Learning-to-Share Based on Finding Groups for Large Scale Image Classification

Li Shen[1], Shuqiang Jiang[2], Shuhui Wang[2], Qingming Huang[1,2]

[1]Graduate University, Chinese Academy of Sciences,
Beijing, 100049, China
[2]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS,
Beijing, 100190, China
{lshen, sqjiang, shwang, qmhuang}@jdl.ac.cn

Abstract. With the large scale image classification attracting more attention in recent years, a lot of new challenges spring up. To tackle the problems of distribution imbalance and divergent visual correlation of multiple classes, this paper proposes a method to learn a group-based sharing model such that the visually similar classes are assigned to a discriminative group. This model enables the class draw support from other classes in the same group, thus the poor discrimination ability with limited available samples can be relieved. To generate effective groups, the intra-class coherence and the inter-class similarity are computed. Then a hierarchical model is learned based on these groups that the classes within the group can inherit the power from the discriminative model of the group. We evaluate our method across 200 categories extracted from *ImageNet*. Experimental results show our model has better performance in large scale image classification.

Keywords: group sharing, large scale classification, hierarchical model

1 Introduction

Visual classification is an important issue in the area of multimedia and computer vision. In recent years, a general trend is towards large scale datasets with many categories [1, 2]. A lot of traditional image classification algorithms have been proposed in the literature. These methods have worked well on small databases. However, they may underperform when the number of categories significantly increases. Some new challenges spring up under the large scale scenario. Firstly, the distribution of multiple classes is usually imbalanced. Many categories have relatively larger available training samples than others, so their classifier may have better performance than the categories with fewer samples. Moreover, visual correlation among the categories is divergent. Some categories are visually similar, meanwhile some categories can be easily discriminated due to the large variance between them. For example, the diversity between the categories of "*duck*" and "*goose* "is slight. They may be confused with each other, however they can easily apart from others such as "*car*" or "*buildings*".

To deal with above challenges, some solutions have been proposed such as multi-task learning [5, 6] or adding exterior information, *i.e.* attribute and tags. Many researches are developed based on multi-task to relieve the imbalance problem by sharing information among tasks. Inspired by the above observations, the similar classes always have some common properties that the irrelative classes do not have. This phenomenon is helpful for modeling the sharing structure.

In this paper, we propose to learn to transfer effective information across related classes by a group-based sharing model. Resembling to the cascade classification [11], the model is hierarchical based on the coarse-to-fine rule. As shown in Figure1, Hierarchical Divisive Clustering [18] is firstly introduced in to effectively analyze intra-class coherence. Based on these analyses, the similarity of classes in pairs is measured to generate the group. Then a hierarchical structure is used to learn the layer-classifiers. The classes in a group are viewed as integral one that can be discriminated from other groups, and they will share the group properties to enhance their own strength. By this method, a hierarchical sharing structure will be learned that can be extended by further researches for classification. In Section 2, we give a brief review about related work. In Section 3 and 4, we will describe the whole system in detail. Experiments are then discussed in Section 5.

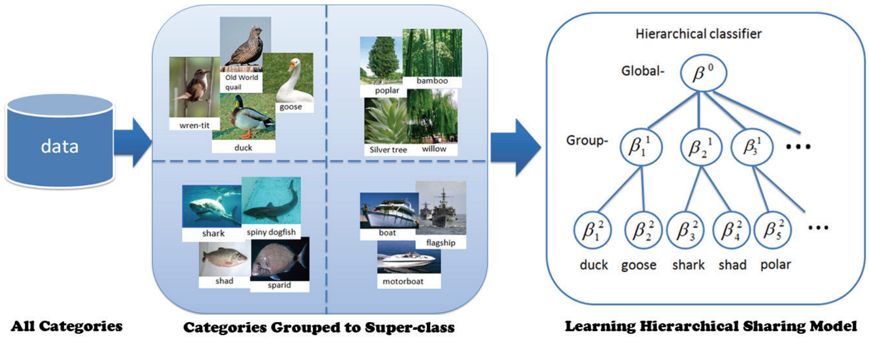

Fig. 1. Framework in the paper. The categories firstly are grouped based on visual similarity. Then use the hierarchical sharing model to train the classifiers, and the categories in a group will share a group vector.

2 Related Works

Considering that each class is a task, the multi-class image classification can be viewed as a mission consisting of multiple related tasks. There have mainly been two strategies to train the classifiers: learning the classifiers for each class separately [3, 4] or learning the classifiers for all categories simultaneously [5, 6]. Many researchers have shown that learning multiple tasks simultaneously can improve performances by virtually sharing information across correlated tasks, whereas it is a critical problem to model the sharing structure among multiple classes. Various attempts have been devised, for example, hierarchical Bayesian modeling assumes that a common hyper prior is shared by all categories [5, 6]. The model ignores the relationship among the categories, *i.e.* to decide which classes should share and what they will share.

How to effectively organize the concepts and data by representing the dependencies among object categories is a critical issue. *WordNet* is often applied to guide the classification as prior [9, 10], while the primitive structure is not completely consistent with visual similarity. Dirichlet Process is also used to identify groups of similar tasks, while the model is so complex [7]. Ruslan *et al.* [8] constructs a hierarchical model by depth-first search strategy to decide which classes a new class should share information with. The method is data-driven, but time-consuming. While adding one more class, all the parameters in structure should be re-trained. Besides, the structure is not coherent because it is influenced by the adding order.

3 Group Model Construction

The target of this step is to find the group which the categories within it are visually similar and can be apart from other ones as larger as possible. Firstly, we need to measure the similarity between classes. Since each class usually has hundred of samples, it is time-consuming to compute the distance between each element in one class and each one in the other. Meanwhile the samples are slightly different in a class, an average vector cannot represent the diversity of the whole class. An effective strategy is conducted to partition one class to a set of sub-clusters with each cluster being compact. There are several typical clustering methods, e.g. k-means [12]. But the specified numbers of clusters are unable to deal with various categories. In this study, we introduce Hierarchical Divisive Clustering[18] method to achieve the goal.

3.1 Partitioning to Sub-clusters

The basic idea of Hierarchical Divisive Clustering [18] algorithm is that, all samples are initialized as a singleton cluster. If the diversity of the cluster is large, the largest margin will split to two smaller clusters with decreased diversity. The partition will stop while the diversity is rather slight. Given a class samples $X_c = \{x_{c,1}, x_{c,2}, \dots x_{c,Nc}\}, X_c \in R^d$ .denote the feature vector of length d for the data belonged to class c and the total number is N_c . For the class c , we aim to get the clusters $S_c^k, k = 1 \dots c_k$. c_k is the number of clusters in class c :

$$S_c^k = \{x_{c,1}^k, x_{c,2}^k \dots x_{c,n_k}^k\}, \sum_{k=1}^{K} n_k = N_c, \tag{1}$$

We firstly compute the Euclidean distance matrix D_{Eu} and Geodesic distance matrix D_{Ge} base on K-nearest Neighbor Graph in pairs, and define a rate $R(x_{c,i}, x_{c,j}) = D_{Ge}(x_{c,i}, x_{c,j}) / D_{Eu}(x_{c,i}, x_{c,j})$ to measure the correlation between $x_{c,i}$ and $x_{c,j}$. We can use average rate to measure the compactness of the cluster:

$$\overline{R}=\frac{1}{n_k\cdot n_k}\sum_{i=1}^{n_k}\sum_{j=1}^{n_k}R\left(x_{c,i},x_{c,j}\right) \tag{2}$$

With these definitions, HDC [18] algorithm is summarized in Algorithm 1.We use the average vector to represent each sub-cluster.

Algorithm 1: Hierarchical Divisive Clustering algorithm

```
Input: threshold $r$ and $X_c=\left\{x_{c,1},x_{c,2},\dots x_{c,N_c}\right\},X_c\in R^d$
Initialize: $S_c^1=\left\{x_{c,1},x_{c,2},\dots x_{c,Nc}\right\}$
While
Choose $S_c^k$ with the largest $\overline{R}$ , If $\overline{R}\le threshold$ break. Otherwise, according to $D_{Ge}$ select two furthest seed points $x_L,x_R$ from $S_c^k$ , $S_c^k\left(L\right)=\left\{x_L\right\}$ , $S_c^k\left(R\right)=\left\{x_R\right\}$ , $S_c^k=S_c^k\setminus\left\{x_L,x_R\right\}$
While $S_c^k\neq\varnothing$
    $Ne_L=\left\{kNN\left(S_c^k(L)\right)\cap S_c^k\right\}$ , $Ne_R=\left\{kNN\ \left(S_c^k(R)\right)\cap S_c^k\right\}$
If $\left(\text{inter=}Ne_L\cap Ne_R\right)\neq\varnothing$
$\text{inter}_L=\left\{dist\left(\text{inter,}S_c^k(L)\right)<dist\left(\text{inter,}S_c^k(R)\right)\right\}$ , others are $\text{inter}_R$
$S_c^k(L)=S_c^k(L)\cup\left\{Ne_L\setminus\text{inter}_R\right\}$ , $S_c^k(R)=S_c^k(R)\cup\left\{Ne_R\setminus\text{inter}_L\right\}$ ,
Else
$S_c^k(L)=S_c^k(L)\cup Ne_L$ , $S_c^k(R)=S_c^k(R)\cup Ne_R$ ,
End
$S_c^k=S_c^k\setminus\left\{Ne_L\cup Ne_R\right\}$
The cluster $S_c^k$ splits into $S_c^k(L)$ and $S_c^k(R)$ , $k=k+1$ ,compute the $\overline{R}_c^k\left(L\right)$ and $\overline{R}_c^k\left(R\right)$ according to (2)
Output: $S_c^k=\left\{x_{c,1}^k,x_{c,2}^k\dots x_{c,n_k}^k\right\}$
```

3.2 Constructing Class Group

There are some traditional methods to calculate the distance between classes with several sub-clusters, *i.e.* single-linkage, complete-linkage and average-linkage. We adopt average-linkage to represent the pair-wise similarity.

After computing the distance among classes, we use Affinity Propagation [13].The method aims to find the exemplar to represent the cluster, so the class dissimilar with others can be separated rather assigning to a group. Moreover, AP can find uniform clusters.

4 Learning the Group-based Sharing Model

4.1 Traditional Classification Model

Suppose we are given a set of N training samples belonging to K categories, $X_k = \{(x_i, y_i) \mid i = 1 \ldots N\}$, where x_i denotes the training samples' feature and y_i denotes the corresponding class label. Considering the multi-class classification problem, it will be equivalent to several two-class ones. For class k, the label y_i of sample x_i can be transferred to a binary value indicating whether the sample belongs to it. It assumes the following probability model:

$$P(y_i = 1 \mid x_i, \beta^k) = \frac{1}{1 + \exp(-\beta^k x_i)} \tag{3}$$

where $\beta^k = [\beta_0^k, \beta_1^k, \ldots \beta_D^k]$ is the regression coefficients for class K ,and the β_0^k is the bias term. We append each instance with an additional dimension $x_i \leftarrow [x_i, 1]$ to adapt to β . Moreover, L_2 penalized term is added to obtain good generalization ability. The regularized logistic regression is in the following form:

$$\min_{\beta} \sum_{k=1}^{K} l^k(\beta^k) + \frac{\lambda}{2} \left\| \beta^k \right\|_2^2, \tag{4}$$

and the loss function is

$$l^k(\beta) = \sum_{x_i \in X_k} \log\left(1 + e^{-\beta^T x_i}\right) \tag{5}$$

4.2 Group Sharing Model

The original C categories have been partitioned to Z groups, and each group has at least one category. z_c represents the group category c belongs to. The categories in a group can be viewed as a generalized category divided from other groups. The classes in lower level will inherit its parents' information .As shown in Figure 2, the classifier of each class is the sum of classifiers along the tree [8].

For example, the classifier of "*duck*" is given by (6). β^0 is the global classifier shared cross all categories, β^1 is the group classifier shared by its child classes, *e.g.* β_1^1 is shared by "*duck*", "*goose*". β^2 is the specific classifier used by the special class.

$$\beta_{duck} = \beta^0 + \beta_1^1 + \beta_1^2 \tag{6}$$

According to the traditional logistic regression, the Group Sharing Model can be formulated as (7). $l^c(\beta^c)$ is computed by (5).

$$\min_{\beta} \frac{\lambda_0}{2} \left\| \beta^0 \right\|_2^2 + \frac{\lambda_1}{2Z} \sum_{z=1}^{Z} \left\| \beta_z^1 \right\|_2^2 + \frac{\lambda_2}{2C} \sum_{c=1}^{C} \left\| \beta_c^2 \right\|_2^2 + \sum_{c=1}^{C} l^c(\beta^c) \tag{7}$$

where $\beta^c = \beta^0 + \beta_{z_c}^1 + \beta_c^2$.

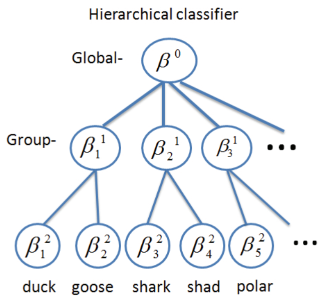

Fig. 2. Group Sharing Model

4.3 Learning the Model

The hierarchical model have established after constructing the group. Given the tree structure, the model can optimized efficiently using iterative procedure [8], as shown in Algorithm 2. The object function can be decomposed into several separated problem. For example, when β^0 and β^2 are given, β_z^1 can be optimized efficiently based on Trust Region Newton method [14] as traditional single class model.

```
Algorithm 2: Group Sharing Model optimization
Input: the group $Z$ and basic-level classes $C$
Initialize: $\beta^0 = 0, \beta^1 = 0, \beta^2 = 0$
While (not Converged)
  (1)  Given $\beta^1$ and $\beta^2$ ,optimize global-level $\beta^0$ using Eq.7
  (2)  for $i = 1:Z$
           Given $\beta^0$ and $\beta^2$ , optimize parent-level $\beta_z^1$ using Eq.7
       end
  (3)  for $j = 1:C$
           Given $\beta^0$ and $\beta^1$ , optimize basic-level $\beta_c^2$ using Eq.7
      end
end
Output: Hierarchical classifiers $\beta^0, \beta^1, \beta^2$
```

5 Experimental Results

In this section, we systematically evaluate our proposed framework on a subset of the *ImageNet* dataset [2] .We randomly select about 200 concepts covering sub-categories from *Animal*, *Plant*, *Instrument, Scene*, and *Food* which distribute different levels across wide domain. The set contain the simple concepts with coherent visual appearance such as "*apple*" and "*goldfish*", also has the concepts with large visual variance *i.e.* "*book*" and "*cup*". And the number of samples is quite different, from several to thousands.

We divide the samples of each class into two equal sets: one is for training, the other is for testing. The feature we used is Color Moment and PHOG-180[15] to represent the color distribution and local shape of the image.

5.1 Constructing Group

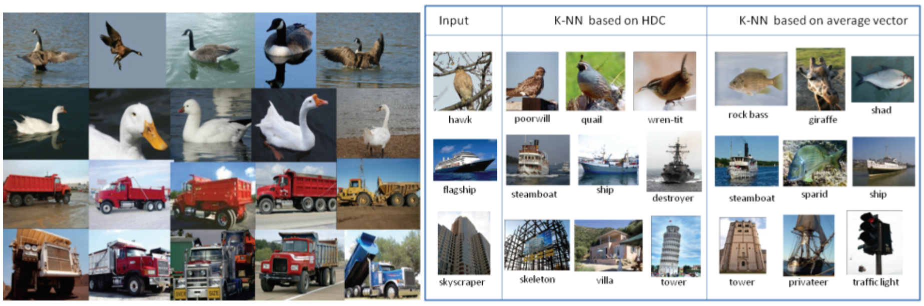

Fig. 3. (Left) Partition example: the subsets from class "*duck*" and "*dump truck*" . (Right)Similarity measure compared with HDC and Average Vector.

In this step, we firstly use HDC[18] to describe the diversity within class. The number of subsets and the variance among them is determined by the property of class. As shown in Figure 3(left), the top two rows are extracted from two subsets of class "*duck*". The partition highlights the color's variance. The below two rows are extracted from two subsets of class "*dump truck*" that represent the multiple views of the truck. We measure the inter-class similarity by computing average distance among their subsets.

We compare the method with average vector. Figure 3(right) shows the K-Nearest Neighbor concepts based on HDC method and average vector. It is shown that HDC method always has stable performance. When the class has a common appearance and the diversity within class is small, better result can be got in terms of average vector, such as "*tower*" is more similar to "*skyscraper*" than "*skeleton*". However, it may be inaccurate when the class has large intra-diversity and varied background.

Figure 4 shows the distribution of the training samples for 207 concepts. The concepts are arranged in groups represented by different colors. Observe that the union of many concepts is consistent with semantic similarity, *i.e.* {"*car*", "*railcar*", "*truck*", "*tractor*", "*pantechnicon*", "*van*"}; {"*hawk*", "*duck*" , "*quail*", "*wren-tit*", "*poorwill*", "*goose*"}; {"*squirrel*", "*kangaroo*", "*wolf*", "*fox*", "*lion*", "*tiger*"}; {"*boat*", "*destroyer*", "*flagship*", "*steamboat*", "*privateer*", "*ship*"}; {"*bed*", "*double bed*", "*sofa*"}. Moreover, the classes with visual concurrence such as {"*aircraft*", "*airplane*"}, {"*ship*", "*ocean*"} and {"*sky*", "*mountain*"} are also in the same groups. However, some unions are different from semantic similar. For example, "*mouse*" is related with "*keyboard*" and "*computer*" though they are not visually similar; a lot of instruments are not similar with each other, such as "*surgical knife*", "*scoop*" and "*reamer*". "*apple*" looks like "*tomato*" rather than "*grape*", and the "*ice bear*" is more similar to "*goose*" with the same color and similar background.

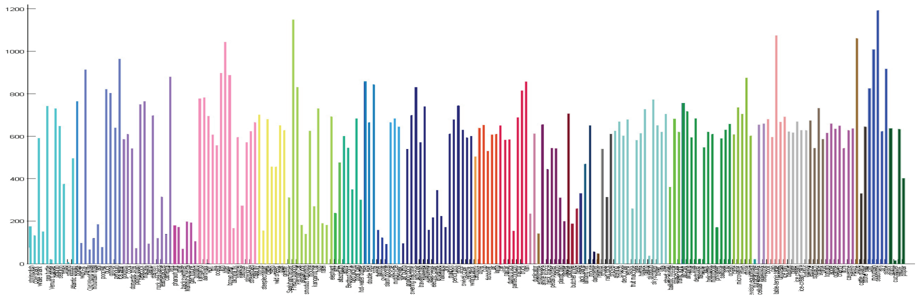

Fig. 4. The distribution of the training samples for 207 concepts

5.2 Performance of the Model

In order to investigate the performance of Group Sharing Model, we compare it with the following three models: Single Class Model, is trained based on'1 against all' rule. The SGD-QN method [16] is introduced to train the model fast and effectively. Global Sharing Model, use a single global classifier for sharing [17]. Ruslan *et al.* [8] uses depth-first search strategy to decide which classes a new class should share information with.

Table 1. Time cost and MAP compared among the methods

Model	**Single Class**	**Global Sharing**	**Ruslan** *et al.*	**Group Sharing**
Time cost(/h)	21.5	6.5	170	12
MAP(%)	1.51	2.95	3.23	3.44

Table 1 show the results of time cost and mean average precision(*MAP*). Due to the categories variance and the influence of complex background, the *MAP* of all categories is not so well. However, Group Sharing Model also have comparatively good performance. In term of the time cost, Global Sharing Model has the lowest the time cost (about 5.5 hours). Due to the process of finding group and three level hierarchical modal, the complex of our Group Sharing Model is increasing (about 12 hours). While it is still lower than the other two methods. The cost is of Single Class Model (about 21.5 hours) is high despite using a fast optimization method. Ruslan *et al.* model is time-costing because of the dynamic structure and duplicate training (more than a week). It can use parallel method to decrease the tremendous complexity.

Figure 5 (left) displays the improvements in *AP*(%) of Group Sharing and Ruslan *et al.* [8] for all the categories over the Global Sharing Model. It shows the mid-level groups contribute to learning the data with large scale. Observe that the decrease in Ruslan *et al.* model is obvious. The category order and amount distribution of related classes may lead to the negative transfer. Figure 5 (right) shows the average improvements of groups (the number is 45) over Global Sharing Model.

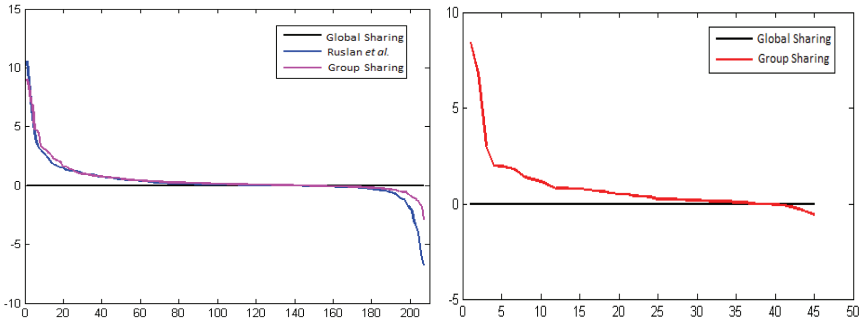

Fig. 5. (Left) *AP* improvements of Group Sharing and Ruslan *et al.* Model over Global Sharing Model.(Right) average *AP* improvements of the groups in Group Sharing Model over Global Sharing Model.

In term of our model, the top 3 largest improvement in *AP* is "*mailbox*"(+9.01) in the group containing {"*pencil box*", "*envelope*"}, "*lettuce*"(+8.42) in the group containing {"*spinach*", "*olive*"}, and "*peacock*"(+7.24) in the group containing {"*cock*" , "*macaw*"}. They are benefit from visually related categories. However, *AP* in some categories decreases, such as "*poniard*" (-2.86), "*slash pocket*" (-1.81) and "*tachina fly*" (-1.60). These concepts always have identical appearance, and the group may bring in extra noise.

Table 2. Most Confuesd Categories based on Group Sharing Model

Categories	Top 3 of Confused Categories		
whale shark (39.08)	dolphin(8.34)	sea turtle (5.97)	cow shark (2.75)
dolphin (8.89)	whale shark (27.03)	sea turtle (5.66)	cow shark (2.56)
sea turtle (7.25)	whale shark (20.30)	dolphin (6.36)	cow shark (3.40)
cow shark (2.94)	whale shark (18.17)	dolphin (7.95)	sea turtle (5.30)
orange (14.41)	cayenne(11.71)	tomato (7.58)	cherry (3.39)
cayenne (15.77)	tomato (7.49)	orange (7.40)	cherry (5.17)
tomato (6.70)	orange (16.89)	cayenne (9.01)	apple (3.16)
cherry (5.83)	cayenne (14.03)	tomato (6.20)	orange (4.12)

In order to describe the group performance, we use a singleton classifier to classify all the test data. Table 2 displays the classifiers' performance and their most confused categories. It is shown that the model always confuses the visually similar samples in the same group. And these confusing may be acceptable.

6 Conclusion

In this paper, we propose to learn a hierarchical group-based sharing model by exploring the visual relatedness among categories. The categories with similar visual appearance are partitioned in a group and can improve the own strength with the aid of their groups. The fixed tree model can be effectively extended to further research, such as kernel learning, multi-feature integration and feature selection.

Acknowledgments. This work was supported in part by National Natural Science Foundation of China: 61025011, 60833006 and 61070108, and in part by Beijing Natural Science Foundation: 4092042.

References

[1] Russell, B., Torralba, A., Murphy, K., and Freeman, W. T. Labelme: a database and web-based tool for image annotation. *IJCV,* 2008, 77, 157–173.

[2] L. Fei-Fei, ImageNet: crowdsourcing, benchmarking & other cool things, *CMU VASC Seminar,* 2010

[3] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems, *Comput. Learing Theory,* 2000,pp. 35–46.

[4] J. Weston and C. Watkins. Multi-class support vector machines, *Proc. ESANN99, M. Verleysen, Ed., Brussels, Belgium,*1999.

[5] Torralba, A., Murphy, K. P., and Freeman, W. T. Sharing visual features for multiclass and multiview object detection. *PAMI,* 2007, 29, 854–869.

[6] Yu, K., Tresp, V., and Schwaighofer, A.. Learning Gaussian processes from multiple tasks. *ICML*, 2005,1012–1019.

[7] Ya Xue, Xuejun Liao, Lawrence Carin. Multi-Task Learining for Classification with Dirichlet Process Priors. *JMLR 8*,2007, 35-63

[8] Ruslan Salakhutdinov, Antonio Torralba, Josh Tenenbaum. Learning to Share Visual Appearance for Multiclass Object Detection. *CVPR*, 2011

[9] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. *CVPR*, 2007

[10] Rob Fergus, Hector Bernal, YairWeiss, and Antonio Torralba. Semantic Label Sharing for Learning with Many categories. *ECCV* , 2010

[11] Paul A. Viola, Michael J. Jones.Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade. *Advances in Neural Information Processing Systems*

[12] MacQueen, J. B. Some methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability,*1967.

[13]Brendan J. Frey and Delbert Dueck. Clustering by Passing Messages Between Data Points. *University of Toronto Science 315, 972–976, February* 2007

[14] Chih-Jen Lin, Ruby C. Weng, S. Sathiya Keerthi. Trust Region Newton Method for Large-Scale Logistic Regression. *JMLR 9,*2008, 627—650

[15] Bosch, A., Zisserman, A. and Munoz, X. Representing shape with a spatial pyramid kernel.*CIVR,*2007

[16] Antoine Bordes, Léon Bottou, Patrick Gallinari. SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent. *JMLR 10* ,2009,1737-1754

[17] T. Evegniou and M. Pontil. Regularized multi–task learning. *KDD* 2004.

[18] L. Kaufman, P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. *Wiley, New York.* 1990.