CNN for CV

AI for CV Group
2020

Week 9.
Classification

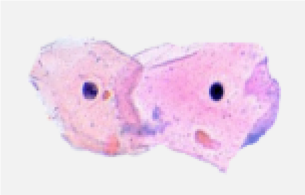
Contents:

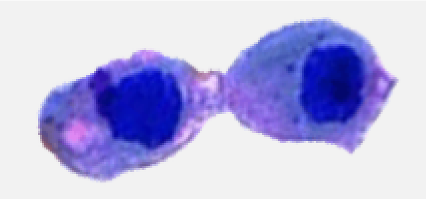
I. Classification Outlines

I. Classification Outlines

I. Classification Outlines

A. Binary Classification:





Non-linearity:

Sigmoid: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

Loss:

Cross Entropy: $H(p,q) = -\sum_i p_i log q_i$

Non-linearity + Loss:

$J(\theta) = -ylog(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$

PyTorch:

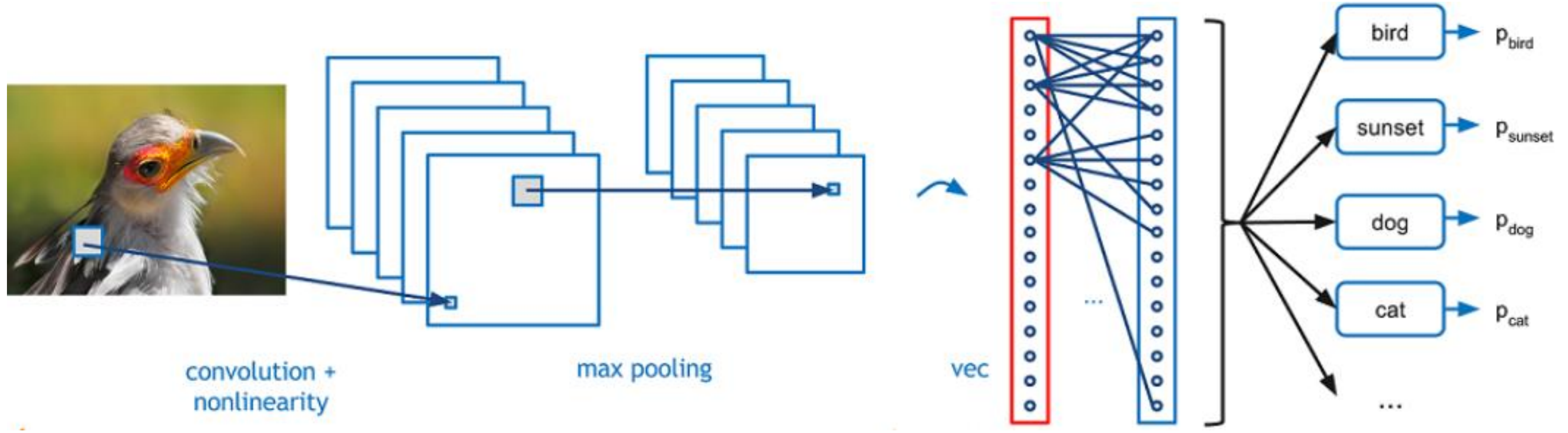
torch.nn.BCEWithLogitsLoss

I. Classification Outlines

B. Multi-Class Classification:

I. Classification Outlines

B. Multi-Class Classification:



Non-linearity

Softmax:

Loss:

Cross Entropy:

PyTorch:

torch.nn.CrossEntropyLoss
torch.nn.logSoftmax+
.NLLloss

I. Classification Outlines

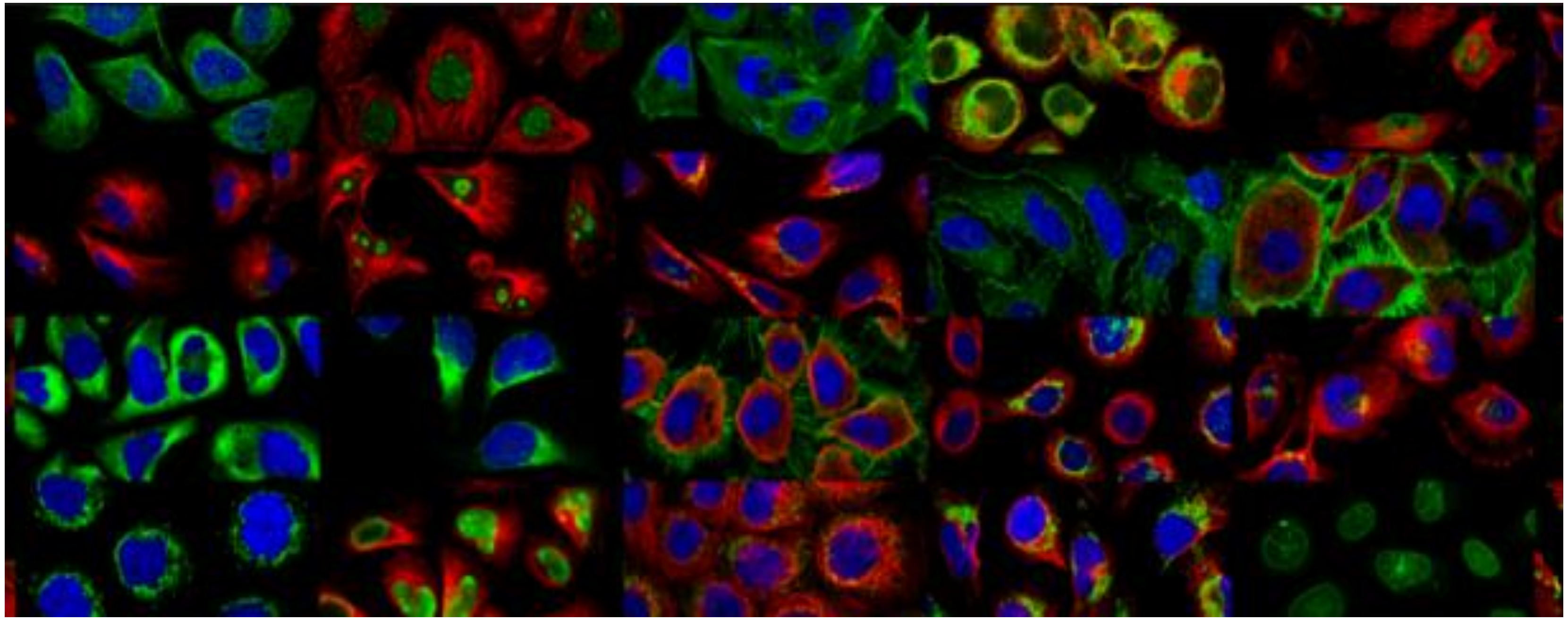
B. Multi-Class Classification:

Softmax:

I. Classification Outlines

C. Multi-Label Classification:

Human Protein Atlas Image Classification

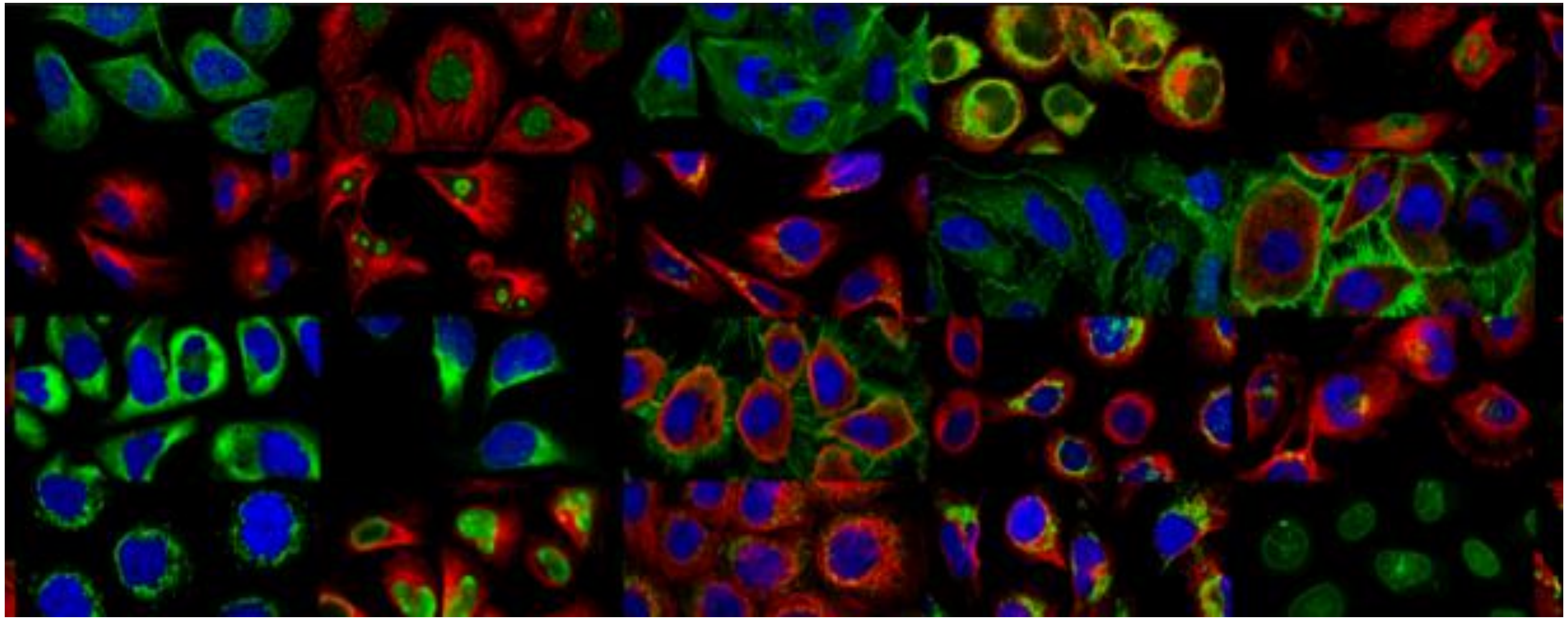


0. Nucleoplasm
1. Nuclear membrane
2. Nucleoli
3. Nucleoli fibrillar center
4. Nuclear speckles
5. Nuclear bodies
6. Endoplasmic reticulum
7. Golgi apparatus
8. Peroxisomes
9. Endosomes
10. Lysosomes
11. Intermediate filaments
12. Actin filaments
13. Focal adhesion sites
14. Microtubules
15. Microtubule ends
16. Cytokinetic bridge
17. Mitotic spindle
18. Microtubule organizing center
19. Centrosome
20. Lipid droplets
21. Plasma membrane
22. Cell junctions
23. Mitochondria
24. Aggresome
25. Cytosol
26. Cytoplasmic bodies
27. Rods & rings

I. Classification Outlines

C. Multi-Label Classification:

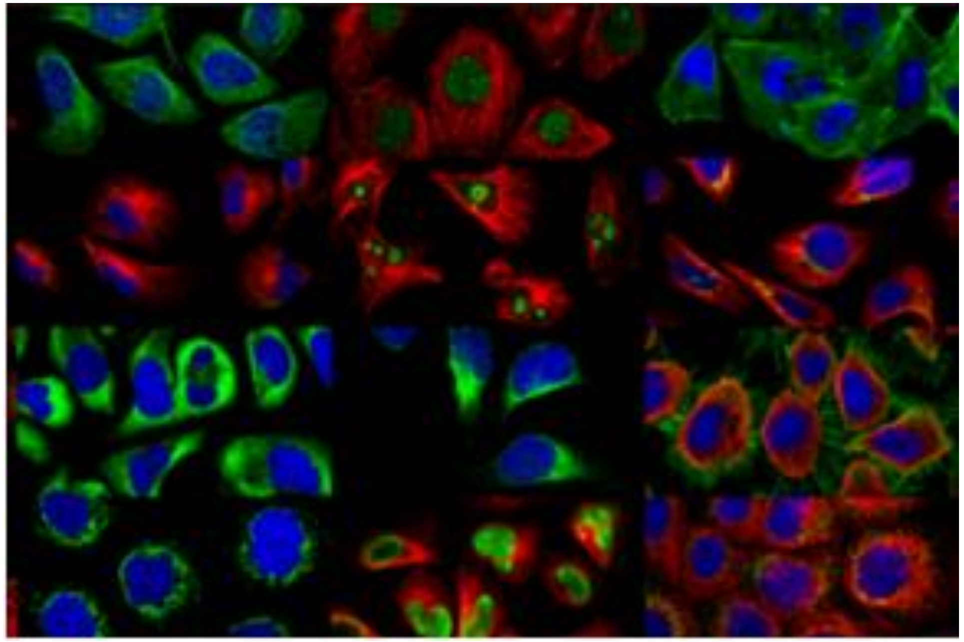
Human Protein Atlas Image Classification



……jpg 0 0 1 0 1 … 1
……jpg 1 0 1 0 0 … 0
……jpg 0 0 1 1 0 … 1
……jpg 1 0 0 0 0 … 0
……jpg 0 0 0 0 1 … 1
……jpg 0 1 1 0 0 … 0
.
.
.
……jpg 0 0 0 1 0 … 1
……jpg 1 1 0 0 0 … 0

I. Classification Outlines

C. Multi-Label Classification:



Non-linearity:

Sigmoid: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

Loss:

Cross Entropy: $H(p,q) = -\sum_i p_i log q_i$

PyTorch:

torch.nn.multiLabelSoftMarginLoss
.multiLabelMarginLoss

I. Classification Outlines

D. Multi-Task Classification:

Gender:

Male / Female / NA

Hat:

Hat / No Hat / NA

Mask:

Mask / No Mask / NA

Glasses:

Glasses / Sunglasses / No Glasses / NA

II. Practical Classification Problem

II. Practical Classification Problem

E. Multi-Class/Label/Task Classification

E1. Multi-Class

[Simple Introduction of Caffe]

II. Practical Classification Problem

E. Multi-Class/Label/Task Classification

E2. Multi-Label Classification

[Implementation of PyTorch]

II. Practical Classification Problem

E. Multi-Class/Label/Task Classification

E3. Multi-Task Classification

[Let’s see the real practical task again]

II. Practical Classification Problem

E. Multi-Class/Label/Task Classification

E3. Multi-Task Classification

Gender:

Male / Female / NA

Hat:

Hat / No Hat / NA

Mask:

Mask / No Mask / NA

Glasses:

Glasses / Sunglasses / No Glasses / NA

II. Practical Classification Problem

E. Multi-Class/Label/Task Classification

E3. Multi-Task Classification

Age:

0-100 or more

Expression:

No Expression / Happy / Sad / Angry / ……

Race:

Asian / Indian / Latino / Black / White / Hispanic

Hair:

Short / Long / Bold / Braid / ……

II. Practical Classification Problem

E. Multi-Class/Label/Task Classification

E3. Multi-Task Classification

Difficulties:

1. Data unbalanced in different tasks
2. Data unbalanced within the same class
3. Data has different task

II. Practical Classification Problem

F. Unbalanced Data

F1. Aspect of data

From paper & practice:

1. **When there's problems of unbalanced data,**
   dueling with data can always gives the best correction

2. **Down sampling & Up sampling [Repeat / Augmentation]**

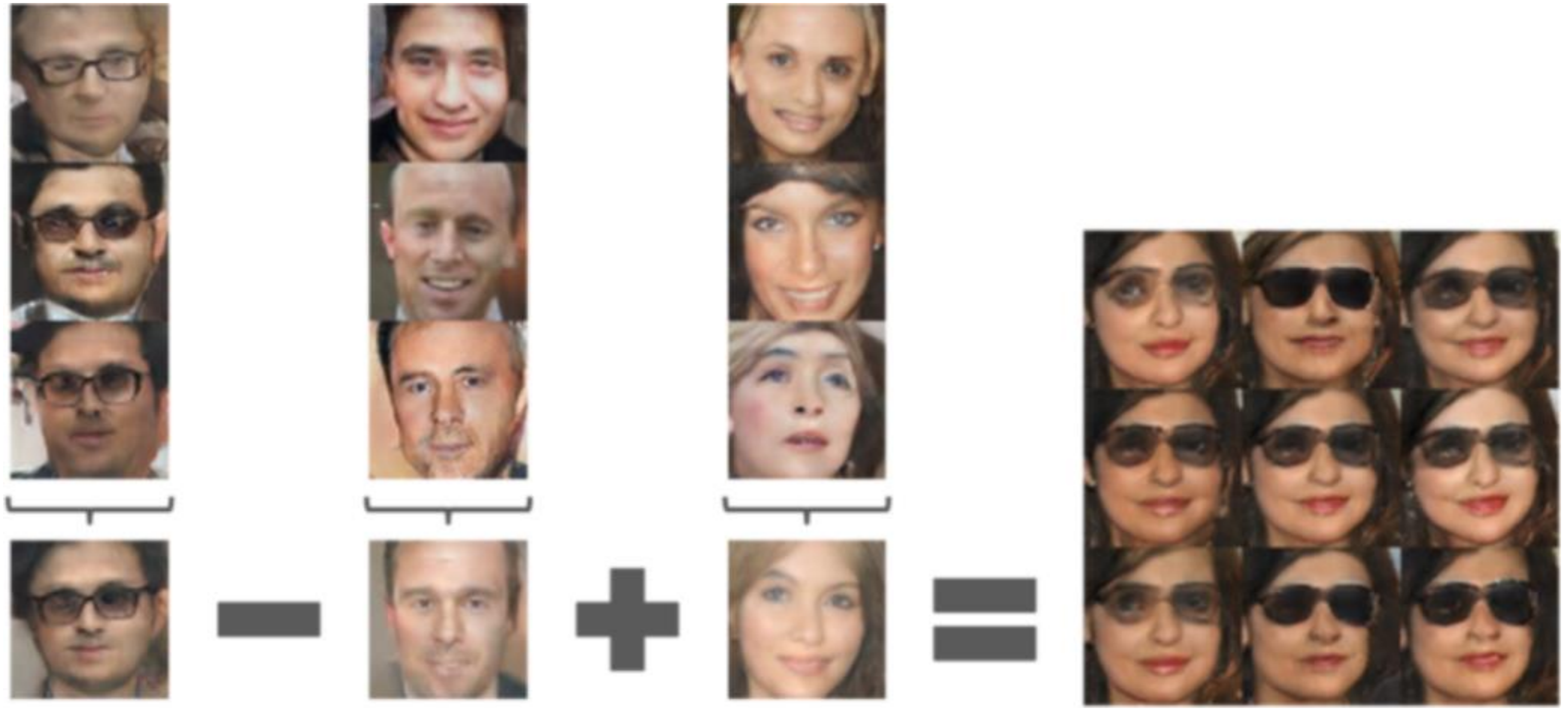
   Rotation / Perspective / Translation / Scale / Noise / Blur / Occlusion / Color / Brightness / …

3. **If possible, GAN could help [from count to style].**

II. Practical Classification Problem

F. Unbalanced Data

F1. Aspect of data



II. Practical Classification Problem

F. Unbalanced Data

F2. Loss

Weighted Cross Entropy Loss

Focal Loss

II. Practical Classification Problem

F. Unbalanced Data

F3. Learning Strategy

Backbone + Branches

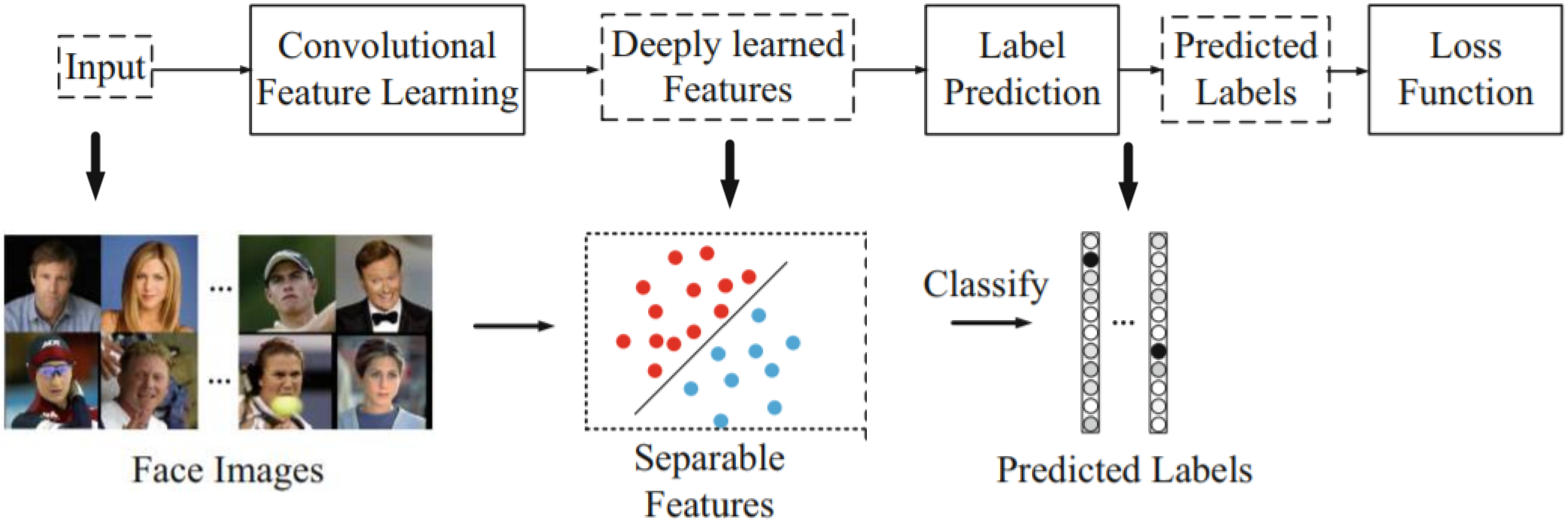
[Let’s see the practical procedure]

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

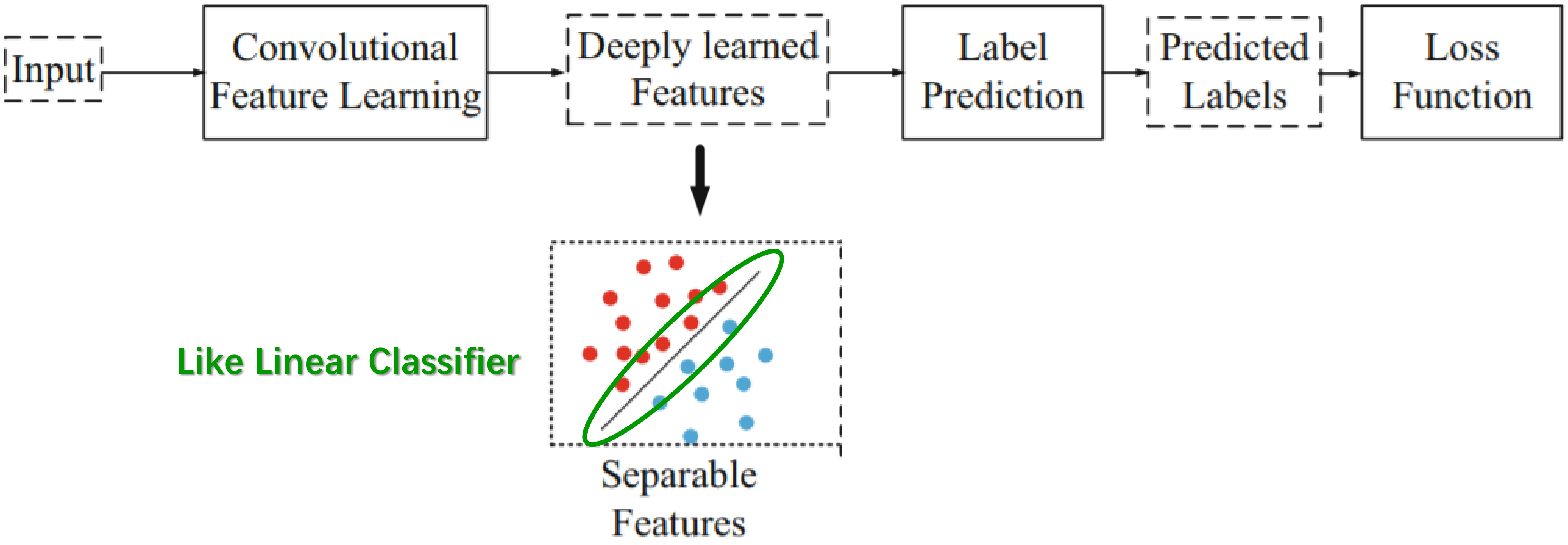


II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

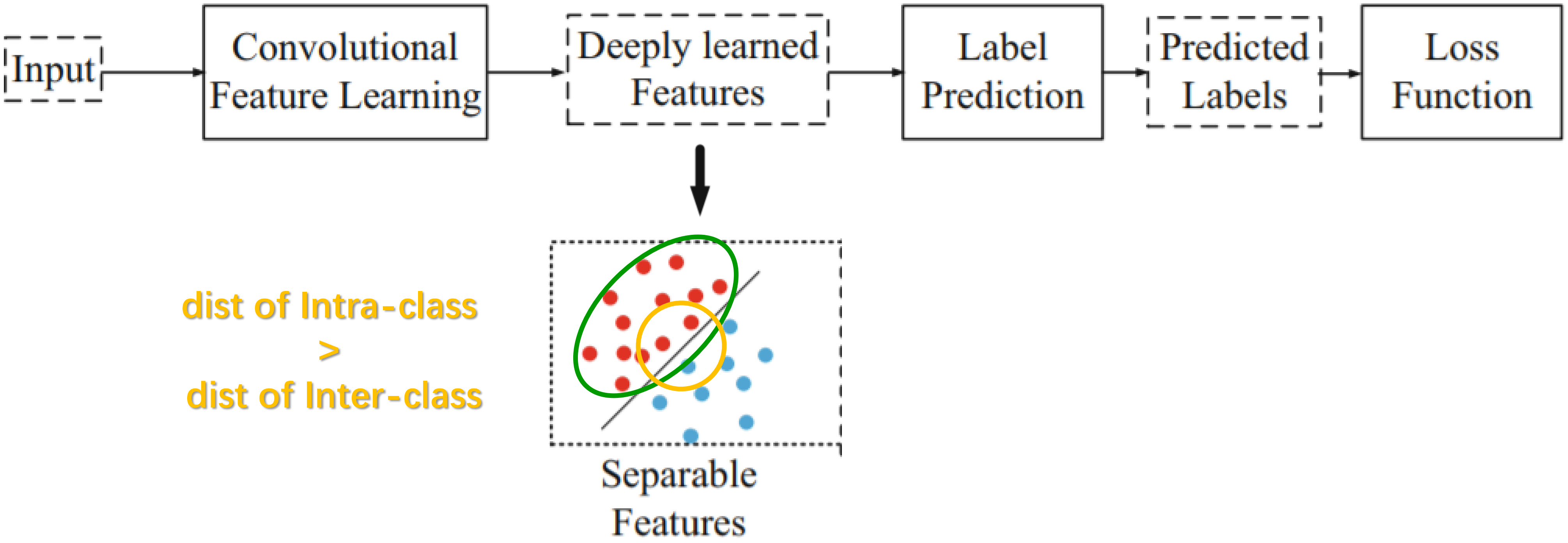


II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

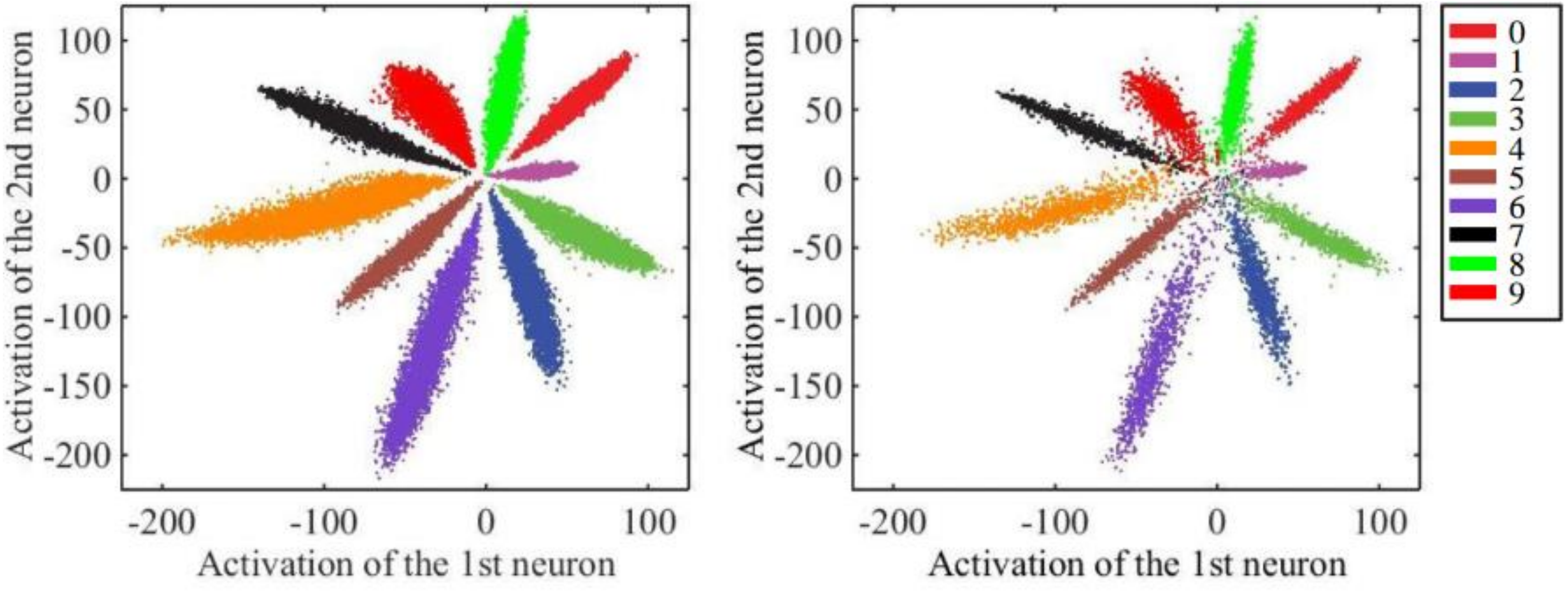


II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:



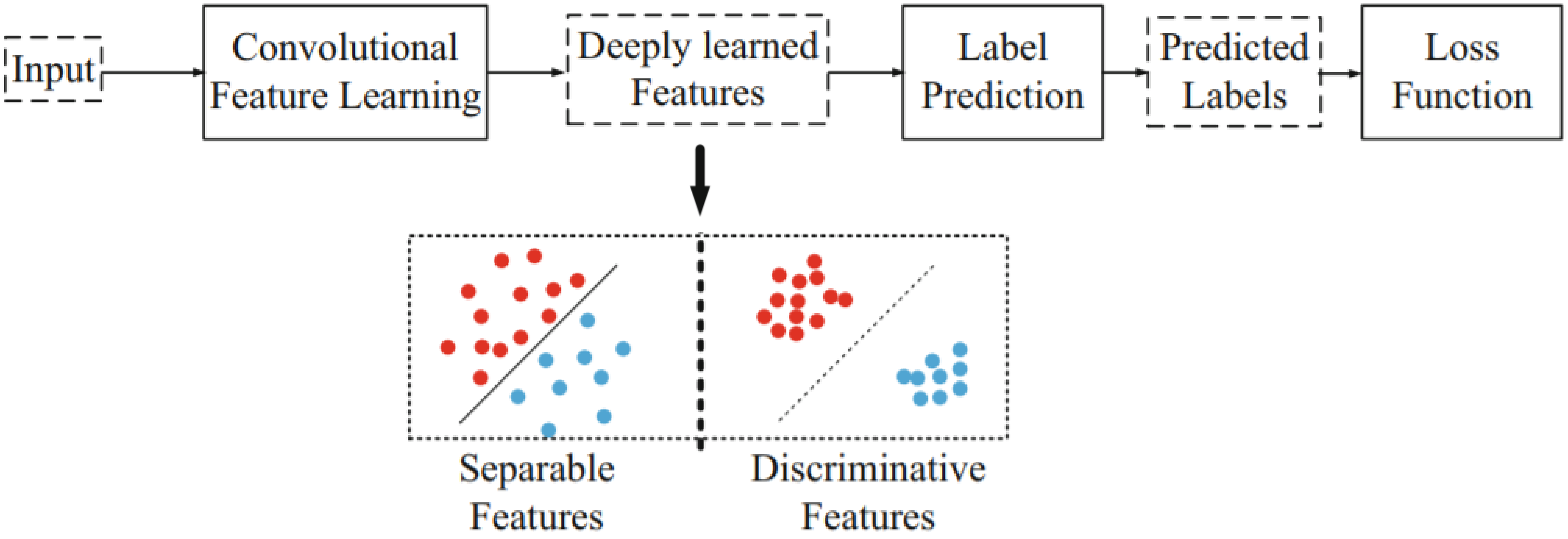
(a) (b)

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

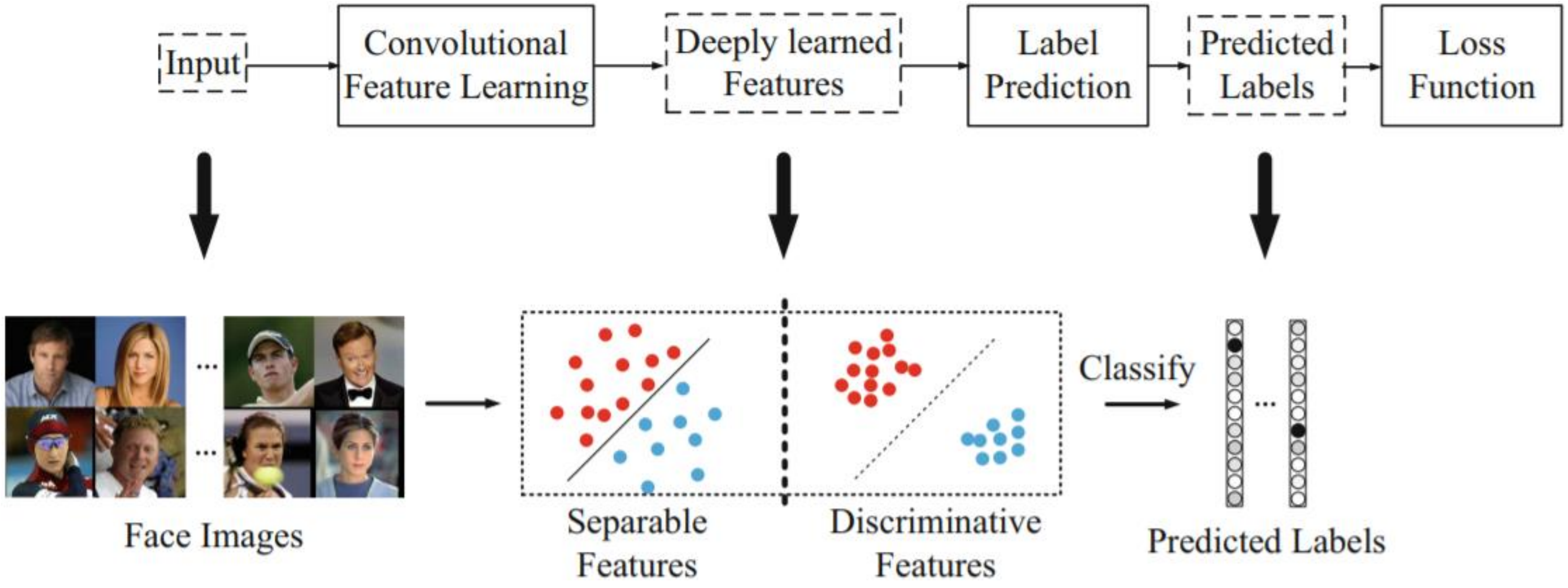


II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:



II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:

Algorithm 1. The discriminative feature learning algorithm

Input: Training data $\{\boldsymbol{x}_i\}$. Initialized parameters θ_C in convolution layers. Parameters W and $\{\mathbf{c}_j | j = 1, 2, ..., n\}$ in loss layers, respectively. Hyperparameter λ, α and learning rate μ^t. The number of iteration $t \leftarrow 0$.

Output: The parameters θ_C.

1: **while** not converge **do**

2: $t \leftarrow t + 1$.

3: Compute the joint loss by $\mathcal{L}^t = \mathcal{L}^t_S + \mathcal{L}^t_C$.

4: Compute the backpropagation error $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t}$ for each i by $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} = \frac{\partial \mathcal{L}^t_S}{\partial \boldsymbol{x}_i^t} + \lambda \cdot \frac{\partial \mathcal{L}^t_C}{\partial \boldsymbol{x}_i^t}$.

5: Update the parameters W by $W^{t+1} = W^t - \mu^t \cdot \frac{\partial \mathcal{L}^t}{\partial W^t} = W^t - \mu^t \cdot \frac{\partial \mathcal{L}^t_S}{\partial W^t}$.

6: Update the parameters $\mathbf{c}_j$ for each j by $\mathbf{c}_j^{t+1} = \mathbf{c}_j^t - \alpha \cdot \Delta \mathbf{c}_j^t$.

7: Update the parameters θ_C by $\theta_C^{t+1} = \theta_C^t - \mu^t \sum_i^m \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} \cdot \frac{\partial \boldsymbol{x}_i^t}{\partial \theta_C^t}$.

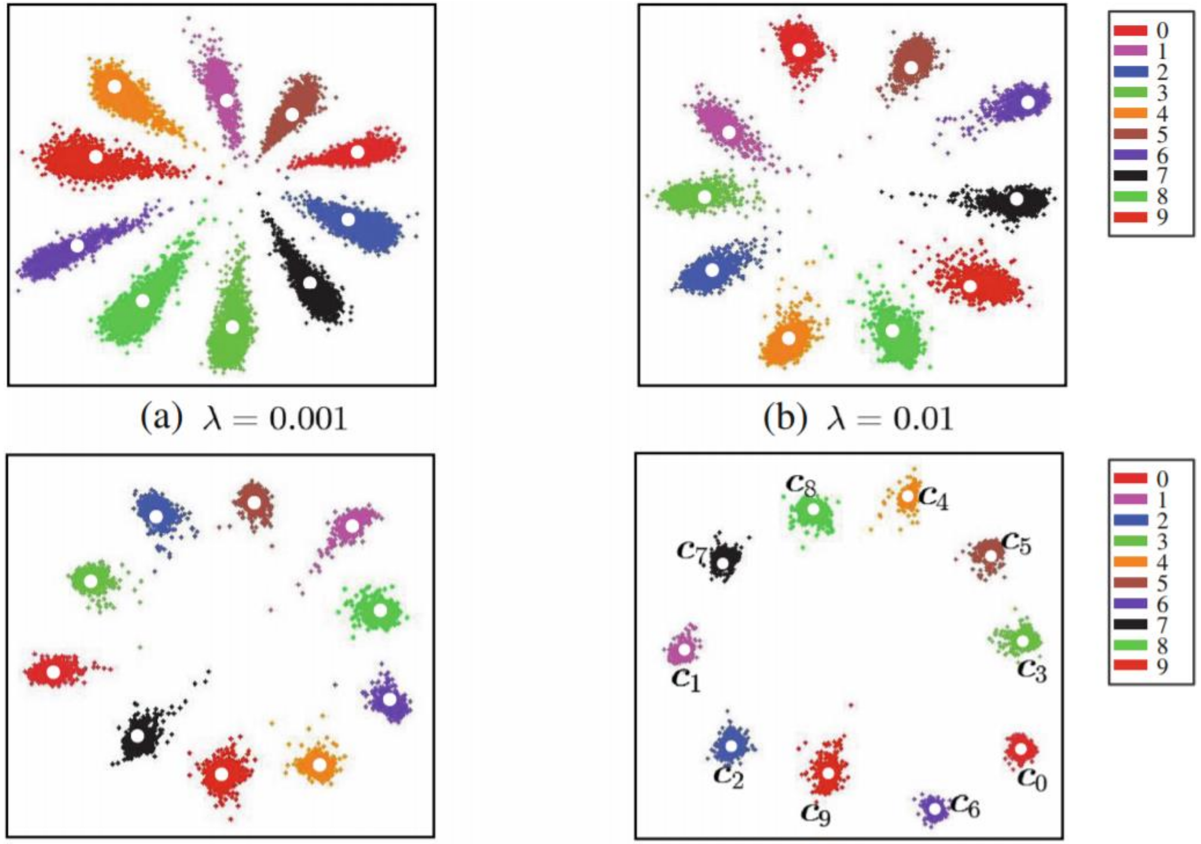
8: **end while**

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Center Loss:



(a) $\lambda = 0.001$ (b) $\lambda = 0.01$

(c) $\lambda = 0.1$ (d) $\lambda = 1$

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Other Losses:

Triplet Loss / Contrastive Loss

II. Practical Classification Problem

G. Fine-Grained Classification

G1. Discriminative Feature

Other Losses:

Triplet Loss / Contrastive Loss

Tips:

1. Usually, they are also good for unbalanced data

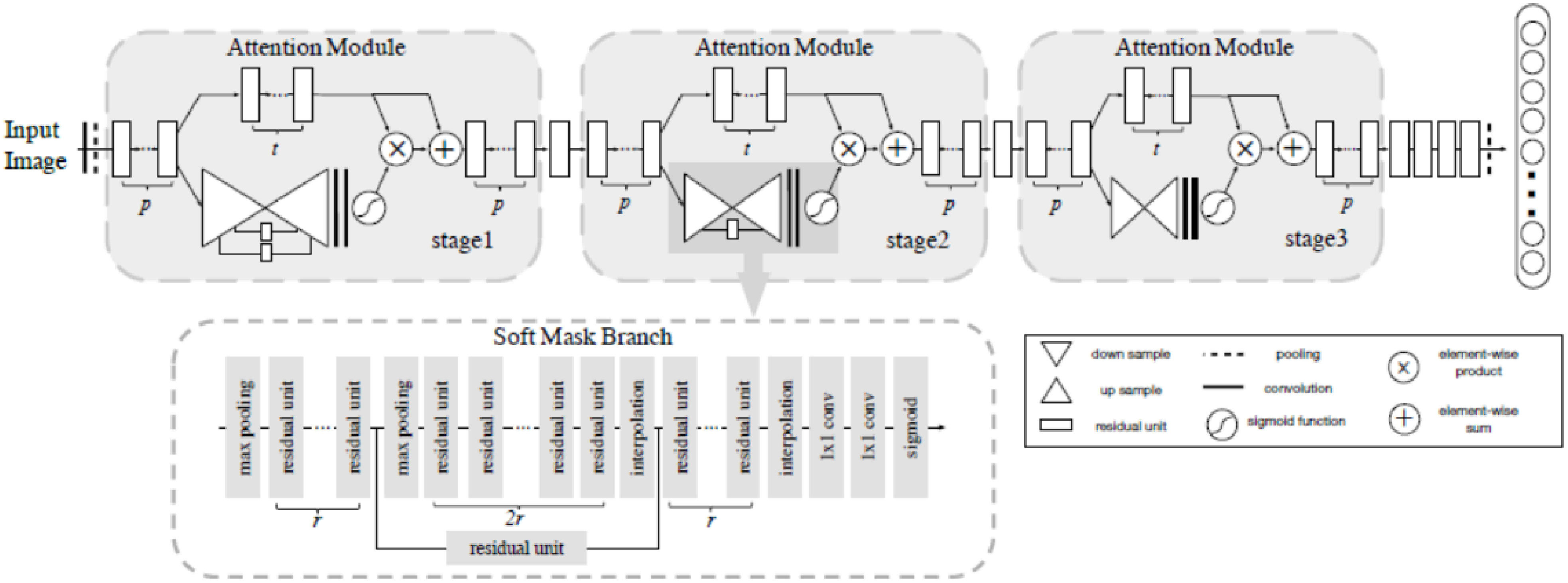
2. Empirically, they are critical for face-related stuff like face recognition……

II. Practical Classification Problem

G. Fine-Grained Classification

G2. Attention Mechanism

Residual Attention Network:



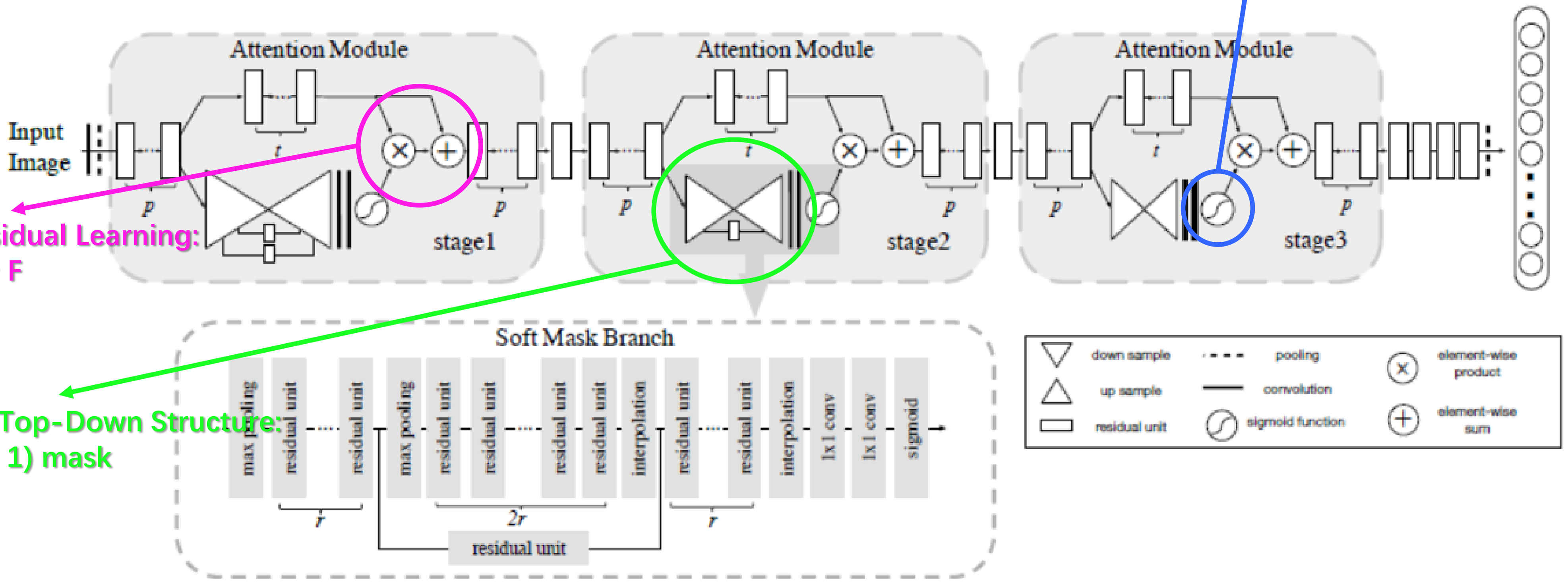
II. Practical Classification Problem

G. Fine-Grained Classification

G2. Attention Mechanism

Residual Attention Network:

Sigmoid:
Generate (0, 1) mask



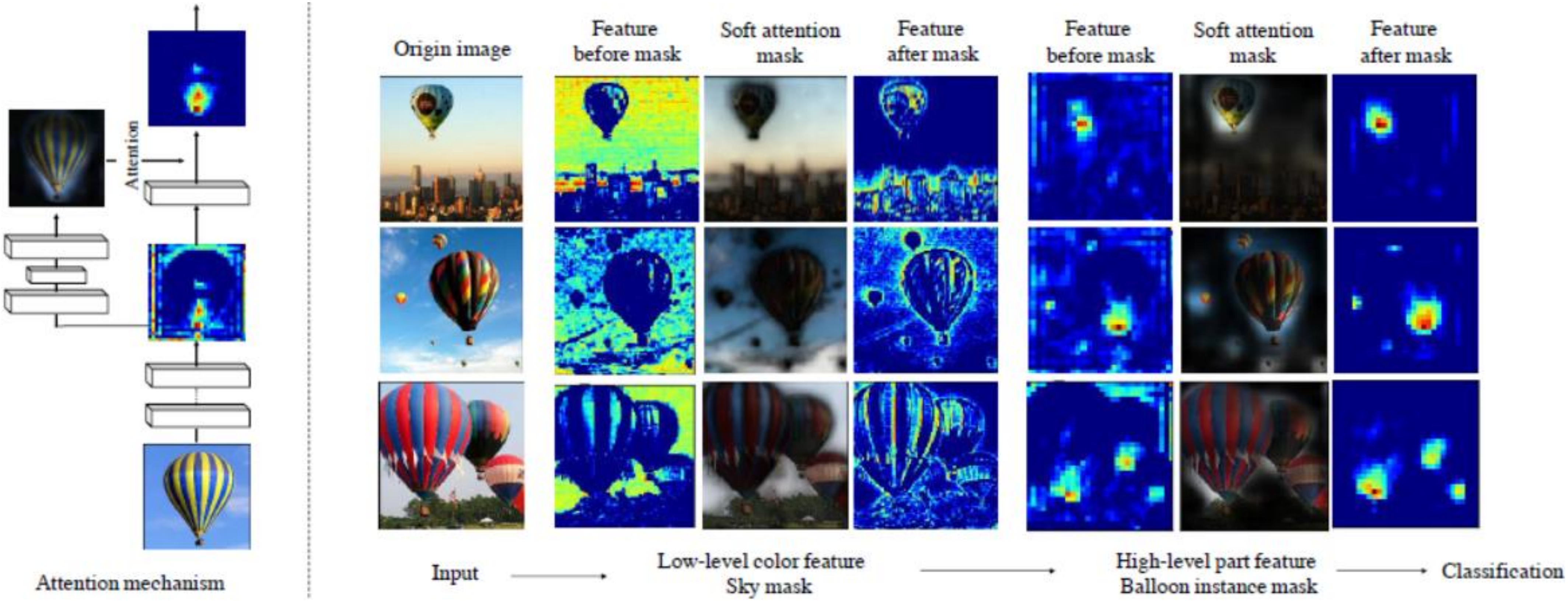
Attention Residual Learning:
H = (1 + M) * F

Bottom-Up Top-Down Structure:
Generate (0, 1) mask

II. Practical Classification Problem

G. Fine-Grained Classification

G2. Attention Mechanism



Projects:

[iWildCam 2019]

[Human Protein Atlas Image Classification]

[Human Face Attribute Recognition]