On the Effects of Spam Filtering and Incremental Learning for Web-Supervised Visual Concept Classification

Matthias Springstein[1]
matthias.springstein@tib.eu

[1]German National Library of Science and Technology (TIB)
Hannover, Germany

Ralph Ewerth[1, 2]
ralph.ewerth@tib.eu

[2]Leibniz Universität Hannover, Faculty of Electrical Engineering and Computer Science
Hannover, Germany

ABSTRACT

Deep neural networks have been successfully applied to the task of visual concept classification. However, they require a large number of training examples for learning. Although pre-trained deep neural networks are available for some domains, they usually have to be fine-tuned for an envisaged target domain. Recently, some approaches have been suggested that are aimed at incrementally (or even endlessly) learning visual concepts based on Web data. Since tags of Web images are often noisy, normally some filtering mechanisms are employed in order to remove "spam" images that are not appropriate for training. In this paper, we investigate several aspects of a web-supervised system that has to be adapted to another target domain: 1.) the effect of incremental learning, 2.) the effect of spam filtering, and 3.) the behavior of particular concept classes with respect to 1.) and 2.). The experimental results provide some insights under which conditions incremental learning and spam filtering are useful.

Keywords

Deep convolutional neural network; visual concept classification; Web-supervised learning

1. INTRODUCTION

Current approaches for semantic image search usually rely on a lexicon of pre-defined visual concepts which have to be detected automatically. For example, deep neural networks [10] are successfully applied to the task of visual concept classification. However, these methods require a large number of positive training examples for each concept category which is a time-consuming and costly task. Common image datasets such as Pascal Visual Object Classes (VOC) [6] or ImageNet [14] have been created by hundreds of people who judged the presence of visual concepts and additionally labeled object locations. Not so long ago, training data creation has been a pure manual task. To solve this problem, it has been proposed to use web and social media. For example, some approaches [1, 15] exploit image search engines to gather images in order to train a concept classifier. Other sources of information, such as Flickr, provide more realistic images with natural background, but this kind of source also includes more noisy tags [2, 18, 16, 9]. Image tags or surrounding text information on websites such as Flickr can be incomplete, subjective, ironic, or simply wrong. Recently, images available in the web are also employed to train a detector for localizing the target concepts in an image [2, 3, 4]. Another kind of approaches uses incremental learning to iteratively optimize the concept model with additional images [3, 11] or to find new concepts in the downloaded data. In general, a further popular alternative is to use pre-trained models, e.g., pre-trained deep neural networks such as GoogleNet or Alexnet [5], but normally these models require fine-tuning based on additional training data for an envisaged target domain.

In this paper, we investigate some properties of a web-supervised learning system for visual concept classification. The envisaged scenario is to build an automatic classification system, which is only initialized with the concept names to be learned, for the target domain Flickr. The baseline system can be an (initially) untrained network as well as a pre-trained network based on ImageNet data. After it has been initialized, the system starts gathering images from Flickr with appropriate tags, i.e., the concept names, and iteratively refines the network model via incremental learning. A spam filtering step based on textual and visual information can precede the model update in each training iteration. The objective of the paper is to provide an insight into 1.) the effect of incrementally adding training data to the network, 2.) the usefulness of spam filtering based on textual and visual information, and 3.) it is investigated if concept classes behave differently with respect to incremental learning and spam filtering. Experimental results are presented for Pascal VOC 2012 data. The performance of the web-supervised system is also compared with a supervised version that is fine-tuned using Pascal VOC 2012 training data.

The remainder of the paper is organized as follows. In section 2, the web-supervised learning system is presented. Experimental results are discussed in section 3 and section 4 concludes the paper and outlines areas for future work.

2. WEB-SUPERVISED LEARNING

In this section, we describe the framework for web-supervised learning of visual concepts. The system consists of

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

ACM ISBN 978-1-4503-4359-6/16/06.

DOI: http://dx.doi.org/10.1145/2911996.2912072

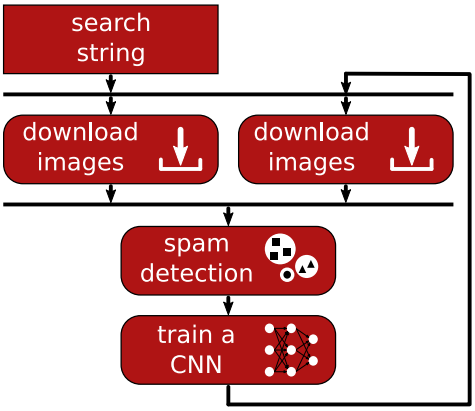

Figure 1: Iterative model of the system.

Figure 2: Example images from a non-spam (2 top rows) and a spam (2 bottom rows) cluster.

three components: crawling, spam detection, and concept learning using convolutional neural networks (CNN). The system is initialized only with the names of the target concepts. Then, our target domain Flickr is used to find positive training examples for all concepts. If an appropriate entry is found, the image and all related information are downloaded and stored. Search engines such as Google or Bing are not exploited since it has been shown that they tend to be biased towards "clear" iconic images [13, 12], i.e., images showing only one visual category. Textual and visual information are exploited to identify "good", i.e., positive training images and to filter out "spam" images from the downloaded content. An image is filtered out if one of the two filters (textual or visual) considers an image as spam. Spam detection based on textual information is described in previous work [7]. The crawling of web content and the spam detection steps are conducted separately for each specified target concept. The system learns in several iterations and is basically designed to learn endlessly. The system's workflow is illustrated in Figure 1.

2.1 Visual Spam Detection

This filter exploits the visual information of each image candidate in order to separate spam images from relevant images which are supposed to be useful for training. To describe an image, a feature vector is generated using a pre-trained CNN. The Caffe model BVLC GoogLeNet [17] is used which is trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 data [14]. The feature vector is computed using the last pooling layer of the GoogLeNet model. Then, the feature vectors are clustered using k-means algorithm. To estimate the best number of clusters, k is iterated from 2 to 20 clusters. The silhouette coefficient describing the homogeneity of clustering results is computed for each k. The clustering with the maximum silhouette coefficient is used in the next step.

Clusters consisting of a low number of images or having a silhouette coefficient below a threshold are discarded as spam. In the next step, the average distance of each cluster member to its cluster center is computed. These average distances are sorted in ascending order and the maximum of two successive pairs are computed. The maximum is used to split the sorted list into relevant and irrelevant clusters (images). All clusters below this threshold are used as training data while the other clusters are discarded as spam. Some examples of clustered images are shown in Figure 2.

2.2 Incremental Learning Process Using CNN

The clusters that remain after spam detection are used as positive training examples for the target concept. We consider two scenarios to train a CNN as classifier. In the first scenario we use CNN with randomly initialized weights, while a CNN pre-trained on ILSVRC 2012 training set is exploited in the second scenario.

The first deep neural network that we are using is the BVLC AlexNet [10] model from Caffe [8]. The system uses a stochastic gradient descent (SGD) learning algorithm with a momentum term of 0.9. In the first scenario, the network is initially trained with 450 000 iterations and a batch size of 256. We start with a learning rate α of 0.01 and after 100 000 iterations, the learning rate is dropped by a factor of 10. In the second scenario, only 100 000 iterations are used to train the network. Also an initial learning rate α of 0.001 is used, but it is increased in the last layer by a factor of 10. After 25 000 iterations, the learning rate is reduced by a factor of 10.

In addition, a pre-trained CNN GoogLeNet [17] model from Caffe [8] is used as a classifier and fine-tuned for the new task. This model has shown good performance on the ILSVRC 2014 [14]. A SGD learning algorithm is used for fine-tuning and the number of neurons in the last layer is set to the number of target concepts. Only 100 000 iterations with a batch size of 128 and a momentum term of 0.9 are used. The initial learning rate α is reduced by the factor 10 to 0.001, but the learning rate in the last layer is increased by the factor of 10. To increase the number of training examples, all images are scaled to a size of 256×256 and a region of size 224×224 pixels is cropped from the image. Additionally, the training examples are mirrored randomly.

For each incremental training step, 2500 images per target concept are used to optimize the CNN. As long as the system finds further (2500) training examples in the web, this process is repeated.

3. EXPERIMENTAL RESULTS

All experiments are based on an incremental learning setting. The deep neural networks are configured as described in the previous section. In each iteration, new training examples (2500 images per concept) gathered from the web are passed as additional input to the spam filter and the network. The validation set of Pascal VOC 2012 [6] is used to

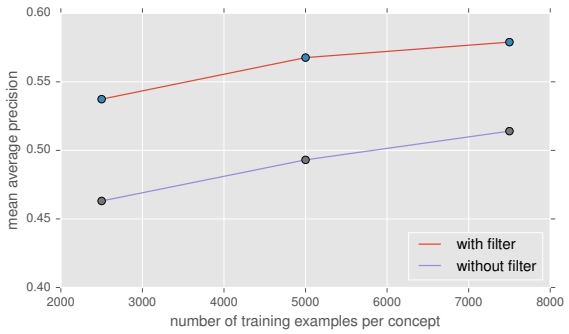

Figure 3: Effect of spam filtering for a CNN with randomly initialized weights.

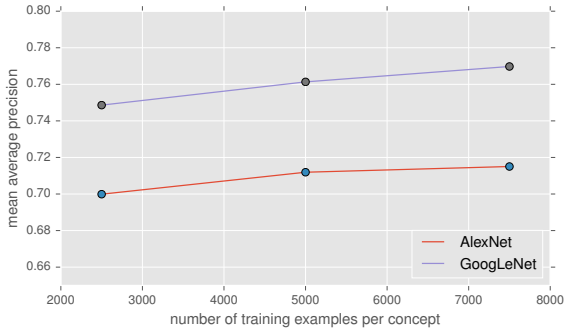

Figure 4: Incremental learning results for two CNNs pre-trained on ILSVRC 2012.

test and compare the performance of the models. This validation set is annotated with respect to 20 visual classes. The first experiment is conducted using 17 classes, since there were not enough training data available (at least 7500) at the time of this experiment for the concepts potted plant, dining table, and TV monitor. This effect was resolved by searching for synonyms as well, which has been employed in the subsequent experiments, i.e., results for the more comprehensive second experiment are reported for all 20 classes.

3.1 Validation of Spam Detection

In the first experiment, the impact of spam detection based on textual and visual information is evaluated. For this purpose, an untrained AlexNet with randomly initialized weights is used. Two AlexNet models are trained: The first model uses the raw image data captured from the Web, whereas the second model employs the spam filters.

Results in Figure 3 show that the mean average precision (MAP) is noticeably better for spam detection for all training iterations. This finding is confirmed in another experiment (described below) using GoogLeNet and 20 classes.

Based on this result, we have tested pre-trained versions of AlexNet and GoogLeNet in conjunction with the spam filtering process. The results displayed in Figure 4 show that the GoogLeNet outperforms the AlexNet network. Hence, a pre-trained GoogLeNet is used in the next experiment.

3.2 Incremental Learning Capability

In this experiment, the behavior of a pre-trained GoogLeNet is investigated for a larger number of eight iterations

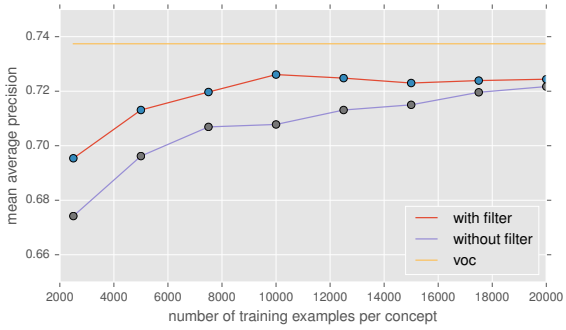

Figure 5: Results of a supervised (VOC training data) and two web-supervised versions, each version building on a CNN pre-trained on ILSVRC 2012.

for two conditions: with and without spam filtering. Again, in each iteration 2500 images per concept class are passed to the CNN and the (optional) spam filtering. That is, in the version without spam filtering 400 000 have been passed to the system. In addition a purely supervised CNN, i.e. without using any image data acquired from the web, is fine-tuned with the Pascal VOC 2012 training data in the same manner as described in section 2.2.

Figure 5 shows the trend of MAP for the eight iterations starting from 2500 up to 20 000 samples per target. The performance of the model using the filter system increases from 69.5 % to 72.4 %.

Also, it can be observed that spam filtering noticeably improves the MAP in the first iterations and MAP of both systems start to converge for 12 500 examples per concepts. Hence, it can be concluded that spam filtering is most helpful when using a smaller number of training examples. Although the web-supervised system version are slightly outperformed by the CNN using "clean" training data (73.7 %), they achieve (nearly) a similar performance – without any supervision and manual efforts for annotating training data.

3.3 Effects on Particular Concepts

The effects of incremental learning in conjunction with (and without) spam filtering can be observed in Table 1 for 10 out of 20 classes. The absolute improvement ranges from -0.7 % for dining table up to 7.6 % TV monitor when using spam filtering. Interestingly, for the system version without spam filtering this behavior is reversed. The absolute improvement ranges from -0.4 % for TV monitor up to 15.7 % for dining table. These concepts tend to be very sensitive to the spam filtering or incremental learning, respectively. Another interesting observation is that additional training examples beyond the size of 10 000 do not improve the system accuracy and the performance converges. From this point on, the learning process has to be modified, for example, by means of network parameters or training data selection.

4. CONCLUSIONS

In this paper, we have investigated the effect of spam filtering and incremental learning for a web-supervised deep learning framework for visual concept classification. A visual spam filter has been suggested which is used in conjunction with a textual filter in order to remove noisy images.

Several conclusions can be drawn. For a target domain

Table 1: Image classification results on Pascal VOC 2012 validation set.

Trained CNN with ...	MAP	table	dog	horse	mbike	person	pplant	sheep	sofa	train	monitor
VOC training data	73.7	61.5	84.7	77.7	83.2	76.4	40.0	83.7	56.2	91.4	71.5
Web & filter (2500)	69.5	48.3	85.6	81.7	84.1	69.3	37.6	78.3	45.6	89.5	32.3
Web & filter (5000)	71.3	51.5	85.0	87.7	84.1	70.3	42.5	80.1	47.4	91.4	38.0
Web & filter (10000)	72.6	51.2	86.1	90.9	83.6	72.9	40.2	80.6	47.5	92.0	49.0
Web & filter (20000)	72.4	47.6	86.8	91.2	84.4	73.1	39.4	80.1	50.7	92.1	39.9
Web & no filter (2500)	67.4	41.6	84.7	85.3	81.5	63.0	39.8	83.1	41.5	85.7	32.2
Web & no filter (5000)	69.6	45.9	84.5	88.3	82.0	62.1	42.1	85.5	48.8	87.6	33.6
Web & no filter (10000)	70.8	49.4	83.9	90.0	83.8	65.3	41.6	86.3	45.1	90.2	35.2
Web & no filter (20000)	72.2	57.3	84.6	89.5	83.1	66.7	45.5	88.7	48.3	90.7	31.8

such as Flickr, a purely web-supervised system based on a pre-trained CNN can achieve nearly the same performance as a system that is fine-tuned with manually annotated data. It has been further shown that the spam filter is useful for system with less training data, up to sample size of 10 000 per concept, or fewer training rounds. From this point on, the accuracy converges. Another insight is that a system without a spam filter can be on a par with the version using spam filter, when it used about twice as many training examples as the latter one. Finally, it has been demonstrated that there are some concept classes which are very sensitive to incremental learning or spam filtering, respectively.

In the future, we plan to investigate which system aspects such as self-validation, training data selection, or network layer modifications can prevent the system's performance from converging. For example, a self-evaluation component is needed that retains parts of a model structure (for particular concepts) which are superior to its own next generation.

5. REFERENCES

[1] K. Chatfield, R. Arandjelović, O. Parkhi, and A. Zisserman. On-the-fly learning for visual search of large-scale image and video datasets. *International Journal of Multimedia Information Retrieval*, 4(2):75–93, 2015.

[2] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. *arXiv preprint arXiv:1505.01554*, 2015.

[3] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1409–1416. IEEE, 2013.

[4] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3270–3277. IEEE, 2014.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[7] R. Ewerth, K. Ballafkir, M. Mühling, D. Seiler, and B. Freisleben. Long-term incremental web-supervised learning of visual concepts via random savannas. *IEEE Transactions on Multimedia*, 14(4):1008–1020, 2012.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[9] P. Kakar and A. Y.-S. Chia. If you can't beat them, join them: Learning with noisy data. In *ACM International Conference on Multimedia*, pages 571–580. ACM, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] L.-J. Li and L. Fei-Fei. Optimol: automatic online picture collection via incremental model learning. *International Journal of Computer Vision*, 88(2):147–168, 2010.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[13] E. Mezuman and Y. Weiss. Learning about canonical views from internet image collections. In *Advances in Neural Information Processing Systems*, pages 719–727, 2012.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[15] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):754–766, 2011.

[16] C. Sun, C. Gan, and R. Nevatia. Automatic concept discovery from parallel text and visual corpora. In *IEEE International Conference on Computer Vision*, pages 2596–2604, 2015.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[18] B. Zhou, V. Jagadeesh, and R. Piramuthu. Conceptlearner: Discovering visual concepts from weakly labeled image collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2015.