Learning from Large-Scale Noisy Images

Jianping Fan
Department of Computer Science
UNC-Charlotte

Course Website:
http://webpages.uncc.edu/jfan/itcs5152.html

TRAINING DEEP NEURAL NETWORKS ON NOISY LABELS WITH BOOTSTRAPPING, ICLR 2015

Current state-of-the-art deep learning systems for visual object recognition and detection use purely supervised training with regularization such as dropout to avoid overfitting. The performance depends critically on the amount of labeled examples, and in current practice the labels are assumed to be unambiguous and accurate. However, this assumption often does not hold; e.g. in recognition, class labels may be missing; in detection, objects in the image may not be localized; and in general, the labeling may be subjective.

It is very interesting to develop new technologies that can directly learn from large-scale images with noisy labels!

TRAINING DEEP NEURAL NETWORKS ON NOISY LABELS WITH BOOTSTRAPPING, ICLR 2015

One generic way to handle noisy and incomplete labeling is to augment the prediction objective with a notion of consistency

If the same prediction is made given similar percepts, a prediction consistent can be considered; if the same prediction is made given similar percepts, the deep features computed from the input data could be similar.

This paper develops a simple consistency objective that does not require an explicit noise distribution or a reconstruction term. The idea is to dynamically update the targets of the prediction objective based on the current state of the model. The resulting targets are a convex combination of (1) the noisy training label, and (2) the current prediction of the model. Intuitively, as the learner improves over time, its predictions can be trusted more. This mitigates the damage of incorrect labeling, because incorrect labels are likely to be eventually highly inconsistent with other stimuli predicted to have the same label by the model.

TRAINING DEEP NEURAL NETWORKS ON NOISY LABELS WITH BOOTSTRAPPING, ICLR 2015

A cross-entropy objective is used, but generate new regression targets for each SGD mini-batch based on the current state of the model. We empirically evaluated two types of bootstrapping.

"Soft" bootstrapping uses predicted class probabilities q directly to generate regression targets for each batch as follows:

$$\mathcal{L}_{soft}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{L} [\beta t_k + (1-\beta) q_k] \log(q_k)$$

"Hard" bootstrapping modifies regression targets using the MAP estimate of $\mathbf{q}$ given $\mathbf{x}$, which we denote as $z_k := \mathbb{1}[k = \text{argmax } q_i, i = 1...L]$:

$$\mathcal{L}_{hard}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{L} [\beta t_k + (1-\beta) z_k] \log(q_k)$$

When used with mini-batch stochastic gradient descent, this leads to an EM-like algorithm: In the E-step, estimate the "true" confidence targets as a convex combination of training labels and model predictions; in the M-step, update the model parameters to better predict those generated targets.

TRAINING DEEP NEURAL NETWORKS ON NOISY LABELS WITH BOOTSTRAPPING, ICLR 2015

Both hard and soft bootstrapping can be viewed as instances of a more general approach in which model-generated regression targets are modulated by a softmax temperature parameter T; i.e.

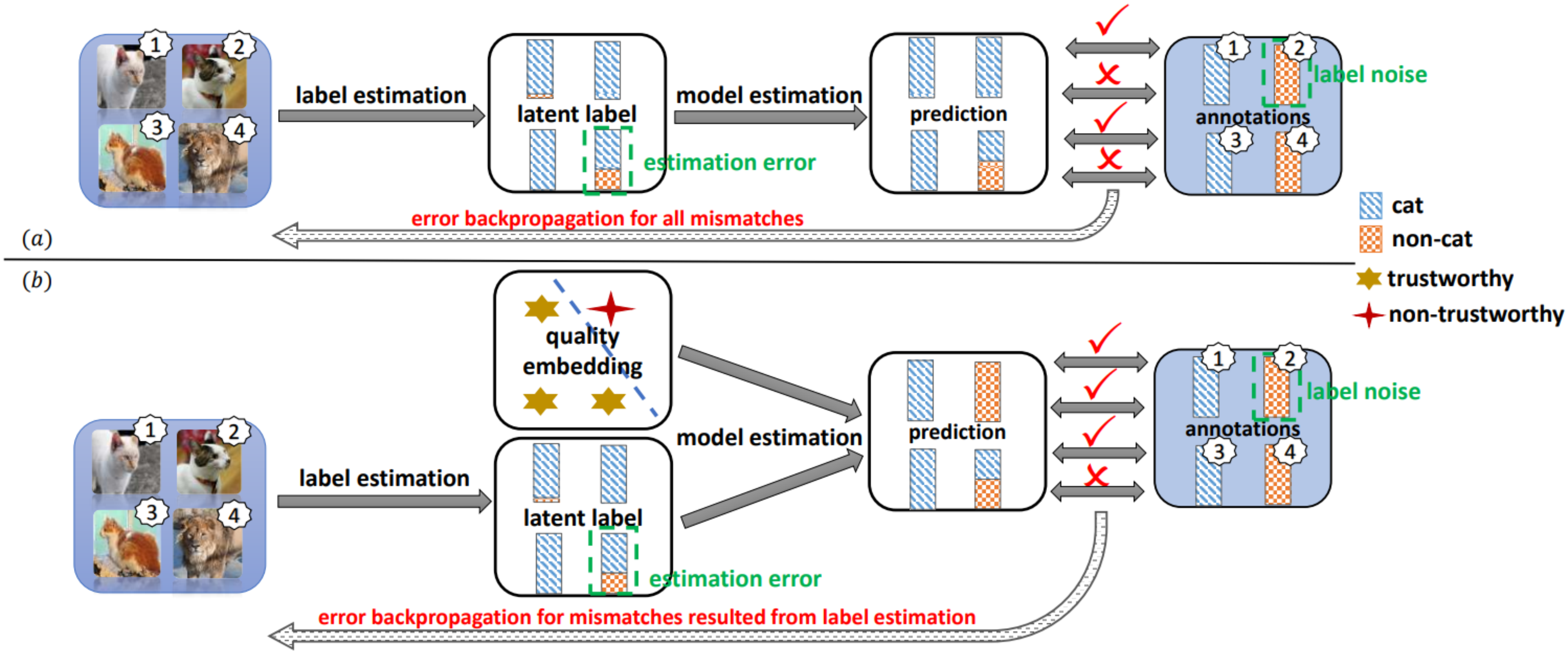
$$P(q_j = 1|\mathbf{x}) = \frac{\exp(T \cdot (\sum_{i=1}^{D} W_{ij}^{(1)} x_i + b_j^{(1)}))}{\sum_{j'=1}^{L} \exp(T \cdot (\sum_{i=1}^{D} W_{ij'}^{(1)} x_i + b_{j'}^{(1)}))}$$

Setting $T = 1$ recovers soft boostrapping, and $T = \infty$ recovers hard bootstrapping. We only use these two operating points in our experiments, but it may be worthwhile to explore other values for T, and learning T for each dataset.

$$\mathcal{L}_{multibox-hard}(\mathbf{c}, \mathbf{t}) = -\sum_{k=1}^{L} [\beta t_k + (1-\beta)\mathbb{1}_{c_k>0.5}] \log(c_k) \\ - \sum_{k=1}^{L} [\beta(1-t_k) + (1-\beta)(1-\mathbb{1}_{c_k>0.5})] \log(1-c_k)$$

$$\mathcal{L}_{multibox-soft}(\mathbf{c}, \mathbf{t}) = -\sum_{k=1}^{L} [\beta t_k + (1-\beta)c_k] \log(c_k) \\ - \sum_{k=1}^{L} [\beta(1-t_k) + (1-\beta)(1-c_k)] \log(1-c_k)$$

Deep Learning from Noisy Image Labels with Quality Embedding



Analysis about back-propagation in previous methods that model the latent label, as well as our idea to avoid the effect of label noise. (a) All images are forward into the model and the mismatch error caused by both label estimation and label noise are back-propagated. (b) With quality embedding as a control from latent labels to predictions, the negative effect of label noise is reduced in the back-propagation.

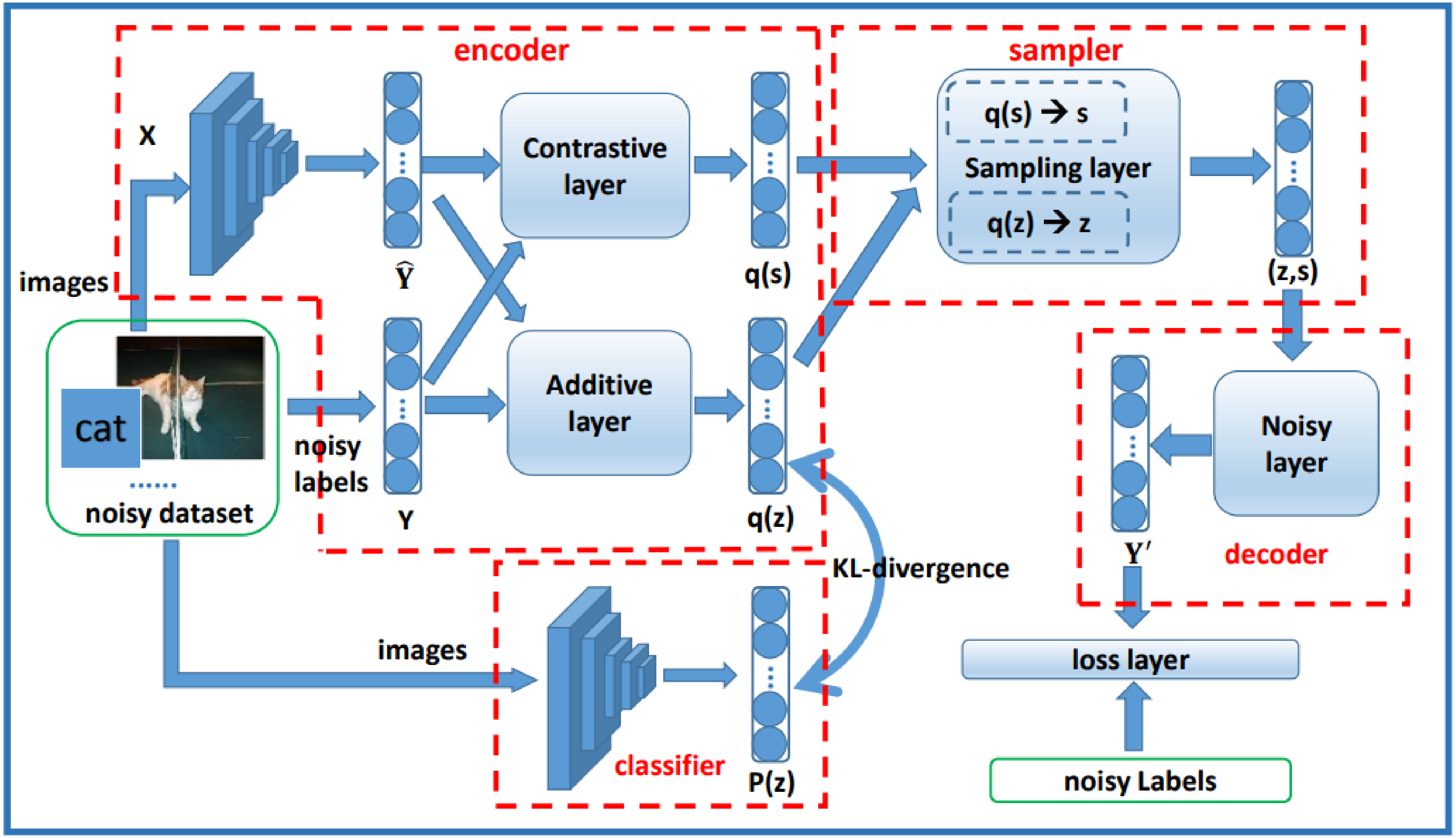
Deep Learning from Noisy Image Labels with Quality Embedding

Fig. 1 illustrates the idea of this paper as well as its advantage to reduce the noise effect.

In Fig. 1(a), the latent labels and predictions of the first three cat images must approximately consistent due to their content similarity. However, mismatch will occur between the second prediction and the corresponding annotation by virtue of the label noise. For the fourth image, the prediction induced by the estimation error of the latent label, also has conflict with the fourth annotation. As a result, these two mismatches will mix together for back-propagation.

On the other hand, if we explicitly introduce a quality variable to model the trustworthiness of noisy labels like Fig. 1(b), label noise can be reduced more effectively. For example, if the quality variable of the second sample is embedded in the non-trustworthy subspace, the latent label can be disturbed accordingly to prevent mismatch error caused by the label noise from back-propagation. While for the fourth sample whose quality variable is estimated in the trustworthy subspace, the latent label still transits to the final prediction causing the mismatch. Then supervision from the correct annotations is normally fed back.

Deep Learning from Noisy Image Labels with Quality Embedding



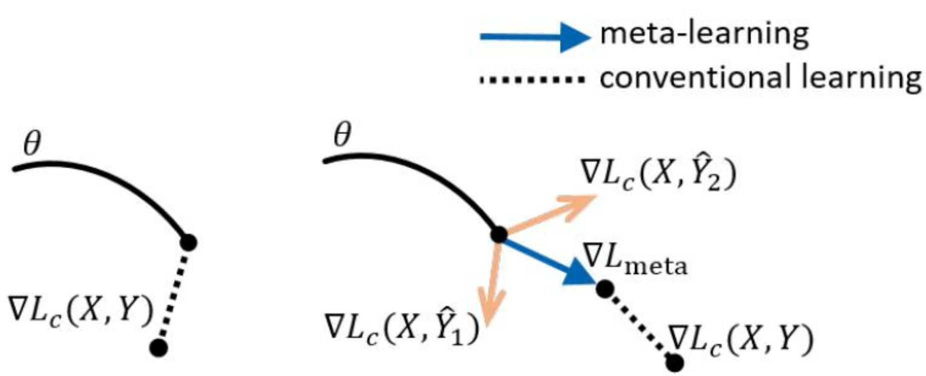
The network consists of four modules, encoder, sampler, decoder and classifier, which are trained end-to-endly. Encoder tries to learn latent labels and evaluate the quality of noisy labels; sampler is used to generate samples from encoder outputs; decoder tries to recover noisy labels from samples. Meanwhile, our classifier is learned based on KL-divergence between q(z) and P(z).

Learning to Learn from Noisy Labeled Data

Despite the success of deep neural networks (DNNs) in image classification tasks, the human-level performance relies on massive training data with high-quality manual annotations, which are expensive and time-consuming to collect.

There exist many inexpensive data sources on the web, but they tend to contain inaccurate labels. Training on noisy labeled datasets causes performance degradation because DNNs can easily overfit to the label noise.

To overcome this problem, we propose a noise-tolerant training algorithm, where a meta-learning update is performed prior to conventional gradient update. The proposed meta-learning method simulates actual training by generating synthetic noisy labels, and train the model such that after one gradient update using each set of synthetic noisy labels, the model does not overfit to the specific noise.



Learning to Learn from Noisy Labeled Data

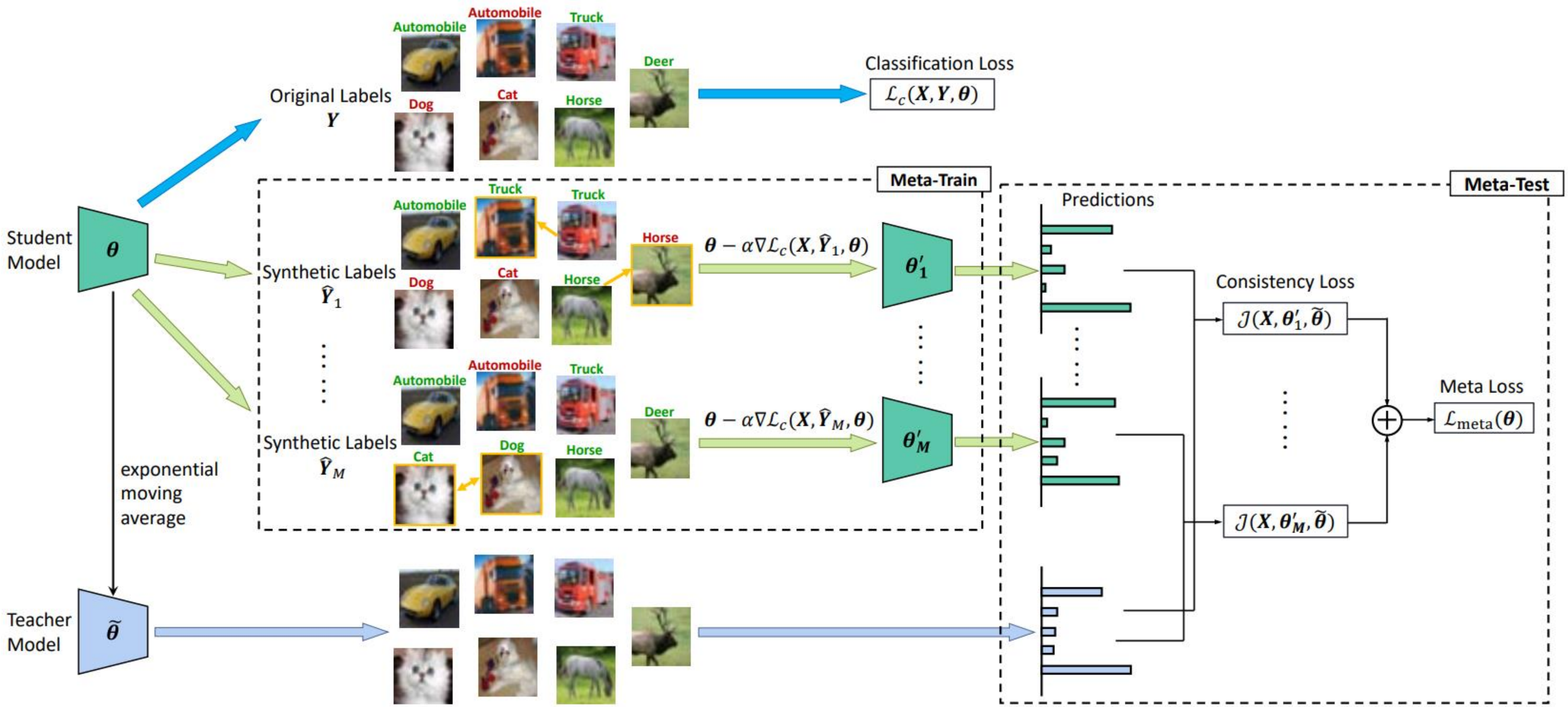


Illustration of the proposed meta-learning based noise-tolerant (MLNT) training. For each mini-batch of training data, a meta loss is minimized before training on the conventional classification loss. We first generate multiple mini-batches of synthetic noisy labels with random neighbor label transfer (marked by orange arrow). The random neighbor label transfer can preserve the underlying noise transition (e.g. DEER → HORSE, CAT ↔ DOG), therefore generating synthetic label noise in a similar distribution as the original data. For each synthetic mini-batch, we update the parameters with gradient descent, and enforce the updated model to give consistent predictions with a teacher model

Learning to Learn from Noisy Labeled Data

Our method can learn the parameters of a DNN model in such a way as to "prepare" the model for label noise. The intuition behind our method is that when training with a gradient-based rule, some network parameters are more tolerant to label noise than others.

How can we encourage the emergence of such noise-tolerant parameters?

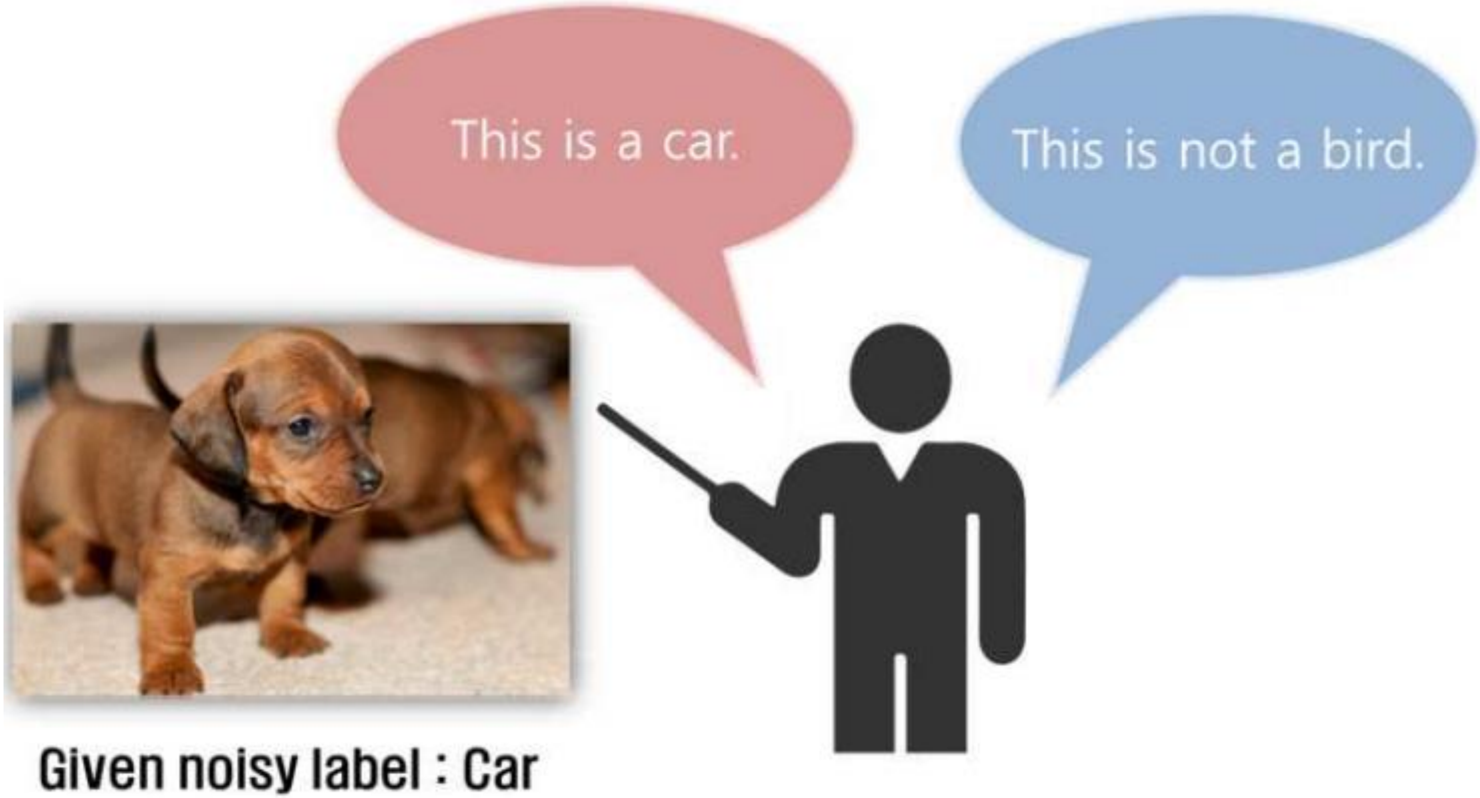
We achieve this by introducing a meta-learning update before the conventional update for each mini-batch. The meta-learning update simulates the process of training with label noise and makes the network less prone to over-fitting. Specifically, for each mini-batch of training data, we generate a variety of synthetic noisy labels on the same images. With each set of synthetic noisy labels, we update the network parameters using one gradient update, and enforce the updated network to give consistent predictions with a teacher model unaffected by the synthetic noise. As shown in Figure 1, the meta-learning update optimizes the model so that it can learn better with conventional gradient update on the original mini-batch. In effect, we aim to find model parameters that are less sensitive to label noise and can consistently learn the underlying knowledge from data despite label noise.

NLNL: Negative Learning for Noisy Labels, ICCV 2019

Convolutional Neural Networks (CNNs) provide excellent performance when used for image classification. The classical method of training CNNs is by labeling images in a supervised manner as in “input image belongs to this label” (Positive Learning; PL), which is a fast and accurate method if the labels are assigned correctly to all images. However, if inaccurate labels, or noisy labels, exist, training with PL will provide wrong information, thus severely degrading performance.

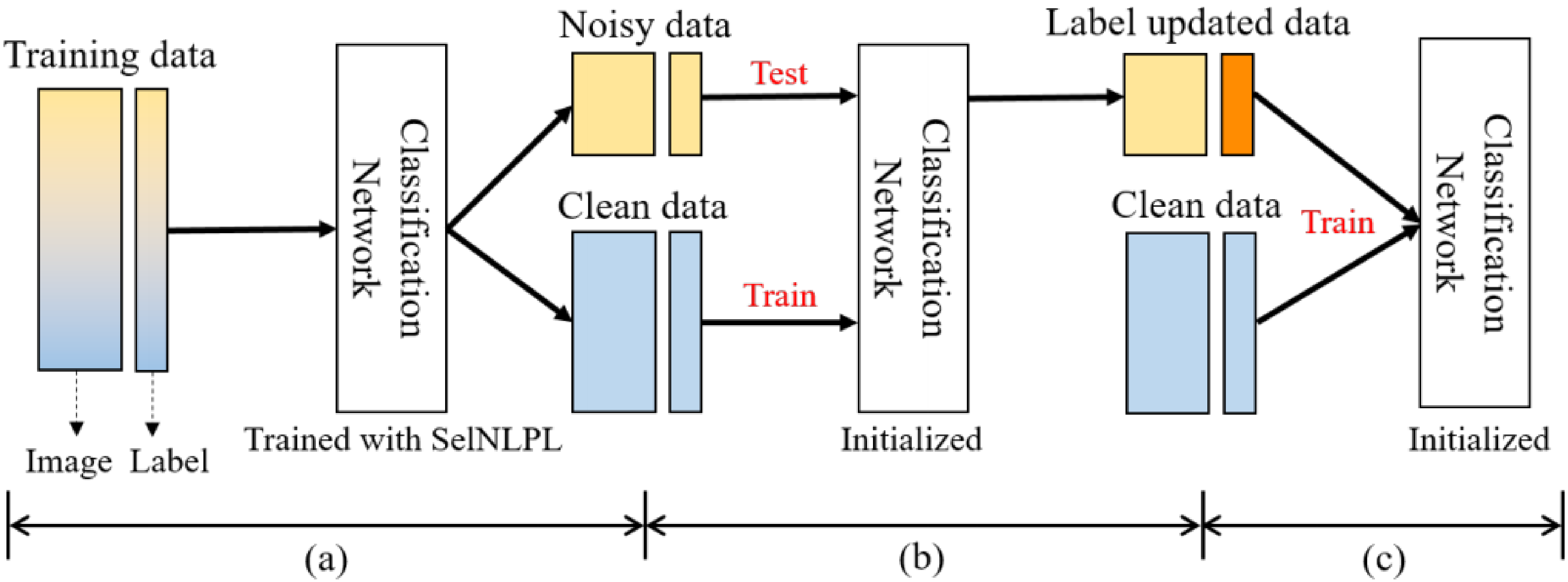
To address this issue, we start with an indirect learning method called Negative Learning (NL), in which the CNNs are trained using a complementary label as in “input image does not belong to this complementary label.” Because the chances of selecting a true label as a complementary label are low, NL decreases the risk of providing incorrect information. Furthermore, to improve convergence, we extend our method by adopting PL selectively, termed as Selective Negative Learning and Positive Learning (SelNLPL). PL is used selectively to train upon expected-to-be-clean data, whose choices become possible as NL progresses, thus resulting in superior performance of filtering out noisy data.

NLNL: Negative Learning for Noisy Labels, ICCV 2019



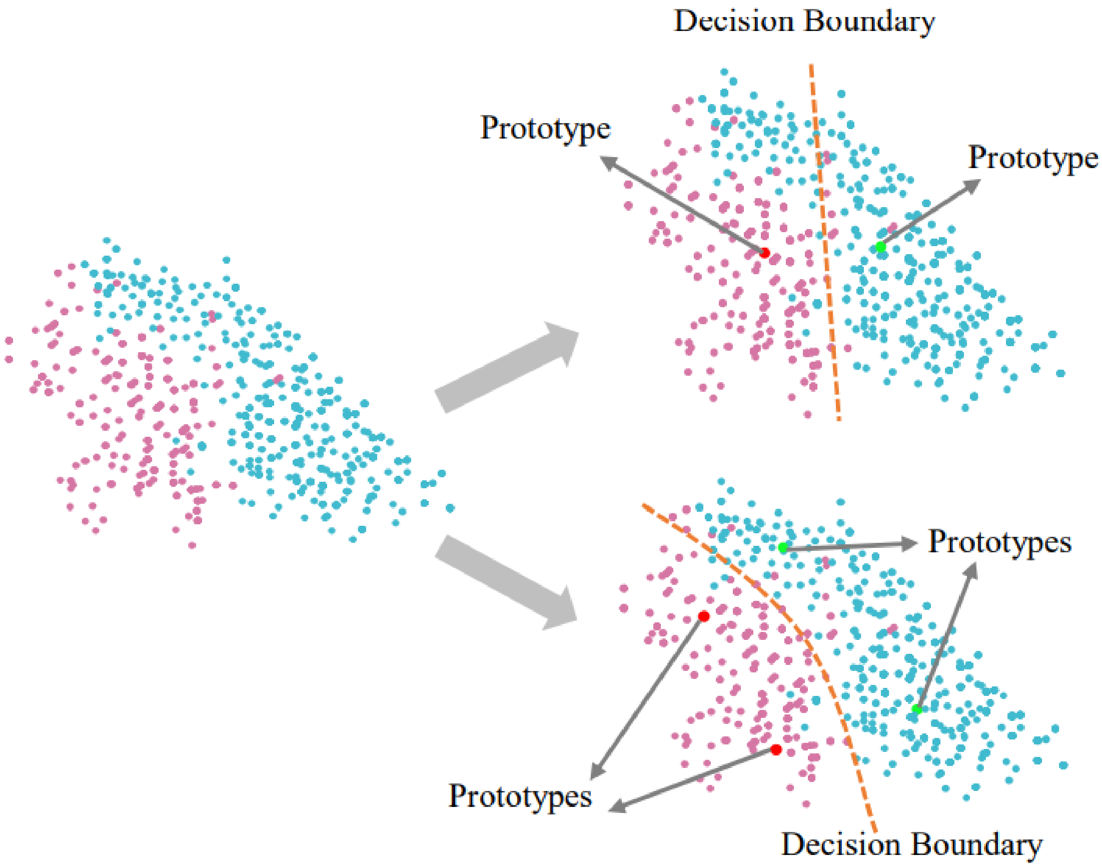
Conceptual comparison between Positive Learning (PL) and Negative Learning (NL). Regarding noisy data, while PL provides CNN the wrong information (red balloon), with a higher chance, NL can provide CNN the correct information (blue balloon) because a dog is clearly not a bird.

NLNL: Negative Learning for Noisy Labels, ICCV 2019



Pseudo labeling for semi-supervised learning. (a): Division of training data into either clean or noisy data with CNN trained with SelNLPL. (b): Training initialized CNN with clean data from (a), then noisy data's label is updated following the output of CNN trained with clean data. (c): Clean data and label-updated noisy data are both used for training initialized CNN in the final step.

Deep Self-Learning From Noisy Labels, ICCV 2019



An example of solving two classes classification problem using different number of prototypes. Left: Original data distribution. Data points with the same color belong to the same class. Upper Right: The decision boundary obtained by using a single prototype for each class. Lower Right: The decision boundary obtained by two prototypes for each class. Two prototypes for each class leads to a better decision boundary.

Deep Self-Learning From Noisy Labels, ICCV 2019

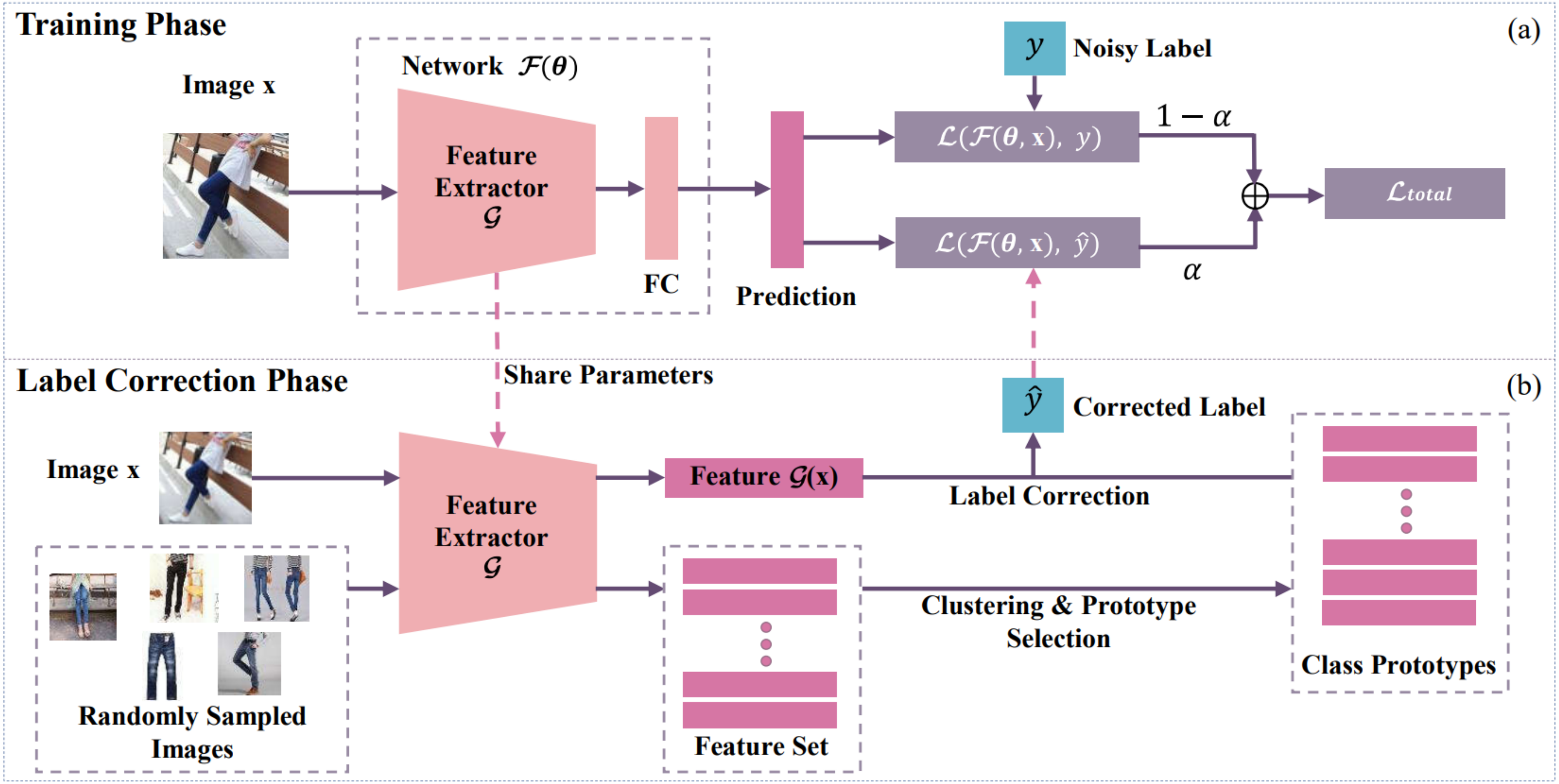
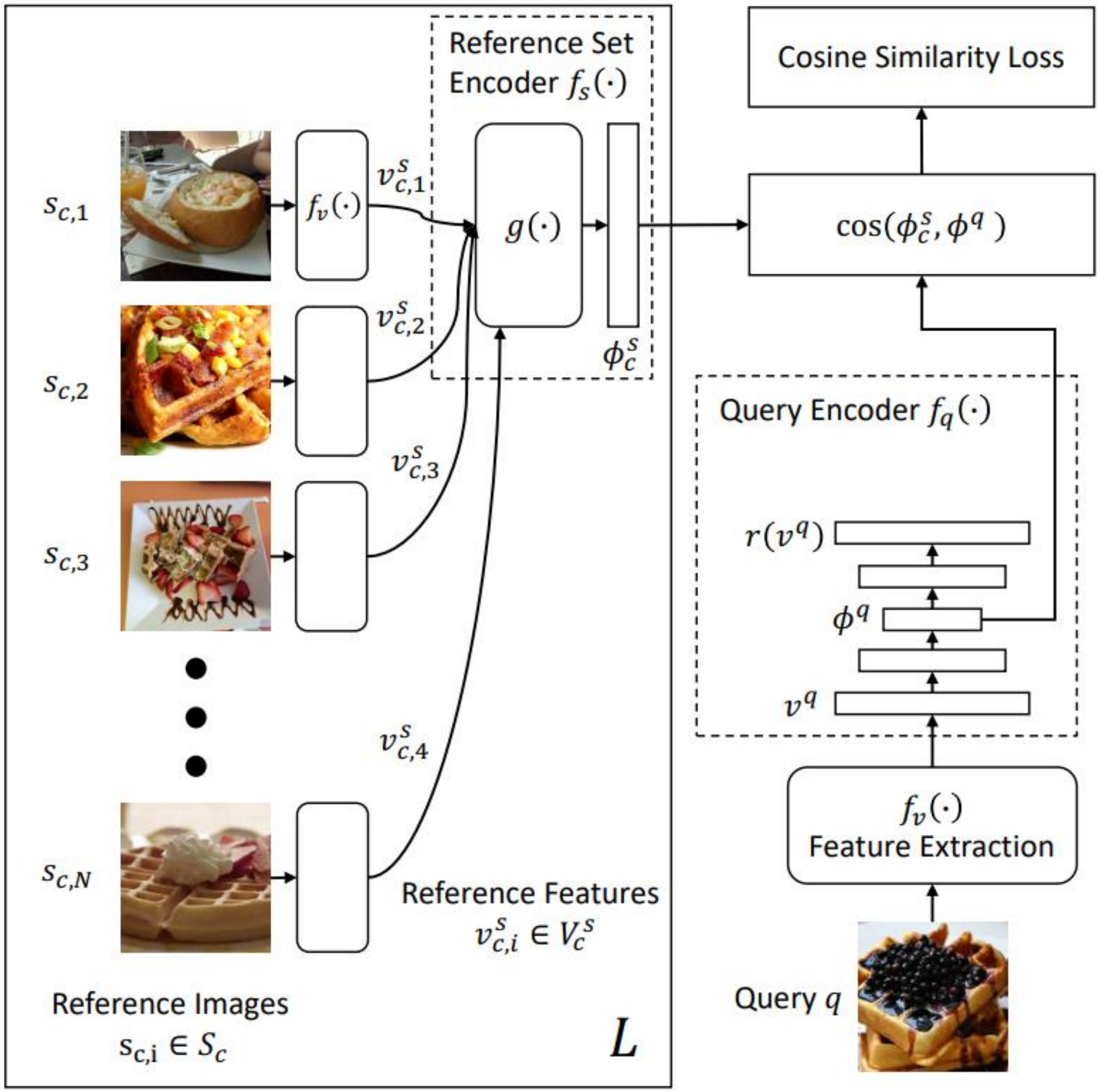


Illustration of the pipeline of iterative self-learning framework on the noisy dataset. (a) shows the training phase and (b) shows the label correction phase, where these two phases proceed iteratively. The deep network G can be shared, such that only a single model needs to be evaluated in testing.

CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise, CVPR 2018



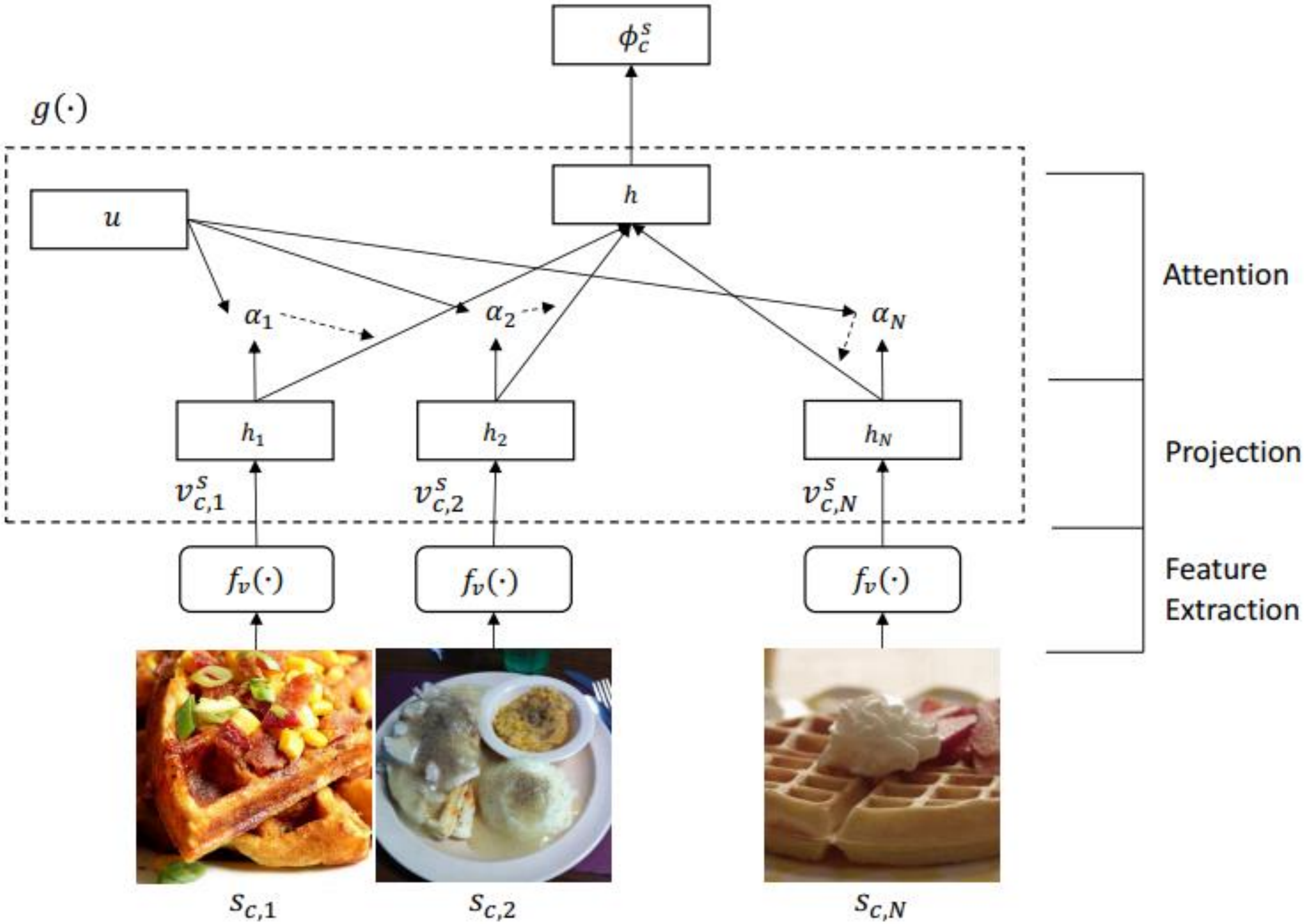
CleanNet architecture for learning a class embedding vector ϕ_c^s and a query embedding vector ϕ_q with a similarity matching constraint. There exists one class embedding for each of the L classes.

CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise, CVPR 2018

The overall architecture of CleanNet consists of two parts: (1) a reference set encoder and (2)a query encoder.

The reference set encoder fs(·) learns to focus on representative features in a noisy reference image set, which is collected for a specific class, and outputs a class-level embedding vector. Since using all the images in the reference set is computationally expensive, we first create a representative subset, and extract one visual feature vector from each image in that subset to form a representative feature vector set, i.e., let V_c^s denotes the representative reference feature vector set for class c (reference feature set).

CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise, CVPR 2018



Reference set encoder fs(·)

CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise, CVPR 2018

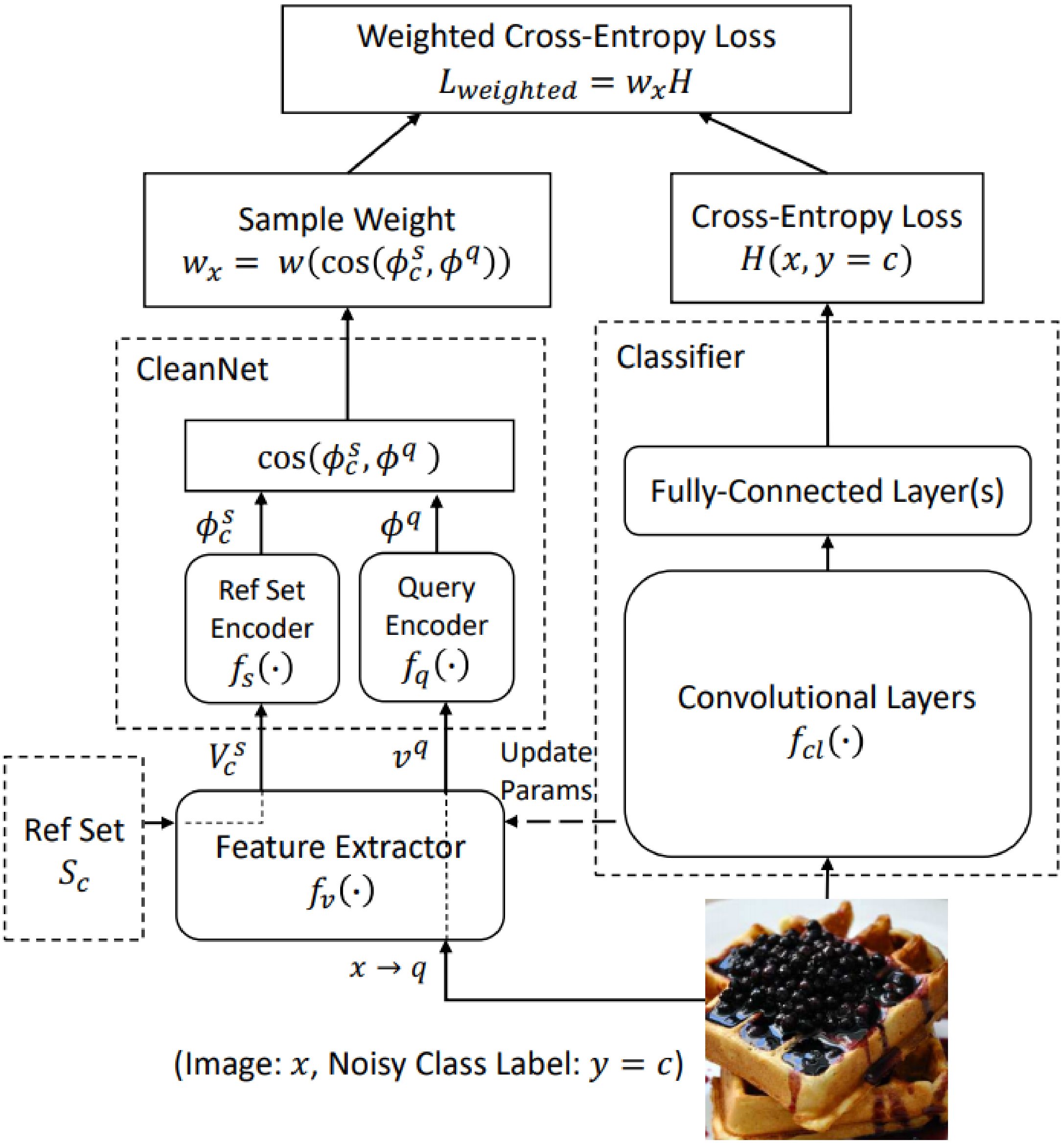
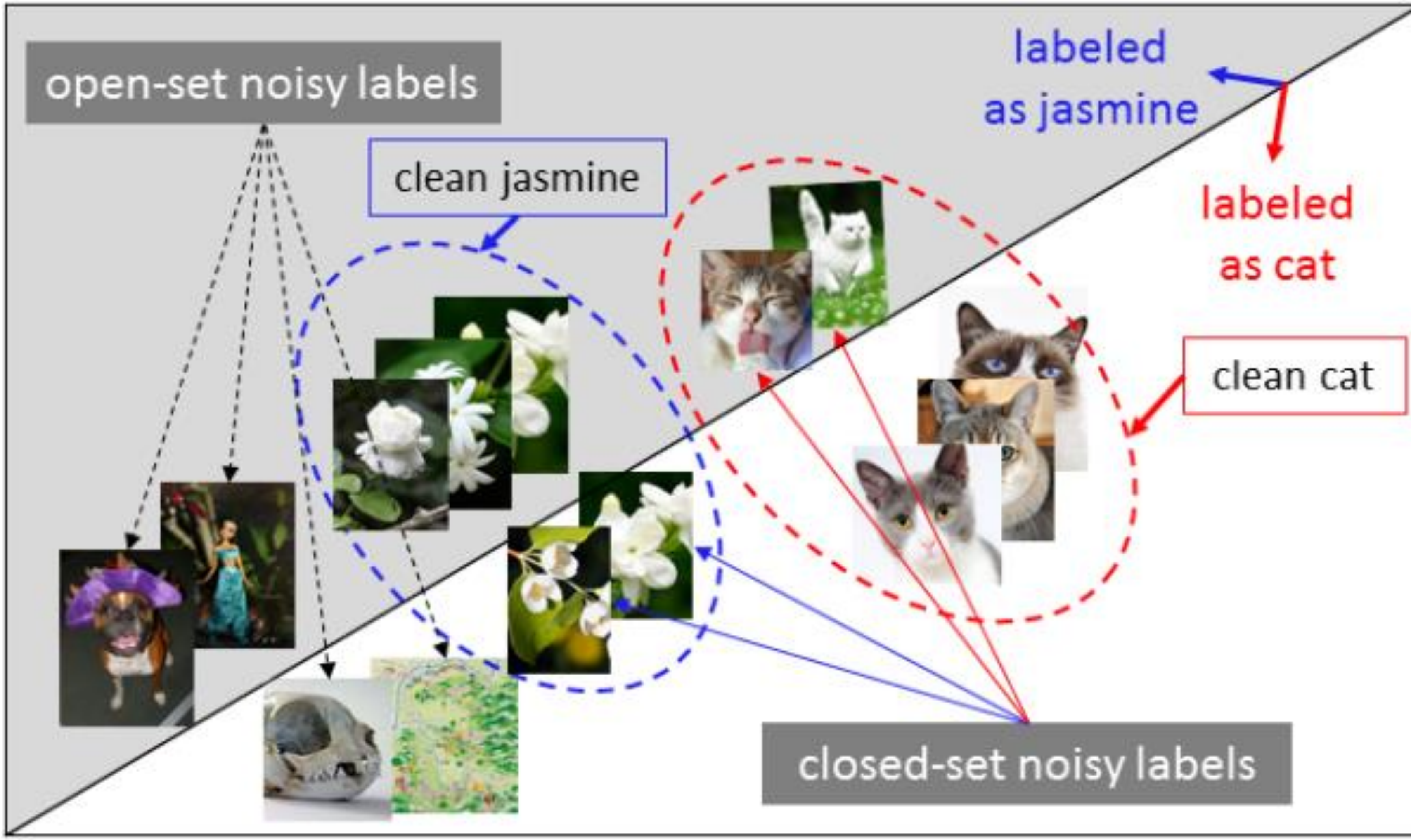


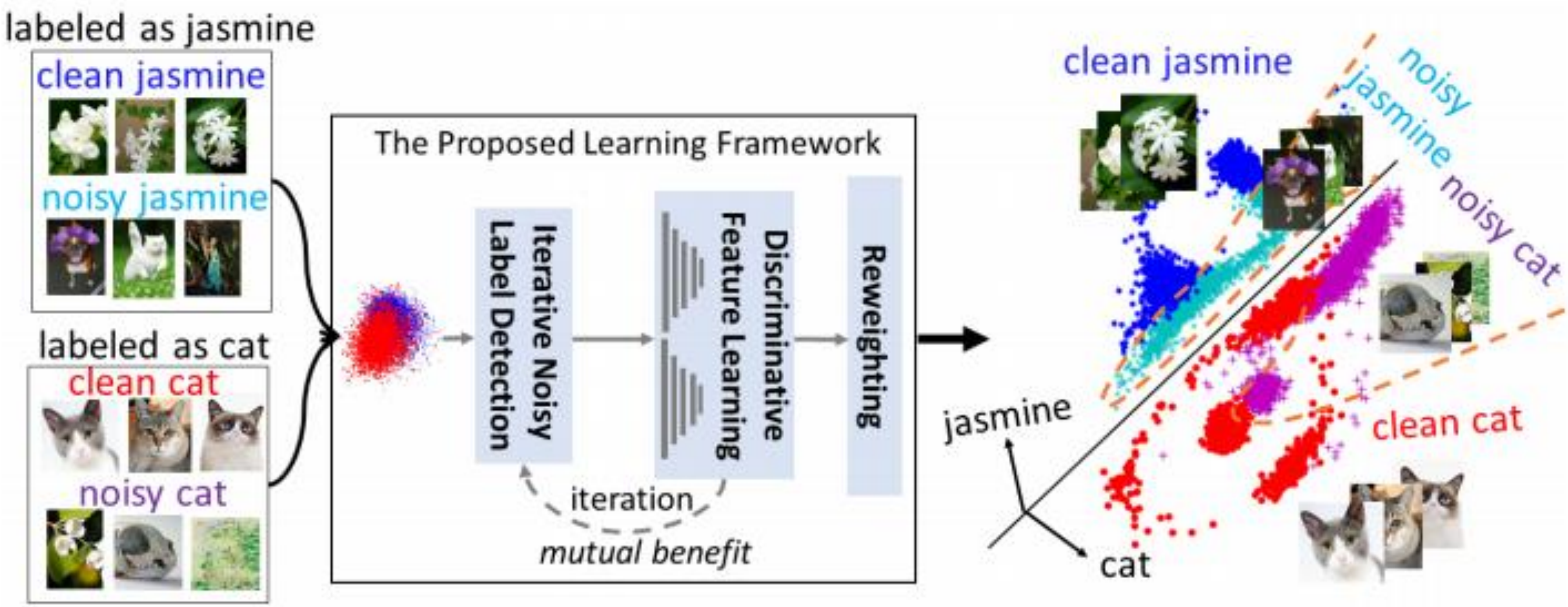
Illustration of integrating CleanNet for training the CNN-based image classifier with label noise.

Iterative Learning with Open-set Noisy Labels, CVPR 2018



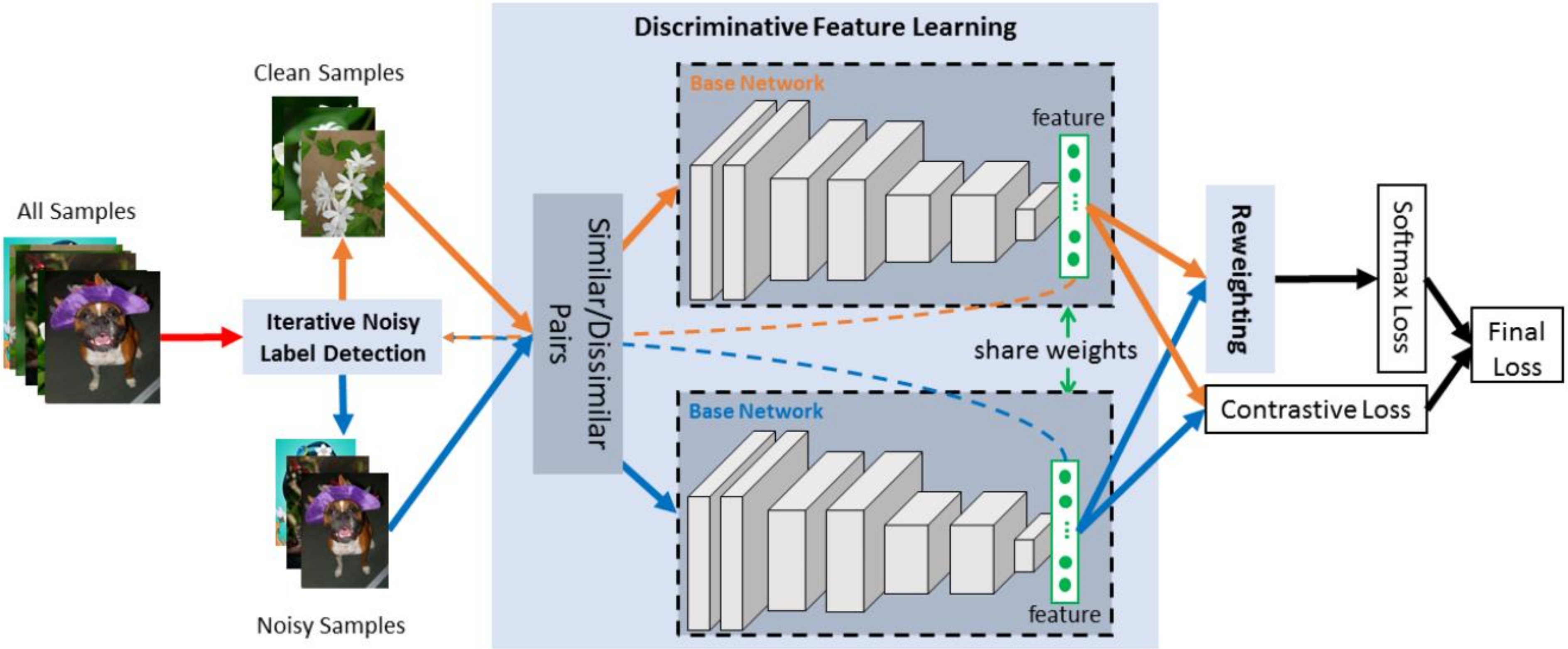
An illustration of closed-set vs open-set noisy labels.

Iterative Learning with Open-set Noisy Labels, CVPR 2018



An overview of our framework that iteratively learns discriminative representations on a “jasmine-cat” dataset with openset noisy labels. It not only learns a proper decision boundary (the black line separating jasmine and cat) but also pulls away noisy samples (green and purple) from clean samples (blue and red).

Iterative Learning with Open-set Noisy Labels, CVPR 2018



The framework of the proposed iterative learning approach. Iterative noisy label detection module and discriminative feature learning module form a closed-loop, i.e., one module's inputs are the other module's output, which can benefit from each other and be jointly enhanced. The network is jointly optimized by two types of losses: reweighted softmax loss and contrastive loss.

Learning from Noisy Labels with Distillation, ICCV2017

Learning from noisy labels

Cost: delay, dollars, manpower

Low cost → High cost

Unsupervised Learning | **Learning From Noisy Labels** | Semi Supervised Learning | Supervised Learning

YFCC100M Dataset

- Yahoo Flickr Creative Commons 100M
- 100,000,000 Flickr photos
- Pixels and metadata:
  - User tags, machine tags, username, title, description, geo tags, device, date
- Visual concept learning with YFCC100M

Text based linking: image candidates → Data labeling: partial clean labels → Model learning: partial clean labels and noisy labels

Learning from Noisy Labels with Distillation, ICCV2017

Types of label noise

Traditional assumption: Random Classification Noise (RCN): Bird ⇒ Cat

In practice: text ambiguity

Learning from Noisy Labels with Distillation, ICCV2017

Related Work

Bootstrap

Reed, et al. "Training deep neural networks on noisy labels with bootstrapping." *ICLR* 2014.

- Make prediction based on current model:

$$z_k := \mathbb{1}[k = \text{argmax } q_i, i = 1 \ldots L]$$

- Update with the modified labels:

$$\mathcal{L}_{hard}(\mathbf{q}, \mathbf{t}) = \sum_{k=1}^{L} [\beta t_k + (1-\beta) z_k] \log(q_k)$$

Reweight

Liu, et al. "Classification with noisy labels by importance reweighting." *IEEE TPAMI 2016*.

- Estimate noise level with a pretrained classifier $P_{D_\rho}(\hat{Y}|X)$

$$\rho_{-\hat{Y}} = \min_{X \in \mathcal{X}} P_{D_\rho}(\hat{Y}|X).$$

- Estimate instance importance: (how likely it is a noise sample)

$$\beta(X, \hat{Y}) = \frac{P_{D_\rho}(\hat{Y}|X) - \rho_{-\hat{Y}}}{(1 - \rho_{+1} - \rho_{-1}) P_{D_\rho}(\hat{Y}|X)},$$

- Retrain the model with the weighted loss

$$\beta(X, \hat{Y}) \ell(f(X), \hat{Y})$$

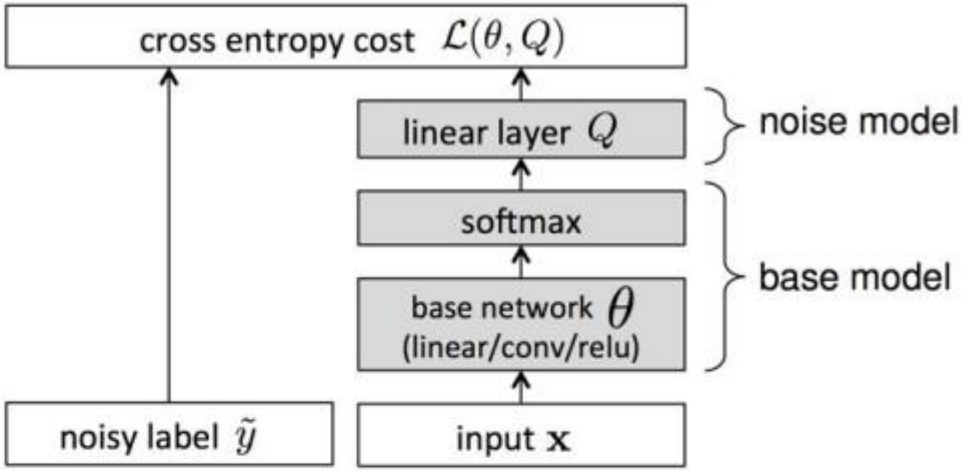
Learning from Noisy Labels with Distillation, ICCV2017

Related Work

Noise layer

Sukhbaatar, et al. "Learning from noisy labels with deep neural networks." ICLR 2014.

- Add a new layer on top of softmax to “absorb” noise



(a)

Learning from Noisy Labels with Distillation, ICCV2017

Related Work

Label smooth Szegedy, et al. "Rethinking the inception architecture for computer vision." *ICLR* 2015

- Modify the label map with smoothed version:

Original label map	0	0	0	1	0	0	0	0	0	0
Smoothed label map	0.01	0.01	0.01	0.91	0.01	0.01	0.01	0.01	0.01	0.01

Just do it! Krause, Jonathan, et al. "The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition." *ECCV 2016*.

- Two kinds of noise
  - Cross domain noise
  - Cross category noise
- The cross domain images are not shown in the evaluation
  - This is true only for fine grained classification

Willet: the name

Willet: the bird

Learning from Noisy Labels with Distillation, ICCV2017

Semantic knowledge graph

Family: Pinaceae

Fir

Larix_laricina

Spruce

Larch

Order: Hemiptera

Leafhopper

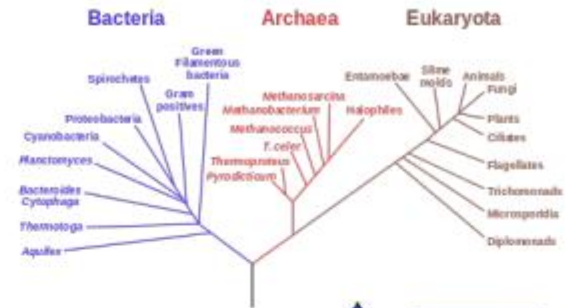
Aphid

Cicada

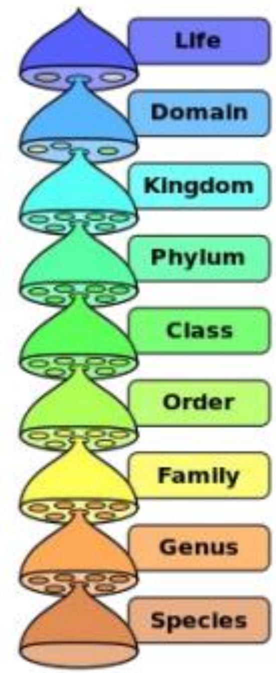
Class: Bird

Hummingbird, Ostrich, Tanager, Ruff, Willet, Darter, …

Phylogenetic Tree of Life



Source: wikipedia



Learning from Noisy Labels with Distillation, ICCV2017

Visual knowledge graph


Bison


Gaur


Takin


Wildebeest


Zebu


Abalone

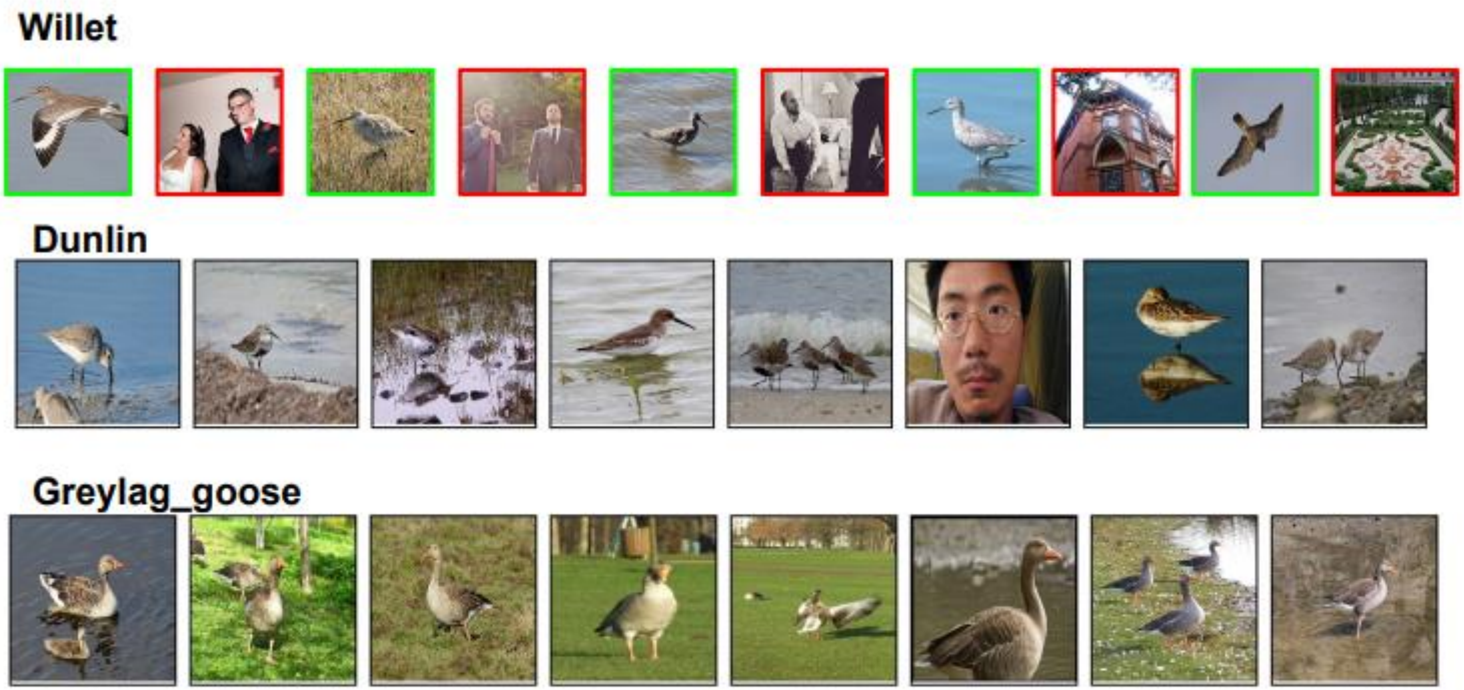

Clam


Mussel


Oyster


Scallop

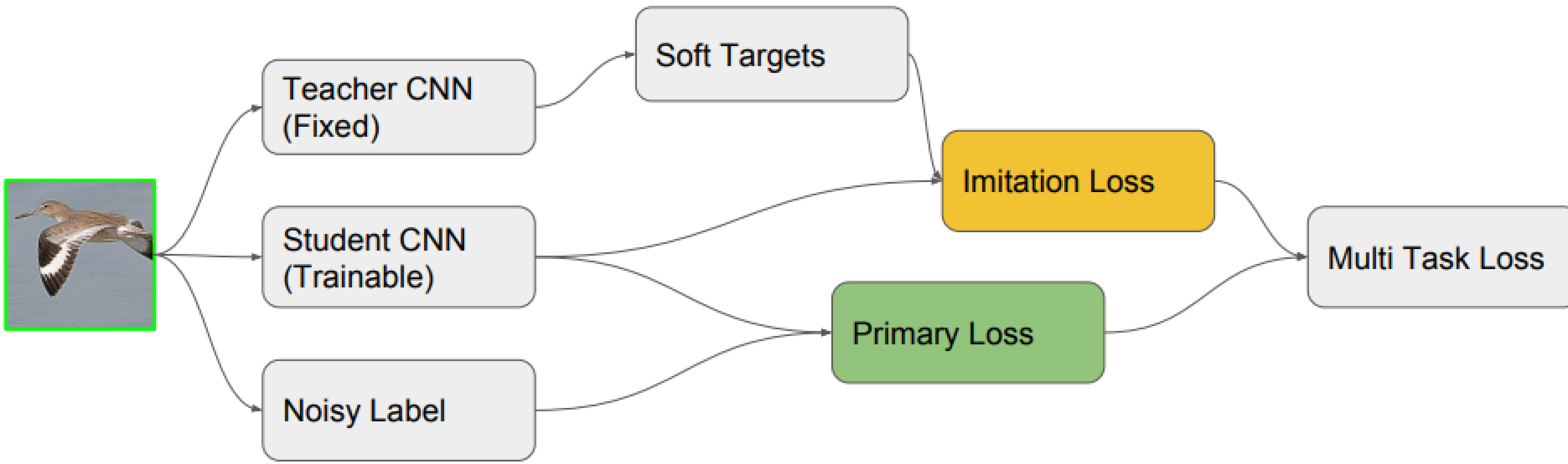
A motivating example



There is no way to get rid of the ambiguity by itself

Learning from Noisy Labels with Distillation, ICCV2017

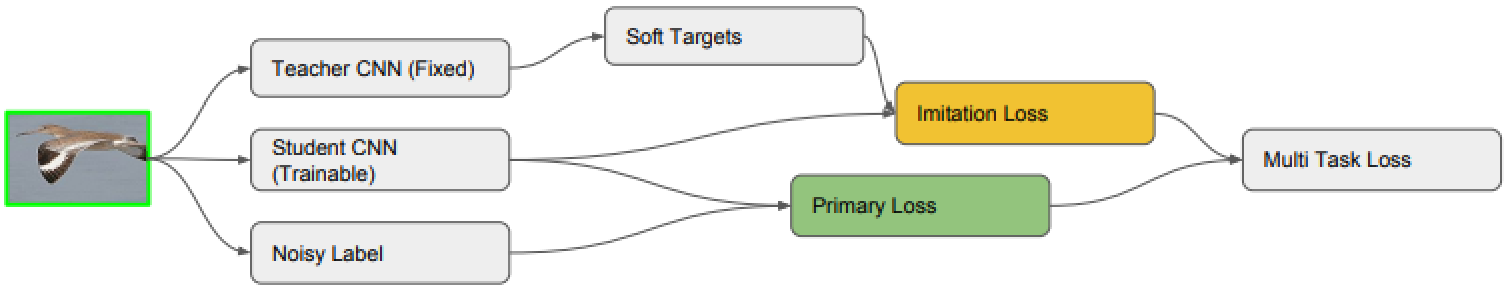
Distillation



$$f_s = \arg\min_{f \in \mathcal{F}_s} \frac{1}{n} \sum_{i=1}^{n} \left[\underbrace{(1-\lambda)\ell(y_i, \sigma(f(x_i)))}_{\text{Primary Loss}} + \underbrace{\lambda \ell(s_i, \sigma(f(x_i)))}_{\text{Imitation Loss}} \right],$$

Learning from Noisy Labels with Distillation, ICCV2017

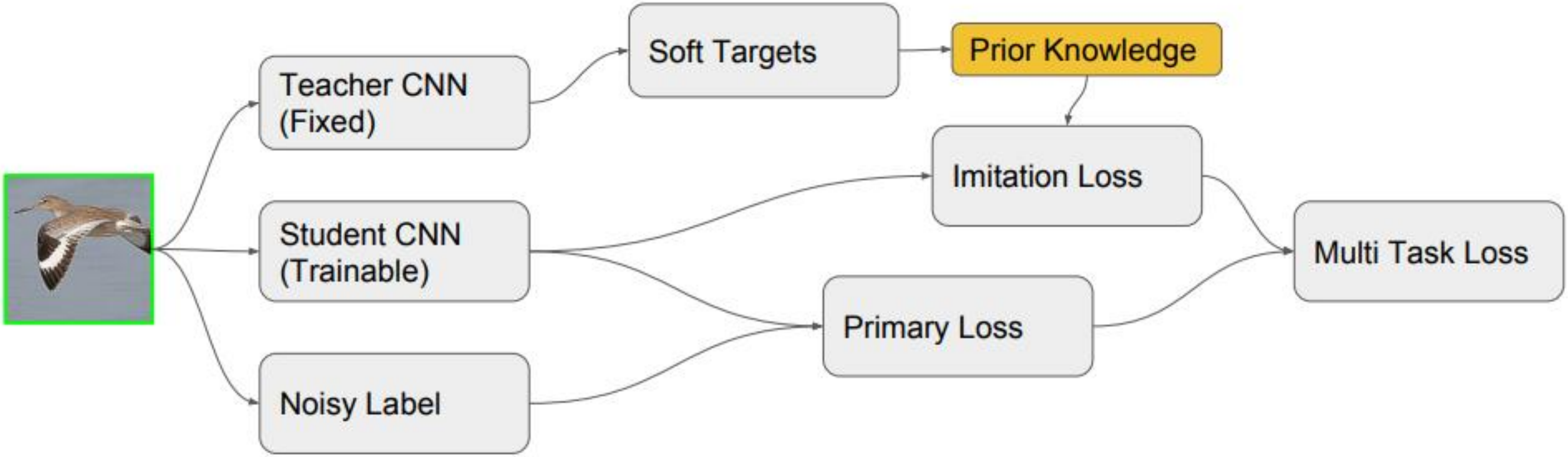
Examples of Distillation



Teacher CNN	Student CNN	Reference
Expensive strong CNN ensemble	Deployable weak CNN	Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).
Privileged features	Generic features	Lopez-Paz, David, et al. "Unifying distillation and privileged information."*arXiv preprint arXiv:1511.03643* (2015).
Small set of clean labels	Large set of noisy labels + Knowledge graph	Ours

Learning from Noisy Labels with Distillation, ICCV2017

Guided Distillation

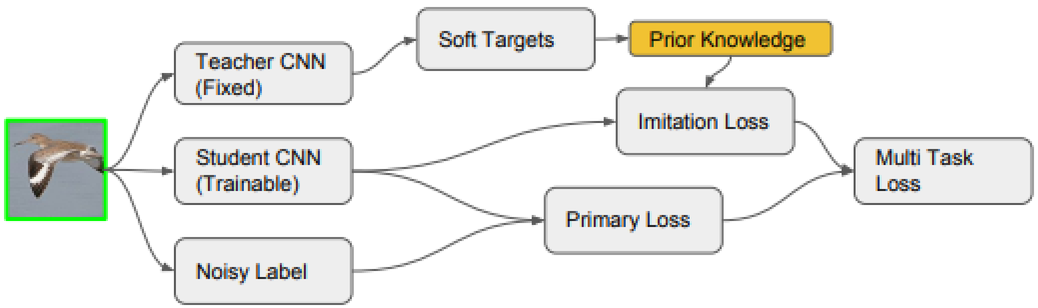


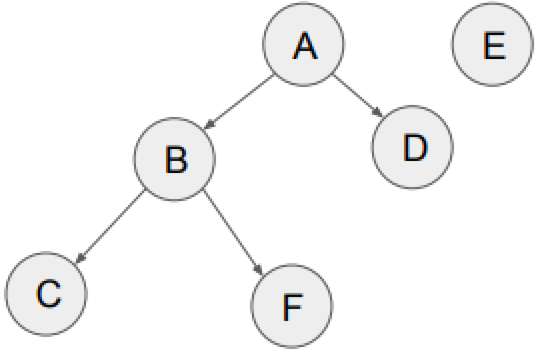
$$f_s = \arg\min_{f \in \mathcal{F}_s} \frac{1}{n} \sum_{i=1}^{n} \left[(1-\lambda)\ell(y_i, \sigma(f(x_i))) + \lambda \ell(s_i, \sigma(f(x_i))) \right],$$

$$s = g(\tilde{s}, \Phi)$$

Learning from Noisy Labels with Distillation, ICCV2017

Knowledge Graph Guided Distillation



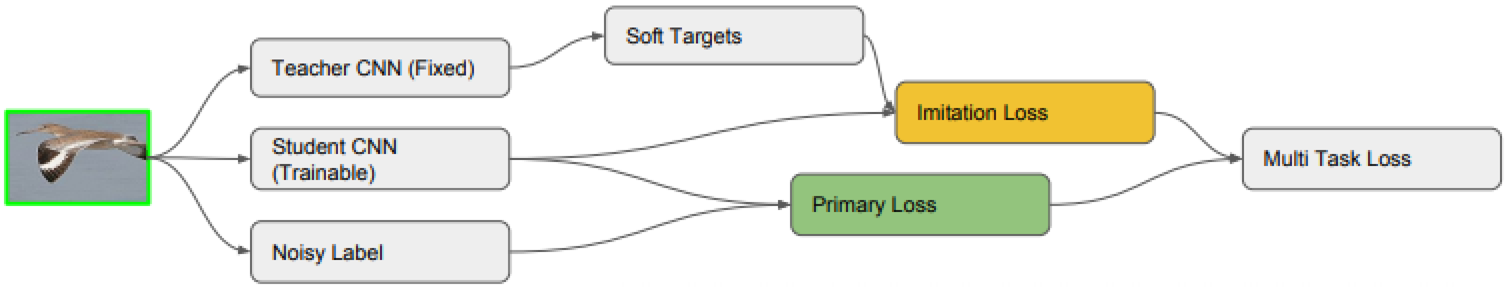


$$f_s = \arg\min_{f \in \mathcal{F}_s} \frac{1}{n} \sum_{i=1}^{n} \Big[(1-\lambda)\ell(y_i, \sigma(f(x_i))) + \lambda \ell(s_i, \sigma(f(x_i))) \Big],$$

$$s \equiv \tilde{s} \times T$$

$$T(m,n) = \begin{cases} 1-\beta & m = n \\ \frac{\beta}{|\mathrm{sibling}(n)|} & m \in \mathrm{sibling}(n) \\ 0 & \text{otherwise} \end{cases}$$

Knowledge Distillation == Risk Hedging



$$\hat{y}_i^{\lambda} = \lambda y_i + (1 - \lambda) s_i$$

Proposition 1. *The optimal risk associated with $\hat{y}^{\lambda}$ is smaller than both risks with y and s,* i.e.

$$\min_{\lambda} R_{\hat{y}^{\lambda}} < \min\{R_y, R_s\}, \tag{7}$$

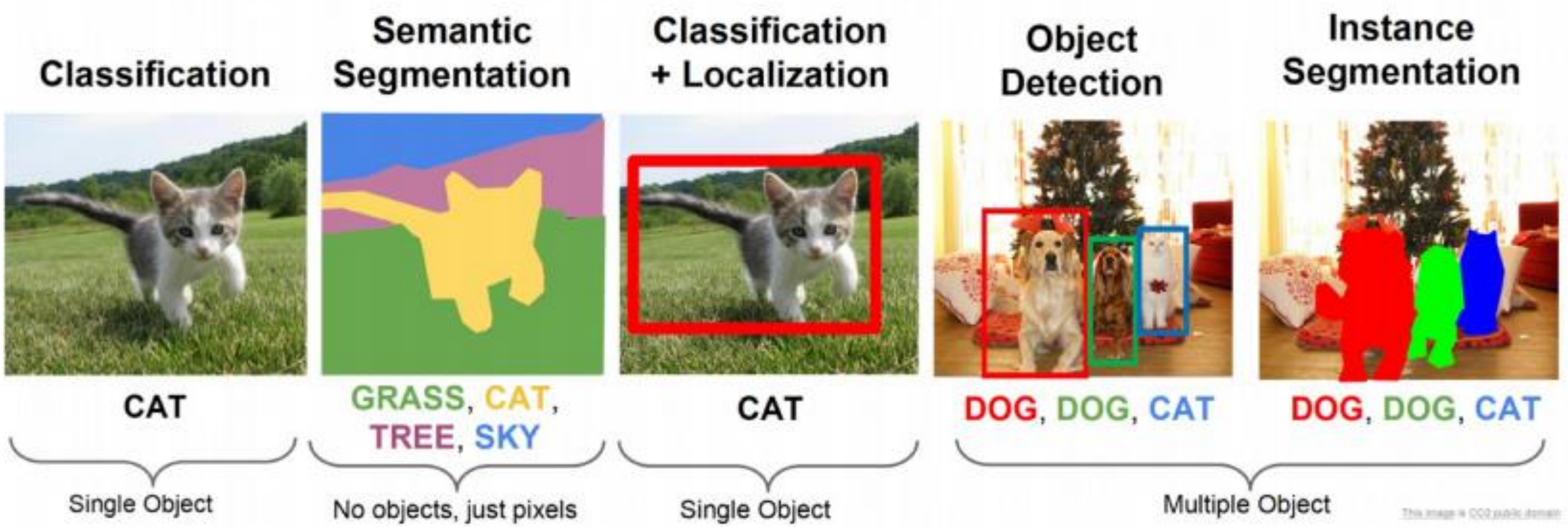
where y is the unreliable label on $\mathcal{D}$, and s is the soft label output from f_{D_c}. By setting $\lambda = \frac{R_s}{R_s + R_y}$, $R_{\hat{y}^{\lambda}}$ reaches its minimum,

$$\min_{\lambda} R_{\hat{y}^{\lambda}} = \frac{R_y R_s}{R_s + R_y}. \tag{8}$$

- Deep Learning is a subfield of Machine Learning that has achieved impressive results outperforming previous techniques and even humans, thus becoming the state-of-the-art in a wide range of tasks
- This success was possible due to 3 main factors:
  - Processing power
  - Data models
  - Data
- Computer Vision has been one of the most benefited areas

Darian Frajberg, Applying Deep Learning with Weak and Noisy labels

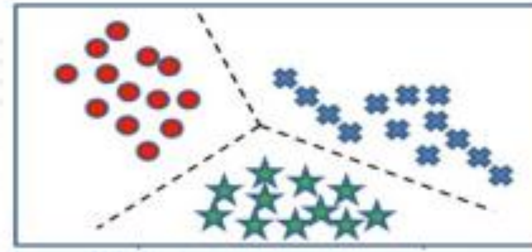
- Computer Vision tasks



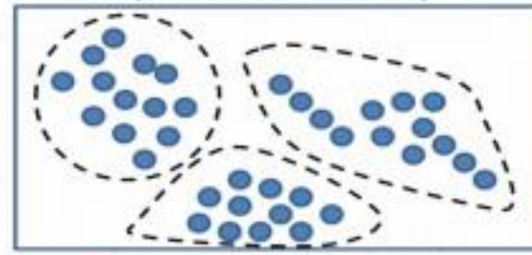
[http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf]

Types of Learning

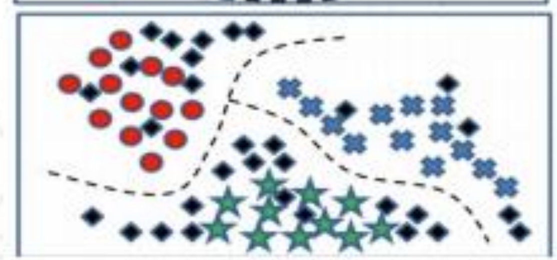
- Supervised learning
  - Training data includes desired outputs



- Unsupervised learning
  - Training data does not include desired outputs



- Weakly or semi-supervised learning
  - Training data includes a few desired outputs

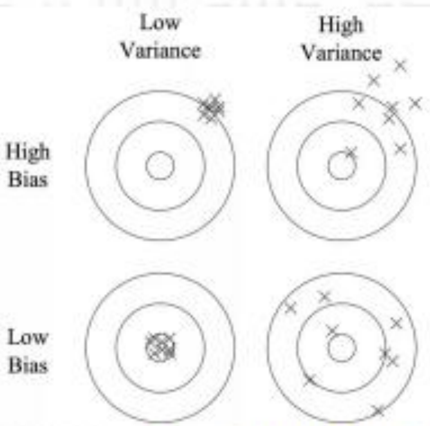


- Reinforcement learning
  - Rewards from sequence of actions

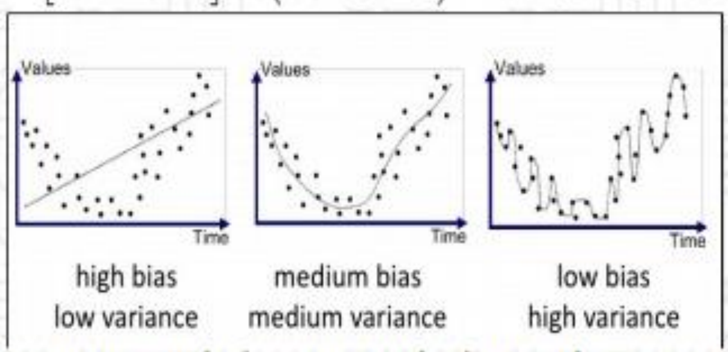
[ECE 5984: Introduction to Machine Learning, Dhuruv Batra]

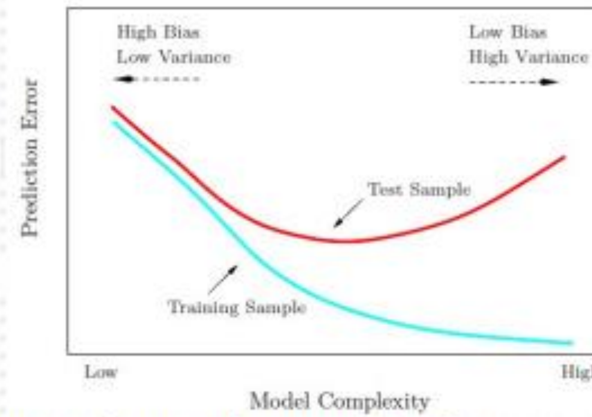
Bias – Variance tradeoff

- **Bias** is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to simplify and miss the relevant relations between features and target outputs (underfitting).
- **Variance** is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).



$$\mathrm{E}\left[\left(y-\hat{f}(x)\right)^2\right] = \left(\mathrm{Bias}\left[\hat{f}(x)\right]\right)^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$





- Supervised learning with large and clean data can generalize achieving high accuracy by reducing both bias and variance.

Large scale datasets labelling

- State-of-the-art machine learning models require massive labeled training sets, which usually do not exist for real-world applications

- High quality data characteristics:
  - Accuracy
  - Completeness
  - Consistency
  - Uniqueness
  - Validity
  - Timeliness

- Constraints:
  - Labor-intensive
  - Expensive
  - Tedious
  - May require domain expertise

Large scale datasets labelling

- Solution:
  - Exploit cheaper sources of labels
- Options:
  - Publicly available data
  - Weakly or semi-supervised approaches
  - Data augmentation
  - Transfer learning
  - Synthetic data generation with GANs

- They can all be combined

- In this presentation we will review some recent works to learn from large scale datasets containing weak and noisy labels

Classification in the presence of Label Noise: a Survey (2014) [1/2]

- Classification consists in predicting the classes of new samples by using a model inferred from training data
- Class label noise (mislabeled instances) is an important issue in classification problems
- Sources
  - Labelers are humans
  - Insufficient information provided to labelers
  - Labels may be collected by non-experts
  - Labeling may be subjective
  - Data encoding or communication problems

- Consequences
  - Decreased accuracy of predictions
  - Increased difficulty to identify relevant features
  - Increased inferred models complexity
  - Increased number of necessary training samples
  - Increased distortion of observed frequencies

Classification in the presence of Label Noise: a Survey (2014) [2/2]

- Taxonomy of methods to deal with label noise
  - Label noise-robust models
  - Algorithms not to sensitive to label noise
  - Common losses are not completely robust
  - Label noise handled by avoiding overfitting
  - Ensemble methods can improve robustness

  - Label noise cleaning methods
  - Filter approaches to remove or relabel mislabeled instances
  - Anomaly detection through model predictions or ad hoc measures and thresholds
  - Cheap and easy to implement
  - Overcleansing may reduce the performance of classifiers

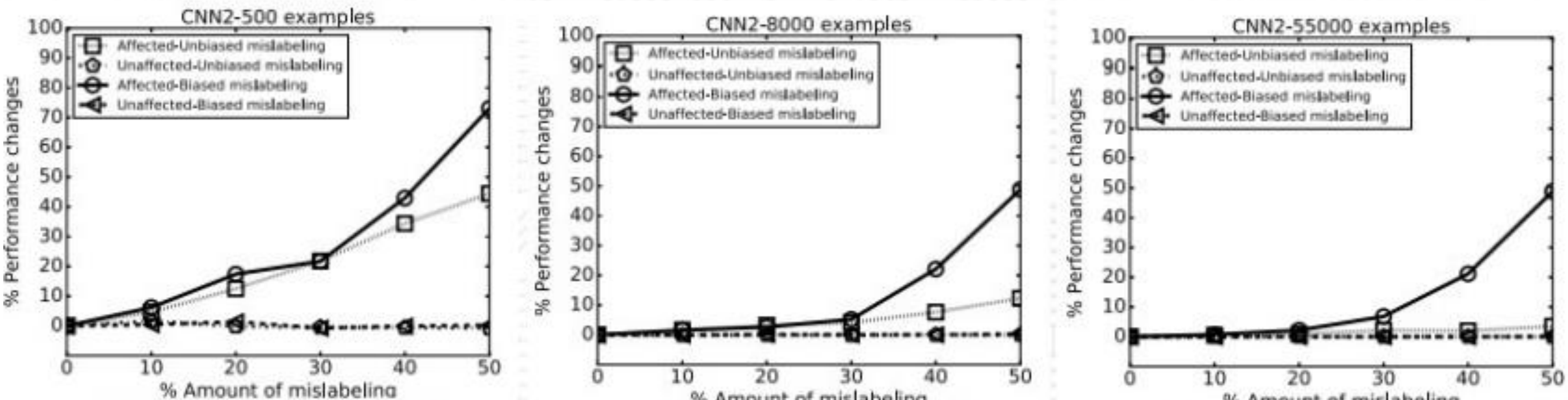
  - Label noise-tolerant algorithms
  - Probabilistic models can take advantage of prior knowledge
  - Label noise can be directly modeled during training
  - Increased complexity and overfitting risk

Impact of biased mislabeling on learning with deep networks (2017) [1/2]

- Types of mislabeling data
  - Unbiased mislabeling: inconsistent noise
  - Biased mislabeling: consistent noise
- Questions
  - What is the degree to which mislabeled data affects the accuracy of classifiers?
  - Is it better to provide a large less accurate training set for deep learning or is it better to put more effort into increasing the accuracy of the data?
- Experiment
  - MNIST dataset
  - Study the impact of mislabeling ratios (0%-10%-20%-30%-40%-50%) on different dataset sizes (500 – 8.000 – 55,000)
  -  biased mislabeling

Impact of biased mislabeling on learning with deep networks (2017) [2/2]

- Results



(a) The CNN2 trained on 500 examples

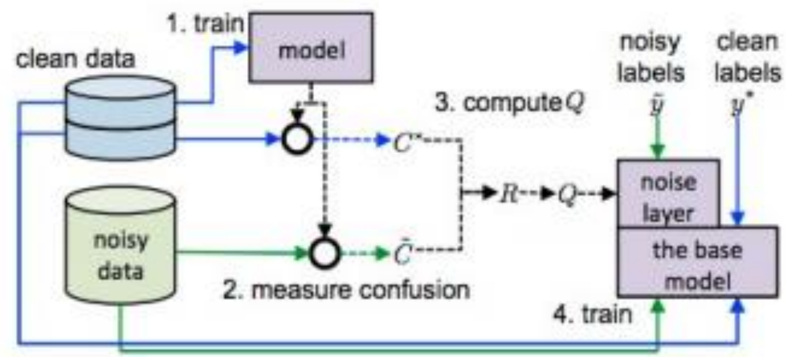
(a) The CNN2 trained on 8000 examples

(a) The CNN2 trained on 55000 examples

- Conclusion
  - Clear tradeoff between data size and amount of mislabeled data
  - Deep Learning can model noise intrinsically on large data sets

Learning from Noisy Labels with Deep Neural Networks (2014) [1/2]

- User tags from social web sites and keywords from image search engines are very noisy labels and unlikely to help training deep neural networks without additional tricks
- Question
  - How to use clean data and high volumes of noisy data for Deep Learning classification tasks?
- Approach
  - Bottom-up noise model



  - Reweighting of noisy data with loss function

$$\mathcal{L}(\theta) = \frac{1}{N_c + N_n}\left(\sum_{n=1}^{N_c} \log p(y = y_n|\mathbf{x}_n, \theta) + \gamma \sum_{n=1}^{N_n} \log p(\tilde{y} = \tilde{y}_n|\tilde{\mathbf{x}}_n, \theta)\right) \tag{7}$$

where N_c and N_n are the number of clean and noisy samples respectively. The hyper-parameter γ is the weight on noisy labels and is set by cross validation.

Learning from Noisy Labels with Deep Neural Networks (2014) [2/2]

- Experiment
  - ImageNet dataset + Web Image Search
  - Study the impact of using 1.3M images with clean labels over 1,000 classes (ImageNet) and 1.4M noisy labeled images (scraped from Internet)

- Results

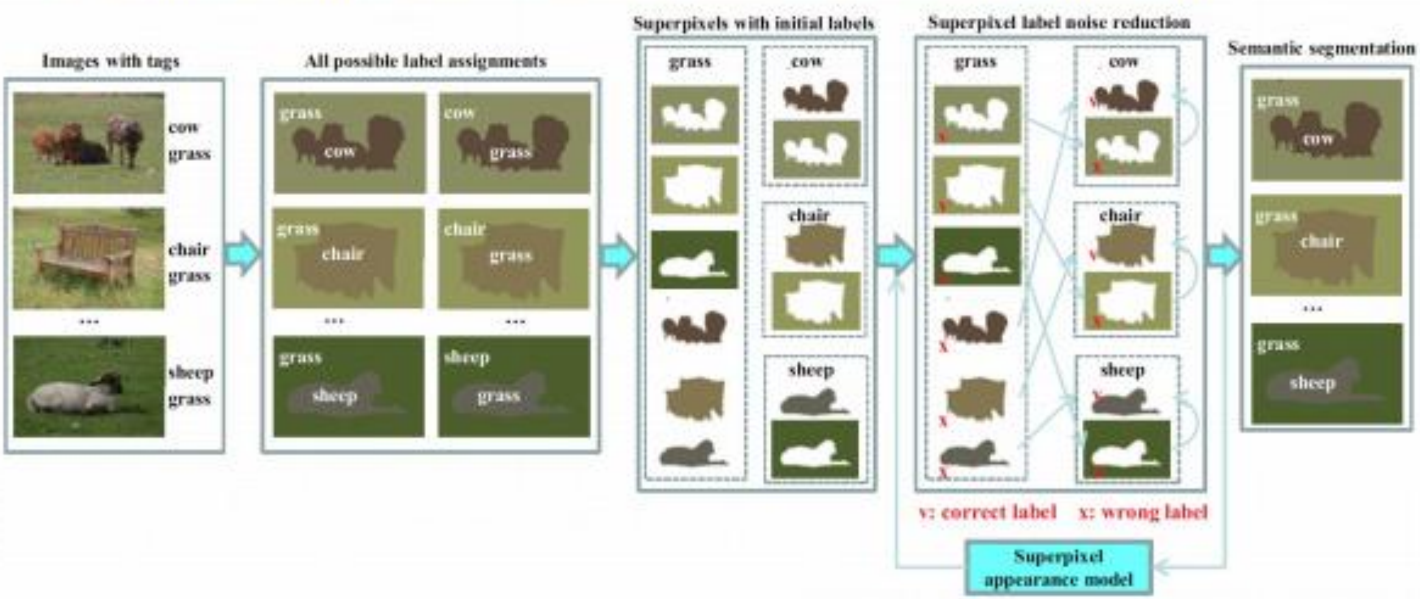
Model	Extra data	Noisy weight γ	Top 5 val. error
Krizhevsky et al. [8]	-	-	18.2%
Krizhevsky et al. [8]	15M full ImageNet	-	16.6%
Conv. net	-	-	18.0%
Conv. net	1.4M noisy images from Internet	1	18.1%
Conv. net		0.1	16.7%
Bottom-up (learned)		0.1	16.5%
Bottom-up (estimated)		0.2	16.6%

Table 3: Error rates of different models on validation images of ImageNet

Darian Frajberg, Applying Deep Learning with Weak and Noisy labels

Learning from Weak and Noisy Labels for Semantic Segmentation (2017) [1/2]

- Weakly supervised semantic segmentation (WSS) aims to learn a segmentation model from weak (image-level) as opposed to strong (pixel-level) labels
- User-tagged images from media sharing sites (e.g. Flicker) can be exploited as unlimited supply of noisy labeled data
- Approach
  - Label noise reduction technique (classes and super-pixels)

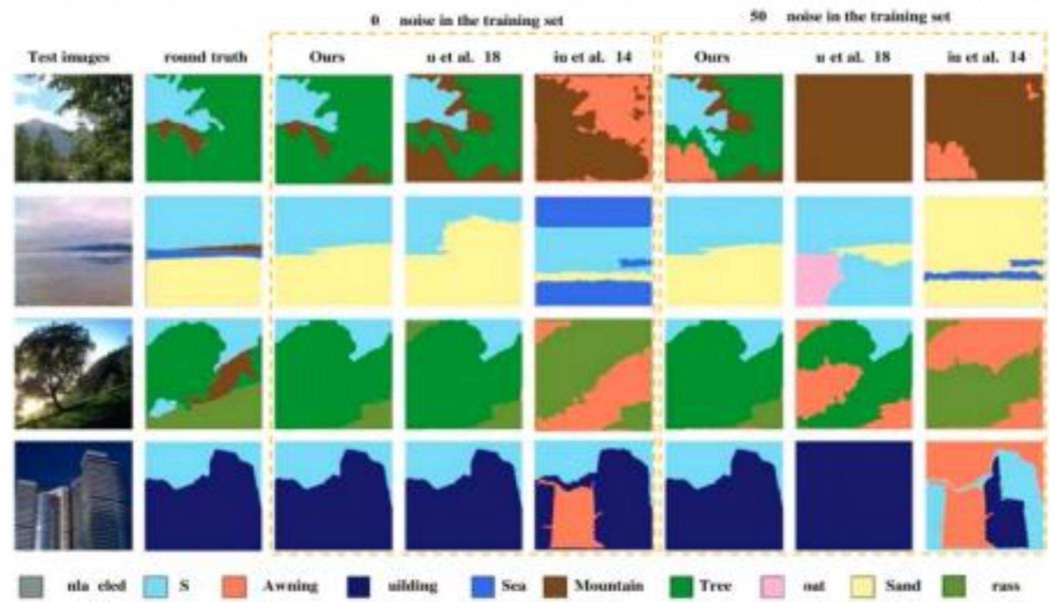


Learning from Weak and Noisy Labels for Semantic Segmentation (2017) [2/2]

- Experiment
  - PASCAL VOC dataset
  - Study the impact of the proposed approach for WSSS
  - Study the impact of noisy class labels for WSSS

- Results

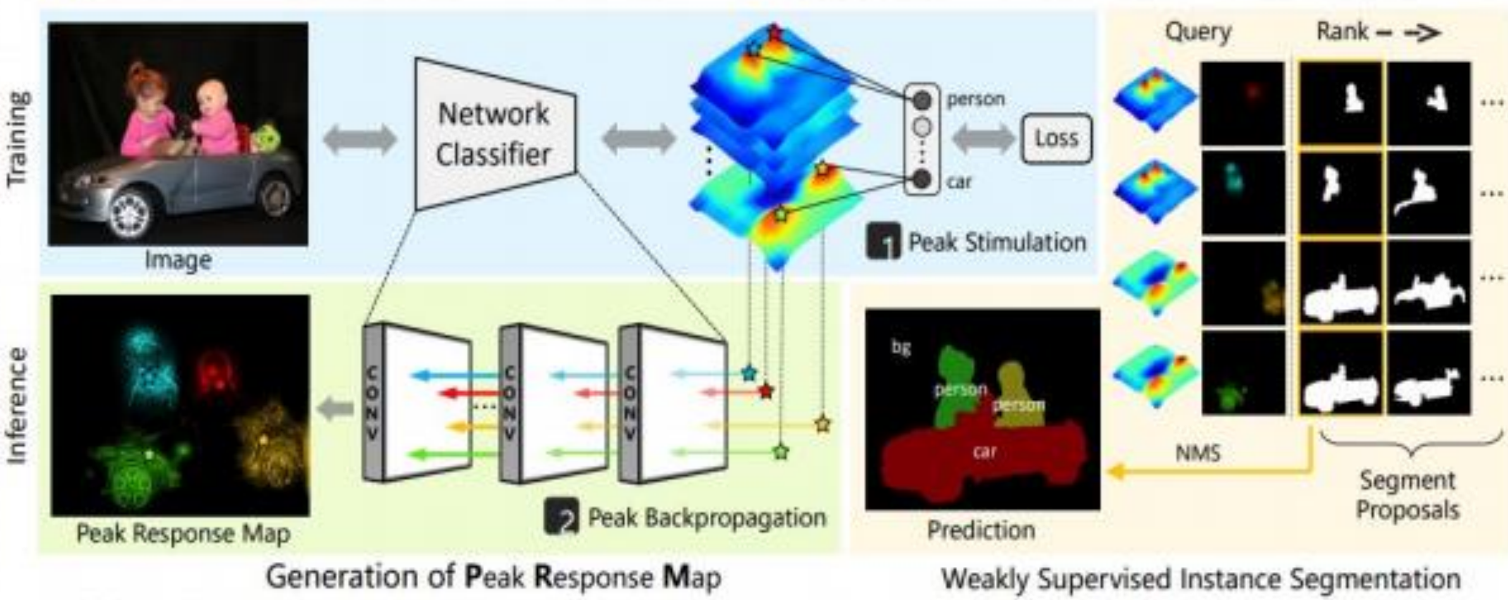
Method	Supervision	Trans.?	per-class (%)	IOU (%)
Upper bound (ours)	full	N	49.2	23.6
Ours (trans.)	weak	Y	**48.9**	**21.6**
Ours	weak	N	48.1	20.8
Xu et al. [18] (trans.)	weak	Y	48.5	19.7
Xu et al. [18]	weak	N	47.8	18.3
Liu et al. [14]	weak	Y	29.8	16.4
Zhang et al. [19]	weak	Y	44.6	N/A
Xie et al. [59]	weak	Y	42.0	N/A
Zhang et al. [15]	weak	N	24.0	N/A
Liu et al. [13]	weak	Y	38.0	N/A
Liu et al. [60]	weak	Y	32.0	N/A
Ladicky et al. [4]	full	N	30.0	N/A
Larlus et al. [61]	full	N	37.2	N/A
Shotton et al. [2]	full	N	42.0	N/A
Girshick et al. [53]	full	N	**43.0**	**26.7**



Weakly Supervised Instance Segmentation using Class Peak Response (2018) [1/2]

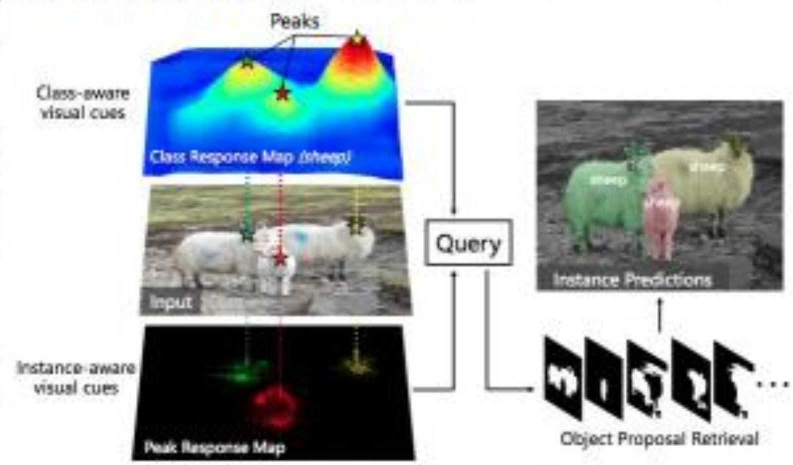
- Weakly supervised instance segmentation (WSIS) is similar to weakly supervised semantic segmentation (WSSS), but it also aims to predict precise masks for each object instance.

- Approach



Weakly Supervised Instance Segmentation using Class Peak Response (2018) [2/2]

- Experiment
  - PASCAL VOC dataset
  - Study the impact of the proposed approach for WSIS



- Results

Method		$mAP^r_{0.25}$	$mAP^r_{0.5}$	$mAP^r_{0.75}$	ABO
Ground Truth	Rect.	78.3	30.2	4.5	47.4
	Ellipse	81.6	41.1	6.6	51.9
	MCG	69.7	38.0	12.3	53.3
Training requires image-level labels and object proposals					
MELM [39]	Rect.	36.0	14.6	1.9	26.4
	Ellipse	36.8	19.3	2.4	27.5
	MCG	36.9	22.9	8.4	32.9
Training requires only image-level labels					
CAM [47]	Rect.	18.7	2.5	0.1	18.9
	Ellipse	22.8	3.9	0.1	20.8
	MCG	20.4	7.8	2.5	23.0
SPN [48]	Rect.	29.2	5.2	0.3	23.0
	Ellipse	32.0	6.1	0.3	24.0
	MCG	26.4	12.7	4.4	27.1
PRM (Ours)		**44.3**	**26.8**	**9.0**	**37.6**

Table 4: Weakly supervised instance segmentation results on the PASCAL VOC 2012 val. set in terms of mean average precision (mAP%) and Average Best Overlap (ABO).

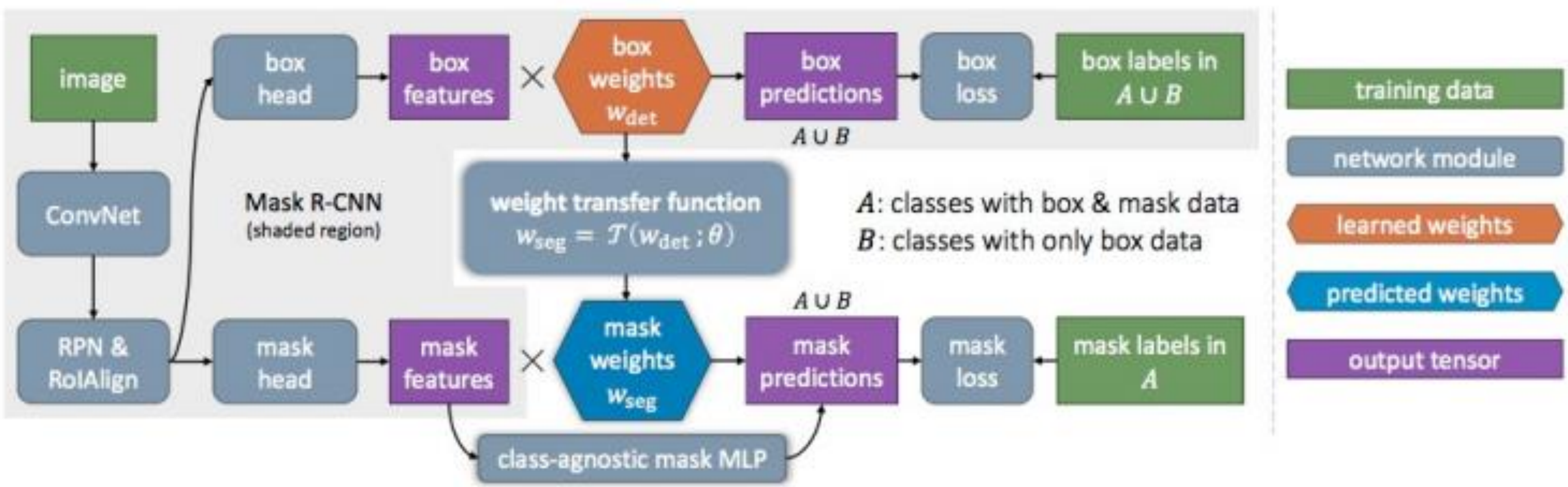
	ResNet50					VGG16	
Peak Stimulation		✓	✓	✓	✓		✓
Instance-aware term	✓		✓	✓	✓	✓	✓
Class-aware term	✓	✓		✓	✓	✓	✓
Boundary-aware term	✓	✓	✓		✓	✓	✓
$mAP^r_{0.5}$	22.8	13.3	16.5	24.3	**26.8**	11.9	22.0

Table 5: Ablation study on the PASCAL VOC2012 val. set based on different network backbones.

Learning to Segment Every Thing (2018) [1/3]

- Partially supervised instance segmentation (PSIS) aims at using bounding box annotations to train a model for instance segmentation

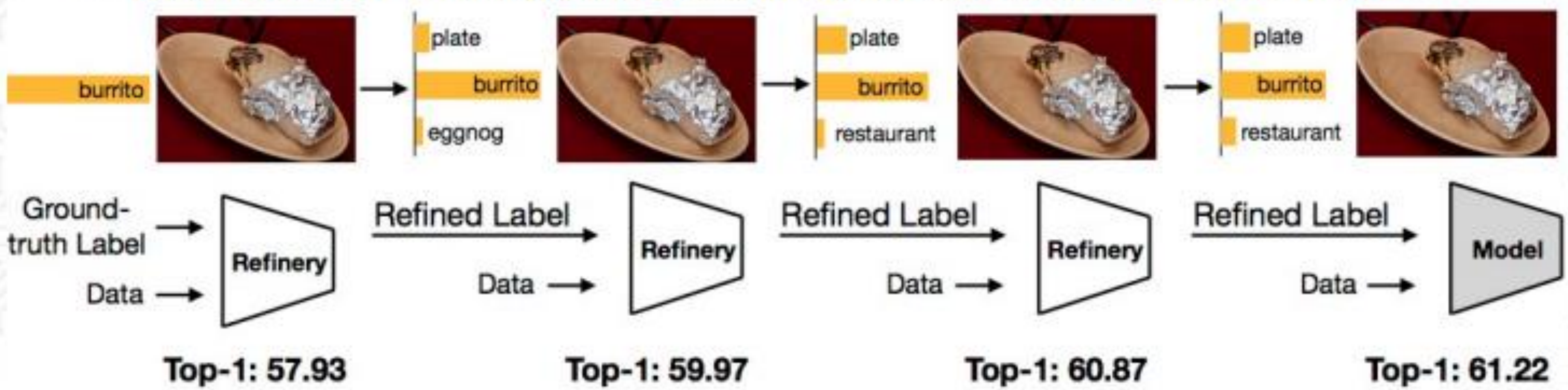
- Approach
  - Transfer learning



Darian Frajberg, Applying Deep Learning with Weak and Noisy labels

Label Refinery: Improving ImageNet Classification through Label Progression (2018) [1/2]

- Supervised learning relies on three main components: data, labels and models. However, labels, which are often incomplete, ambiguous and redundant, are not usually studied.
- Question
  - How to modify labels during training so as to improve generalization and accuracy of learning models?
- Approach
  - Iterative soft, multicategory, dynamically-generated label refinery



Conclusions

- Data collection for supervised learning tasks is not always feasible
- There are many alternatives to exploit public available noisy data
- Weakly supervised learning techniques can be very effective
- Supervised learning can be boosted by complementing it with the presented techniques