Deep Self-Learning from Noisy Labels

Jiangfan Han[1] Ping Luo[2] Xiaogang Wang[1]

[1]The Chinese University of Hong Kong

[2]The University of Hong Kong

The IEEE International Conference on Computer Vision (ICCV), 2019

Presented by Fatir Qureshi

Overview

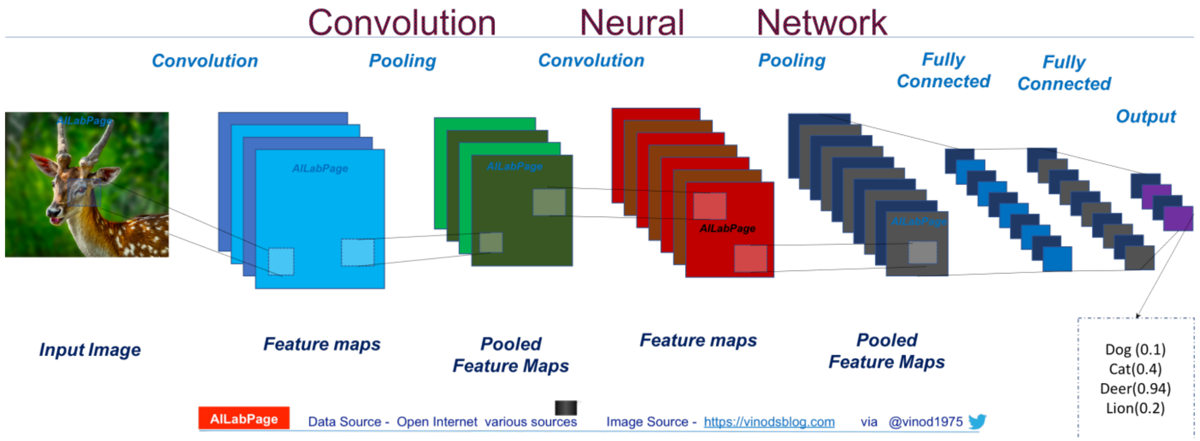
- Introduction
- Related Work
- Approach
- Experiments

Introduction

- In order to perform well on deep learning tasks, large scale datasets with clean annotations are required
- Collecting such difficult is costly and time consuming and incorrect labels may persist
- By developing ways to utilize image-level tags as queries it may be possible to overcome the constraints of relying on highly cleaned and processed data

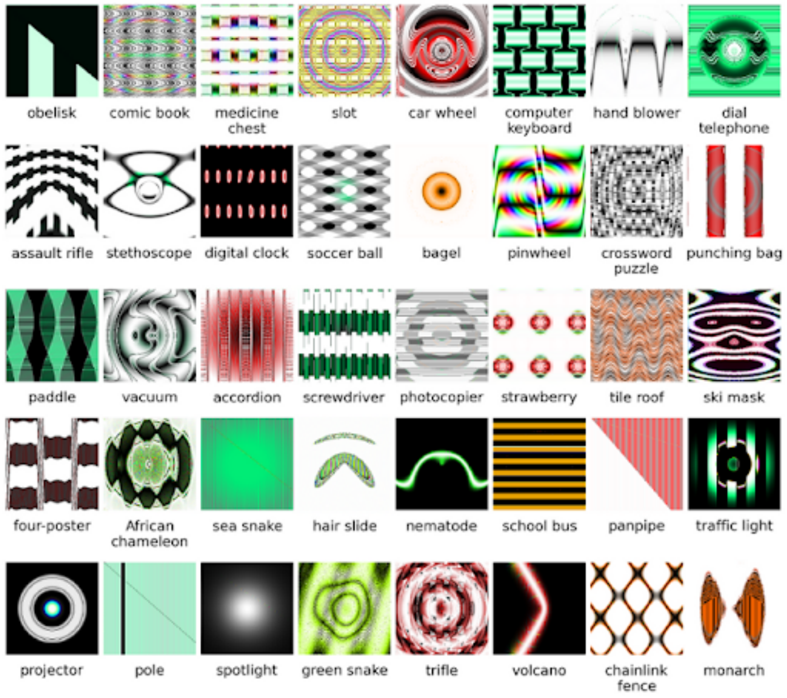
Introduction

- Convolutional neural networks have achieved great successes when trained with clean data for computer vision tasks
- As annotating a large-scale clean and unbiased dataset is expensive and time-consuming, many efforts have been made to improve the robustness of CNNs trained on noisy datasets
- Four major approaches have been used to improve robustness to noisy labeling
  - Transition Matrices
  - Robust loss functions
  - Additional supervision
  - CleanNet derived additional network



Data Source - Open Internet various sources Image Source - https://vinodsblog.com via @vinod1975

Introduction



- Such techniques have several limitations
- Assumption that there is a single transition probability between the noisy label and ground-truth label, and this probability is independent of individual samples
- Alternatively, these approaches are costly to implement are difficult to apply in large-scale real-world scenarios
- Additional supervision impractical for large datasets

Introduction

- CleanNet, a joint neural embedding network, achieved the existing state-of-the-art performance on real-world dataset such as Clothing1M, but needed additional information or supervision to train
- A single class prototype is hard to represent all characteristics of a category.
- Authors propose a novel framework of Self-Learning with Multi-Prototypes (SMP), which aims to train a robust network on the real noisy dataset without extra supervision

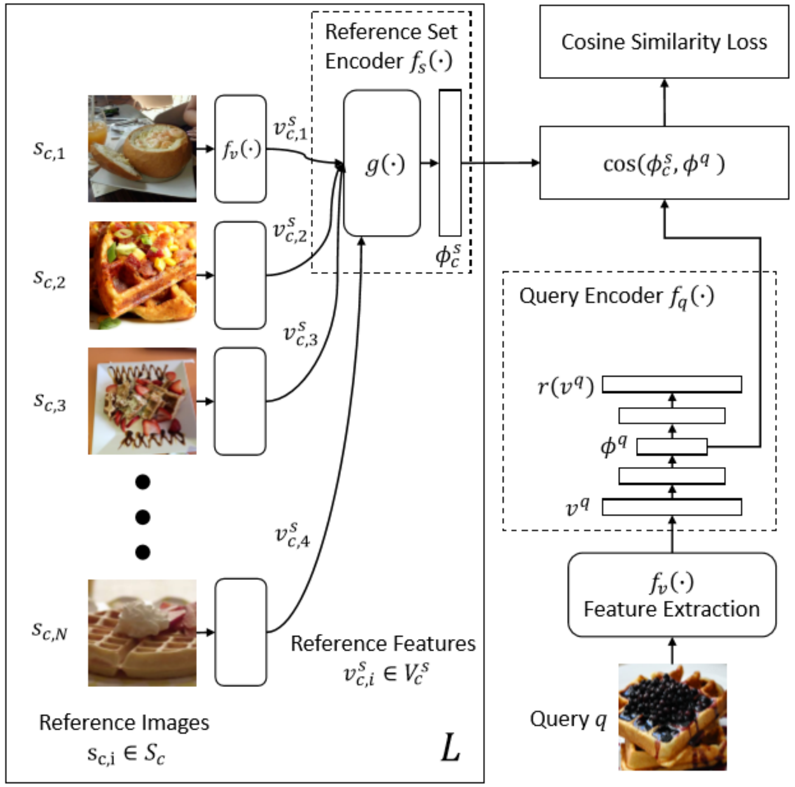


Figure 1. CleanNet architecture for learning a class embedding vector ϕ^s_c and a query embedding vector ϕ_q with a similarity matching constraint. There exists one class embedding for each of the L classes.

Introduction

SMP trains in an iterative manner which contains two phases

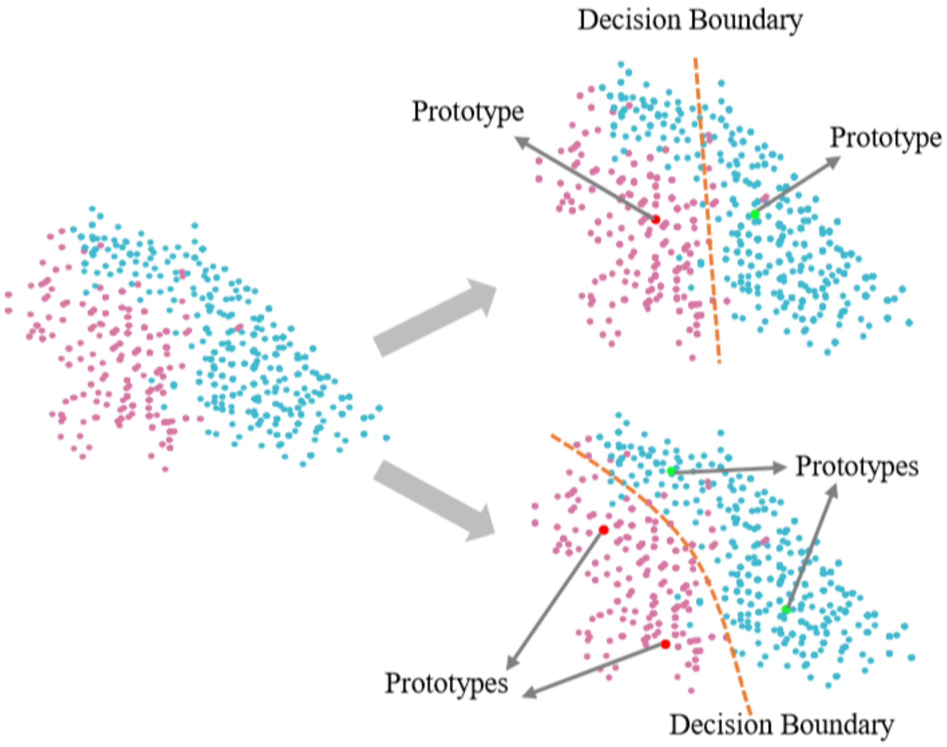
- Train a network with the original noisy label and corrected label generated in the second phase
- Use the network trained in the first stage to select several prototypes. These prototypes are used to generate the corrected label for the first stage

Authors lay out two key contributions of this work

- Propose an iterative learning framework SMP to relabel the noisy samples and train ConvNet on the real noisy dataset, without using extra clean supervision
- SMP results in state-of-the-art findings for learning from noisy data

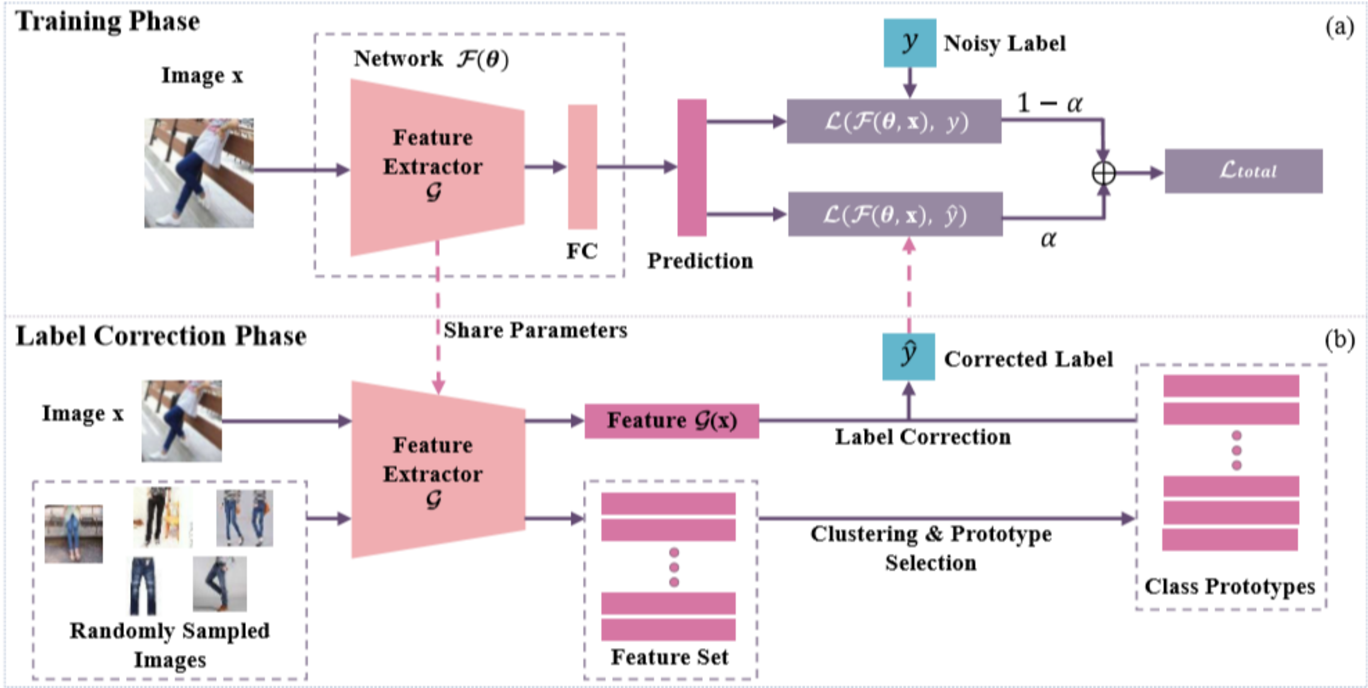
Approach

- By observing the characteristics of samples in the same noisy category, the authors believe that these samples have widely spread distribution
- A single class prototype is hard to represent all characteristics of a category
- Self-Learning with Multi-Prototypes (SMP) will subsequently overcome this limitation

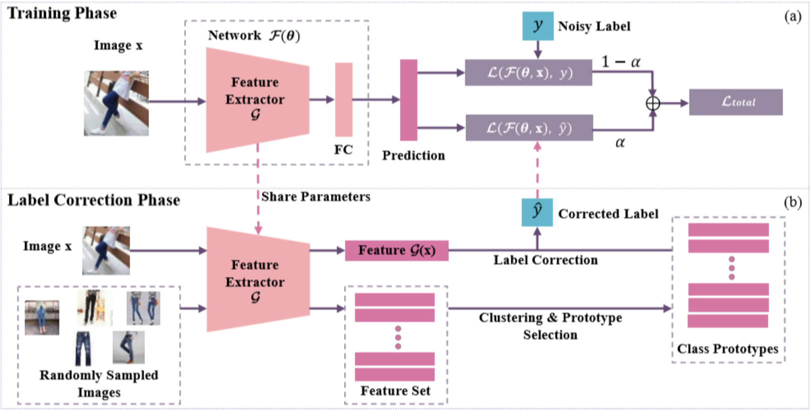


Approach

- Approach consists of two main phases, training and label corrector
- In the training phase, a neural network with parameters θ is trained, taking image x as input and producing the corresponding label prediction F(θ, x).


Training Phase
(a)
Image x
Network $\mathcal{F}(\theta)$
Feature Extractor $\mathcal{G}$
FC
Prediction
y Noisy Label
$\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), y)$
$1-\alpha$
$\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), \hat{y})$
α
$\mathcal{L}_{total}$
Share Parameters
Label Correction Phase
(b)
$\hat{y}$ Corrected Label
Image x
Feature Extractor $\mathcal{G}$
Feature $\mathcal{G}$(x)
Label Correction
Randomly Sampled Images
Feature Set
Clustering & Prototype Selection
Class Prototypes

Approach


Training Phase
(a)
Image x
Network $\mathcal{F}(\theta)$
Feature Extractor $\mathcal{G}$
FC
Prediction
y Noisy Label
$\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),\ y)$
$\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),\ \hat{y})$
$1-\alpha$
α
$\mathcal{L}_{total}$
Share Parameters
Label Correction Phase
(b)
Image x
Feature Extractor $\mathcal{G}$
Feature $\mathcal{G}(x)$
$\hat{y}$ Corrected Label
Label Correction
Randomly Sampled Images
Feature Set
Clustering & Prototype Selection
Class Prototypes

- When training on a cleanly-labeled dataset, the optimization problem is defined as

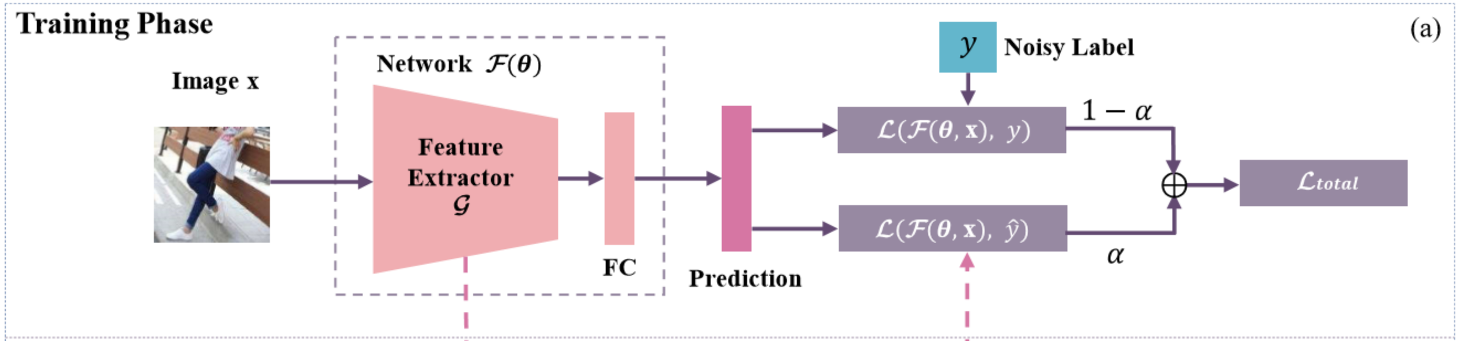
$$\theta^* = \text{argmin}_\theta\, \mathcal{L}(Y, \mathcal{F}(\theta, \mathbf{X}))$$

, where dataset D = {X,Y } = {(x1,y1),...,(xN,yN)},

- Authors propose to attain the corrected label ˆY (X, Xs) in a self-training manner, where Xs indicates a set of class prototypes to represent the distribution of classes

$$\theta^* = \text{argmin}_\theta\, \mathcal{L}(Y, \hat{Y}(\mathbf{X}, \mathbf{X}_s), \mathcal{F}(\theta, \mathbf{X}))$$

Training Phase


Training Phase
(a)
Image x
Network $\mathcal{F}(\theta)$
Feature Extractor $\mathcal{G}$
FC
Prediction
y Noisy Label
$\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),\ y)$
$1-\alpha$
$\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),\ \hat{y})$
α
$\mathcal{L}_{total}$

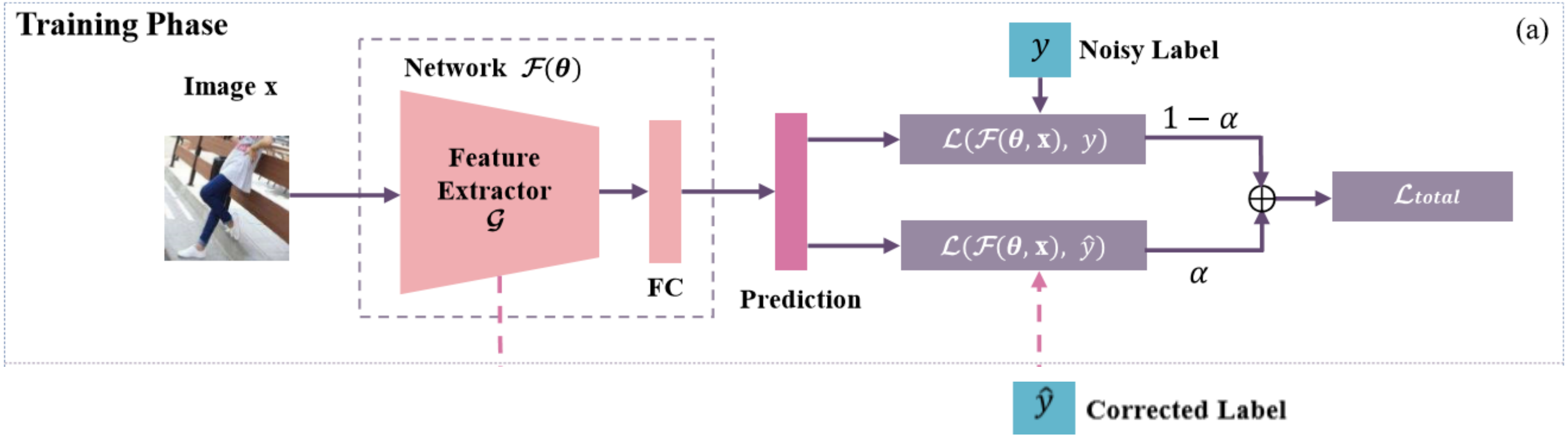
- The objective function is the empirical risk of cross-entropy loss, which is formulated by:

$$\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),y)=-\frac{1}{n}\sum_{i=1}^{n}\log(\mathcal{F}(\theta,\mathbf{x}_i)_{y_i})$$

where n is the mini-batch size and yi is the label corresponding to the image xi

- When learning on a noisy dataset, the original label yi may be incorrect, so this approach introduces another corrected label as a complementary supervision

Training Phase

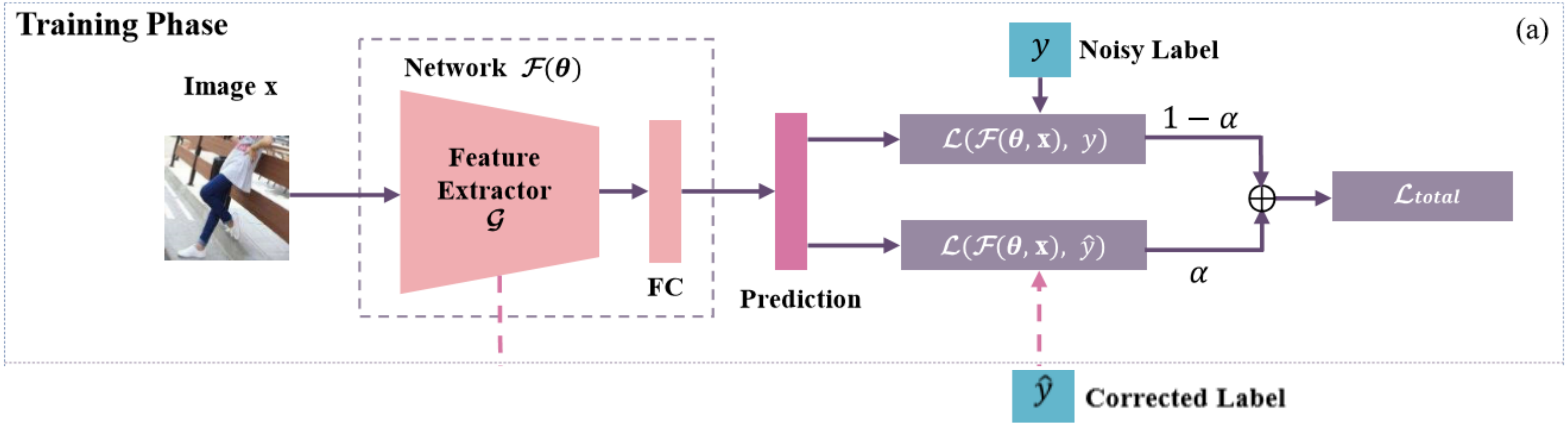

Training Phase
(a)
Image x
Network $\mathcal{F}(\boldsymbol{\theta})$
Feature Extractor $\mathcal{G}$
FC
Prediction
y Noisy Label
$\mathcal{L}(\mathcal{F}(\boldsymbol{\theta},\mathbf{x}),\ y)$
$1-\alpha$
$\mathcal{L}(\mathcal{F}(\boldsymbol{\theta},\mathbf{x}),\ \hat{y})$
α
$\mathcal{L}_{total}$
$\hat{y}$ Corrected Label

- With the corrected signal, the objective loss function is:

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),y) + \alpha\mathcal{L}(\mathcal{F}(\theta,\mathbf{x}),\hat{y})$$

where L is the cross entropy loss as shown in the original objective function; y is the original noisy label, and ˆ y is the corrected label

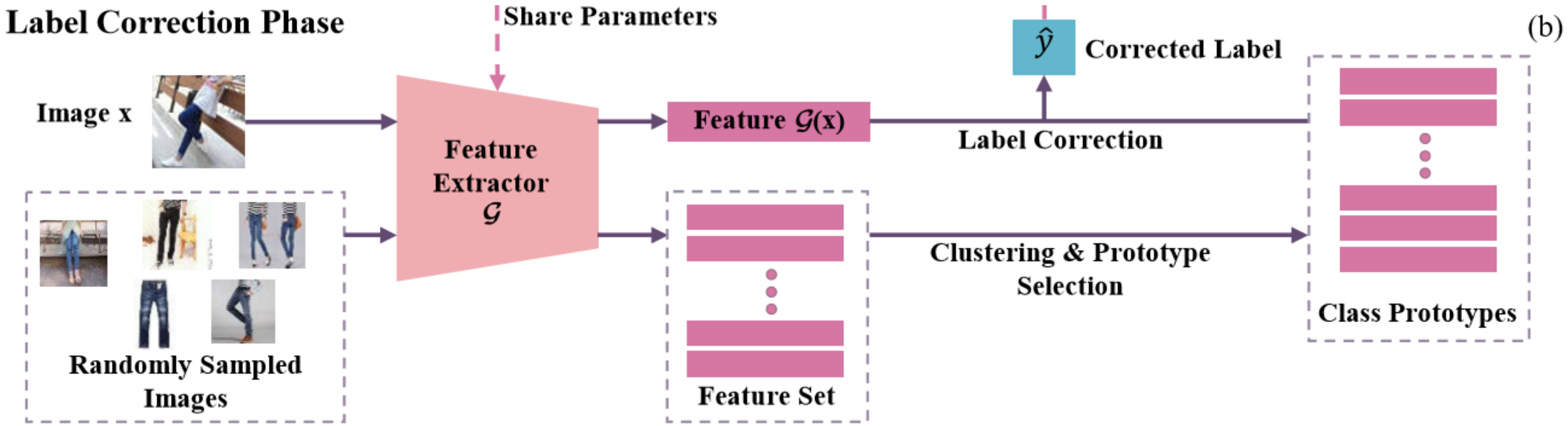
Training Phase



Training Phase
(a)
Image x
Network $\mathcal{F}(\theta)$
Feature Extractor $\mathcal{G}$
FC
Prediction
y Noisy Label
$\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}),\ y)$
$1-\alpha$
$\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}),\ \hat{y})$
α
$\mathcal{L}_{total}$
$\hat{y}$ Corrected Label

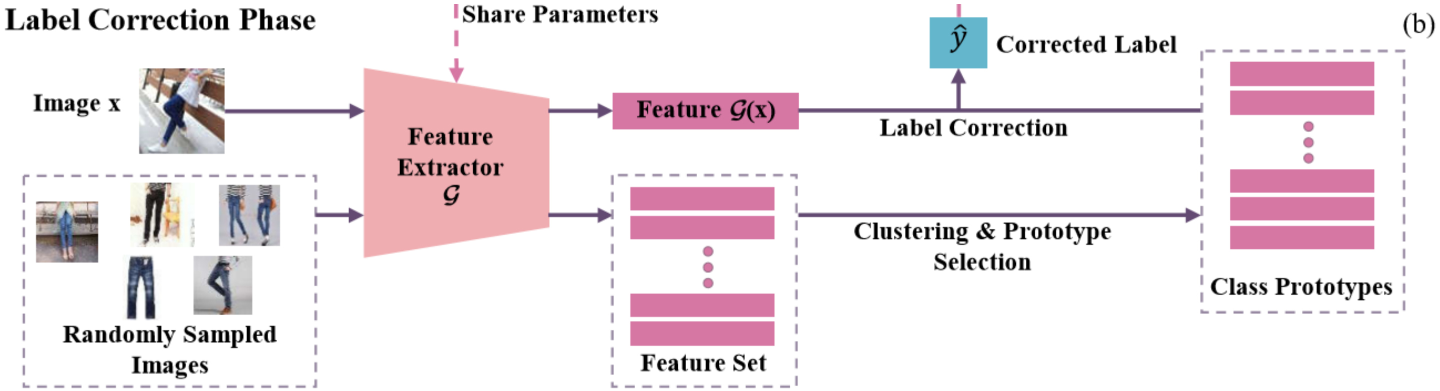
- For the first iteration of training, α is set to 0 and they train the network F by using only the original noisy label y
- After a preliminary network was trained, it is possible to step into the second phase and obtain the corrected label

Label Correction Phase



- The goal of this step is to obtain a corrected label for each image in the training set
- Initially, several class prototypes for each category must be selected

Label Correction Phase – Select prototypes

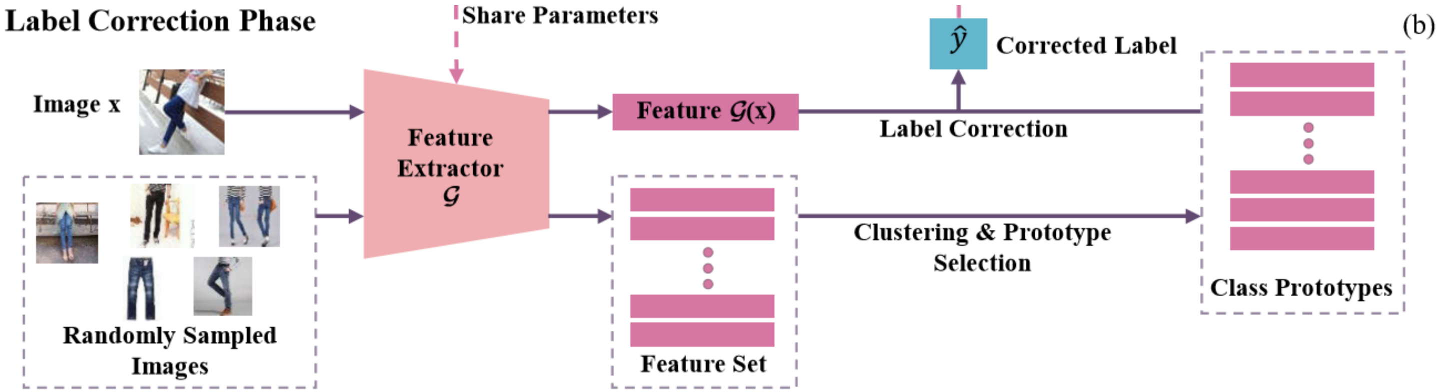


- Preliminary network trained in the first phase to extract deep features of images in the training set

- Method employs the ResNet architecture, where the output before the fully connected layer is regarded as the deep features, denoted as G(x)

- The relationship between F(θ, x) and G(x) is F(θ, x) = f(G(x)), where f is the operation on the fully-connected layer of ResNet

Label Correction Phase – Select prototypes



- To select class prototypes for a category, the method extracts a set of deep features, {G(xi)}n i=1, corresponding to a set of images {xi}n i=1 in the dataset with the same noisy label
- Once extracted, the method calculates the cosine similarity between the deep features and constructs a similarity matrix
- S ∈ Rn×n, n is the number of images with noisy label c and Sij ∈ S with

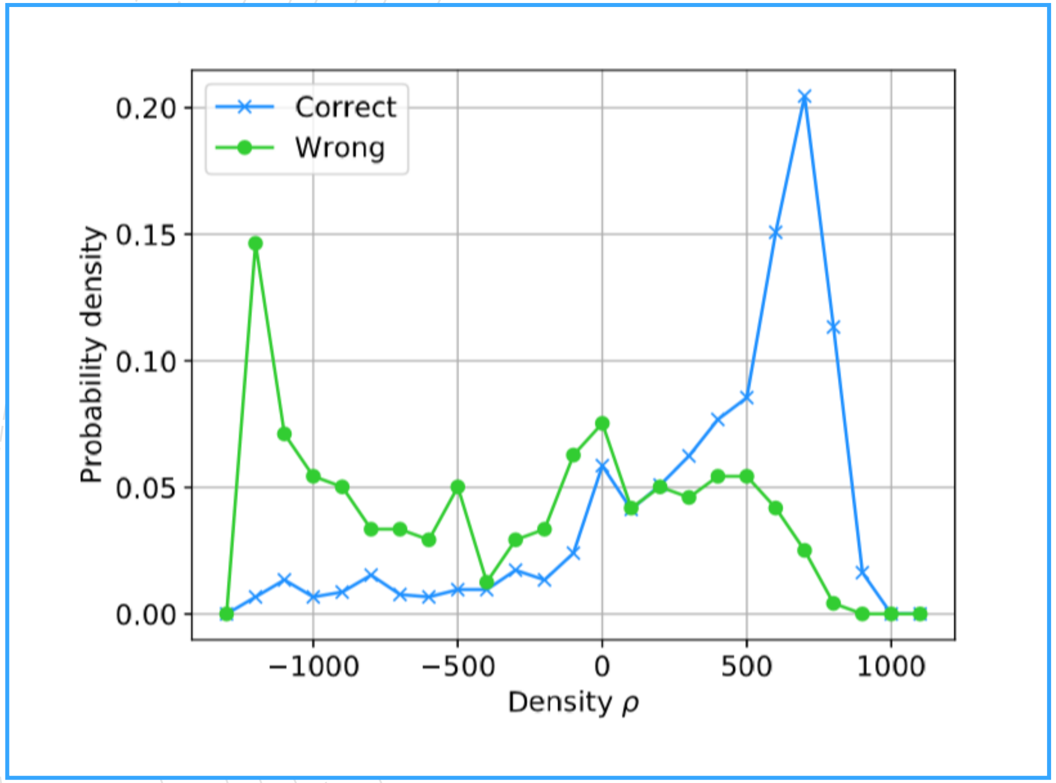
$$S_{ij} = \frac{\mathcal{G}(\mathbf{x}_i)^T \mathcal{G}(\mathbf{x}_j)}{||\mathcal{G}(\mathbf{x}_i)||_2 ||\mathcal{G}(\mathbf{x}_j)||_2}$$

Label Correction Phase

- Sij is a measurement of the similarity between two images xi and x_j and x_j
- Cosine similarity was found by experimental trials to be better for improving label accuracy through experimental trials than Euclidian distance
- Calculating cosine similarity matrix S time-consuming, so the approach randomly samples m images (m < n) in the same class to calculate the similarity matrix S^{m*m}
- To select prototypes, the authors define a density ρi for each image xi, where sign(x) is the sign function and S_c is a constant

$$\rho_i = \sum_{j=1}^{m} \operatorname{sign}(S_{ij} - S_c)$$

Choosing Prototypes

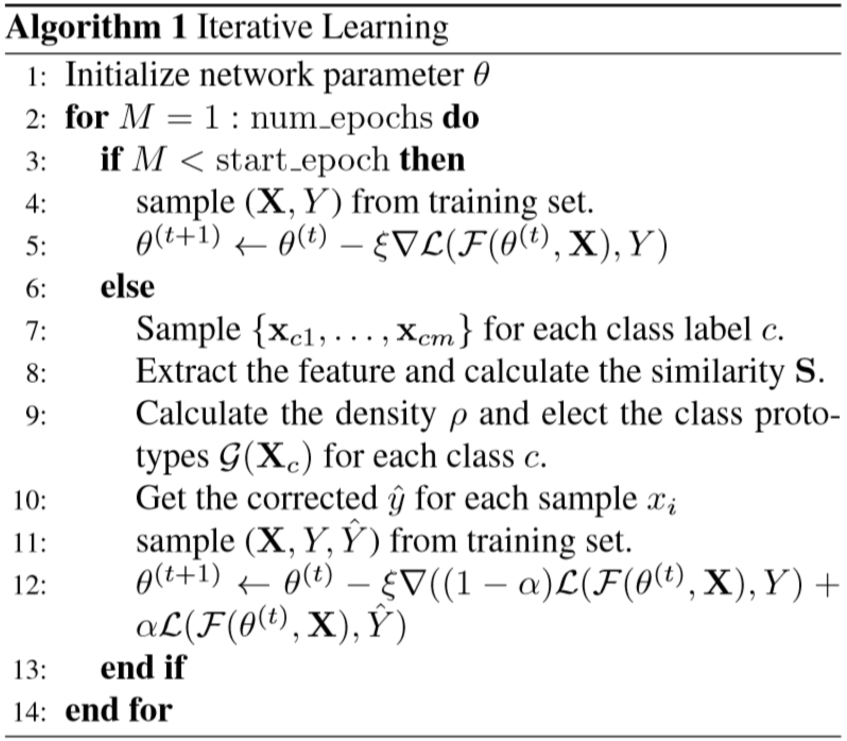


- The images with larger ρ have more similar images around them
- Images with correct labels are more likely to have large ρ value while those images with wrong labels appear in the region with low ρ.
- If we need p prototypes for a class, we can regard the images with the top-p highest density values as the class prototypes
- If the chosen p prototypes belonging to the same class are very close to each other, little advantage to SMP

Recap

- The training phase and the label correction phase proceed iteratively
    - Training phase first trains an initial network by using image x with noisy label y, as no corrected label is determined yet

- Cosine similarity used as a metric for determining the prototypes
- Randomly sample m images from the noisy dataset for each class and extract features by F, and then the prototype selection procedure selects p prototypes for each class.
- Corrected label ˆ y is assigned to every image x by calculating the similarity between its features G(x) and the prototypes
- Corrected label ˆ y is then used to train the network F in the next epoch



Algorithm 1 Iterative Learning

```
Initialize network parameter $\theta$
for $M = 1$ : num_epochs do
    if $M <$ start_epoch then
        sample $(\mathbf{X}, Y)$ from training set.
        $\theta^{(t+1)} \leftarrow \theta^{(t)} - \xi\nabla\mathcal{L}(\mathcal{F}(\theta^{(t)}, \mathbf{X}), Y)$
    else
        Sample $\{\mathbf{x}_{c1}, \ldots, \mathbf{x}_{cm}\}$ for each class label $c$.
        Extract the feature and calculate the similarity $\mathbf{S}$.
        Calculate the density $\rho$ and elect the class prototypes $\mathcal{G}(\mathbf{X}_c)$ for each class $c$.
        Get the corrected $\hat{y}$ for each sample $x_i$
        sample $(\mathbf{X}, Y, \hat{Y})$ from training set.
        $\theta^{(t+1)} \leftarrow \theta^{(t)} - \xi\nabla((1-\alpha)\mathcal{L}(\mathcal{F}(\theta^{(t)}, \mathbf{X}), Y) + \alpha\mathcal{L}(\mathcal{F}(\theta^{(t)}, \mathbf{X}), \hat{Y})$
    end if
end for
```

Experiments

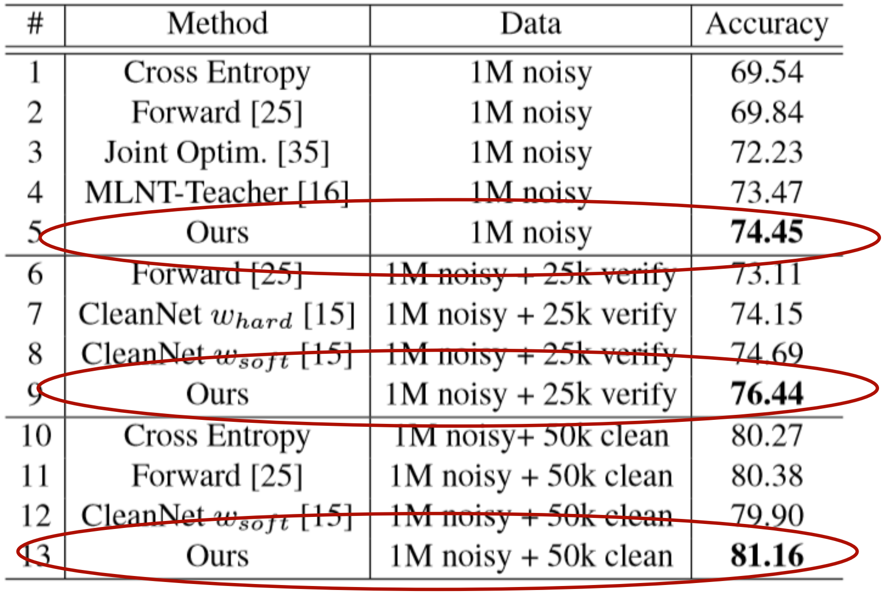
- The authors carried out three different types of experiments to evaluate their method
- They evaluated:
  - Ability to correctly classify by training on Clothing1M
  - Label correction accuracy on Clothing1M
  - Ability to correctly classify by training on Food-101N

Experiments: Clothing1M

- Clothing 1M contains 1 million images of clothes, which are classified into 14 categories
- It is partitioned into training, validation and testing sets, containing 50k, 14k and 10k images respectively
- Human annotators are asked to clean a set of 25k labels as a clean set, but these will not be used with the authors classification approaches
- Experimental approach used ResNet50 pretrained on the ImageNet
- Data preprocessing procedure includes resizing the image with a short edge of 256 and randomly cropping a 224×224 patch from the resized image

Experiments: Clothing1M

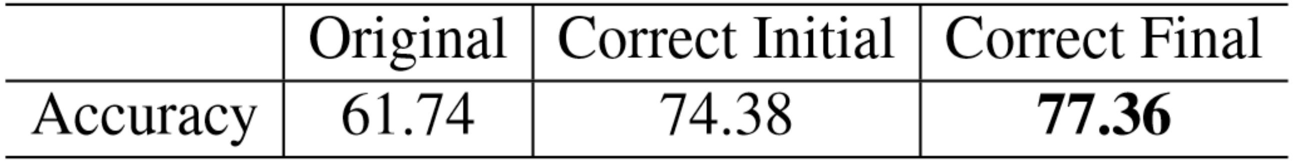
- Three settings were adopted for the approach taken by the authors using the Clothing1M dataset
  - Only the noisy data is used to train their method
  - Verification labels are provided, but they are not used to train the network directly
  - Both noisy dataset and 50k clean labels are available for training



#	Method	Data	Accuracy
1	Cross Entropy	1M noisy	69.54
2	Forward [25]	1M noisy	69.84
3	Joint Optim. [35]	1M noisy	72.23
4	MLNT-Teacher [16]	1M noisy	73.47
5	Ours	1M noisy	**74.45**
6	Forward [25]	1M noisy + 25k verify	73.11
7	CleanNet w_{hard} [15]	1M noisy + 25k verify	74.15
8	CleanNet w_{soft} [15]	1M noisy + 25k verify	74.69
9	Ours	1M noisy + 25k verify	**76.44**
10	Cross Entropy	1M noisy+ 50k clean	80.27
11	Forward [25]	1M noisy + 50k clean	80.38
12	CleanNet w_{soft} [15]	1M noisy + 50k clean	79.90
13	Ours	1M noisy + 50k clean	**81.16**

Ablation Study

- Authors sought to examine the classification accuracy in the label-correction phase



	Original	Correct Initial	Correct Final
Accuracy	61.74	74.38	**77.36**

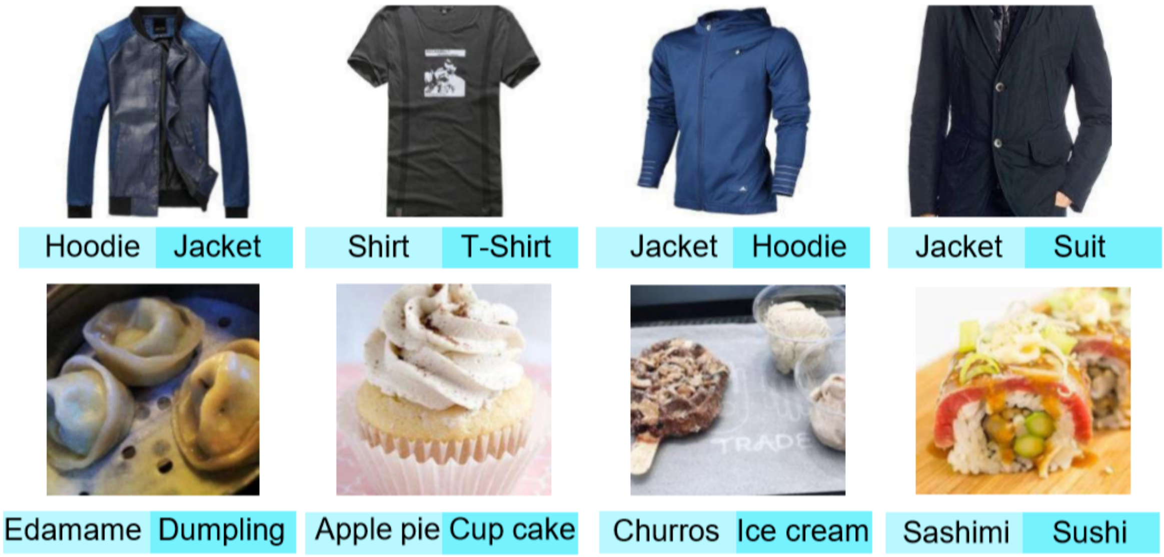
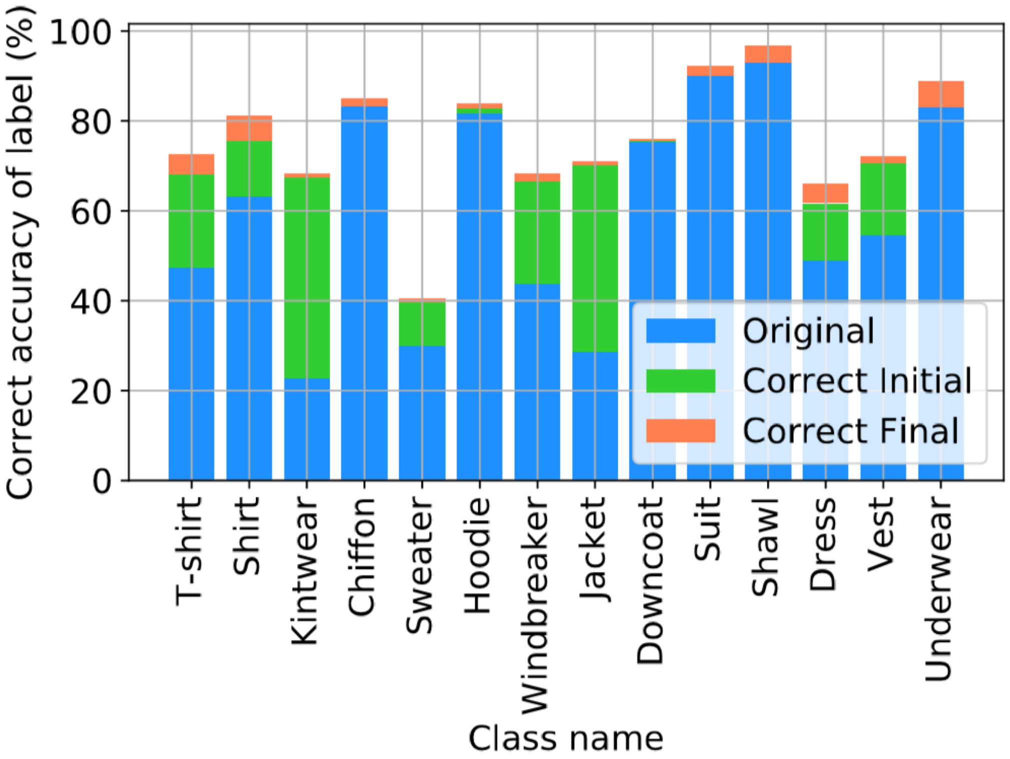
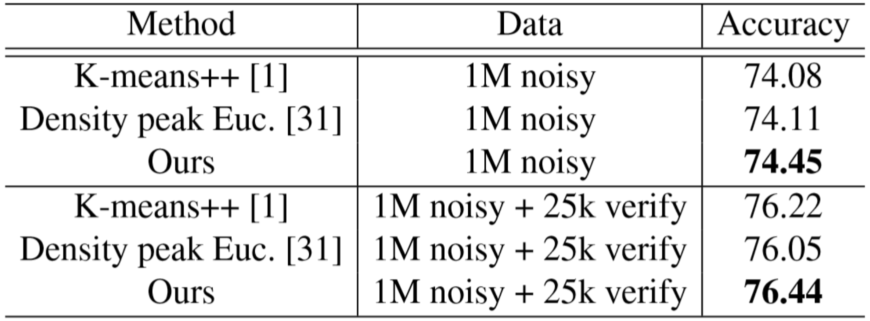


Figure 5. Samples corrected by our method. **Left**: The original



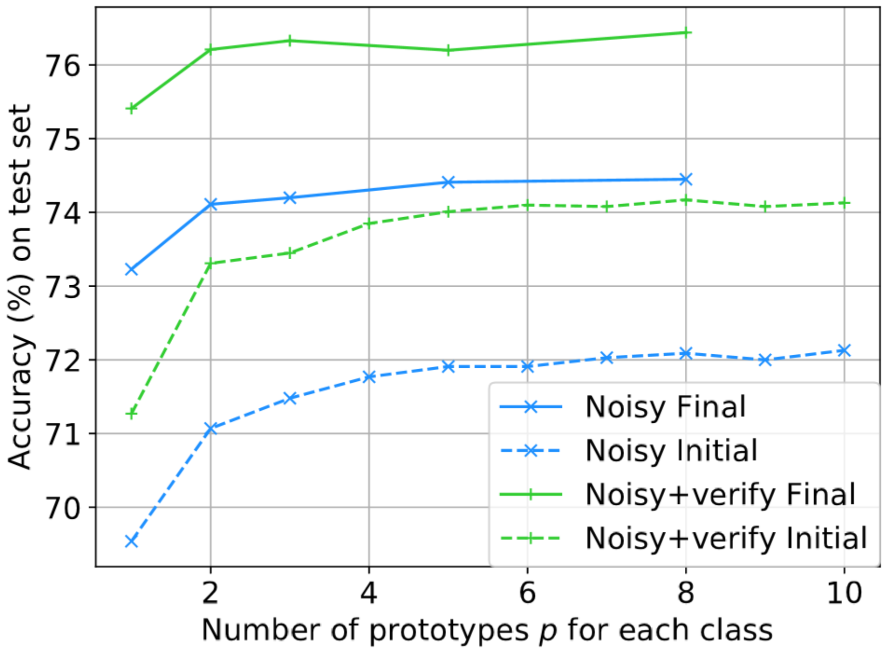
Experiments: Clothing1M

- The number of class prototypes is the key to the representation ability to a class
- The plot below shows the effect of changing the number of prototypes for each class
- Alternative prototype selection methods were also considered
  - K means
  - Euclidian distance



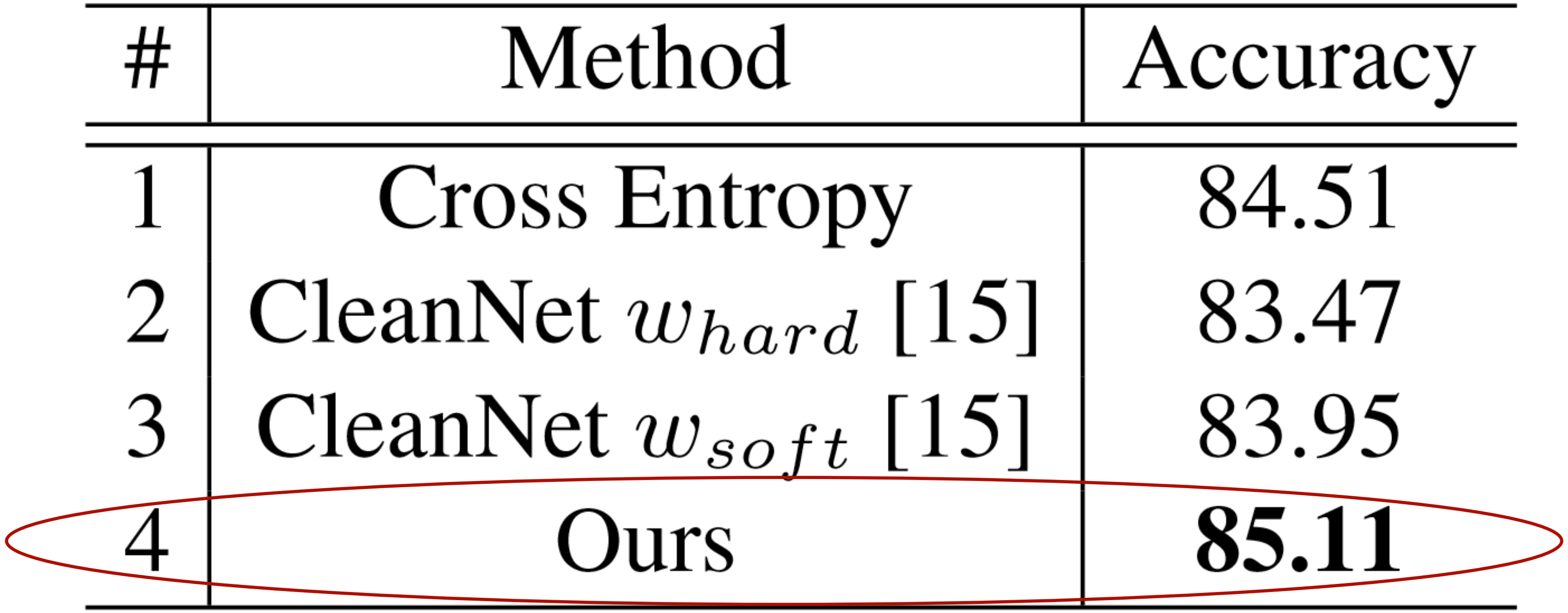
Method	Data	Accuracy
K-means++ [1]	1M noisy	74.08
Density peak Euc. [31]	1M noisy	74.11
Ours	1M noisy	**74.45**
K-means++ [1]	1M noisy + 25k verify	76.22
Density peak Euc. [31]	1M noisy + 25k verify	76.05
Ours	1M noisy + 25k verify	**76.44**

Table 4. The classification accuracy (%) on Clothing1M with different cluster methods used to select the prototypes.



Food 101-N

- Similar protocol implemented for Food 101-N as with the clothing dataset
- Their method had also achieved state-of-the-art performance



#	Method	Accuracy
1	Cross Entropy	84.51
2	CleanNet w_{hard} [15]	83.47
3	CleanNet w_{soft} [15]	83.95
4	Ours	**85.11**

Conclusions

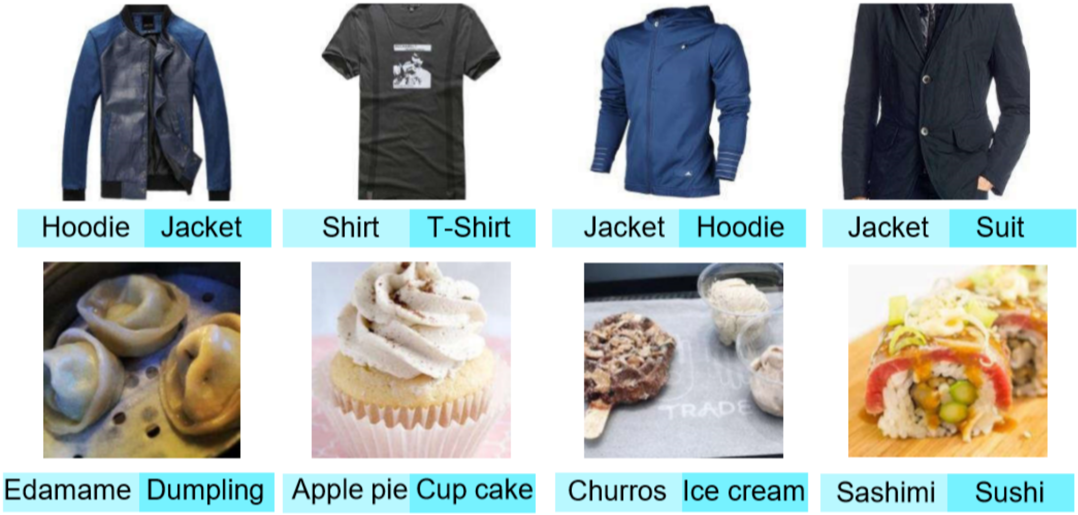


Figure 5. Samples corrected by our method. **Left**: The original

- Demonstrated the advantages of having a multiple prototype approach to label correction
- By correcting the label using several class prototypes and training the network jointly using the corrected and original noisy iteratively, this work provides an effective end-to-end training framework without using an accessorial network or adding extra supervision on a real noisy dataset
- State-of-the-art performance achieved on noisy datasets

Questions?