Incremental Learning for Fine-Grained Image Recognition

Liangliang Cao
Yahoo Labs
New York, NY, USA
liangliang@yahoo-inc.com

Jenhao Hsiao
Yahoo Taiwan
Taipei, Taiwan
jenhaoh@yahoo-inc.com

Paloma de Juan
Yahoo Labs
New York, NY, USA
pdjuan@yahoo-inc.com

Yuncheng Li
University of Rochester
Rochester, NY, USA
yli@cs.rochester.edu

Bart Thomee
Yahoo Labs
San Francisco, CA, USA
bthomee@yahoo-inc.com

ABSTRACT

This paper considers the problem of fine-grained image recognition with a growing vocabulary. Since in many real world applications we often have to add a new object category or visual concept with just a few images to learn from, it is crucial to develop a method that is able to generalize the recognition model from existing classes to new classes. Deep convolutional neural networks are capable of constructing powerful image representations; however, these networks usually rely on a logistic loss function that cannot handle the incremental learning problem. In this paper, we present a new method that can efficiently learn a new class given only a limited number of training examples, which we evaluate on the problems of food and clothing recognition. To illustrate the performance of our proposed method on the task of recognizing different kinds of food, when using only 1.3% of training examples per category we achieved about 73% of the performance (as measured by F1-score) compared to when using all available training data.

Keywords

Incremental Learning; Dynamic Vocabulary; Deep Convolutional Neural Networks; Image Recognition

1. INTRODUCTION

Fine-grained image recognition is the task of recognizing specific object classes present in images, such as different breeds of dogs, species of birds, kinds of food, and types of clothing. In recent years there have been a number of studies on this topic [4, 9, 5], although all of them addressed the problem using a fixed vocabulary, i.e. the classes to detect were predefined. In realistic scenarios, however, it may be necessary to define new classes over time, as well as to redefine or remove existing classes. Consider for example the problem of clothing recognition (see Figure 1), where we may initially have defined a number of high-level categories (e.g. coat, shirt, pants, etc.), but later realize that we also need to recognize more specific subclasses of some of these categories (e.g. types of skirts: straight, mermaid, ruffed, etc.). Given an unseen example (e.g. a kilt), and with limited access to examples of the various clothing categories, we need to decide whether the new instance belongs to one of these existing categories, or to a new one. The same applies to the identification of specific colors or patterns. To address such scenarios, we propose a novel incremental learning strategy based on deep convolutional neural networks for handling the fine-grained recognition problem using a dynamic vocabulary, which only needs a limited number of training examples to accurately learn an object class.

Figure 1: The problem of incremental learning. Based on a limited number of examples from multiple object classes, we need to determine whether a new example belongs to any of the existing classes, or to a new class.

How to learn fine-grained classes efficiently is a challenging problem, considering that the more specific a class is the fewer training examples of the class are available. However, the fine-grained image recognition problem enjoys the three following characteristics: (i) it follows strictly multiclass learning, where we know that any class is mutually exclusive with its sibling classes, as well as with all those from different branches if a class hierarchy is used, which we thus can use as negative examples, (ii) sibling classes are similar to each other in some sense, so we can employ a shared representation to facilitate the classification task, and (iii) each new class corresponds to a simple concept that has limited diversity, which can be learned with relatively few examples.

In this paper, we present an incremental learning method for fine-grained image recognition that exploits the three aforementioned characteristics. Since the fine-grained objects may share similar patterns, we can explore the models and examples for existing classes that can reduce the amount of training required. A surprising fact is that our method can learn a new class from just ten examples, while obtaining comparable accuracy to a model that uses all available data.

ICMR'16, June 06-09, 2016, New York, NY, USA

DOI: http://dx.doi.org/10.1145/2911996.2912069

2. RELATED WORK

There have been a number of studies on image recognition using deep learning. Domain adaptation [1] focuses on predicting auxiliary problems based on the current features, while a never-ending learning frame [6, 7] was used to learn new visual concepts. More recently, context information from videos [8, 11] was also used for learning new concepts. A key difference between these works and our proposed method is that existing classes are mutually exclusive to any new classes that have to be modeled in fine-grained image recognition. Moreover, our method focuses on using as few examples as possible to learn a new (sub-)class.

The most closely related technique to our work is the k-nearest neighbor (k-NN) method that was extended to incremental learning [13, 16]. Since a k-NN method can easily separate a new category from the old categories, such a method can be applied to the fine-grained recognition problem. However, the two aforementioned methods are based on a histogram of SIFT words, which is not as efficient as the deep convolutional neural network representation we use in our work.

In recent years, deep convolutional neural networks have shown impressive results in large-scale image recognition [10, 12, 17, 18]. Borrowing a network with transferred features from a related task can greatly boost the performance. For example, off-the-shelf deep convolutional neural networks trained on ImageNet have been applied to other datasets, such as Caltech-256, and achieved good results [15]. Even better performance can be obtained by fine-tuning the network [9]. However, these studies did not consider the fine-grained recognition problem and can not be applied to the incremental learning scenario.

3. METHOD

Classic deep convolutional neural networks [10, 17, 18] use the cost function of logistic loss (also known as cross entropy), given by

$$C = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{1}(y_i = k) \cdot \log p_k(\mathbf{x}_i),$$

where $i = 1, \cdots, N$ denotes the index of training examples, $k = 1, \cdots, K$ the index of class labels, y_i the label of example i and $\mathbf{x}_i$ the feature vector of example i, which in a neural network is the output of the penultimate layer; the indicator function $\mathbb{1}(\cdot)$ equals 1 if the condition is true or 0 otherwise, and $p_k(\cdot)$ is the prediction score of the neural network that satisfies

$$\sum_{k=1}^{K} p_k(\mathbf{x}) = 1, \ \ \forall\, \mathbf{x}. \tag{1}$$

In practice, $p_k(\cdot)$ is usually obtained by the output of a softmax layer.

The challenge of incremental learning lies in the fact that Equation (1) does not hold when a new class is added. Namely, when a new class $K+1$ is introduced,

$$\sum_{k=1}^{K} p_k(\mathbf{x}) + p_{K+1}(\mathbf{x}) > 1.$$

To overcome this problem, we consider the softmax function

$$p_l(\mathbf{x}) = \frac{\exp(\mathbf{w}_l^T\mathbf{x})}{\sum_{k=1}^{K}\exp(\mathbf{w}_k^T\mathbf{x})},$$

where l and k are indexes of class labels, and $\mathbf{w}_l$ and $\mathbf{w}_k$ are the parameters corresponding to l and k, respectively[1]. We observe that this function can also be written as

$$\begin{aligned} p_l(\mathbf{x}) &= \frac{1}{\sum_{k=1}^{K}\exp((\mathbf{w}_k - \mathbf{w}_l)^T\mathbf{x})} \\ &= \frac{1}{1 + \sum_{k=1,k\neq l}^{K}\exp((\mathbf{w}_k - \mathbf{w}_l)^T\mathbf{x})}. \end{aligned}$$

Now it becomes clear that the prediction score is determined by the differences between $\mathbf{w}_k$ and $\mathbf{w}_l$. To emphasize this, we introduce a new function for incremental learning that we call the "softratio", given by

$$r_{k,l}(\mathbf{x}) = \exp((\mathbf{w}_k - \mathbf{w}_l)^T\mathbf{x}).$$

If the classifier successfully predicts that $\mathbf{x}$ belongs to the l-th class, then $\mathbf{w}_k^T\mathbf{x} - \mathbf{w}_l^T\mathbf{x} < 0$; this tends to be a large negative number, such that

$$r_{k,l}(\mathbf{x}) = \exp((\mathbf{w}_k - \mathbf{w}_l)^T\mathbf{x})) \approx 0. \tag{2}$$

Using the softratio function, we can now define the cost for an example from the new class ($l = K+1$) as

$$\begin{aligned} C_l^+(\mathbf{x}) &= -\log\frac{1}{1+\sum_{k=1}^{K} r_{k,l}(\mathbf{x})} \\ &= \log[1 + \sum_{k=1}^{K} r_{k,l}(\mathbf{x})], \end{aligned} \tag{3}$$

where the definition of this cost is no longer constrained by Equation (1). The cost for an example from one of the existing classes ($l \leq K$) is defined as

$$\begin{aligned} C_l^-(\mathbf{x}) &= \log[1 + \sum_{k=1,k\neq l}^{K+1} r_{k,l}(\mathbf{x})] \\ &= \log[1 + r_{K+1,l}(\mathbf{x}) + \sum_{k=1,k\neq l}^{K} r_{k,l}(\mathbf{x})] \\ &\approx \log[1 + r_{K+1,l}(\mathbf{x})], \end{aligned} \tag{4}$$

where $\approx$ holds following Equation (2). Our new cost function for incremental learning is then a combination of Equations (3) and (4), given by

$$C = \sum_{i=1}^{N} \mathbb{1}(y_i \leq K)C_{y_i}^-(\mathbf{x}_i) + \mathbb{1}(y_i = K+1)C_{y_i}^+(\mathbf{x}_i). \tag{5}$$

With this cost function, we can keep adding new classes into our model in an incremental manner. Given a network trained with K classes, we can add a new class $K+1$ by introducing new examples and then retrain the model using the cost function in Equation (5). We can simply repeat this process to add additional classes. In practice, for an incremental class $K+1$, we only learn the parameter vector $\mathbf{w}_{K+1}$, while keeping the existing parameters fixed.

To ensure learning is balanced, we usually require every class (including classes $1, \cdots, K$) to have the same amount of training examples. Since our objective is to learn with a limited number of training examples, we cannot afford to adjust too many parameters as this will lead to significant overfitting.

To further reduce the requirement of training examples from existing classes, we can take the $\bar{\mathbf{x}}_k$ as the mean statistic of $\mathbf{x}$ in the

[1]For simplicity, we let $\mathbf{x} \leftarrow [\mathbf{x}, 1]$ so that $\mathbf{w}^T\mathbf{x}$ denotes a general linear function.

k-th existing class, i.e.

$$\bar{\mathbf{x}}_k = \frac{\sum_{i=1}^{N} \mathbb{1}(y_i = k)\mathbf{x}_i}{\sum_{i=1}^{N} \mathbb{1}(y_i = k)},$$

such that the new cost function becomes

$$C = \sum_{k=1}^{K} C_k^-(\bar{\mathbf{x}}_k) + \sum_{i=1}^{N} \mathbb{1}(y_i = K+1) C_{y_i}^+(\mathbf{x}_i). \quad (6)$$

Although we will show that the performance of Equation (6) is worse than that of Equation (5), it is particularly suitable for the scenario where we cannot access the examples of previously added classes, but only mean statistics of their features.

4. EXPERIMENTS

We evaluated our method using two datasets. In both datasets, our method can be viewed as a two-step process. First, we train a deep convolutional neural network on a number of predefined fine-grained classes, where each class has as many training examples as are available in the dataset. Second, we apply our incremental learning approach to learn new classes, where each class may only have a limited amount of training examples.

Food-101 [4] is a large-scale fine-grained public dataset containing 101 classes of food. Each class contains 750 training images and 250 test images. In this paper, we used the first 90 classes as the set of existing classes and the last 11 classes as the new classes for incremental learning.

Clothing-33 is a proprietary dataset containing 33 different clothing attributes, of which 23 refer to different colors (e.g. white, apricot, cream) and 10 to different patterns (e.g. floral, dotted, striped). The practice of clothing recommendation prefers a dynamic vocabulary in which a new color or a pattern may be added from time to time, a scenario for which incremental learning is very well suited. To learn color models, we randomly selected 19 colors as the existing classes and the remaining 4 colors as the new classes, while to learn pattern models, we randomly selected 7 patterns as the existing classes and the remaining 3 patterns as the new classes.

We split the datasets into training and test sets, each including all classes.

We initialized our deep convolutional neural network using a model trained on ImageNet [5], which is based on a million images annotated with 1,000 object classes, and then tailored it to either the food or clothing dataset using Nesterov stochastic gradient decent (SGD) [14]. We implemented the SGD using Theano [3] on K80 GPU cards, with the batch size fixed to 128 and the learning rate decreasing from 0.002, where we trained the network for 100 epochs over existing classes. Once the network was tailored to the dataset, we then started the process of incremental learning. During incremental learning, we found that stochastic optimization is less preferred, since there are only a limited amount of training examples. Instead, we placed all of training examples in memory and employed limited-memory BFGS [2], which can quickly converge to the solution in both smooth and non-smooth functions.

We first present our experiments on food recognition. We tailored the deep network to the Food-101 dataset using the first 90 classes, and then learned an incremental model for all the other 11 classes using the cost function given by Equation (5). In Figure 2 we see that the incremental model is able to learn well from a limited number of examples, achieving 73.18% of the average F_1 score (the harmonic mean of precision and recall) which is comparable with the result of using all 750 examples. In contrast, when

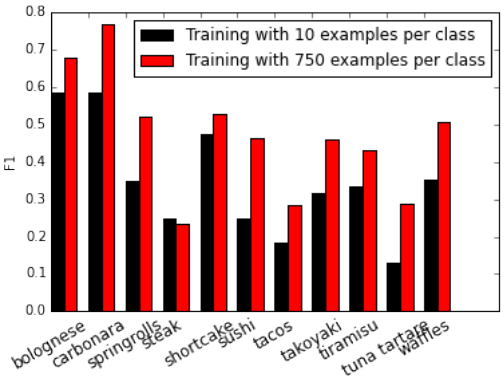

Figure 2: Measuring the F_1 score of incremental learning using the cost function in Equation (5) on the Food-101 dataset for different numbers of training examples per class.

we learned an incremental model using Equation (6) the average F_1 score dropped to 53.50% when we only use the mean statistics of the existing classes.

To illustrate why our incremental learning method is able to obtain a good performance with only a few training examples, in Figure 3 we show how the F_1 score, precision and recall change with additional examples for the class `bolognese` (class 91) of the Food-101 dataset. Although we intuitively would expect the performance to grow with more training data, the performance appears to plateau after only a small number of training examples. The reason is that our method is able to effectively utilize the representation from existing classes. As shown in Equation (5), we only need to learn $\mathbf{w}_{K+1}$, while we can keep reusing $\mathbf{w}_1, \cdots, \mathbf{w}_K$. By reducing the parameters to learn, we can reach a good performance with a limited number of training examples from every new class. For all classes, we found that the performance generally started to plateau after about 10 examples.

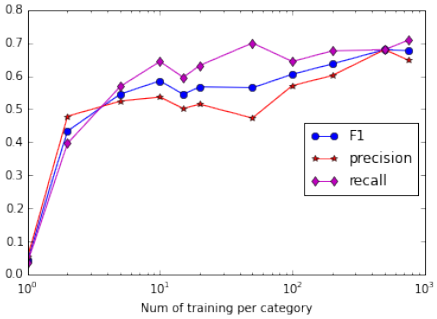

Figure 3: The performance of incremental learning using the cost function in Equation (5) for varying numbers of training examples for the `bolognese` class of the Food-101 dataset.

To further demonstrate the effectiveness of our method, Table 1 compares our incremental learning method with several baseline methods, namely the random forest discriminant components [4] (RFDC), the fine-tuned deep convolutional network [9] (FCNN), and the nearest neighbor-based incremental learning [16] (NNIL), where the latter uses the same deep features as FCNN. The methods in the "standard learning" group used all 750 examples per class, while the methods in the "incremental learning" group only used a few examples per class. For a fair comparison with the results reported in the RFDC and FCNN papers, we calculated the accuracy over the same data (25,250 test images, 101 classes). We can see

that that our method significantly outperforms RFDC, and is also much better than NNIL. The performance of our method is close to that of FCNN, which is impressive since our own incremental method used less training examples.

Table 1: Comparing our method against baselines on the Food-101 dataset.

Setting	Methods	Accuracy (%)
Standard learning	RFDC [4]	50.76
	FCNN [9]	64.37
Incremental learning	NNIL [16]	45.93
	Our method	60.32

We now apply our method to incrementally learn clothing colors and patterns. In the Clothing-33 dataset there are 6K images with pattern labels and 32K images with color labels. We use 80% of the images for training and the other 20% for testing. We apply our method using the cost function given by Equation (5) to incrementally learn four colors (apricot, white, leopard print, cream) and three patterns (floral, dotted, stripe) from the Clothing-33 dataset. Figure 4 compares the performance when using only a few examples with when using all examples. Our incremental learning method again achieves a performance that is competitive for a number of classes. While the striped pattern and the white color were more difficult to capture using just a few examples, the performance of learning the dotted pattern and the cream color was comparable to learning them with all available examples.

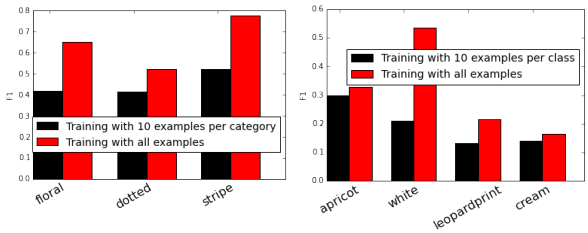

Figure 4: Measuring the F_1 score when incrementally learning clothing patterns (left) and colors (right).

Table 2 offers another view on how effective our method is using limited training examples. For the problem of incremental food recognition, when using only 1.3% of all available training examples per class, we can achieve about 73% of the performance (as measured by F_1 score) compared to when using all examples. The task of incrementally learning clothing patterns and colors appears to be more difficult than food recognition, but the relative performance obtained with incremental learning is nonetheless still about 65%.

Table 2: Comparing the sampling percentage and relative performance of using 10 examples per class versus using all examples per class.

Task	Avg. sampling %	Avg. F_1 %
Food	1.33%	73.18%
Clothes color	3.02%	64.48%
Clothes pattern	4.32%	65.00%

5. CONCLUSIONS

This paper considers the problem of incrementally learning a new category in fine-grained image recognition using only a few examples. Our solution benefits from the recent advances of deep convolutional networks, as well as from the fact that fine-grained image categories are correlated but mutually exclusive. We generalized the traditional softmax function to a new cost function, so it can be applied in an incremental learning scenario. The experimental results on food and clothing suggest that our method can learn a reliable model with only a few examples per class. In future work we aim to generalize our method to the YFCC100M dataset.

6. REFERENCES

[1] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. In *JMLR*, pages 1817–1853, 2005.

[2] G. Andrew and J. Gao. Scalable training of L1-regularized log-linear models. In *ICML*, 2007.

[3] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *SciPy*, 2010.

[4] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: delving deep into convolutional nets. In *BMVC*, 2014.

[6] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting Visual Knowledge from Web Data. In *ICCV*, 2013.

[7] S. Guadarrama, E. Rodner, K. Saenko, N. Zhang, R. Farrell, J. Donahue, and T. Darrell. Open-vocabulary object retrieval. *Robotics: Science and Systems*, 2014.

[8] M. Hasan and A. Roy-Chowdhury. Incremental activity modeling and recognition in streaming videos. In *CVPR*, 2014.

[9] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. In *BMVC*, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[11] A. Kuznetsova, S. Hwang, B. Rosenhahn, and L. Sigal. Expanding object detector's horizon: incremental learning framework for object detection in videos. In *CVPR*, 2015.

[12] M. Lin, Q. Chen, and S. Yan. Network in network. *ICLR*, 2013.

[13] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*, 35(11):2624–2637, 2013.

[14] Y. Nesterov. A method of solving a convex programming problem with convergence rate O(1/k2). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[15] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[16] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool. Incremental learning of NCM forests for large-scale image classification. In *CVPR*, 2014.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.