Object Classification Using a Semantic Hierarchy

Somayah Albaradei[1,2] and Yang Wang[1]

[1] Department of Computer Science, University of Manitoba
[2] Department of Computer Science, King Abdulaziz University

Abstract. We consider the problem of object classification by exploiting the hierarchy structure of object categories. Our proposed method first train a collection of binary classifiers to differentiate pairs of object categories at different levels of the object hierarchy. Then we use the outputs of these classifiers and the object hierarchy to define a new image representation. Our experimental results show that our proposed method outperforms other baseline methods on several image classification datasets.

1 Introduction

Object recognition is a cornerstone problem in computer vision. Most previous work in this area approaches object recognition as a pattern classification based on low-level image representations. For example, a popular image representation is the bag-of-words (BoW) representation based on local image descriptors, such as SIFT [1].

Although low-level image features have shown promise in many applications, there are inherently limited since they do not capture high-level semantic information about object classes. When we move towards high-level recognition tasks, these low-level features often do not offer enough discriminative powers. This is commonly referred to as the "semantic gap" problem in computer vision. To address this limitation, some recent work has proposed to use semantically more meaningful features. Examples of such features include object bank [2], classme [3], action bank [4], etc. These methods first learn a set of classifiers for certain high-level concepts, then use the responses of these classifiers as mid-level features for various visual recognition problems.

Object categories naturally form a hierarchy (also called taxonomy) with many levels of abstraction. For example, ImageNet [5] organizes all the object categories according to the WordNet hierarchy. Nodes closer to the root of the hierarchy correspond to more abstract concepts, while nodes closer to leaves correspond to finer-grained concepts. For example, a path in the hierarchy might correspond to "living thing $\rightarrow$ animal $\rightarrow$ mammal $\rightarrow$ dog $\rightarrow$ shepard dog". The object hierarchy provides a very rich semantic information about various object categories. In this paper, we exploit the hierarchical structure to develop a new mid-level image representation for object classification.

2 Related Work

Hierarchical classification is an active area of research in computer vision. Some work in this area focuses on using the hierarchy for improving efficiency. For example, Bengio et al. [6] proposed to use the hierarchical tree of object classes to achieve sublinear running time during testing. Deng et al. [7] developed an improved method by learning the hierarchy jointly with the classification model. Gao et al. [8] further improved the method by allowing overlapping object classes at different child nodes. Sun et al. [9] proposed to use the branch-and-bound technique on object hierarchy for efficient classification. Object hierarchy has also been used to improve image retrieval [10] and to provide accuracy-specificity trade-offs in large scale recognition [11].

Our work is related to visual recognition using mid-level features. Li et al. [2] proposed an image representation called *object bank* for scene recognition. This representation first learns a large collection of object detectors. For a given image, these detectors are applied and their responses are used as mid-level features for recogition. Sadanand and Corso [4] adapted the object bank representation to action recognition in videos. Torresani et al. [3] developed a similar representation called *classme* for object classification. Classme first learns classifiers for a sef of basis classes. Any new object category is then represented as combinations of these basis classes.

Our work is most closely related to [12] and [13]. Cao et al. [12] proposed a framework for learning mid-level features called "learning by focus". Their method learns a set of one-vs-one classifiers between pairs of object classes. The responses of these classifiers are then used as features for recognition. The reason for using one-vs-one (instead of one-vs-rest) classifiers is that it is easier to distinguish different concept pairs. Albaradei et al. [13] extended [12] by learning binary classifiers for concept pairs at different levels of the object hierarchy. The main difference between our work and [13] is how to construct image representations from those binary classifiers. The method in [13] chooses to concatenate the scores of all binary classifiers as the image representation. In contrast, our image representation will exploit the hierarchical structure of the object categories.

3 Our Approach

We assume that we are given a tree-structured taxonomy of object categories, e.g. the WordNet hierarchy in ImageNet[5]. The taxonomy organizes object classes into many levels of abstractions (also called "synset" in [5]). A path in the taxonomy indicates the "is-a" relationship between various object classes. For example, a "shepard" is a "dog", which in turn is a "mammal". Our goal is to classify an input image into one of the object categories corresponding to leaf nodes in the taxonomy.

An overview of our approach is shown in Fig. 1. For a given image, we first extract standard low-level visual features (e.g. color, texture, shape, etc). We then apply a large collection of binary classifiers on the low-level features. The

responses of these binary classifiers are used to construct a mid-level image representation.

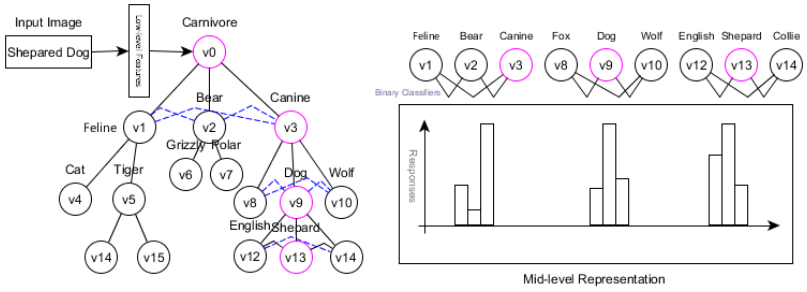

Fig. 1. An overview of our approach: (left) Given a tree-structured hierarchy, we construct a set of binary classifiers. Each dash blue line represents a classifer between two concepts in the hierarchy; (right) For a given image, we represent the image using a vector of mid-level features. The entries of this vector are the responses of the corresponding binary classifiers on this image.

3.1 Mid-Level Features

Most previous approaches in image classification use low-level features (e.g. color, texture, shape) to represent an image. A discriminative classifier is trained based on the low-level features. The limitation of low-level features is that they do not have any semantic information about object categories. In particular, they do not capture the hierarchical structure of object categories. This is commonly known as the "semantic gap" in high-level computer vision tasks.

To address this semantic gap, previous work has proposed mid-level features that offer more semantic meaning. For example, [12, 13] apply a large collection of binary classifers on the low-level image features. The scores of these binary classifiers are used as the mid-level features. In our work, we use the method in [13] to learn the collection of binary classifiers, since it allows exploiting the hierarchical structure of object classes.

For each internal node V with k children in the hierarchy, we learn $k(k-1)/2$ binary classifiers by selecting each pair of its children as positive/negative classes. For example, suppose that V corresponds to the concept "mammal" and it has three children: "dog", "cat" and "horse". We will construct three concept pairs "dog-vs-cat", "dog-vs-horse", "cat-vs-horse". We do the same for all internal nodes in the hierarchy. In the end, we have a large collection of concept pairs. Some concept pairs (e.g. "animal-vs-plant") correspond to coarse object categories, where others (e.g. "shepard" vs "husky") correspond to finer object categories. For each concept pair, we learn a binary linear SVM classifier

based on low-level image features to differentiate these two concepts. In the end, we will have a collection of binary SVM classifiers. For a new image, we can apply these binary classifiers and each of them will produce a score. In Sec. 3.2, we will describe how to construct a mid-level image representation using these scores.

3.2 Image Representation Using Object Hiearchies

Given the collection of binary linear SVMs in Sec. 3.1, we would like to construct an image representation using the scores of these linear classifiers. The image representation in [13] uses the concatenation of the scores from all the binary linear SVMs. We believe this approach has some limitations. First of all, the method in [13] has to evaluate all the binary linear classifiers on an image. But intutively, only a subset of these classifiers will produce meaningful scores on a given image. As an example, let us consider an image of "shepard dog". If a concept pair (e.g. "apple-vs-banana") is irrelevant to "shepard dog", the score of the corresponding linear classifer will likely to be close to 0. This suggests that we only need to evaluate a subset of the binary classifiers and approximate the scores of the remaining ones with 0.

To operationalize this intuiton, we propose to construct the image representation by only considering the most relevant concept pairs. We first describe how to construct the image representation for training images, where we know the ground-truth labels. In Sec. 3.3, we will explain how to handle test images where the ground-truth labels are unknown.

Given a training image, since we know its ground-truth label, we can find the path (from the root to a leaf) of its object class in the hierarchy. For each internal node V along the path, we consider the concept pairs between each pair of its children to be "relevant". Our intuition is that these concept pairs are most likely to provide discriminative information for this path. For example, let us consider a training image of "shepard dog" in the hierarchy in Fig. 1. We first find the ground-truth path (from the root to a leaf node) corresponding to "shepard dog". In this case, the path is "$v_0 \rightarrow v_3 \rightarrow v_9 \rightarrow v_{13}$". The relevant concept pairs at v_0 are $(v_1, v_2), (v_1, v_3), (v_2, v_3)$. Similarly, the relevant concept pairs at v_3 are $(v_8, v_9), (v_8, v_{10}), (v_9, v_{10})$.

The final image representation is a vector of SVM scores. An entry of this vector is nonzero only when its corresponding concept pair is "relevant". Note that the number of relevant concept pairs can be different for different object classes. But the length of the image representation is identical for all classes. If the total number of linear SVMs (from Sec. 3.1) is M, the length of this vector is M. This vector is also sparse, since a lot of the entries correspond to irrelevant concept pairs and will be set to zero.

In the end, each training image is represented as a M-dimensional sparse vector. We then learn a non-linear multi-class SVM to classify the image into one of the K classes. Given the image representation, we can also use this nonlinear classifer to obtain the score of predicting each of the K classes.

3.3 Prediction on Unseen Images

For an unseen image during testing, we cannot directly construct the image representation in Sec. 3.2 since we do not know its ground-truth path in the hierarchy. A naïve approach is to traverse each of the K possible paths from the root to leaves. For the i-th path, we can construct the image representation using the method in Sec. 3.2. We then use the K-class nonlinear classifier to obtain a score of predicting the i-th class. After traversing all paths, we will have the score for each of the K classes. In the experiments (Sec. 4), we will show that this naïve approach gives reasonably good results. But the limitation of this approach is that it is very computationally expensive, since we need to repeatedly traverse paths in the hierarchy. In the following, we propose a more efficient approach based on simple heurstics.

The naïve approach requires traversing the hierarchy starting from the root. At each internal node, it recusively visit all of the children of this node in a depth-first search manner. This procedure is repeated recursively until all nodes in the hierarchy are processed. In contrast, our approach only choose a subset of the children to visit at each internal node. In other words, we effectively prune many branches in the search tree.

Let x be an unseen image, v be an internal node with n children $\{c_1, c_2, ..., c_n\}$. Sec. 3.1 gives us a binary linear SVM classifier between each pair c_i and c_j $(i, j \in \{1, 2, ..., n\})$. Suppose we consider c_i to be the positive class and c_j to be the negative class. We use $f_{ij}(x)$ to denote the score of this classifier on the image x. Note that $f_{ij}(x) = -f_{ji}(x)$ for any i and j. Using these binary classifiers, we first define the score of picking c_i as a child to visit: $h_i = \sum_{j:j\in\{1,2...,n\},j\neq i} f_{ij}(x)$. We will visit the child c_i only when h_i is greater than certain threshold T. In our experiment, we choose the threshold as the median of these scores, i.e. $T = \text{median}_{i\in\{1,2,...,n\}} h(i)$. The same procedure is iteratively applied to all child nodes. In the end, we would have traversed a subset of of the K possible paths in the hierarchy. We use the K-class nonlinear SVM (see Sec. 3.2) to obtain a final score for each of the traversed path. The path with the maximum score will give us the final prediction.

For example (see Fig. 2), suppose we have a test image "Collie dog". Starting from root node v_0 = "*carnivore*", we collect the output scores L_0 from the corresponding binary classifiers (Canine-vs-Feline, Canine-vs-Bear, Feline-vs-Bear). Suppose the scores of visiting "Canine" and "Feline" are greater than the threshold T. We will then prune "Bear" and only visit "Canine" and "Feline" at the next level in the hierarchy. We repeat this process. At each visited internal node v_i, we collected scores L_i from its binary classifiers, and pick some children to visit until leaf nodes are reached. At the end, we will have explored more than one path in the hierarchy. Each path will result in a mid-level representation. Then, we feed this representation to the learned non-linear SVM (Sec. 3.2) which gives a score for predicting the class k. After traversing several paths, we will pick the class with the best score from the non-linear SVM as the best predicted class label for the given image.

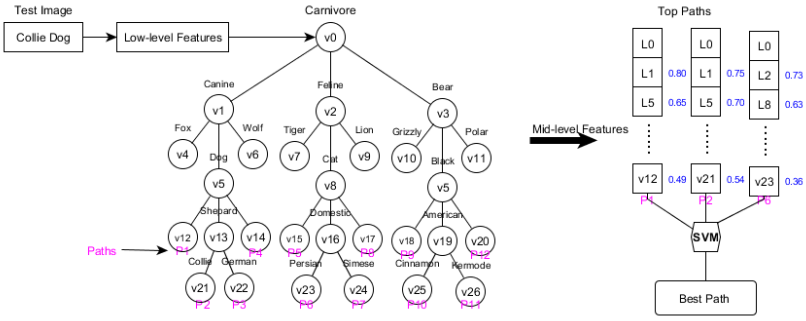

Fig. 2. For each test image, we explore more than one path, and construct more than one mid-level representation. We feed these mid-level representations to the learned non-linear SVM (Sec. 3.2) which predicts the best path of the given image. Please refer to Sec. 3.3 for details.

4 Experiments

4.1 Datasets

In order to evaluate our method, we use the same four datasets that have been used in [13]. Category labels in each of the dataset are organized in a tree-structured hierarchy.

ImageNet65: this dataset consists of subtrees for "plant" "animal" and "vehicles" in the ImageNet hierarchy [5]. There are 39600 images in this dataset correspondining to 65 categories.

Animal-with-Attributes (AwA): this dataset contains 30474 images. Each image belongs to one of the 50 animal categories [14]. The WordNet is used to organize these animal classes into a hierachy.

CIFAR: this dataset consists of 60000 images of animals and vehicles [15]. There are in total 100 object classes organized into a two-layer hierarchy. The hierarchy has 20 internal nodes and 50 leaf nodes.

Yahoo Shoes: this dataset contains 5250 images of shoe images collected by Yahoo [16]. These images are organized into a hierarchy with 10 internal nodes and 107 leaf nodes.

4.2 Experimental results

Figure 3 visualizes the confusion matrices of our method on these four datasets.

We compare our approach with the following three baseline methods:

- Raw features: this baseline method learns a nonlinear kernel SVM based on the raw low-level features.
- Cao et al. [12]: this baseline method considers each pair of leaf nodes and learns a binary classifier. The concatenation of these classifier responses are

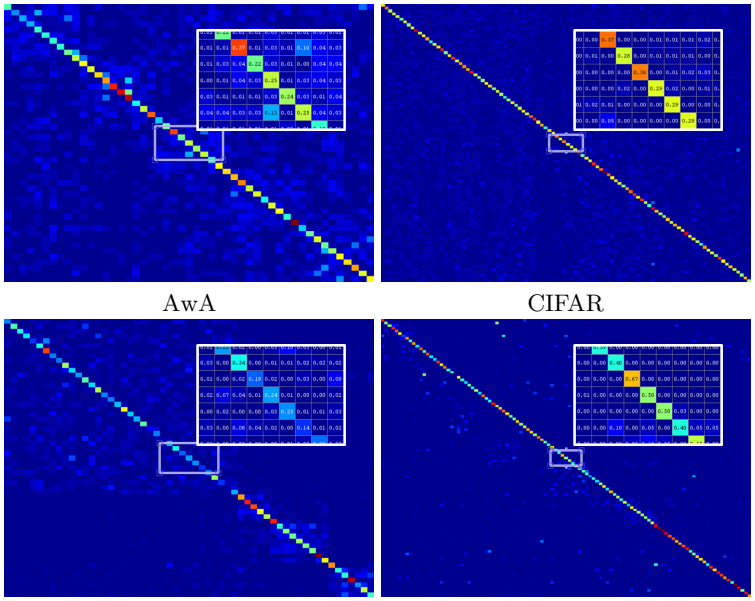

Fig. 3. Confusion matrices of our method on four datasets.

used as mid-level features. This method does not exploit the hierarchical structure of object classes.

– Albaradei et al. [13]: this baseline approach selects concept pairs similarly to our method. But it uses the concatenation of all binary classifiers as the mid-level feature. In other words, the hierarchical structure is used when selecting concept pairs, but not used when constructing the final mid-level image representation.

The mean per-class accuracies of these methods are shown in Table 4.2. We can see that our method performs significantly better than the raw features and the method in [12]. It also outperforms the method in [13]. Compared with [13], our method has the additional advantage that we only need to apply a subset of the binary SVM classifiers on a given image. In Fig. 4, we visualize the average number of nodes visited on test images for both our method and the method in [13]. We can see that our method visit significantly less nodes than [13]. This demonstrates that our pruning strategy is very effective.

To further illustrate the trade-off of accuracy and efficiency of our approach, we also consider the following two baselines.

– Hierachy (single path): this method is essentially the same as Bengio et al. [6]. At each internal node, it chooses only one child to visit. In this end, we will reach a leaf node. This leaf node will give the predicted label. This method is very efficient, since it only needs to explore one single path in the hierarchy.

Table 1. Comparison of overall accuracies of our approach with three baseline methods on four datasets.

Method \ Dataset	ImageNet65	AwA	CIFAR	Yahoo Shoes
Raw features	23.82%	23.10%	25.73%	57.14%
Cao et al. [12]	29.7%	24.5%	28.6%	62.4%
Albaradei et al. [13]	36.21%	27.50%	30.52%	64.73 %
Ours	**37.85%**	**29.30%**	**31.75%**	**64.85%**

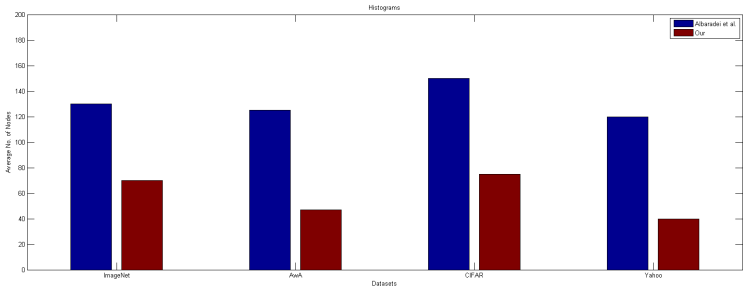

Fig. 4. Comparison of the average number of nodes visited on test images between our method and [13] on the four datasets. Compared with [13], our approach visits significantly less nodes due to the pruning strategy.

– Hierarchy (all paths): this method is an extreme case of our approach, where no children are pruned.

The comparison with these two baselines are shown in Table 4.2. We can see that although exploring a single path is efficient, the performance is much worse. The reason is that if any internal node picks the wrong child to visit, the error cannot be corrected by any decendants. This issue can be addressed by exploring more paths in the hierarchy. Figure 5 shows some predictions of our method and the single path.

Table 2. Comparison of overall accuracies of our approach with two baseline methods on four datasets.

Method \ Dataset	ImageNet65	AwA	CIFAR	Yahoo Shoes
Hierarchy (single path) [6]	28.14%	24.50%	27.41%	59.43%
Hierarchy (all paths)	36.70%	29.30%	31.10%	63.88%
Ours	**37.85%**	**29.30%**	**31.75%**	**64.85%**

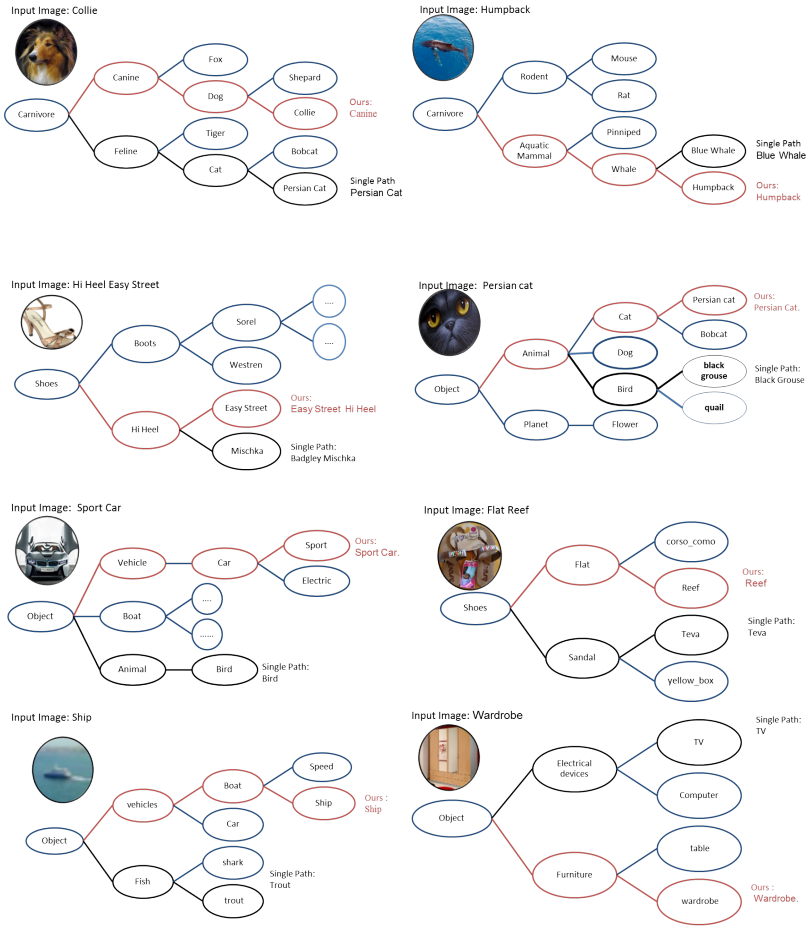

Fig. 5. Some example predictions of our method and the single path method.

4.3 Conclusion and Future Work

In this paper, we have presented a new method for object classification using semantic hierarchy. Our proposed method exploits the semantic hierarchy in two aspects. First, it uses the hierarchy to select concept pairs and learn binary SVM classifiers. Second, it exploits the hierarchy to construct the mid-level representation using the responses of the binary SVM classifiers. Our experimental results show that our approach outperforms other baseline methods. In the future, we would like to extend our work to exploit other semantic relations (e.g. non-tree structures) between object classes.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
2. Li, L.J., Su, H., Xing, E.P., Fei-Fei, L.: Object bank: A high-level image representation for scene classification and semantic feature sparsification. In: Advances in Neural Information Processing Systems. MIT Press (2010)
3. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: European Conference on Computer Vision. Springer (2010) 776–789
4. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2012) 1234–1241
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2009) 248–255
6. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: Advances in Neural Information Processing Systems. (2010) 163–171
7. Deng, J., Satheesh, S., Berg, A.C., Li, F.: Fast and balanced: Efficient label tree learning for large scale object recognition. In: Advances in Neural Information Processing Systems. (2011) 567–575
8. Gao, T., Koller, D.: Discriminative learning of relaxed hierarchy for large-scale visual recognition. In: IEEE International Conference on Computer Vision. (2011)
9. Sun, M., Huang, W., Savarese, S.: Finding the best path: an efficient and accurate classifier for image hierarchies. In: IEEE International Conference on Computer Vision. (2013)
10. Deng, J., Berg, A.C., Fei-Fei, L.: Hierarchical semantic indexing for large scale image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2011) 785–792
11. Deng, J., Krause, J., Berg, A.C., Fei-Fei, L.: Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2012) 3450–3457
12. Cao, L., Gong, L., Kender, J.R., Codella, N.C., Smith, J.R.: Learning by focusing: A new framework for concept recognition and feature selection. In: IEEE International Conference on Multimedia and Expo, IEEE (2013) 1–6
13. Albaradei, S., Wang, Y., Cao, L., Li, J.: Learning mid-level features from object hierarchy for image classication. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision. (2014)
14. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2009) 951–958
15. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25. (2012) 1106–1114
16. Yahoo research labs: Yahoo! shopping shoes image content (2013)