Augmenting Flower Recognition by Automatically Expanding Training Data from Web

Cheng-Yu Huang, Yen-Liang Lin, Winston H. Hsu
National Taiwan University, Taipei, Taiwan
(lairoeaes,yenliang)@cmlab.csie.ntu.edu.tw, whsu@ntu.edu.tw

Abstract—Aiming to improve recognition rate, we propose a novel flower recognition system that automatically expands the training data from large-scale unlabeled image pools without human intervention. Existing flower recognition approaches often learn classifiers based on a small labeled dataset. However, it is difficult to build a generalizable model (e.g., for real-world environment) with only a handful of labeled training examples, and it is labor-intensive for manually annotating large-scale images. To resolve these difficulties, we propose a novel framework that automatically expands the training data to include visually diverse examples from large-scale web images with minimal supervision. Inspired by co-training methods, we investigate two conceptually independent modalities (i.e., shape and color) that provide complementary information to learn our discriminative classifiers. Experimental results show that the augmented training set can significantly improve the recognition accuracy (from 65.8% to 75.4%) with a very small initially labeled training set. We also conduct a set of sensitivity tests to analyze different learning strategies (i.e., co-training and self-training) and show that co-training is more efficient in our multi-view flower dataset.

Index Terms—semi-supervised learning, flower recognition, self-training, co-training

I. INTRODUCTION

Object recognition is one of the important tasks in computer vision. In this task, flower recognition is an interesting and challenging problem to recognize the flower species from its visual appearance. It can help the non-professional people to know the information about flowers (e.g. toxicity) and avoid danger. However, flower recognition has two main challenges: (1) the small inter-class variation (i.e., the flowers of different species have similar appearance both in shape or color) and (2) the large intra-class variation (i.e., the flowers of the same species vary in shape and color.)

We investigate a number of research works on flower recognition. Nilsson and Zisserman [1] propose the combinations of multi-modality features to resolve the problem of the inter-class and intra-class variation. Ito and Kubota [2] introduce three heterogeneous co-occurrence features and get a significant improvement in the Oxford 17-category flower dataset [1]. Those works pay much attention in combining different features or developing a discriminative feature for improving the performance. However, one problem for a realistic flower recognition system is that we require a large amount of training data to learn the generalizable models. Existing flower datasets only provide a small portion of example images for each species. In Figure 1, for example, all petals of daffodils in the Oxford 17-category dataset are yellow, but the petals of daffodils on the web could have different colors. Therefore, we need to supply these diverse data to ensure that the learned classifiers are robust in the real-world environment.

Nowadays, the Internet provides us plentiful resources of images, and we can use a keyword to retrieve thousands of images through web image search engines such as Google, Flickr, and so on. However, those images can not be used as training data directly. There are several reasons. First, in text-based image search engines, homonyms confuse search engines and the search results vary in the semantic meaning. This causes a portion of the returned images to possibly be unrelated to the intended category. Second, the intended objects may be within a scenic photo and the intended objects are too small to distinguish what they are. Third, the region of the intended objects in images is unknown and it is hard to learn a classifier model from a noisy environment, where the intended objects are surrounded by the cluttered backgrounds. Moreover, the mislabeled images also lead to a wrong classifier model. To create a good training dataset, we can manually remove those unsuitable images. However, manual annotation is time-consuming and requires domain knowledge especially in plant species annotation. In recent years, there is a substantial amount of work to apply semi-supervised learning techniques.

Fig. 1. The example images of daffodils in (a) the Oxford 17-category dataset and (b) search results from the web image search engines by the keywords, “Daffodil flower”. In Oxford 17-category dataset, the flowers have the same color of the petals; however, the petals of daffodils on the web could have different colors. The classifiers learned from a small labeled dataset may fail in these diverse data.
Fig. 2. The overview of the proposed system. First, we use the keywords (e.g., daffodils) to crawl web images from image search engines. Second, we extract the foreground region of the web images automatically to reduce the influence of background noise. Third, shape and color features are extracted on the segmented images. Fourth, we use the same process to extract features from the labeled images and train two modality classifiers. Fifth, these classifiers collaborate in the semi-supervised learning task, where learning from the unlabeled (or weakly labeled) images. Finally, we combine the improved modality classifiers into a mixed classifier by a late fusion scheme. The proposed method is scalable to extend to different modalities.

II. RELATED WORK

There is a few work that incorporates the unlabeled data into the training process of supervised classification frameworks [4], [6]. Self-training is historically the earliest idea about using unlabeled data in classification. This is a wrapper-algorithm and applicable to any supervised learning method. It starts by training a single classifier on the labeled data and predicting the labels of the unlabeled data by this classifier. Next, the confidently predicted examples are used as the additional training data and to retrain the original classifiers. This procedure continues until the unlabeled data is exhausted. To sum up, the classifier teaches itself based on its own predictions.

Co-training [6] is an extension of self-training where multiple classifiers trained on different modalities collaborate on semi-supervised learning tasks. The object we want to learn may have different modalities to describe it. Each modality is enough to distinguish a part of instances of objects but not all of them. If one modality can distinguish the part of instances, those modalities can learn from each other. Co-training starts by training two classifiers from different modalities. Each classifier predicts the labels of the unlabeled data. Then, it is retrained based on the confidently predicted examples by the other. That is, the classifier uses the prediction results of the other to teach itself. This procedure continues until the unlabeled data is exhausted.

In flower recognition, there are two important modalities, shape and color, to describe the flower categories.1 Inspired by co-training, we leverage the shape and color modalities as the two disjoint views of flower categories. The idea is that we distinguish the flower from the same species by one of the modalities. Even though the flowers are very

1Note that the proposed method can extend to multiple modalities. Here we take to common ones, color and shape, as the example.
different in one modality, we can distinguish whether they are in the same category through other modalities. Based on this observation, we use those modalities in different learning strategies to improve the flower recognition accuracy without human intervention.

III. SYSTEM FRAMEWORK

We illustrate our system framework in Figure 2. For each category in an existing dataset, we first use the related keywords to crawl the images from web image search engines. On the dataset and crawled images, we resize these images to the same width and detect their salient region [7]. Then, we use GrabCut [8] to extract the foreground region based on salient values. On the segmented images, we extract the shape and color features as two modalities of the object category. Next, we train two classifiers on different modalities. These classifiers learn from the web images in different training strategies. Finally, we obtain two improved classifiers and combine them into a mixed classifier by a late fusion scheme.

A. Automatically Salient Region Detection

The web images are varied in different sizes and the foreground objects are in different positions and scales in each image. We resize all of the images to 500 pixels wide to reduce the computational complexity. To obtain the foreground region automatically, we first use the implementation of [7] provided by the authors to get the salient value per pixel, and define four ranks of thresholds to segment the salient values into four regions: foreground region, possible foreground region, possible background region and background region. Then, these regions are used as the initial masks and optimized by using GrabCut. We show a segmentation example in Figure 3. Thresholds are chosen empirically, which get the maximum performance in the labeled dataset. All of categories use the same settings.

B. Feature Extraction and Representation

In flower recognition, shape and color information are the most important modalities to describe the visual appearance of different flower species. Each modality has its own properties and is sufficient to train a good classifier. In this paper, we extract SURF [9] features from gray-scale images and OpponentSURF [10] features from color images as shape and color features, respectively. OpponentSURF describes all channels in opponent color space using SURF descriptors. Opponent color space is a better model for perception of color. It can be converted from RGB color space with Equation 1.

\[
\begin{pmatrix}
O_1 \\
O_2 \\
O_3
\end{pmatrix} = \begin{pmatrix}
\frac{R-G}{\sqrt{3}} \\
\frac{R+G-2B}{\sqrt{6}} \\
\frac{R+G+B}{\sqrt{6}}
\end{pmatrix}.
\]

(1)

In the bag-of-words model, local descriptors are represented as the bag of visual words. We use the VLAD [11] representation which is an extension of bag-of-words model. The primary advantages of VLAD over bag-of-words are more discriminative by adding the difference of each descriptor from the assigned cluster. Thus, it can contain more information with few visual words and is a more compact representation. We cluster all local descriptors in the training images of the labeled dataset using k-means clustering to get a visual vocabulary and each image is represented by a VLAD vector.

C. Flower Classifiers

We aim to expand the training images from the web image search results. However, the unrelated images also exist in the search results. Thus, we train a two-class classifier (e.g. daffodil vs. background) to remove unrelated images. For each flower species, a weak classifier is trained by each modality. To predict the final label of a test image, we use the late fusion scheme to combine the modality classifiers.

We use L2-regularized logistic regression SVM classifier implemented by liblinear [12]. Given a set of instance-label pairs \((x_i, y_i), i = 1, \ldots, l, x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}\), the method solves the following optimization problem:

\[
\min_w \frac{1}{2}w^Tw + C \sum_{i=1}^{l} \log(1 + e^{-y_i w^T x_i}),
\]

(2)

where \(C > 0\) is a penalty parameter. To remove unrelated images in the web images, we train a two-class SVM classifier from one of the flower categories in the dataset and the general background images by a published background image dataset (e.g. daffodils vs. background). The SVM classifier predicts the label of a testing example and the probability:

\[
P(y = 1 \mid x) = \frac{1}{1 + e^{-w^T x}}.
\]

(3)
The final label and the probability can be calculated by the following late fusion scheme:

\[
y = \begin{cases} 
+1, & s \geq 0.5 \\
-1, & s < 0.5 
\end{cases}
\]

\[
s = \alpha \cdot p_{\text{shape}} + (1 - \alpha) \cdot p_{\text{color}},
\]

where \(y\) is the label of the example, \(s\) is the confident score, \(p_{\text{shape}}\) is the probability of the shape classifier, \(p_{\text{color}}\) is the probability of the color classifier and \(\alpha\) is the linear weight learned by maximizing the performance in the validation set of the labeled dataset.

D. Semi-supervised Learning

We have two different learning strategies to expand the training data from the web images. In Figure 4, we illustrate the cooperation of two modality classifiers in different strategies, and the algorithms are given in Algorithm 1 and Algorithm 2.

1) Self-training: The first strategy, self-training, is a straightforward method, which uses the final prediction to select the confidently predicted examples from the web images. Those selected examples with the predicted label are added into the training data as resources for retraining. This method takes different modalities into consideration to select the most confidently predicted examples.

2) Co-training: The second strategy is co-training. Unlike self-training, each modality classifier predicts its own labels on the web images. Those images confidently predicted by one classifier are added to the training data pool of the other classifier. Each modality classifier is retrained with the examples from the prediction of the other classifier. This method improves the recall of prediction in each step of the training data expanding process. That is, some instances of the category can be distinguished clearly in one of the modalities but not in the other, and those instances are the important training data for the other modality classifier. After co-training, we use the late fusion scheme to merge the improved classifiers into the final classifier.

Algorithm 1 Self-training

1: procedure
2: \(L \leftarrow\) labeled training examples
3: \(U \leftarrow\) unlabeled examples
4: loop \(k\) iterations
5: Train a classifier \(h\) by \(L\)
6: The classifier \(h\) picks \(p\) positives and \(n\) negatives from \(U\)
7: Add those examples to \(L\)
8: Remove those \(p + n\) examples from \(U\)

Algorithm 2 Co-training

1: procedure
2: \(\{L_{\text{shape}}, L_{\text{color}}\} \leftarrow\) labeled training examples
3: \(U \leftarrow\) unlabeled examples
4: loop \(k\) iterations
5: Train a shape classifier \(h_{\text{shape}}\) by \(L_{\text{shape}}\)
6: Train a color classifier \(h_{\text{color}}\) by \(L_{\text{color}}\)
7: \(h_{\text{shape}}\) picks \(p\) positives and \(n\) negatives from \(U\) and adds those examples to \(L_{\text{shape}}\)
8: \(h_{\text{color}}\) picks \(p\) positives and \(n\) negatives from \(U\) and adds those examples to \(L_{\text{color}}\)
9: Remove those \(2p + 2n\) examples from \(U\)

IV. EXPERIMENTS

A. Dataset

We use two published datasets and construct a new dataset, Flickr10k, in our experiments.

1) Oxford 17-category flower dataset

One of the state-of-the-art flower datasets [1]. It contains 17 flower categories and each category has 80 images. The dataset are randomly split into training, validation and test sets in [1]. We use the same data splits in our experiments.

2) The background images (Caltech101 dataset [13])

We only use the background category of this dataset as the general background images. Those background images obtained from the search results of the keyword “things” in Google.

3) Flickr10k

We crawl the images of all categories in Oxford flower dataset from Flickr. It contains 10k images totally. The scientific name for each flower category is a precise keyword which prevents the problem of synonyms of the common name. Thus, we use the scientific name as the keyword to crawl the web images. The number of images in each category varies from 136 (Tulipa sylvestris) to 1000 (Galanthus nivalis) and the average number is 600. We also annotate those images and there are 52% of correct images (by keyword) in each category (average).

B. Experimental Settings

We define four sets from the above datasets to evaluate our proposed framework in a realistic web environment. First, training set contains 40 images from each category of the Oxford flower dataset and 40 images from the background category of the Caltech101 dataset. We combine the images from a flower species with the general background category to train the classifiers of this species. Second, evaluation set contains 20 images of each category and 20 images of the background class for learning the fusion weight of the modality classifiers. Third, we choose 75% of images of each category in the Flickr10k dataset as the supply set which is the unlabeled (or weakly labeled) resources in semi-supervised learning. Finally, the remaining 25% of images of each category is the test set for evaluating the performance. To compare co-training with self-training, we use the same algorithm parameters. In each iteration, those algorithms select \(p\) positive and \(n\) negative predictions for expanding training data, where \(p\) and \(n\) are set to 10. Both of algorithms perform
Fig. 4. The difference between self-training and co-training. In self-training, only one classifier updates its model by itself. In co-training, on the other hand, multiple classifiers updates their model collaboratively. To compare the performance improvement in these strategies fairly, we train two modality classifiers in the training set. In self-training, these classifiers are combined by a linear weight to predict the label of test images; in co-training, each classifier predicts the label of test images and provides the prediction to the other classifier as the re-training resource.

$k$ iterations and $k$ is set to 5, where the images in the supply set is exhausted.

### C. Learning Curves

The learning curves are shown in Figure 5. Figure 5 (a) shows the performance of the classifiers trained on the non-segmented images. We observe that the initial classifier achieves a low performance of 65.8%, which indicates that the small labeled dataset is not enough to train a robust classifier in a realistic environment. In self-training, the best performance is 70.2% of accuracy. And the best performance achieved by co-training is 69.7% of accuracy, which is smaller than self-training. The main reason is that the features extracted from the non-segmented images are so noisy that the initial classifiers have a low accuracy, and self-training takes two modalities into consideration to select the confidently predicted examples, which is more robust than co-training. Self-training is more suitable in the noisy environment.

Figure 5 (b) shows that we can achieve a high initial accuracy by extracting the foreground region of images. The automatic salient region detection provides the initial seeds for segmentation without human supervision and the performance is significantly improved from 65.8% to 73.6%. Both self-training and co-training improve the performance from 73.6% to 75.4%. However, co-training improves the performance more efficiently than self-training because it makes use of the complementary information of two modalities. When we crawl more images from the image search engines, co-training is an effective and efficient strategy to expand training data from web.

### D. Discussion

The learning curve of each strategy drastically rises up in the beginning, and then drops gradually. The main performance improvement comes from the diversity in the supply set. The original training set is biased and lack of diverse examples. After adding the confidently predicted examples from the supply set, the updated classifiers are able to classify the diverse examples in the testing set. However, the process is automatic and without human intervention, it may include false examples as its training resources. Moreover, the distribution of the positive examples is unknown which makes us hard to decide when to stop the training process. When the positive examples are exhausted, the classifiers may add the negative examples with positive labels and are retrained with those false examples. It causes the performance degradation in the end of training process. To prevent this problem, we can stop the process when the percentage of the predicted negative examples in the unlabeled data is over a fixed threshold.

In our experiments, we observe that when the percentage of the predicted negative examples in the supply set increases to 80%, the training process can be stopped and the performance of the classifier is best. It is reasonable because the classifier detects that there are few positive examples which can be added to the next training step. We analyze the training process and conclude that this stop decision is suitable for the noisy supply set. There are two situations in our training process. In the first case, the classifier predicts a true positive example as a negative one and adds it to the training set in the previous iteration. It will increase the false positive rate, and the classifier will predict the remaining examples as negative with a high probability. The performance of the new classifier starts to decrease at this iteration. In the other case, the classifier predicts the example with a correct label in each iteration. If the number of the true positive examples is less than the true negative examples in the supply set and the classifier removes a true positive example and a true negative example from this set, the ratio of the true negative examples is increased in the remaining supply set. The performance of the classifier is improved and repeats this process until reaching the stop decision.

Actually, the ratio of the negative examples in the supply set is greater than the positive in each category. It usually happens when we crawled thousands of images from the web. The reasons may be due to the mistagged images, the ambiguous tags, the images where the flower is too small to identify and so on. In this paper, we focus on the improvement of semi-
Fig. 5. The classification accuracy versus the number of iterations. The plots show the performance of the classifiers trained on (a) the non-segmented images and (b) segmented images. The line with star markers represents the supervised classifier before expanding training data from web, the line with circle markers represents the improved classifier in self-training, and the line with plus sign markers represents the improved classifier in by the proposed co-training method.

supervised learning in the content-based classifier system. The order of the search result in the text-based search engine will be helpful to improve our training process. Moreover, we can limit the number of the search result to prevent a high ratio of negative examples. However, the text-based method has its own issues, such as mistagged images or semantic ambiguity, and adding this modality makes the system more complicated to analyze the improvement of semi-supervised learning. Also, to limit the number of search result will be influenced by the previous problem.

V. CONCLUSIONS AND FUTURE WORK

To leverage the freely available web images for improving recognition diversity, we propose a novel flower recognition framework to improve the recognition accuracy by automatically expanding training data from web. Inspired by co-training, we use two conceptually independent modalities as the disjoint views in the multiview learning task. Besides, we analyze different learning strategies to expand training data from unlabeled (or weakly labeled) data. Through our experiments, we conclude the advantage of these learning strategies. Self-training is a robust strategy to select the confidently predicted examples when the initial classifiers are at a low precision level. In contrast, co-training is an effective and efficient strategy when the initial classifiers are at a high precision level.

Our framework can also be extended to general object recognition tasks with a very small initially labeled training set. It removes noisy background automatically by salient region detection and GrabCut, and uses shape and color features, which are general to any object, to learn the classification model from a tremendous amount of web images. These learning strategies (i.e., self-training and co-training) can also be extended to use three or more features when the object contains various modalities.

In flower recognition, we can include GPS information embedded in the web images. Flowers are native in limited regions. For example, daffodils are native in Europe, North Africa, and West Asia. Thus, geolocation could provide another view to recognize the flower species. This new modality can be involved in these learning strategy and improve the recognition performance.

REFERENCES