domain.

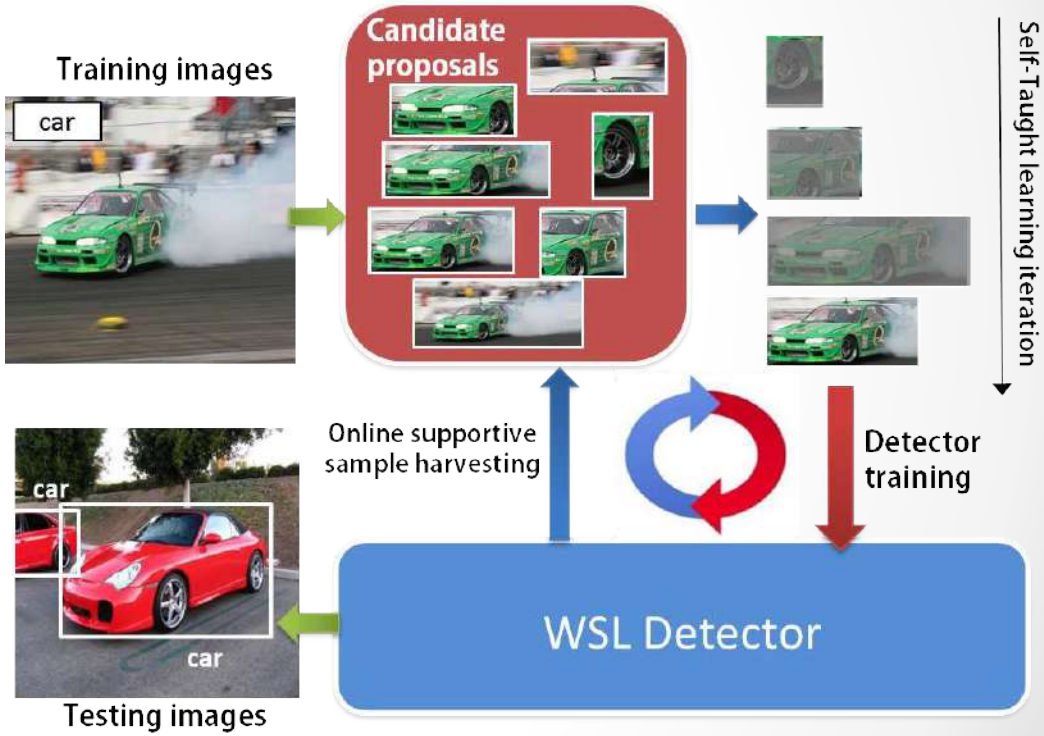
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359.

Example of Self-taught Learning



Self-Taught Learning for Weakly Supervised Object Localization

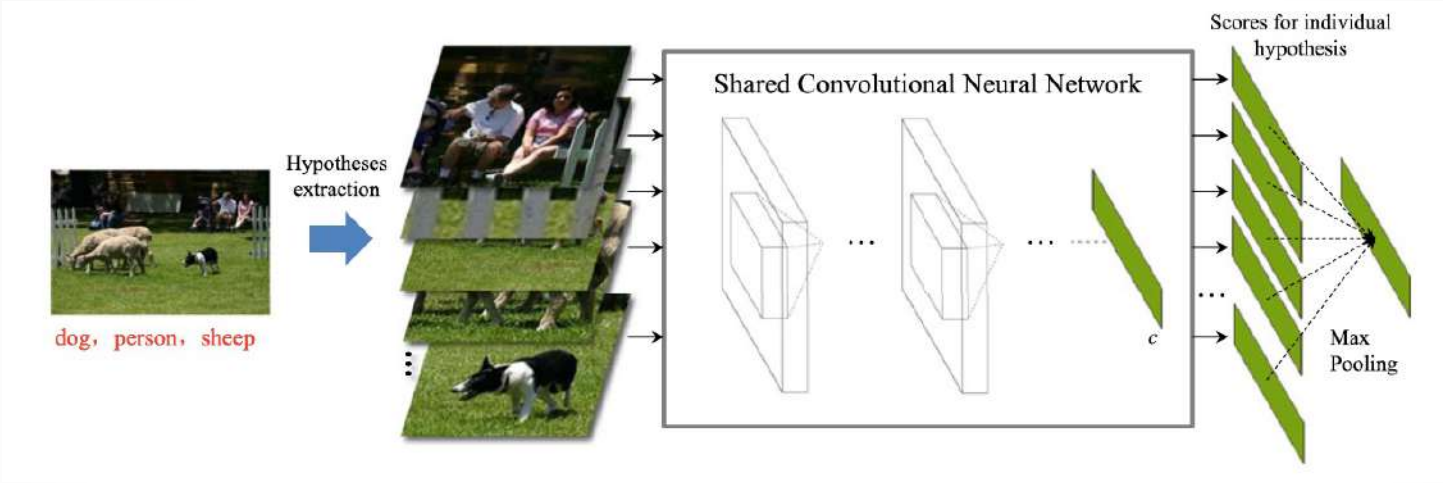
- Given image-level supervision, **seed positive proposals** are first obtained as initial positive samples for a CNN detector.
- The CNN detector is then trained with self-taught learning which **alternates between training and online supportive sample harvesting**
  - relying on the relative improvement of CNN scores predicted by the detector.



weakly supervised localization (WSL)

Jie, Zequn, et al. "Deep self-taught learning for weakly supervised object localization." CVPR. 2017.

HCP: A Flexible CNN Framework for Multi-Label Image Classification



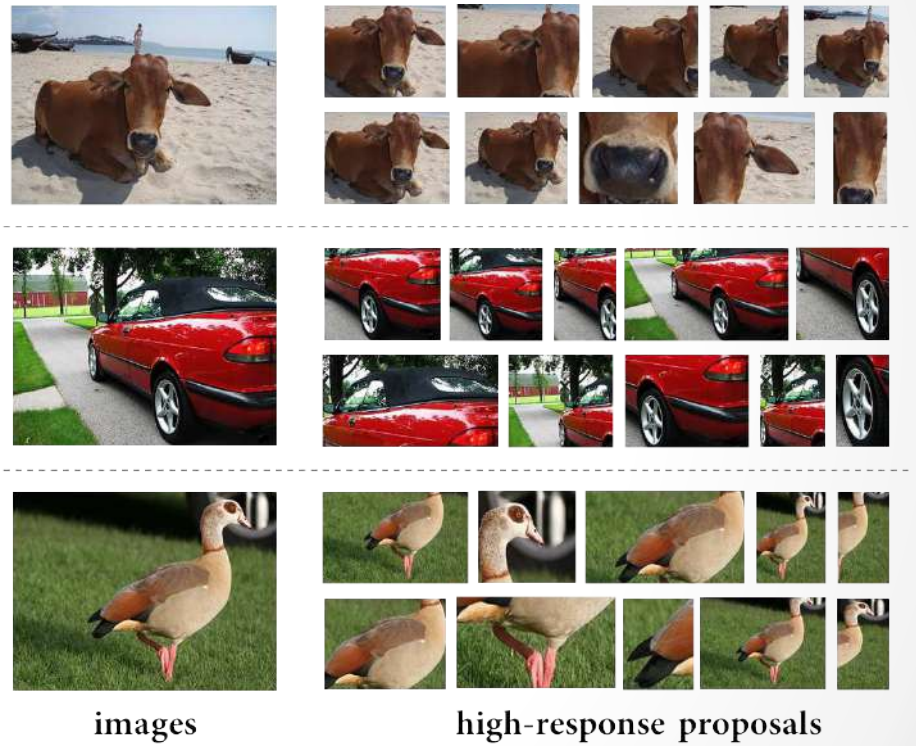
- **Hypothesis-CNN-Pooling:** For a given multi-label image**, a set of input hypotheses to the shared CNN is selected** based on the proposals generated by an objectness detection techniques.
- Feed the selected hypotheses into the shared CNN and **fuse the outputs into a c-dimensional prediction vector with cross-hypothesis max-pooling operation**.
- Finally, retrain the whole HCP to further fine-tune the parameters for multi-label image classification.

Yunchao Wei, et.al.. HCP: A flexible CNN framework for multi-label image classification PAMI, 38(9), 2016.

Self-Taught Learning for Weakly Supervised Object Localization

Image-to-Object transformation:

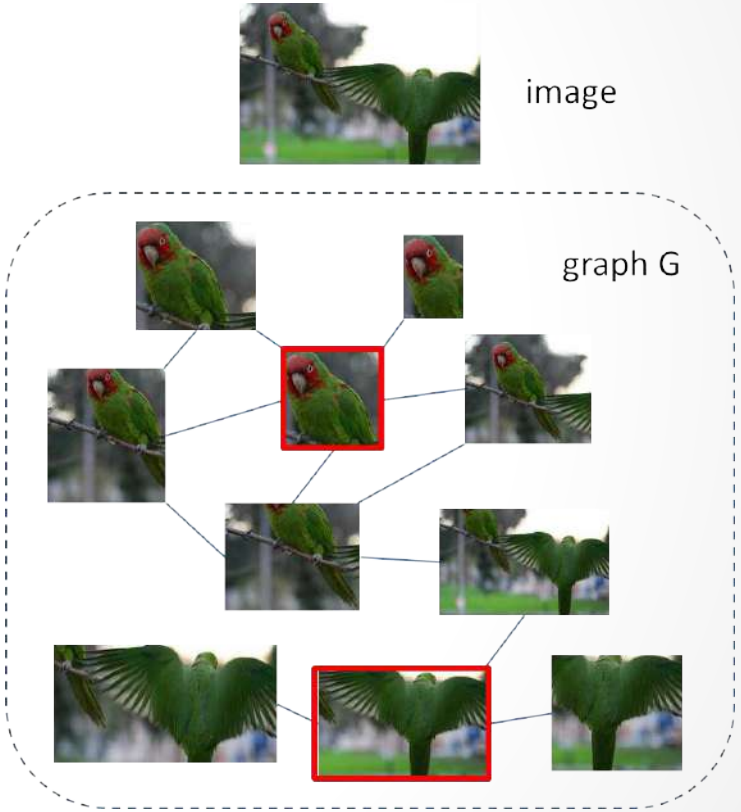
- Use HCP algorithm[1] for region proposal to detect candidate proposals with the **highest responses to represent an object.**
- **Top 10 proposals** for each image are shown.
  - The top-ranked proposals may contain context or only a key discriminative part of the object.
- These top-ranked proposals are mostly **spatially concentrated around** the true object instances.



[1]Yunchao Wei, et.al.. HCP: A flexible CNN framework for multi-label image classification PAMI, 38(9), 2016.

Self-Taught Learning for Weakly Supervised Object Localization

- **Construct a graph** G whose nodes are the proposals in the N -candidate proposal pool.
- Each candidate proposal is connected to the others with **IoU 0.5.**
- By dense subgraph discovery, X (here, 2) spatially concentrated proposals are **selected among all the proposals**, framed in red boxes.



Jie, Zequn, et al. "Deep self-taught learning for weakly supervised object localization." CVPR. 2017.

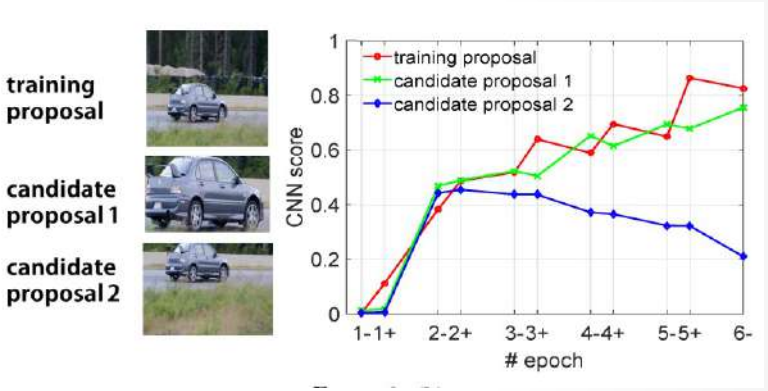
Self-Taught Learning for Weakly Supervised Object Localization

- For an image, denote
- A^t_i = the objectness predicted score for the i-th proposal at the t -th epoch.

- To compute the **Relative Improvement (RI)=** objectness score at the (t+1)-th epoch as B^{t+1}_i.

- Then among the N candidate proposals, the proposal P^*_{t+1} with the largest RI is selected for the (t+1) -th training epoch:

$$P^*_{t+1} = \arg\max_i (B^{t+1}_i - A^t_i).$$



CNN score on the target class vs. number of epochs during training Fast R-CNN for different proposals. The training proposals are the seed positive samples to train Fast R-CNN. "**1** -" and "**1** +" indicate the CNN score right before and after training on this image in the **1** st epoch, respectively.

Jie, Zequn, et al. "Deep self-taught learning for weakly supervised object localization." CVPR. 2017.

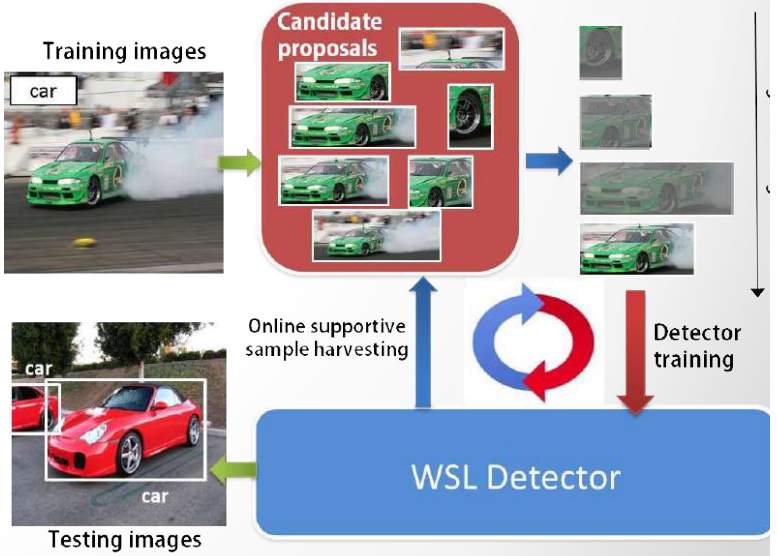
Deep Self-Taught Learning: implementation

The method performs Negative rejection after several epochs of online supportive sample harvesting.

Removes 10% samples with the minimal CNN scores and their corresponding images in the subsequent Fast R-CNN training.

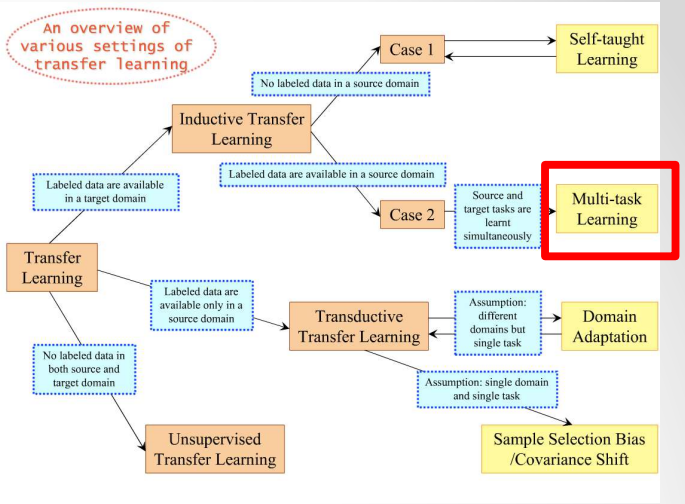
This is inspired by the observation that even the best positive samples selected from the difficult positive images are of unsatisfactory quality (low IoU to true objects).

- For data augmentation, the method:
  - Uses all the proposals in this image that overlap with the selected proposal by IoU >0.5 .
  - Negative examples: the proposals which have IoU in [0.1, 0.5) overlap with the selected proposal.



Jie, Zequn, et al. "Deep self-taught learning for weakly supervised object localization." CVPR. 2017.

Index



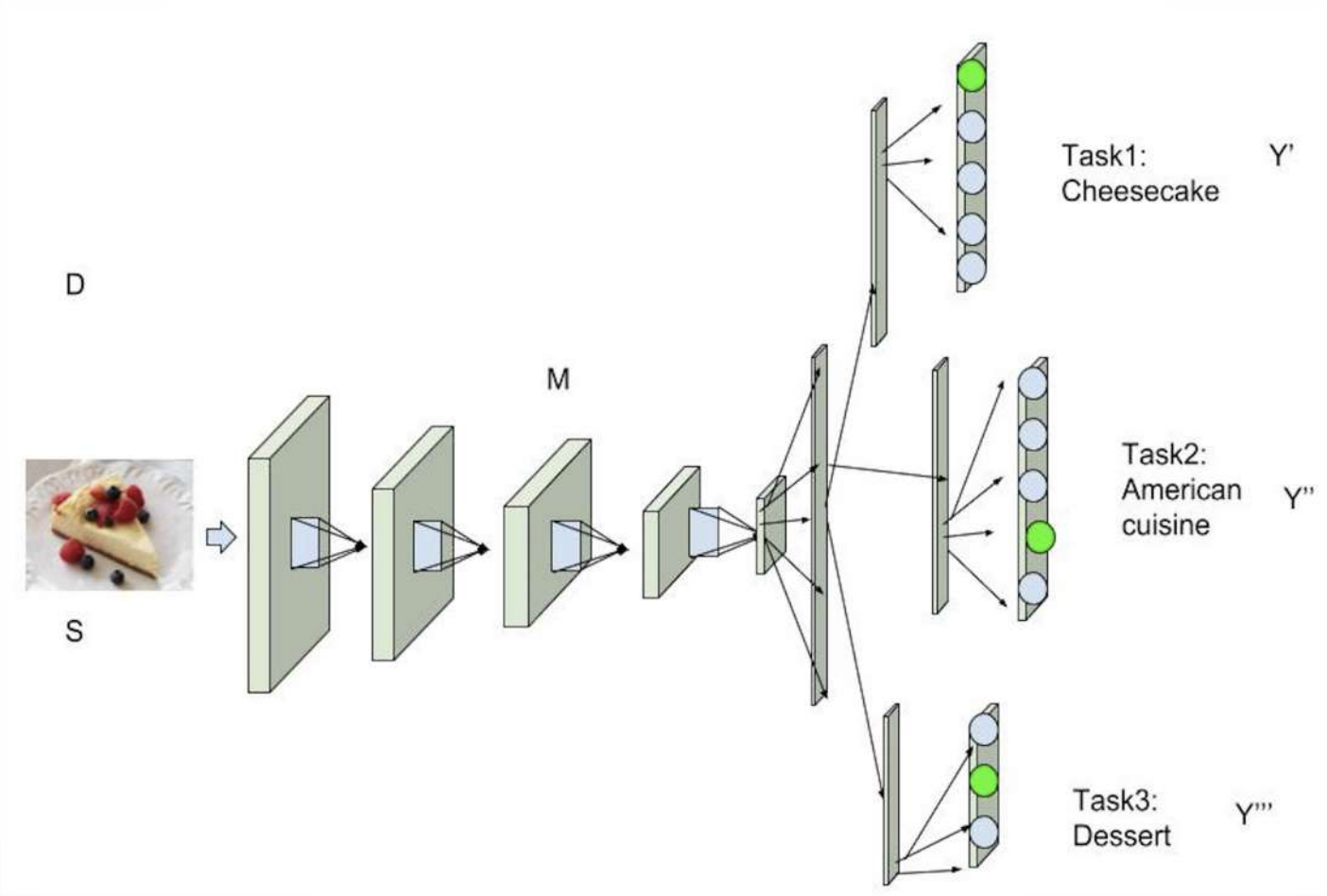
1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Domain adaptation
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Multi-Task Learning

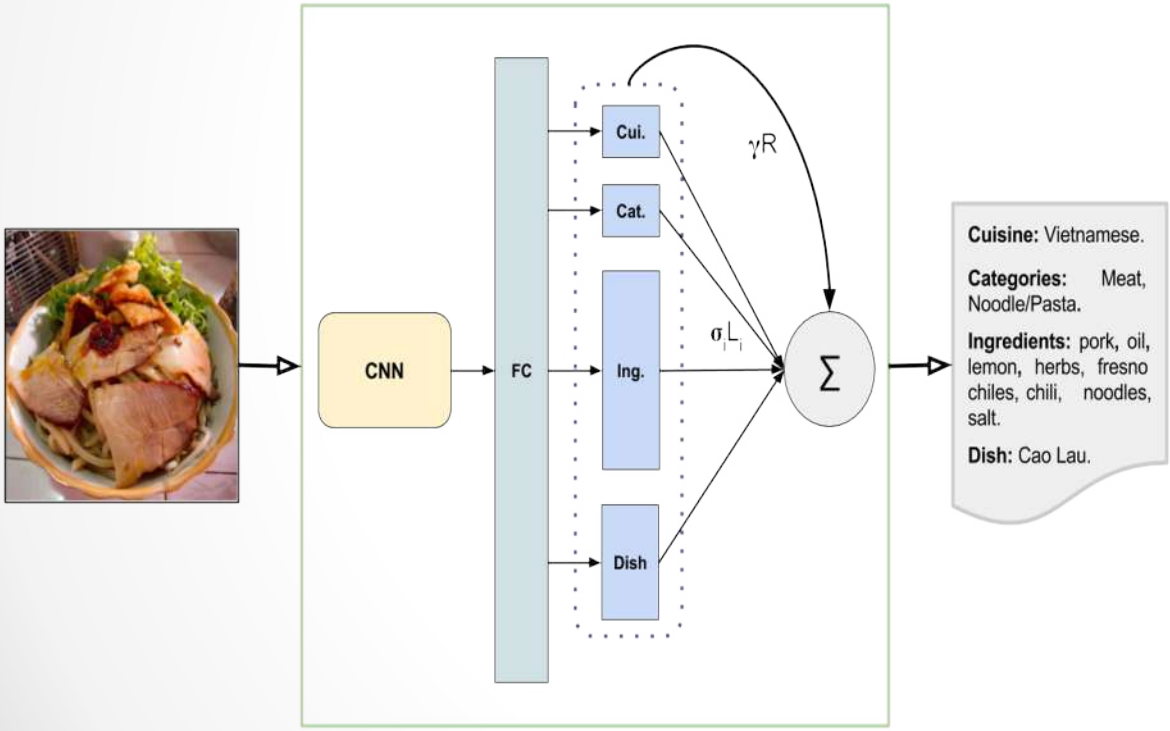
- Input: A lot of labeled data in a source domain and the target domain that coincide.
- Goal: Classifying the target and source domain data simultaneously .
- Assumption: The source domain and target domain data share some common features, which can help classifying sharing knowledge.
- Main Idea: To train simultaneously both tasks.

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Multi-Task Learning



Loss function in the MTL

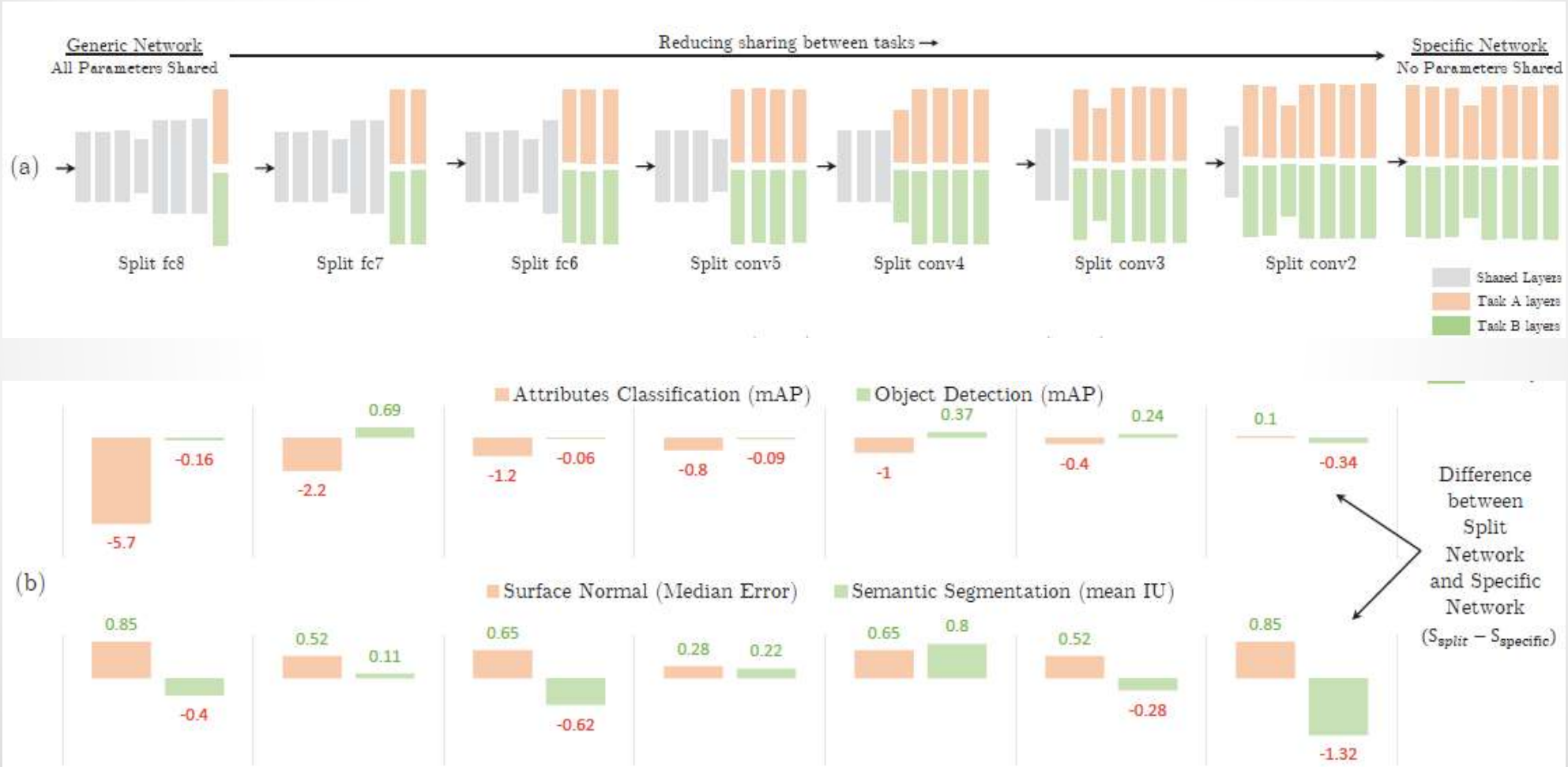


$$L_{total} = \sum_i \omega_i L_i$$

Multi-Task Learning (MTL)

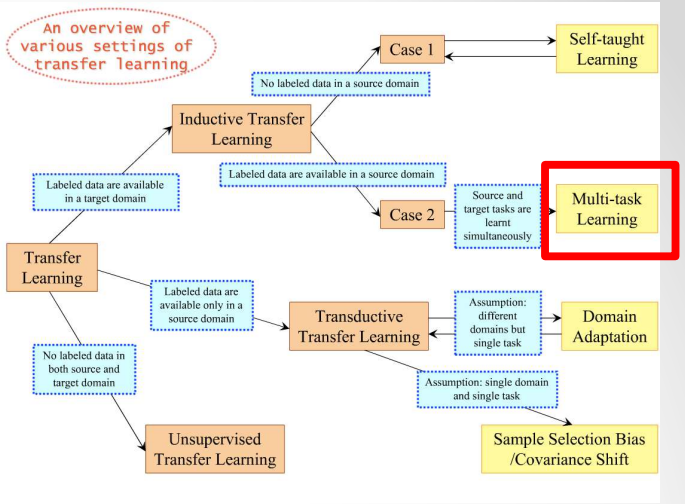
- Learning **multiple objectives** from a shared representation
  - *Efficiency* and prediction *accuracy*.
- Crucial importance in systems where **long computation** run-time is prohibitive
  - Combining all tasks *reduces computation*.
- Inductive **knowledge transfer**
  - *Generalization* by sharing the domain information between complimentary tasks.

MTF possible architectures



I. Misra, A. Shrivastava A. Gupta, M. Hebert, "Cross-stitch Networks for Multi-task Learning", CVPR'2016

Index



1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Domain adaptation
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Transductive Transfer Learning: Domain Adaptation

- Input: A lot of labeled data in D_S and only unlabeled data in D_T.
- Assumption: Single task across domains, which means $P(Y_S|X_S)$ and $P(Y_T|X_T)$ are the same while $P(X_S)$ and $P(X_T)$ may be different.
- Output: A common representation between D_S and D_T and a model on the new representation for use in D_T.
- Main Idea: Find a "good" feature representation that reduces the "distance" between both domains.

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Making representations more similar

- To improve transferability, look for **representations as similar as possible btw both domains.**
  - The model should not take into account domain-specific characteristics that may hinder transfer but the commonalities between the domains.
- Apply as
  - Pre-processing step to the data representatons, or
  - Encourage representations of the domains to be more similar to each other.

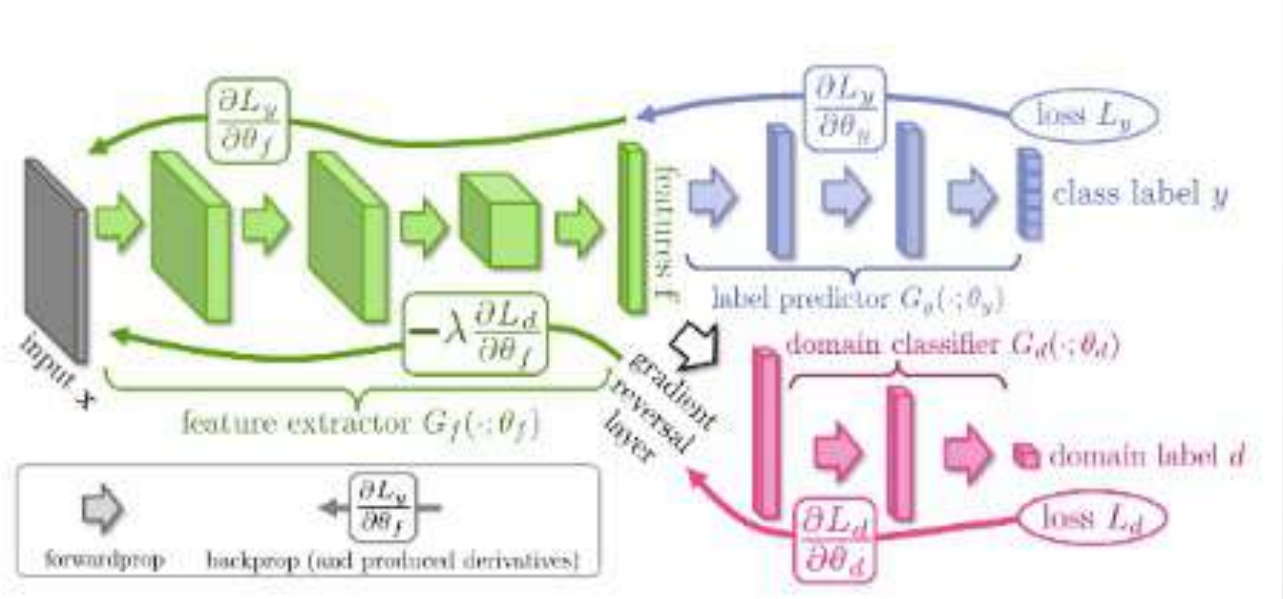
Confusing domains

A way to ensure similarity between the representations of both domains:

- add another objective to an existing model that encourages it to confuse the two domains.

- This domain confusion loss is a regular classification loss where the model tries to predict the domain of the input example.

- The difference to a regular loss: the gradients that flow from the loss to the rest of the network are reversed.

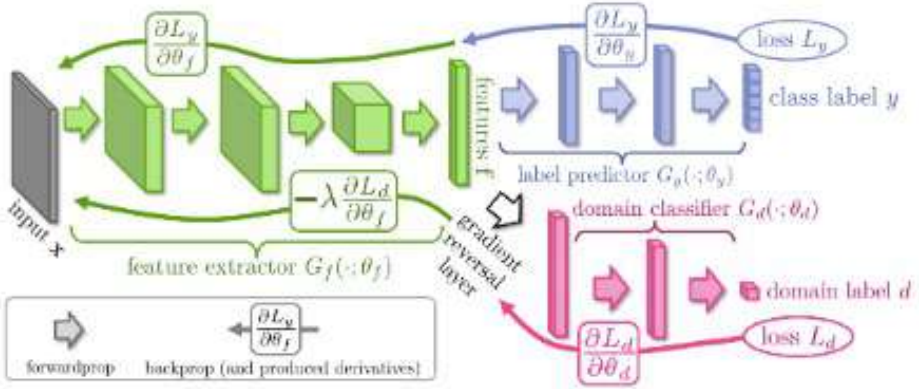


Confusing domains

Hypothesis: for effective domain transfer to be achieved, predictions must be made based on features that cannot discriminate between the training (source) and test (target) domains.

The proposed architecture includes:

a deep feature extractor (green) and a deep label predictor (blue), which together form a standard feed-forward architecture.



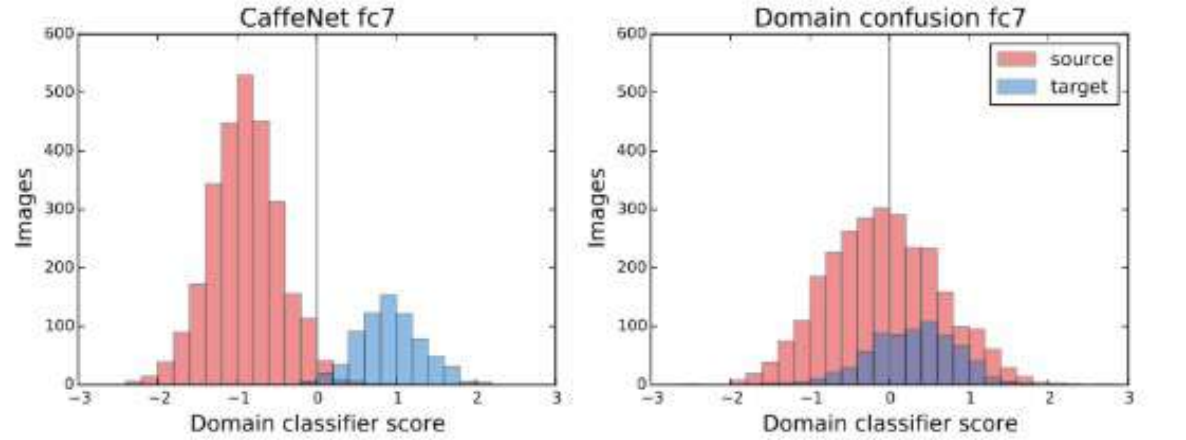
Unsupervised domain adaptation is achieved by adding a domain classier (red) connected to the feature extractor via a **Gradient Reversal Layer** that multiplies the gradient by a certain negative constant during the backpropagation-based training.

The training minimizes:

- the label prediction loss (for source examples) and
- the domain classication loss (for all samples).

Confusing domains

While a model trained only with the regular objective is to be clearly able to separate domains based on its learned representation, a model whose objective has been augmented with the domain confusion term is unable to do so.



- Domain classifier score of a regular and a domain confusion model (Tzeng et al, 2015)

Domain Separation Networks

- Existing approaches for domain adaptation:
  - Map a domain to another, or
  - Extract features invariant to both domains.
  - However, they ignore individual features of both domains.

- **Proposal**: explicitly modelling what is unique help construct invariant features.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain Separation Networks. NIPS. ↩

Domain Separation Networks

Hypothesis:

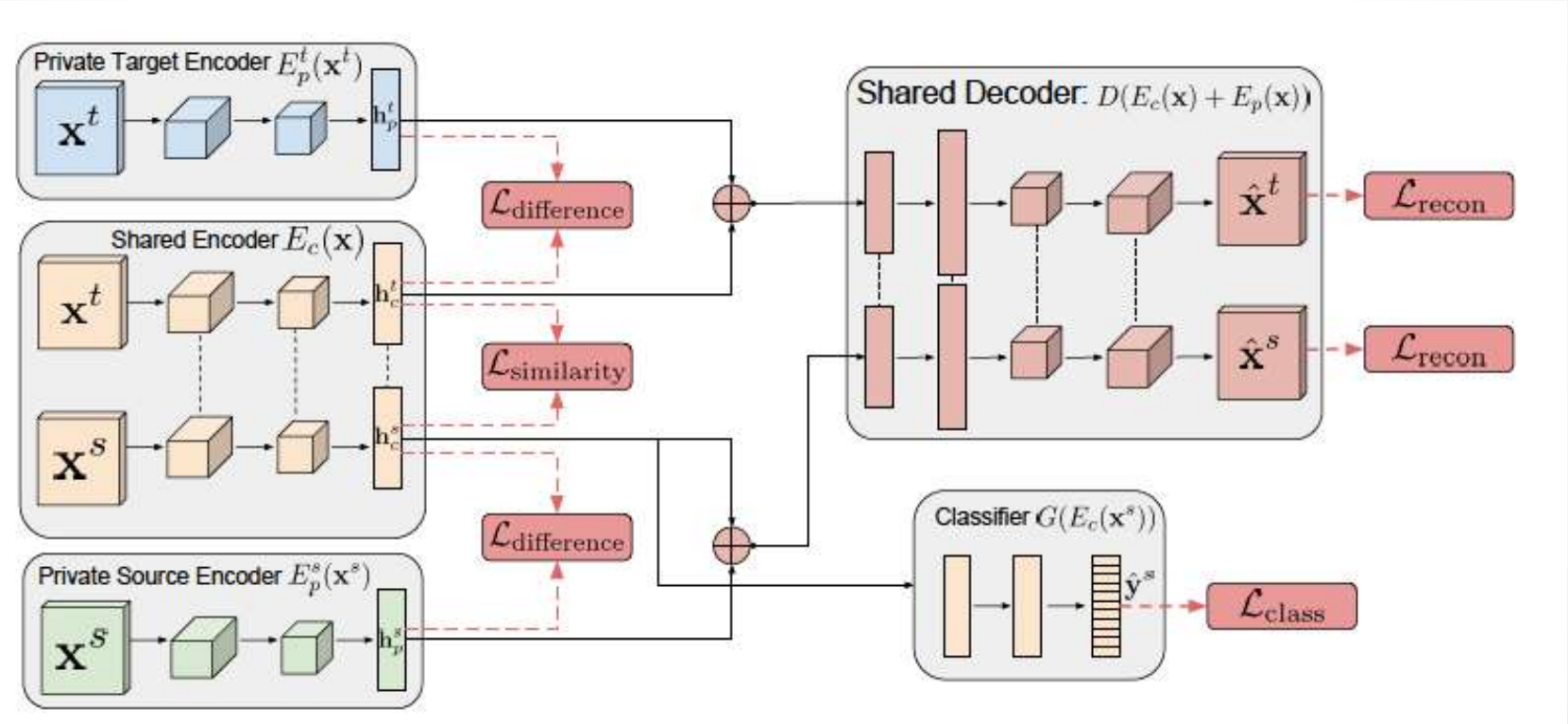
S & T differ mainly in terms of the low-level image features (illumination, colour) distributions but high-level parameters (objects, 3D pose) have similar distributions and the same label space.

- Extract image representations partitioned into 2 subspaces:
  - Particular to each domain
  - Shared across domains.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain Separation Networks. NIPS. ↩

Domain Separation Networks

- Cross-domain task



They assume the target domain is unlabelled, the loss is applied only to the source domain.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain Separation Networks. NIPS. ↩

Domain Separation Networks

- Final Loss function:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \alpha\,\mathcal{L}_{\text{recon}} + \beta\,\mathcal{L}_{\text{difference}} + \gamma\,\mathcal{L}_{\text{similarity}}$$

Classification loss

Assures data are reconstructed

Separates domain-specific vs domain-generic

Assures domain-generic features

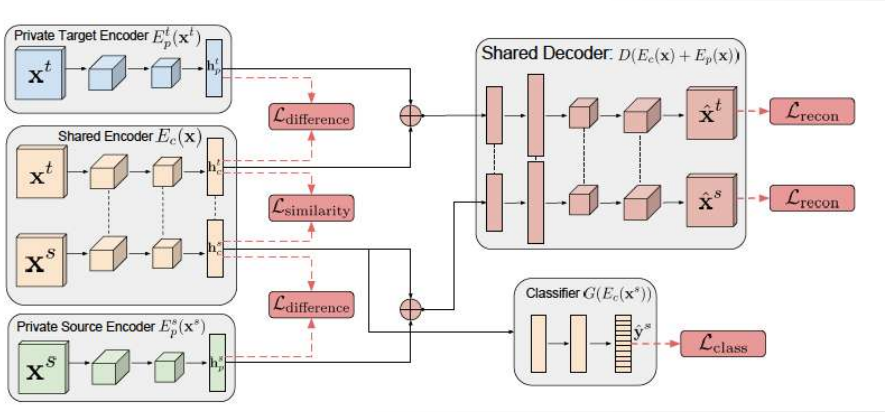
$$\mathcal{L}_{\text{task}} = -\sum_{i=0}^{N_s} \mathbf{y}_i^s \cdot \log \hat{\mathbf{y}}_i^s,$$

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{N_s} \mathcal{L}_{\text{si\_mse}}(\mathbf{x}_i^s, \hat{\mathbf{x}}_i^s) + \sum_{i=1}^{N_t} \mathcal{L}_{\text{si\_mse}}(\mathbf{x}_i^t, \hat{\mathbf{x}}_i^t)$$

$$\mathcal{L}_{\text{si\_mse}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{k}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 - \frac{1}{k^2}([\mathbf{x} - \hat{\mathbf{x}}] \cdot \mathbf{1}_k)^2,$$

$$\mathcal{L}_{\text{difference}} = \left\|\mathbf{H}_c^{s\top}\mathbf{H}_p^s\right\|_F^2 + \left\|\mathbf{H}_c^{t\top}\mathbf{H}_p^t\right\|_F^2,$$

$$\mathcal{L}_{\text{similarity}}^{\text{DANN}} = \sum_{i=0}^{N_s+N_t} \left\{ d_i \log \hat{d}_i + (1 - d_i)\log(1 - \hat{d}_i) \right\}.$$



Zero-shot learning

Aims to learn from **only a few, one or even zero instances of a class, we arrive at few-shot, one-shot, and zero-shot learning** respectively.

Enabling models to perform one-shot and zero-shot learning is admittedly among the **hardest problems** in machine learning.

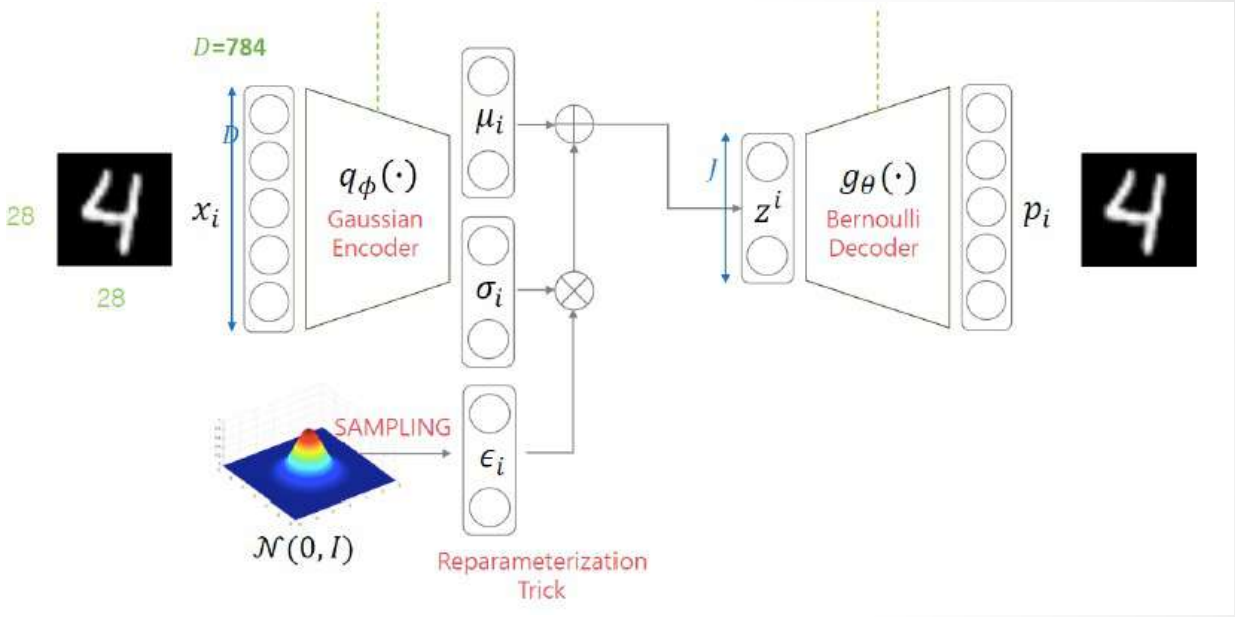
Recent advances:

- **models need to be trained explicitly to perform one-shot learning** in order to achieve good performance at test time,
- **zero-shot learning**: training classes are present **at test time**

https://ruder.io/transfer-learning/index.html#fn44

Variational Autoencoder

Objective : Minimize reconstruction error + regularization loss

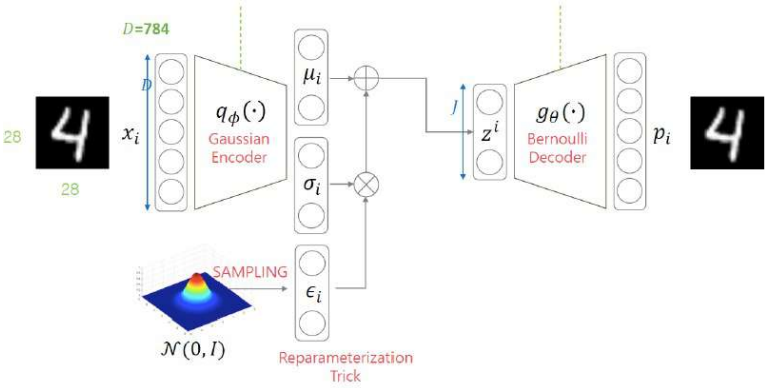


Approximated by finding its closest proxy posterior, **q(z|x)**, through minimizing their distance using a variational lower bound limit.

Schonfeld, Edgar, et al. "Generalized zero-and few-shot learning via aligned variational autoencoders." Proceedings of the IEEE Conf CVPR. 2019.

Variational Autoencoder

- The objective function of a VAE is the variational lower bound on the marginal likelihood of a given datapoint:



$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z))$$

the first term is the reconstruction error and the second term is the unpacked Kullback-Leibler divergence between the inference model **q(z|x)** and **p(z).**

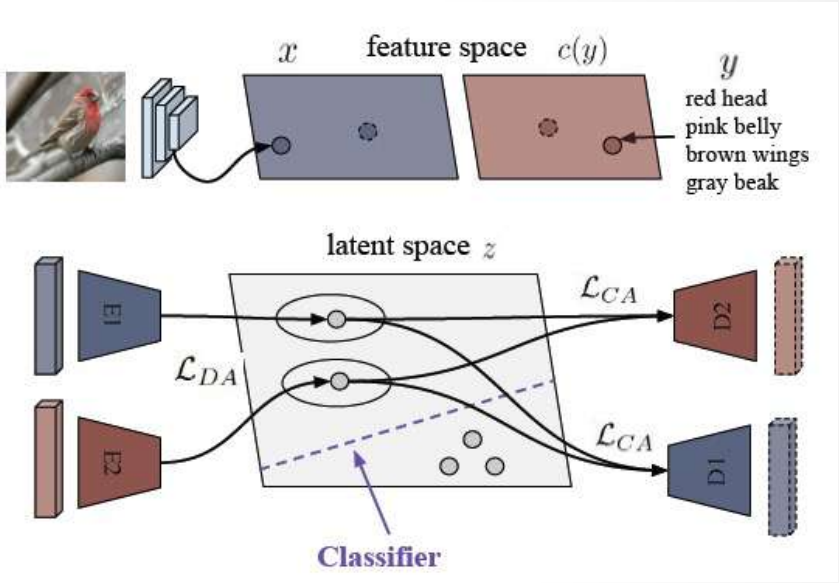
A common choice for the prior is a multivariate standard Gaussian distribution.

Schonfeld, Edgar, et al. "Generalized zero-and few-shot learning via aligned variational autoencoders." Proceedings of the IEEE Conf CVPR. 2019.

Zero-shot learning

Generalized zero-shot learning rely on *cross-modal mapping between the image feature space and the class embedding space*.

As labeled images are expensive, one direction is to *augment the dataset by generating either images or image features.*



Proposal: a shared latent space of image features and class embeddings is learned by modality-specific aligned variational autoencoders.

- Requires discriminative information about the image and classes in the latent features, on which we train a softmax classifier.

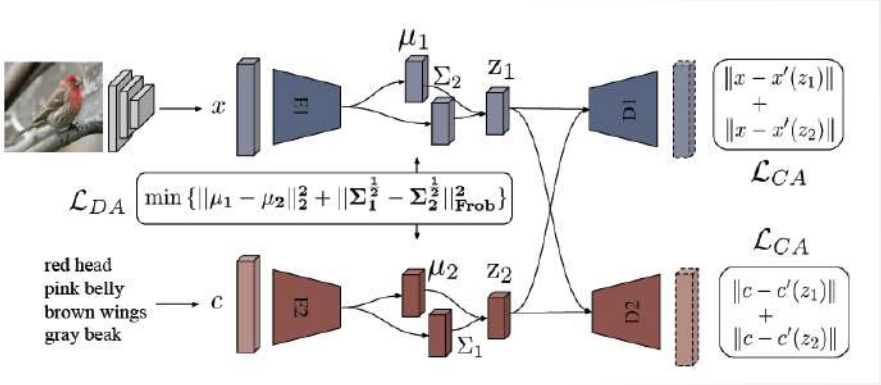
- Aligns the distributions earned from images and from side-information to construct latent features that contain the essential multi-modal information associated with unseen classes.

Schonfeld, Edgar, et al. "Generalized zero-and few-shot learning via aligned variational autoencoders." Proceedings of the IEEE Conf CVPR. 2019.

Cross- and Distribution-Aligned VAE

- It can include M encoders corresponding to M different modalities

$$\mathcal{L}_{VAE} = \sum_{i}^{M} \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x^{(i)}|z)] - \beta D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z))$$



- **Cross-Aligned Loss**

-minimize difference btw modalities

$$\mathcal{L}_{CA} = \sum_{i}^{M} \sum_{j \neq i}^{M} |x^{(j)} - D_j(E_i(x^{(i)}))|.$$

Distribution-Alignment Loss:

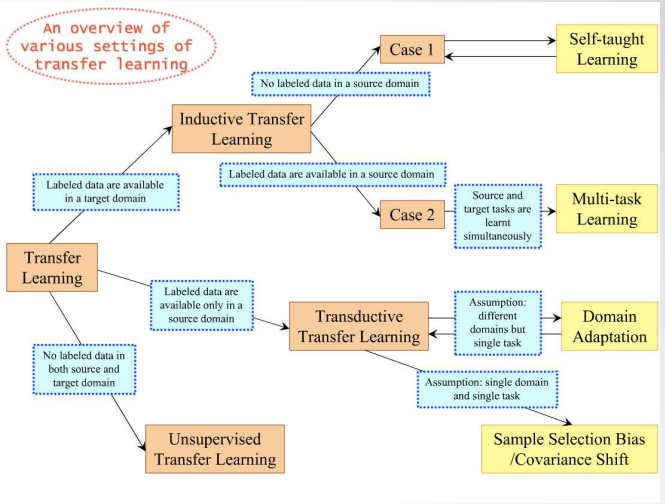
-minimize difference btw distributions

$$W_{ij} = [||\mu_i - \mu_j||_2^2 + Tr(\Sigma_i) + Tr(\Sigma_j) - 2(\Sigma_i^{\frac{1}{2}} \Sigma_i \Sigma_j^{\frac{1}{2}})^{\frac{1}{2}}]^{\frac{1}{2}}.$$

- **Total Loss**

$$\mathcal{L}_{CADA-VAE} = \mathcal{L}_{VAE} + \gamma \mathcal{L}_{CA} + \delta \mathcal{L}_{DA}$$

Index



1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Sample Selection Bias

- To correct sample selection bias:

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n_S} \frac{\mathcal{P}(x_{T_i})}{\mathcal{P}(x_{S_i})} l(x_{S_i}, y_{S_i}, \theta)$$

weights for source domain data

- How to estimate ($P(X_T)/P(X_S)$)?
  - One straightforward solution is to estimate $P(X_S)$ and $P(X_T)$, respectively.

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Index

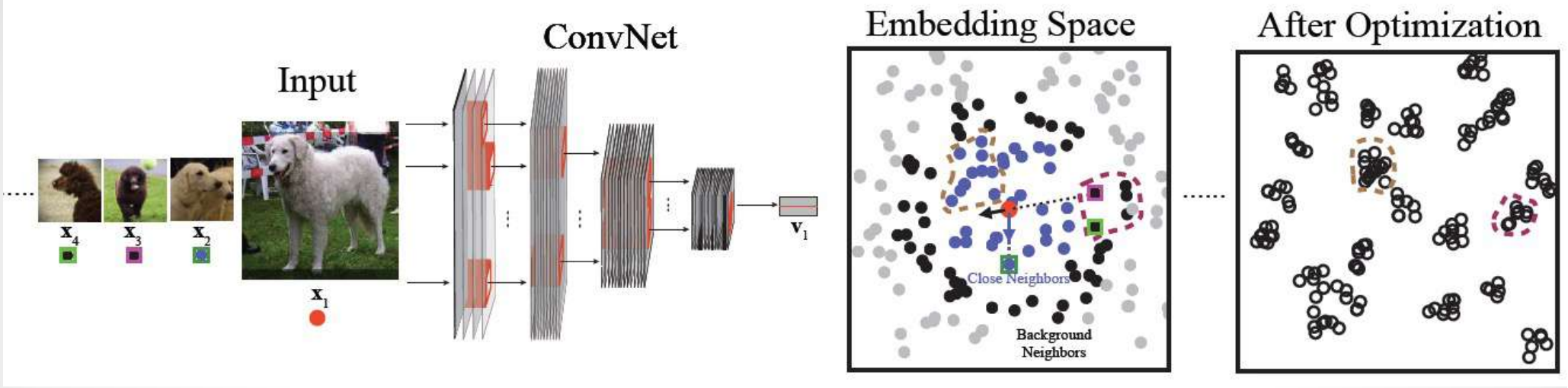
1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Unsupervised learning

- Assumption: No labels and distributions are known in both D_S and D_T.
- Main Idea: Find a "good" feature representation that reduces the "distance" between domains.
- Input: A lot of unlabeled data in D_S and D_T.
- Output: A common representation between D_S and D_T and a model on the new representation for better clustering of both.

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Local Aggregation for Unsupervised Learning of Visual Embedding



Deep neural network to embed it into a lower dimension space (”Embedding Space” panel).

Identify its close neighbours (blue dots) and background neighbours (black dots).

The optimization seeks to push the current embedding vector (red dot) closer to its close neighbours and further from its background neighbours.

Zhuang et.al.. "Local aggregation for unsupervised learning of visual embeddings." *CVPR*. 2019.

Index

1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
   - Self-taught learning
   - Multi-task learning
   - Sample selection Bias
   - Unsupervised learning
   - Negative transfer
4. Application of Transfer Learning to food recognition
5. Conclusions

Negative Transfer

- Most approaches to transfer learning assume transferring knowledge across domains be always positive.
- However, in some cases, when two tasks are too dissimilar, brute-force transfer may even hurt the performance of the target task, which is called negative transfer.
- Some researchers have studied how to measure relatedness among tasks.
- How to design a mechanism to avoid negative transfer needs to be deeply studied theoretically and practically.

- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Index

1. Definition of Transfer Learning
2. Scenarios of Transfer Learning
3. Transductive vs Inductive learning – taxonomy
4. Application of Transfer Learning to food recognition
5. Conclusions

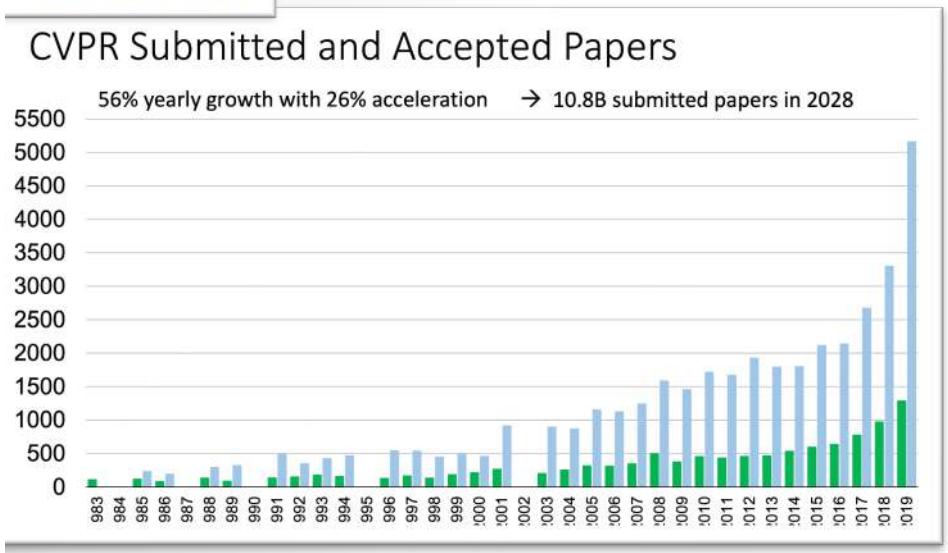
Why food recognition?

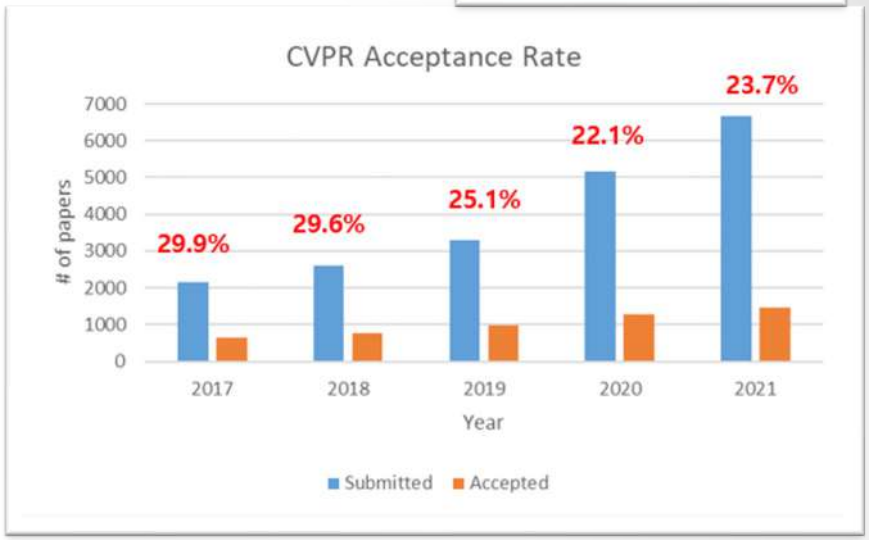
Did you know that:

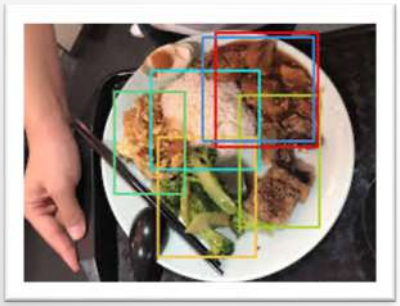
- **180 million photos** with the hashtag #food on Instagram
- **90 new photos** hash-tagged #foodporn are uploaded to Instagram **every minute**.
- **54%** of 18–24 year olds take a food photo while eating out,
- **39%** have posted it somewhere online.
- **5% of over-50s** share food snaps on forums as Facebook & Twitter

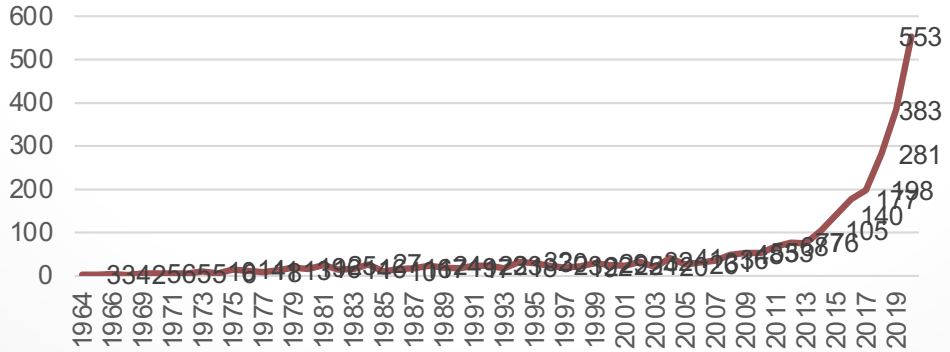
"**Camera eats first**"

CVPR evolution

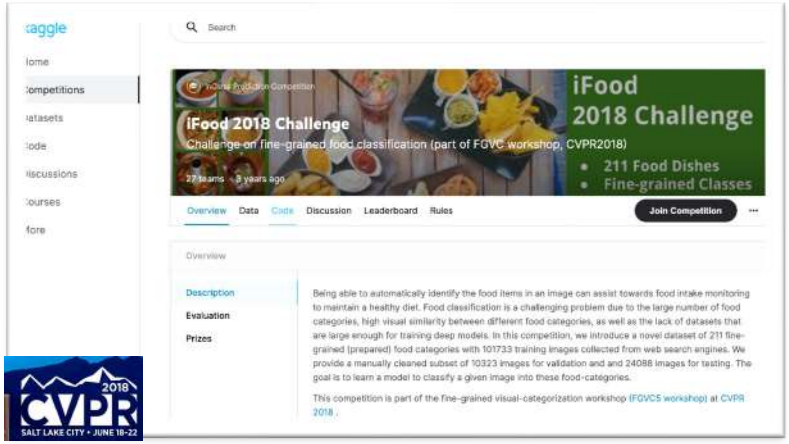




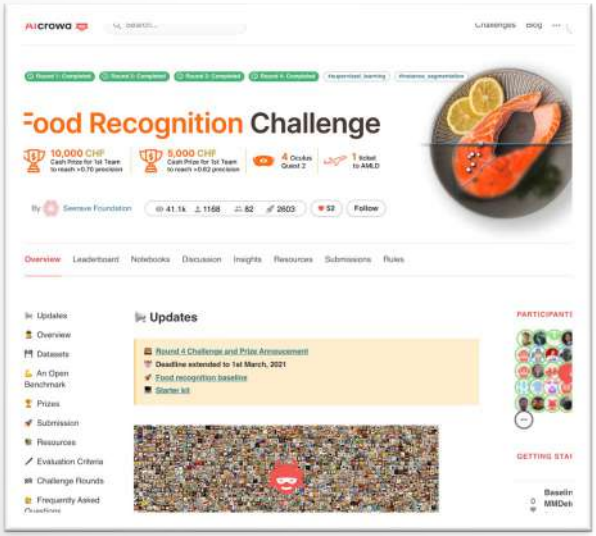




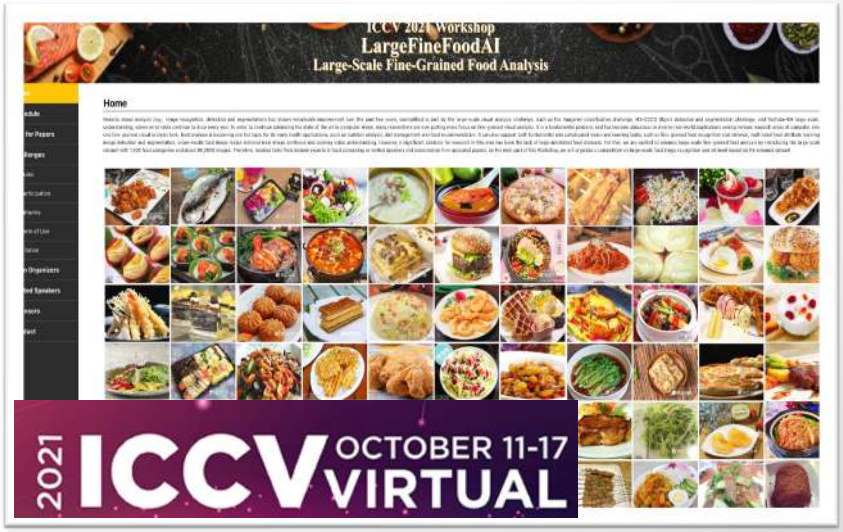
Food recognition popularity



iFood
211 fine-grained (prepared) food categories with 135733



AICrowd
26000 annotated segmented images



LargeFineFoodAI
1,000 fine-grained food categories and over 50,000 images.

Why is the food recognition a challenge?

Difficulties

Huge intra-class variations

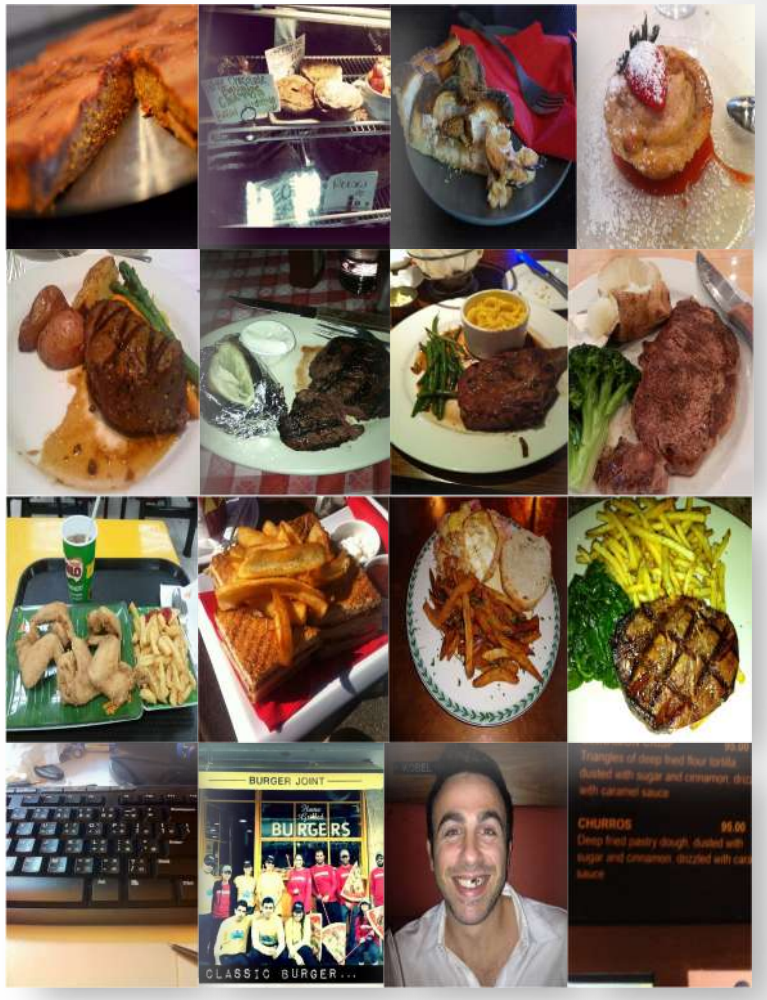
Ambiguous definition

Inter-class similarities

Mixed items

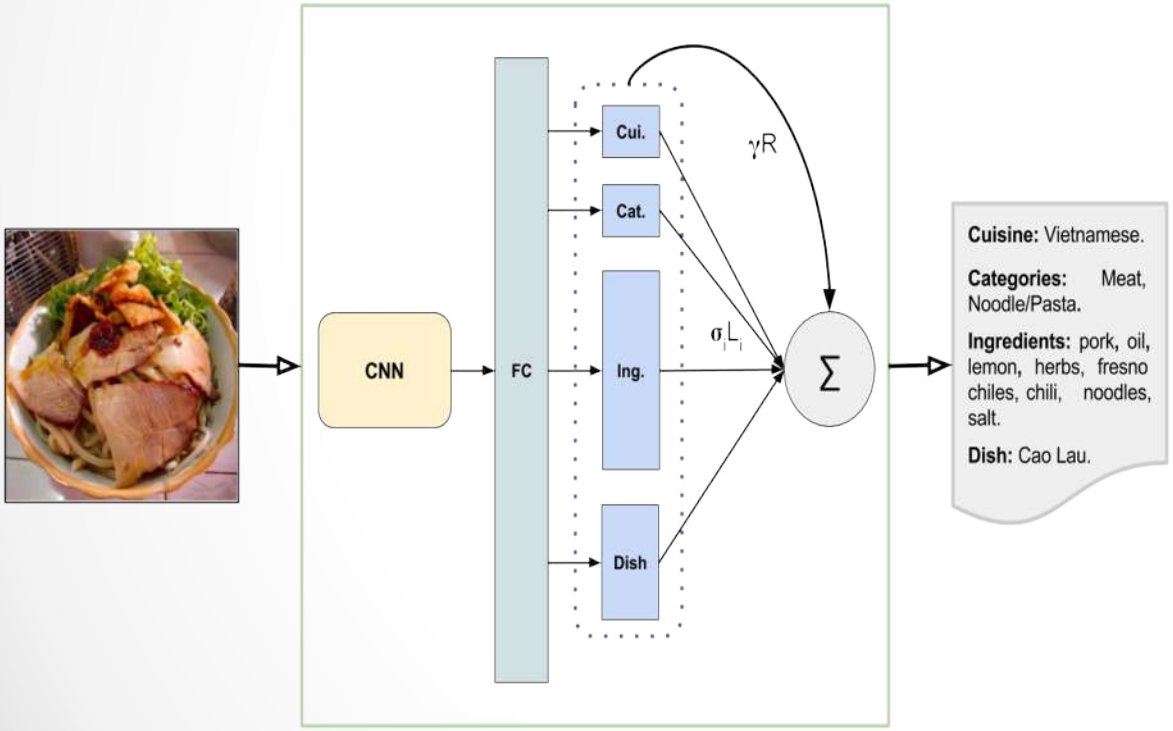
Need of huge datasets

Bad Labeled



What to do when you have a really complicate problem?

Food Recognition as a MTL



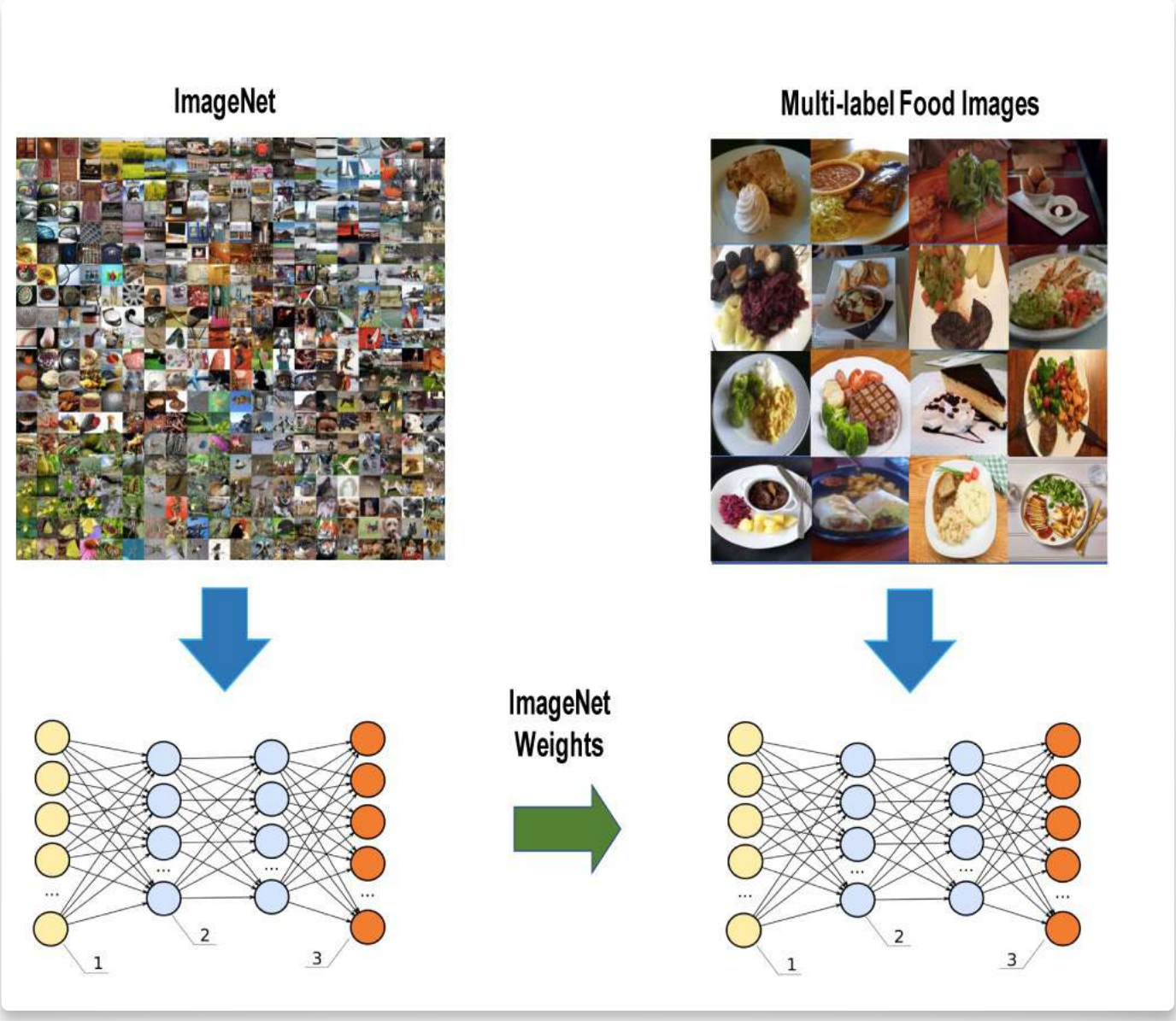
$$L_{total} = \sum_i \omega_i L_i$$

How to define the importance of each task?

- Weighted uniformly the losses.
- Manually tuned the losses.
- Dynamic weighted of the losses.
  - The main task is fixed and weights are learned for each side-task ([1]).
  - Weight the tasks according to the homoscedastic uncertainty ([2]).

[1] X. Yin and X. Liu. Multi-task convolutional neural network for face recognition.
[2] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics.

Transfer learning in Food Recognition



Negative transfer

- Negative transfer: when source domain data and task contribute to reduced performance of learning in the target domain

- Causes:
  - Domains are too dissimilar [RMKD05]
  - Tasks are not well-related [BH03], etc.

- Similarity measures
  - Cosine similarity
  - Kullback-Leibler divergence
  - Jensen-Shannon divergence
  - Maximum Mean Discrepancy [BGR+02], etc.

Single- vs Multi-Label FR

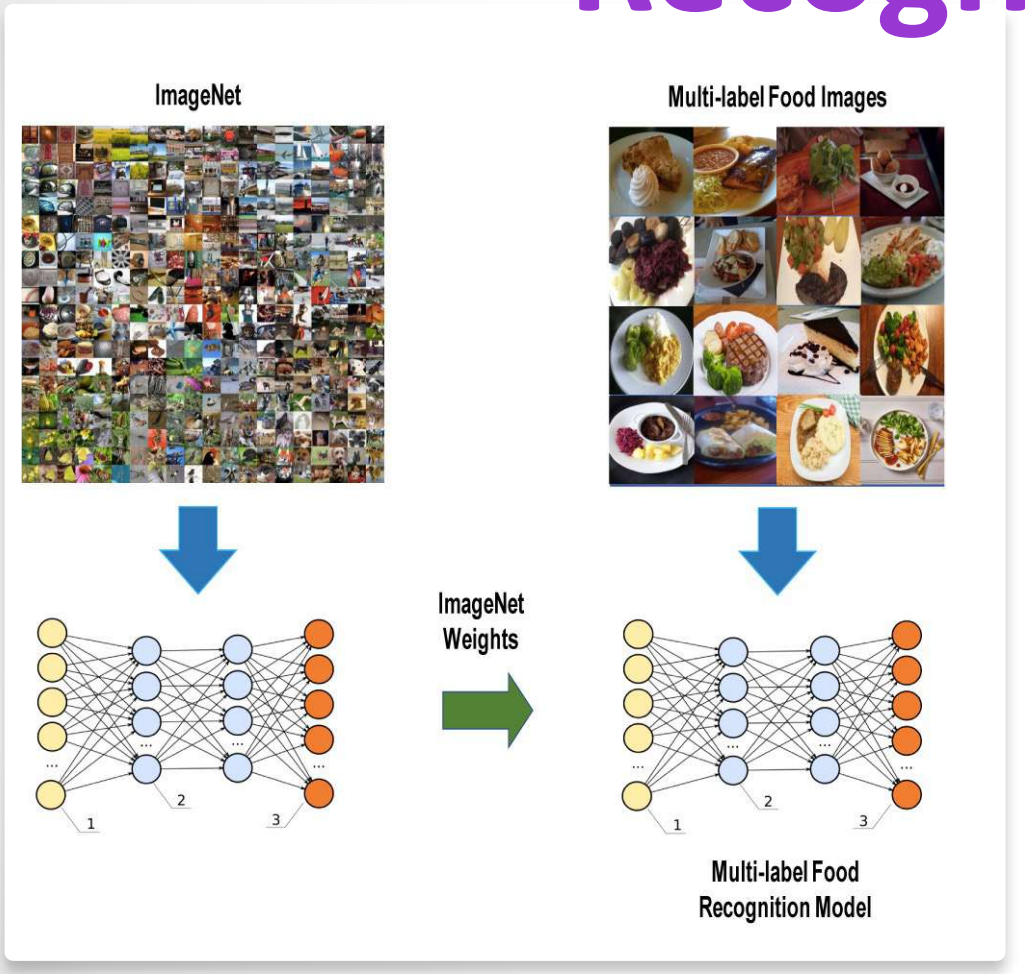
- Food-101 Food-201 our Combo-plates

Single-Label Recognition always has better performance than Multi-label Rec.

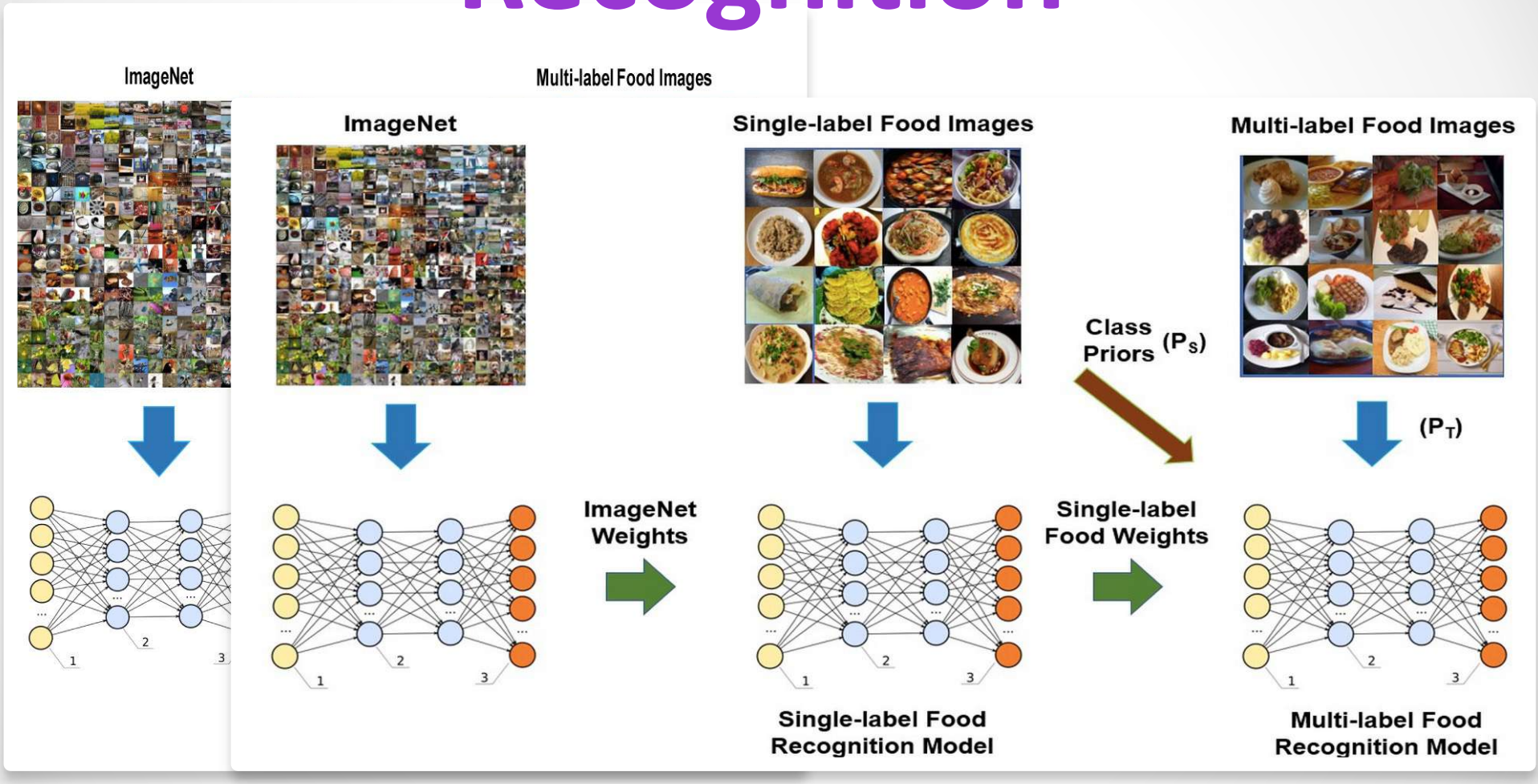
Most public datasets for FR are single-label.

However, we used to eat multiple food (so real FR should be a multi-label FR).

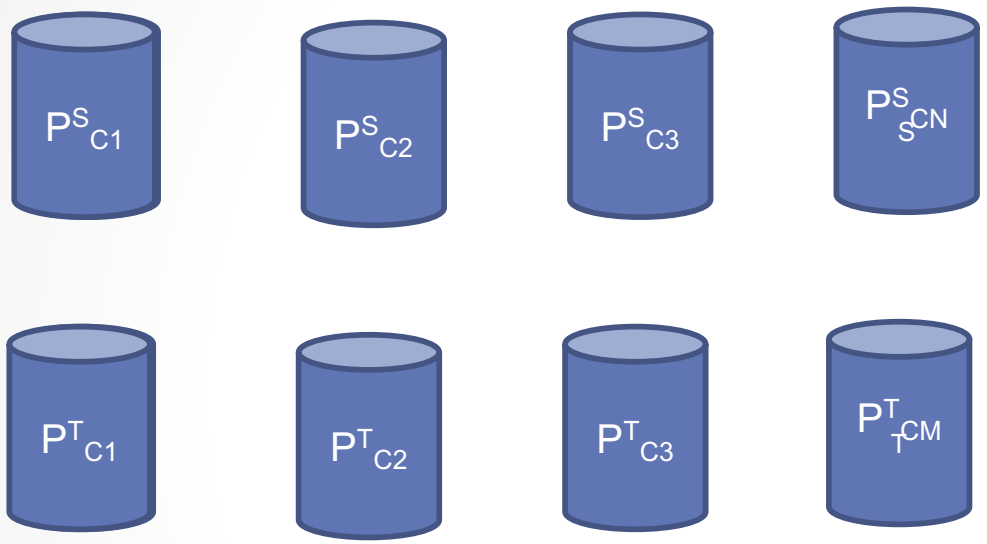
Single-to-Multi-Label Recognition



Single-to-Multi-Label Recognition



Class Priors for Improved Transferability



P^S is the prior of the source domain classes

P^T is the prior of the target domain classes

- The ratio of priors, P^T/P^S, can be used to enhance the transferability of the domain knowledge.

- Bhalaji Nagarajan, Eduardo Aguilar, Petia Radeva, S2ML-TL Framework for Multi-label Food Recognition. ICPR (5) 2020: 629-646

S2ML-TL Framework

- Prior Computation: $P(T_i) = \frac{1}{N_t} \sum_{n=1}^{N_t} y_i^n$

$$P(S_i \quad \frac{1}{N_s} \sum_{i} \sum_{j=1}^{N_s} y_j,\ j \in \{all\ source\ classes\ containing\ i\}$$

- During TL, the training of the target domain starts with initialization of weights from the source domain.

$$r_i^b = \beta \left[\alpha \frac{P_T}{P_S} + (1 - \alpha) P_T \right]$$

- Prior-induced ML loss function:

$$l_p(x, y) = \frac{1}{C} \sum_{c=1}^{C} [\ y_i.log(p(y_i)) + (1 - y_i).log(1 - p(y_i))\ ] * r_i^b$$

Validation

Food datasets

Food256: 25.600 images (100 images/class)

Classes: 256

Food101 – 101.000 images (1000 images/class)
Classes: 101

UECFood
Classes: 256

ChinaFood
Classes: 1000

Food DB	ImageNet	Future Food DB
150.000 images 231 categories	1.400.000 images 1000 categories	????? images 200.000 categories

09:56

FoodImageNet

- Food – 550 dishes, 11 categories, 11 cuisines
- Ingredients – 65
- Drinks – 40

In total:

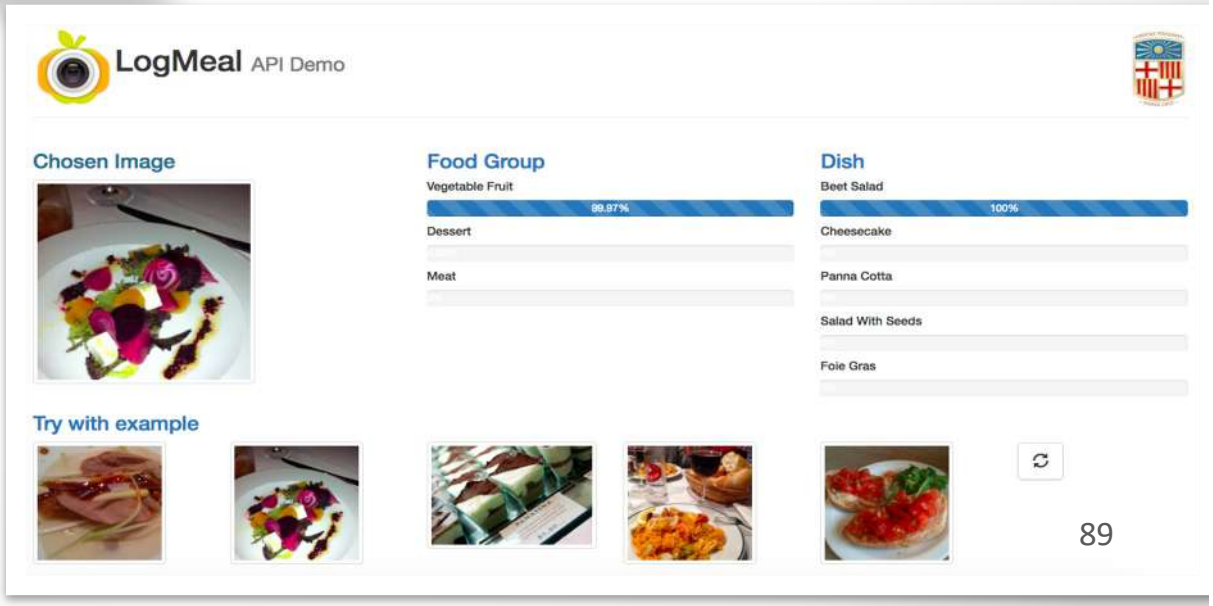
more than

550.000 images

Eduardo Aguilar, Marc Bolaños, Petia Radeva: **Regularized uncertainty-based multi-task learning model for food analysis.** J. Visual Communication and Image Representation 60: 360-370 (2019)

Food ingredients recognition

[Food category and class recognition]



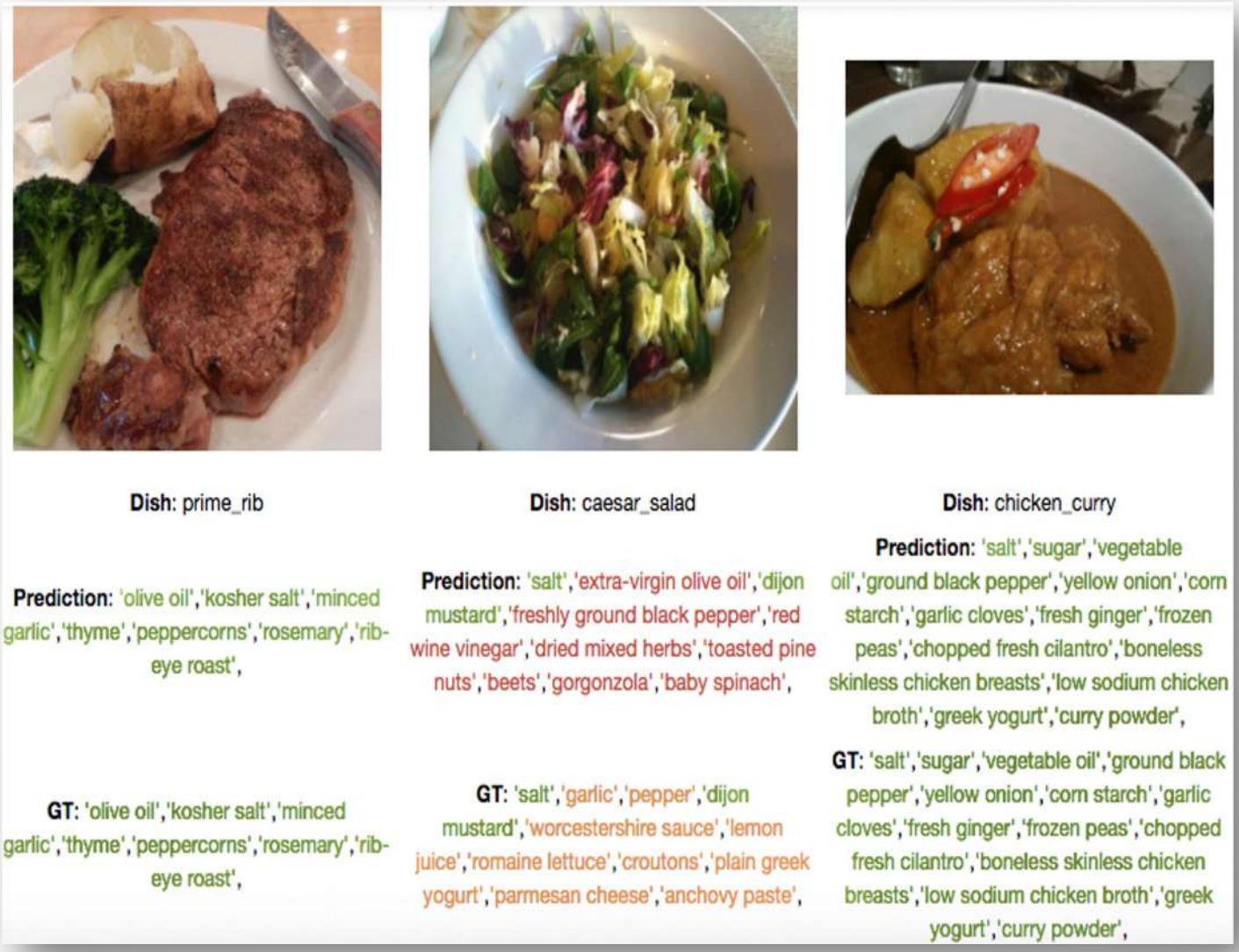
MAFood

- Food – 550 dishes, 11 categories, 11 cuisines
- Ingredients – 65
- Drinks – 40

In total:
more than
550.000 images

Eduardo Aguilar, Marc Bolaños, Petia Radeva: Regularized uncertainty-based multi-task learning model for food analysis. J. Visual Communication and Image Representation 60: 360-370 (2019)

Food ingredients recognition



- Aguilar, Eduardo, Marc Bolaños, and Petia Radeva. "Regularized uncertainty-based multi-task learning model for food analysis." *Journal of Visual Communication and Image Representation* 60 (2019): 360-370.

Validation

Model	Validation data			Test data		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Standard TL	0.7209	0.5865	0.6468	0.7152	0.5840	0.6430
ERM	0.6991	0.5667	0.6260	0.6900	0.5700	0.6200
KL	0.7030	0.6212	0.6596	0.6984	0.6173	0.6553
Priors	0.7045	0.6229	**0.6612**	0.7000	0.6200	**0.6600**

Model performance of InceptionResnetV2 on Combo-plates

Model	Validation data			Test data		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Standard TL	0.7204	0.4215	0.5319	0.7322	0.4636	0.5678
ERM	0.7518	0.4546	0.5666	0.7493	0.4800	0.5852
KL	0.7918	0.4317	0.5587	0.7740	0.4370	0.5586
Priors	0.7767	0.5877	**0.6691**	0.7313	0.5400	**0.6213**

Model performance of Resnet50 on Food201

- Bhalaji Nagarajan, Eduardo Aguilar, Petia Radeva, S2ML-TL Framework for Multi-label Food Recognition. ICPR (5) 2020: 629-646

Conclusions

- Transfer learning – one of the main advantages of Deep learning

- Besides the well known Fine-tunning, we can define define several other approaches:

- Self-taught learning
- Domain adaptation
- Multi-task learning
- Sample selection bias
- Unsupervised learning

- Transfer learning allows to solve complex problems and applications as food recognition


Thank you!