MIL at ImageCLEF 2014: Scalable System for Image Annotation

Atsushi Kanehira, Masatoshi Hidaka, Yusuke Mukuta, Yuichiro Tsuchiya, Tetsuaki Mano, and Tatsuya Harada

Machine Intelligence Lab., The University of Tokyo
{kanehira,hidaka,mukuta,tsuchiya,mano,harada}@mi.t.u-tokyo.ac.jp
http://www.mi.t.u-tokyo.ac.jp

Abstract. In this working note, we describe details of our method in ImageCLEF2014 Scalable Concept Image Annotation task. We are given images from the Web and some additional information, including web pages, in which images exist. Using this information, we must construct an annotation system that has high performance and scalability. To assign labels to each image, we use the page title and attributes of an image tag extracted from the web page. As visual features, we propose the use of the combination of two complementary features, which are Fisher Vector and deep convolutional neural network based feature. They are generative and discriminative feature respectively. We then train linear classifiers using Passive–Aggressive with Averaged Pairwise Loss. After training, we calculate the score of each concept for test data and label some concepts having the best scores. Results show that the combination of two features contributes to the improvement of recognition performance.

Keywords: ImageCLEF, deep convolutional neural network, Fisher vector, Image annotation

1 Introduction

For the ImageCLEF2014 Scalable Concept Image Annotation task [1][2], our task is to construct an image annotation system that yields high-performance with scalability.

As visual features, we use a convolutional neural network (CNN) based feature as well as the Fisher Vector (FV) [3]. For scalability, our method of assigning labels to training images is simple. We use only the page title and attributes of the image tag extracted from the web page. To train linear classifiers, we use Passive–Aggressive with Averaged Pairwise Loss (PAAPL) [4] because of its scalability and robustness to noise of label assignment.

In our experiment, we combine two types of features, which are FV and CNN-based features. Actually, FV is often used in image recognition tasks because of its recognition performance. However, many results of recent studies show that deep CNN achieves high performance on many tasks. Therefore, we expect that

the feature, which is the neuron activation pattern in the hidden layers of the network, has high-representational ability.

These two features are extracted in completely different ways. We obtain FV by coding local descriptors considering their probabilistic distribution. Therefore FV can be regarded as the feature expressing generative information of an image. In contrast to FV because deep CNN based feature is extracted from the network trained for recognition task, we can regard it as a discriminative feature.

Assuming that these two types of features, which represent different kinds of information, mutually compensate for representational ability, we propose their combined use. Our contribution is the usage of a combination of features that have complementary properties to improve the performance of annotation systems.

The remainder of this working note is structured as follows. Section 2 presents a description of two types of visual features: FV and deep CNN based features. In Section 3, we explain details of how we obtain labels from training data. Then, in section 4, we introduce a multi-label linear classifier training method: PAAPL. In section 5, we present the results of experiments, using either or both of these visual features. Finally, in section 6, we discuss the analysis of the results obtained in our experiment.

2 Visual Feature

2.1 Fisher Vector

As a visual feature, we use Fisher Vector (FV) because FV can achieve better recognition performance than Bag of Visual Words with a linear classifier. In general, the linear classifier is less costly than a nonlinear one such as kernel-SVM when the amount of the training sample increases. Therefore, FV is suitable for this task, which requires scalability.

In our experiments, we extract four local descriptors: SIFT, GIST, LBP, and C-SIFT. The dimensions of all these local descriptors are reduced to 64 dimensions using Principal Component Analysis (PCA). These local descriptors are densely extracted from five scales of patches (squares 16, 25, 36, 49, 64 pixels on a side) sliding with a step of six pixels. Using some of the obtained local descriptors, we train a Gaussian Mixture Model (GMM) with 256 components, which have diagonal matrices as covariance matrices. After training GMM, we extract FV from each image by calculating the gradient of log-likelihood of local descriptors with respect to parameters of GMM. Then we normalize it using the Fisher information matrix. Power normalization and L2 normalization are applied to the extracted FVs. To include spatial information, we divide images into 1×1, 2×2, and 3×1 cells, extract features from each region and concatenate them into one vector. The final dimension of FV is 262,144 ($64 \times 256 \times 2 \times 8$).

As described above, FV is obtained by coding local descriptors considering their probabilistic distribution. Therefore FV can be regarded as a feature expressing generative information of the image.

2.2 Deep CNN based feature

In addition to FV, we use a deep convolutional neural network (CNN) based feature extracted from a deep CNN model that had been pre-trained with ImageNet dataset.

In recent years, many studies that specifically examine deep CNN have shown that such models can perform better than conventional feature representation in object recognition and other tasks. On ImageNet Large Scale Visual Recognition Challenge, one system [5] dramatically outperformed all other methods including a state-of-the-art method using FV.

However, training deep architecture of CNN requires large-scale data to prevent overfitting and to ensure generalization ability. Trained models will have low recognition performance if the training data are few.

For this task, because we must label training data automatically, the number of reliably labeled data we can obtain using our method is roughly 100,000 on the development set and about 200,000 on the test set. This fact implies that the approximate number we can use in training models is only 1,000 per concept, on average. Therefore, because of limitations in the amount of data in the given dataset, it is difficult to train deep CNN models to produce high recognition performance.

According to [6], features extracted from the activation of a CNN pre-trained in supervised fashion can be re-purposed to generic tasks. In his experiment, he shows that such feature representation has such high generality that it outperforms conventional methods in several tasks despite their simple training algorithm.

In consideration of the discussion presented above, we use feature representations extracted from a deep CNN model pre-trained with the ImageNet dataset. Following the method of [6], we extract features from sixth and seventh layers of the network having architecture that is the same as that proposed by [5], which won ILSVRC2012 and which includes five convolutional and three fully connected layers. In some layers, the Hinge function is used as the activation function. It is designated as ReLU.

Contrary to FV, because deep CNN-based features are extracted from the network, which is trained for recognition task, we can regard it as a feature that expresses discriminative information of an image.

In our experiment, we use DeCAF, an open source library produced by [6], to extract deep feature representation. We use features of four types. The feature vector dimensions are 4,096.

3 Label Assignment

Because no explicit labels are assigned to training images, we must label them using additional information or external sources. In this section, we describe how to assign labels to images. The pipeline of label assignment is shown in Fig.1. We label images in the following two steps.

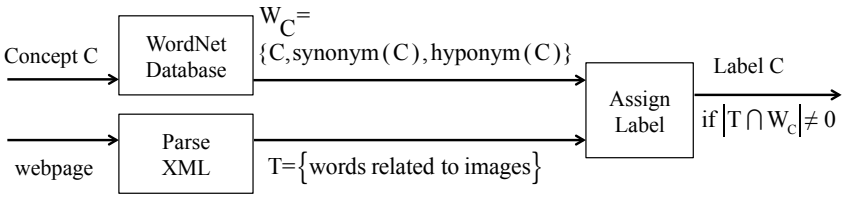

Fig. 1. Pipeline of label assignment.

In the first step, we parse the xml files of the web page, in which an image exists. Then we extract page titles and attributes of the image tag, which include src, title, and alt. The hope is that these attributes include important information about what the image represents. Then we split them into a set of single words T. For example, if there is an image tag in an xml file shown in Fig.2, then we obtain

$$T = \{\text{Queen}, \text{Prince}, \text{corgi}, \text{family}, \text{abroad}\}.$$

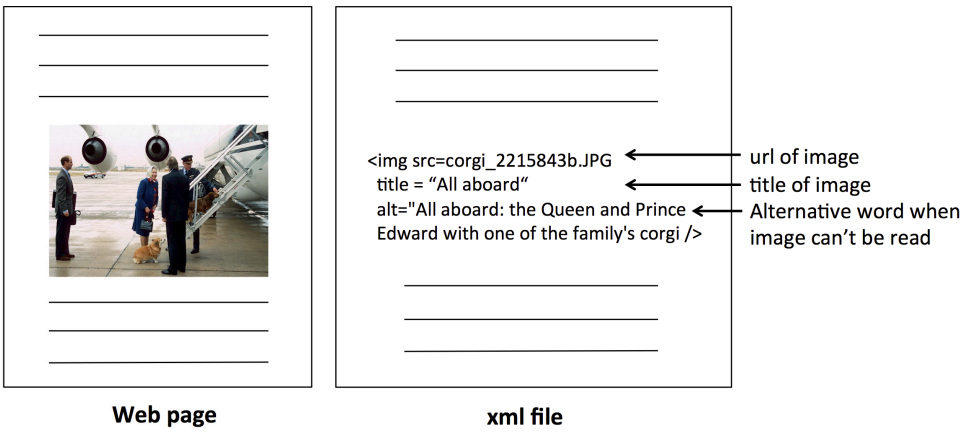

Fig. 2. Attributes of image tag.

In the second step, we collect a set of synonyms and hyponyms for each concept C using WordNet [7]. We denote the collected sets by W_C, which is expressed as

$$W_C = \{C, \text{synonym}(C), \text{hyponym}(C)\}$$

where synonyms (C) and hyponyms (C) respectively represent sets of synonyms and hyponyms of the concept C. For example, given a target concept "dog", we obtain

$$W_{dog} = \{\text{dog}, \text{puppy}, \text{corgi}, ...\}.$$

We assign the concept C as the label to the image if at least one word in W_C appears in T.

4 Training Classifier

In this section, we introduce the multi-label linear classifier training method Passive–Aggressive with Averaged Pairwise Loss (PAAPL) [4]. Because PAAPL is based on Passive–Aggressive (PA) [8] method, which is known to be robust to outliers, PAAPL also has robustness to outliers. In addition, PAAPL has scalability because the trained classifier is linear. Such properties of PAAPL are suitable for our task, in which it is necessary to construct a scalable system that handles data including some outliers.

First, we describe the model update rule of PA. Given the t-th training sample, we designate the visual feature by $\boldsymbol{x}_t$. We define Y_t as the set of labels assigned to the t-th sample, and $\bar{Y}_t$ as the set of labels not assigned. The model (weight) of the linear classifier corresponding to concept label C before updating by the t-th sample is denoted by $\boldsymbol{w}_t^C$.

1. Fetch the t-th training sample. Then compute scores for each label using current models. Because classifiers we are training are linear, scores are given by simple calculation of the inner product of weight and feature.
2. Based on scores, find a combination of labels $r_t \in Y_t$ and $s_t \in \bar{Y}_t$ in the following way.

$$r_t = \arg\min_{r \in Y_t} \boldsymbol{w}_t^r \cdot \boldsymbol{x}_t$$
$$s_t = \arg\max_{s \in \bar{Y}_t} \boldsymbol{w}_t^s \cdot \boldsymbol{x}_t$$

3. For a combination of r_t and s_t, compute the hinge-loss l

$$l(\boldsymbol{w}_t^{r_t}, \boldsymbol{w}_t^{s_t}; (\boldsymbol{x}_t, Y_t)) = \begin{cases} 0 & if\ \boldsymbol{w}_t^{r_t} \cdot \boldsymbol{x}_t - \boldsymbol{w}_t^{s_t} \cdot \boldsymbol{x}_t > 1 \\ 1 - (\boldsymbol{w}_t^{r_t} \cdot \boldsymbol{x}_t - \boldsymbol{w}_t^{s_t} \cdot \boldsymbol{x}_t) & \text{otherwise} \end{cases}$$

4. Update models using hinge-loss according to the following rule.

$$\boldsymbol{w}_{t+1}^{r_t} = \boldsymbol{w}_t^{r_t} + \frac{l}{2|\boldsymbol{x}_t|^2 + \frac{1}{D}} \boldsymbol{x}_t$$
$$\boldsymbol{w}_{t+1}^{s_t} = \boldsymbol{w}_t^{s_t} - \frac{l}{2|\boldsymbol{x}_t|^2 + \frac{1}{D}} \boldsymbol{x}_t$$

Therein, D is a Passive–Aggressive parameter that reduces the negative influence of noisy labels.

Then we describe the method of training classifiers with PAAPL.

1. Pick the t-th training sample, compute scores for target labels using current models.
2. For a randomly selected combination of labels $r_t \in Y_t$ and $s_t \in \bar{Y}_t$, hinge-loss is calculated as PA and remove r_t and s_t from Y_t and $\bar{Y}_t$. Continue this process until $|Y_t| = 0$ or $|\bar{Y}_t| = 0$.
3. For combinations satisfying the condition that the hinge-loss is not 0, update the models according to the update rule of PA.

In PAAPL, convergence of models is faster than in PA because PAAPL updates multiple pairs of models for one sample, whereas PA updates only one pair of models.

5 Results

At the training phase, we first extract visual features and assign labels. Then, linear classifiers are trained. At the test phase, we calculate scores for test images using the linear classifiers we trained. The concepts are labeled on those images. When training classifiers, the number of iterations is set to 5 and Passive Aggressive parameter D is set to 1.0×10^5. After the training process, we average scores from different models trained using different visual features. Also, we decide concepts by selecting those with scores in the top 4% of all given concepts to each test sample. In our experiments, we compare two types of visual features. The settings, except for the combinations of visual features, are fixed throughout all of our experiments.

First, for each type of feature, we respectively search for the best combination. For FV, we use four local descriptors, SIFT, C-SIFT, GIST, and LBP. Because the respective properties of these four features differ, we try all possible combinations of them. As for deep CNN-based features, we extract them from sixth and seventh layers. From each layer, we obtain features of two types, such as activations and outputs of each unit. Therefore we also obtain four types of visual features. In contrast to the case of FV, we need not try all possible combinations because combinations of features from the same layer do not make sense theoretically: they are expected to have similar properties. As shown in Tables Table 1 and Table 2, in both cases, the results of combining all features achieves higher performance than the others. These results respectively correspond to our Run 1 and Run 2.

Finally, we combine these two types of visual features. Using the information presented above, we combine all four FVs and four deep CNN based features. The final results are presented in a Table in Table 3, which correspond to all Runs we submitted. We achieved better performance using both features than by using either one.

As a result, we achieved the second score among all participants with our best run.

Table 1. Results of score combinations (FV).

C-SIFT	GIST	LBP	SIFT	MF-samples
✓	-	-	-	0.286
-	✓	-	-	0.292
-	-	✓	-	0.284
-	-	-	✓	0.294
✓	✓	-	-	0.329
✓	-	✓	-	0.325
✓	-	-	✓	0.330
-	✓	✓	-	0.328
-	✓	-	✓	0.332
-	-	✓	✓	0.324
✓	✓	✓	-	0.347
✓	✓	-	✓	0.350
✓	-	✓	✓	0.348
-	✓	✓	✓	0.344
✓	✓	✓	✓	**0.356**

Table 2. Results of score combinations (deep CNN).

sixth (ReLU)	sixth	seventh (ReLU)	seventh	MF-samples
✓	-	-	-	0.325
-	✓	-	-	0.348
-	-	✓	-	0.346
-	-	-	✓	0.360
✓	-	✓	-	0.358
-	✓	-	✓	0.371
✓	-	-	✓	0.356
-	✓	✓	-	0.366
✓	✓	✓	✓	**0.373**

Table 3. Score combinations of two types of visual feature. Each row corresponds to results we submitted on this task.

Run	4 FVs	4 CNNs	MF-samples (devel)	MF-samples (test)
1	✓	-	0.356	0.240
2	-	✓	0.373	0.265
3	✓	✓	**0.394**	**0.275**

6 Conclusion

In this working note, we described our annotation method for the ImageCLEF 2014 Scalable Concept Image Annotation task. As visual features, we used FV and deep CNN based feature. Assuming that these two types of features mutually express different kinds of information complementarily, we tried combining them. In our experiment, we showed how the combination of features contributes to the improvement of recognition performance. Results show that the combination of generative features and discriminative features proved effective in image recognition tasks.

References

1. Barbara Caputo, Henning Müller, Jesus Martinez-Gomez, Mauricio Villegas, Burak Acar, Novi Patricia, Neda Marvasti, Suzan Üsküdarlı, Roberto Paredes, Miguel Cazorla, Ismael Garcia-Varea, and Vicente Morell. ImageCLEF 2014: Overview and analysis of the results. In *CLEF proceedings*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014.

2. Mauricio Villegas and Roberto Paredes. Overview of the ImageCLEF 2014 Scalable Concept Image Annotation Task. In *CLEF 2014 Evaluation Labs and Workshop, Online Working Notes*, 2014.
3. F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. *European Conference on Computer Vision*, 2010.
4. Y. Ushiku, T. Harada, and Y. Kuniyoshi. Efficient image annotation for automatic sentence generation. *The 20th ACM International Conference on Multimedia*, 2012.
5. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, Vol. 1, p. 4, 2012.
6. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
7. C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
8. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online Passive–Aggressive Algorithms. *The Journal of Machine Learning Research*, Vol. 7, pp. 551–585, 2006.