

A Hybrid Architecture Based on CNN for Image Semantic Annotation

Yongzhe Zheng¹, Zhixin Li^{1,2}(✉), and Canlong Zhang^{1,2}

¹ Guangxi Key Lab of Multi-source Information Mining and Security,
Guangxi Normal University, Guilin 541004, China

zhengyzms@163.com, {lizx, clzhang}@gxnu.edu.cn

² Guangxi Experiment Center of Information Science, Guilin 541004, China

Abstract. Due to semantic gap, some image annotation models are not ideal in semantic learning. In order to bridge the gap between cross-modal data and improve the performance of image annotation, automatic image annotation has become an important research hotspots. In this paper, a hybrid approach is proposed to learn automatically semantic concepts of images, which is called Deep-CC. First we utilize the convolutional neural network for feature learning, instead of traditional methods of feature learning. Secondly, the ensembles of classifier chains (ECC) is trained based on obtained visual feature for semantic learning. The Deep-CC corresponds to generative model and discriminative model, respectively, which are trained individually. Deep-CC not only can learn better visual features, but also integrates correlations between labels, when it classifies images. The experimental results show that this approach performs for image semantic annotation more effectively and accurately.

Keywords: Semantic learning · Image auto-annotation · Convolutional neural network

1 Introduction

In the past decades, several state-of-the-art approaches have been proposed to solve the problems of automatic image annotation, which can be roughly categorized into two different models. The first one is based on generative model. The auto-annotation is first defined as a traditional supervised classification problem [1, 7], which mainly depends on similarity between visual features and predefined tags to model the classifier, then a unknown image is annotated relevant tags by computing similarity of visual level. The other is based on discriminative model, which are treat image and text as equivalent data. These methods try to mine the correlation between visual features and labels on an unsupervised basis by estimating the joint distribution of multi-instance features and words of each image [7, 16]. In brief, these methods extract various low-level visual features. These approaches greatly reduces the ability of feature presentation, therefore it makes the semantic gap become more serious between image and semantic.

Furthermore, the performances of image annotation are highly dependent on the representation of visual feature and semantic mapping. In view of the fact that deep

convolutional neural networks (CNNs) has been demonstrated a outstanding performance in computer vision recently. Besides, Mahendran and Vedaldi [3, 4, 9] and [11] have demonstrated that CNN has a better effect over existing methods of hand-crafted features in many applications, such as object classification, face recognition, and image annotation. Inspired these articles, this paper proposes a hybrid architecture based on CNN for image semantic annotation to improve the performances of image annotation.

In this paper, our main contributions are the following. Firstly, we use redesigned CNN model to learn high-level visual features. Secondly, we employ the ensembles of classifier chains (ECC) to train model on visual features and predefined tags. Finally, we propose a hybrid framework to learn semantic concepts of images based CNN (Deep-CC). Deep-CC not only can learn better visual features, but also integrates correlations between labels when it classifies images. The experimental results show that our approach performs more effectively and accurately.

2 CNN Visual Feature Extraction

In the past few years, some recent articles [14, 17] have demonstrated that the CNN models pre-trained on large datasets with data diversity, e.g., AlexNet [4] which can be directly transferred to extract CNN visual features for various visual recognition tasks such as image classification and object detection. CNN is a special form of neural network that consists of different types of layers, such as convolutional layers, spatial pooling layers, local response normalization layers and fully connected layers. Different network structures will show different ability of visual features representation. Krizhev et al. [4] have proved that the Rectified Linear Units (ReLU) not only saves the computing time, but also implements the features of sparse representation, and ReLU also increases the sample characteristic diversity. So in order to improve the generalization ability of the feature representation, we extract *fc7* visual vectors after ReLU. As shown in the top of the Fig. 1, our CNN model has the similar network structure to the AlexNet. As reflected in Fig. 1, which contains five convolutional layers (short as conv) and three fully-connected layers (short as fc). The CNN model is pre-trained in 1.2 million images of 1000 categories from ImageNet [14].

2.1 Extracting Visual Features from Pre-trained CNN Model

Li and Yu [5] and Razavian et al. [12] have demonstrated the outstanding performance of the off-the-shelf CNN visual features in various recognition tasks, so we utilize the pre-trained CNN model to extract visual features. Particularly, each image is resized to $227 * 227$ and fed into the CNN model. As shown in Fig. 1, it represents the feature flow extracted from the convolutional neural network. The *fc7* features are extracted from the secondly convolution layer after ReLU. The *fc7* denote the 4096 dimensional features of the last two fully-connected layers after the rectified linear units (ReLU) [4].

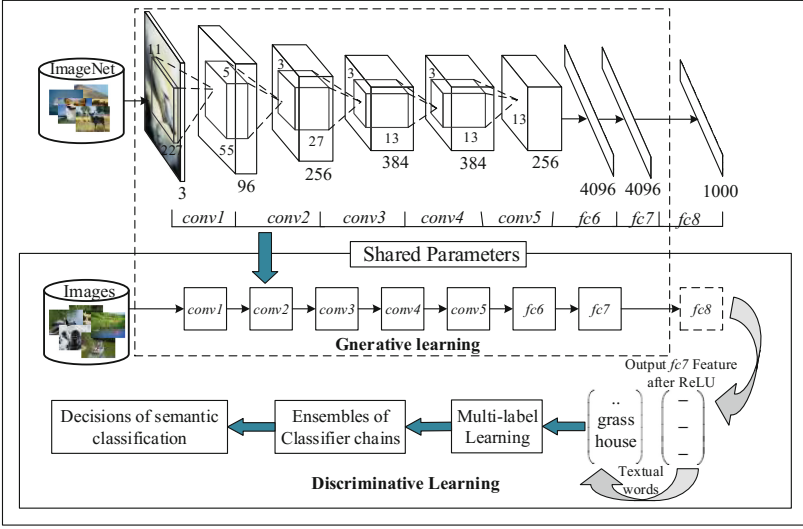


Fig. 1. The Pipeline of Image Annotation

2.2 Extracting Visual Feature from Fine-Tuned CNN Model

Taking into account the different categories (and the number of categories) between the target dataset and ImageNet, if we directly utilize the pre-trained model on the ImageNet to exact image visual features, it may not be the optimum strategy. To make the model fit the parameters better, we redesign the last hidden layer for visual feature learning task, later re-designed CNN model by fine-tuning parameters with each of images in the target dataset. Considering the rationality of the design of the convolutional neural networks, our CNN model has the similar network structure to the AlexNet. As show in the mid of Fig. 1, the overall architecture of our CNN model still contains five conv layers, followed by a pooling layer and three fully-connected layers. We redesign the last hidden layer for feature learning task. The number of neural units of the last fully-connected layer is modified from 1000 to m , where m is the number of the target dataset's categories. The output of the last fully-connected layer is then fed into a m -way softmax which produces a probability distribution over m categories.

Given one training sample x , the network extracts layer-wise representations from the first conv layer to the output of the last fully connected layer $fc8 \in \mathbb{R}^m$, which can be viewed as high level features of the input image. Followed by a softmax layer, $fc8$ is transformed into a probability distribution $p \in \mathbb{R}^m$ for objects of m categories, and cross entropy is used to measure the prediction loss of the network. Specifically, we define the following formula.

$$p_i = \frac{\exp(\hat{v}_i)}{\sum_i \exp(\hat{v}_i)} \text{ and } L = t_i \log(p_i) \quad (1)$$

In formula (1), L is the loss of cross entropy. The gradients of the deep convolutional neural network is calculated via back propagation

$$\frac{\partial L}{\partial \hat{h}_i} = p_i - t_i \quad (2)$$

In formula (2), $t = \{t_i | t_i \in \{0, 1\}, i = 1, \dots, m, \sum_{k=1}^m t_k = 1\}$ denotes the true label of the sample x_j , where the $\{x_j | j = 1, 2, \dots, n\}$ is a bag of instances. $t = \{t_i | t_i \in \{0, 1\}, i = 1, \dots, m.\}$ is the label of the bag; Convolutional neural network extracts representations of the bag, it can get a feature vector $v = \{v_{ij}\} \in \mathbb{R}^{m \times n}$, in which each column is the representation of an instance. The aggregated representation of the bag for visual vectors are defined as follows.

$$\hat{v}_i = f(v_{i1}, v_{i2}, \dots, v_{in}) \quad (3)$$

In the training phase, similarly back propagation algorithm is used to optimize the loss function L . Suppose that we have a set of training images $I = \{M_i\}$. The trained instances of traditional supervised learning in which training instances are given as pairs $\{(m_i, l_i)\}$, where $m_i \in \mathbb{R}^m$ is a feature vector and $l_i \in \{0, 1\}$ is the corresponding label. In visual feature learning, trained sample is regarded as bags $\{I_i\}$, and there are a number of instances x_{ij} in each bag. Finally, the network extracts layer-wise representations from the first conv layer to the output of the last fully connected layer visual vectors v_i , which can be viewed as high level features of the input image. By fine-tuning like this, the parameters can better adapt to the target dataset by rectifying the transferred parameters. For the task of visual feature learning, we first employ existing model to fine-tune the parameters in the target dataset, then we apply the fine-tuned CNN model to learn image visual features. Similarly, the FT-*fc7* denotes the 4096 dimensional features of the last two fully-connected layers after the rectified linear units (ReLU).

3 Ensembles of Classification Classifiers for Semantic Learning

In the discriminative learning phase, the ensemble of classification classifiers (ECC) [13] are used to accomplish the task of multi label classification, and each of the binary classifier is implemented by SVM. Taking into account the semantic correlations between tags, ECC can classify images into multiple semantic classes, with a high degree of confidence and acceptable computational complexity. Furthermore, by learning the semantic relevance between labels, classifier chain can effectively overcome the problems of label independence in image binary classification.

The classifier chain model consists of $|L|$ binary classifiers, where L denotes the truth label set. Classifiers are linked along a chain where each classifier deals with the binary relevance problem associated with label $l_j \in L$. The feature space of each linked in the chain is extended with the $\{0, 1\}$ label associations of all previous links. The training

procedure is outlined in Algorithm 1 in the left of Table 1. Lastly, we can note the notation for a training example (x, S) , where $S \subseteq L$ and x is an instance feature vector.

Hence a chain C_1, C_2, \dots, C_i of binary classifier is formed. Each classifier C_j in the chain is responsible for learning and predicting the binary association of label l_j , which is given in the feature space and is augmented by all prior binary relevance predictions in the chain $\{l_1, l_2, \dots, l_{j-1}\}$. The classification procedure begins at C_1 and propagates along the chain C_1 determines $\Pr(l_1|x)$ and every following classifier C_2, \dots, C_j predicts $\Pr(l_j|x_i, l_1, l_2, \dots, l_{j-1})$. This classification procedure is described in Algorithm 2 in the right of Table 1.

This training method passes label information between classifiers, with classifier chain taken into account label correlations, so it overcomes the label independence problem of binary relevance method. However, classifier chain still remains advantages of binary relevance method including low memory and runtime complexity. Although $|L|/2$ features are added to each instance on an average, this item is negligible in computational complexity because $|L|$ is invariably limited in practice.

Different order of the chain clearly has a different effect on accuracy. This problem can be solved by using an ensemble framework with a different random train ordering for each iteration. Ensembles of classifier chains train m classifier chains C_1, C_2, \dots, C_m . Each C_k model is trained with a random chain which can order the L outputs and get a random subset of D . Hence each C_k model is likely to be unique and able to give different multi-label predictions. These predictions are summed by label so that each label receives a number of votes. A threshold is used to select the most popular labels which form the final predicted multi-label set. These predictions are summed by label so that each label receives a number of votes. A threshold is used to select the most popular labels which form the final predicted multi-label set.

Table 1. Training and prediction procedures of ensembles of classifier chains for multi-label learning

Processing	Algorithm 1. Training steps of classifier chain	Algorithm 2. Classifying procedure ECC
Input	Training set $I = \{(x_1, S_1), (x_2, S_2), \dots, (x_n, S_n)\}$	Test example x .
Output procedures	Classifier chains $\{C_1, C_2, \dots, C_{ L }\}$	$Y = \{l_1, l_2, \dots, l_{ L }\}$.
1	For $i \in 1, 2, \dots, L $	$Y \leftarrow \{\}$
2	Semantic learning	For $i \in 1, 2, \dots, L $
3	$I' \leftarrow \{\}$	Do $Y \leftarrow Y \cup (l_i \leftarrow C_i: (x_i, l_1, l_2, \dots, l_{j-1}))$
4	For $(x, S) \in I$	Return (x, Y)
5	Do $I' \leftarrow I' \cup ((x, l_1, l_2, \dots, l_{i-1}), l_i)$	
6	Train C_i to predict binary relevance of l_i	
7	$C_i: I' \rightarrow l_i \in \{0,1\}$	

Each k th individual model predicts vector $y_k = (l_1, l_2, \dots, l_{|L|}) \in \{0, 1\}^{|L|}$. The sums are stored in a vector $W = (\lambda_1, \lambda_2, \dots, \lambda_{|L|}) \in \mathbb{R}^{|L|}$, where λ_j is defined as $\lambda_j = \sum_{k=1}^m l_j \in y_k$. Hence each $\lambda_j \in W$ represents the sum of the votes for label $l_j \in L$. We then normalize W to W_{norm} , which represents a distribution of scores for each label in $[0,1]$. A threshold is used to choose the final multi-label set Y such that $l_j \in Y$ where $\lambda_j \geq t$ for threshold t . Hence the relevant labels in Y represent the final multi-label prediction.

4 Hybrid Framework for Image Annotation

On the deep model and ensembles of classifier chains, we propose a hybrid learning framework to address cross-modal semantic annotation problem between images and text with Multi-label. Figure 2 shows two setups of the hybrid architecture approach for semantic learning based on deep learning. The first path (generative learning) feeds training image to the fine-tune pre-trained CNN step which is also called the feature learning phase, then in the discriminative learning phase, we utilize ensemble of classifier chains to model the visual vectors which are co-occurrence matrix consisting of texture and exacted visual features by pre-trained CNN model. This hybrid pipeline model is called Deep-CC image annotation system.

Bases on the learning feature, the trained CNN model output visual features after ReLU. Suppose that we have a set of images $M = \{m_1, m_2, \dots, m_i\}$, this model extracts visual vectors by pre-trained CNN model and we denote the space of visual vectors as $V = \{v_1, v_2, \dots, v_i\}$, where v_i denotes the visual vector of i th image. Noting the notation for a training example (v_i, S) , where $S \in L$, L denotes the label set and v is a feature vector. Then, by making use of the aspect distribution and original labels of each training image, we build a series of classifiers in which every word in the vocabulary is treated as an independent class. The classifier chain model implements the feature classification task and it can effectively learn the semantic correlation between labels in discriminative step. Finally, given a test image, the Deep-CC system will return a correlative label subset $l \in L$.






Image					
Ground Truth	temple, sky, buddhist, mountains	elephant, trees, planes, sky	cabin, trees, autumn, field	trees, sky, road, park	people, water, trees, sand
HGDM annotations	house, sky, water, cloud, mountains	Africa, sky, land, animal, beach	field, grass, land, trees, mountains	trees, road, sky, pant, sea	sand, water, tress, people, cloud
Deep-CC annotations	temple, sky, palace, land, mountains	elephant, land, plane, trees, sky	chair, trees, field, land, mountains	park, road, trees, sky, mountains	sand, coast, trees, water, people

Fig. 2. Comparison of annotations made by HGDM and Deep-CC on Corel5k

As a comparison, we evaluate the deep feature’s performance from the AlexNet CNN on those same benchmarks. Following by [10], we choose 5 words with highest confidence as annotations of the test image. After each image in the database is annotated, the retrieval algorithm ranks the images labeled with the query word by decreasing confidence.

5 Experiments and Results

In this section, we conduct experiments of our Deep-CC learning framework on both image classification and image auto-annotation. We choose a dataset Core5K which is widely used in image classification and annotation. In order to make the experimental result more convinced, we simultaneously compare the experimental results with the existing traditional model and deep model.

5.1 Datasets and Evaluation Measures

In order to test the effectiveness and accuracy of the proposed approach, we conduct our experiments on a baseline annotated image datasets Core5K [2]. Core5k is a basic comparative dataset for recent research works on image annotation. The dataset contains 5000 images from 50 Corel Stock Photo cds. We divided this dataset into 3 parts: a training set of 4000 images, a validation set of 500 images and a test set of 500 images. Like the Duygulu et al. [2], We divide separately the training set of 4500 images and the test set of 500 images.

Image annotation performance is evaluated by comparing the captions automatically generated for the test set with the human-produced ground truth. It is essential to include several evaluation measures in multi-label evaluation. Similar to Monay and Gatica-Perez [10], we use mAP as evaluation measures. Naturally, we define the automatic annotation as the top 5 semantic words of largest posterior probability, and compute the recall and precision of every word in the test set.

5.2 Results for Image Annotation on Core5 K

In this section, we demonstrate the performance of our model on the core5 k data set for image multi-label annotation, and compare the results with some existing image annotation methods, e.g. PLSA-WORDS [10], HGMD [8] and DNN [15]. We evaluate the returned keywords in a class-wise manner. The performance of image annotation is evaluated by comparing the captions automatically generated with the original manual annotations. Similar to Monay and Gatica-Perez [10], we compute the recall and precision of every word in the test set and use the mean of these values to summarize the system performance.

Table 2 reports results of several models on the set of all 260 words which occur in the training set. Data in precision and recall columns denotes mean precision and mean recall of each word. The off-the-shelf CNN features (i.e. fc7 and FT-fc7) obtain significant improvements (7.8 % based on PLSA-WORDS, 3.4 % based on HGDM)

compared with these traditional feature learning methods. After fine-tuning, a further improvement (8.2 % based on PLSA-WORDS, 4.6 % based on HGDM) can be achieved with the best performance of the CNN visual features FT-fc7.

Annotations of several images obtained by our Deep-CC annotation system are show in Fig. 2. We can see that annotations generated by Deep-CC are more accurate than HGDM in most cases. In order to be more intuitive to observe different precision-recall in various methods, the Fig. 3 presents the precision-recall curves of several annotation models on the Corel5k data set. As is shown in Fig. 3, Deep-CC performs consistently better than other models. Where the precision and recall values are the mean values calculated over all words.

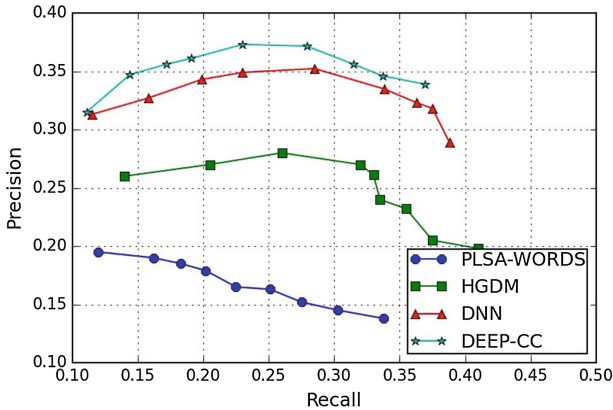


Fig. 3. Precision–recall curves of several models for image annotation on Corel5K

5.3 Result Analysis

In summary, the experimental results on Corel5k show that Deep-CC outperforms many state-of-the-art approaches, which proves that the redesigned CNN and the hybrid framework is effective in learning visual features and semantic concepts of images. We compare the CNN visual features with traditional visual features for learning semantic concepts of images over two traditional learning approaches and a deep model. Especially, the comparison in terms of rigid and articulated visual features among Corel5k is shown in Table 2, from which it can be seen that CNN feature outperforms almost all the original hand-crafted features. To verify this assumption, different visual features between traditional models (also from the authors of this paper) and CNN mode, and FT-fc7 is executed to make an enhanced prediction for Corel5k. Incredibly, the mAP score on Corel5k can surge to 35.2 % as shown in Table 2, which demonstrates the great dominance in the deep networks. To sum up, based on the above reported experimental results, we can see that CNN visual features are very effective for semantic image annotation.

Table 2. Performance (mAP in %) comparison in terms of different methods and visual features algorithms on Corel5k. (The “-” means to use their method)

Method	Visual features	Result on all words		mAP
		Precision	Recall	
PLSA-WORDS	-	22.1	12.1	19.1
	Fc7	27.5	21.7	26.9
	FT-fc7	29.3	22.6	27.3
HGDM	-	32.1	29.3	26.3
	Fc7	36.4	30.5	29.7
	FT-fc7	37.6	32.9	30.9
DNN	-	37.5	40.5	32.7
Deep-CC(our)	-	39.7	37.6	35.2

6 Conclusion

In this paper, we utilize CNN model to learn deep visual features, and we redesign the last hidden layer for feature learning task, and in order to obtain high performance of feature representation, we first train our deep model on ImageNet, then the pre-trained parameters are fine-tuned on target dataset. We showed under what conditions each visual feature can perform better, and propose a hybrid architecture. We demonstrated that re-designed CNN model and ensembles of classifier chains can effectively improve annotation accuracy.

In comparison to many state-of-the-art approaches, experimental results show that our method achieves superior results in the tasks of image classification and annotation on Corel5K. However, in the process of learning visual features, Deep-CC only employ single convolution neural network not fully understanding multiple instance in the image, and how to excavate the high-level semantic relevance between the tags, it can be deeply studied. In future research, we aim to take semi-supervised learning based on a large number of unlabeled data to improve its effectiveness.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (Nos. 61663004, 61262005, 61363035), the Guangxi Natural Science Foundation (2013GXNSFAA019345, 2014GXNSFAA118368), the Guangxi “Bagui Scholar” Teams for Innovation and Research Project, Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

References

1. Cusano, C., Ciocca, G., Schettini, R.: Image annotation using SVM. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 5304, pp. 330–338 (2003)
2. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)

3. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2016)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, vol. 25, no. 2 (2012)
5. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2015)
6. Li, J., Wang, J.Z.: Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Machine Intell.* **25**(9), 1075–1088 (2003)
7. Liu, Y., Zhang, D., Lu, G., et al.: A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.* **40**(1), 262–282 (2007)
8. Li, Z., Shi, Z., Zhao, W., et al.: Learning semantic concepts from image database with hybrid generative/discriminative approach. *Eng. Appl. Artif. Intell.* **26**(9), 2143–2152 (2013)
9. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. *Comput. Sci.*, 5188–5196 (2015)
10. Monay, F., Gatica-Perez, D.: Modeling semantic aspects for cross-media image indexing. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(10), 1802–1817 (2007)
11. Mnih, V., Kavukcuoglu, K., Silver, D., et al.: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015)
12. Sharif Razavian, A., Azizpour, H., Sullivan, J., et al.: CNN features off-the-shelf: an astounding baseline for recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 512–519. IEEE Computer Society (2014)
13. Read, J., Pfahringer, B., Holmes, G., et al.: Classifier chains for multi-label classification. *Mach. Learn.* **85**(3), 254–269 (2011)
14. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
15. Sermanet, P., Eigen, D., Zhang, X., et al.: OverFeat: integrated recognition, localization and detection using convolutional networks. *Eprint Arxiv* (2013)
16. Smeulders, A.W.M., Worring, M., Santini, S., et al.: Content-based image retrieval at the end of the early. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(12), 1349–1380 (2000)
17. Wu, J., Yu, Y., Chang, H., et al.: Deep multiple instance learning for image classification and auto-annotation. In: *Computer Vision and Pattern Recognition*, pp. 3460–3469. IEEE (2015)