# Deep Mixture of Diverse Experts for Large-Scale Visual Recognition

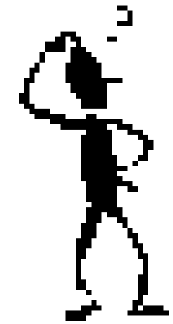
**Jianping Fan**
**Dept. of Computer Science**
**UNC-Charlotte, NC, USA**

# Outlines of Presentation

- Research Motivation

Large-scale visual recognition

- Deep Mixture of Diverse Experts
- Ontology-driven Task Group Generation
- Deep Multi-Task Learning, Deep Boosting & Deep Collaborative Learning
- Conclusions & Future Work



# 1. Research Motivation

- **Many applications rely on large-scale visual recognition**

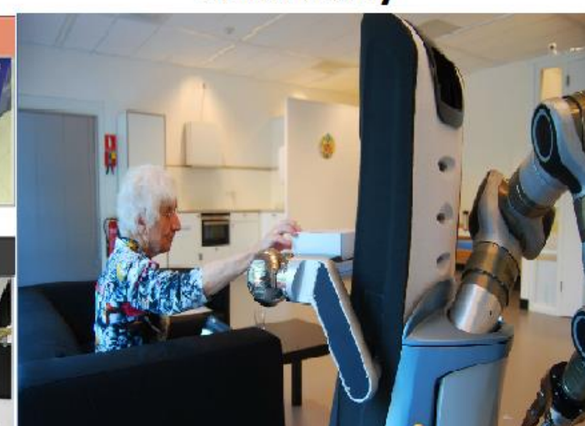
car damage assessment

face recognition


smart city

document electronify


surveillance

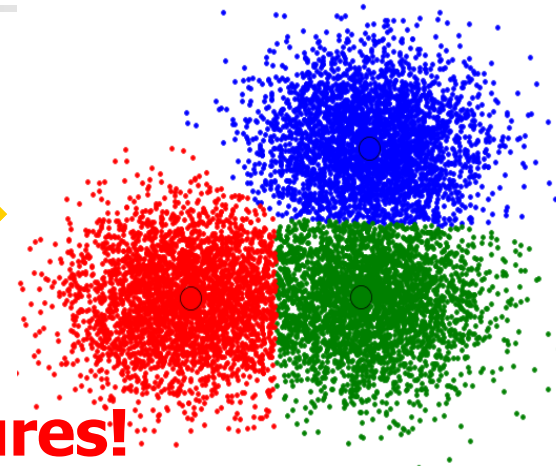


elder care

# Traditional Solutions

**Separate processes for feature learning & classifier training**

**Feature Extraction**



**Hand-crafted features!**

**Classifier Training**

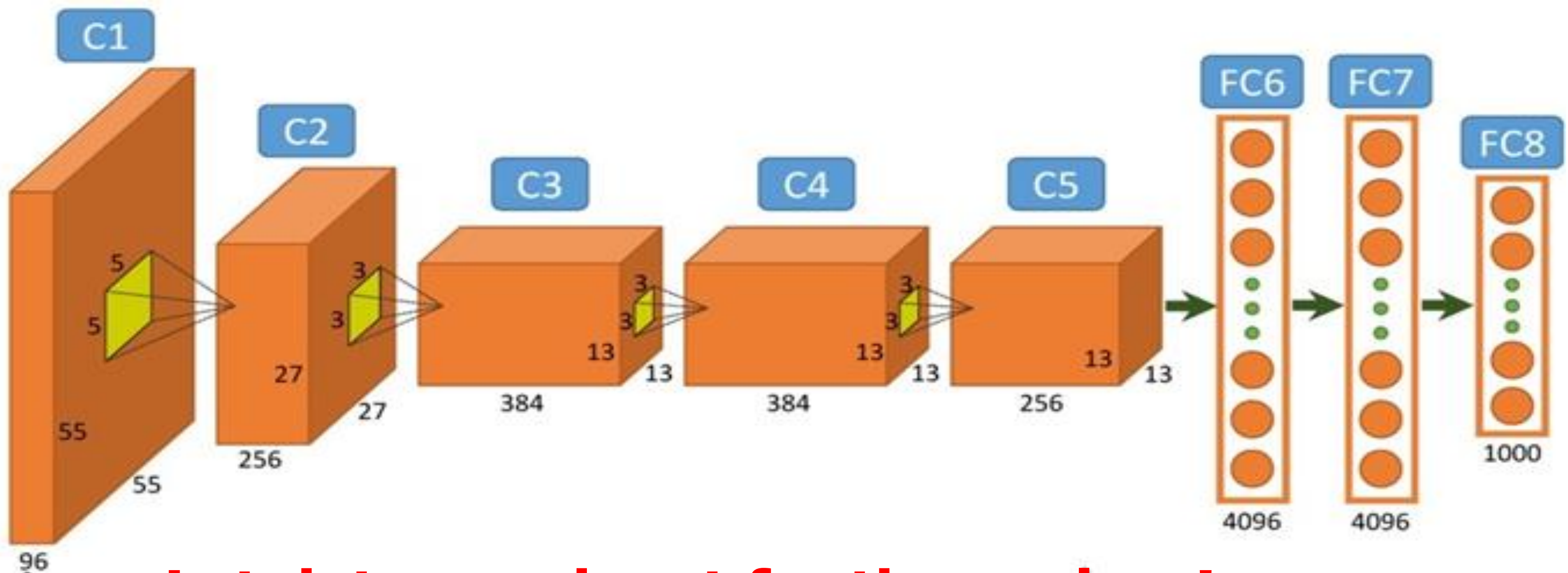
**Feature Extraction**

**Classifiers**

**Prediction**

# Deep Learning Approach

**Joint process for feature learning & classifier training**




**Let data speak out for themselves!**

**SGD for back-propagation**

# Flat Softmax

- **Problems of Flat Softmax**
  - **Inter-class visual correlations are completely ignored!**
  - **Differences on their learning complexities are completely ignored!**
  - **Back-propagation may pay more attentions on hard object classes but it may easily achieve higher accuracy rates on easy object classes!**

# Traditional Deep Networks

- Problems of Traditional Deep Networks

Network structures
& node weights

- They are optimized for recognizing 1,000 object classes or less than 1,000 classes!
- They train one joint network for the hard and easy object classes even they have significantly different learning complexities!

Plant Species: more than 200K
Taobao Product Categories: over 100K
ImageNet10K: over 10K

# Traditional Deep Networks

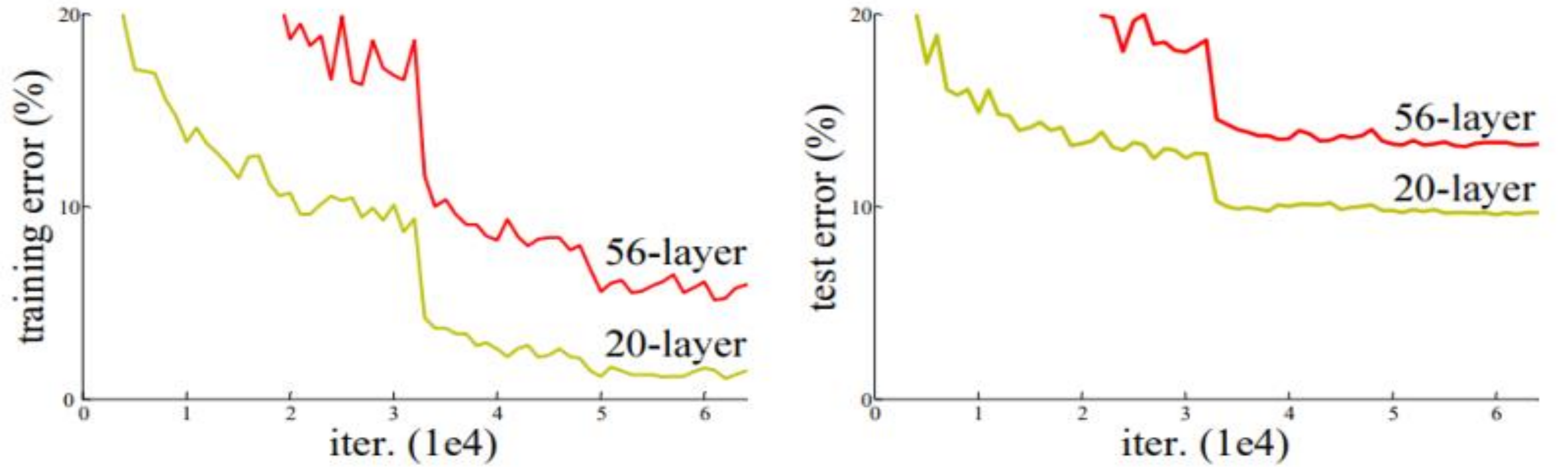
**Some well-designed traditional deep networks**

**Recognizing 10,000 categories or even larger**

## How to Configure Huge Deep Networks?

- **Potential Solutions for Network Extension**

  - **More layers & more units on each layer, …..**
    **---too expensive for trials & errors, & may not be doable for most academic researchers**



# How to Configure Huge Deep Networks?

- **Potential Solutions for Network Extension**

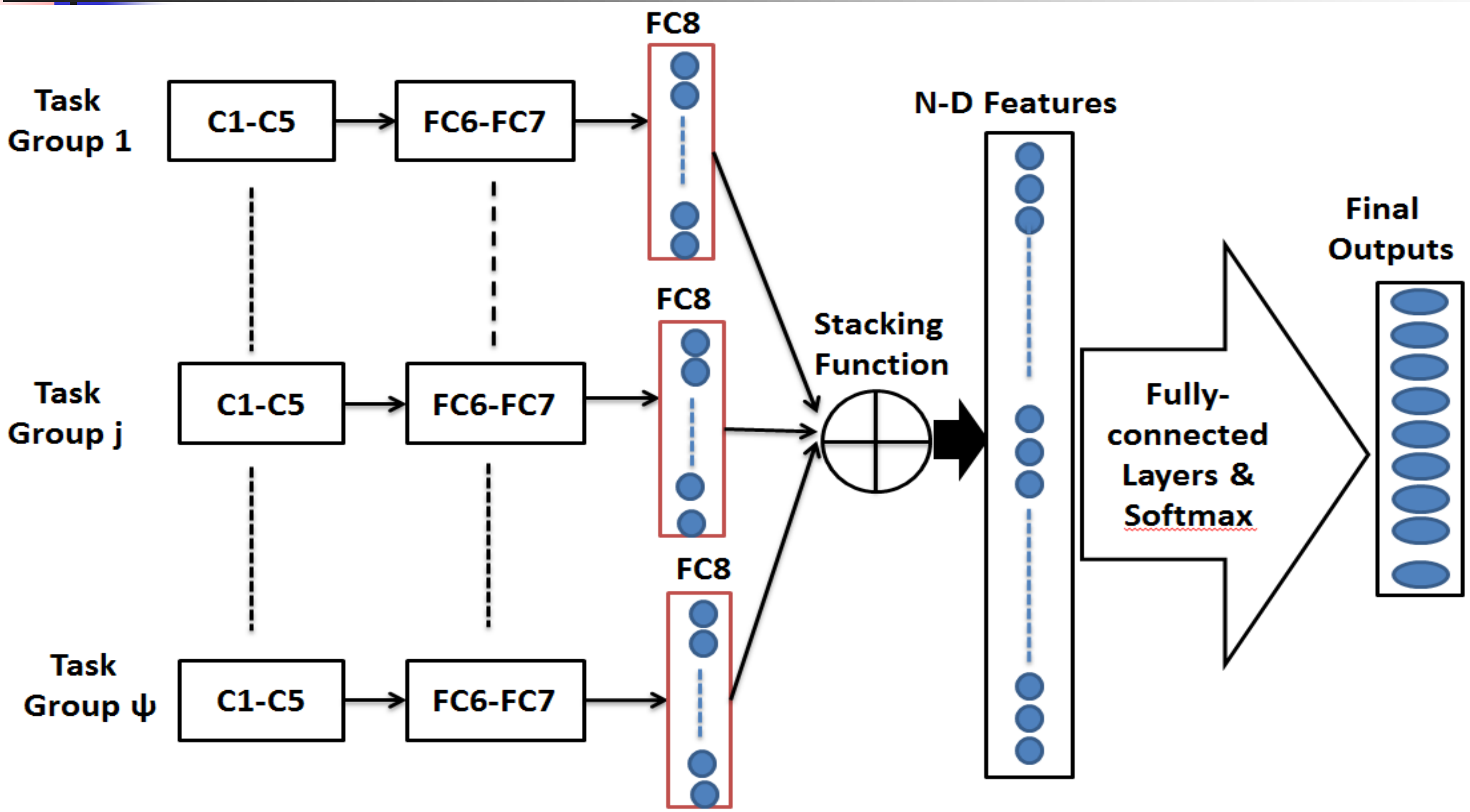
  - **Deep mixture?**

    ---they require same task space

  - **Transfer learning?**

    ---they can transfer the common knowledge from larger task space to smaller ones

# Simple Mixture & Random Task Group Generation




**Problems:**

- **Some bench NBA players may become MVP in CBA**
- **L. James vs. M. Jordan**

# Simple Mixture & Random Task Group Generation

- **Global Optimum**

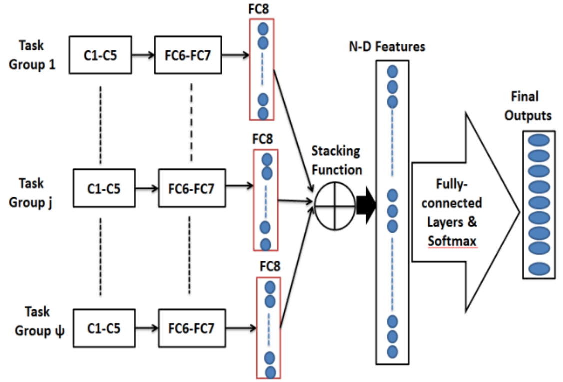
  --- The gradients of objective function are not uniform for all the classes in the same task group!

- **Task Assignment for Group Generation**

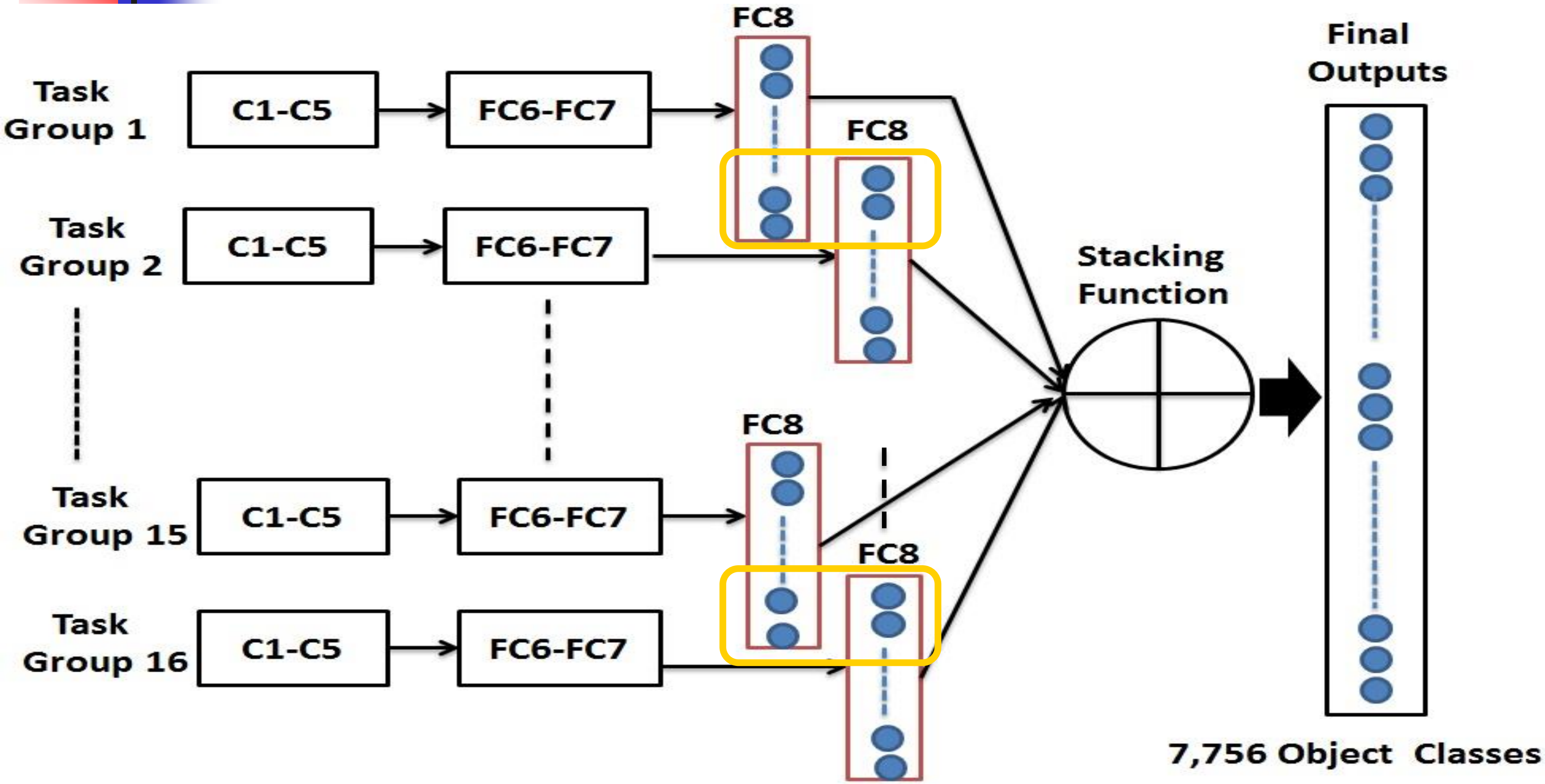
  ---Different classes may have significant differences on their learning complexities!

- **Prediction Comparability**

  ---L. James *vs.* M. Jordan



# Deep Mixture of Diverse Experts



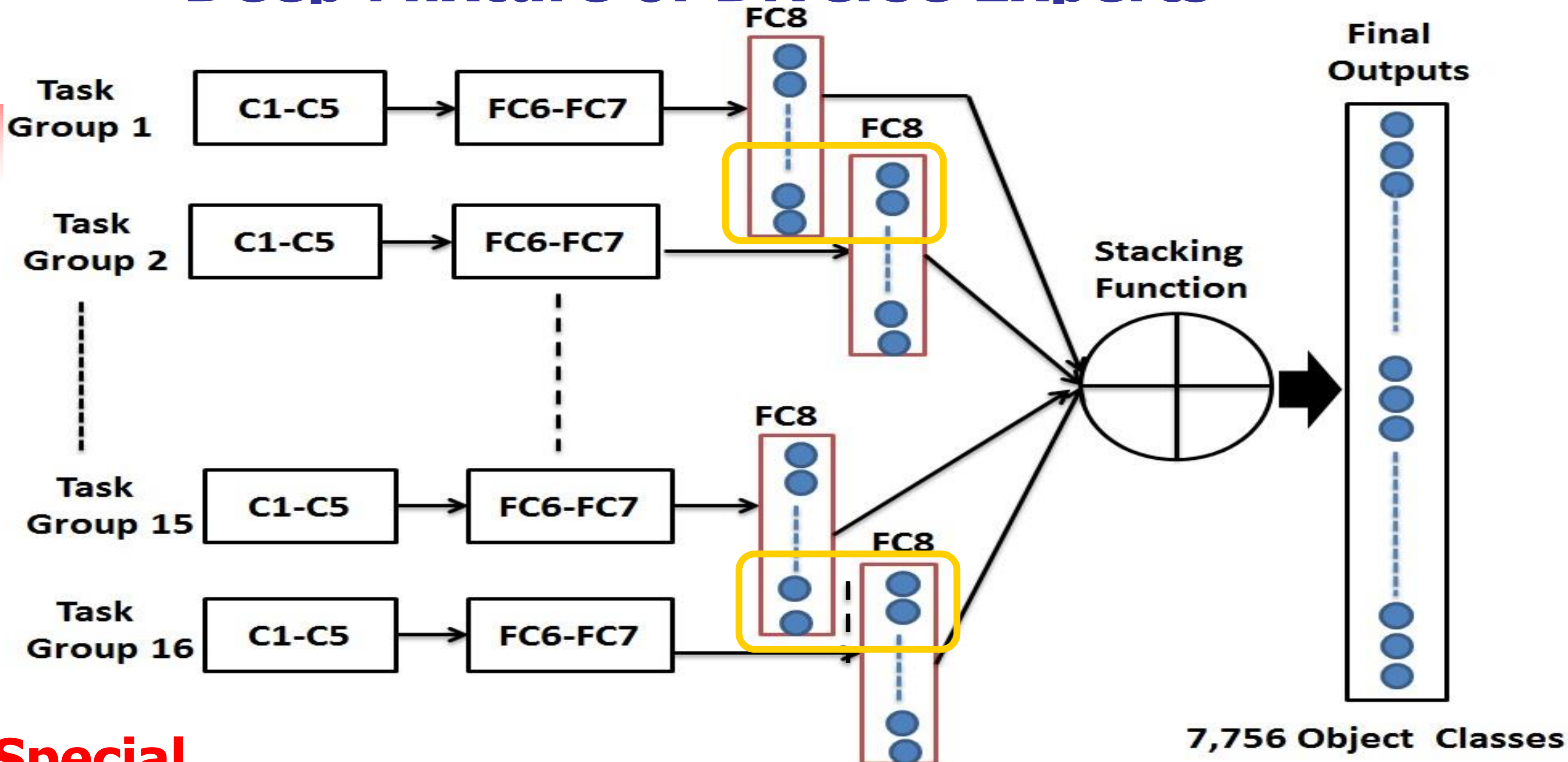
# Deep Mixture of Diverse Experts

- **Potential Problems:**
  - **They are not learned jointly, thus their predictions may not be comparable directly!**
  - **For each test sample, all these base deep CNNs will provide their individual predictions, which one is more believable?**
  - **Object classes may have different learning complexities!**

# Deep Mixture of Diverse Experts

- **Wish List:**
  - The object classes with similar learning complexities are assigned into the same task group!
  - The predictions from multiple base deep CNNs could be comparable at certain level!
  - The prediction conflicts among different groups are predictable or identifiable!

# Deep Mixture of Diverse Experts




**Special Design**

- Object classes with similar learning complexities are assigned into the same task group;
- Task overlapping to enable inter-group communication;
- Special class of ``not-in-group" to enhance comparability;
- ------

# Deep Mixture of Diverse Experts

10,000 object classes

Ontology-driven task group generation

Task Groups with certain overlapping

Each Task Group with 1000 classes

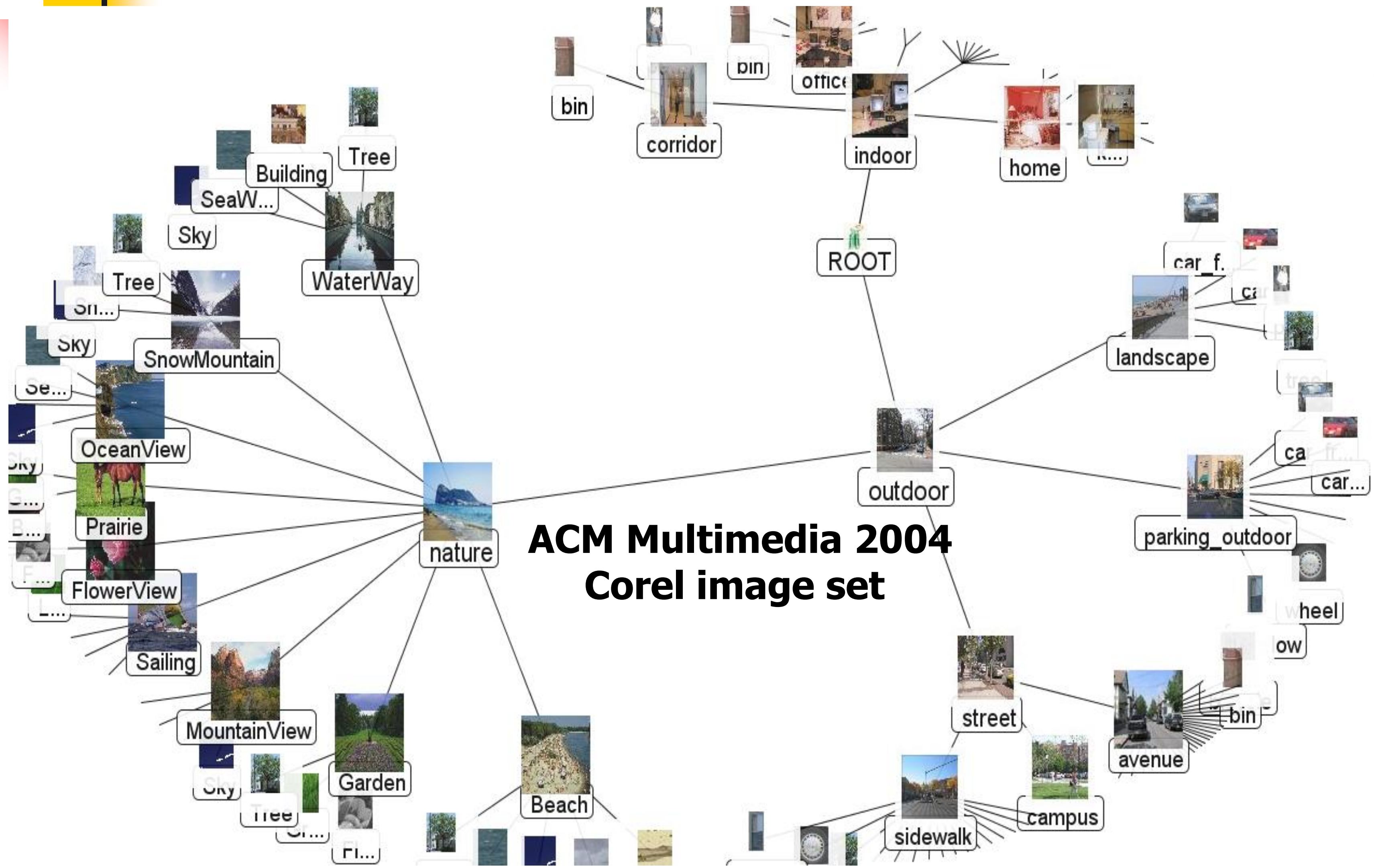
a. Deep multi-task learning for base deep CNNs training
b. Deep Boosting for base deep CNNs Training
c. Hierarchical learning for base deep CNNs training
d. Deep Collaborative Learning
e. Knowledge Distillation from Big Brother or Teacher

Outputs for all task groups

a. Stacking function
b. Gating network
c. Hierarchical mixture

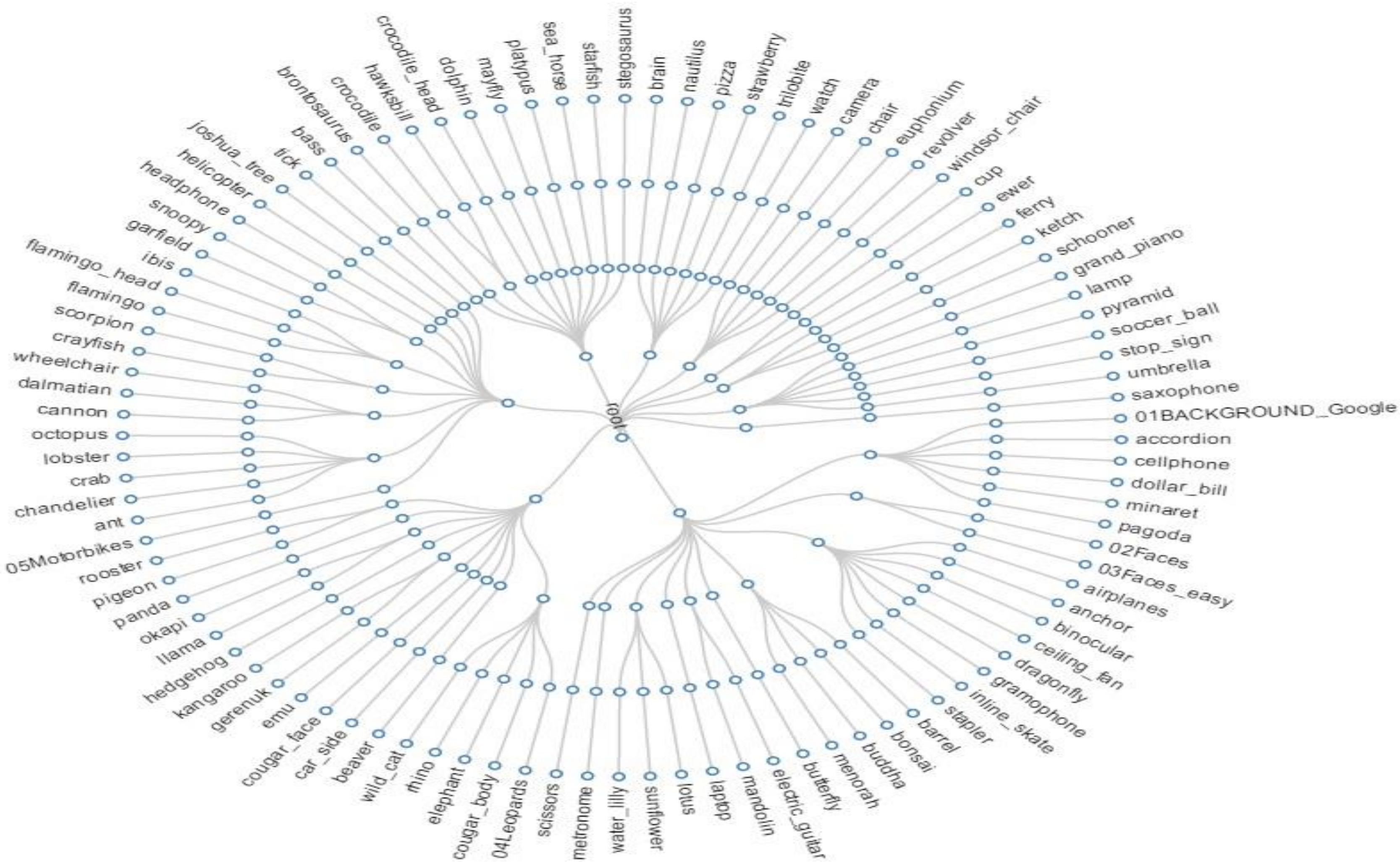
Mixture network with larger outputs

# 2. Ontology for Task Group Generation



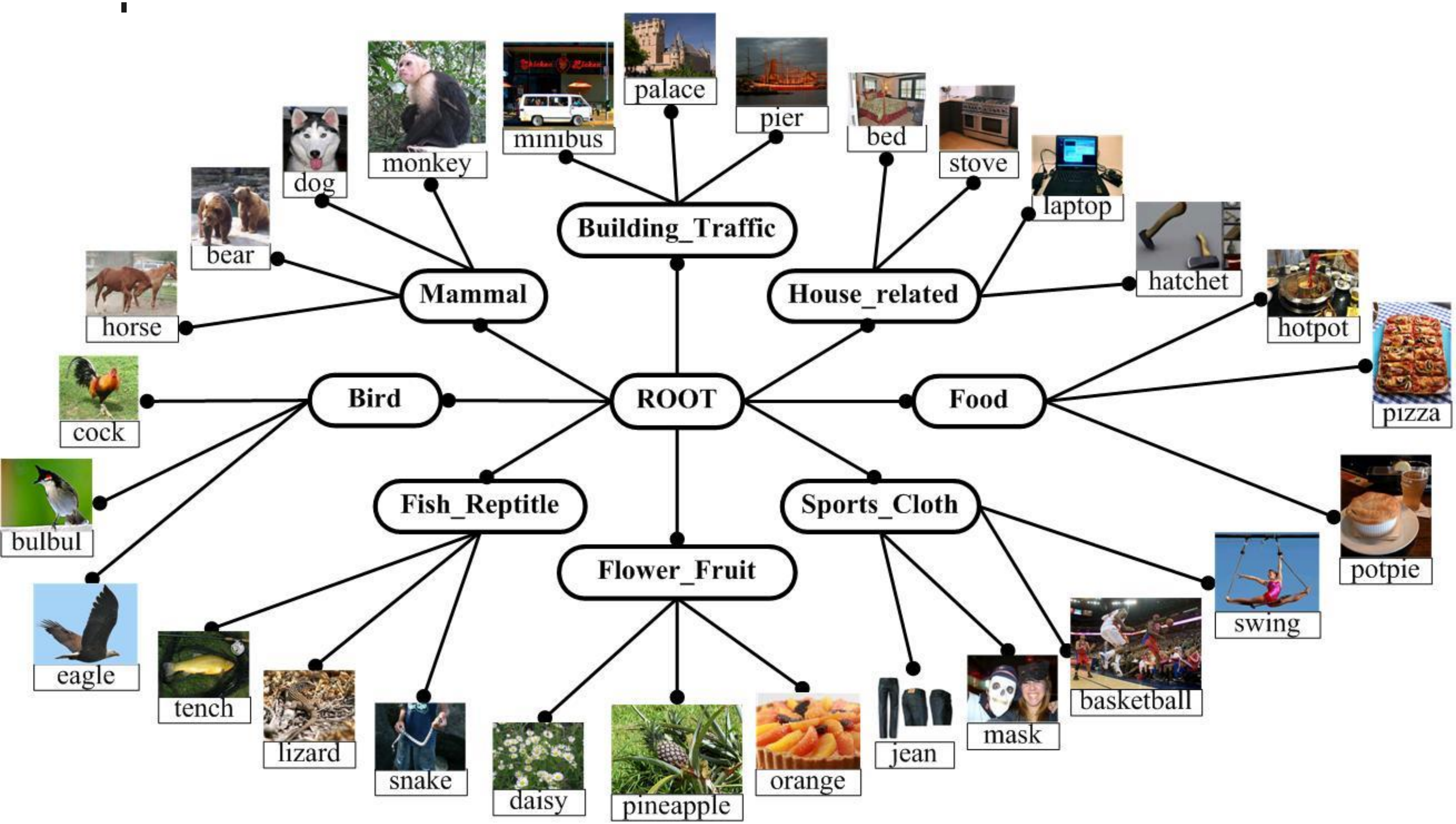

**ACM Multimedia 2004 Corel image set**

# 2. Ontology for Task Group Generation



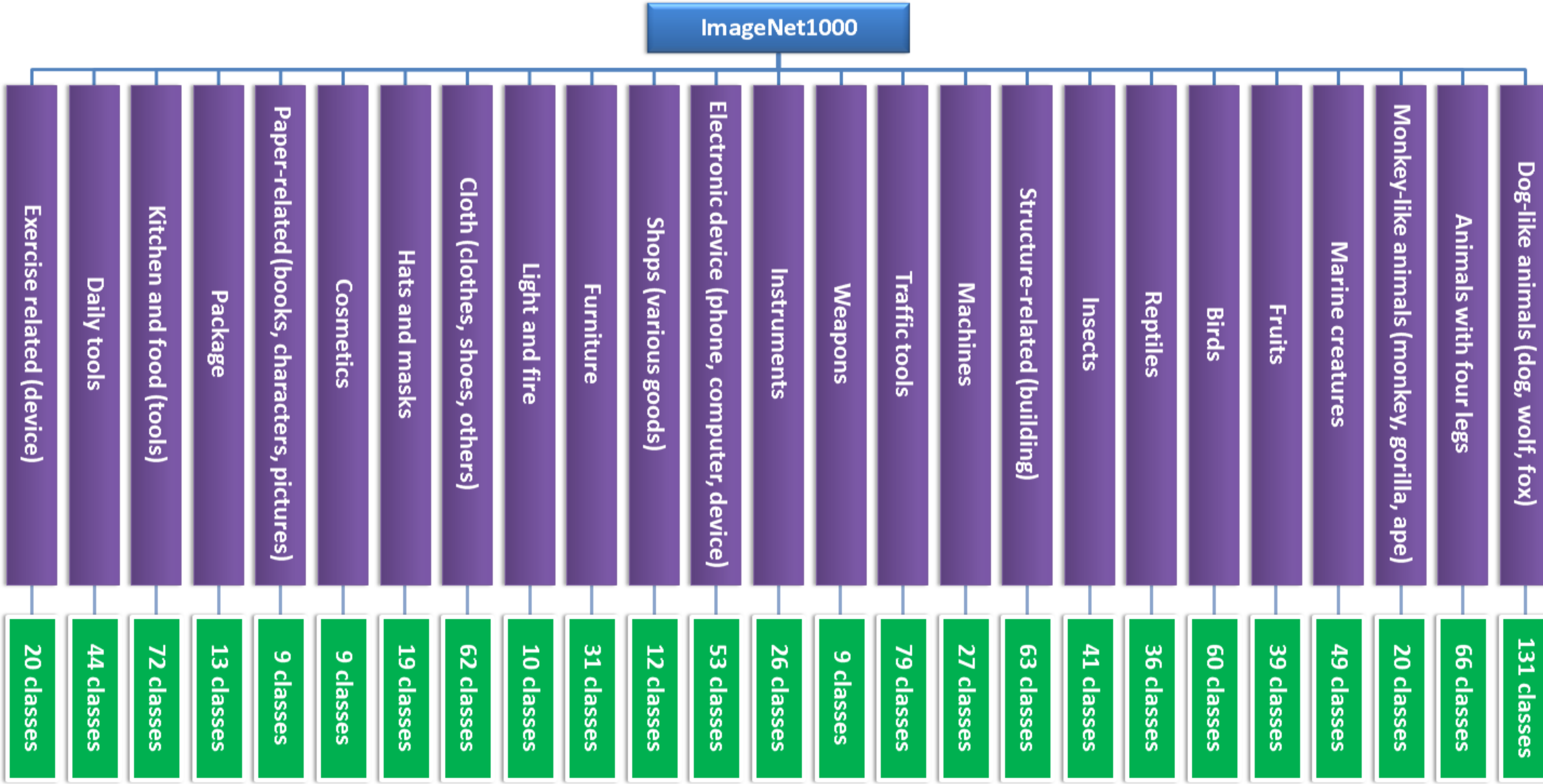

**CalTech101**

# 2. Ontology for Task Group Generation

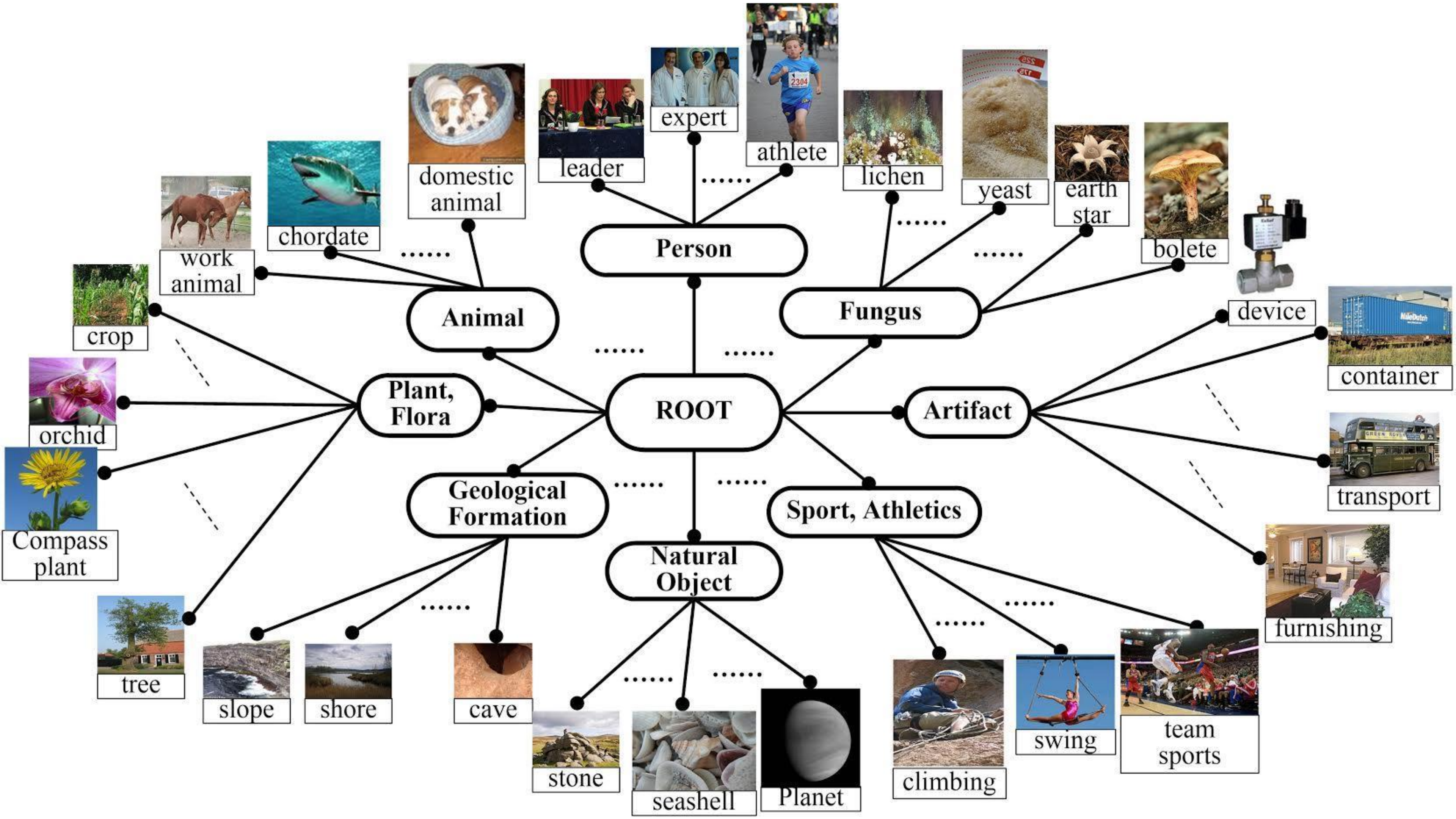



**Two-Layer Ontology for ImageNet1K**

# 2. Ontology for Task Group Generation

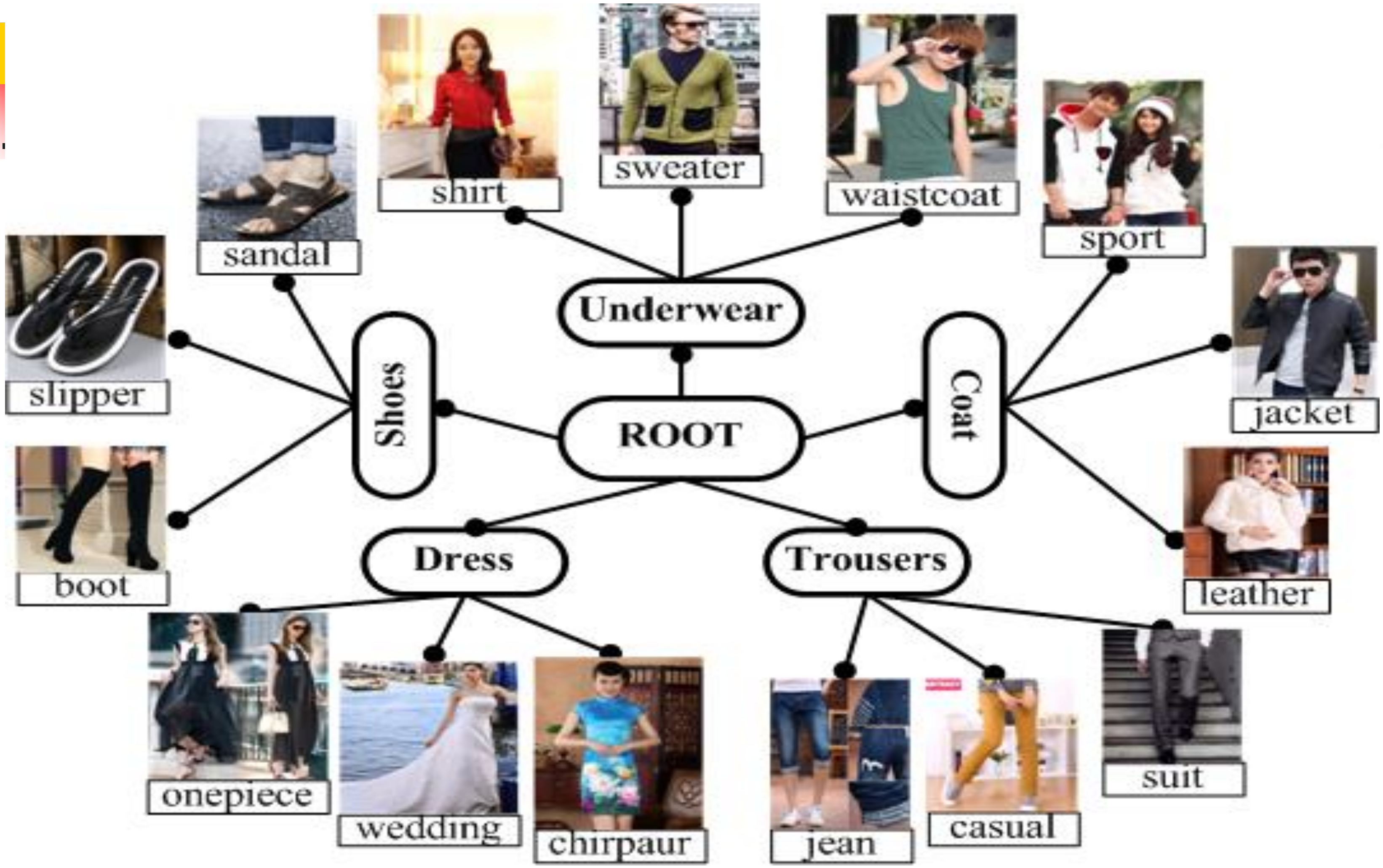


# 2. Ontology for Task Group Generation



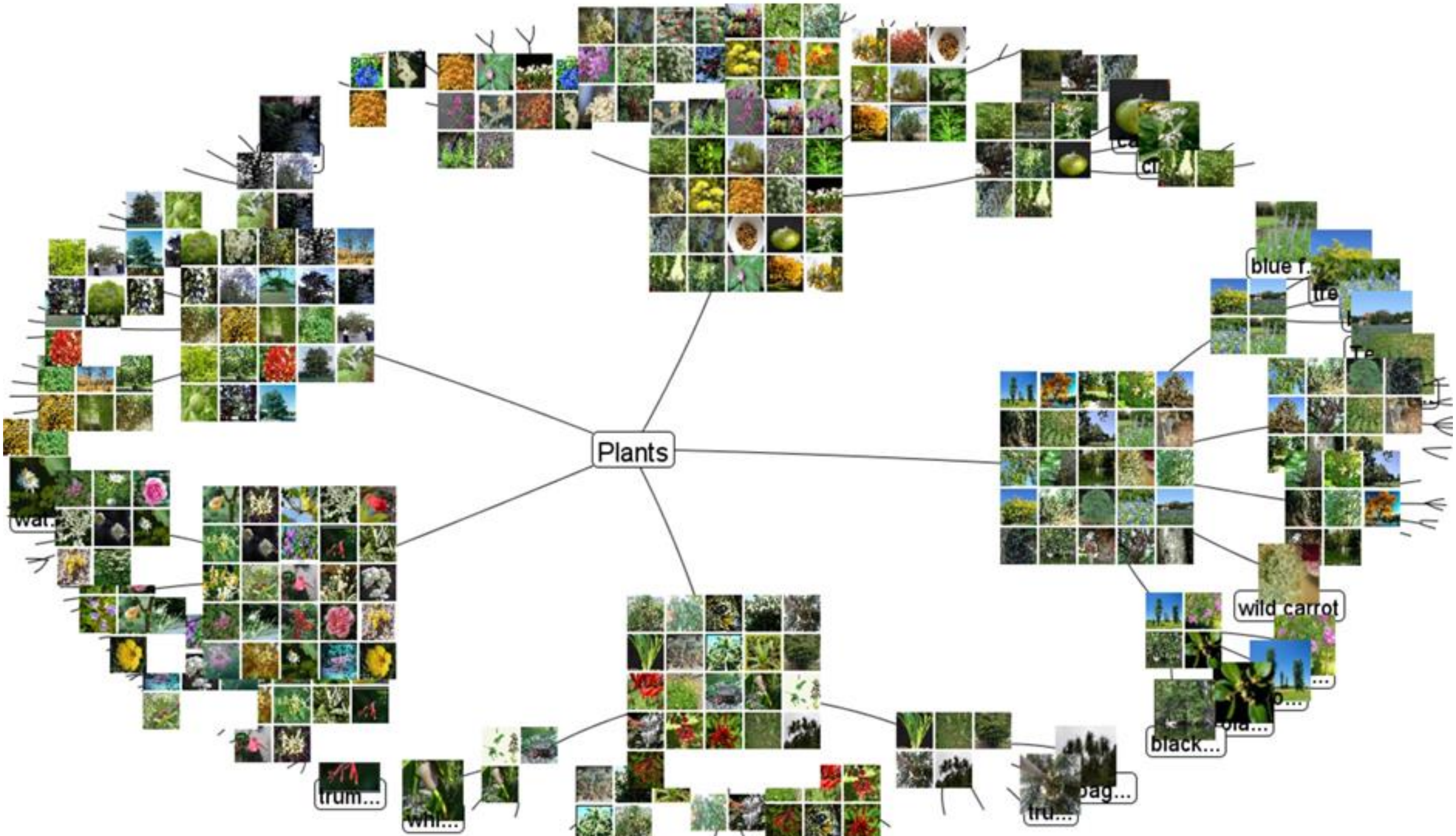

**Two-Layer Ontology for ImageNet10K**

## 2. Ontology for Task Group Generation



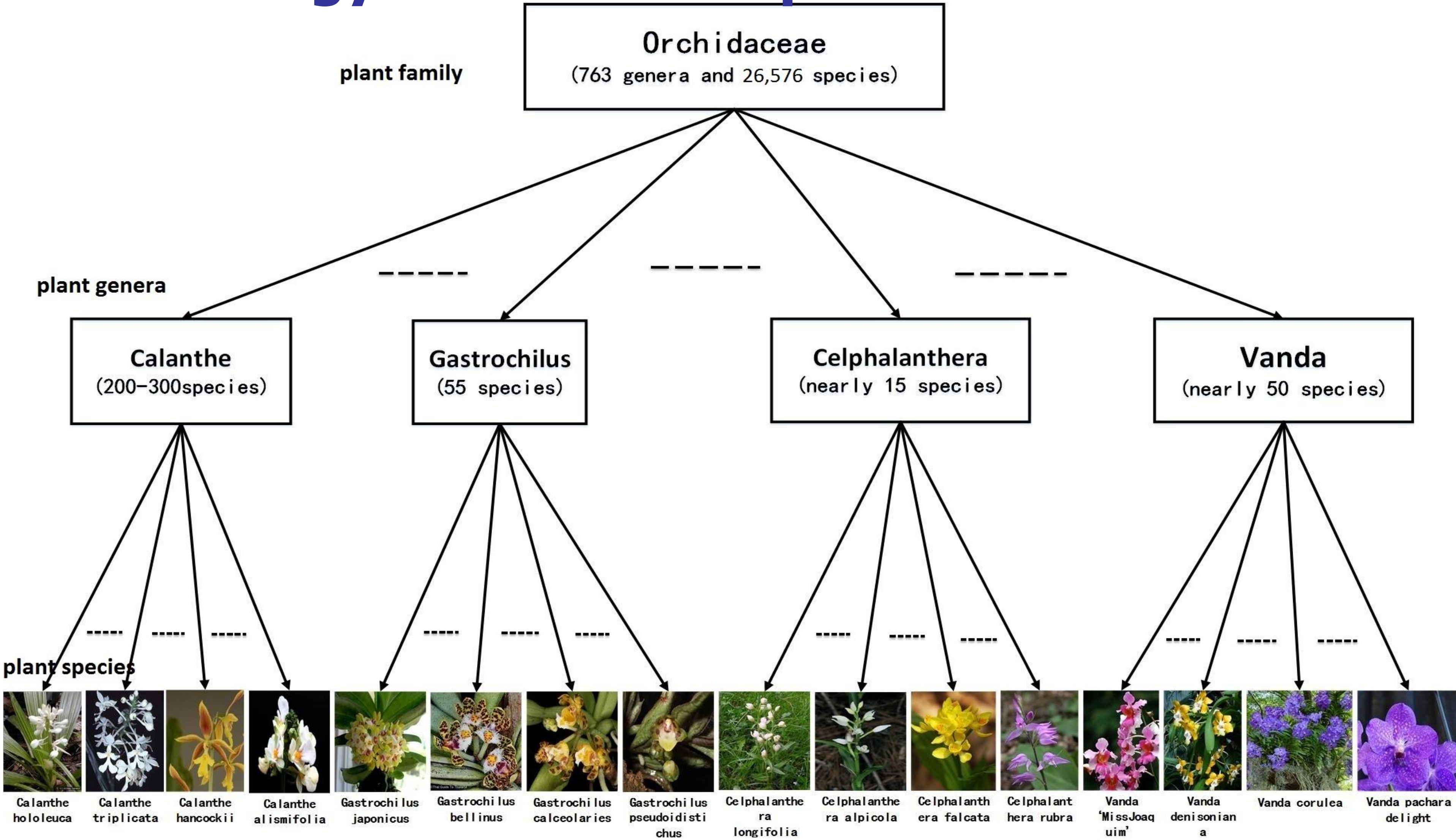

**Two-Layer Ontology for Taobao Products**

# 2. Ontology for Task Group Generation



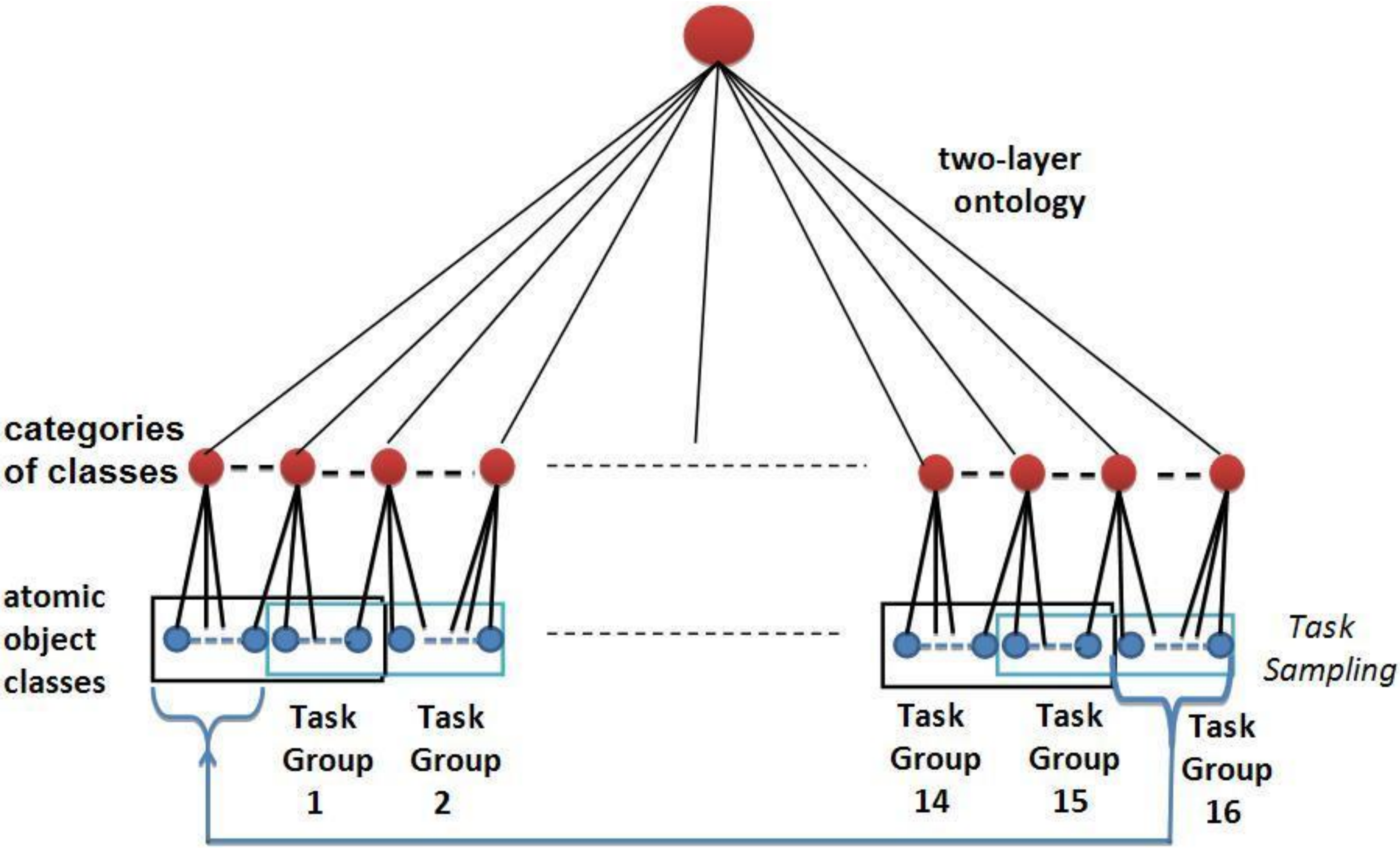

**Plant Ontology**

# 2. Ontology for Task Group Generation

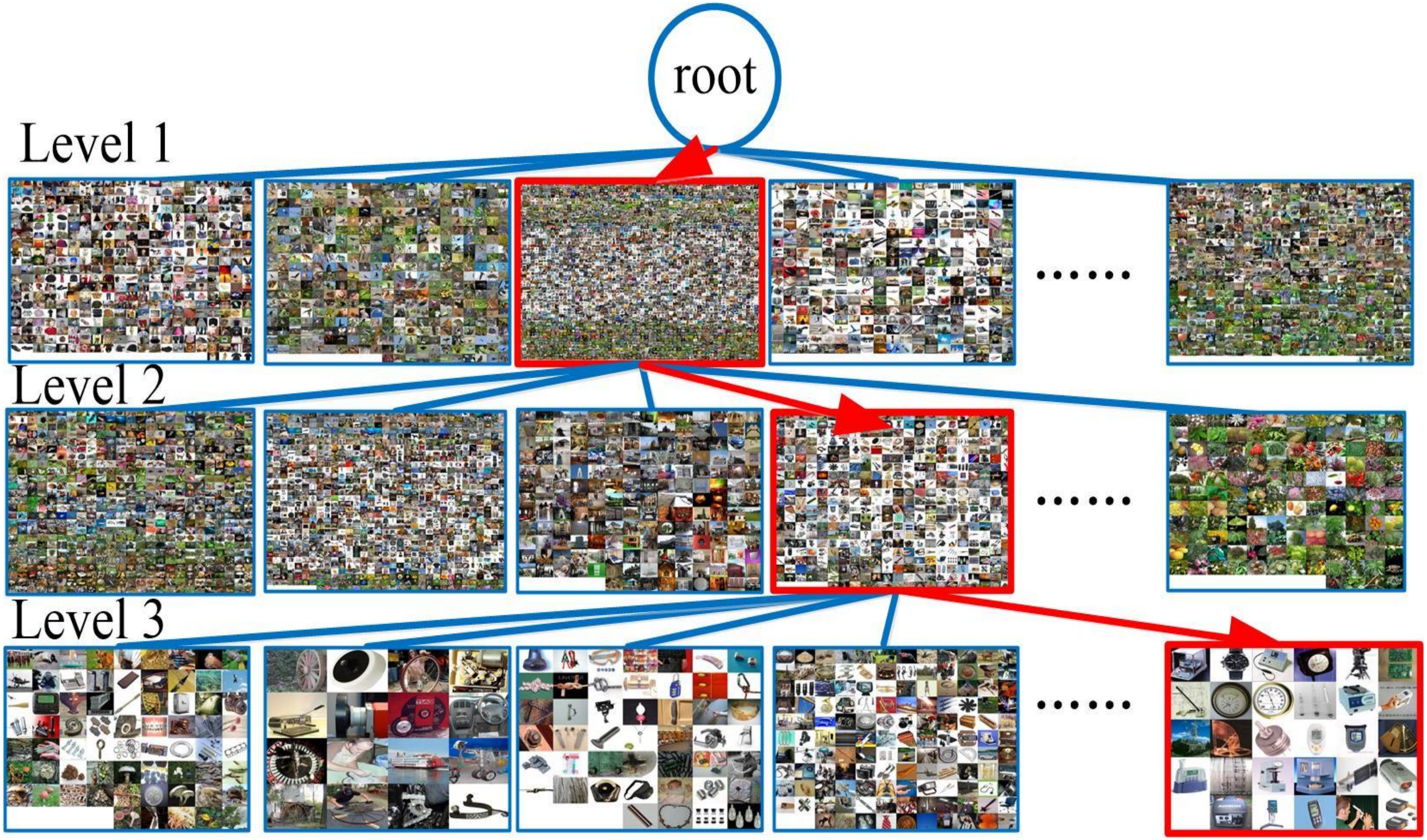



**Two-Layer Ontology for Orchidaceae**

# 2. Ontology-Driven Task Group Generation

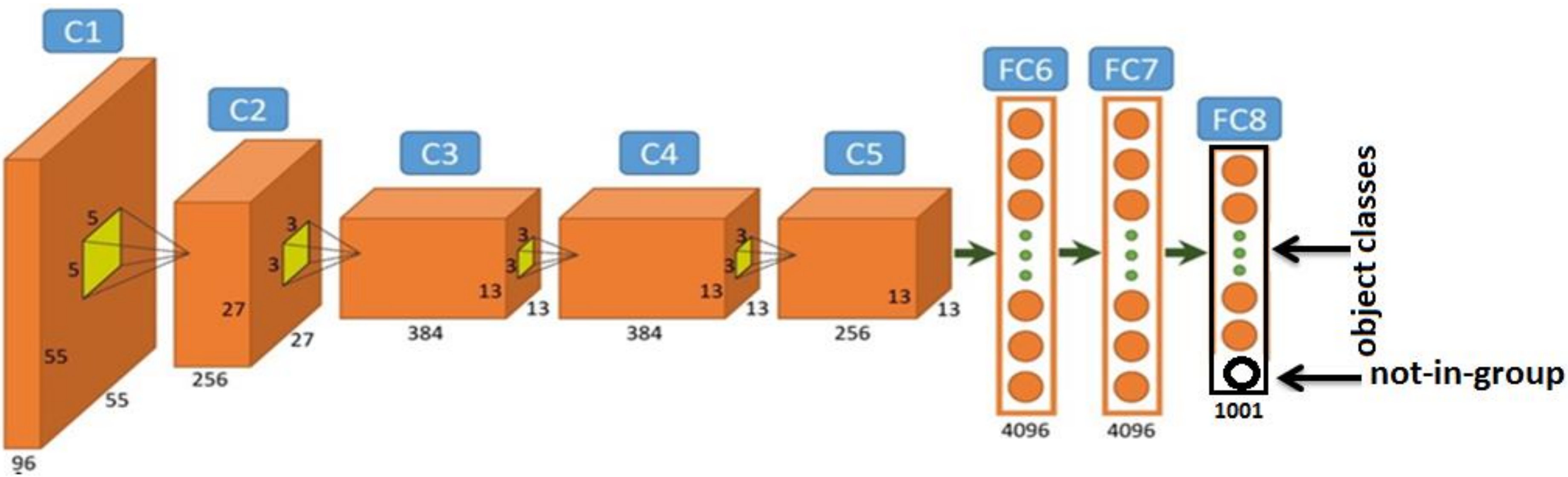


# 2. Ontology-Driven Task Group Generation



# 3. Learning Base CNNs for Each Task Group

- **Design of Base Deep CNNs**




**AlexNet, VGG, GoogleNet, ResNet, ……**

**MobileNet can be selected for smartphone applications!**

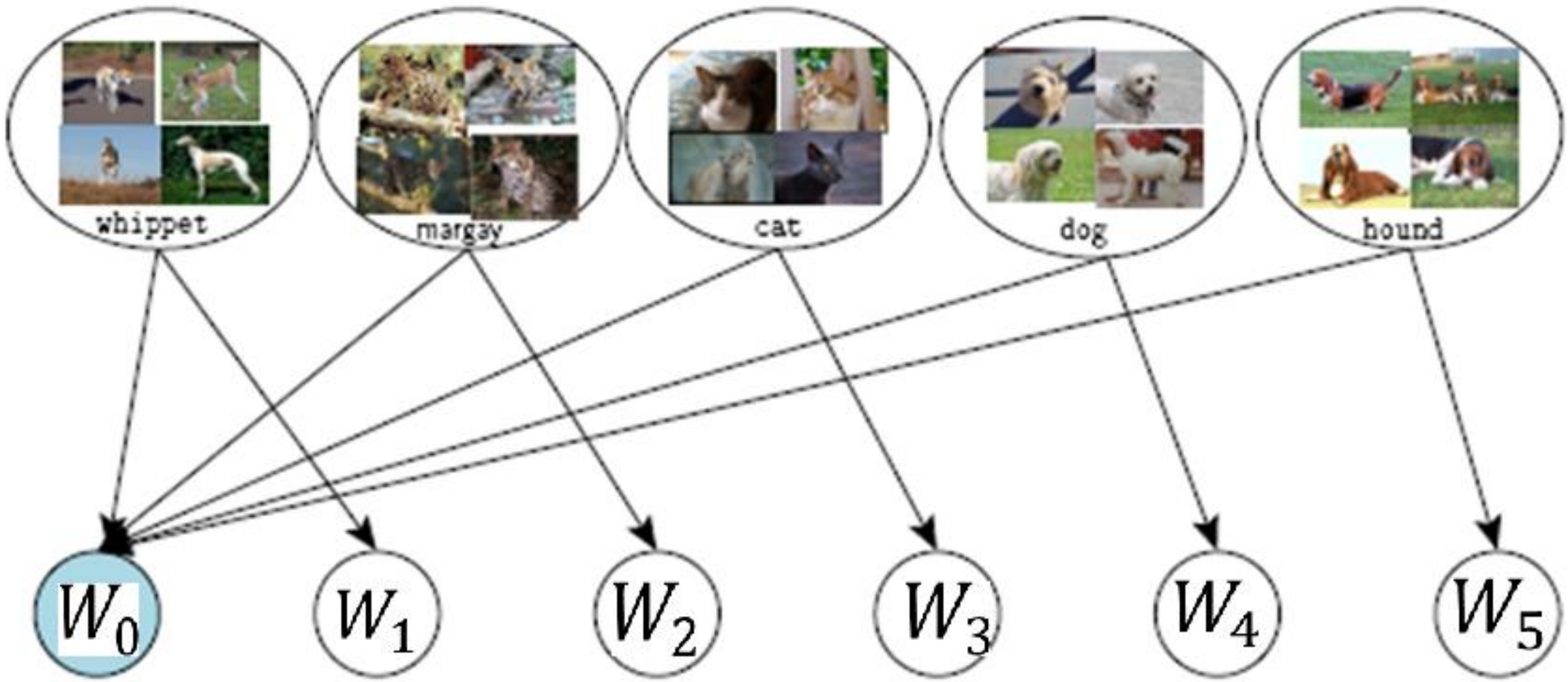
# 3. Learning Base CNNs for Each Task Group

- **Approaches for Learning Base Deep CNNs**
  - **Deep Multi-Task Learning**
  - **Hierarchical Deep Multi-Task Learning**
  - **Deep Boosting**
  - **Deep Collaborative Learning**

***IEEE TIP 2017, 2018, PAMI 2017, 2018***

# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning:** $F_j(\mathbf{x}) = (W_0 + W_j)^T \mathbf{x} + \mathbf{b}$



# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**

$$min \left\{ C \sum_{l=1}^{R} \sum_{j=1}^{B} \xi_j^l + \delta_1 Tr\left(WW^T\right) + \frac{\delta_2}{2} Tr\left(WLW^T\right) \right\}$$

***subject to:***

$$\forall_{l=1}^{R} \forall_{j=1}^{B} : \quad y_j^l \left(W_j^T \cdot x_j^l + b\right) \geq 1 - \xi_j^l, \;\; \xi_j^l \geq 0$$

# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**

$$min\left\{\sum_{j=1}^{B}\sum_{l=1}^{R}\beta_l^j-\frac{1}{2\delta_1}\beta^TY\Re\left(\Re+\frac{\delta_2}{\delta_1}\Re\left(L\bigotimes I\right)\Re\right)^{-1}\Re Y\beta\right\}$$

***subject to:***

$$\forall_{l=1}^{R}\forall_{j=1}^{B}:\ \sum_{l=1}^{R}\beta_l^j\cdot y_l^j=0,\ \ 0\leq\beta_l^j\leq 1$$

# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**

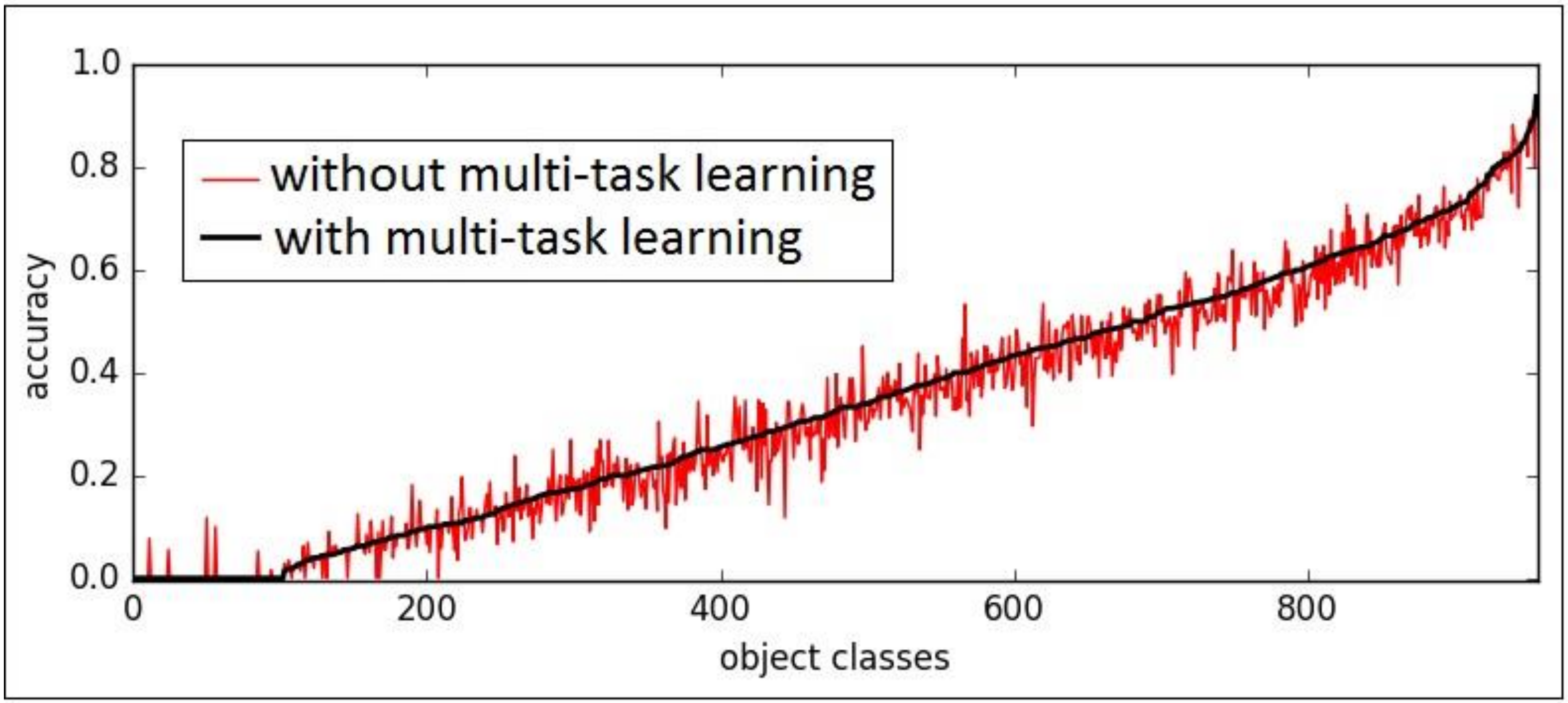
$$\alpha^* = \frac{1}{2\delta_1}\left(\Re + \frac{\delta_2}{\delta_1}\left(\Re\left(L \bigotimes I\right)\Re\right)^{-1}\Re Y \beta^*\right)$$

**Multi-Task Classifiers at Sibling Leaf Nodes**

$$\forall_{j=1}^{B}: \quad f_{c_j}^1(x)\mid_{F_{c_j}^1} = \sum_{l=1}^{R} \alpha_j^{l*}\kappa(x_j^l, x) + b_j^*, \quad c_j \in c_h$$

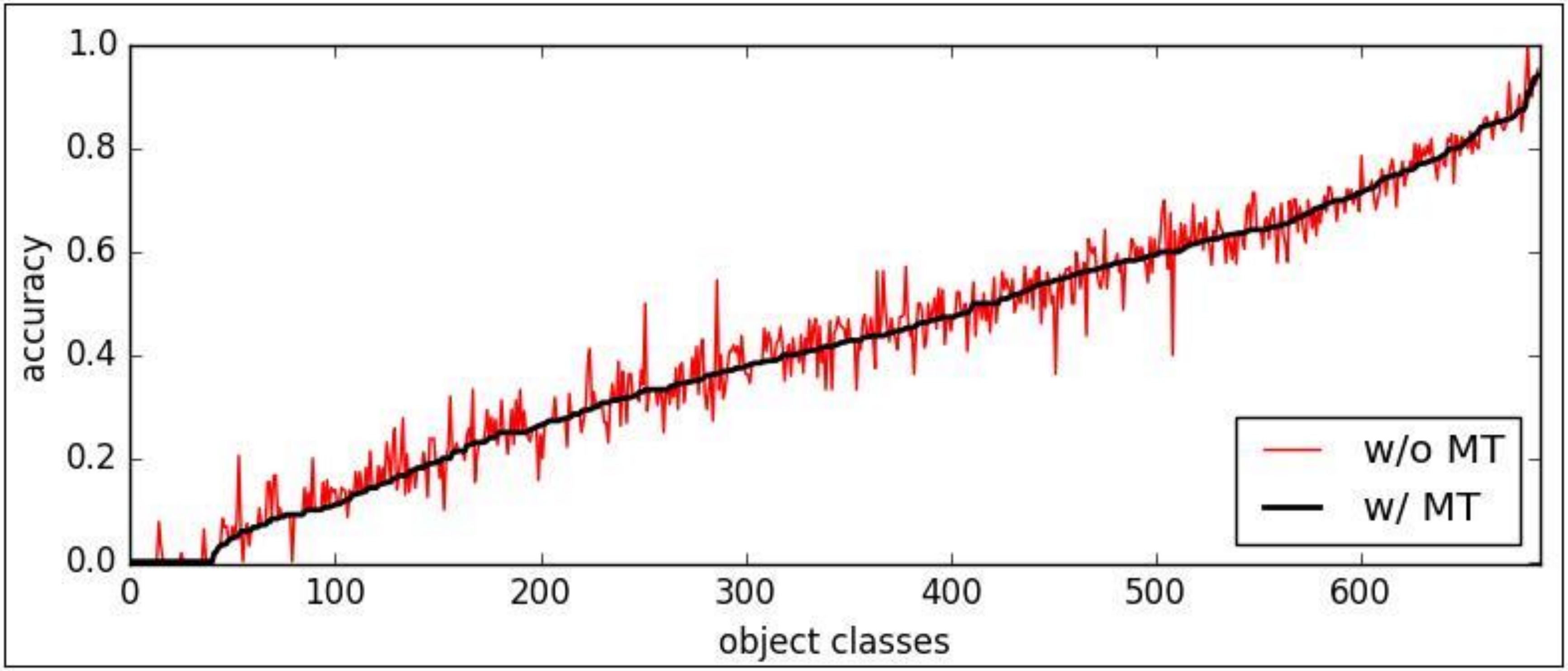
# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**



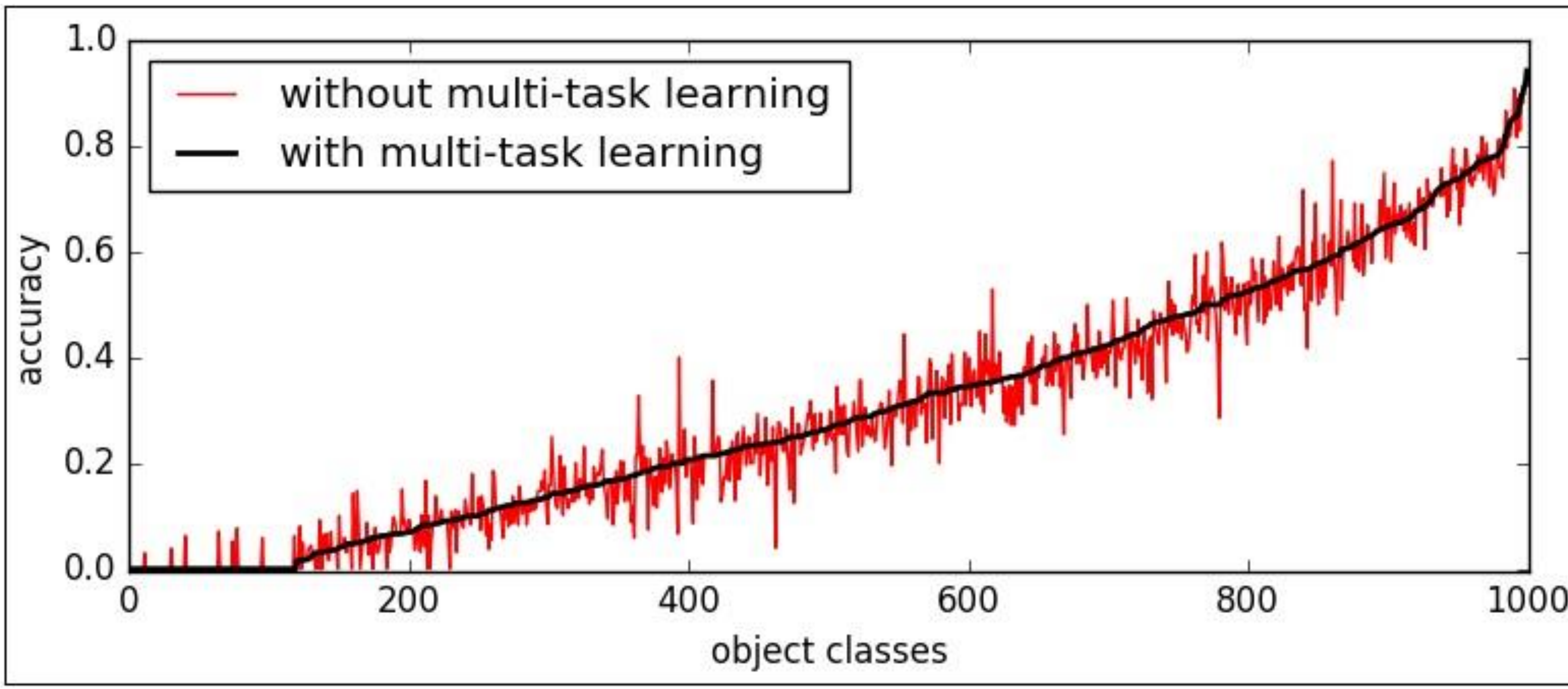
# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**



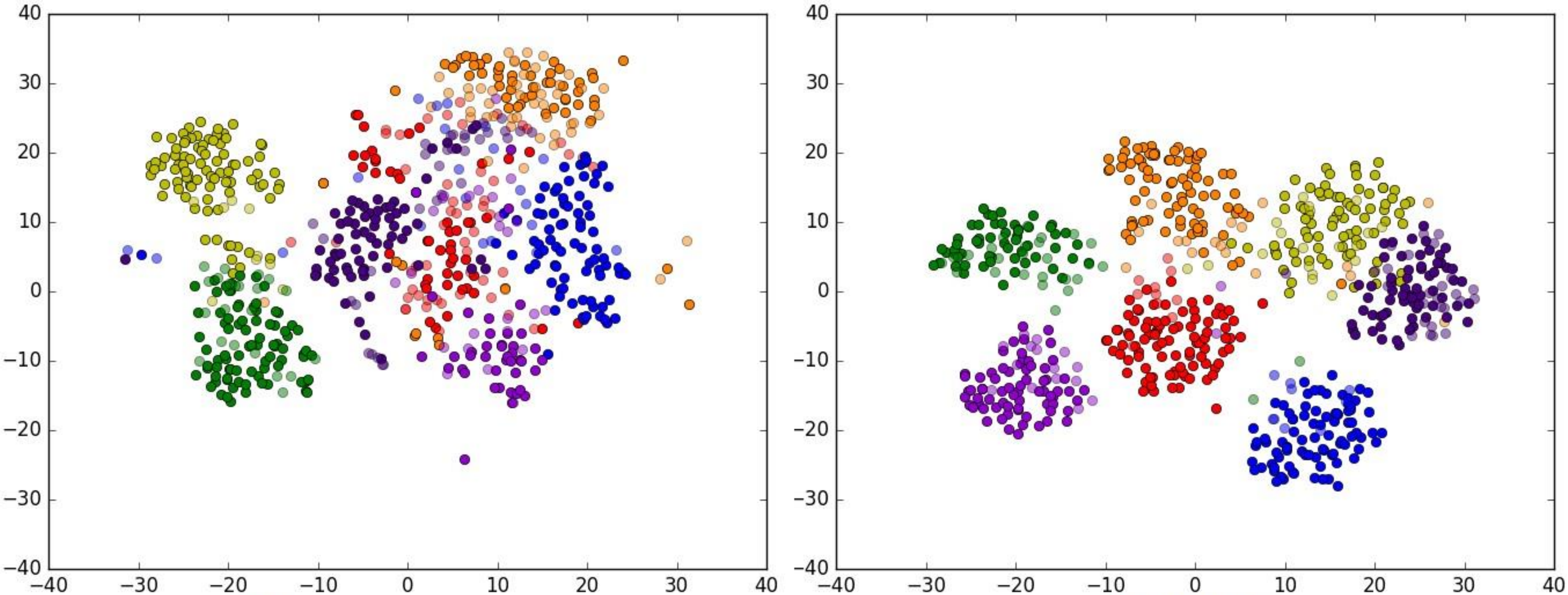
# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**



# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**

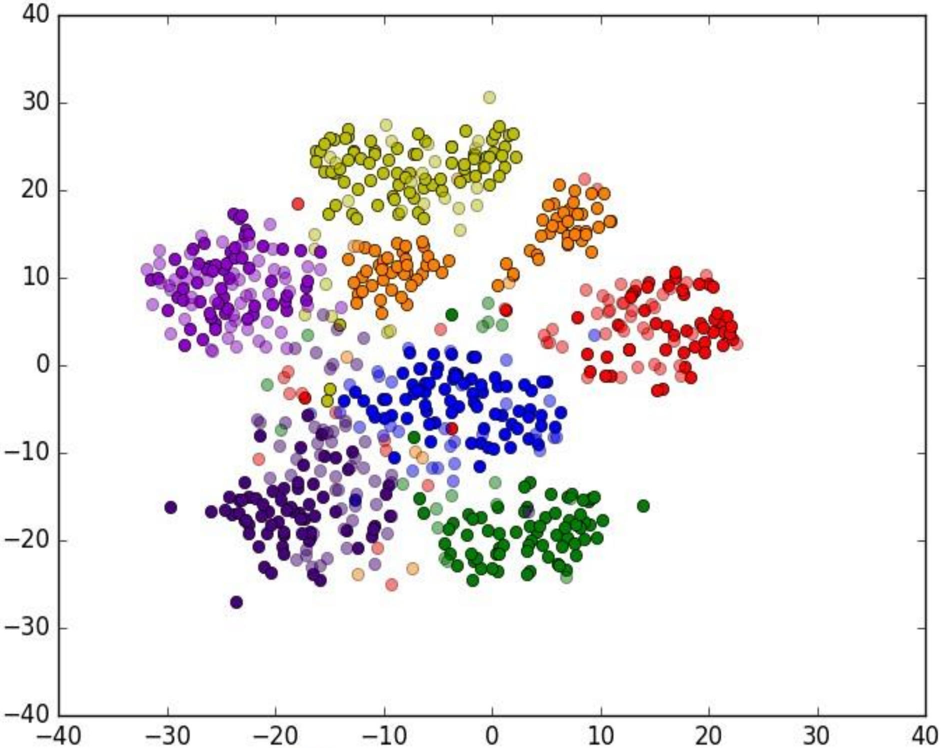



(a) N-way Softmax

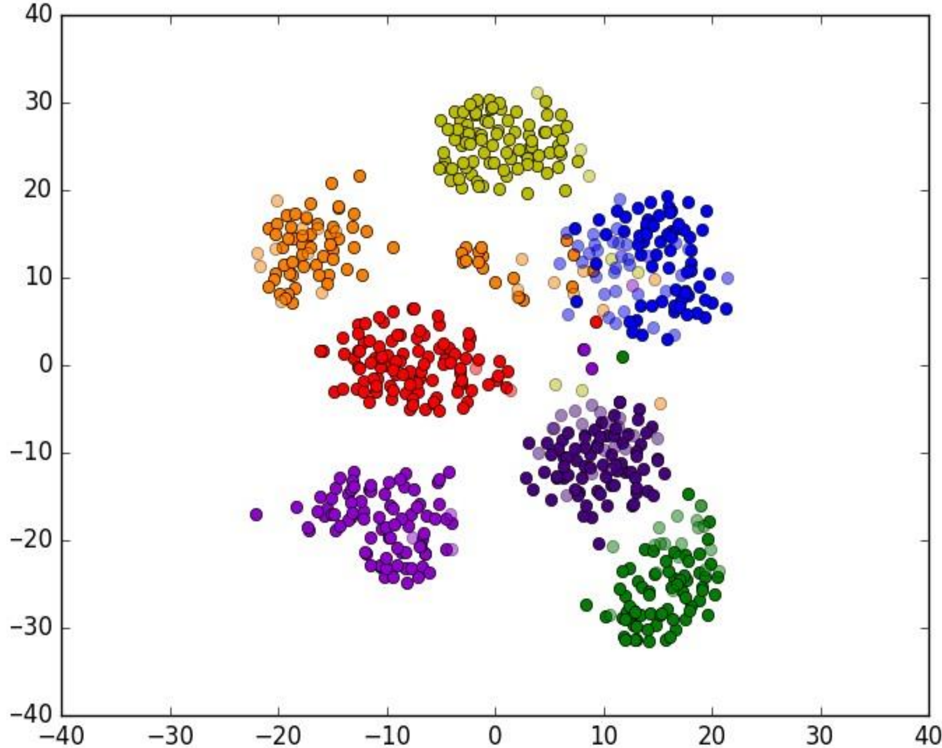
(b) HD-MTL

# 3. Learning Base CNNs for Each Task Group

- **Deep Multi-Task Learning**



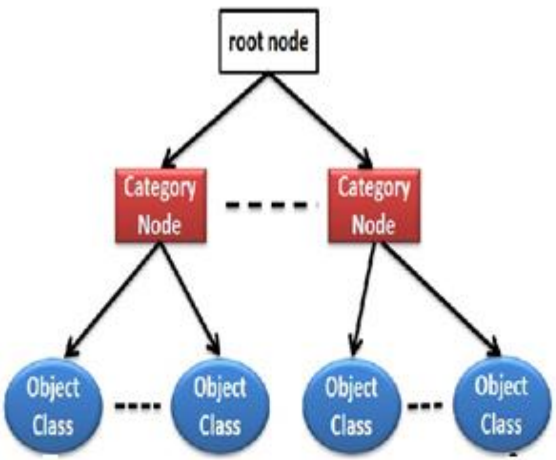

(a) N-way Softmax

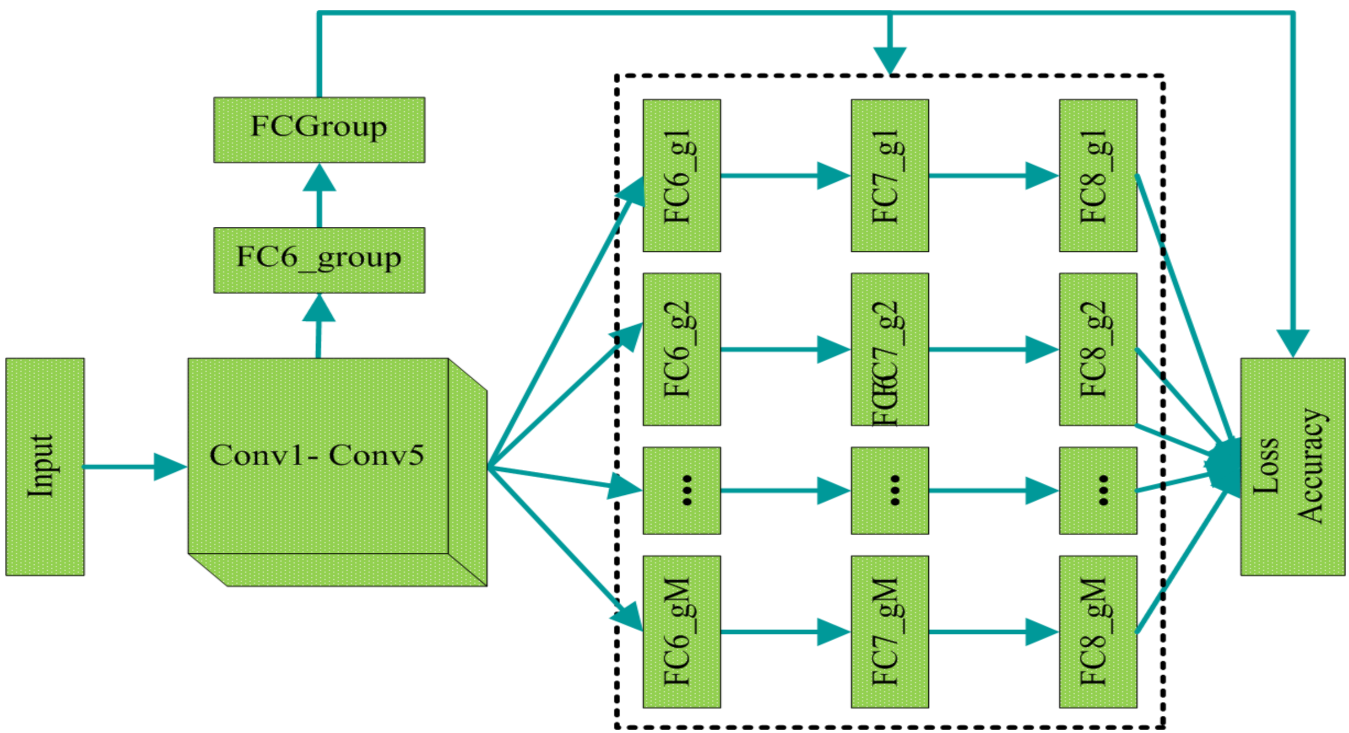



(b) LMM

# 3. Learning Base CNNs for Each Task Group

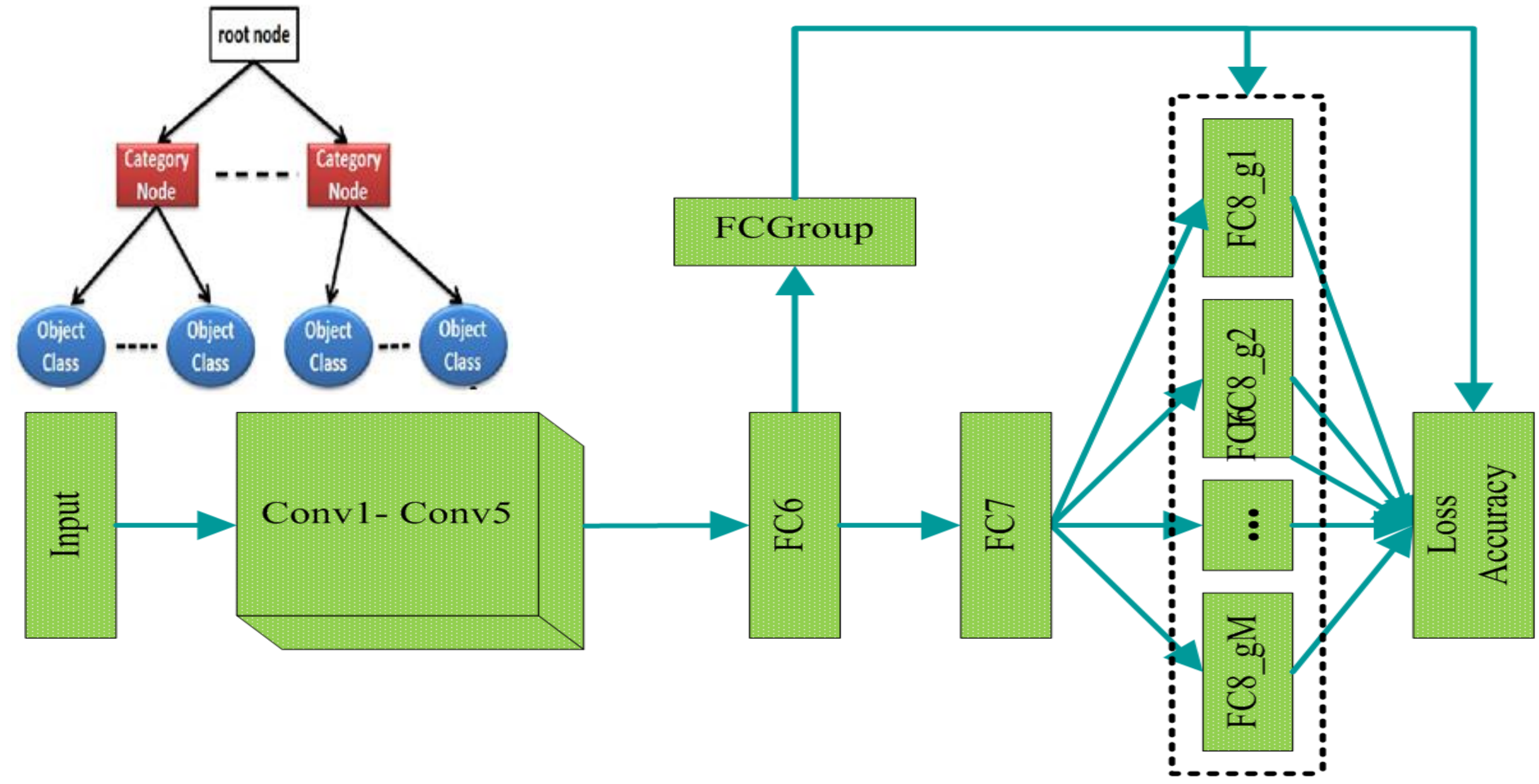
- **Hierarchical Deep Multi-Task Learning**



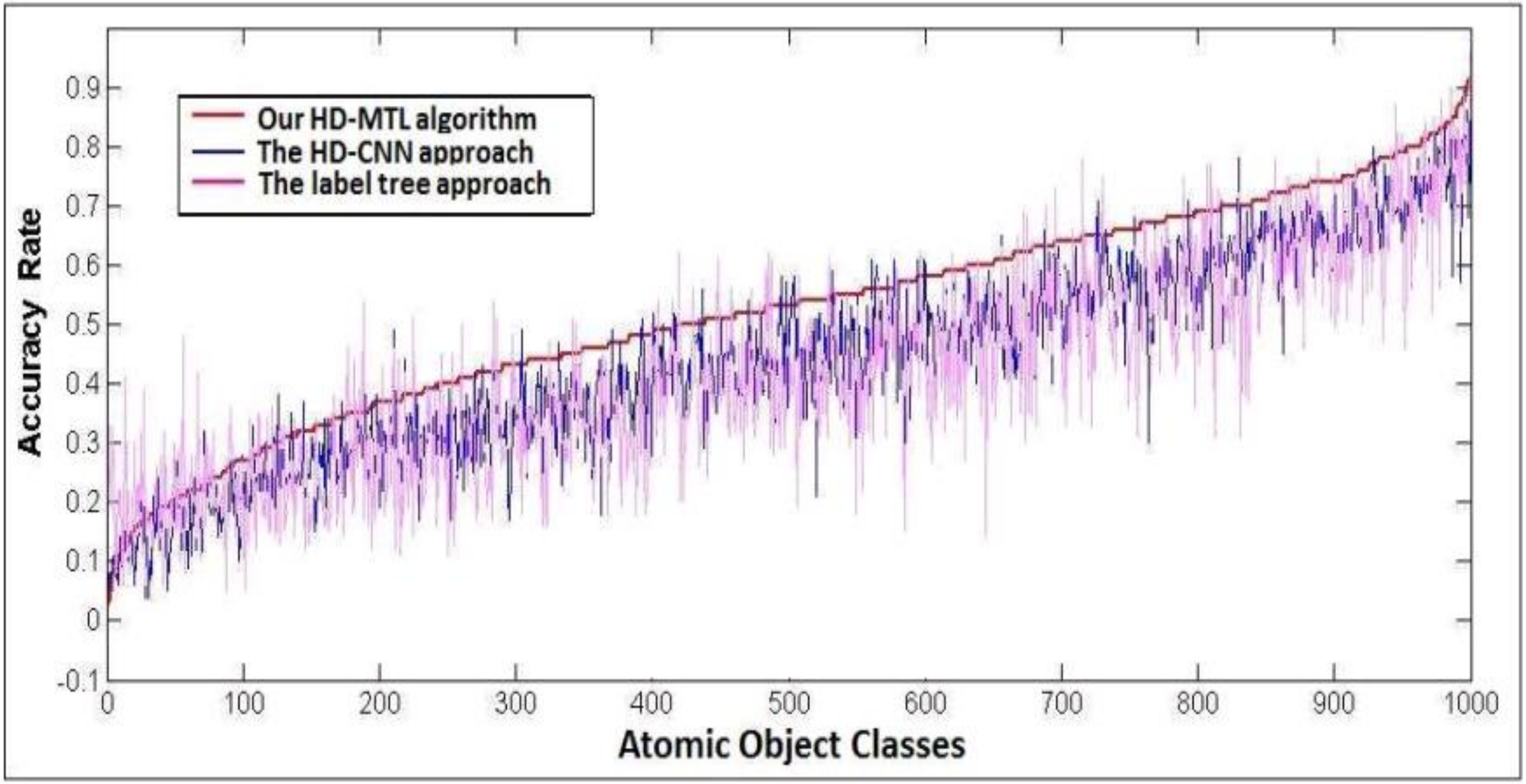
# 3. Learning Base CNNs for Each Task Group

- **Hierarchical Deep Multi-Task Learning**



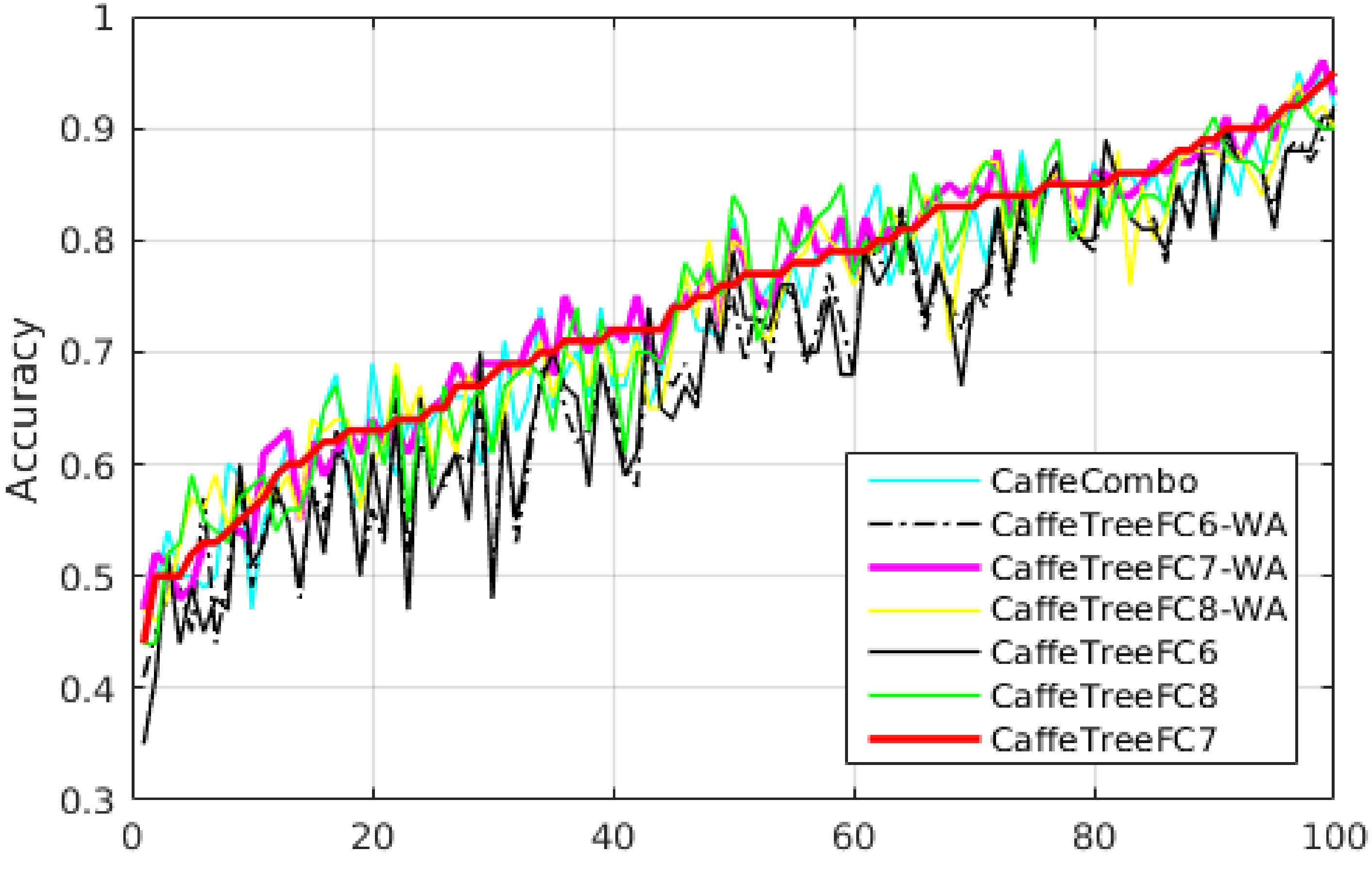
# 3. Learning Base CNNs for Each Task Group

- **Hierarchical Deep Multi-Task Learning**



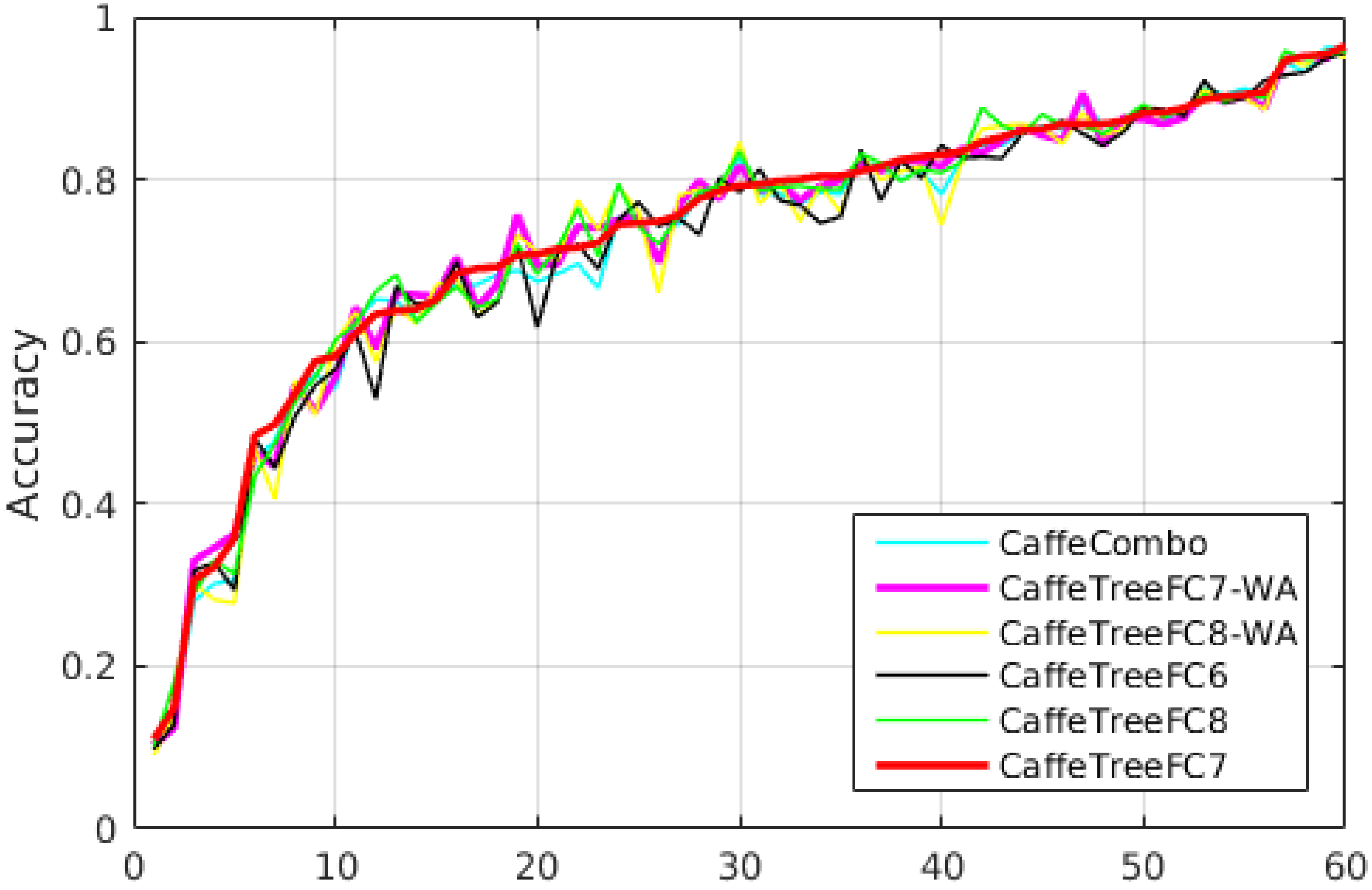
# 3. Learning Base CNNs for Each Task Group

- **Hierarchical Deep Multi-Task Learning**



# 3. Learning Base CNNs for Each Task Group

- **Hierarchical Deep Multi-Task Learning**



# 3. Learning Base CNNs for Each Task Group

- **Hierarchical Deep Multi-Task Learning**

## 3. Learning Base CNNs for Each Task Group

- **Deep Boosting**
  - **Hard object classes** may have higher learning complexities, but **easy** ones may have lower learning complexities;
  - Learning a **joint network** for both of them may not make sense, e.g., their errors may have significantly different effects on optimizing their joint objective function!

# Deep Boosting

## Visualizing object classes according to their Learning Complexities

**IEEE Trans. on Multimedia, 2012**

## Visualization of Seperability

## Easy Classes

## Hard Classes

# 3. Learning Base CNNs for Each Task Group

- **Deep Boosting**

**Weighting on object classes not samples**

**Algorithm 1** Deep Boosting of Complementary Networks

**Require:** Training set for $N$ object classes: $\mathcal{S} = \{(x_i^l, y_i^l) \mid l \in \{1, ..., N, i \in \{1, ..., R\}$ ; Initializing the distribution of importances over $N$ object classes: $\phi_1(C_1) = ... = \phi_1(C_N) = \frac{1}{N}$; Number of complementary deep networks or iterations: $T$.

1: **for** $t = 1, \ldots, T$ **do**

2: Normalizing the distribution of importances over $N$ object classes: $\varphi_t(C_l) = \frac{\phi_t(C_l)}{\sum_{j=1}^{N} \phi_t(C_j)}, l = 1, ..., N$

3: Training the $t^{th}$ complementary deep network $f_t(x)$ according to the normalized distribution of importances over $N$ object classes $[\varphi_t(C_1), ..., \varphi_t(C_N)]$;

4: Calculating the error rate $\varepsilon_t(C_l)$ for each object class;

5: Computing the weighted error rate for the $t^{th}$ complementary deep network $f_t(x)$: $\varepsilon_t = \sum_{l=1}^{N} \varphi_t(C_l)\varepsilon_t(C_l)$;

6: Setting the parameter $\beta_t = \frac{\lambda\varepsilon_t}{1-\lambda\varepsilon_t}$;

7: Updating the distribution of importances over $N$ object classes $\phi_{t+1}(C_l)$ as: $\phi_{t+1}(C_l) = \phi_t(C_l)\beta_t^{1-\lambda\varepsilon_t(C_l)}$, $l = 1, ..., N$, so that the hard object classes, which have larger error rates and are misclassified by $f_t(x)$, can receive larger weights (importances) when we train the $(t+1)^{th}$ complementary deep network $f_{t+1}(x)$ at the next round;

8: **end for**

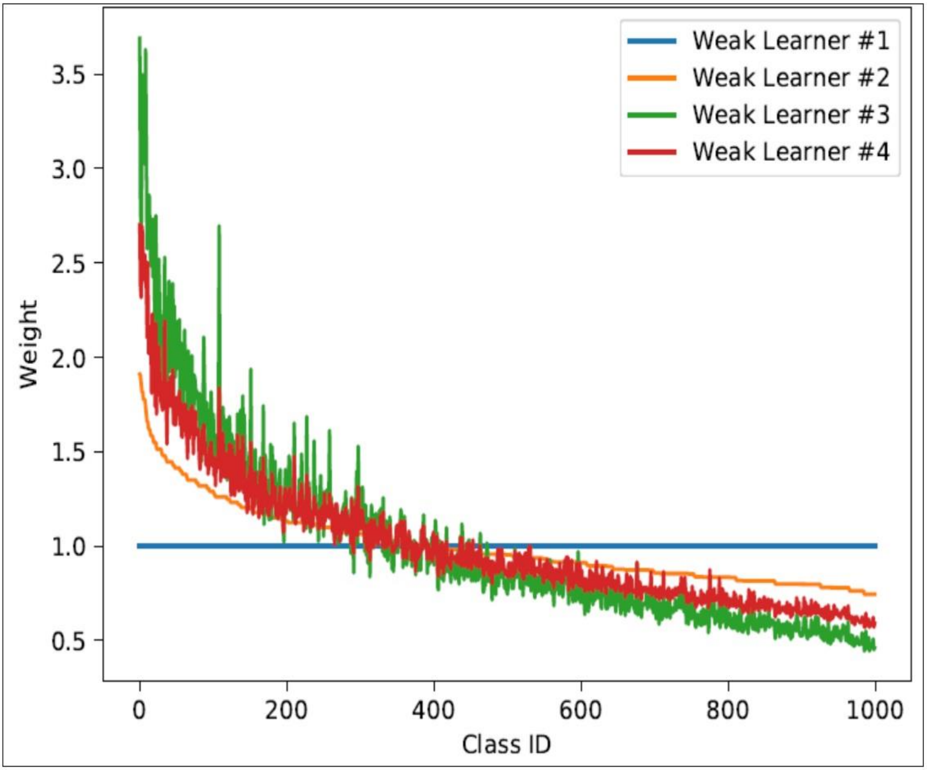
9: Outputting the ensemble network: $\mathbb{F}(x) = \frac{1}{\mathbb{Z}} \sum_{t=1}^{T} \log\left(\frac{1}{\beta_t}\right) f_t(x)$

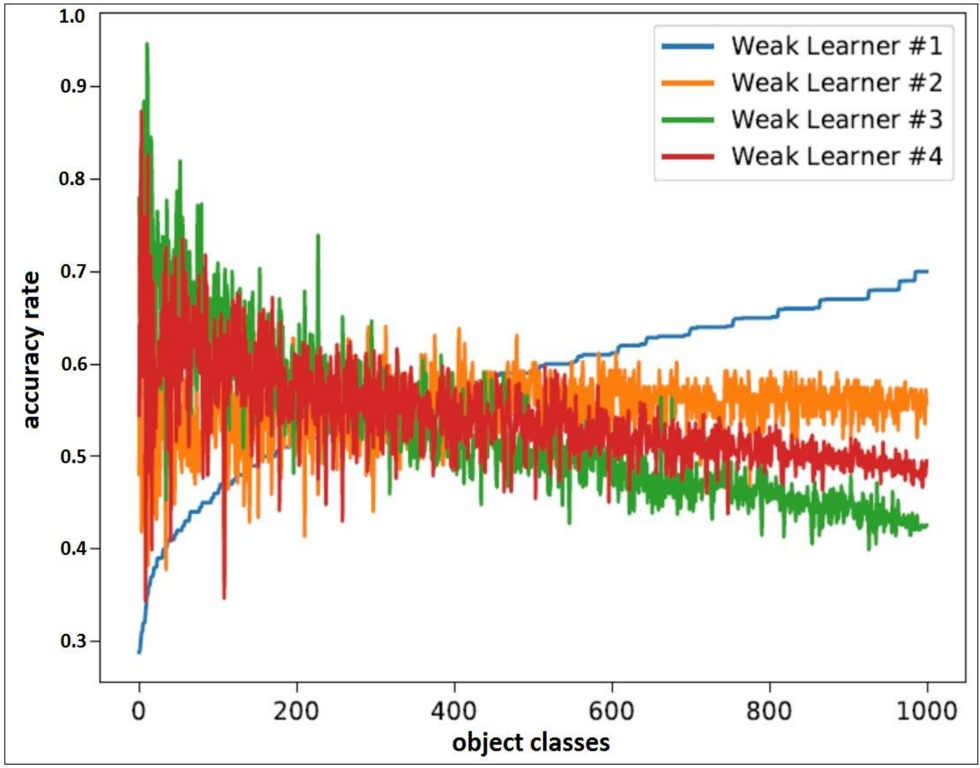
# 3. Learning Base CNNs for Each Task Group

- Deep Boosting
  - All the complementary networks focus on different subsets of 1000 object classes in the same task group;
  - They can enhance each other
  - Their importance or contributions depends on their performances

# 3. Learning Base CNNs for Each Task Group

- **Deep Boosting:** distribution of importance & accuracy rates

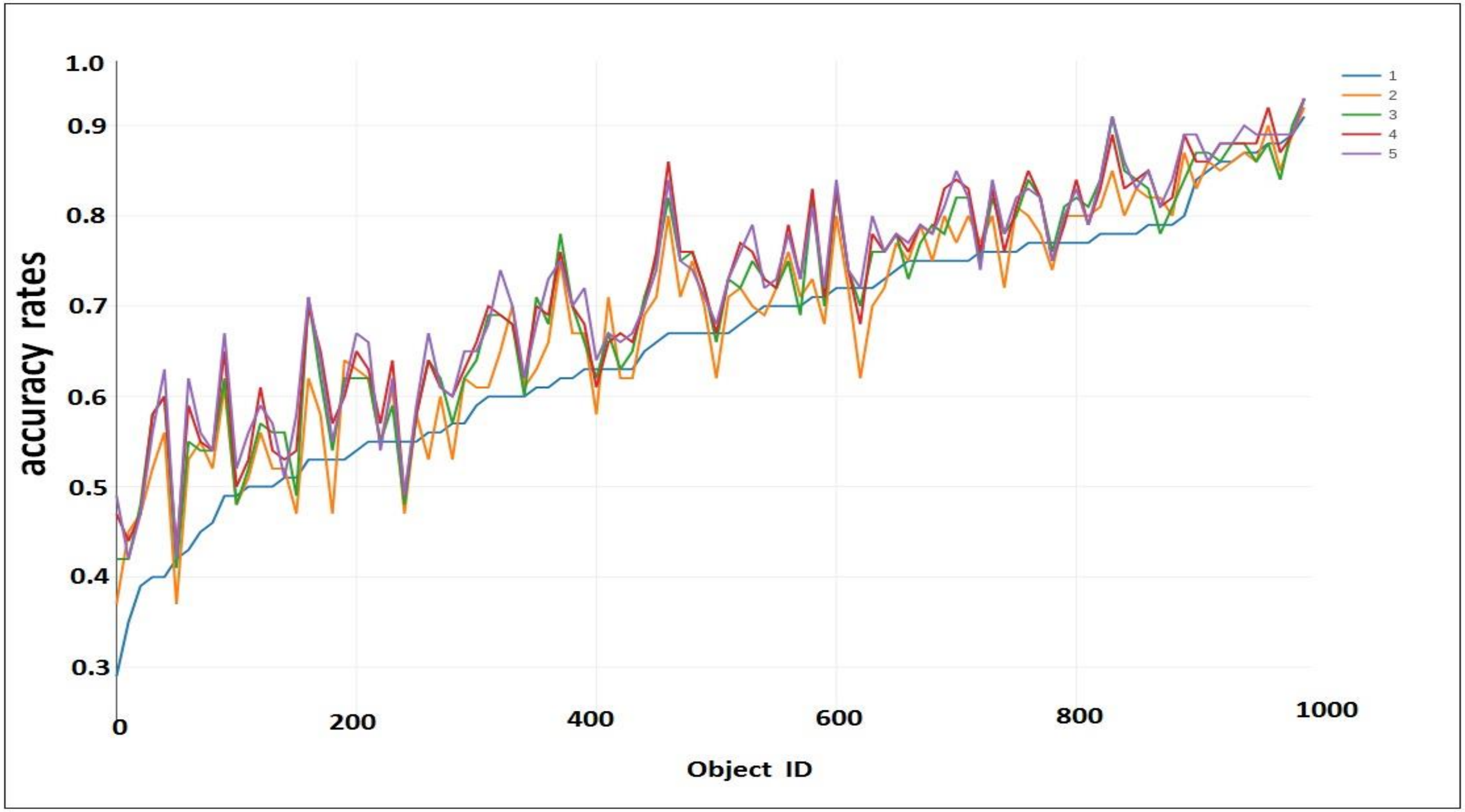



**always-hard object classes**

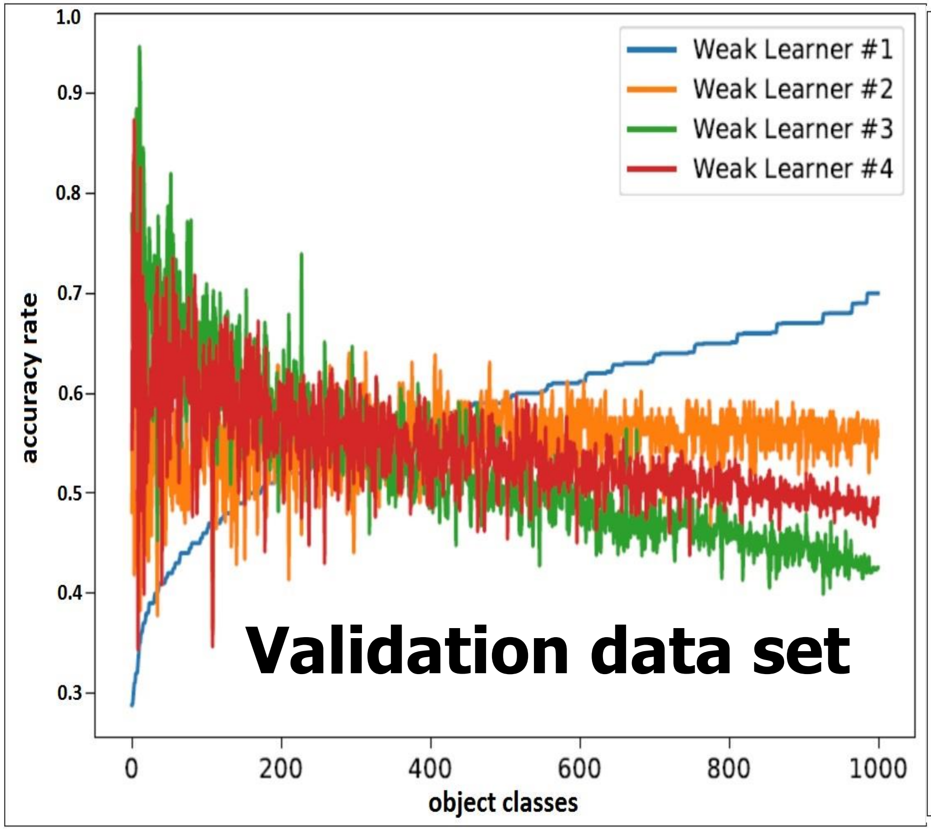
# 3. Learning Base CNNs for Each Task Group

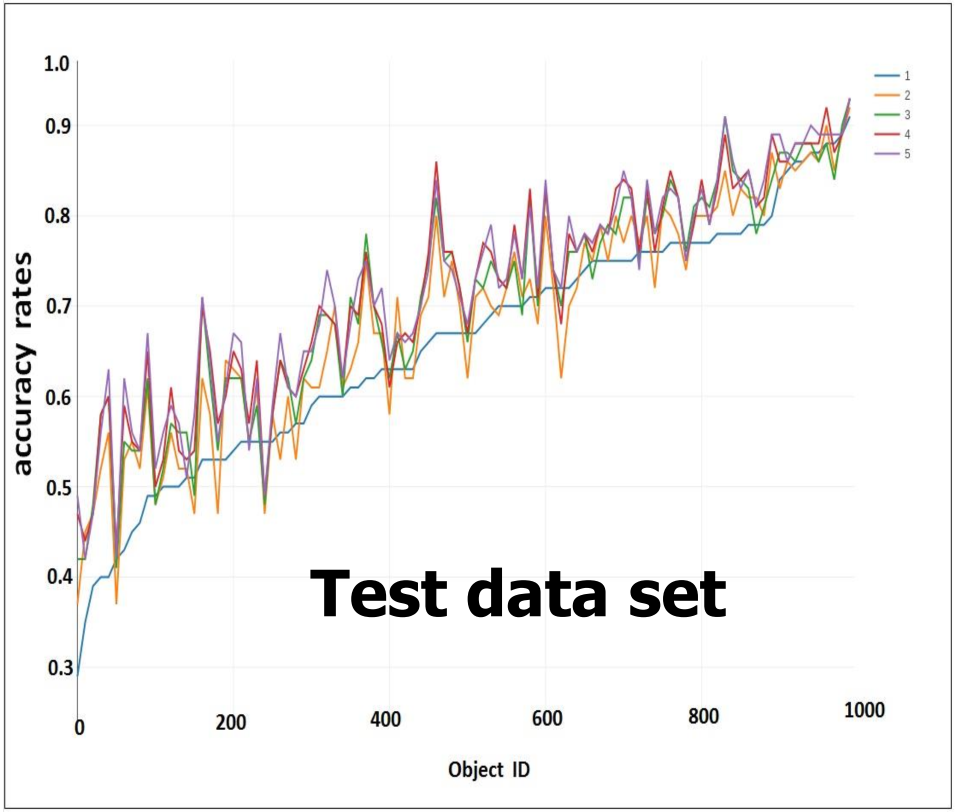
- **Deep Boosting**



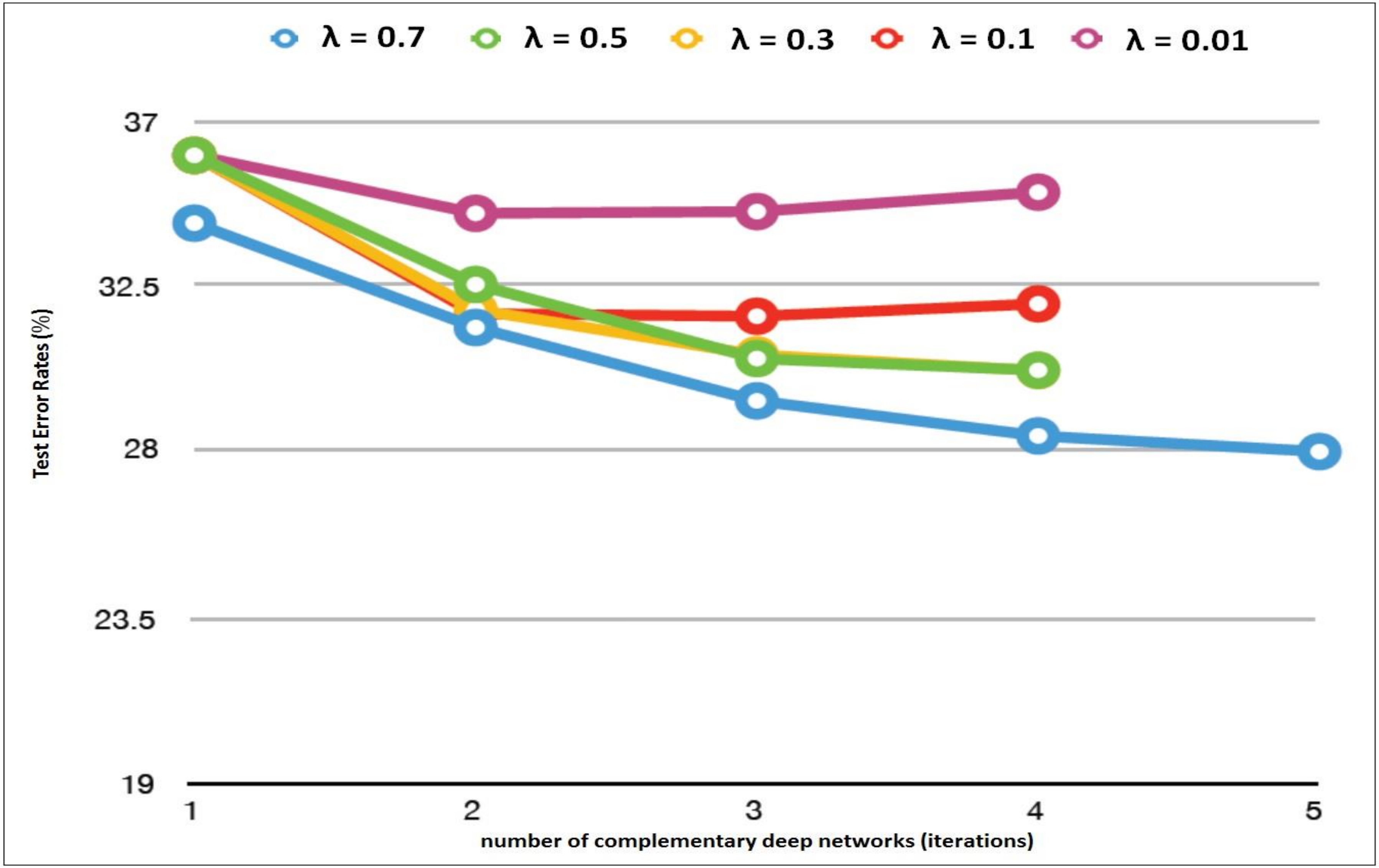
# 3. Learning Base CNNs for Each Task Group

- **Deep Boosting**

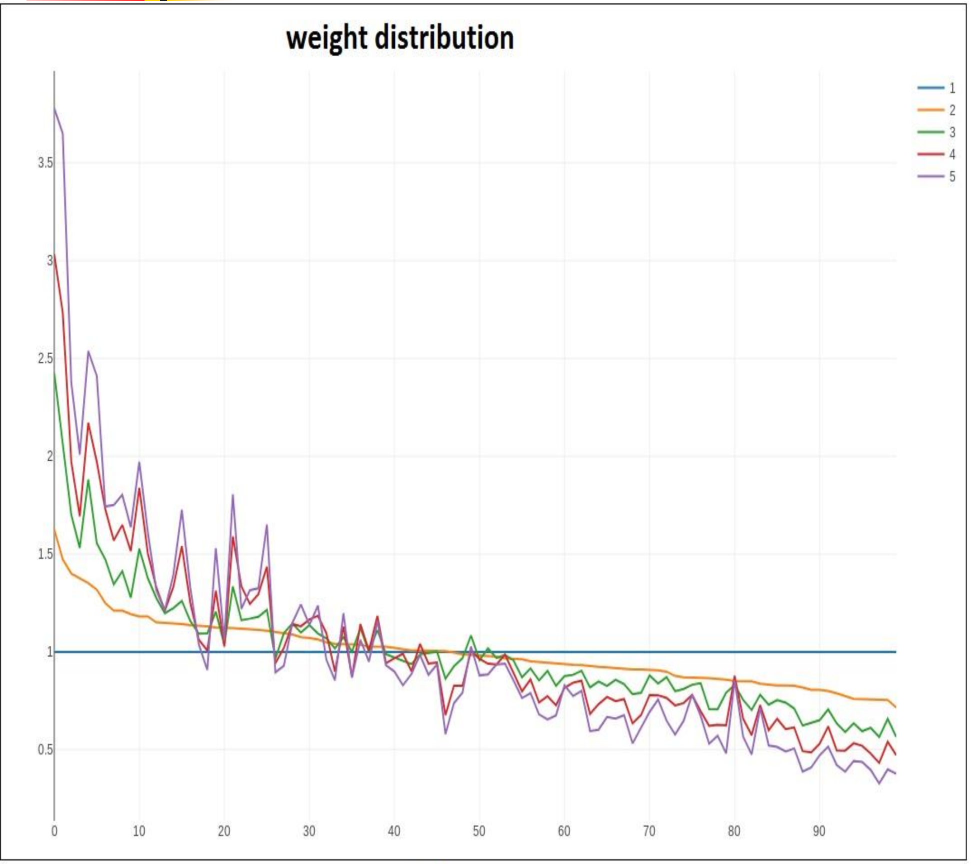


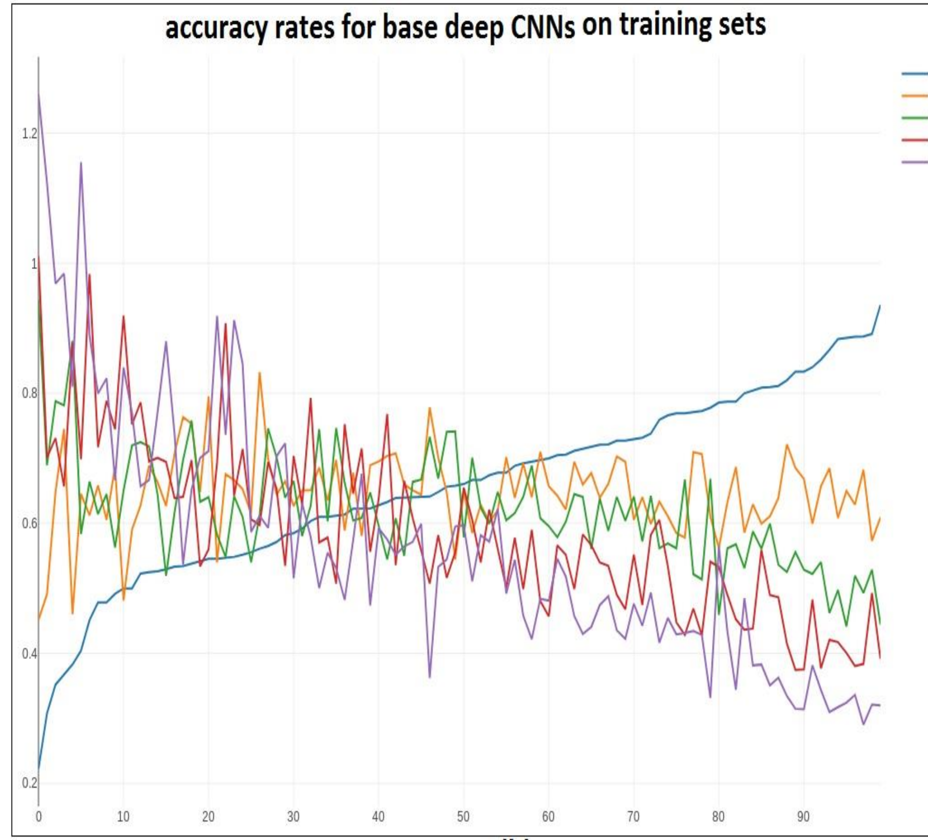
# 3. Learning Base CNNs for Each Task Group



# 3. Learning Base CNNs for Each Task Group



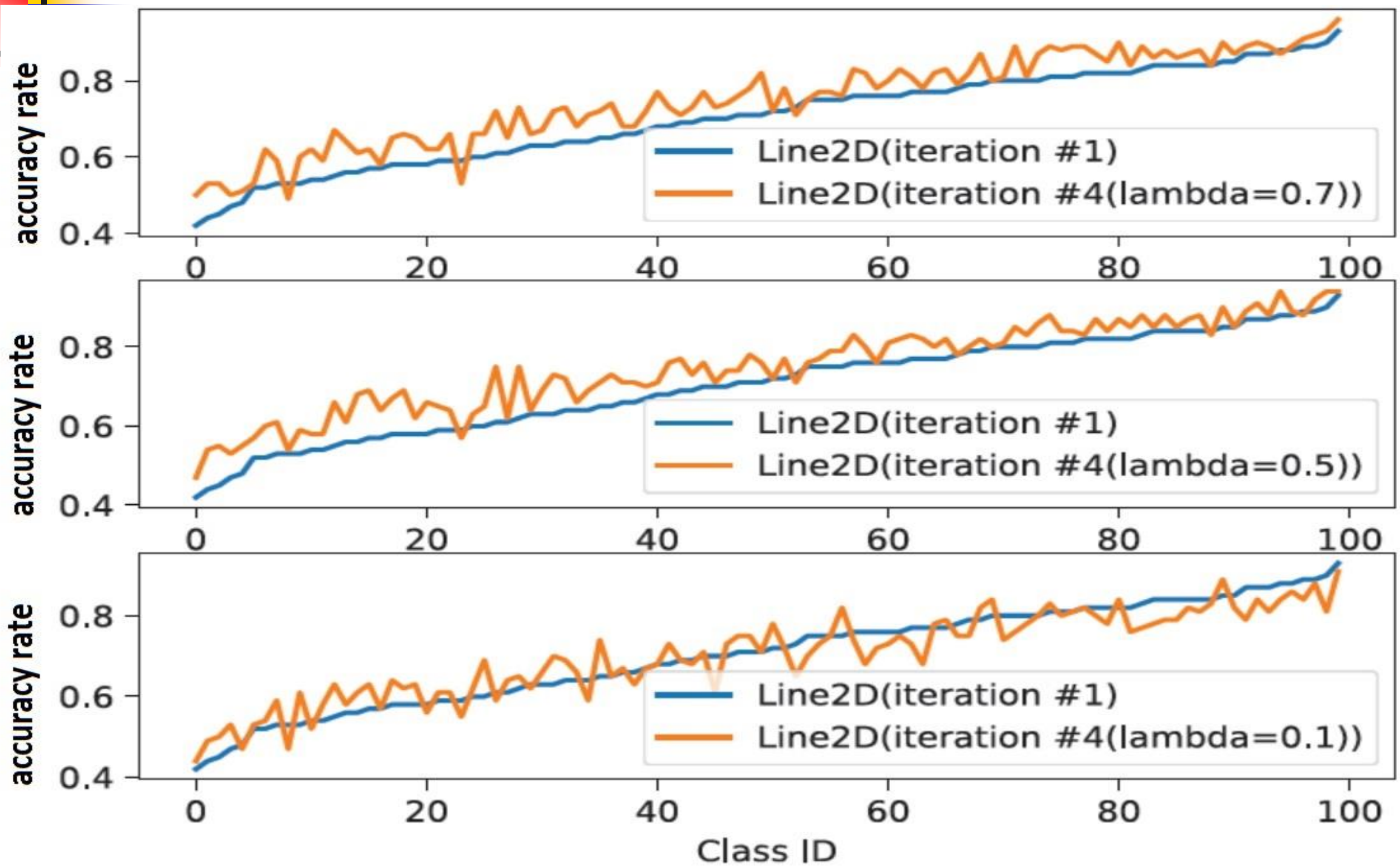

(a)




(b)

**CIFAR-100**

# 3. Learning Base CNNs for Each Task Group




**CIFAR-100**

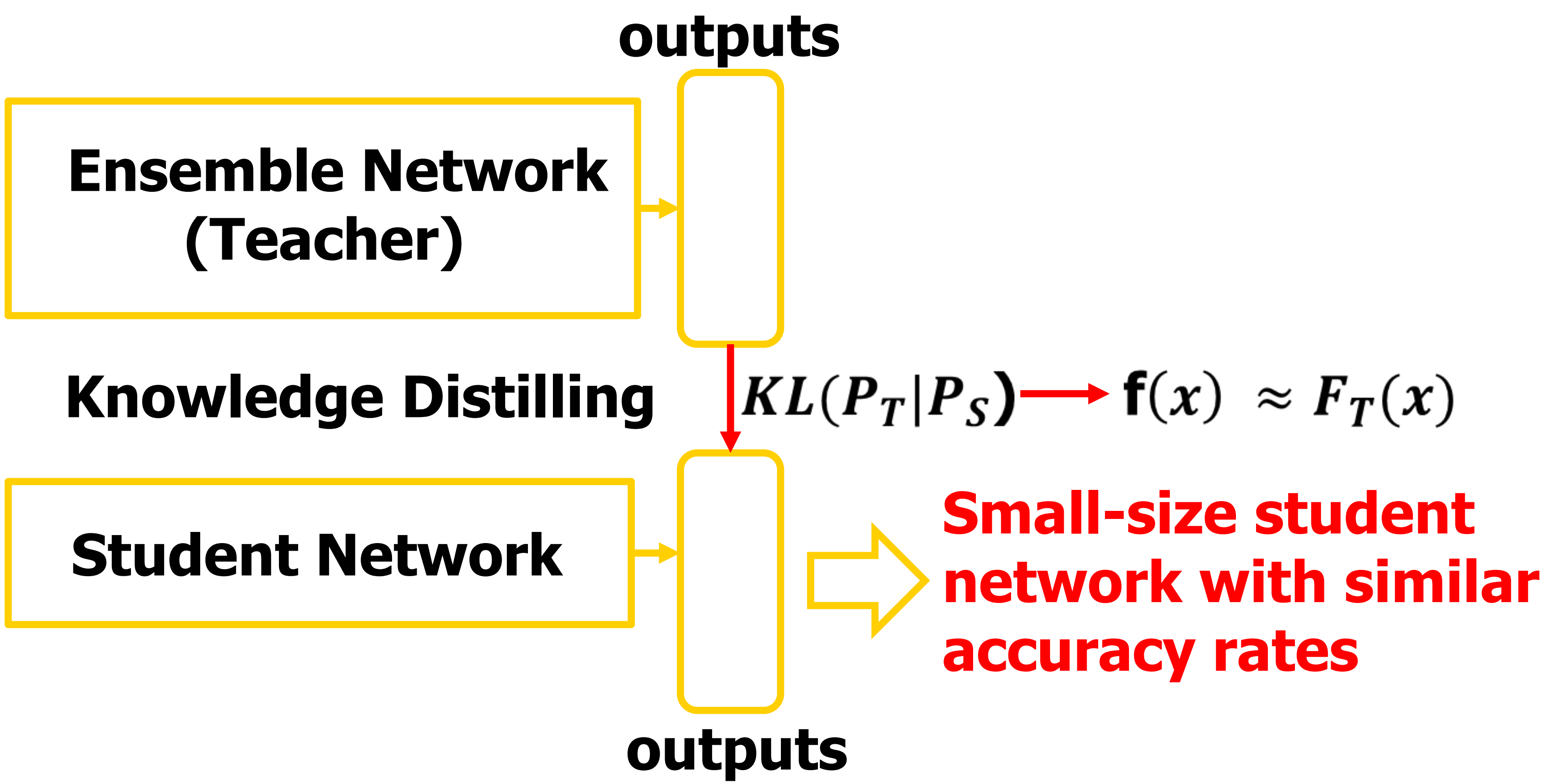
# 3. Learning Base CNNs for Each Task Group

- **Shortages for Using Ensemble Network**
  - **Large space for parameter storage**
  - **Large memory for execution**
  - **Huge computation cost**
  - **Low comprehensibility**

**Even using ensemble network can achieve better accuracy rates, it may be unsuitable for smartphone applications**

## 3. Learning Base CNNs for Each Task Group

- **Knowledge Distillation for Network Learning**



# 3. Learning Base CNNs for Each Task Group

- **Deep Collaborative Learning**


**(a) Individual Learning** **(b) Collaborative Learning**

# 3. Learning Base CNNs for Each Task Group

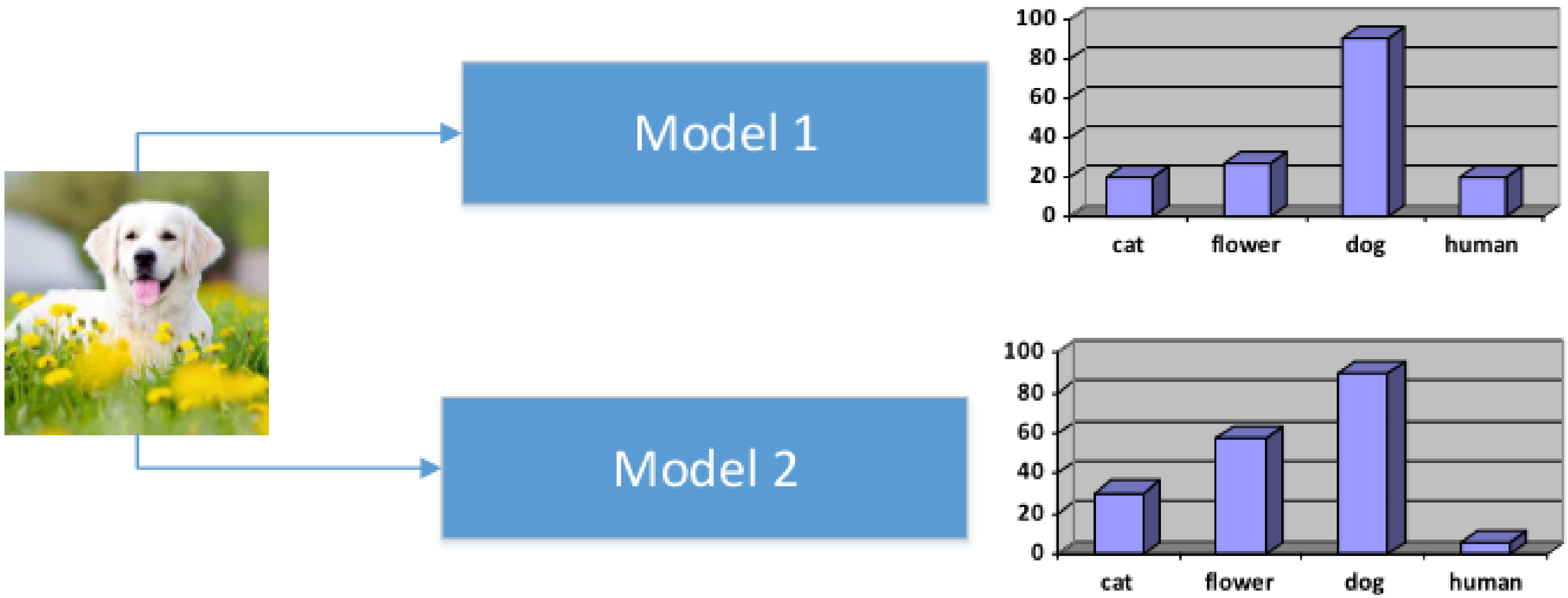
- **Deep Collaborative Learning**

**benefits**

- easier to get "unstuck" with others' help
- exposed to and exchange diverse viewpoints/beliefs
- the opportunity to converse with peers, present and defend ideas

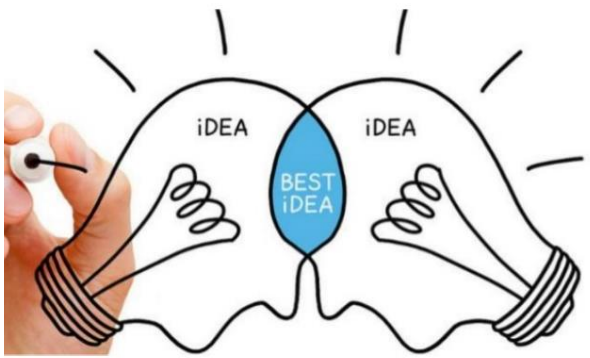
# 3. Learning Base CNNs for Each Task Group

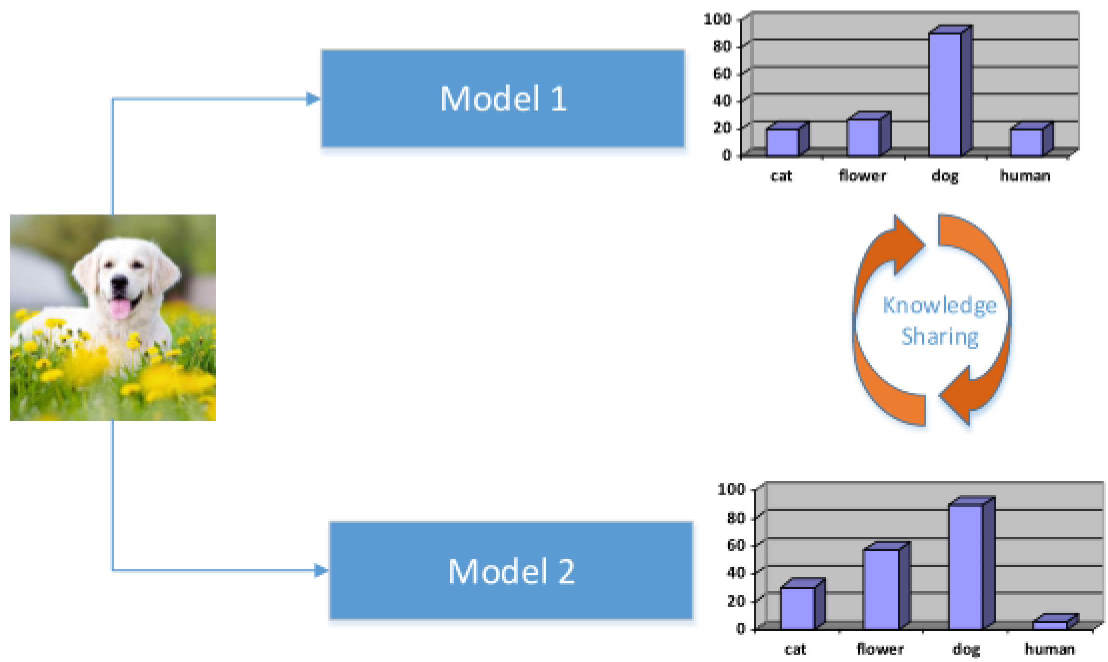
- **Deep Collaborative Learning**



# 3. Learning Base CNNs for Each Task Group

- **Deep Collaborative Learning**



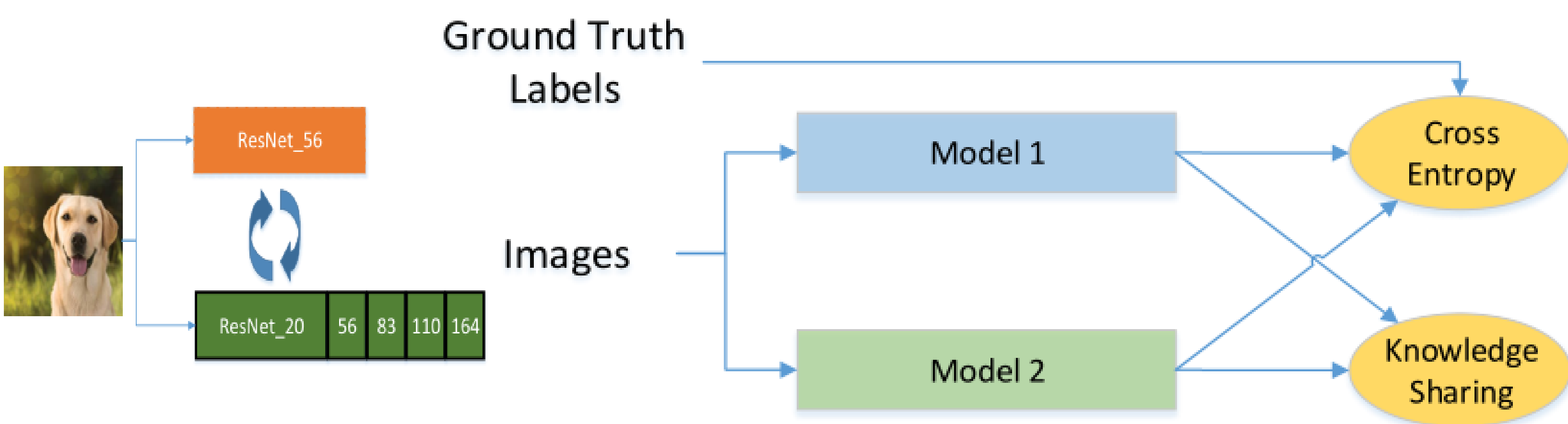
# 3. Learning Base CNNs for Each Task Group

- **Deep Collaborative Learning**

**Loss**

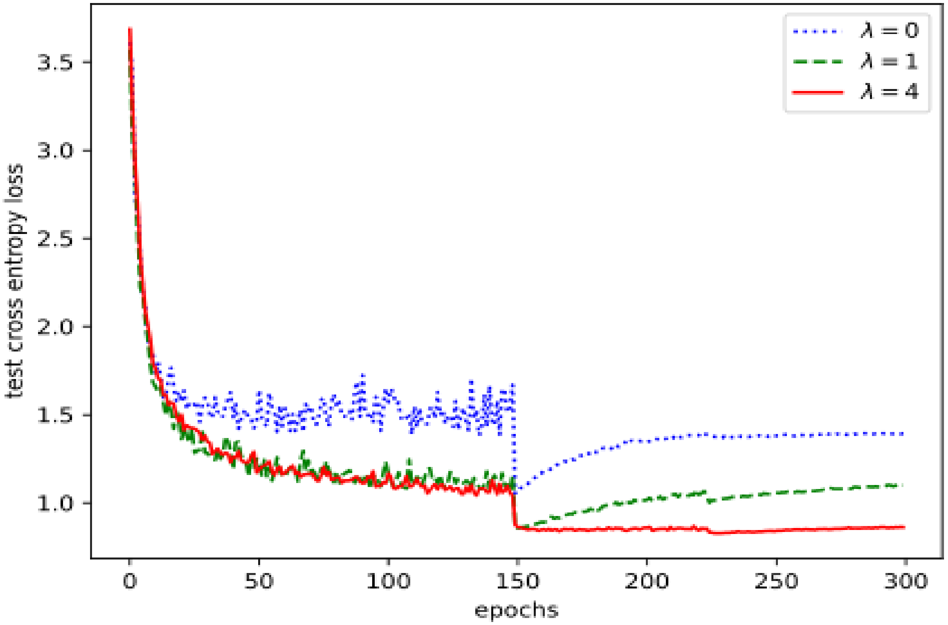
$$\mathcal{L} = \mathcal{L}_{cross} + \lambda \mathcal{L}_{share}$$

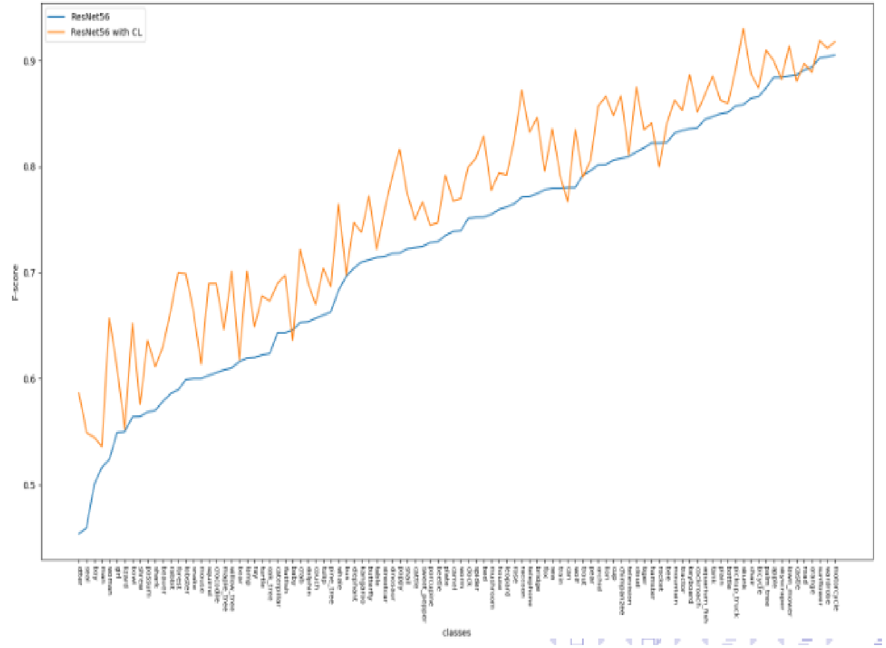
- Cross Entropy Loss: $\mathcal{L}_{cross} = \frac{\sum_i \hat{y}_i \log y_{1,i}}{N} + \frac{\sum_i \hat{y}_i \log y_{2,i}}{N}$
- $\lambda$ is a hyper parameter (coefficent) to balance the losses



# 3. Learning Base CNNs for Each Task Group

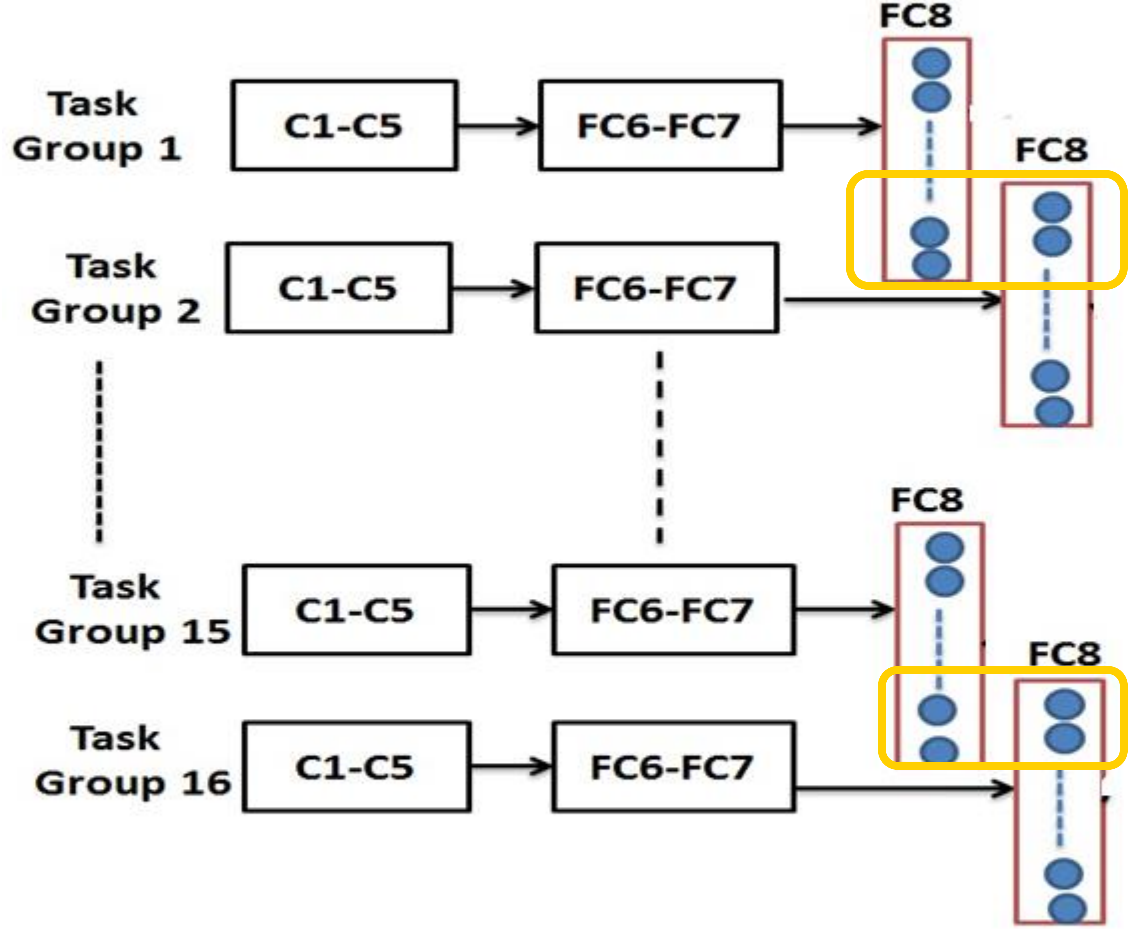
- **Deep Collaborative Learning**



# 3. Learning Base CNNs for Each Task Group

- **Deep Collaborative Learning**




$J_{ind}$

$J_{com}$

$J_{diff}$

## 3. Learning Base CNNs for Each Task Group

- **Deep Collaborative Learning**

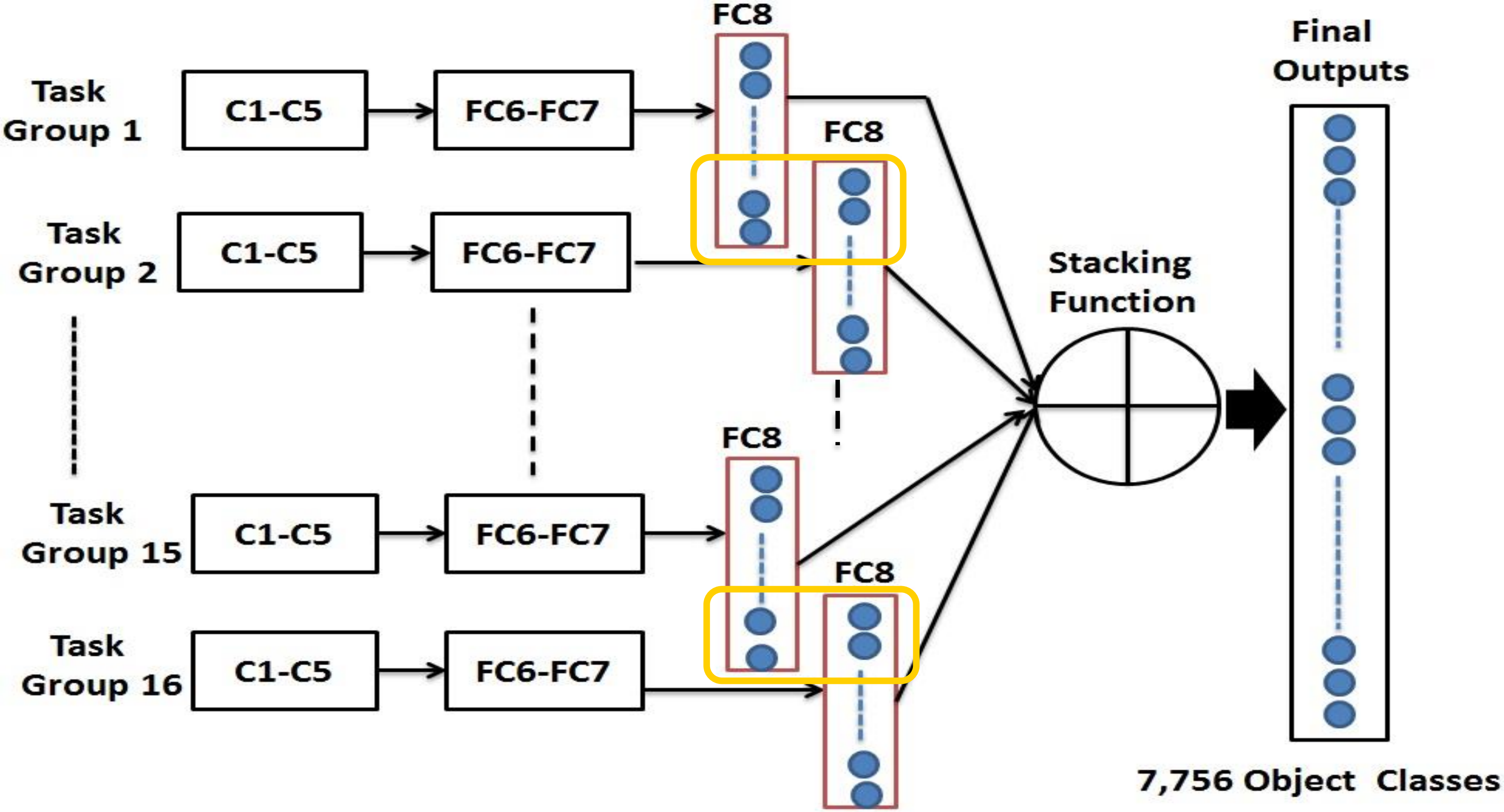
$$J_{toal} = J_{ind} + \alpha J_{com} + \beta J_{diff}$$

$$J_{ind} = \sum_{i=1}^{\tau} \sum_{j=1}^{N} \delta(y_i^j, \bar{y}_i^j)$$

$$J_{com} = \sum_{i,j \in common\ tasks} KL(P_i|$$

$$J_{diff} = \sum_{i,j \in non-overlapped\ tasks}$$

# 4. Deep Mixture of Diverse Experts



# 4. Deep Mixture of Diverse Experts

- Mixture Approaches
  - Stacking function
  - Gating network
  - Hierarchical Deep Mixture

# 4. Deep Mixture of Diverse Experts

- **Three factors for Output Integration**
  - **Prediction Scores**

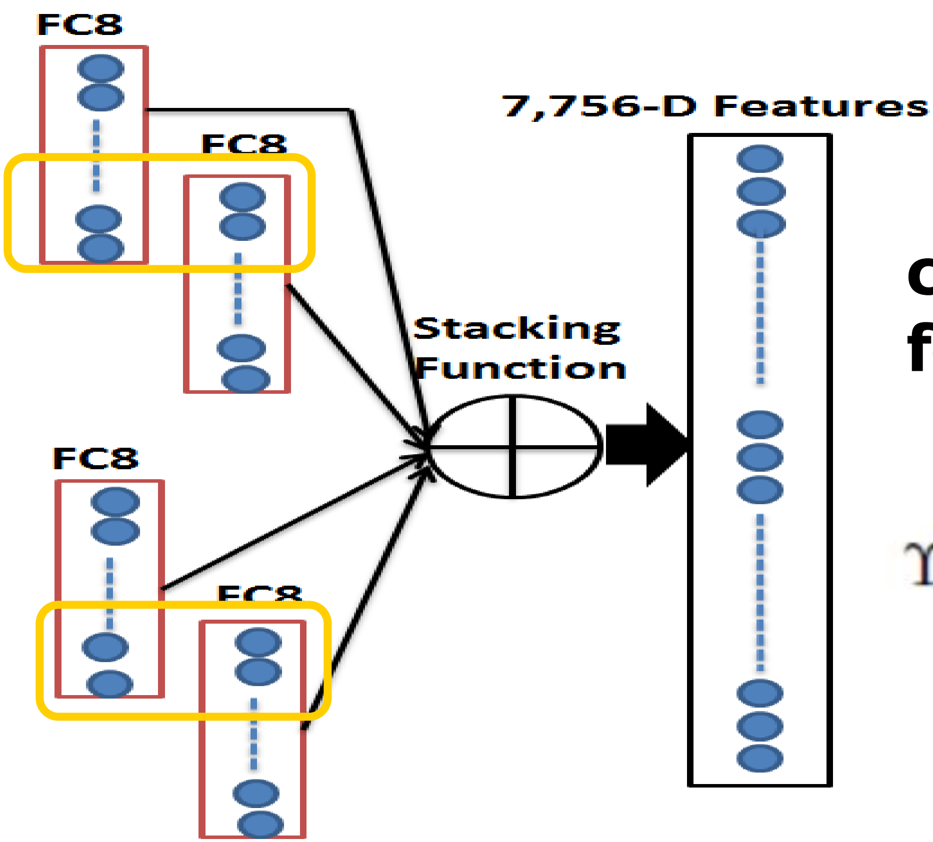
    ---each base deep CNNs will provide their individual predictions for each sample!
  - **Inter-Group Conflict---score for ``not-in-group"**

    ---the predictions from different groups may conflict
  - **Inter-Group Overlapping**

    ---more inter-group overlapping may provide more comparable results

# 4. Deep Mixture of Diverse Experts

- **Output Integration from Diverse Experts**



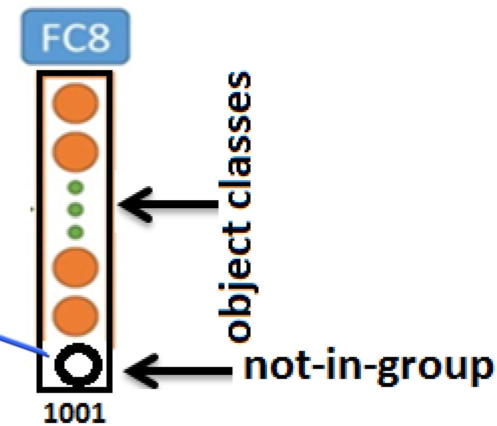

**cumulative prediction score for *ith* object class:**

$$\Upsilon(c_i) = \sum_{j=1}^{\vartheta} \Lambda_j(c_i) p_j(c_i) \frac{(1-\phi_j)}{\phi_j}$$

# 4. Deep Mixture of Diverse Experts

- **Output Integration from Diverse Experts**

$$\Upsilon(c_i) = \sum_{j=1}^{\vartheta} \Lambda_j(c_i) p_j(c_i) \frac{(1-\phi_j)}{\phi_j}$$




$$\Lambda_j(c_i) = \begin{cases} 1, & \text{if } c_i \text{ is in the jth task group} \\ \lambda, & \text{otherwise} \end{cases}$$

$$p_j(i) = \frac{exp(W_i^T x + b)}{\sum_{l=1}^{M} exp(W_l^T x + b)} \qquad 0 \leq p_j(i) \leq 1$$

## 4. Deep Mixture of Diverse Experts

- **Wish Lists:**
  - **Classes with similar learning complexities can be learned together! ---ontology-driven task assignment for group generation!**
  - **The predictions from multiple base deep CNNs are comparable at certain level! ---inter-group overlapping & not-in-group!**
  - **The conflicts on inter-group predictions are known or identifiable! ---not-in-group!**

# 4. Deep Mixture of Diverse Experts

- **Another Wish List:**
  - **The special category ``not-in-group" in each task group may suffer from the problem of ``huge sample imbalance"!**
  - **hard object classes may need more training samples, using the same number of training samples does not make sense to them!**

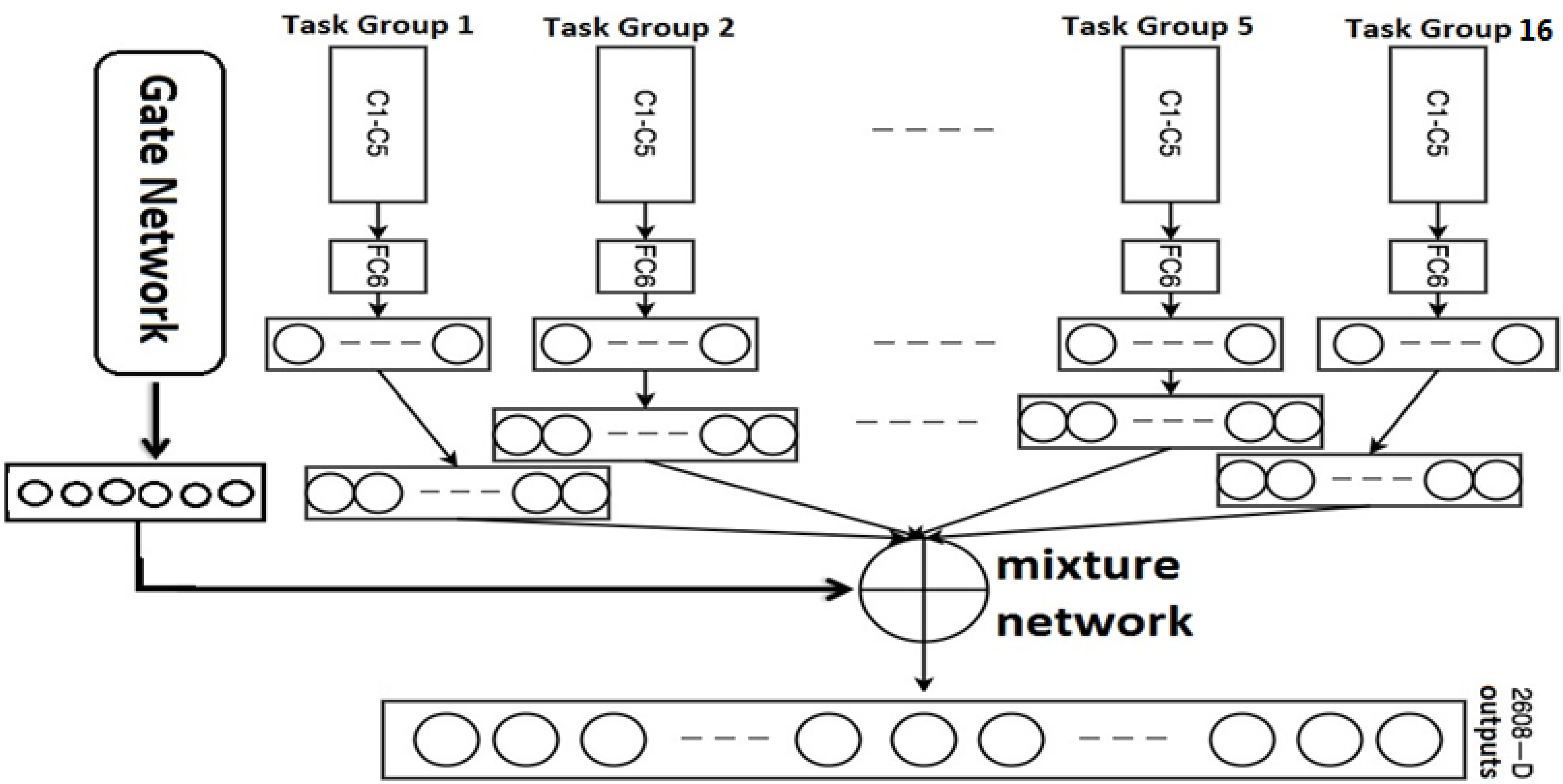
**Future Research: Imbalance Deep Learning**

# 4. Deep Mixture of Diverse Experts

- **Another Wish List:**
  - The difference between the highest score and the second one?
  - Cost-sensitive classifier training?

# 4. Deep Mixture of Diverse Experts

- ## Gate Network for Deep Mixture



# 4. Deep Mixture of Diverse Experts

- **Gate Network for Deep Mixture**

(a) $\tau$ base deep CNNs $\{f_1(x), \cdots, f_t(x), \cdots, f_\tau(x)$

(b) a $\tau$-D gate network $\vartheta = \{\phi_1, \cdots, \phi_t, \cdots, \phi_\tau\}$

$$\mathbb{F}(x) = \sum_{t=1}^{\tau} \phi_t^T f_t(x), \qquad \sum_{t=1}^{\tau} \phi_t^T \phi_t = 1$$

$\phi_t = \left[\phi_t^1, \cdots, \phi_t^j, \cdots, \phi_t^M\right]$ is the confidence score for the $t$-th base deep CNNs $f_t(x)$ and $\phi_t^j$ is the confidence score for identifying the $j$-th plant species in the $t$-th task group $\Omega_t$, $f_t(x) = \left[f_t^1(x), \cdots, f_t^j(x), \cdots, f_t^M(x)\right]$ denotes the $t$-th base deep CNNs with $M$ outputs

# 4. Deep Mixture of Diverse Experts

- **Gate Network for Deep Mixture**

$$\min_{\phi, W, \vartheta} \quad \mathcal{L}(\mathcal{D}) = \sum_{t=1}^{\mathcal{T}} \pounds(W_t) + \sum_{t=1}^{\mathcal{T}} \sum_{h=1}^{\mathcal{T}} \ell(\phi_t, \phi_h)$$
$$+ \sum_{l=1}^{R} \sum_{j=1}^{N} \alpha \max\left(P_{opt}(x_j^l, c_j) - P_{opt}(x_j^l, y_j^l) + \beta, 0\right)$$

where $\phi_t$ and $\phi_h$ are used to indicate the confidence scores for the $t$-th and $h$-th base deep CNNs $f_t(x)$ and $f_h(x)$, $\ell(\cdot)$ is the loss function to emphasize the confidence consistency among the predictions from two base deep CNNs $f_t(x)$ and $f_h(x)$ when they share some common plant species because of inter-group task overlapping, $P_{opt}(x_j^l, c_j)$ is the prediction probability for the plant image $x_j^l$ to be identified as the species $c_j$ and it is aggregated over $\tau$ base deep CNNs, $\beta$ is a hyper-parameter to denote the confidence margin, $\alpha$ is a hyper-parameter that is used to make trade-off for the importance of the margin-based loss.

# 4. Deep Mixture of Diverse Experts

- **Gate Network for Deep Mixture**

$$\phi_t^j = \frac{1}{R}\sum_{l=1}^{R} I\{y_j^l, c_j\}\frac{exp(W_{t_j}^T x_j^l + b)}{\sum_{i=1}^{M} exp(W_{t_i}^T x_i^l + b)}$$

$$\phi_h^j = \frac{1}{R}\sum_{l=1}^{R} I\{y_j^l, c_j\}\frac{exp(W_{h_j}^T x_j^l + b)}{\sum_{k=1}^{M} exp(W_{h_k}^T x_k^l + b)}$$

$$\ell(\phi_t, \phi_h) = \sum_{c_j \in \Omega_t \cap \Omega_h} H(\phi_t^j, \phi_h^j)$$

$$P_{opt}(x_j^l, c_j) = \sum_{t=1}^{\mathcal{T}} I\{y_j^l, c_j\}\phi_t^T f_t(x_j^l), \quad P_{opt}(x_j^l, y_j^l) = \sum_{t=1}^{\mathcal{T}} I\{y_j^l, 1 - c_j\}\phi_t^T f_t(x_j^l)$$

# 4. Deep Mixture of Diverse Experts

## ▪ Gate Network for Deep Mixture

$$\min_{\phi, W, \vartheta} \quad \mathcal{L}(\mathcal{D}) = \sum_{t=1}^{\mathcal{T}} \pounds(W_t) + \sum_{t=1}^{\mathcal{T}} \sum_{h=1}^{\mathcal{T}} \ell(\phi_t, \phi_h)$$

$$+ \sum_{l=1}^{R} \sum_{j=1}^{N} \alpha \max\Big(P_{opt}(x_j^l, c_j) - P_{opt}(x_j^l, y_j^l) + \beta, 0\Big)$$

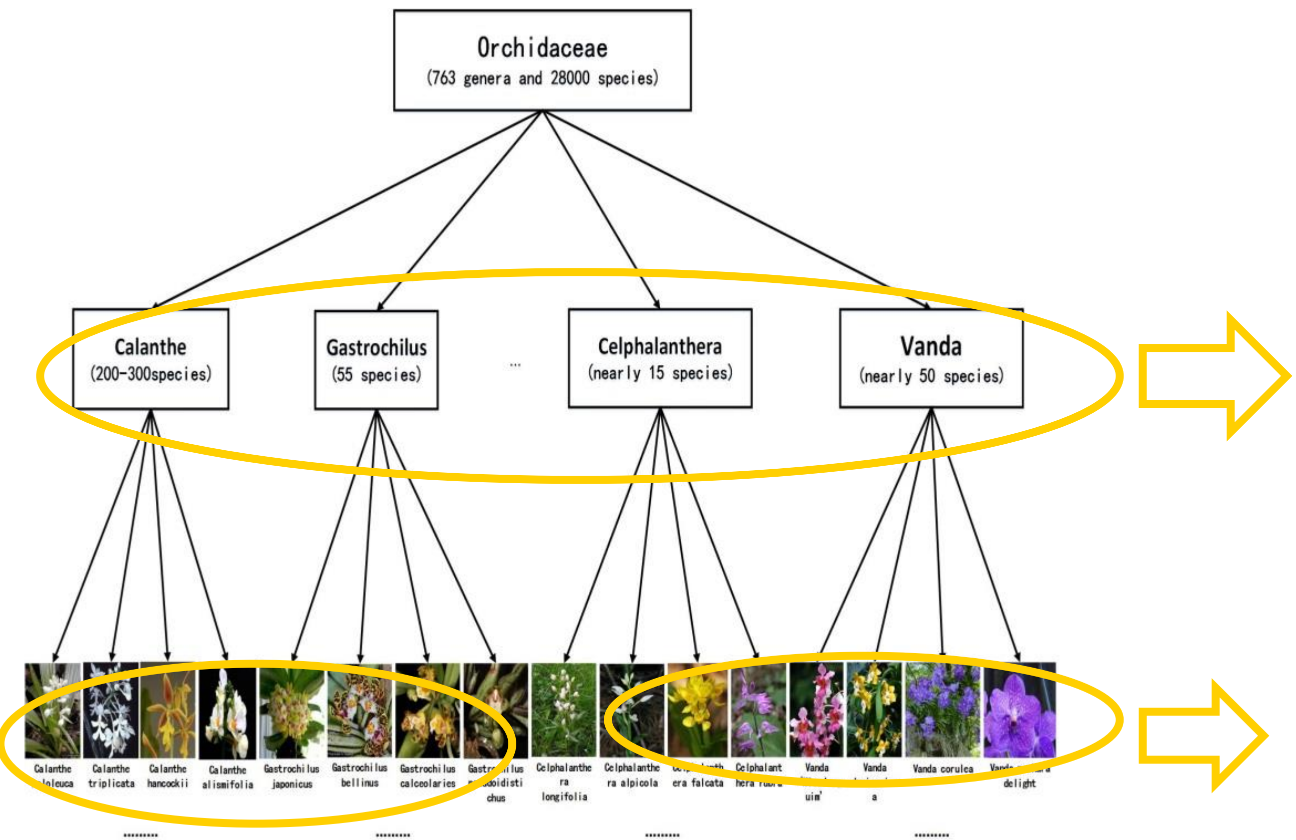
(a) The ***first part*** is the loss as defined in Eq. (4), which aims to minimize the loss of the relevant base deep CNNs.

(b) The ***second part*** is the gate network loss to emphasize that: (1) for the same plant image $x_j^l$, its best-matched plant species $c_j$ can be identified correctly

(c) The ***third part*** aims to address the overconfidence issue and guarantees that the best-matched plant species for each image has higher probability than others in a reasonable margin $\beta$, e.g., $P_{opt}(x_j^l, y_j^l) - P_{opt}(x_j^l, c_j) \geq \beta$.

# 4. Deep Mixture of Diverse Experts

- **Hierarchical Deep Mixture over Ontology**

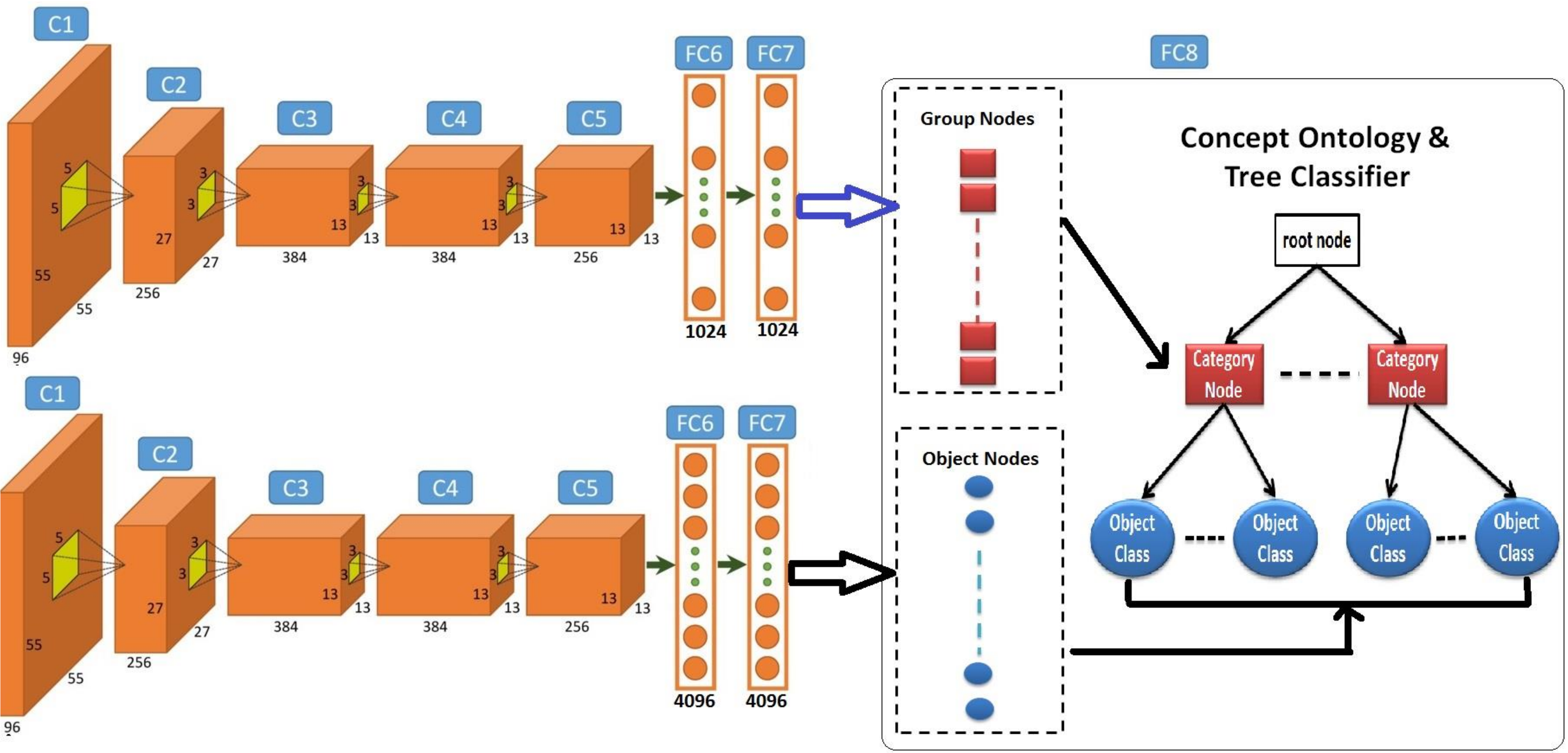



**Deep Network for High-level Categories**

**Deep Networks for Low-level Categories**

# 4. Deep Mixture of Diverse Experts

- **Hierarchical Deep Mixture over Ontology**



# 4. Deep Mixture of Diverse Experts

- Hierarchical Deep Mixture over Ontology
  - Benefits
    - Less storage memory because of less parameters, good sample balance, less test cost， ….
  - Shortages
    - Inter-level error propagation

# 4. Deep Mixture of Diverse Experts

- **Knowledge Distillation for Smartphone Application**

outputs

**Mixture Network (Teacher)**

**Knowledge Distilling**

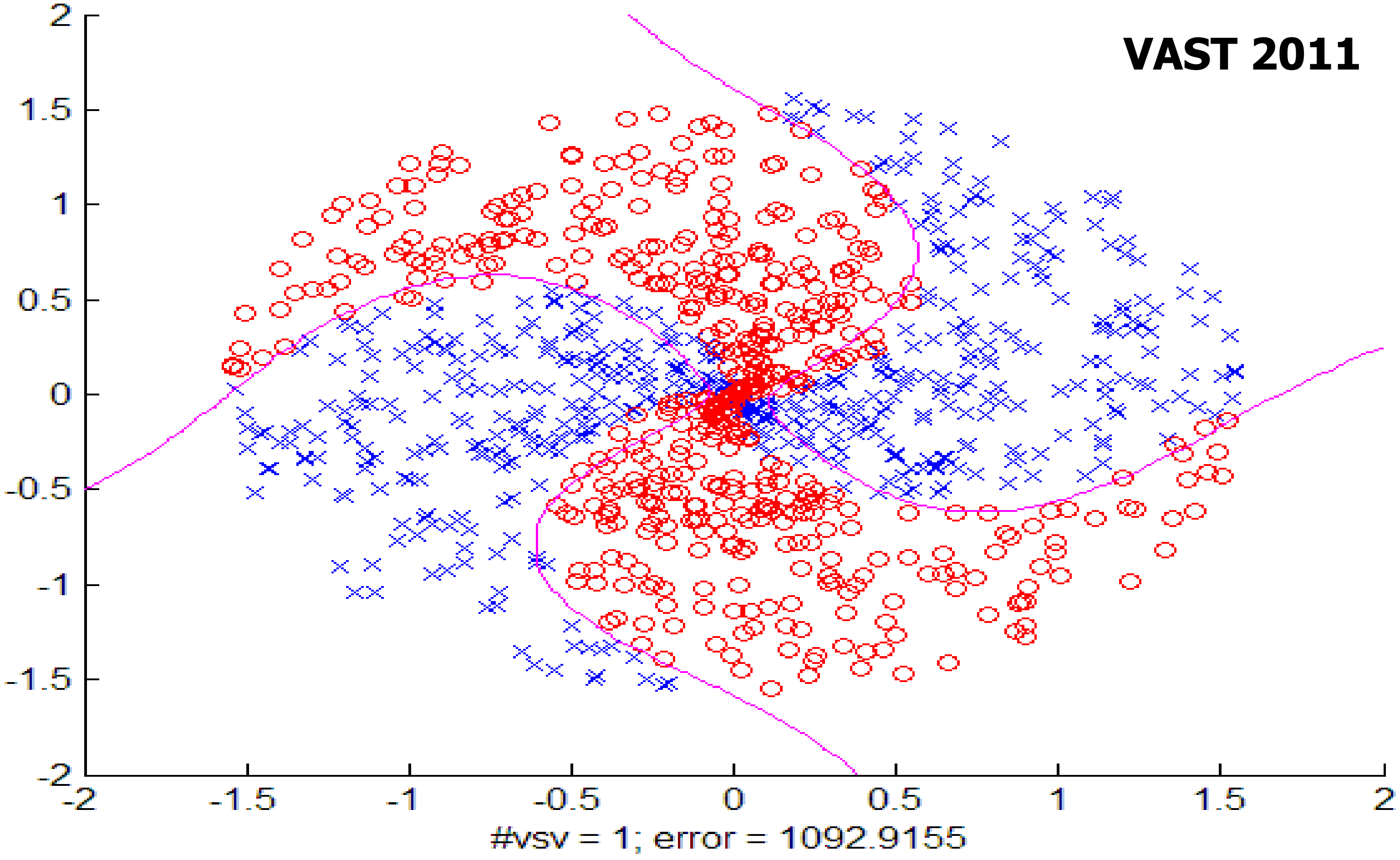
$$KL(P_T|P_S) \longrightarrow \mathbf{f}(x) \approx F_T(x)$$

**Student Network**

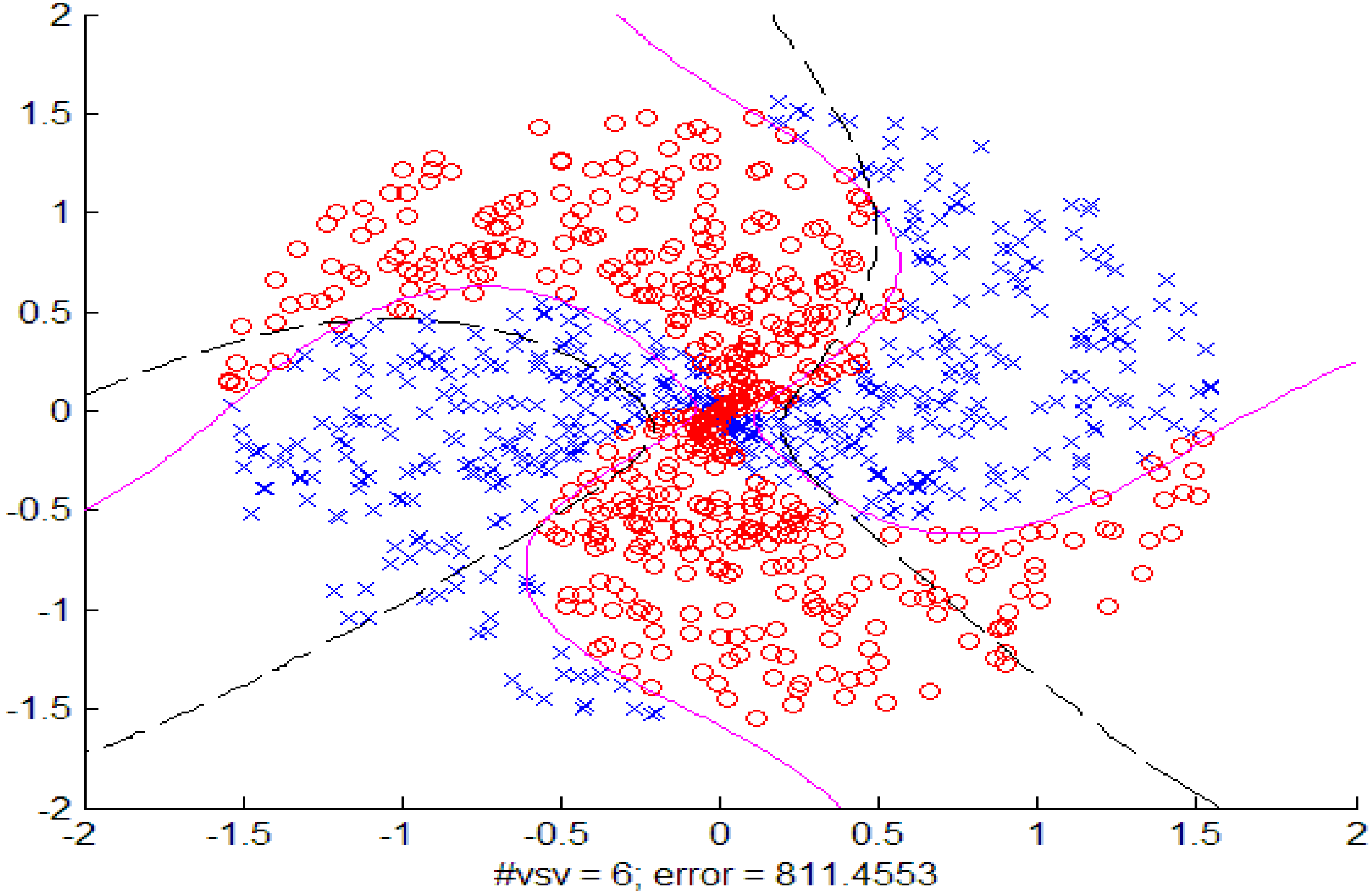
outputs

**Small-size student network with similar performance**

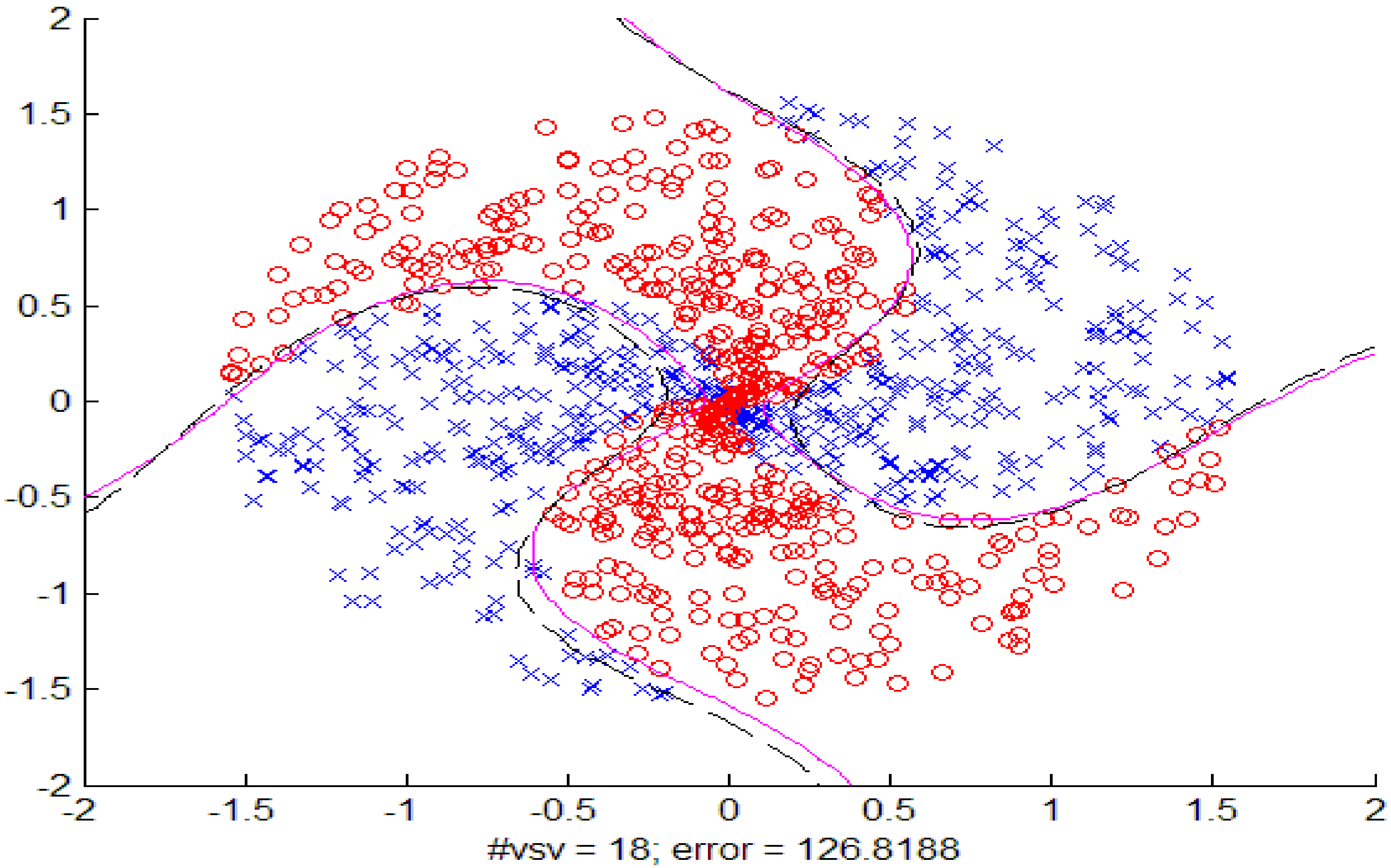
# 5. Interactive Classifier Assessment



# 5. Interactive Classifier Assessment



# 5. Interactive Classifier Assessment



# 6. Experimental Results

- **Image Sets for Algorithm Evaluation**
  - **ImageNet with 1000 atomic object classes**
  - **ImageNet10K with 10184 categories**

# 6. Experimental Results

- **Components for Evaluation**

  - **Deep mixture of diverse experts**
  - **Deep multi-task learning**
  - **Task assignment for group generation**

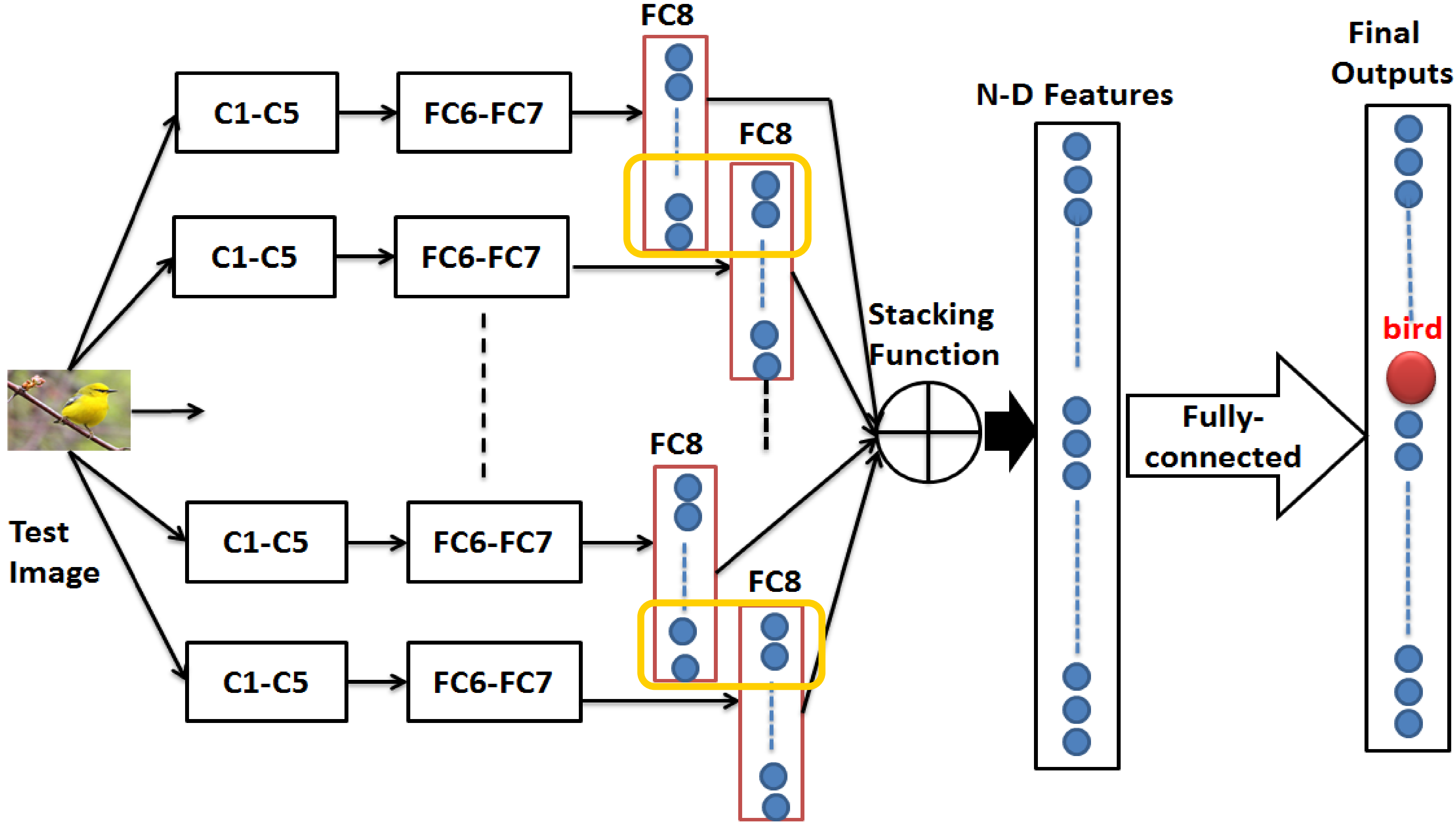
# 6. Experimental Results

- **ImageNet10K**

  - **7756** atomic object classes (at the leaf nodes of the concept ontology) are identified in ImageNet10k image set, **2428** high-level image concepts (at the non-leaf nodes of the concept ontology) are identified.

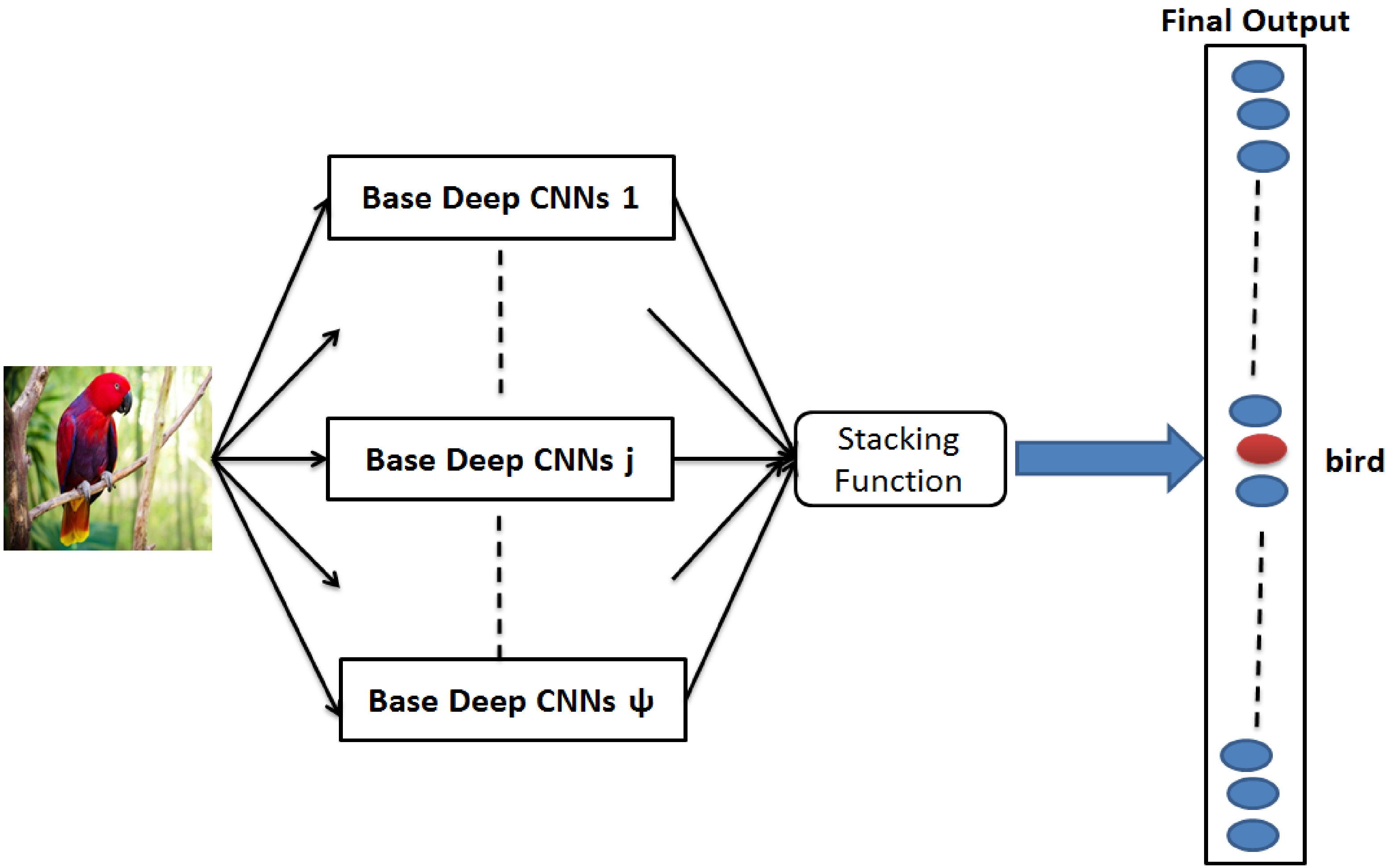
# 6. Experimental Results

- **Testing**



# 6. Experimental Results

## ▪ Testing



# 6. Experimental Results

## Testing

- **Output Integration from Diverse Experts**

$$\Upsilon(c_i) = \sum_{j=1}^{\vartheta} \Lambda_j(c_i) p_j(c_i) \frac{(1-\phi_j)}{\phi_j}$$

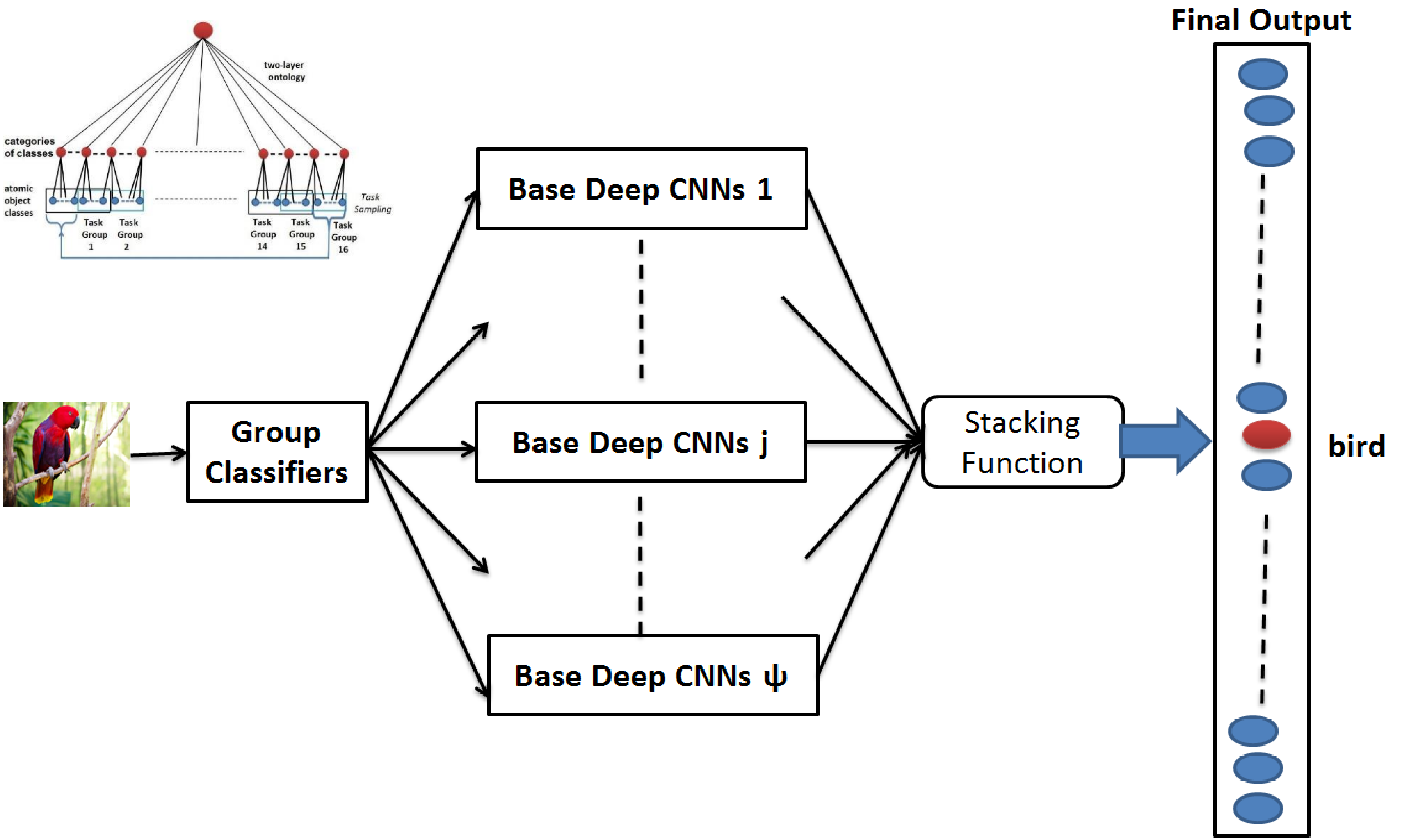
$$\Lambda_j(c_i) = \begin{cases} 1, & \text{If } \varnothing_j < T \\ \lambda, & \text{otherwise} \end{cases}$$

**It is different from training time**

$$p_j(i) = \frac{exp(W_i^T x + b)}{\sum_{l=1}^{M} exp(W_l^T x + b)} \qquad 0 \le p_j(i) \le 1$$

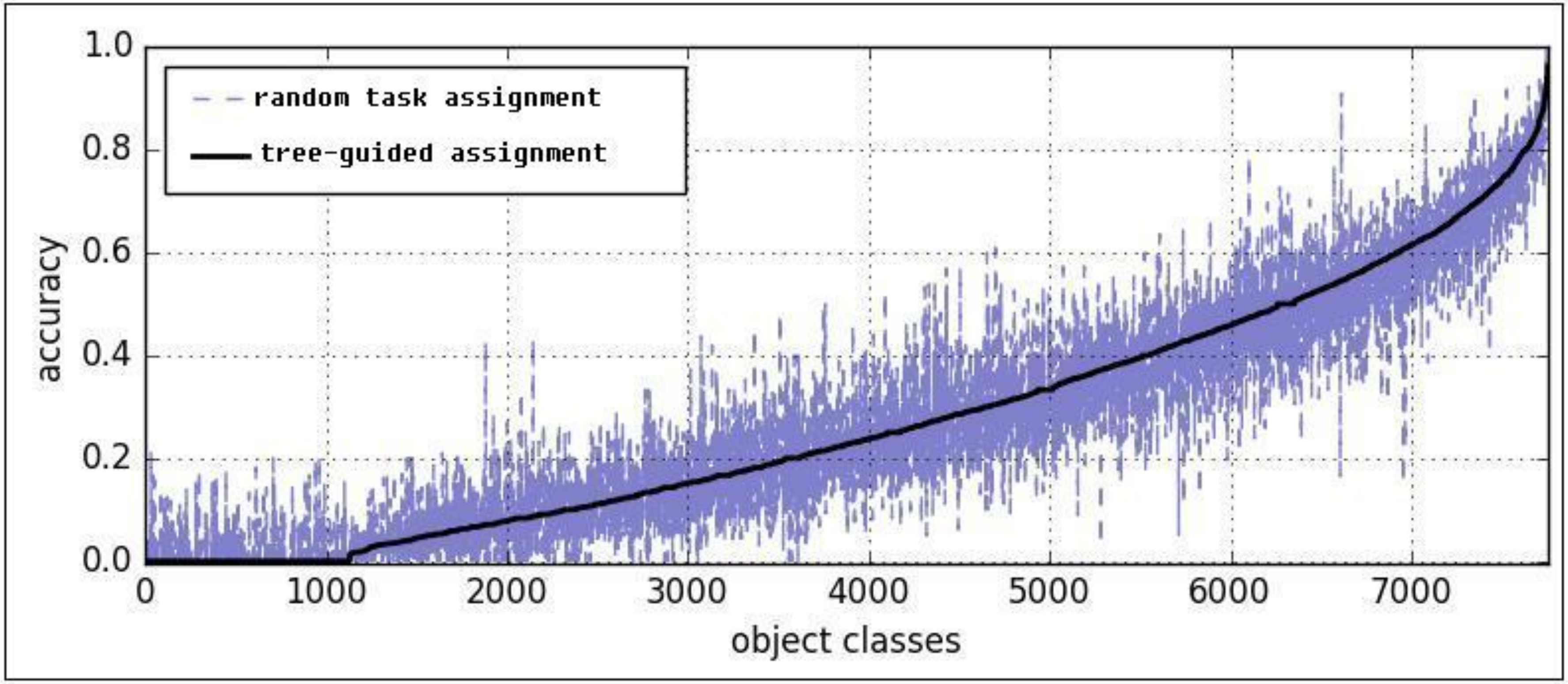
# 6. Experimental Results

- **Testing: when tree classifier is learned**



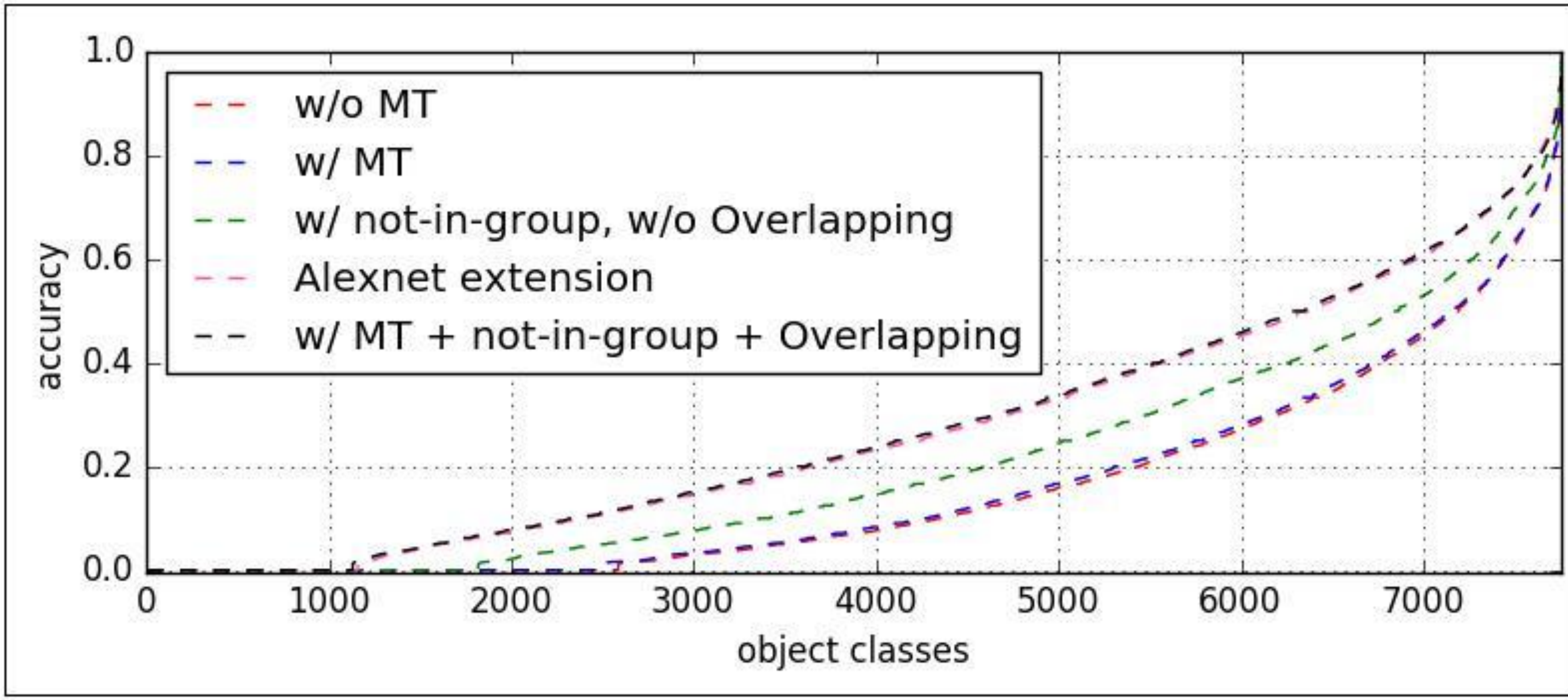
# 6. Experimental Results

- **Effects of Ontology-Driven Task Assignment**



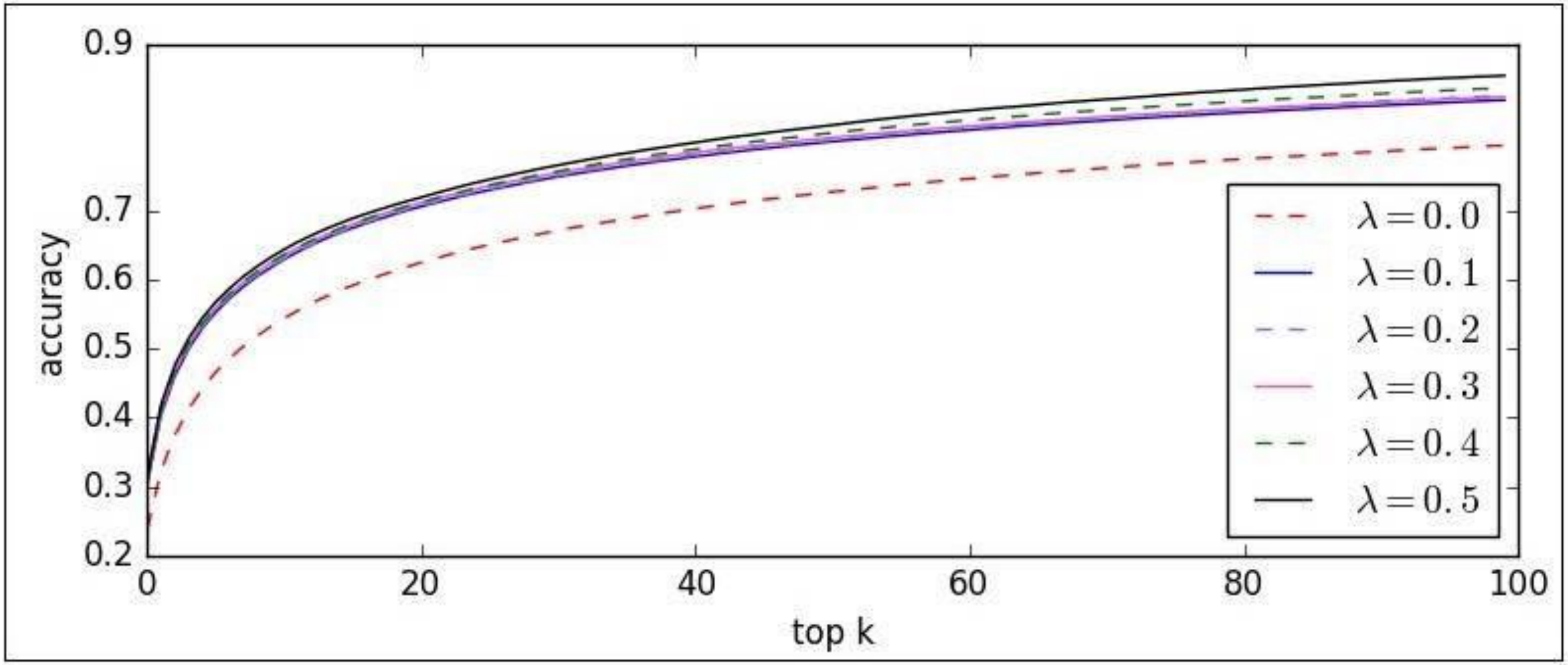
# 6. Experimental Results

- **Effects of Deep Multi-Task Learning**



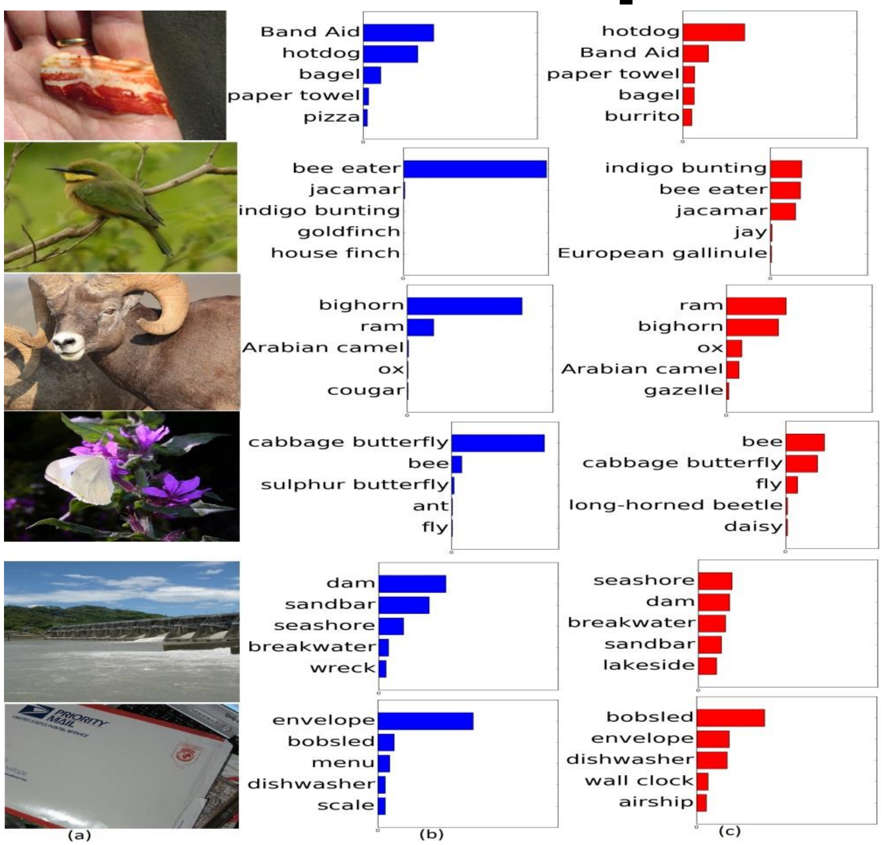
# 6. Experimental Results

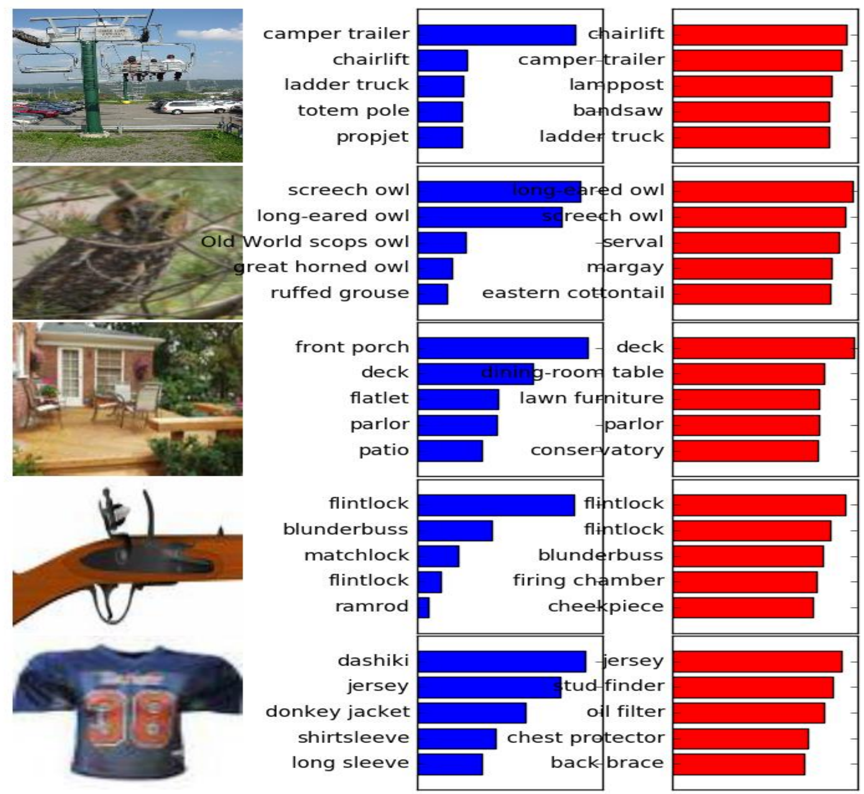
- **Effects of Inter-Group Overlapping**



# 6. Experimental Results

- **Effects of Deep Mixture**



# 6. Experimental Results

- **Effects of Deep Mixture**

(a) (b) (c)

# 6. Experimental Results

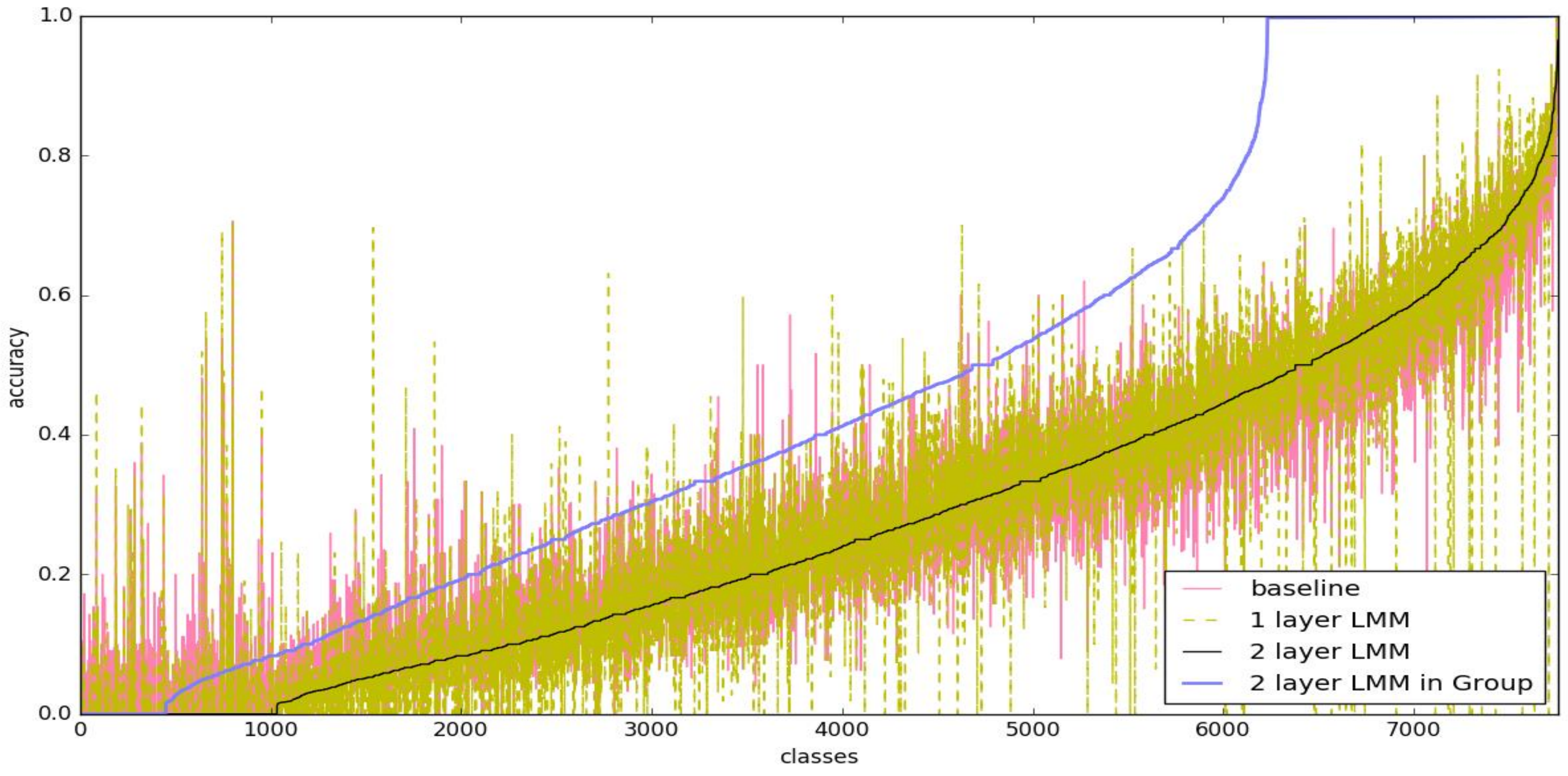
- **Effects of Deep Mixture**

TABLE I
**The comparisons on the average accuracy rates.**

| approaches | accuracy rate (top k) | | |
|---|---|---|---|
| | 1 | 5 | 10 |
| our deep mixture algorithm | **38.65%** | **55.41%** | **64.32%** |
| AlexNet Extension | 31.70% | 46.23% | 52.18% |
| random assignment | 34.53% | 47.39% | 53.25% |
| visual tree | 37.55% | 53.29% | 62.02% |
| Stack 2 | 37.63% | 54.37% | 63.29% |

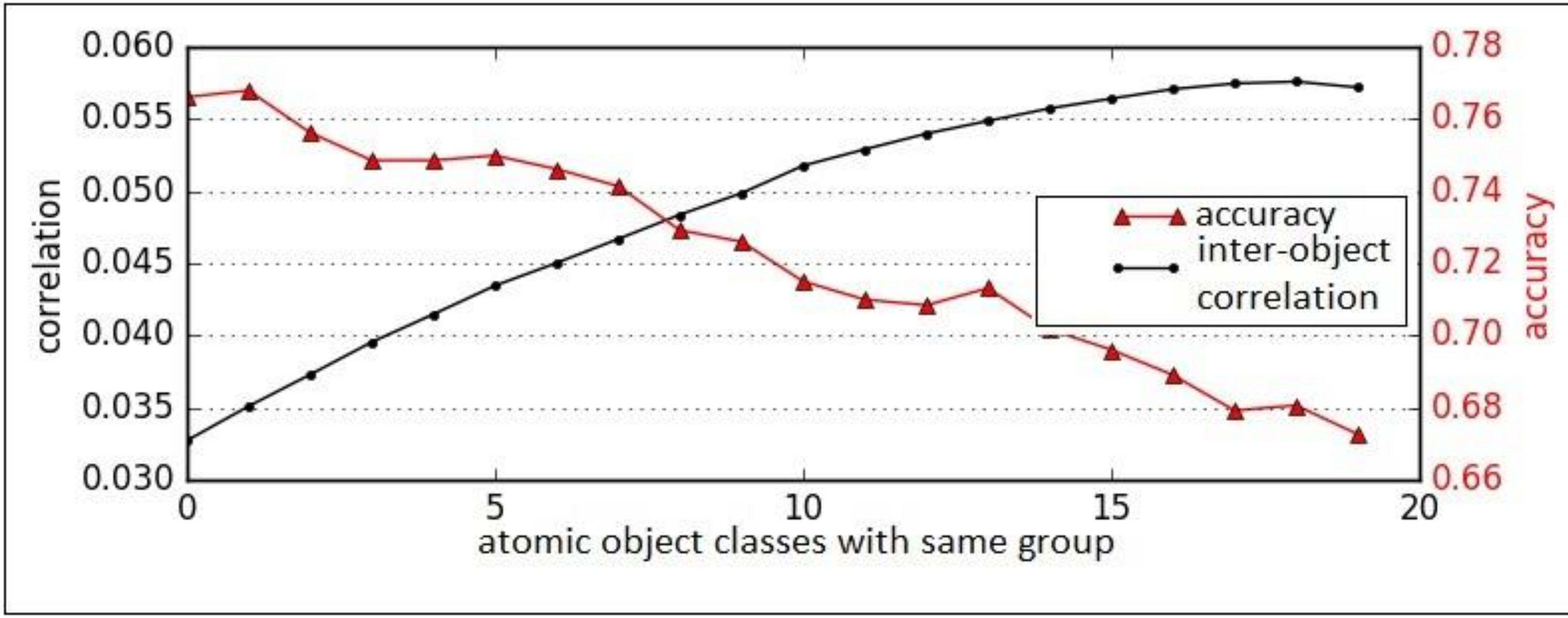
# 6. Experimental Results

- **Effects of Deep Mixture**



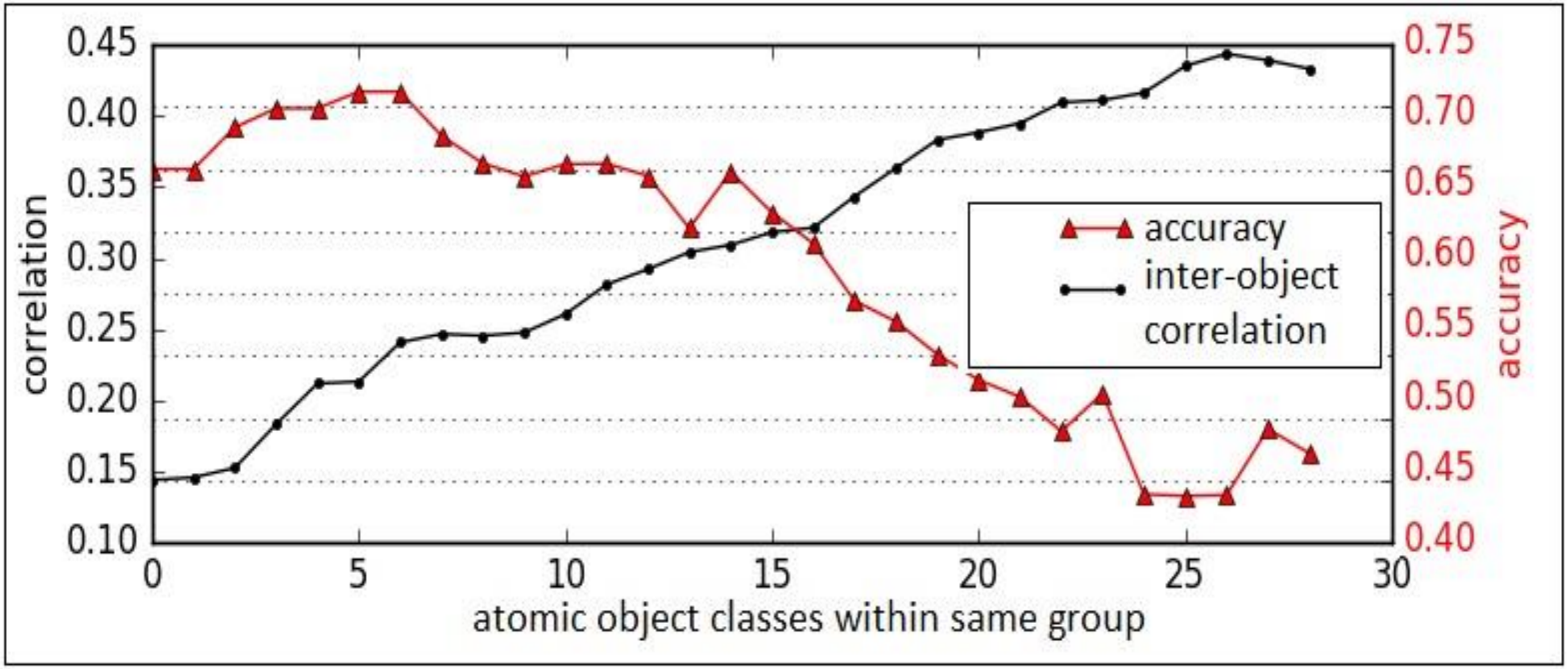
# 6. Experimental Results

- **Impacts of Inter-Task Relationships**



## 6. Experimental Results

- **Impacts of Inter-Task Relationships**



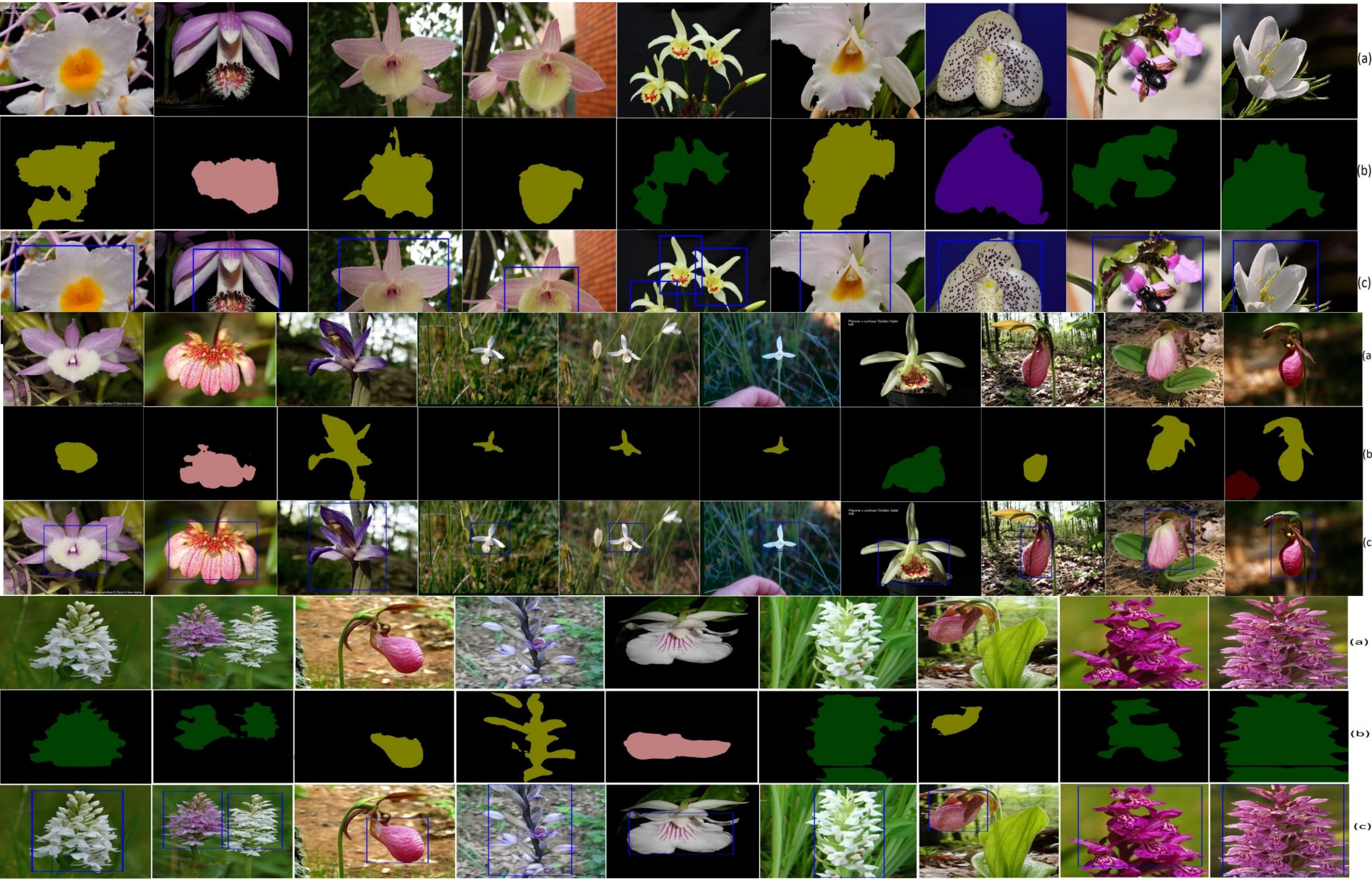
## 6. Experimental Results

- **Late fusion *vs.* Early fusion**

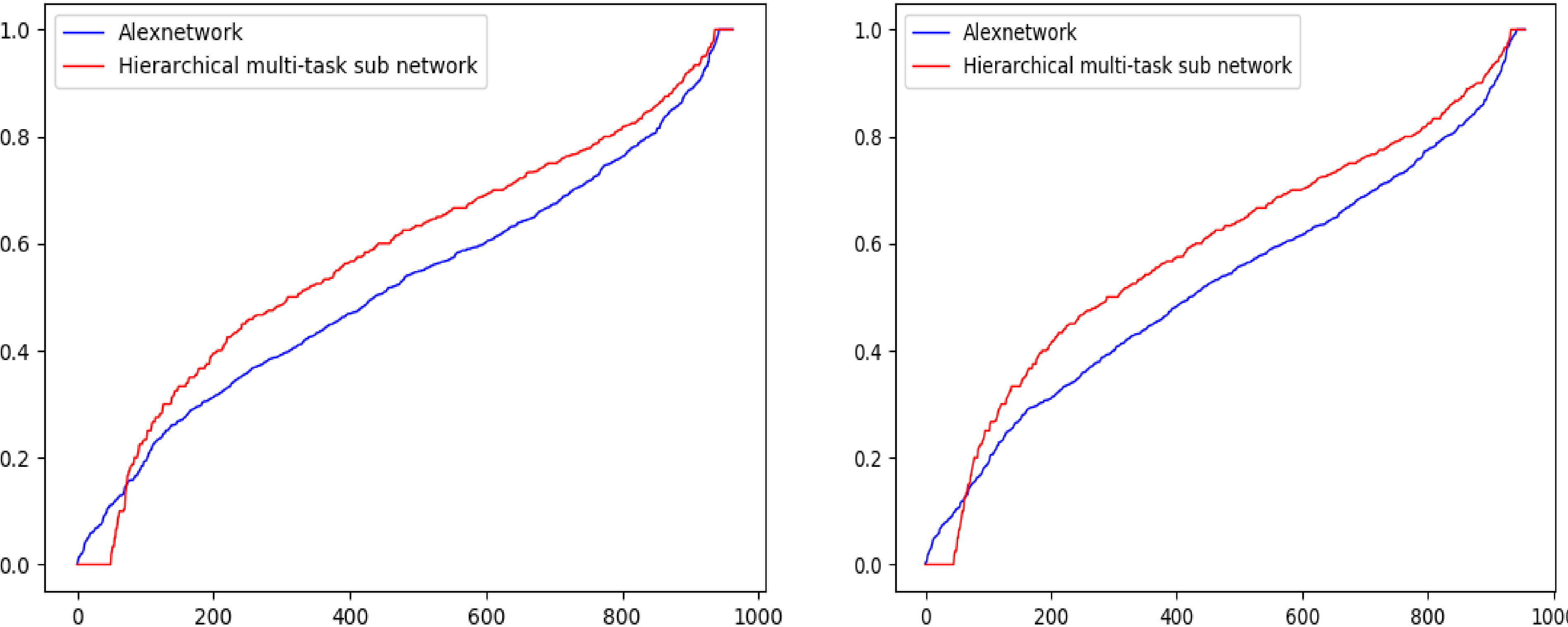
The comparison on the average accuracy rates between late fusion and early fusion.

| approaches | accuracy rate (top k) | | |
|---|---|---|---|
| | 1 | 5 | 10 |
| late fusion | **38.65%** | **55.41%** | **64.32%** |
| early fusion | 36.23% | 52.45% | 61.38% |

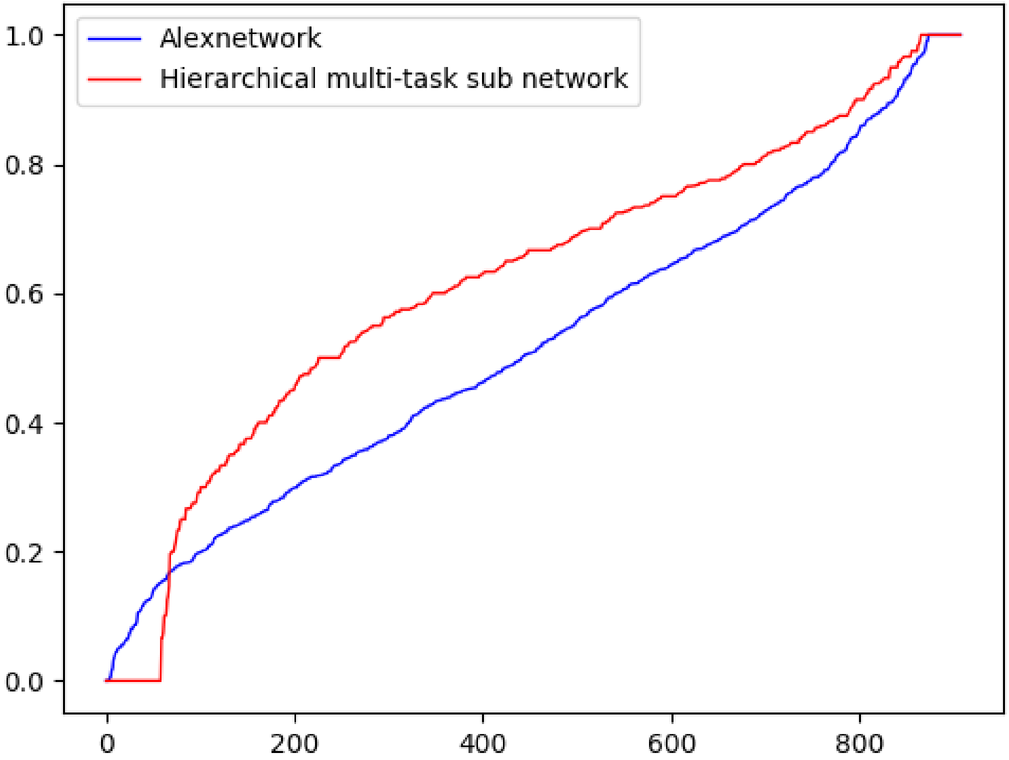
# **Applications**: Large-Scale Plant Species Recognition

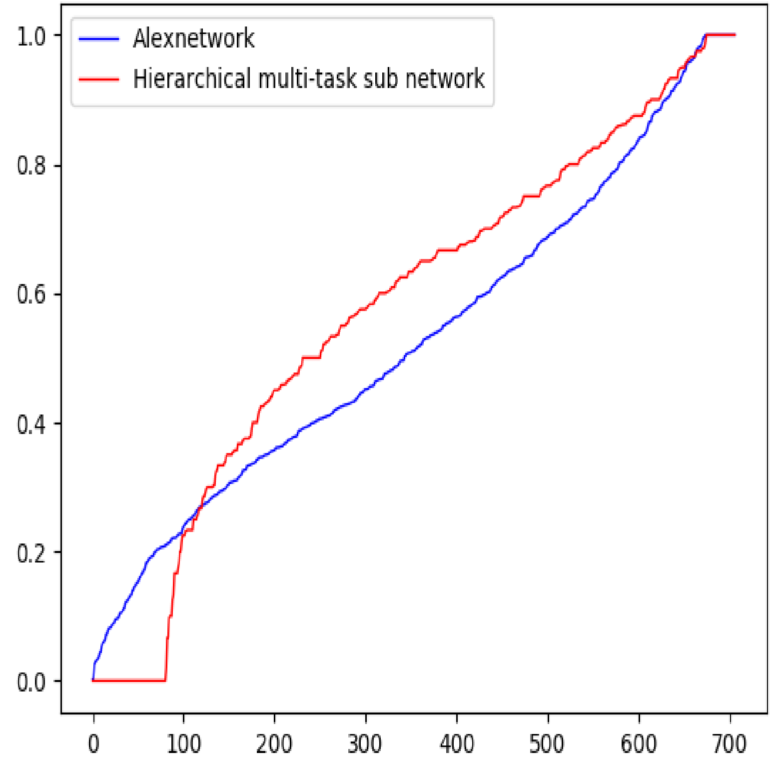


# **Applications**: Large-Scale Plant Species Recognition

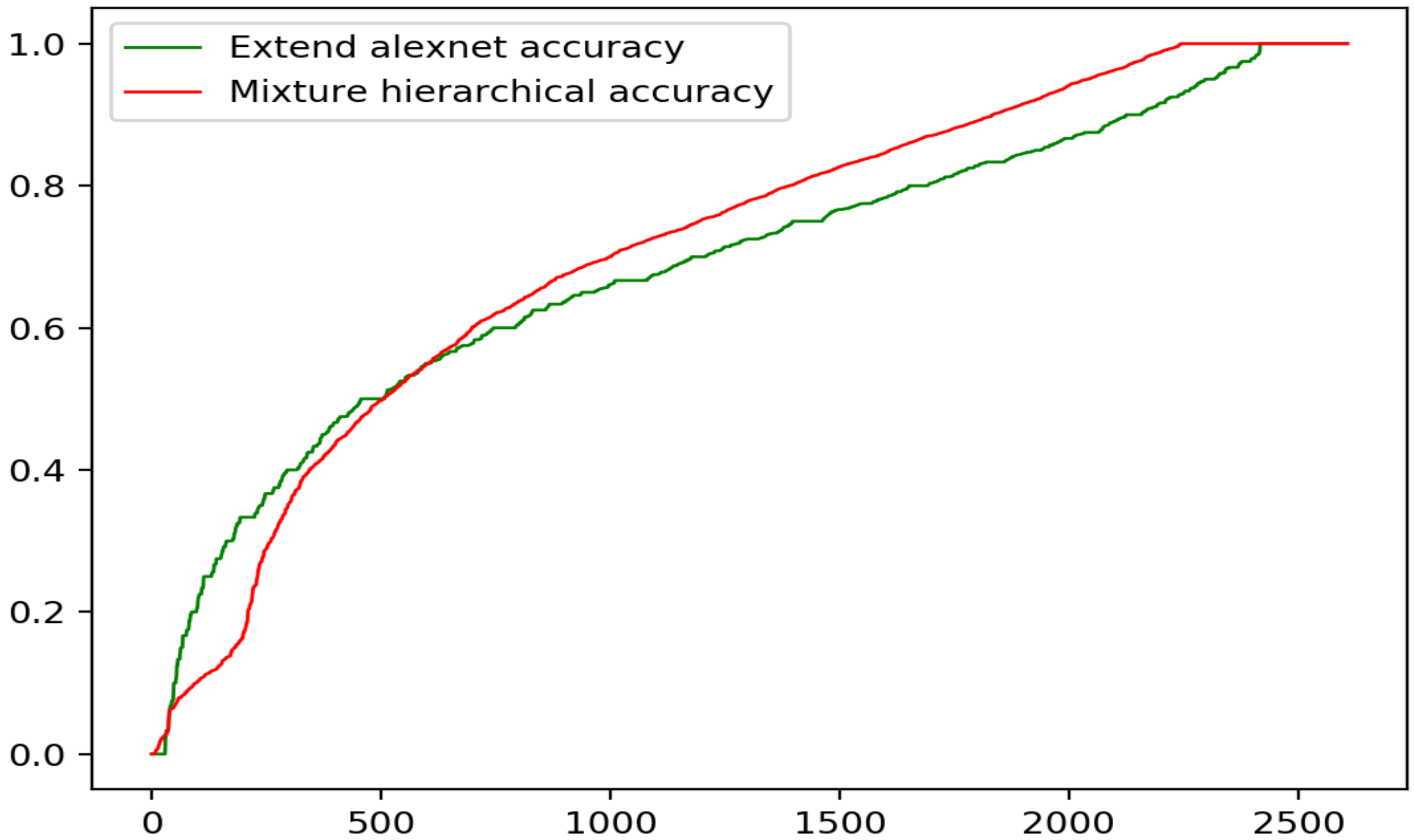


# **Applications**: Large-Scale Plant Species Recognition



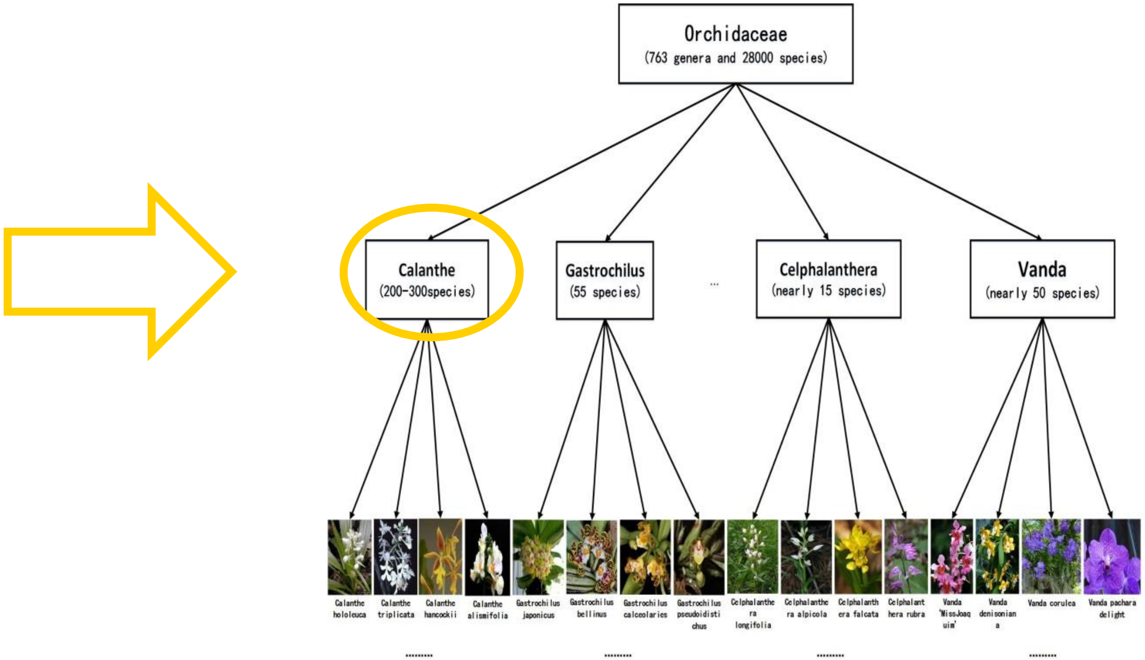



# **Applications**: Large-Scale Plant Species Recognition



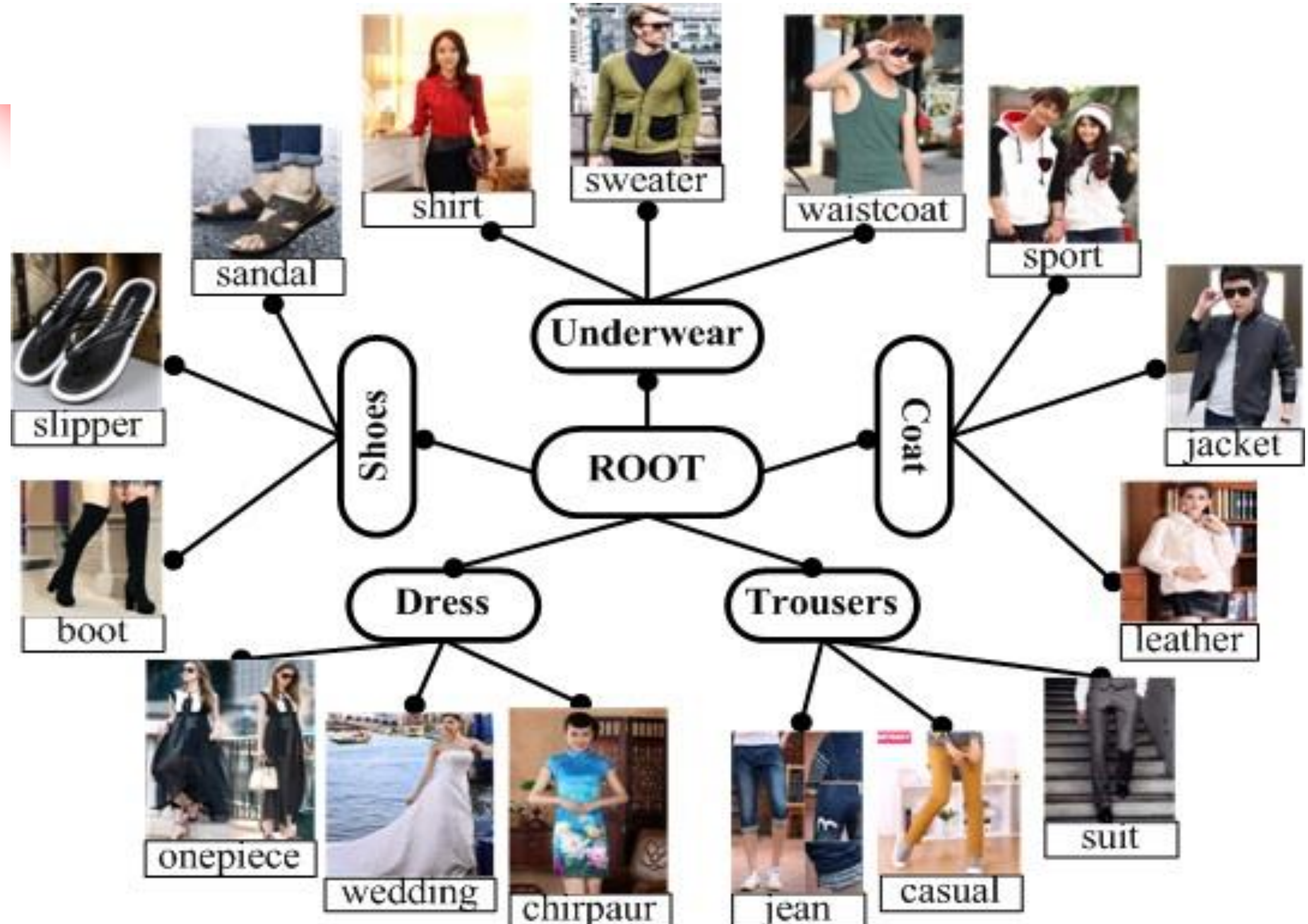
# Why we use concept ontology?

- **Early Stop**

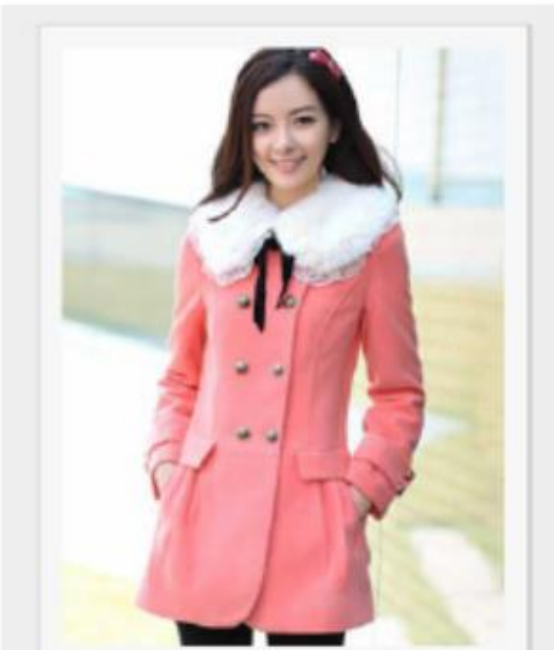



- **Semantic Interpretation**

# **Applications**: Fashion Recognition and Search

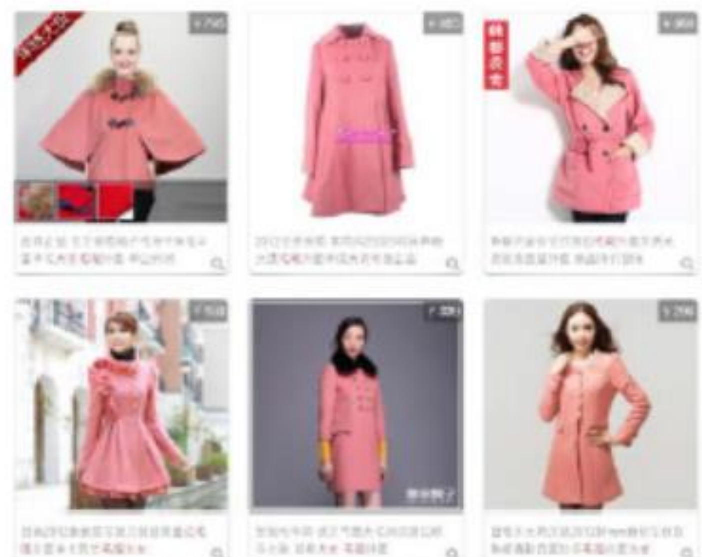


# **Applications**: Fashion Recognition and Search

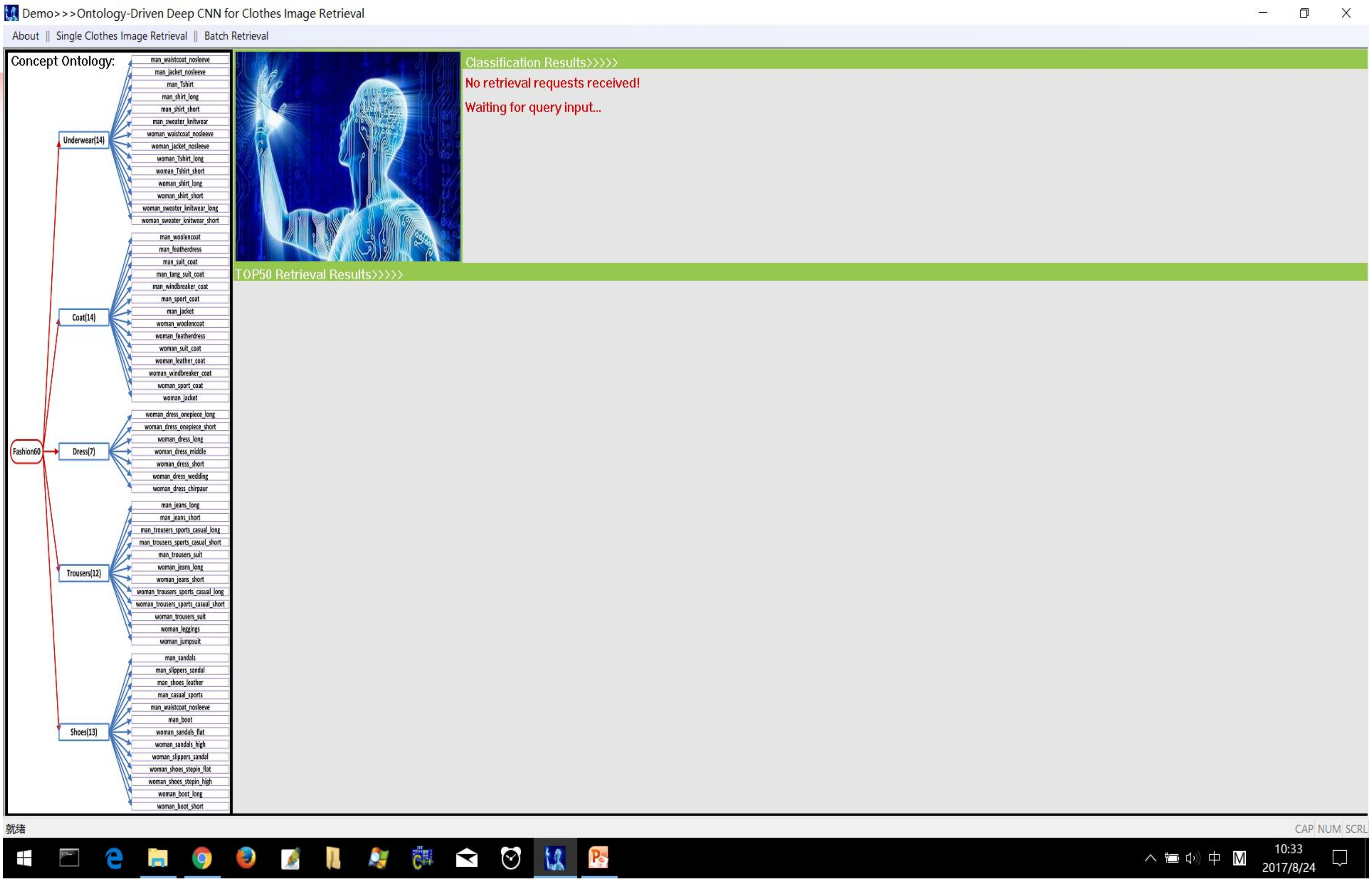


similarity search

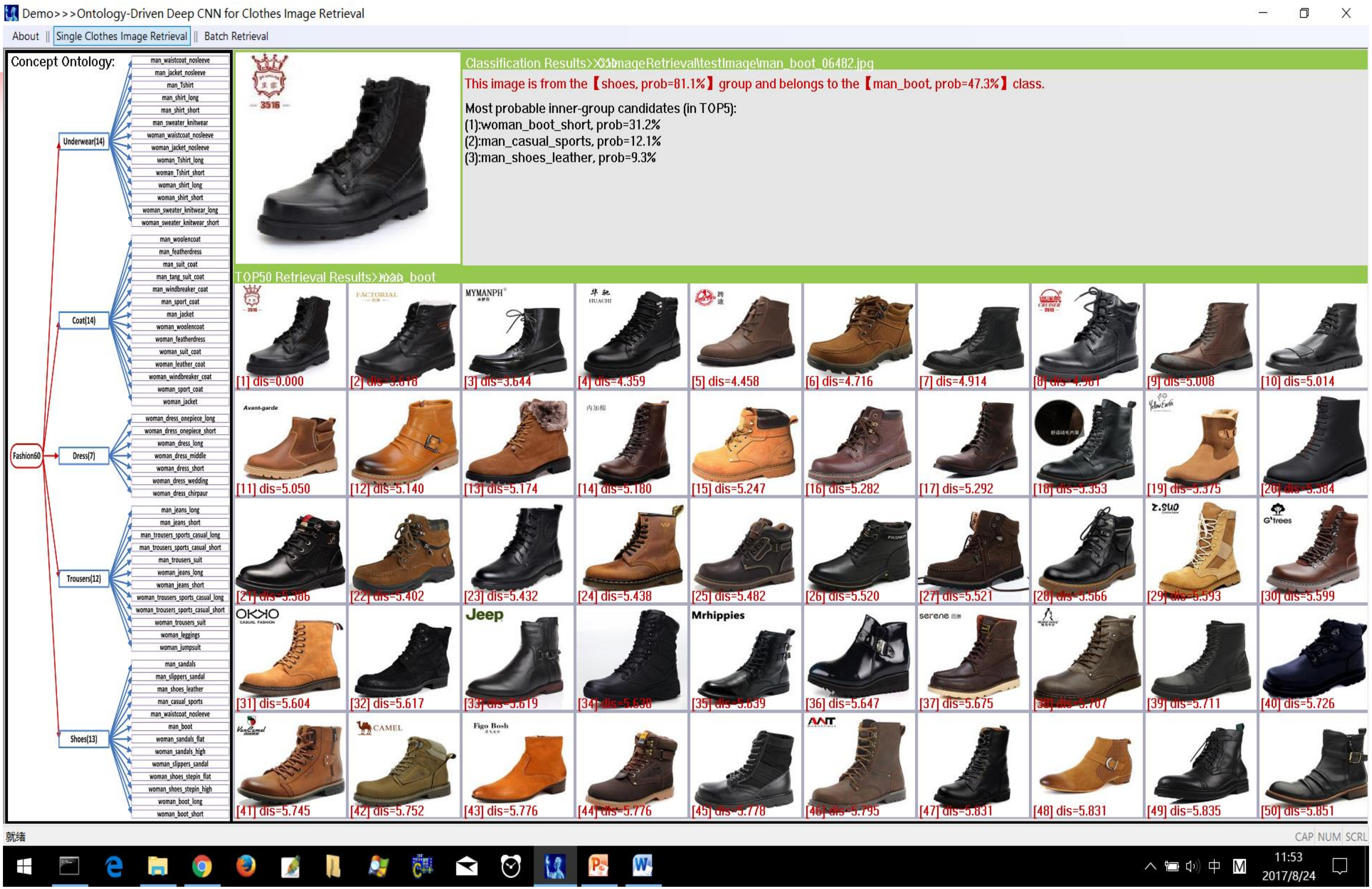
**recognition + search**



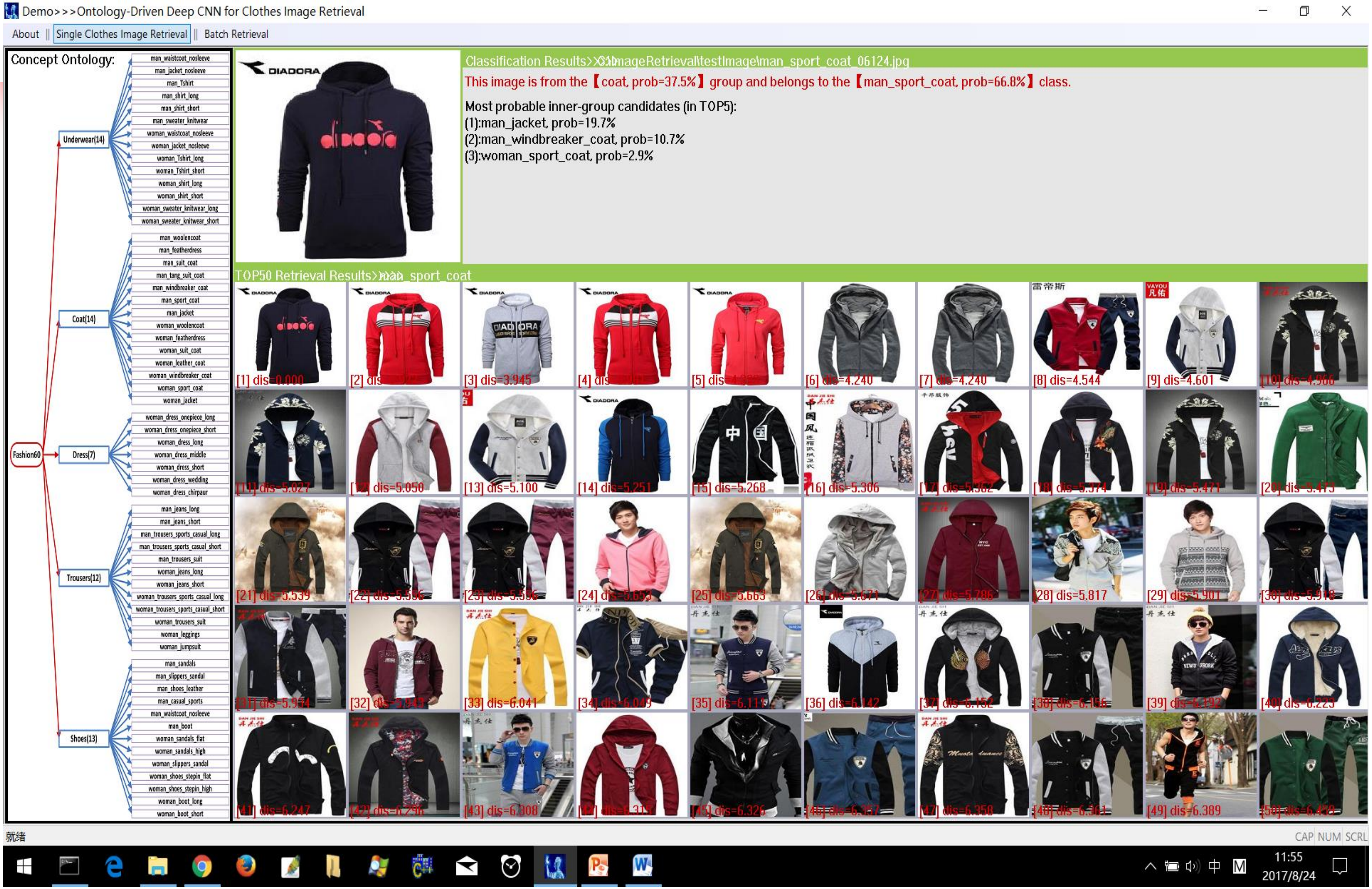
# **Applications**: Fashion Recognition and Search



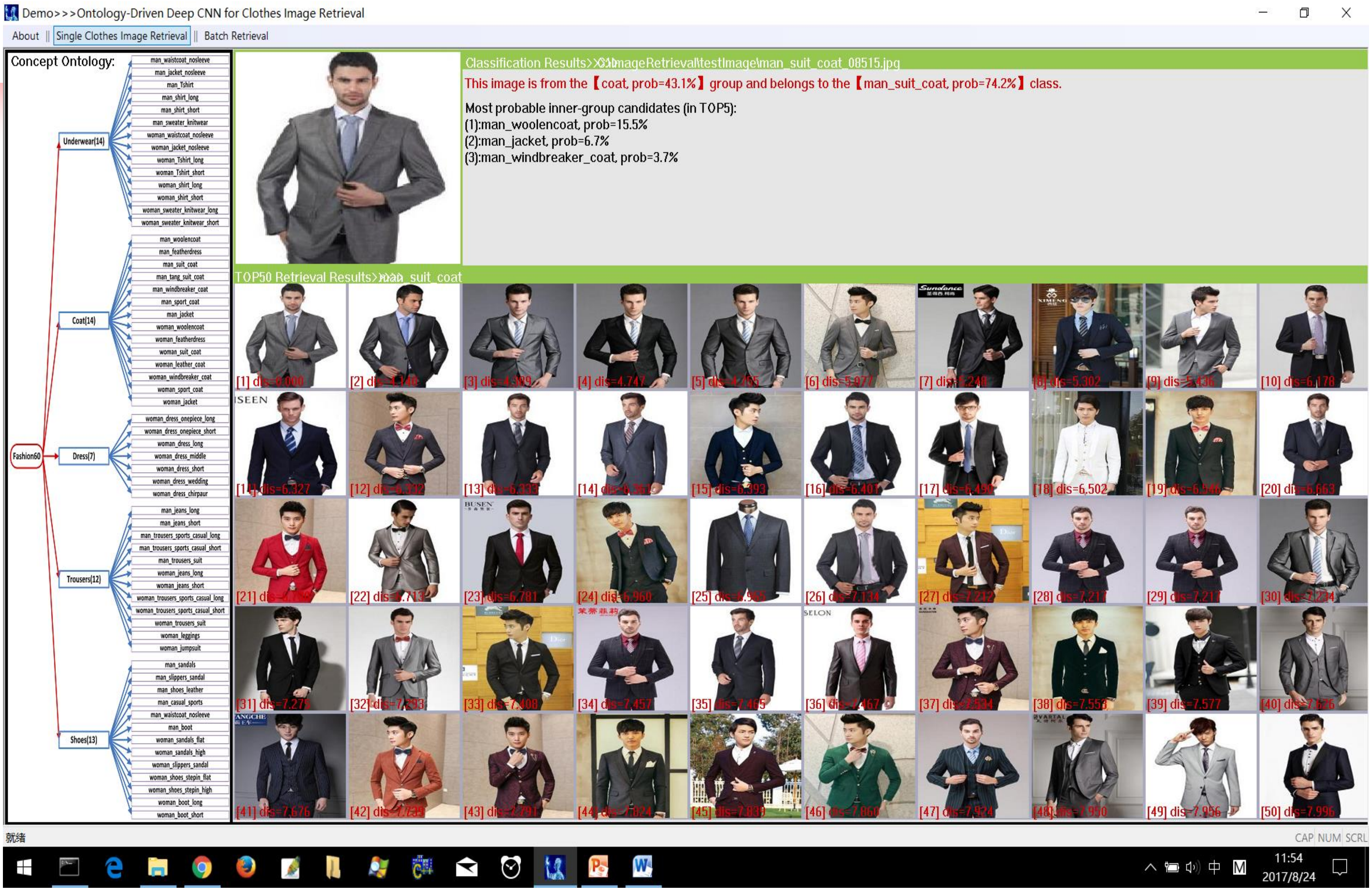
# **Applications**: Fashion Recognition and Search



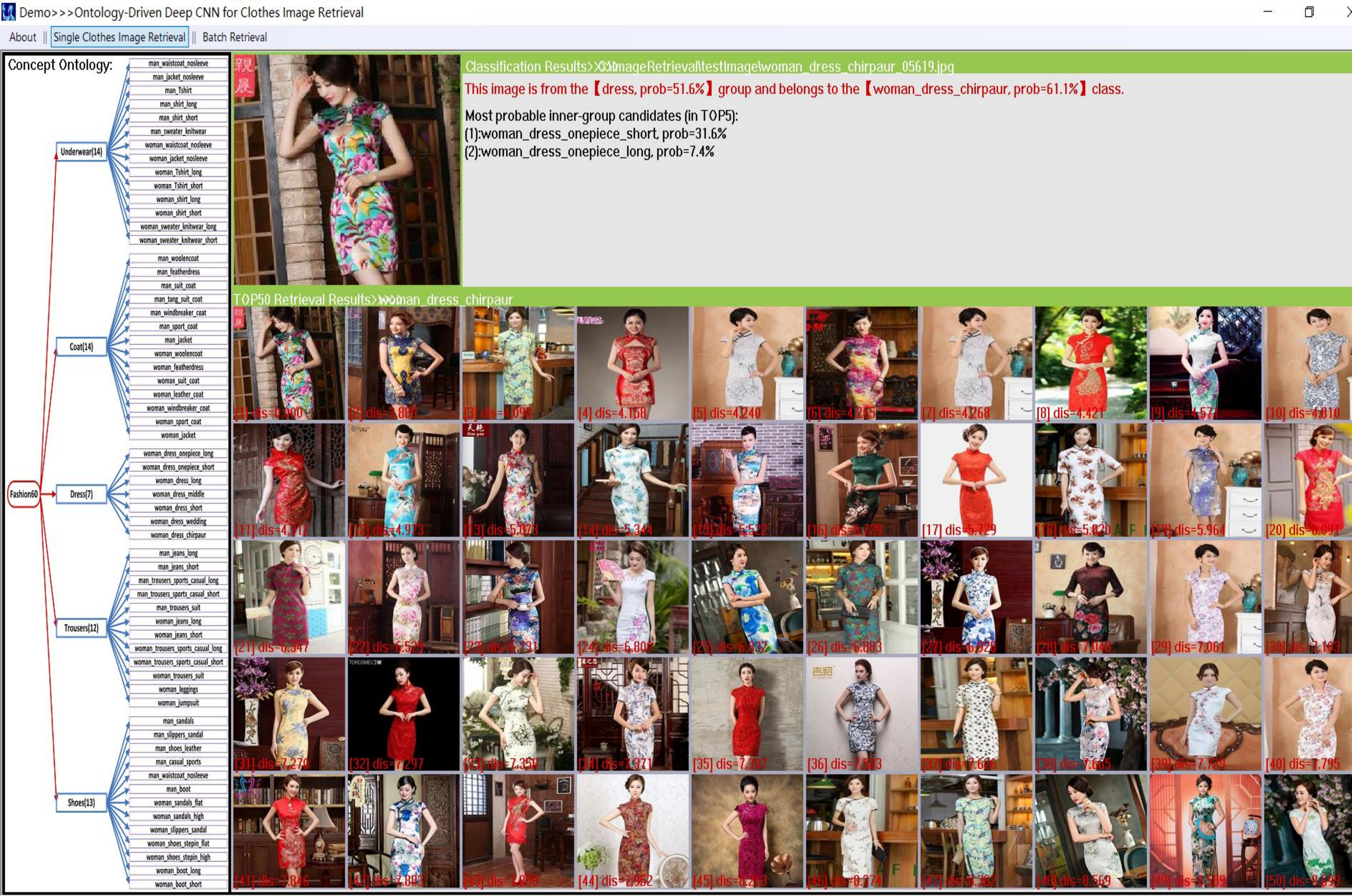
# **Applications**: Fashion Recognition and Search



# **Applications**: Fashion Recognition and Search



# **Applications**: Fashion Recognition and Search



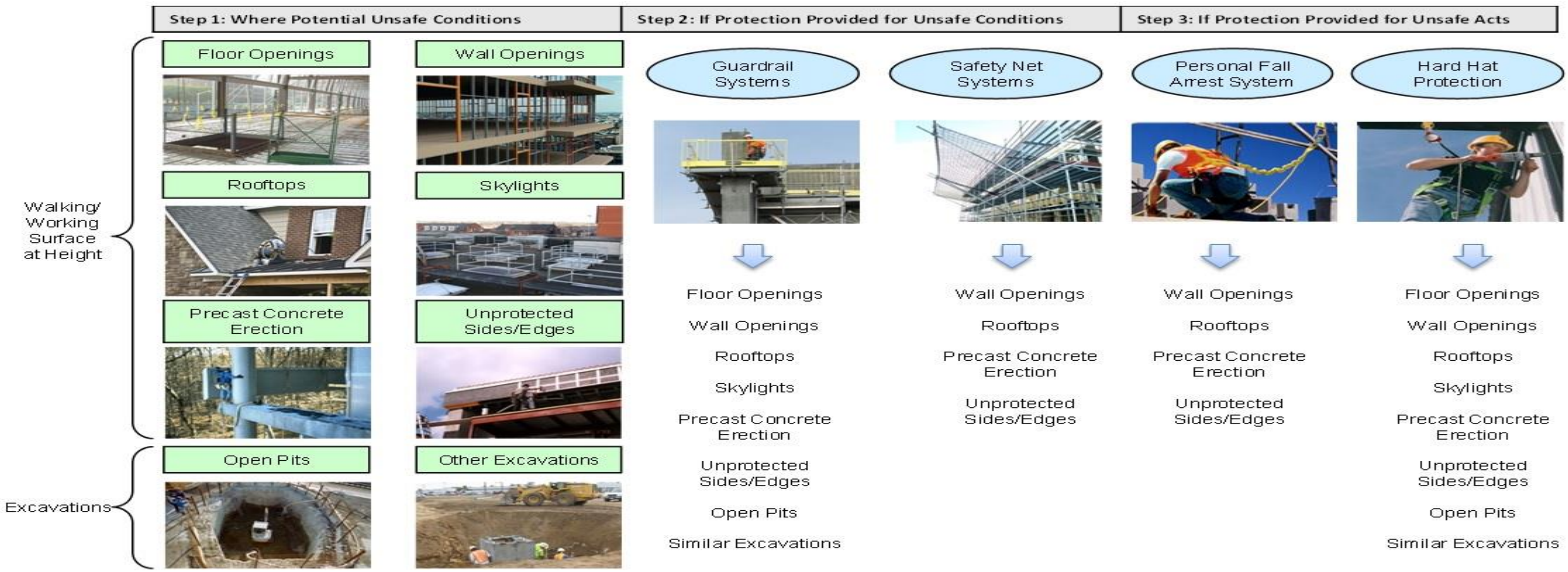
# **Applications**: Fashion Recognition and Search

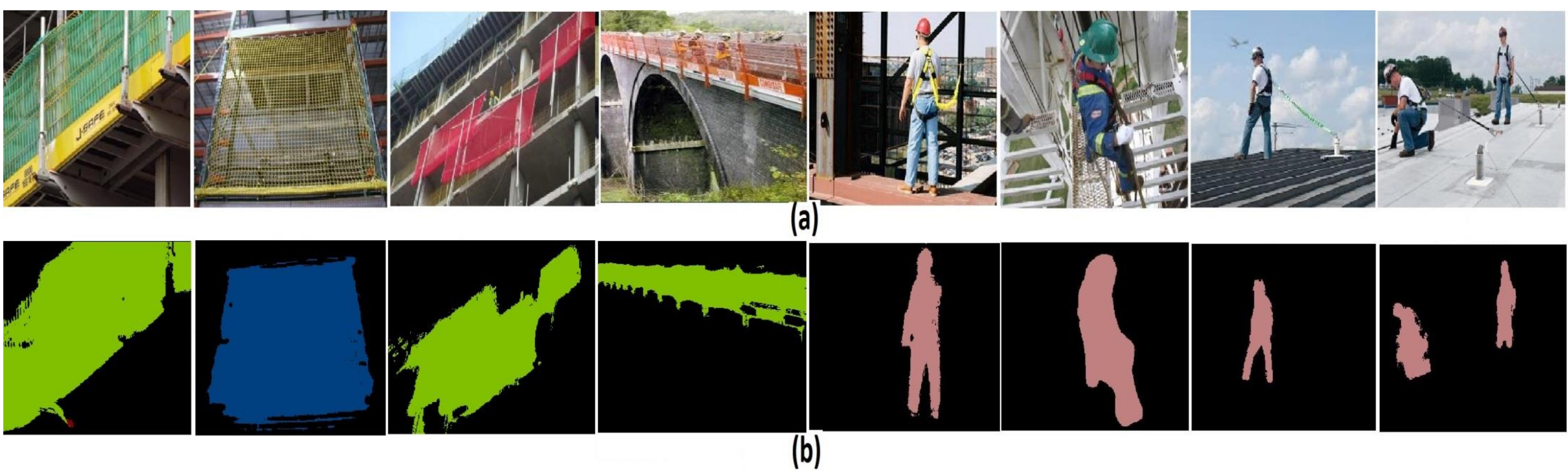
# **Applications**: Smart Home for Elder Care


## Two-stream CNNs

**IEEE Trans. on T-IFS**, vol.13, no.2, 2017
**IEEE Trans. on T-IFS**, vol.14, no.1, 2018
**IEEE J. Biomedical and Health Informatics**, 2014, 2015

# **Applications**: Construction Safety



# Predictions of High-Level Image Concepts

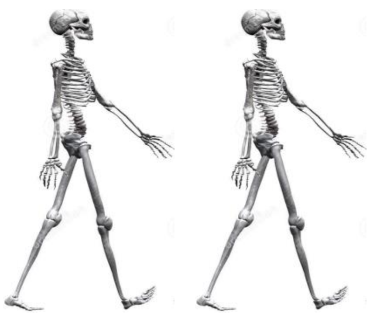
- **Ideas from Chinese Characters**

人

**Human: animal can stand to walk**

# Predictions of High-Level Image Concepts

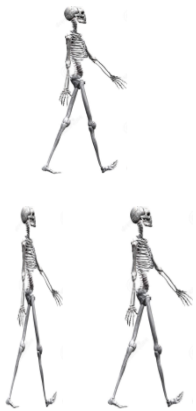
- **Ideas from Chinese Characters**



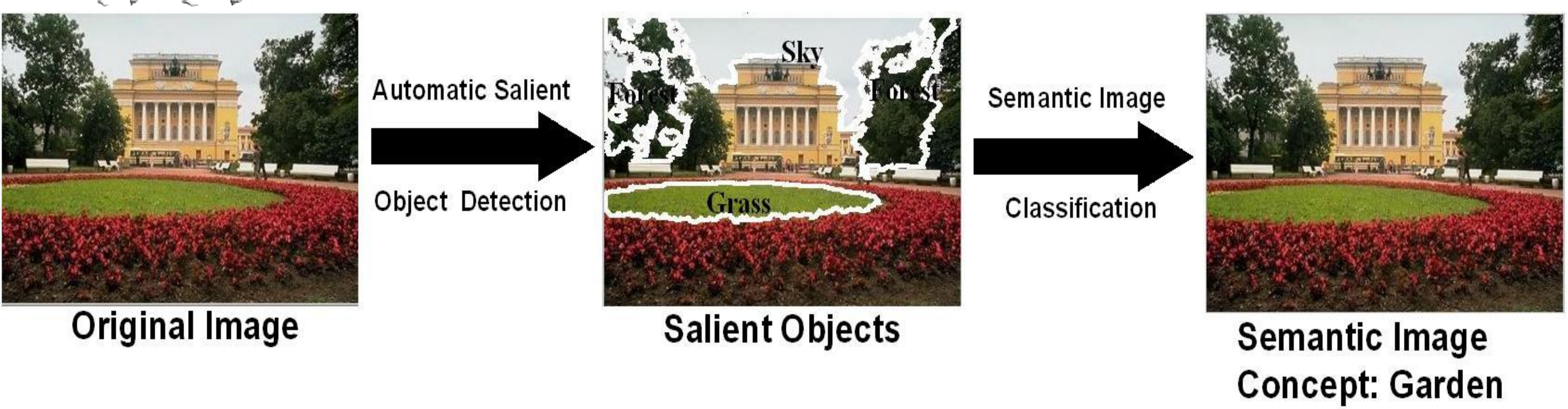
从 **Follow up: one person follows another**

# Predictions of High-Level Image Concepts

- **Ideas from Chinese Characters**



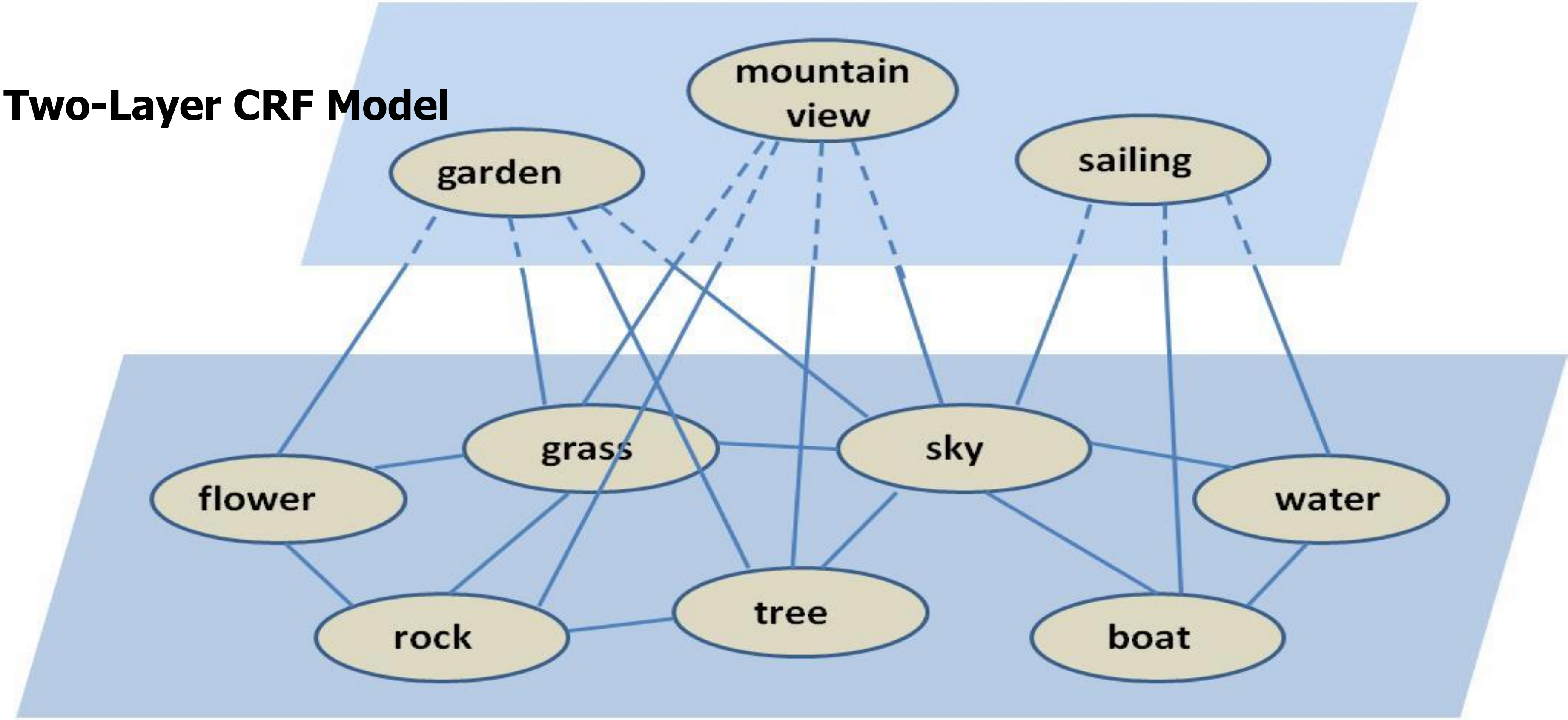
众 **Crowd: many people with leader**



# Predictions of High-Level Image Concepts

- **Context-Driven Prediction of High-Level Semantics**

**Two-Layer CRF Model**



# 7. Conclusions

- **Deep mixture algorithm to integrate diverse outputs from multiple experts with different but overlapping task spaces to generate a mixture network with larger outputs!**
- **Ontology-driven task group generation & identifying the inter-related learning tasks;**
- **Deep multi-task learning to exploit inter-class visual similarities and enhance their seperability;**
- **Deep Boosting to train the deep networks for the hard and easy object classes sequentially in an easy-to-hard way;**
- **Deep Collaborative Learning to learn multiple networks simultaneously & enhance each other;**
- **Knowledge Distillation for model compression for mobile usages.**

# Future Work: Human-like machine learning

- Developing Human-Like Learning Techniques
  ---directly learn from large-scale dirty data
  - *Noise-free machine learning*: machines should know which sources can be trusted more! *–MIL?*
  - *Integrating decisions from different sources with different quality levels*: machines should know which sources are more important and reliable!
  - Human Assistants: where, how & what? mixture intelligence?
  - Chemical Actions in Machine Learning
  - Multi-modal decision or information fusion