Object recognition using deep convolutional neural networks with complete transfer and partial frozen layers

Maarten C. Kruithof, Henri Bouma [1], Noëlle M. Fischer, Klamer Schutte

TNO, Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands

ABSTRACT

Object recognition is important to understand the content of video and allow flexible querying in a large number of cameras, especially for security applications. Recent benchmarks show that deep convolutional neural networks are excellent approaches for object recognition. This paper describes an approach of domain transfer, where features learned from a large annotated dataset are transferred to a target domain where less annotated examples are available as is typical for the security and defense domain. Many of these networks trained on natural images appear to learn features similar to Gabor filters and color blobs in the first layer. These first-layer features appear to be generic for many datasets and tasks while the last layer is specific. In this paper, we study the effect of copying all layers and fine-tuning a variable number. We performed an experiment with a Caffe-based network on 1000 ImageNet classes that are randomly divided in two equal subgroups for the transfer from one to the other. We copy all layers and vary the number of layers that is fine-tuned and the size of the target dataset. We performed additional experiments with the Keras platform on CIFAR-10 dataset to validate general applicability. We show with both platforms and both datasets that the accuracy on the target dataset improves when more target data is used. When the target dataset is large, it is beneficial to freeze only a few layers. For a large target dataset, the network without transfer learning performs better than the transfer network, especially if many layers are frozen. When the target dataset is small, it is beneficial to transfer (and freeze) many layers. For a small target dataset, the transfer network boosts generalization and it performs much better than the network without transfer learning. Learning time can be reduced by freezing many layers in a network.

Keywords: Defense, security, object recognition, deep learning, transfer learning, big-data analytics.

1. INTRODUCTION

Object recognition is important to understand the content of video and allow flexible querying in a large number of cameras, especially in the security domain where the number of CCTV cameras is growing exponentially. Typically, the object detectors that perform well on public benchmarks are trained on large collections (e.g., ImageNet [11]) or annotated subsets (e.g., the Pascal Visual Object Challenge (VOC) [13] and the ImageNet Large-Scale Recognition Challenge (ILSVRC) [16]). The deep convolutional neural network (CNN) has been demonstrated to be an effective approach [10] of which several implementations have been proposed, such as Decaf/Caffe [12][15][16], Overfeat [19] and R-CNN [14]. Many CNNs trained on natural images appear to learn features similar to Gabor filters and color blobs in the first layer. These first-layer features appear to be general for many datasets and tasks, while the latter layers appears to more-and-more specific for a particular class [1][17]. It was shown that it is possible to flexibly learn novel objects with only a low number of training samples by freezing many initial layers and retraining a classifier on the outputs of the initial layers [8][14]. This approach has been reused in an interactive demonstrator [6][7][18]. In transfer learning [4], a base network is pre-trained on a base dataset, and these learned features are transferred to be further trained on a target dataset. The learning rate for fine-tuning is typically set to be less than the initial learning rate to ensure that features learnt on the larger set are not forgotten and the step used to shrink the learning rate schedule is decreased to avoid overfitting [1][2]. Transfer of the features works well if the features are general, and not specific for the base task. Yosinski e.a. [20] already performed a valuable study about the transferability from a base dataset to a target dataset of features from each layer of the network. They showed that: initializing a network with transferred features can produce a boost to generalization performance, fine-tuning on the transferred target dataset improves performance, and random initialization on higher layers may drop performance due to fragile co-adaptation. However, they did not make a distinction between the number of layers that are copied and the number of layers that are fine-tuned.

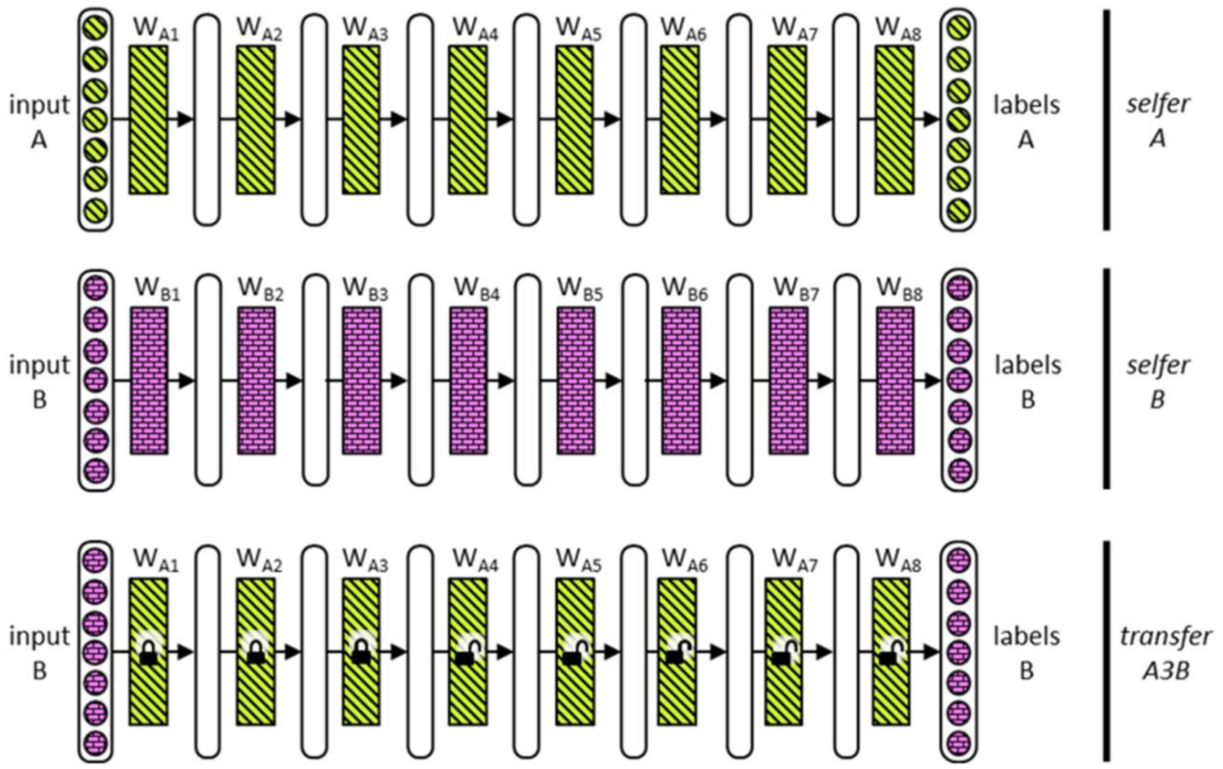
[1] henri.bouma@tno.nl; phone +31 888 66 4054; http://www.tno.nl

In this paper, our main contribution is that we transfer all layers and study the effect of retraining a variable number of layers and a variable target dataset size. We performed an experiment with a Caffe-based network [15] on 1000 ImageNet classes [11][16] that are randomly divided in two equal subgroups for the transfer from one to the other. We copy all layers and vary the number of layers that is fine-tuned and we vary the size of the target dataset. Furthermore, we performed other experiments with the Keras platform (which is a deep learning library for Theano) on CIFAR-10 dataset [16]. We show with both platforms and both datasets that the accuracy on the target dataset improves when more target data is used. When the target dataset is large, it is beneficial to freeze only a few layers. A network that was trained and applied to the same dataset is called a 'selfer' network and a network that was trained on one dataset and then applied to another dataset is called a 'transfer' network. For a large target dataset, the selfer network performs better than the transfer network, especially if many layers are frozen. When the target dataset is small, it is beneficial to transfer (and freeze) many layers. For a small target dataset, the transfer network boosts generalization and it performs much better than the selfer network. Computation time can be reduced by freezing many layers in a network.

The outline of this paper is as follows. Section 2 presents the method. Section 3 describes the experimental setup and it the results. Finally, Section 4 summarizes the conclusions.

2. METHOD

In this study we define a base dataset *A* with input values and labels and a target dataset *B* with its own input values and labels. Yosinski [20] proposed a transfer network where only a few layers were copied and the other layers were randomly initialized. However, this resulted in a poor connection between the copied layers and the random layers, which was called 'fragile co-adaptation'. We propose to implement a transfer network by copying all layers from the base network to avoid this effect and then freezing the first *N* layers (Figure 1). We will use a notation of transfer network *XnY*, where '*A3B*' refers to a network that was pre-trained on *A*, then transferred and fine-tuned on *B* with the lower 3 layers frozen.



W_{A1} W_{A2} W_{A3} W_{A4} W_{A5} W_{A6} W_{A7} W_{A8}
input A
labels A
selfer A
W_{B1} W_{B2} W_{B3} W_{B4} W_{B5} W_{B6} W_{B7} W_{B8}
input B
labels B
selfer B
W_{A1} W_{A2} W_{A3} W_{A4} W_{A5} W_{A6} W_{A7} W_{A8}
input B
labels B
transfer A3B

Figure 1: Top layer shows the selfer network that was trained on the base dataset *A*. The middle layer shows the selfer network that was trained on target dataset *B*. The bottom layer shows the transfer network *A3B*, which was originally pre-trained on *A*, then the first 3 layers were frozen and the other layers were free to be fine-tuned for *B*.

3. EXPERIMENTS AND RESULTS

3.1 Experimental setup

We performed two sets of experiments using other implementations and datasets, to improve generality and avoid conclusions that are only valid for one specific implementation and dataset.

The first experiment was performed on random splits on 1000 classes of ImageNet data [16] – which contains more than one million images – with the CNN implementation of Caffe [15]. The network consists of eight layers, of which five convolutional layers and three inner-product layers. The layers 1, 2 and 5 perform pooling, 6 and 7 have a dropout, 1 to 7 have rectified linear unit (ReLU) activation and the last layer has softmax activation. This experiment is similar to the transfer learning setup used by Yosinski [20], with the distinction that we copy the information of all layers rather than to randomly re-initialize retrained layers to avoid distortion in fragile co-adaptation during training.

The second set of experiments was performed on random splits on 10 classes of CIFAR-10. The training set consists of 50k images and the test set consists of 10k images. The CNN was implemented with Keras. Keras is a neural network library that can run in Python on top of Theano. The network consists of five layers, of which four convolutional layers with ReLU activations and one fully-connected dense layer with softmax activation. The first two layers are each followed by max pooling. The second and fourth layer are followed by a dropout.

3.2 Transfer learning with Caffe on ImageNet-1000

First, we analyzed transfer-learning with a Caffe-based neural network on ImageNet-1000 data, similar to the work of Yosinski [20]. The 1000 class dataset is randomly split in 500 classes for set A and 500 classes for set B. For the computed accuracies, we used 210,000 iterations. The results are shown in Figure 1. Two points of the 10% target data curve are missing because the process failed to complete. Our key observations are the following:

- Of course, the accuracy on the target dataset improves when more target data is used.
- When the target dataset is large, it is beneficial to freeze only a few layers. For example on ImageNet-1000, when the target dataset is larger than 50% of the base dataset, is better to freeze only the first (0, 1 or 2) layers.
- When the target dataset is small, it is beneficial to transfer and freeze many layers. For example, if only 1% of the target data is available (which is approximately 10 images per class), the performance improves with almost a factor three when 7 layers are frozen instead of only 1.

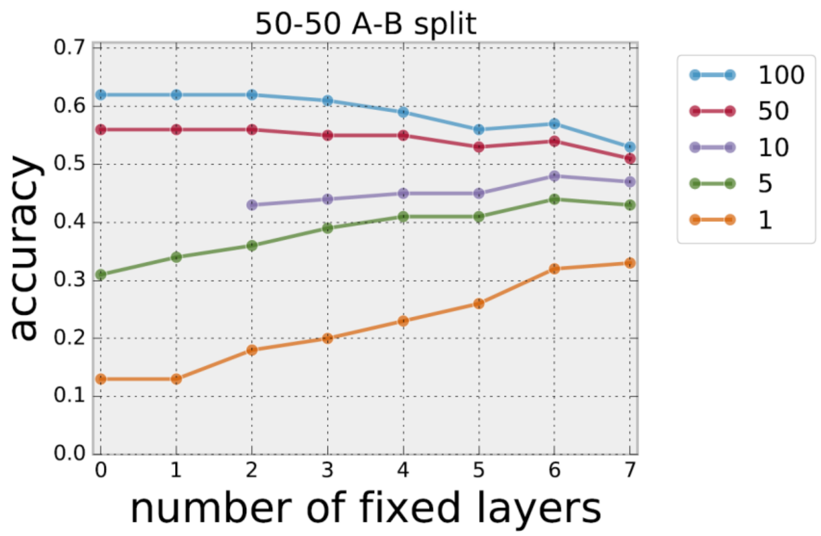


Figure 2: Accuracy of Caffe on ImageNet-1000 with varying number of fixed layers and varying target dataset size (using 1, 5, 10, 50 and 100% of the target data).

3.3 Transfer learning with Keras on CIFAR-10

Second, we analyzed transfer-learning with a Keras-based neural network on CIFAR-10 data. Besides the experiment that we performed on ImageNet-1000, we also looked at a different distribution of classes between the base and target dataset (30/70, 50/50, 70/30) and we looked at the computation time. For the computed final accuracies, we used 200 epochs and for the estimation of the required computation time was based on an early-termination criterion at 0.98 times the final accuracy. The accuracy results on the balanced dataset are shown in Figure 3, the accuracy results on the unbalanced dataset are shown in Figure 4, and the computation times are shown in Figure 5. The accuracies are much higher on the CIFAR-10 dataset than on ImageNet-1000, because the problem of discriminating 5 classes is much simpler than discriminating 500 classes. Our key observations on CIFAR-10 are similar to what we observed with Caffe on ImageNet-1000:

- The accuracy on the target dataset improves when more target data is used (Figure 3 and Figure 4).
- When the target dataset is large, it is disadvantageous to transfer and freeze many layers. For example on CIFAR-10, when the target dataset is larger than 50% of the base dataset, is better to freeze only the first (1) layer (Figure 3 and Figure 4). The transfer network performs worse than the selfer network, especially if many layers are frozen.
- When the target dataset is small, it is beneficial to transfer and freeze many layers (Figure 3 and Figure 4). The transfer network performs much better than the selfer network. Typically, this has always been the main reason for using transfer learning: to enable training without overfitting on small target datasets.
- With few classes in the target dataset the performance with even a very small amount of training data can be acceptable (Figure 4, 70-30 split.)
- Freezing many layers results in a network that can be trained faster than a network with many free layers that can be fine-tuned (Figure 5). A similar result was obtained on the ImageNet-1000 (results not shown).

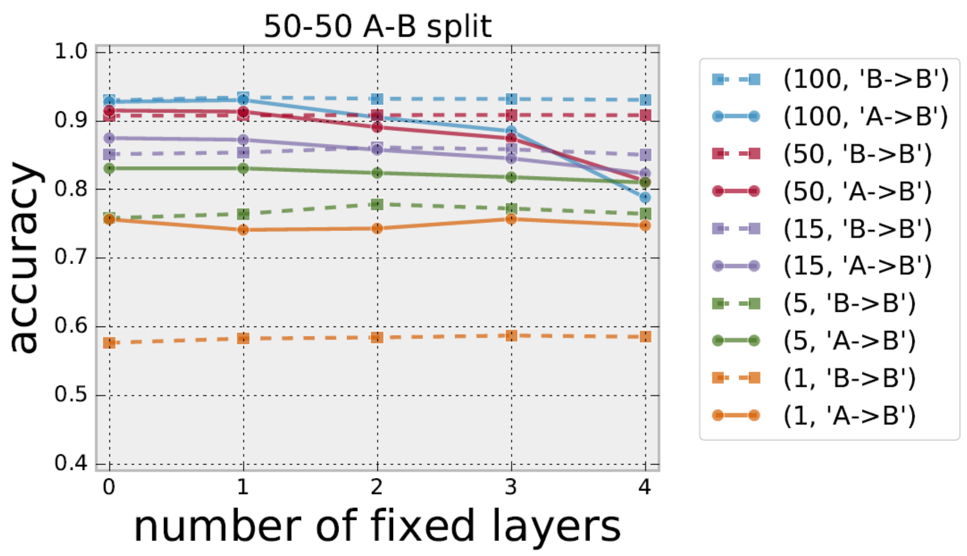


Figure 3: Accuracy of Keras on Cifar-10 with varying number of fixed layers and varying target dataset size. Equal amount of classes in base (A) and target (B) dataset.

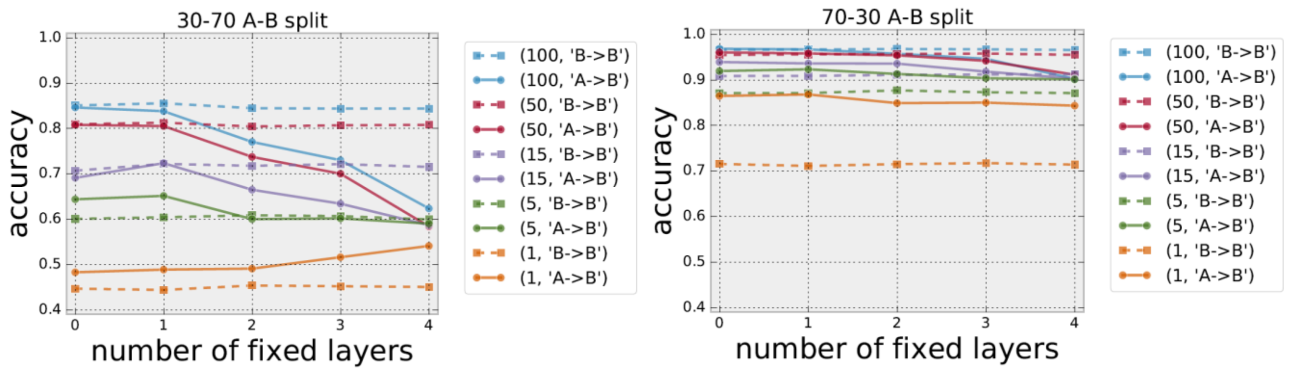


Figure 4: Accuracy of Keras on Cifar-10 with varying number of fixed layers and varying target dataset size. Unequal amount of classes in base (A) and target (B) datasets.

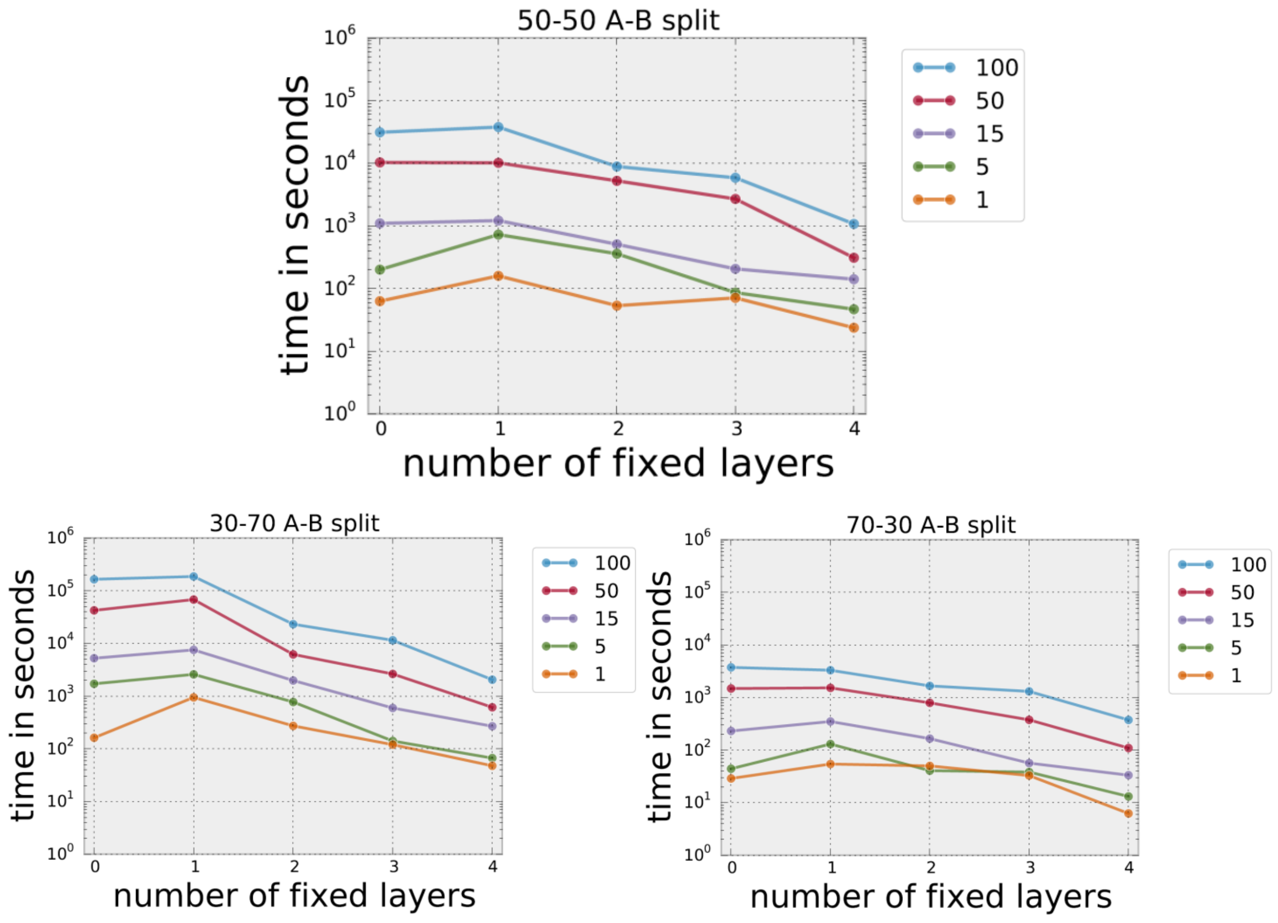


Figure 5: Computations time of Keras on Cifar-10 with varying number of fixed layers and varying target dataset size. Symmetric (50/50) and asymmetric (30/70 and 70/30) class distribution between base and target datasets.

3.4 Discussion

The work of Yosinski [20] stated that if transfer network *A3B* performs as well as selfer network *B*, there is evidence that the third- (or lower-)layer features are general, at least with respect to *B*. If performance suffers, there is evidence that the third-layer features are specific to *A*. In contrast to this earlier work, we cannot easily make the same statement. Figure 3 shows that the transfer network *A3B* improves the performance when *B* contains 1% of the samples and it deteriorates the performance when *B* contains 100% of the samples. Hence, it is not just an effect that a layer is generic or specific for a task, but also the effect that sufficient samples are needed to allow fine tuning to increase over fixed layer transfer. To the best of our knowledge, this is the first time that these two separate effects have been decoupled and analyzed layer by layer in a CNN.

The work of Yosinski [20] concluded that transferring features and then fine-tuning them results in networks that perform slightly (1.6%) better than those trained directly on the target dataset. We performed extended experiments, which allow us to conclude that the performance improvement is substantial for small target datasets and marginal (or absent) for large target datasets .

Applying Caffe to 1% target data of ImageNet without frozen layers resulted in only 13% accuracy. Freezing all layers resulted in a large accuracy increase of almost a factor 3 (to 33%). However, 33% accuracy is commonly not sufficient for practical applications. This is related to the complexity of the recognition of 500 classes. Recognition of less classes in CIFAR-10 shows that 85% accuracy can be obtained with only 1% of the data (Figure 4). Hence, practical applications may become feasible by increasing the relevancy of classes in the base dataset, the size of the target dataset or by focusing on a reduced number of classes.

The ImageNet data consists of a large number of classes. The 500 classes in the base dataset allow the creation of generic and reusable layers that are also representative for the classes in the target dataset. Therefore, it was expected that the transfer network performs much better than the selfer network when the target dataset is small (e.g., 1%). The CIFAR-10 dataset consists of only 10 classes of which 70%, 50% or 30% are used in the target dataset, in demonstrating the effect when reducing the number of classes. Here the results clearly show that a small amount of target classes allows the network to perform at an acceptable level (performance over 85%) when using a little as 1% of the training sample.

Security and defense applications often involve object recognition challenges where we have only a small amount of classes but only few training examples for each mission. The proposed methodology would allow to pre-train a CNN on a pre-arranged corpus, and subsequently fine-tune the network on the few training examples, and still reach an acceptable, mission relevant target discrimination level!

4. CONCLUSIONS

In this paper, we show that, especially for small target datasets, it is beneficial to copy all layers in the neural network to boost performance, and that the number of fine-tuned layers can be reduced to only the top layers to minimize training time. This was tested with multiple CNN implementations (based on Caffe and Keras) and on multiple datasets (ImageNet-1000 and Cifar-10). We show with both methods and both datasets that the accuracy on the target dataset improves when more target data is used. It is not always beneficial to use a transfer network, but it is a trade-off that depends on the size of the target dataset. When the target dataset is large, it is beneficial to freeze only a few layers. For a large target dataset, the network without transfer learning even outperforms the transfer network, especially if many layers are frozen. When the target dataset is small, it is beneficial to transfer (and freeze) many layers. For a small target dataset – which is common in practical applications – the transfer network boosts performance and it performs much better than the network without transfer learning. Learning time can be reduced by freezing many layers in a network.

ACKNOWLEDGEMENT

The work for this paper was conducted in the Early Research Program “Making Sense of Big Data” (MSoBD).

REFERENCES

[1] Azizpour, H., Razavian, A., Sullivan, J., Maki, A., Carlsson, S. “From generic to specific deep representations for visual recognition,” IEEE CVPR, (2015).

[2] Azizpour, H., Sharif-Razavian, A., Sullivan, J., Maki, A., Carlsson, S., “Factors of transferability for a generic ConvNet representation,” IEEE Trans. PAMI, (2015).

[3] Bakri, A., Elhoseiny, M., El-Gaaly, T., Elgamal, A., “Digging deep into the layers of CNNs: In search of how CNNs achieve view invariance,” ICLR, (2016).

[4] Bengio, Y., “Deep learning of representations for unsupervised and transfer learning,” JMLR Proc. Unsupervised and Transfer Learning 27, 17-36 (2012).

[5] Bengio, Y., Bastien, F., Bergeron, A., et al., “Deep learners benefit more from out-of-distribution examples,” JMLR Proc. AISTATS, 164 – 172 (2011).

[6] Boer, M. de, Schutte, K. and Kraaij, W., “Knowledge based query expansion in complex multimedia event detection,” Multimedia Tools and Applications, 1-19 (2015).

[7] Boer, M. de, Brandt, P., Sappelli, M., Daniele, L.M., Schutte, K., Kraaij, W., “Query Interpretation–an Application of Semiotics in Image Retrieval,” Int. J. Adv. Software 3/4(8), 435-449 (2015).

[8] Bouma, H., Eendebak, P., Schutte, K., Azzopardi, G., Burghouts, G., “Incremental concept learning with few training examples and hierarchical classification,” Proc. SPIE 9652, (2015).

[9] Bouma, H., Azzopardi, G., Spitters, M., et al., “TNO at TRECVID 2013: multimedia event detection and instance search,” Proc. TRECVID, (2013).

[10] Chatfield, K., Simonyan, K., Vedaldi, A. et al., “Return of the devil in the details: delving deep into convolutional nets,” BMVC, (2014).

[11] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L., “Imagenet: a large-scale hierarchical image database,” IEEE CVPR, 248–255 (2009).

[12] Donahue, J., Jia, Y., Vinyals, O., et al., “Decaf: A deep convolutional activation feature for generic visual recognition,” ICML, (2014).

[13] Everingham, M., van Gool, L., Williams, C., Winn, J., and Zisserman, A., “The PASCAL visual object classes (VOC) challenge,” IJCV 88, 303–338 (2010).

[14] Girshick, R., Donahue, J., Darrell, T., and Malik, J., “Rich feature hierarchies for accurate object detection and semantic segmentation,” IEEE CVPR, 580–587 (2014).

[15] Jia, Y., Shelhamer, E., Donahue, J., et al., “Caffe: convolutional architecture for fast feature embedding,” Proc. ACM Multimedia, 675–678 (2014).

[16] Krizhevsky, A., Sutskever, I., and Hinton, G., “ImageNet classification with deep convolutional neural networks,” NIPS, (2012).

[17] Long, M., Cao, Y., Wang, J., Jordan, M., “Learning transferable features with deep adaptation networks,” ICML, (2015).

[18] Schutte, K., Bouma, H., Schavemaker, J., et al., “Interactive detection of incrementally learned concepts in images with ranking and semantic query interpretation,” IEEE Content-Based Multimedia Indexing CBMI, (2015).

[19] Sermanet, P., Eigen, D., Zhang, X., et al., “Overfeat: integrated recognition, localization and detection using convolutional networks,” ICLR, (2014).

[20] Yosinski, J., Clune, J., Bengio, Y., Lipson, H., “How transferable are features in deep neural networks?,” NIPS, 3320 – 3328 (2014).

[21] Zeiler, M., Fergus, R., “Visualizing and understanding convolutional networks,” ECCV LNCS 8689, 818-833 (2014).