From Generic to Specific Deep Representations for Visual Recognition

Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, Stefan Carlsson
KTH (Royal Institute of Technology)
{azizpour, razavian, sullivan, atsuto, stefanc}@csc.kth.se

Abstract

Evidence is mounting that ConvNets are the best representation learning method for recognition. In the common scenario, a ConvNet is trained on a large labeled dataset and the feed-forward units activation, at a certain layer of the network, is used as a generic representation of an input image. Recent studies have shown this form of representation to be astoundingly effective for a wide range of recognition tasks. This paper thoroughly investigates the transferability of such representations w.r.t. several factors. It includes parameters for training the network such as its architecture and parameters of feature extraction. We further show that different visual recognition tasks can be categorically ordered based on their distance from the source task. We then show interesting results indicating a clear correlation between the performance of tasks and their distance from the source task conditioned on proposed factors. Furthermore, by optimizing these factors, we achieve state-of-the-art performances on 16 visual recognition tasks.

1. Introduction

The history of convolutional networks (ConvNets) traces back to early work on digit and character recognition [11, 20]. Prior to 2012, though, in computer vision field, neural networks were more renowned for their propensity to overfit than for solving difficult visual recognition problems. And within the computer vision community it would have been considered ludicrous, given the overfitting problem, to think that they could be used to train image representations for transfer learning.

However, these perceptions have had to be radically altered by the experimental findings of the last three years. First, deep networks [19, 13], trained using large labelled datasets such as ImageNet [1], produce by a huge margin the best results on the most challenging image classification [1] and detection datasets [9]. Second, these deep ConvNets learn powerful generic image representations [32, 8, 24, 44] which can be used off-the-shelf to solve many visual recognition problems [32]. In fact the performance of these representations is so good that at this juncture in computer vision, a *deep ConvNet image representation* combined with a *simple classifier* [32, 13] should be the first alternative to try for solving a visual recognition task.

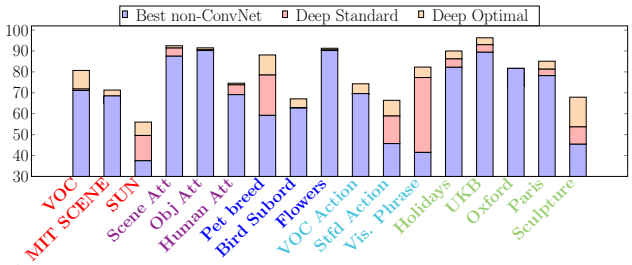

Figure 2: **The improvements achieved by optimizing the transferability factors are significant.** This optimization boosted the performance by up to **50%** relative error reduction. The violet bars show the performance of non-ConvNet state of the art systems on different datasets. The pink stacked bars shows the improvement when using off-the-shelf ConvNet features with standard settings and a linear SVM classifier. The burnt orange stacked bars show the boost gained by finding the best transferability factors for each task. Detailed results can be found in Table 6. The accuracy is measured using the standard evaluation criteria of each task, see the references in Table 2.

Following this advice you may ask, if I use ImageNet to train a deep ConvNet representation: *How can I then maximize the performance of the resulting representation for my particular target task?* The question becomes especially pertinent if you only have a limited amount of labelled training data, time and computational resources because training a specialized deep ConvNet from scratch is not an option. The question rephrased in more technical terminology is: how should a deep ConvNet representation be learned and adjusted to allow better transfer learning from a source task producing a generic representation to a specific target task? In this paper we identify the relevant factors and demonstrate, from experimentally evidence, how they should be set given the categorization of the target task.

The first set of factors that effect the transferability of a ConvNet representation are those defining the architecture and training of the initial deep ConvNet. These include the source task (encoded in the labelled training data), network width and depth, optimization parameters and whether you fine-tune the network using labelled data from the target task. The next set, after learning the "raw" representation, are what we term post-learning parameters. These include

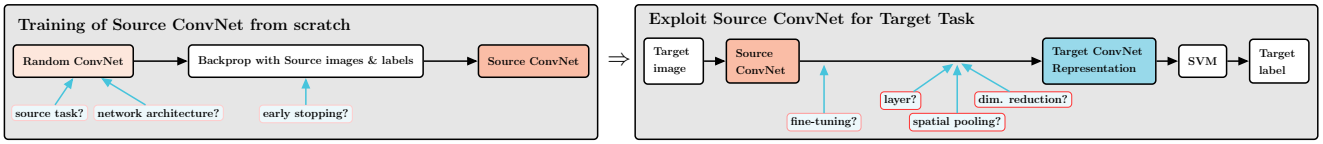

Figure 1: **Transferring a ConvNet Representation** ConvNet representations are effective for visual recognition. The picture above shows the pipeline of transferring a source ConvNet representation to a target task of interest. We define several factors which control the transferability of such representations to different tasks (questions with blue arrow). These factors come into play at different stages of transfer. Optimizing these factors is crucial if one wants to maximize the performance of the transferred representation (see Figure 2).

the network layer from which the representation is extracted and whether the representation should be post-processed by dimensionality reduction.

Figure 1 gives an graphical overview of how we transfer a ConvNet representation trained for a source task to target task and the factors we consider that affect its transferability and at what stage in the process the factors are applied. While figure 2 shows how big a difference an optimal configuration for these factors can make for 17 different target tasks.

How should you set these factors? Excitingly we observe that there is an interesting pattern for these factors. Their optimal settings are clearly correlated with the distance of the target task's distance from the source task. When occasionally there is an exception to the general pattern there is a plausible explanation. Table 1 lists some of our findings[1], driven by our quantitative results, and shows the best settings for the factors we consider and illustrates the correlations we mention.

To summarize deep ConvNet representation are very amenable to transfer learning. The concrete evidence we present for this assessment is that in 16 out of 17 diverse standard computer vision databases the approach just described, based on a deep ConvNet representation trained with ImageNet and optimal settings of the transferability settings, outperforms all published non-ConvNet based methods, see Table 6.

Contributions of the paper

- We systematically identify and list the factors that affect the transferability of ConvNet representation for visual recognition tasks (Table 1, Section 3).
- We provide exhaustive experimental evidence showing how these factors should be set (Table 1, Section 3).
- We show these settings follow an interesting pattern which is correlated with the distance between the source and target task, (Figures 3-7 in Section 3).

[1]Interestingly Yosinski et al.[43] just before submission reported on observing similar correlations for the choice of representation layer

[2]In general the network should be as deep as possible but in the final experiments a couple of the instance retrieval tasks defied this advice!

	Target task				
Factor	Source task ImageNet	···	Fine-Grained recognition	···	Instance retrieval
Early stopping	Don't do it				
Network depth	As deep as possible[2]				
Network width	Wider		Moderately wide		
Fine-tuning	Yes, more improvement with more labelled data				
Dim. reduction	Original dim		Reduced dim		
Rep. layer	Later layers		Earlier layers		

Table 1: **Best practices** to transfer a ConvNet representation trained for the source task of ImageNet to a target tasks summarizing some of our findings. The target tasks above are listed from left to right according to their increased distance from the source task (ImageNet image classification). The table summarizes qualitatively the best setting for each factor affecting a ConvNet's transferability given the target task. Although the optimal setting for some factors is similar for all tasks we considered, for other factors their optimal settings depend on the target task's distance from the source task. Table 2 shows the ordering of all tasks. For more detailed analysis please refer to section 3.

- By optimizing the transferability factors we significantly improve (up to 50% error reduction) state-of-the-art on 16 popular visual recognition datasets (Table 6) using a linear SVM for classification tasks and euclidean distance for instance retrieval.

Related Works

The concept of learning from related tasks using neural networks and ConvNets has appeared earlier in the literature see [28, 3, 14, 21] for a few examples. We describe two very recent papers which are the most relevant to our findings in this paper.

In [2] the authors investigate experimentally issues related to the training of ConvNets for the tasks of image classification (SUN image classification dataset) and object detection (PASCAL VOC 2007 & 2012). The result of two of their investigations are especially relevant to us. The first is that they show fine-tuning a network, pre-trained with the

Increasing distance from ImageNet →

Image Classification	Attribute Detection	Fine-grained Recognition	Compositional	Instance Retrieval
PASCAL VOC Object [9]	H3D human attributes [6]	Cat&Dog breeds [25]	VOC Human Action [9]	Holiday scenes [16]
MIT 67 Indoor Scenes [29]	Object attributes [10]	Bird subordinate [39]	Stanford 40 Actions [42]	Paris buildings [27]
SUN 397 Scene [41]	SUN scene attributes [26]	102 Flowers [23]	Visual Phrases [30]	Sculptures [4]

Table 2: Range of the 15 visual recognition tasks sorted categorically by their similarity to ILSVRC12 object image classification task.

ImageNet dataset, towards a target task, image classification and object detection, has a positive effect and this effect increases when more data is used for fine-tuning. They also show that when training a network with ImageNet one should not perform early stopping even if one intends to transfer the resulting representation to a new task. These findings are consistent with a subset of ours though our conclusions are supported by a larger and wider set of experiments including more factors.

Yosinski et al. [43] interestingly show that the transferability of a network trained to perform one source task to solve another task is correlated with the distance between the source and target tasks. Yosinski et al.'s source and target tasks are defined as the classification of different subsets of the object categories in ImageNet. Their definition of transferability comes from their training set-up. First a ConvNet is trained to solve the source task. Then the weights from the first n layers of this source network are *transferred* to a new ConvNet that will be trained to solve the target task. The rest of the target ConvNet's weights are initialized randomly. Then the random weights are updated via fine-tuning while the transferred weights are kept fixed. They show that for larger n the final target ConvNet, learned in this fashion, performs worse and the drop in performance is bigger for the target tasks most distant from the source task. This result corresponds to our finding that the performance of the layer used for the ConvNet representation is correlated to the distance between the source and target task. Yosinki et al. also re-confirm that there are performance gains to be made by fine-tuning a pre-trained network towards a target task. However, once again our results are drawn from a wide range of target tasks which are being used in the field of computer vision. Furthermore, we have investigated many more factors than just the representation layer as listed in Table 1.

2. Range of target tasks examined

To evaluate the transferability of the ConvNet representation we use a wide range of 17 visual recognition tasks. The tasks are chosen from 5 different subfields of visual recognition: object/scene image classification, visual attribute detection, fine-grained classification, compositional semantic recognition, instance retrieval (see Table 2). There are multiple ways one could order these target tasks based on their similarity to the source task of object image classification as defined by ILSVRC12. Table 2 gives our ordering and we now give the rationale for the ranking.

The group of tasks we consider furthest from the source task is instance retrieval. Each task in this set has no explicit category information and is solved by explicit matching to exemplars. While all the other group of tasks involve classification problems and require an explicit learning phases.

We place attribute detection earlier than fine-grained recognition because these visual attributes[3] are usually the explanatory factors which separate the original object classes in ILSVRC and are thus expected to be naturally selected/highlighted by the ConvNet. Also, some attributes (*e.g.* four-legged) are defined as a superset of object classes (*e.g.* cat, dog, *etc.*). Another aspect of this pairwise ordering is that fine-grained recognition involves sometimes very subtle differences between members of a visual category. We suspect that a network trained for higher levels of object taxonomy (*e.g.* flowers in general) would not be sensitive to micro-scale visual elements necessary for fine-grained recognition.

Next comes perhaps the most interesting and challenging set of category tasks – the compositional recognition tasks. These tasks include classes where the type of interactions between objects is the key indicator and thus requires more sophistication to recognize than the other category recognition tasks.

There are other elements which determine the closeness of a target task to the source task. One is the distribution of the semantic classes and images used within each category. For example the Pet dataset [25] is the closest of the fine-grained tasks because the ILSVRC classes include many different dog breeds. While, sometimes the task just boils down to the co-occurrence of multiple ILSVRC classes like the MIT indoor scenes. However, compositional recognition tasks usually encode higher level semantic concepts to be inferred from the object interactions, for instance a person holding violin is not considered a positive sample for playing the violin in [9] nor is a person standing beside a horse considered as the action "riding horse".

[3] As an aside, depending on the definition of an attribute, the placement of an attribute detection task could be anywhere in the spectrum. For instance, one could define a fine-grained, local and compositional attribute which would then fall furthest from all other tasks (*e.g.* "wearing glasses" in H3D dataset).

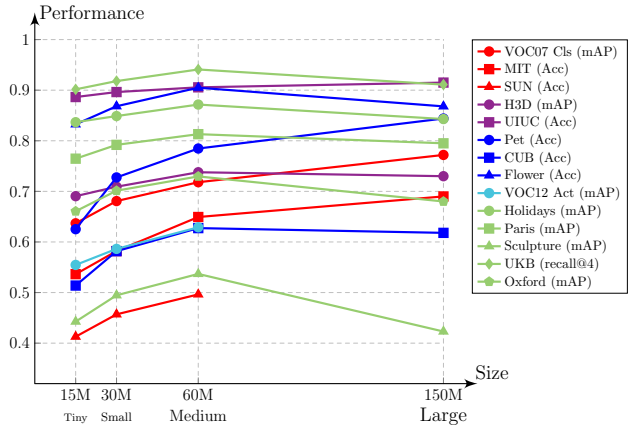

Figure 3: **Network Width:** Over-parametrized networks (`OverFeat`) can be effective when the target task is close to the labelled data. However, the performance on more distant tasks can suffer from over-specialization when the number of network parameters is increased. But overall under-parametrized 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.

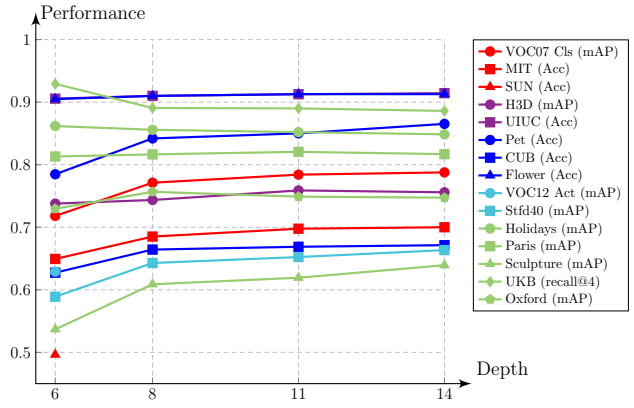

Figure 4: **Network Depth:** Over-parametrizing networks by the number of convolutional layers is effective for nearly all the target tasks. While a saturation can be observed in some tasks there is no significant performance drop as opposed to the trend observed when over-parametrizing by increasing its width. The number on the x-axis indicates the number of convolutional layers of the network. The representation is taken from the first fully connected layer right after the last convolutional layer.

3. Experiments

Now, we analyze the effect of each individual factor on the transferability of the learnt representation. We divide the factors into those which should be considered before learning a representation (learning factors) and those which should be considered when using an off-the-shelf network model (post-learning factors).

3.1. Learning Factors

3.1.1 Network Width

The ConvNet `AlexNet`[19], the first very large network successfully applied to the ImageNet challenge, has around 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 unfeasibly 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 `OverFeat`[31]. Which of these networks produces the best generic image representation and how important is its size to its performance?

Here we examine the impact of the network's size (keeping its depth fixed) on different tasks including the original ImageNet image-level object classification. We trained 3 networks 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 as many parameters as we progress from the smallest to the largest network. For all the networks we kept the number of units in the 6th layer, the first fully connected layer, to 4096. It is this layer that we use for the experiments where we directly compare networks. 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 3 displays the effect of changing the network size on different visual recognition tasks/datasets. The largest network works best for Pascal VOC object image classification, MIT 67 indoor scene image classification, UIUC object attribute, and Oxford pets dataset. 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 those of 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 sublinear in some cases.

3.1.2 Network Depth

Increasing the network width (number of parameters at each layer) is not the only way of over-parameterizing a ConvNet. In fact, [36, 34] have shown that deeper convolutional networks with more layers achieve better performance on the ILSVRC14 challenge. In a similar spirit, we over-parametrize the network by increasing the number of convolutional layers before the fully connected layer from which we extract the representation. Figure 4 shows the results by incrementally increasing the number of convolutional layers from 5 to 13. As this number is increased, the performance on nearly all the datasets increases.

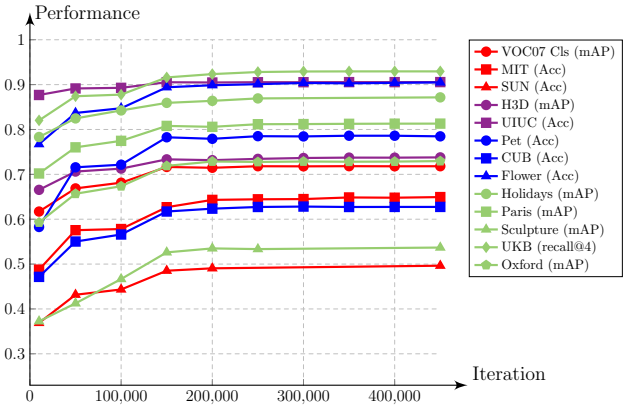

Figure 5: **Early Stopping:** Plotted above is the performance of the representation extracted from layer 6 of the `AlexNet` ConvNet versus the number of iterations of SGD used to train the initial network. It can be seen that early stopping, which can act as a regularizer, does not help to produce a more transferable representation.

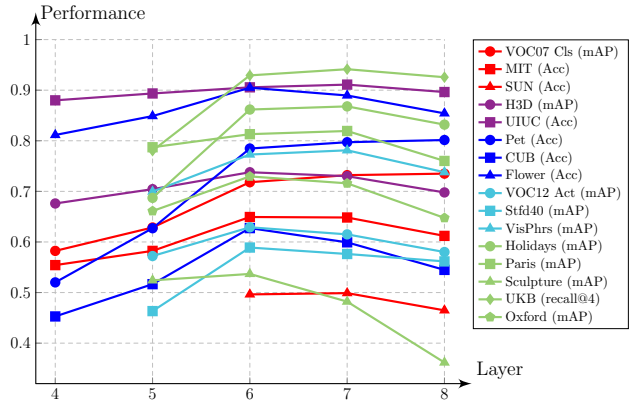

Figure 6: **Representation Layer:** Efficacy of representations extracted from `AlexNet`'s different layers for different visual recognition tasks. A distinct pattern can be observed: the further the task moves from object image classification, the earlier layers are more effective. For instance, layer 8 works best for PASCAL VOC image classification which is very similar to ImageNet while the best performance for all retrieval tasks is at layer 6.

The only tasks for which the results degrade are the retrieval tasks of UKB and Holidays. Interestingly, these two tasks involve measuring the visual similarity between specific instances of classes strongly presented in ImageNet (*e.g.* a specific book, bottle or musical instrument in UKB, and wine bottle, Japanese food in Holidays dataset). It is, thus, expected that the representation becomes more invariant to instance level differences as we increase the complexity of the representation with more layers.

If we compare the effect of increasing network depth to network width on the final representation's performance, we clearly see that increasing depth is a much more stable over-parametrization of the network. Both increasing width and depth improve the performance on tasks close to the source task. However, increasing the width seems to harm the transferability of features to distant target tasks more than increasing depth does. This could be attributed to the fact that increasing depth is a more efficient (in terms of the number of parameters) way of representing more complex patterns. Finally, more layers means more sequential processing which hurts the parallelization. We have observed the computational complexity for learning and using deep ConvNets increases super-linearly with the number of layers. So, learning a very wide network is computationally cheaper than learning a very deep network. These issues means the practitioner must decide on the trade-off between speed and performance.

3.1.3 Early Stopping

Early stopping is used as a way of controlling the generalization of a model. It is enforced by stopping the learning before it has converged to a local minima as measured by monitoring the validation loss. This approach has also been used to improve the generalization of over-parametrized networks [5]. It is plausible to expect that the transferability increases with generalization. Therefore, we investigate the effect of early stopping on the transferability of learnt representation. Figure 5 shows the evolution of the performance for various target tasks at different training iterations. The performance of all tasks saturates at 200K iterations for all the layers and even earlier for some tasks. Surprisingly, it can be seen that early stopping does not improve the transferability of the features whatsoever.

3.1.4 Source Task

As discussed earlier, the most important factor for a learnt representation to be generic is the properties of the source task and its distance from the target task. The recent development of another large scale dataset called the Places Dataset [47] labelled with scene classes enabled us to analyze this factor. Table 3 shows the results for different source tasks of ImageNet, Places, and a hybrid network. The hybrid network is made by combining the ImageNet images with those of the Places dataset. The label set is increased accordingly [47]. It can be observed that results for the tasks very close to the source tasks are improved with the corresponding models (MIT, SUN for Places network). Another interesting observation is that ImageNet features seem to achieve a higher level of generalization for further away tasks. One explanation is that the set of labels is more diverse. Since the number of images in ImageNet is smaller, it shows the importance of diversity of labels as opposed to the number of annotated images when the objective is to achieve a more transferable representation.

	Image Classification			Attribute Detection		Fine-grained Recognition			Compositional	Instance Retrieval		
Source task	VOC07	MIT	SUN	H3D	UIUC	Pet	CUB	Flower	Stanf. Act40	Oxf.	Scul.	UKB
ImageNet	71.6	64.9	49.6	73.8	**90.4**	**78.4**	**62.7**	**90.5**	58.9	71.2	52.0	93.0
Places	68.5	69.3	55.7	68.0	88.8	49.9	42.2	82.4	53.0	70.0	44.2	88.7
Hybrid	72.7	69.6	56.0	72.6	90.2	72.4	58.3	89.4	58.2	**72.3**	52.3	92.2
Concat	**73.8**	**70.8**	**56.2**	**74.2**	**90.4**	75.6	60.3	90.2	**59.6**	72.1	**54.0**	**93.2**

Table 3: **Source Task:** Results on all tasks using representations optimized for different source tasks. ImageNet is the representation used for all experiments of this paper. Places is a new ConvNet trained on 3.5M images labeled with scene categories [47]. Hybrid is a model proposed by Zhou et al. [47] which combines the ImageNet and Places datasets and train a single network for the combination. Concat indicates results of concatenating the feature obtained from ImageNet ConvNet and Places ConvNet for each input image. All results are for first fully connected layer (FC6).

The Hybrid model boosts the transferability of the Places network but still falls behind the ImageNet network for more distant tasks. This could be again due to the fact that the number of images from the Places dataset dominates those of the ImageNet dataset in training the Hybrid model and as a consequence it is more biased toward the Places Network. In order to avoid this bias, in another experiment, we combined the features obtained from the ImageNet network and the Places network as opposed to Hybrid network, and interestingly this late fusion works better than Hybrid model (the Hybrid model where the number of dimensions of the representation is increased to 8192 works worse [7]).

3.2. Post-learning Factors

3.2.1 Network Layer

Different layers of a ConvNet 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 for our tasks' training/testing procedures. The performance of different layers of the pre-trained ConvNet (size: Medium) on ImageNet is shown in Figure 6 for multiple tasks.

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 a subset of those in ILSVRC12, the same is true for the Pet dataset classes. The second fully connected layer (Layer 7) is most effective for the UIUC attributes (disjoint groups of ILSVRC12),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 ConvNet 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.

Although the last layer units act as probabilities for ImageNet classes, note that results using the last layer with 1000 outputs are surprisingly effective for almost all the tasks. This shows that a high order of image-level information lingers even to the very last layer of the network. It should be mentioned that obtaining results of instance retrieval on convolutional layers is computationally prohibitive and thus they are not included. However, in a simplified scenario, the retrieval results showed a drastic decrease from layer 6 to 5.

3.2.2 Dimensionality Reduction

We use principal component analysis (PCA) to reduce the dimensionality of the transferred representation for each task. We observed that dimensionality reduction helps all the instance retrieval tasks (most of the time insignificantly though). The main difference between the retrieval task and other ones is that in retrieval we are interested in the Euclidean distances between samples in the ConvNet representational space. In that respect, PCA can decrease the curse of dimensionality for $L2$ distance. However, one could expect that dimensionality reduction would decrease the level of noise (and avoid potential over-fitting to irrelevant features for each specific task). Figure 8 in the supplementary paper shows the results for different tasks as we reduce the dimensionality of ConvNet representations. The results show that the relative performance boost gained by additional dimensions is correlated with the distance of the target task to the original task. We see that saturations appear earlier for the tasks closer to ImageNet. It is amazing to know that effective dimensionality of the ConvNet representations (with 4096 dims) used in these experiments is at most 500 for all visual recognition tasks from different domains. Another interesting observation is that many of the tasks work reasonably well with very low number of dimensions (5-50 dimensions). Remember that these features are obtained by a *linear* transformation of the original ConvNet representation. This can indicate the capability of ConvNets

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

Table 4: **Fine-tuning:** The first row shows the original ConvNet results. The second row shows the results when we fine-tune the ConvNet toward the target task and specialize the learnt representation. Fine-tuning is consistently effective. The proportional improvement is higher for the more distant tasks from ImageNet.

in linear factorization of the underlying generating factors of semantic visual concepts.

3.2.3 Fine-tuning

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 ConvNet to solve the task of interest would be the most sensible path to take. Here we focus on the issue of fine-tuning the ConvNet's representation with labelled data similar to those we expect to see at test time.

[13, 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 ConvNet for ILSVRC12. This ensures that 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 4 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 parameters it is always at least marginally helpful.

3.2.4 Increasing training data

Zhu et al.[48] suggest that increasing data is less effective than increasing the complexity of models or richness of representation and the former is prone to early performance saturation. Those observations are made using HOG features to perform object detection. Here, we want to investigate whether we are close to saturation point with ConvNet representations.

To measure the effect of adding more data to learn the representation we consider the challenging task of PASCAL VOC 2007 object detection. We follow the procedure of Girshick et al.[13] by fine-tuning the `AlexNet` 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 4. This further adds to the evidence that specializing a ConvNet representation by fine-tuning, even when the original task contained the same labels, is helpful.

Representation	bird	cat	dog
ConvNet [13]	38.5	51.4	46.0
ConvNet-FT VOC [13]	50.0	60.7	56.1
ConvNet-FT VOC+CUB+Pet	**51.3**	**63.0**	**57.2**

Table 5: **Additional data (fine-tuning):** The table presents the mAP accuracy of a sliding window detector based on different ConvNet representations for 3 object classes from VOC 2007. ImageNet 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 are used to fine-tune the ConvNet representation[13]. It results in a big jump in performance. But including cat, dog and bird images from the Oxford Pet and Caltech bird datasets boosts the performance even further.

Furthermore, we investigate how important it is to increase training data for the original ConvNet training. We train two networks, one using SUN397 [41] with 130K images and the other using the Places dataset [47] with 2.5M images. Then we test the representations on the MIT Indoor Scenes dataset. The representation trained from SUN397 (**62.6%**) works significantly worse than that of the Places dataset (**69.3%**). The same trend is observed for other datasets (refer to Table 7 in supplementary material). Since ConvNet representations can model very rich representations by increasing its parameters, we believe we are still far from saturation in its richness.

4. Optimized Results

In the previous section, we listed a set of factors which can affect the efficacy of the transformed representation from a generic ConvNet. We studied how best values of these factors are related to the distance of the target task to the ConvNet source task. Using the know-hows obtained from these studies, now we transfer the ConvNet representations using "Optimized" factors and compare the "Standard" ConvNet representation used in the field. The "Standard" ConvNet representation refers to a ConvNet of medium size and depth 8 (`AlexNet`) trained on 1.3M images of ImageNet, with the representation taken from first fully connected layer (FC6). As can be seen in Table 6 the remaining error of the "Standard" representation can be decreased by a factor of up to 50% by optimizing its transferability factors.

	Image Classification			Attribute Detection			Fine-grained Recognition			Compositional			Instance Retrieval				
	VOC07	MIT	SUN	SunAtt	UIUC	H3D	Pet	CUB	Flower	VOCa.	Act40	Phrase	Holid.	UKB	Oxf.	Paris	Scul.
non-ConvNet	[35] 71.1	[22] 68.5	[40] 37.5	[26] 87.5	[38] 90.2	[45] 69.1	[25] 59.2	[12] 62.7	[18] 90.2	[24] 69.6	[42] 45.7	[30] 41.5	[37] 82.2	[46] 89.4	[37] **81.7**	[37] 78.2	[4] 45.4
Deep Standard	71.8	64.9	49.6	91.4	90.6	73.8	78.5	62.8	90.5	69.2	58.9	77.3	86.2	93.0	73.0	81.3	53.7
Deep Optimized[4]	**80.7**	**71.3**	**56.0**	**92.5**	**91.5**	**74.6**	**88.1**	**67.1**	**91.3**	**74.3**	**66.4**	**82.3**	**90.0**	**96.3**	79.0	**85.1**	**67.9**
Err. Reduction	32%	18%	13%	13%	10%	4%	45%	12%	8%	17%	18%	22%	28%	47%	22%	20%	31%
Source Task	ImgNet	Hybrid	Hybrid	Hybrid	ImgNet	ImgNet	ImgNet	ImgNet	ImgNet	ImgNet	ImgNet	ImgNet	Hybrid	ImgNet	ImgNet	ImgNet	ImgNet
Network Width	Medium	Medium	Medium	Medium	Large	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium
Network Depth[4]	16	8	8	8	8	16	16	16	16	16	16	16	8	8	16	16	16
Rep. Layer	last	last	last	last	2nd last	2nd last	2nd last	3rd last	3rd last	3rd last	3rd last	3rd last	4th last	4th last	4th last	4th last	4th last
PCA	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
Pooling	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	1×1	1×1	2×2	2×2	3×3

Table 6: **Final Results:** Final results of the deep representation with optimized factors along with a linear SVM compared to the non-ConvNet state of the art. In the bottom half of the table the factors used for each task are noted. We achieve up to a **50% reduction of error** by optimizing transferability factors. Relative error reductions refer to how much of the remaining error (from Deep Standard) is decreased. "Deep Standard" is the common choice of parameters - a Medium sized network of depth 8 trained on ImageNet with representation taken from layer 6 (FC6).

5. Implementation details

The `Caffe` software [17] is used to train our ConvNets. Liblinear is used to train the SVMs we use for classification tasks. Retrieval results are based on the $L2$ distance of whitened ConvNet representations. All parameters were selected using 4-fold cross-validation. Learning choices are the same as [32]. In particular, the pipeline for classification tasks is as follows: we first construct the feature vector by getting the average ConvNet feature vector of 12 jittered samples of the original image. The jitters come from crops of 4 corners of the original image, its center and the whole image resized to the size needed by the network (227x227) and their mirrors. We then $L2$ normalize the ConvNet feature vector, raise the absolute value of each feature dimension to the power of 0.5 and keep its sign. We use linear SVM trained using one-versus-all approach for multilabel tasks (e.g. PASCAL VOC image classification) and linear SVM trained using one-versus-one approach and voting for single label tasks (e.g. MIT Indoor Scene). The pipeline for the retrieval tasks are as follows: Following [15] The feature vectors are $L2$ normalized, then the dimensions are reduced using PCA and whitened and the resulting feature is renormalized to unit length. Since buildings (Oxford and Paris) and scupltures datasets include partial images or the object can appear in small part of the whole image (zoomed in or out images of the object of interest) we use spatial search to match windows from each pair of images. We have 1 sub-patch of size 100% of the whole image, 4 sub-patches of each covering 4/9 size of the image. 9 sub-patches of each covering 4/16 and 16 sub-patches of each covering 4/25 of the image. The minimum distance of all sub-patches is considered as the distance of the two images. For more details of the instance retrieval pipeline refer to [33].

[4] **Note:** "Deep Optimized" results in this table are not always the optimal choices of factors studied in the paper. For instance one would expect a very deep network trained using hybrid model would improve results on MIT and SUN, or a deep and large network would perform better on VOC image classification. Another example is that we could do fine-tuning with the optimal choices of parameters for nearly all tasks. Obviously, it was highly computationally expensive to produce all the existing results. We will update the next versions of the paper with further optimized choices of parameters.

6. Conclusion

ConvNet representations trained on ImageNet are becoming the standard image representation. In this paper we presented a systematic study, lacking until now, of how to effectively transfer such representations to new tasks. The most important elements of our study are: We identify and define several factors whose settings affect transferability. Our experiments investigate how relevant each of these factors is to transferability for many visual recognition tasks. We define a categorical grouping of these tasks and order them according to their distance from image classification.

Our systematic experiments have allowed us to achieve the following. First, by optimizing the identified factors we improve the state-of-the-art performance on a very diverse set of standard computer vision databases, see table 6. Second, we observe and present empirical evidence that the effectiveness of a factor is highly correlated with the distance of the target task from the source task of the trained ConvNet. Finally, we empirically verify that our categorical grouping and ordering of visual recognition tasks is meaningful as the optimal setting of the factors remain constant within each group and vary in a consistent manner across our ordering. Of course, there are exceptions to the general trend. In these few cases we provide simple explanations.

Acknowledgement This work has been funded by the Swedish Foundation for Strategic Research (SSF) within the project VINST. We gratefully acknowledge NVIDIA for donation of K40 GPU.

References

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

[2] P. Agrawal, R. B. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 2014. 2

[3] Andreas, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006. 2

[4] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *ICCV*, 2011. 3, 8

[5] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade (2nd ed.)*, pages 437–478, 2012. 5

[6] L. D. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011. 3

[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. 6, 7

[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. 1

[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. 1, 3

[10] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 3

[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. 1

[12] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, pages 1713–1720, 2013. 8

[13] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 7

[14] S. Gutstein, O. Fuentes, and E. Freudenthal. Knowledge transfer in deep convolutional neural nets. *IJAIT*, 17(3):555–567, 2008. 2

[15] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *ECCV*, pages 774–787, 2012. 8

[16] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 3

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

[18] P. Koniusz, F. Yan, P.-H. Gosselin, and K. Mikolajczyk. Higher-order Occurrence Pooling on Mid- and Low-level Features: Visual Concept Detection. Technical report, 2013. 8

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 4

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of IEEE*, 86(11):2278–2324, 1998. 1

[21] 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. 2

[22] D. Lin, C. Lu, R. Liao, and J. Jia. Learning important spatial pooling regions for scene classification. In *CVPR*, pages 3726–3733, 2014. 8

[23] 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. 3

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1, 8

[25] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 3, 8

[26] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proceeding of the 25th Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3, 8

[27] 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. 3

[28] L. Y. Pratt. Discriminability-based transfer between neural networks. In *NIPS*, 1992. 2

[29] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 3

[30] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. 2011. 3, 8

[31] 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. 4

[32] A. Sharif 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. 1, 8

[33] A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. A baseline for visual instance retrieval with deep convolutional networks. In *ICLR workshop proceedings*, 2015. 8

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4, 12

[35] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011. 8

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 4

[37] G. Tolias, Y. S. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, pages 1401–1408, 2013. 8

[38] G. Tsagkatakis and A. E. Savakis. Sparse representations and distance learning for attribute based category recognition. In *ECCV Workshops (1)*, pages 29–42, 2010. 8

[39] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 3

[40] J. Xiao, K. A. Ehinger, J. Hays, A. Oliva, and A. Torralba. Sun database: Exploring a large collection of scene categories. In *IJCV*, 2014. 8, 11

[41] 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. 3, 7

[42] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei. Action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011. 3, 8

[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *arXiv:1411.1792 [cs.LG]*, 2014. 2, 3

[44] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. 1

[45] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 8

[46] W.-L. Zhao, H. Jégou, G. Gravier, et al. Oriented pooling for dense and non-dense rotation-invariant features. In *BMVC*, 2013. 8

[47] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. *NIPS*, 2014. 5, 6, 7, 11

[48] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, pages 1–11, 2012. 7