From Generic to Specific Deep Representations for Visual Recognition

Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, Stefan Carlsson
Computer Vision and Active Perception (CVAP)
KTH (Royal Institute of Technology)
Stockholm, SE-10044
{azizpour, razavian, sullivan, atsuto, stefanc}@csc.kth.se

Abstract

Evidence is mounting that CNNs are currently the most efficient and successful way to learn visual representations. This paper address the questions on why CNN representations are so effective and how to improve them if one wants to maximize performance for a single task or a range of tasks. We assess experimentally the importance of different aspects of learning and choosing a CNN representation to its performance on a diverse set of visual recognition tasks. In particular, we investigate how altering the parameters in a network's architecture and its training impacts the representation's ability to specialize and generalize. We also study the effect of fine-tuning a generic network towards a particular task. Extensive experiments indicate the trends; (a) increasing specialization increases performance on the target task but can hurt the ability to generalize to other tasks and (b) the less specialized the original network the more likely it is to benefit from fine-tuning. As by-products we have learnt several deep CNN image representations which when combined with a simple linear SVM classifier or similarity measure produce the best performance on 12 standard datasets measuring the ability to solve visual recognition tasks ranging from image classification to image retrieval.

1 Introduction

The history of deep CNNs traces back to early work on digit and character recognition [11, 16]. However, prior to 2012 there was skepticism within the computer vision community that they could be trained to solve difficult visual recognition problems bedevilled with clutter and large intra-class variations. But this perception has been radically altered by the success since 2012 of deep networks [15, 12, 29] trained with large scale datasets, such as ImageNet [1], to solve the hardest visual recognition tasks, see figure 1.

Excitingly deep CNNs can also learn powerful generic image representations[27, 8, 21]. These representations can be exploited very simply to solve a large range of recognition tasks [27]. In fact the performance of these representations are so good that at this juncture in computer vision, if you want to solve a visual recognition task you should first try a *deep CNN image representation* combined with a *simple classifier* [27, 8, 12, 29, 30, 33]. In this paper our experimental results overwhelmingly re-confirm this message. In 12 of the 14 diverse standard computer vision databases the approach just described, with the same representation in all cases, outperforms all reported non-CNN based methods, tables 5 and 6 in Appendix A.4. But if you follow this advice which deep CNN representation should you use to maximize performance for a particular task or set of tasks?

To maximize performance for a single task one can train a highly specialized deep network by using a large amount of labelled training data similar to the task you want to solve at test time. Examples of such an approach are the facial recognition network DeepFace [30] and the visual word recognition system PhotoOCR [5]. Both train an appropriately sized deep network with a large amount of data and the results they obtain are impressive. However, it is far from ideal to have to collect a huge number of labelled data and train a large network for every task you consider.

Therefore we rephrase the question as which deep CNN representation should I use to maximize performance for a particular task or set of tasks if I have limited labelled training data outside of ImageNet? There is no simple answer as it requires an understanding of the interplay between the size of the network and its architecture, the amount and diversity of labelled training data and the diversity of the tasks the network is trained to solve. The ability of deep network's representation has been also studied from various perspectives [4] including transfer learning [18] and domain adaptation [28]. The concept of learning from related tasks itself is not new and has appeared earlier in the literature including those on neural networks and CNN; see [25, 3, 13, 17] for a few examples. Another direction to efficiently help improve performance is to learn a *generic* representation followed by fine-tuning of the representation to the particular task when appropriate task specific labelled training data is available. In this paper we examine these factors experimentally and analyze the results through the idea of generic and specialized representations.

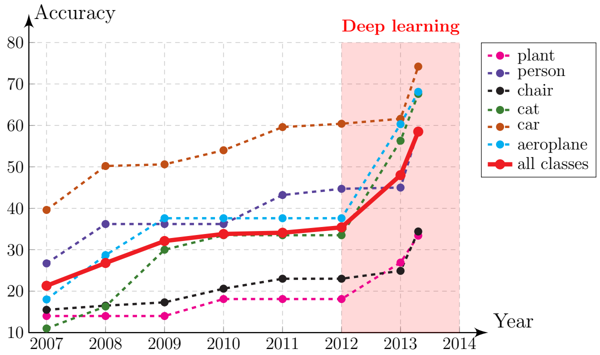


Figure 1: Convolutional Neural Networks (CNN) can be used to learn image representations for visual recognition. Recent results using CNN image representations show large performance gains for the hardest visual recognition tasks. These representations can serve as an off-the-shelf **generic** representation but also benefit from domain adaptation to achieve **specific** representation for each task. Thus a closer study of these different kinds of deep representations has become necessary and is the focus of this paper. The graph above plots the accuracy of the best performing method on the PASCAL VOC 2007 Object Detection dataset for each year since 2007. Note in the past two years with the advent of deep learning there has been sharp jump in the quality of the results. Girshick et al.[12], using a deep CNN representation, produced a bigger improvement than was cumulatively achieved in the years prior to 2012.

We call a representation specialized if it can solve a task, close to the one its CNN was trained to solve, with high accuracy, whereas we term the representation generic if it can accurately solve tasks far from those it was originally trained to solve. Throughout the paper we consider representations from CNNs, with architectures similar to `AlexNet`[15] such as `Caffe`[8] and `OverFeat` [29], that are trained on ImageNet to perform image classification. We measure the generality of a representation by measuring its performance across 14 diverse standard databases taken from the fields of visual recognition and retrieval. The first set of experiments investigate how correlated a representation's specialization and generalization ability is with the size of its underlying CNN, the number of iterations used during the back-propagation training and the layer of the network used for the representation. A summary of the findings for these experiments can be found in section 2.

The next set of experiments examine if it is possible to increase the performance of a generic CNN representation by fine-tuning it towards a particular task of interest [21, 12, 7]. Fine-tuning involves these steps. Replace the original output layer of a trained network with an output layer matching the labels for the task of interest. Then use your limited labelled training data in the backpropagation training process to update the weights of the network. The fine-tuning strategy is a way to leverage the limited amount of labelled training data you have for a specific task and to by-pass the time-consuming network training from scratch. Our experiments show that specialization of a network separately towards each task defined by 9 visual recognition databases improves the performance of the generic representation from 1% to 5%.

Crucially, all the representations learnt in this paper can be easily re-produced by others in the community given the software package Caffe software and following the simple procedures outlined in this paper. From performing these experiments we have amassed know-how of how to train and design these CNNs for visual recognition and retrieval problems (see tables 5 and 6 in Appendix A.4) and these are summarized in the next section for the practitioner and researcher alike.

2 Outline of the paper, its findings and contributions

As we are interested in visual recognition tasks in our experiments we use the deep networks `Caffe`, and `OverFeat` and similar networks, also trained to perform object classification using ImageNet. First we investigate how altering the parameters in a network's architecture and its training impacts the network's representation's ability to specialize and generalize. The different factors we investigate are the size of the network, the number of iterations used during the backpropagation training, the layer in the network used for the representation and the effect of augmenting the test and training datasets with jittering. The extensive experiments indicate the trend that increasing specialization increases performance on the target task but can hurt the ability to generalize to other tasks. In more detail the results of these experiments give definite indications that you should:

1. Over-parametrize your network when you know the final task and have large scale labelled training data close to that final task, see subsection 3.1.1.
2. Try earlier layers of the network to produce your CNN representation when your test task is far from the task your CNN was trained to solve, see subsection 3.1.2.
3. Monitor the network's loss score on validation sets during training and schedule the decrease in the learning rate appropriately. Doing this you can decrease the number of SGD training iterations by orders of magnitude without any loss in performance, see subsection 3.1.3.
4. Always do jittering both during training and testing. It makes the distribution around the mean more compact and helps to train a more discriminative representation and increases test time performance, see subsection 3.1.4 and 4.2.

Our next set of experiments is with regard to the effect of fine-tuning a generic network towards a particular task. Once again we focus on large deep networks originally trained to perform object classification. The overall trend of the results of these experiments is that the less specialized the original network the more likely it is to benefit from fine-tuning, for a more nuanced review of the results see section 3.2.1. More specially these experiments highlight that you should

1. Always try fine-tuning when you have a specific task at hand, even with a low number of training samples. It does not take much time and in the worst scenario has no effect on performance, see section 3.2.1.
2. Increase the data you use as much as possible for fine-tuning. Do not stop until you observe a saturation in performance. There appears to be the general rule that training with more data always improves the results, see section 3.2.2.

Alongside our extensive quantitative experiments is a statistical and separability analysis of the feature representations we learn when applied to images from different datasets. We analyze and visualize the representation space generated by both the generic and fine-tuned deep networks and compare them to other representations used in vision, see section 4. They highlight the fact that CNN representations cluster samples with the labels more than traditional vision representations.

3 Experiments - Factors relevant to a CNN's image representation

Evidence is mounting that CNNs are currently the most efficient and successful way to learn visual representations[1]. Razavian et al.[27] showed that a CNN representation can be very effective for many visual recognition tasks if the CNN is optimized for object image classification on a large set of labelled images (in particular the very large `OverFeat` network [29]). However, it is still

[1] It should be mentioned that by treating a CNN as a *representation learning* technique, one can apply most (if not all) previously established visual classification methods using this new representation space.

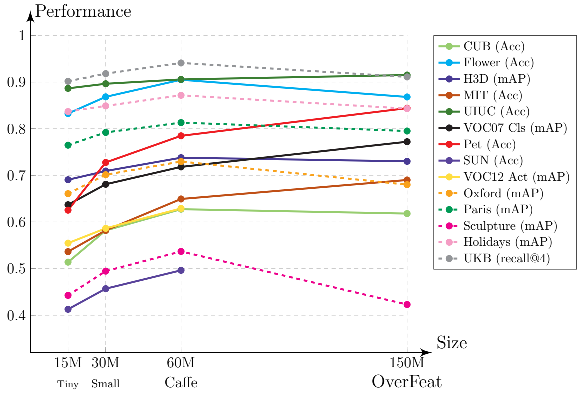


Figure 2: Over-parametrized networks (`OverFeat`) can be effective when the target task is close to the labelled data. However, performance on more distant tasks can suffer from over-specialization when the number of network parameters is increased. Though overall under-parametrized networks (Tiny) are unable to generalize as well. Since the Tiny network has 10 times fewer parameters than `OverFeat`, while preserving most of the performance, it could be useful for scenarios where real-time computation is an issue.

unclear why CNN representations are so effective and how to improve them if one wants to maximize performance for a single task or a range of tasks. Is it just a case of more data and bigger networks? Or do these questions require more subtle answers? We begin to address these questions by assessing experimentally the importance of different aspects of learning and choosing a CNN representation to its performance on a diverse set of visual recognition tasks.

The wide range of visual recognition tasks we consider are: **1**) Object image classification (pascal VOC 2007 [9]), **2**) Scene image classification (MIT indoor scenes 67[26], SUN 397 scenes [32]), **3**) Object attribute detection (H3D human attributes [6], UIUC 64 object attributes [10]), **4**) Fine grained category recognition (Oxford 102 flowers [19], Caltech-UCSD birds[31], Oxford pets [22]), **5**) Instance retrieval (Oxford buildings [23], Paris buildings [24], Sculptures 6k [2], Holidays [14], UK Benchmark [20]), **6**) Still image action classification (PASCAL VOC 2012). The range of the visual recognition tasks is chosen so that they increasingly deviate from the original task the CNN representation was optimized to perform. Our experiments investigate the effect of different settings in training a CNN representation on performance on our visual recognition tasks in particular: **a**) Size of the network. **b**) Different number of training iterations and the layer of the network used for the representation. **c**) Training iteration versus network size. **d**) Fine-tuning the representation towards the final task. **e**) Effect of jittering samples for training and testing the final classifier exploiting a CNN representation. **f**) Effect of adding more data during the fine-tuning process.

3.1 Factors controlling the genericity of a CNN image representation

In our first set of experiments we explore the factors that effect the image representation produced by a CNN when it is trained from labelled data covering a wide range of tasks.

3.1.1 Network size

The CNN `AlexNet`[15], the first very large network successfully applied to the ImageNet challenge, has an order of 60 million parameters consisting of ∼5 million parameters in the convolution layers and ∼55 million parameters in its fully connected layers. Although this appears to be an infeasibly large parameter space the network was successfully trained using the ImageNet dataset of 1.2 million images labelled with 1000 semantic classes. More recently networks larger than `AlexNet` have been trained in particular `Caffe`[8] and `OverFeat`[29]. Which of these networks produce the best generic image representation and how important is its size to its performance?

Therefore here we examine the impact of the network's size on different tasks including the original ImageNet image-level object classification. We trained 3 networks (using the Caffe software) of different sizes using the ILSVRC 2012 dataset and also included the `OverFeat` network in our experiments as the large network. Each network has roughly twice many parameters as we progress from the smallest to the largest network. For all the networks we kept the number of neurons in the 6th layer, the first fully connected layer, to 4096. It is this layer we use for the experiments where we directly compare networks (Table 1 in the appendix). The number of parameters is changed mainly by halving the number of kernels and the number of fully connected neurons (except the fixed one).

Figure 2 displays the effect of changing the network size on different visual recognition tasks/datasets (table of the exact numbers is available in the supplementary material). The largest network works best for Pascal VOC object image classification, MIT 67 indoor scene image classification, Oxford pets dataset and UIUC object attribute. The common feature of these datasets is that their semantic labels are either directly present in the ILSVRC12 set (true for Pascal VOC and Oxford dogs) or are shared between disjoint groups of ILSVRC12 set (e.g. has wheel, has legs, etc. for UIUC attributes) or are simple compositions of the ILSVRC12 label set (book+chair+table+shelves→library, pan+stove+oven→kitchen for MIT indoor scenes). On the other hand, for all the retrieval tasks the performance of the over-parametrized `OverFeat` network consistently suffers because it appears the generality of its representation is less than the smaller networks. Another interesting observation is that, if the computational efficiency at test time is critical, one can decrease the number of network parameters by orders of 2 (Small or Tiny network) for different tasks but the degradation of the final performance is linear and insignificant in some cases. Tables 2 and 3 in the appendices have the exact numbers for different tasks and network sizes.

3.1.2 Network layer

Different layers of a CNN potentially encode different levels of abstraction. The first convolutional layer is usually a collection of Gabor like gray-scale and RGB filters. On the other hand the output layer is directly activated by the semantic labels used for training. It is expected that the intermediate layers span the levels of abstraction between these two extremes. Therefore, we used the output of different layers as the representation fed into the our tasks' training/testing procedures. The performance of different layers of the pre-trained CNN (size: `Caffe`) on ImageNet is demonstrated in figure 3 for multiple tasks (for the table of exact numbers refer to the supplementary material). Observe the same pattern as for the effect of network size. The last layer (1000-way output) is only effective for the PASCAL VOC classification task. In the VOC task the semantic labels are almost a subset of those in ILSVRC12. The second fully connected layer (Layer 7) is most effective for the UIUC attributes (disjoint groups of ILSVRC12), Oxford Pet (dogs are a subset of ILSVRC12 while cats are not) and MIT indoor scenes (simple composition of ILSVRC12 classes). The first fully connected layer (Layer 6) works best for the rest of the datasets which have semantic labels further away from those used for optimizing the CNN representation. An interesting observation is that the first fully connected layer demonstrates a good trade-off when the final task is unknown and thus is the most generic layer within the scope of our tasks/datasets. Refer to Table 4 in appendix A.3 for the actual numbers.

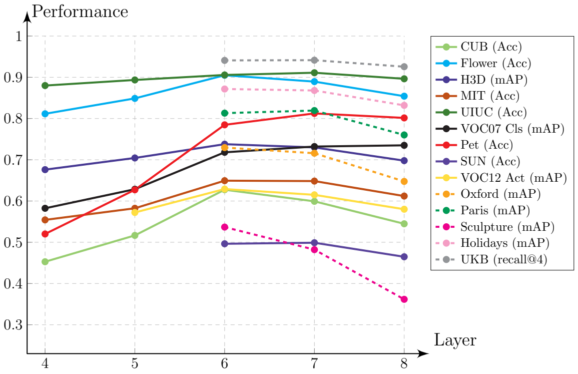
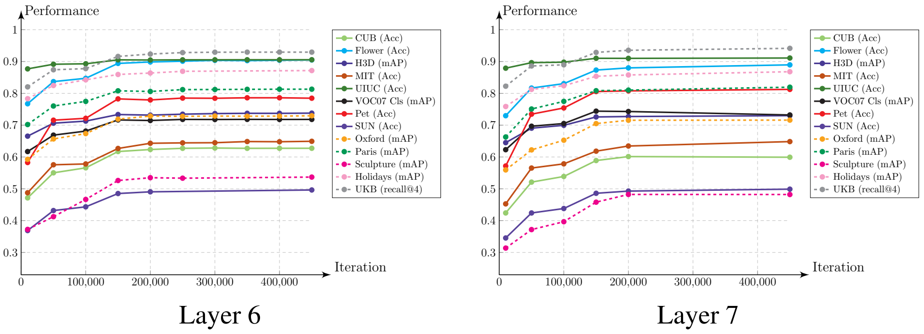


Figure 3: Efficacy of representations extracted from `Caffe`'s different layers for different visual recognition tasks. A distinct pattern can be observed: the further the task moves from object image classification, the earlier layers are more effective.

3.1.3 Training iterations

Figure 4 shows the evolution of the performance at different training iterations for multiple tasks. The performance of all tasks saturates at 200K iterations for all the layers and even earlier for some tasks/layers.

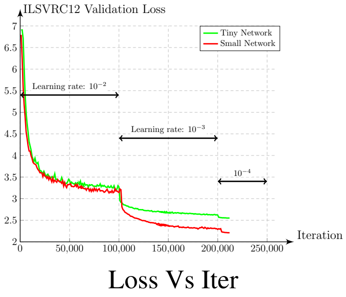


Layer 6

Layer 7

Figure 4: Effect of the number of iterations in the SGD during initial training on the performance of the representation extracted from layer 6 and layer 7 of the `Caffe` CNN.

We further look at the evolution of loss on the ILSVRC12 validation set at different iterations in figure 5. The validation performance also saturates before the final 450K iterations. The most interesting observation is that most of the performance boost is achieved at iterations a multiple of 100K. These iterations are where the learning is decreased by a factor of 10. We observed that the network, in most of the iterations, just wanders around a local minima due to a large learning rate. Thus, a more intelligent learning parameter schedule is necessary for faster training of the network.



Loss Vs Iter

Figure 5: The plot shows the amount by which the loss on the ImageNet validation set decreases during training iterations.

3.1.4 Jittering

Jittering the CNN representation for both the training and test samples input to the simple classifier consistently improves performance [27] (table 2). For us jittering corresponds to cropping, flipping and rotating image a few times and using the average of all the jittered samples as the original sample representation. The positive effect of jittering can be attributed to the low (single) number of modes in the representation space for each semantic class. If this assumption is true, then jittering potentially takes each sample closer to the mean of its class distribution which makes its separability from other classes more effective. This property is examined in more detail in Section 4.

3.2 Specializing a CNN image representation

Frequently the goal is to maximize the performance of a recognition system for a specific task or a set of tasks. In this case intuitively specializing the CNN to solve your task of interest would be the most sensible path to take. Here we focus on the issue of fine-tuning the CNN's representation with labelled data similar to those we expect to see at test time.

3.2.1 Fine-tuning

[12, 7] have shown that fine-tuning the network on a target task helps the performance. Fine-tuning is done by initializing a network with weights optimized for ILSVRC12. Then, using the target task training set, the weights are updated. The learning rate used for fine-tuning is typically set to be less than the initial learning rate used to optimize the CNN for ILSVRC12. This ensures the features learnt from the larger dataset are not forgotten. The step used to shrink the learning rate schedule is also decreased to avoid over-fitting. We have conducted fine-tuning on the tasks for which labels are mutually exclusive. The table in figure 6(**a**) shows the results. The gains made by fine-tuning increase as we move further away from the original image-level object classification task. Fine-tuning on a relatively small target dataset is a fast procedure. With careful selection of learning parameters it is always at least marginally helpful.

Representation	MIT	CUB	Flower
Caffe FC7 jitter	65.9	62.9	90.4
Caffe FT jitter	**66.3**	**66.4**	**91.4**

(**a**) Fine-tuning helps performance

Representation	bird	cat	dog
CNN [12]	38.5	51.4	46.0
CNN-FT VOC [12]	50.0	60.7	56.1
CNN-FT VOC+CUB+Pet	**51.3**	**63.0**	**57.2**

(**b**) Relevant fine-tuning helps detection.

Figure 6: (**a**) Results for fine-tuning a representation towards different target tasks. The CNN is originally trained on ImageNet for object image classification (row 1). Second row shows the results when we further optimize the learnt representation separately toward each task using a lower learning rate. Fine-tuning shows consistent effectiveness. However, the boost is higher for the more distant tasks from ImageNet (bird subcategory recognition achieves 3.5% while MIT indoor scenes is boosted only by 0.4%). (**b**) The table presents the mAP accuracy of a sliding window detector based on different CNN representations combined with a linear SVM classifier for 3 object classes from the 2007 Pascal VOC detection challenge. ImageNet (CNN) contains more than 100,000 dog images and Pascal VOC has 510 dog instances. For the representation in the second row, image patches extracted from the VOC training set, each with one of 21 different labels (20 object labels + the background class), are used to fine-tune the CNN representation towards solving the Pascal VOC detection challenge[12]. It results in a big jump in performance. But fine-tuning the network by also including the very relevant cat, dog and bird images from the Oxford Pet and Caltech bird datasets boosts the final detection performance on these classes even further.

3.2.2 Increasing training data for fine-tuning

To measure the effect of adding more data to learn the representation we consider the challenging task of PASCL VOC 2007 object detection. We follow the procedure of Girshick et al.[12] by fine-tuning the `Caffe` network using samples from the Oxford pet and Caltech-UCSD birds datasets. We show that although there exists a large number of samples for those classes in ImageNet (more than 100,000 dogs) adding around ∼3000 dogs from the Oxford pet dataset helps the detection performance significantly. The same improvement is observed for cat and bird, see the table in figure 6(**b**). This further adds to the evidence that specializing a CNN representation by fine-tuning, even when the original task contained the same labels, is helpful.

4 Analysis

The experimental results from section 3, in combination with tables 6 and 5, provide strong evidence that a CNN image representation is very powerful independent of whether it has been learnt generically or has been specialized towards a specific task or set of tasks. In this section we present both quantitative and qualitative results to give some intuition about the CNN representation space. Our results indicate that images from the same semantic class are grouped into distinct clumps in the CNN representation space and these clumps are made more distinct and compact by extended training iterations and jittering. This clustering in CNN feature space is in marked contrast to the HOG feature space where significant overlap occurs between datapoints from different classes and there is a significant distance between points from the same class. To highlight these issues we focus on images from the Caltech-2011-200 Birds and Oxford Pet dataset. These datasets include three classes (bird, cat, dog) and 237 sub-ordinate classes.

4.1 Quantitative measure

First we quantitatively measure the efficacy of the CNN representation compared to HOG features. As we are interested aiding our intuition we just focus on binary classification problem. Our two separate classes are cats and dogs samples from the Oxford Pet dataset. We fit a single multi-variate Gaussian distribution to the CNN representation of each class training set and measure the accuracy of a Bayes' classifier based on these two distributions on the test set (iter@10K: 88.8%, iter@450K: 90.7%, **iter@450K+Jitter: 93.1%**). For the HOG representation we allow a more flexible generative model and fit a Gaussian Mixture Models (GMM) with 1 to 50 mixtures and different random initializations. The idea here is to investigate if we can compensate for a less powerful image representation with more sophisticated generative modelling and final classifier. In this case we cannot as the best performance achieved is (HOG-GMM: 72.3%). Therefore a simple classifier combined with a CNN representation outperforms sophisticated mixture models using HOG representation. Even if we try to maximize HOG's performance by learning a non-linear classifier discriminatively, kernel SVM with an RBF kernel, the HOG representation only achieves (HOG RBF-SVM: 76.1%). Figure 7 displays the results for all the scenarios just described.

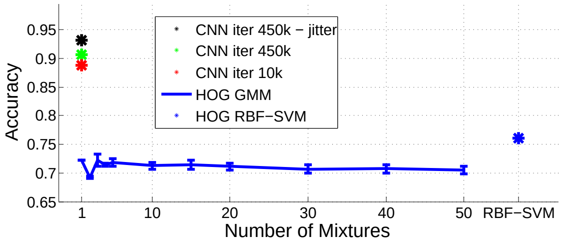
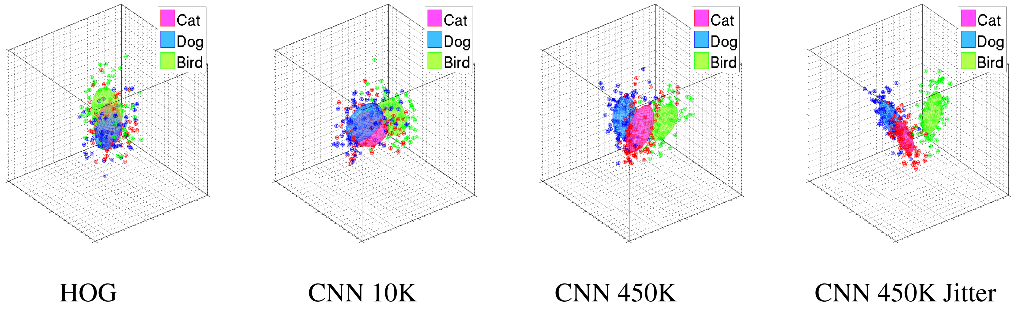


Figure 7: **CNN features cluster semantic classes better than HOG** This graph compares the accuracy of different CNN representations to a HOG representation for the binary classification task of distinguishing between cats and dogs. For each representation we fit a parametric probability distribution function to the training data from each class. The resulting Bayes' classifier is then computed and applied to the test data. For each CNN representation we fit a single multi-variate Gaussian as the class-conditional distribution and a GMM for the HOG representation. We plot the accuracy achieved by the HOG classifier as the number of components in the GMM increase. The CNN representations combined with a simple classifier derived from a simple generative model outperform by a large margin the HOG representation combined with a complicated classifier (derived generatively or discriminatively).

There are a couple of implementation details we should mention. The HOG descriptor from each image was extracted at a bounding box level to minimize of effect of clutter. The dimensionality of both representations, HOG and CNN, was reduced to 100 using PCA for all the experiments to allow a fair comparison.

4.2 Qualitative results

Using eigenvalue decomposition of the $L2$ distance matrix we apply classical Multi-Dimensional Scaling (MDS) to visually demonstrate the degree of separation of data points from different classes when encoded by different image representations. Figure 8 shows the result for the classes of "cat", "dog" and "bird", three different CNN representations and HOG. An ellipsoid is fit to the data points from each class to aid comparison. The original MATLAB .fig files for these plots are available in the supplementary material and with these you can change the viewing angle of the MDS plots. As expected, because of the increased classification performance, the low-dimensional representation produced by MDS for the CNN+Jitter representation has the least amount of overlap between the different classes.



HOG CNN 10K CNN 450K CNN 450K Jitter

Figure 8: Using Multi-Dimenensional Scaling (MDS) we embed 100 randomly selected samples from the classes of "cat", "dog" and "birds" in a 3D dimensional space. You can see the resulting separation of the samples when different representations are used. We also fit an ellipsoid to each class data to allow a better comparison. As we move from left to right the plots show there is less overlap between the 3 different classes and this indicates that this is also the case in the original representation spaces.

5 Conclusion and future work

The concrete conclusions we feel that can be drawn from our experiments for both the computer vision researcher and practitioner are: **1**) Over-parametrize your network when you know your final task and have large scale labelled training data close to that final task. **2**) Always try fine-tuning when you have a specific task to solve, even with a low number of training samples, it does not take time and it is, in the worst case, harmless. **3**) Always do jittering, it squeezes the distribution around the mean and helps the training and test time performance. **4**) Carefully select the learning rate parameters and you can decrease the number of required training iterations by orders of magnitude. **5**) Increase your fine-tuning training data as much as possible. Do not stop until you observe saturation: *more data achieves better results* (Object Detection). **6**) Last, but most importantly, at this moment in time you should replace the visual representation in your visual recognition task with a CNN representation.

Our experiments and analysis indicate that the following directions should be very fruitful for further exploration. First we should design and optimize multi-task networks to find better and more generic features. These multi-task networks should promote more feature sharing and help regularize over-parametrized networks. Embedding CNN representations within existing structured computer vision models should greatly improve the performance of these models. To conclude an intriguing question this paper opens but does not answer is whether it is ImageNet, the CNN architecture, or their combination that provides such a rich image representation?

Acknowledgments We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPUs used for this research.

References

[1] Imagenet large scale visual recognition challenge 2013 (ilsvrc2013). http://www.image-net.org/challenges/LSVRC/2013/.

[2] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *ICCV*, 2011.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.

[4] Y. Bengio, A. C. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[5] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In *ICCV*, pages 785–792, 2013.

[6] L. D. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011.

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arxiv:1405.3531 [cs.CV]*, 2014.

[8] 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. In *ICML*, 2014.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[10] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[11] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):93–202, 1980.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] S. Gutstein, O. Fuentes, and E. Freudenthal. Knowledge transfer in deep convolutional neural nets. *IJAIT*, 17(3):555–567, 2008.

[14] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

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

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of IEEE*, 86(11):2278–2324, 1998.

[17] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.

[18] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. J. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, P. Vincent, A. C. Courville, and J. Bergstra. Unsupervised and transfer learning challenge: a deep learning approach. In *ICML Unsupervised and Transfer Learning*, volume 27 of *JMLR Proceedings*, pages 97–110, 2012.

[19] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[20] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[22] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[25] L. Y. Pratt. Discriminability-based transfer between neural networks. In *NIPS*, 1992.

[26] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[27] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for visual recognition. In *CVPR workshop of DeepVision*, 2014.

[28] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

[30] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[32] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010.

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

A Tables with quantitative details and results

A.1 Details of the size of networks used in the experiments

Table 1: **Network size details** of the different CNNs used in our experiments. The description of the notation in the table: N_T is the total number of weights parameters in the network, n_k is the number of kernels at a convolutional layer and n_h is the number of nodes in a fully connected layer. For each network the output layer applies a SoftMax function and has 1000 output nodes.

		Convolutional layers			**FC layers**	
Network	N_T	#	n_k per layer	kernel sizes per layer	#	n_h per layer
Tiny	15M	5	(24, 64, 96, 96, 64)	(11×11, 5×5, 3×3, 3×3, 3×3)	2	(4096, 1024)
Small	30M	5	(48, 128, 192, 192, 128)	(11×11, 5×5, 3×3, 3×3, 3×3)	2	(4096, 2048)
Caffe	62M	5	(96, 256, 384, 384, 256)	(11×11, 5×5, 3×3, 3×3, 3×3)	2	(4096, 4096)
OverFeat	144M	6	(96, 256, 512, 512, 1024,4096)	(7×7, 7×7, 3×3, 3×3, 3×3, 5×5)	2	(4096, 1000)

A.2 Effect of network size on performance

Table 2: **Effect of network size** (**Classification tasks**). For tasks closer to object image classification the largest network (`OverFeat`) works better while as we move further away from that task the medium-sized network (`Caffe`) performs better.

Network	VOC 2007	MIT 67	SUN 397	UIUC	Cat/Dog	H3D	Birds	Flowers	Action
Tiny	63.7	53.6	41.3	88.7	62.5	69.1	514	83.3	55.4
Small	68.1	58.2	45.7	89.6	72.8	70.9	58.2	86.8	58.6
Caffe	71.8	64.9	50.0	90.6	78.5	**73.8**	**62.7**	**91.5**	**63.5**
OverFeat	**77.2**	**69.0**	—	**91.5**	**84.4**	73.0	61.8	86.8	—

Table 3: **Effect of network size** (**Retrieval tasks**). The medium sized network (`Caffe`) consistently performs better.

Network	Paris	Oxford	Sculpture	Holiday	UKBench
Tiny	76.5	66.1	44.3	83.7	90.2
Small	79.2	70.3	49.5	84.9	91.8
Caffe	**81.3**	**71.2**	**52.0**	**87.1**	**93.0**
OverFeat	79.5	68.0	42.3	84.3	91.1

A.3 Best Layer for different tasks

Table 4: **Best performing layer** (**Classification tasks**). For all the image-level classification tasks/datasets we compare the performance of the best performing layer to the first fully connected layer (FC6). In the last row the layer which results in the best performance is the number in parenthesis. As we move further away from the object image classification task, it is the earlier layers that perform better.

Representation	VOC 2007	MIT 67	SUN 397	UIUC	Cat/Dog	H3D	Birds	Flowers	Action
Caffe FC 6	71.6	64.9	49.6	90.5	78.4	73.8	62.7	90.5	62.9
Caffe Best Layer	75.9(8)	64.9(7,6)	49.9(7)	91.1(7)	81.2(7)	73.8(6)	62.7(6)	90.5(6)	62.9(6)

A.4 Performance: linear SVM + CNN representation Vs non-CNN state of art methods

Table 5: **CNN representation Vs non-CNN methods** (**Classification tasks**). Comparison of the `Caffe` representation baseline, best layer and non-CNN state of the art methods. The 3rd row of results corresponds to the best parameters of the CNN representation found for each task using a combination of the strategies proposed in section 3.

Representation	VOC 2007	MIT 67	SUN 397	UIUC	Cat/Dog	H3D	Birds	Flowers	Action
Caffe FC 6	71.6	64.9	49.6	90.5	78.4	73.6	62.7	90.5	62.9
Caffe FC 6 Jitter	72.2	65.5	—	90.8	78.4	73.6	63.2	91.0	—
Caffe Best results	**75.9**	**66.3**	**49.9**	**91.5**	**82.1**	**73.8**	**66.4**	**91.4**	62.9
non-CNN s.o.a.	71.1	64.0	47.2	90.2	59.2	69.9	56.8	80.7	**69.9**

Table 6: **CNN representation Vs non-CNN methods** (**Retrieval tasks**). Comparison of the `Caffe` representation baseline to non-CNN state of the art methods. CNN representation based retrieval outperforms state of the art methods on 4 out of 5 datasets.

Representation	Paris	Oxford	Sculpture	Holiday	UKBench
Caffe FC 6	**81.3**	71.2	**52.0**	**87.1**	**93.0**
non-CNN s.o.a.	78.2	**81.7**	45.4	82.2	89.3