Convolutional Network Representation for Visual Recognition

ALI SHARIF RAZAVIAN

Doctoral Thesis
Stockholm, Sweden, 2017
Abstract

Image representation is a key component in visual recognition systems. In visual recognition problem, the solution or the model should be able to learn and infer the quality of certain visual semantics in the image. Therefore, it is important for the model to represent the input image in a way that the semantics of interest can be inferred easily and reliably. This thesis is written in the form of a compilation of publications and tries to look into the Convolutional Networks (ConvNets) representation in visual recognition problems from an empirical perspective.

Convolutional Network is a special class of Neural Networks with a hierarchical structure where every layer’s output (except for the last layer) will be the input of another one. It was shown that ConvNets are powerful tools to learn a generic representation of an image. In this body of work, we first showed that this is indeed the case and ConvNet representation with a simple classifier can outperform highly-tuned pipelines based on hand-crafted features. To be precise, we first trained a ConvNet on a large dataset, then for every image in another task with a small dataset, we feed-forward the image to the ConvNet and take the ConvNets activation on a certain layer as the image representation. Transferring the knowledge from the large dataset (source task) to the small dataset (target task) proved to be effective and outperformed baselines on a variety of tasks in visual recognition. We also evaluated the presence of spatial visual semantics in ConvNet representation and observed that ConvNet retains significant spatial information despite the fact that it has never been explicitly trained to preserve low-level semantics.

We then tried to investigate the factors that affect the transferability of these representations. We studied various factors on a diverse set of visual recognition tasks and found a consistent correlation between the effect of those factors and the similarity of the target task to the source task. This intuition alongside the experimental results provides a guideline to improve the performance of visual recognition tasks using ConvNet features. Finally, we addressed the task of visual instance retrieval specifically as an example of how these simple intuitions can increase the performance of the target task massively.

Keywords: Convolutional Network, Visual Recognition, Transfer Learning.
List of Papers

The thesis is based on the following papers:


This paper is an extension of the following award winning paper:


This paper is an extension to the following one:

In addition to the papers [A]-[D], the author of this thesis has contributed to the following papers:


---

1I designed some of the pipelines, network architectures and factors. Also, I performed the experiments for hundreds out of approximately a thousand reported results in this paper.
## Contents

### I Introduction

1 Introduction
   1 Tasks .................................................. 4
   2 Pipeline .................................................. 6
   3 Factors .................................................. 7
   4 Summary .................................................. 9

### 2 Background

1 Descriptors ................................................. 11
2 Encoding Methods .......................................... 14
3 Convolutional Networks ................................. 15
4 Comparison .................................................. 17

### 3 Summary of papers

- A CNN Features off-the-shelf: an Astounding Baseline for Recognition .... 19
- B Persistent Evidence of Local Image Properties in Generic ConvNets .......... 21
- C Factors of Transferability for a Generic ConvNet Representation .......... 23
- D Visual Instance Retrieval with Deep Convolutional Networks ................ 25

### 4 Conclusion

References .................................................. 29

### II Included Publications

- A CNN Features off-the-shelf: an Astounding Baseline for Recognition A1
  1 Introduction ............................................. A2
  2 Background and Outline .................................. A5
CONTENTS

3 Visual Classification ............................................. A5
4 Visual Instance Retrieval ....................................... A12
5 Conclusion ......................................................... A14
References ............................................................. A15

B Persistent Evidence of Local Image Properties in Generic ConvNets B1
1 Introduction ......................................................... B2
2 Flow of information through a ConvNet ......................... B4
3 Experiment ......................................................... B5
4 Semantic directions in representation space ..................... B11
References ............................................................. B12

C Factors of Transferability for a Generic ConvNet Representation C1
1 Introduction ......................................................... C2
2 Range of target tasks examined ................................ C6
3 Experiments ......................................................... C8
4 Optimized Results ................................................ C21
5 Implementation details .......................................... C22
6 Closing Discussion ................................................ C22
7 Acknowledgement ................................................ C23
References ............................................................. C23

D Visual Instance Retrieval with Deep Convolutional Networks D1
1 Introduction ......................................................... D2
2 Related work ....................................................... D4
3 The ConvNet representation of an image ......................... D4
4 Pipeline for measuring similarities of images ................... D6
5 Results ............................................................... D10
6 Conclusion ......................................................... D13
References ............................................................. D13
Part I

Introduction
Chapter 1

Introduction

Visual recognition is a family of tasks in computer vision that try to model and infer the state of visual semantics in an image. To do so, models usually try to represent the input image in a way that relevant information is easily accessible to the inference module. There are numerous algorithms proposed to represent an image. Before 2012, the most successful representations were usually built upon well engineered and handcrafted modules describing the image. The performance of models based on these handcrafted descriptors was increasing steadily and slowly. But in 2012, with the advent of large manually annotated datasets and the increasing power of processors, a Convolutional Network (ConvNet) [60] won [58] the ImageNet challenge [85] and outperformed pipelines based on handcrafted descriptors by a huge margin (see figure 1).

ConvNets are a family of feed forward deep neural networks that are trained in a supervised and end-to-end fashion. Studies suggest that the magic associated with ConvNet is its ability to learn a good image representation from a large dataset [20, 10, 19, 71, 26, 17, 80].

This compilation thesis is comprised of four papers that are originally written to shed more light on the general questions of "What should be expected from ConvNet representation in visual recognition?" from an empirical perspective.

In paper A [80], we trained a ConvNet on a large dataset and used the ConvNet representation space to address a wide variety of visual recognition tasks. Our experiments solidify the superiority of ConvNet representations over other representations based on handcrafted descriptors. In paper B [79], we showed that the spatial information that persisted in the ConvNet representation space is reliable and easily accessible. In paper C [4], we studied the factors of transferability for a ConvNet representation. Finally, in paper D [81] we used the finding of the previous publications on the task of visual instance retrieval.

In all our experiments, we employed transfer learning. Transfer learning is a technique that aims to improve the performance of the task at hand (target task) by transferring the knowledge harvested from another task (source task). In all our experiments, we used large datasets (e.g., ImageNet [85]) as the source tasks and relatively small yet frequently used dataset as the target task.

The structure of this chapter is as follows: We define the visual recognition tasks that
CHAPTER 1. INTRODUCTION

Figure 1: Reported results for object detection on Pascal VOC07 [29] and ImageNet top-5 classification [85] benchmarks. ConvNet based models (circle marks) have significantly outperformed pipelines based on handcrafted features (square marks).

we tackled in section 1. Our pipeline is explained in section 2 and the factors of transferability that we studied in section 3. We summarize the discussion of this chapter in section 4.

The next chapters are organized as following: In chapter 2, we brief the details of some of the most famous handcrafted features and common encodings. Then we continue this chapter by mentioning the important components of ConvNets and finally try to make a connection between these two. Chapter 3 provides the summary of papers that this thesis is based upon. Finally in chapter 4 we conclude this thesis by providing a list of our contributions. The second part of this thesis contains full text of aforementioned publications.

1 Tasks

In our work, we studied the performance of ConvNet representations on a diverse set of visual recognition tasks. In some tasks, the model should infer about high-level visual semantics while in some others low-level visual semantics like spatial information must be modeled. Below is the list of tasks and their associated benchmarks:

Object Classification In this task, the model should infer about the presence or absence of a specific category of object in the image (For example, whether there is a cat in the image or not). Object classification is one of the core problems in computer vision. We
1. TASKS

used the highly cited benchmark of Pascal VOC 2007 dataset [29] as a benchmark for this task.

**Scene Classification** tries to predict the category of the scene in an image. An object classifier can be helpful in narrowing down the set of possible scenes. For example, detecting multiple books in an image implies that the scene is most likely a bookstore or a library. But to exactly recognize the scene, the model should also be aware of the structure of these objects in the scene. We used MIT67 Indoor scene dataset [78] and SUN 397 dataset [101] for our experiments on this task.

**Fine Grained Classification** is a narrow family of tasks in Object Classification where the objects of interest are visually similar. Unlike general object classification tasks in computer vision, in this family of tasks the outline of the object is far less informative and the model should be more aware of other families of features like color or texture (For example, the outlines of Sphinx and Cornish Rex are very similar but the texture of their skin is different). To evaluate this task, we considered the following benchmarks: Pet dataset [73], CUB-200 bird dataset [100] and Flower dataset [68].

**Attribute Detection** tries to describe the content of an image in terms of attributes as opposed to nouns. If an object detector is based on attributes, it could still function to some extent when encountering a new category of object. For example, a model may not have seen any cup before but it can still tell whether it is round or cylindrical. We used H3D human attributes dataset [13], Object attribute dataset [33] and SUN scene attribute dataset [74] for this task.

**Action Recognition** tries to model the *verb* in an image. For example, a man *cutting* vegetables is performing a different action than a man *cooking* them. In this task, objects, their spatiotemporal relation with each other and their pose are all important factors. For evaluation, we used Pascal VOC12 action dataset [31] and Stanford Action40 dataset [103].

**Visual Phrases** describe an image in terms of phrases like "A man riding a horse" by combining the words associated with visual concepts in the image. The same way that a phrase can be seen as an intermediate between a word and a sentence, a visual phrase task can also be viewed as the successor of the aforementioned tasks and the predecessor to visual caption generators. We used Visual Phrases dataset [86] to evaluate this task.

**Visual Instance Retrieval** aims to retrieve images of the same instance of an object in a reference set by sorting them according to their distance to the query image. This can be interpreted as embedding images into a representation space where visually similar images have similar representation. One of the challenges in this task is that the definition of

---

1 We excluded temporal information from our pipelines and only focused on the still-image action recognition tasks for the sake of unified framework.
similarity is vague. Similarity can be related to the presence of a particular object, pattern, shape or a scene structure. A major difference between visual instance retrieval and other visual recognition tasks is that retrieval tasks usually do not have a training set and sorting the reference set usually has to be done on the fly. For evaluation, we used five standard benchmarks in this field: Oxford building dataset [75], Paris Building dataset [76], Oxford Sculpture dataset [3], Holidays dataset [52] and UKBench dataset [69].

**Pose Estimation** aims to find the spatial properties of the item of interest in an image. A model in this task should be sensitive to low-level visual semantics like spatial position and orientation of an object of interest. Also, a model should be good enough to detect the object of interest in the first place, but the standard benchmarks in this task usually take the localized item of interest as an input. For this task, we used Pascal VOC11 keypoints[30], Helen [59], IBUG [8] and LFPW [87] datasets.

**Semantic Segmentation** combines spatial and class information all together and assigns labels to each pixel. In this task, the model should be sensitive to both high-level class information from low-level spatial information in an image. High-level visual semantics should be as accurate as possible while the model is still sensitive to low-level spatial information. For evaluation, we used VOC 2012 semantic segmentation dataset [31].

### 2 Pipeline

Our approach to solve these tasks is by employing transfer learning. Transfer learning tries to incorporate the knowledge learned from one domain (source task) into another one (target task). The idea of transfer learning combined with neural networks is not new (e.g., the work of Caruana (1998) [16]). But before 2012 in computer vision community, neural networks were usually perceived to have a tendency toward over-fitting and therefore, not considered for transfer learning. The recent findings of [80] suggest that ConvNets trained on large datasets learn a generic image representation and therefore, well suited for transfer learning in visual recognition. For all the categories of tasks that we studied, we stayed faithful to the simple pipeline:

- Train a ConvNet on a large dataset (source task).
- For every image in the task at hand (target task):
  - Feed each image to the pre-trained ConvNet.
  - Take the mid-level response of the network as the vector representation of the image.
  - Normalize the vector.
- Train a linear model (SVM [21] or regression [48]) over the representations (For the task of retrieval, sort reference images based on their distance to the query image in the representation space).
• Measure the performance according to the task’s criteria.

This simple pipeline proved to be powerful and the reported performance was on a par (if not better) with heavily engineered pipelines based on handcrafted features. To improve on these results, we simply augment our training set by a simple, yet effective technique. We extracted multiple patches from the image and based on the objective of the tasks, we aggregated the model’s response. For classification tasks, we average the vectors of the training and the test sets. For pose estimation tasks, we apply bounding box regression to have a higher resolution image and for retrieval, we compute the similarity between each patch in reference and query image.

It should be mentioned and highlighted that our simple off-the-shelf pipeline is by no means the most efficient pipeline. For example, while our simple pipeline resulted in the accuracy of 67.1% on CUB dataset [98], the work of [14] reported 85.4% using a more sophisticated ConvNet-based pipeline. The reasons we used this simple pipeline are two-fold. First, we wanted to show the effectiveness of ConvNet representation and second, we could build a unified framework to measure the importance of different ConvNet related factors on the performance.

3 Factors

The use of ConvNet representations has proven to be more effective than handcrafted features over a wide variety of tasks [80], yet different ConvNets yield different performances. In the next part, we identified and studied the factors affecting the transferability of ConvNet representation. We group these factors into two: Learning factors (see section 3.1) and post-learning factors (see section 3.2).

3.1 Learning Factors

Learning factors are the factors that are related to the training of the source task. We defined these factors as the following:

Network width Wider networks have more parameters in each layer while moderately wide networks have fewer. Our finding suggests that wider networks tend to be more specialized on the source task while moderately wide networks generalize better.

Network depth defines the number of layers that a network has. The deeper a network, the more linear transformations (followed by nonlinearity) will be applied to an image. We also observed that depth is a good regularizer and in general, deeper networks tend to generalize better.

Early stopping is a regularization technique that avoids over-fitting by stopping the training procedure early. We observe that early stopping only helps when the network on the source task also exhibits over-fitting (like the case of fine-tuning). Otherwise, it does not affect the performance of the target task.
CHAPTER 1. INTRODUCTION

<table>
<thead>
<tr>
<th>Factor</th>
<th>Source task</th>
<th>FineGrained recognition</th>
<th>Instance retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early stopping</td>
<td></td>
<td>Don’t do it</td>
<td></td>
</tr>
<tr>
<td>Network depth</td>
<td></td>
<td>As deep as possible</td>
<td></td>
</tr>
<tr>
<td>Network width</td>
<td>Wider</td>
<td>Moderately wide</td>
<td></td>
</tr>
<tr>
<td>Diversity/Density</td>
<td></td>
<td>More classes better than more images per class</td>
<td></td>
</tr>
<tr>
<td>Fine-tuning</td>
<td></td>
<td>Yes, more improvement with more labelled data</td>
<td></td>
</tr>
<tr>
<td>Dim. reduction</td>
<td>Original dim</td>
<td>Reduced dim</td>
<td></td>
</tr>
<tr>
<td>Rep. layer</td>
<td>Later layers</td>
<td>Earlier layers</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: The relation between factors of transferability and the performance of the model based on transferred representation.

Source task’s training data is probably the most important factor in learning a generic representation. We studied the training data from various points of view from the similarity of the objective in the source task to the target task, all the way to the density and diversity of the training data.

3.2 Post-learning Factors

Post-learning factors are the factors that should be considered when using a pre-trained ConvNet representation to target task.

Network Layer plays an important role on the performance of the target task. Early layers are more generic while later ones are more specific toward the source task.

Spatial Pooling is necessary to reduce the mid-layer representations’ size while preserving some spatial consistency. This pooling, in particular, is useful when the presence of spatial information is important (e.g., in retrieval tasks).

Dimensionality reduction We studied dimensionality reduction of the representation and observed that the representation for target tasks that are more distant to the source tasks can be more compressed. But in general, ConvNet representations can be compressed to an order of magnitude smaller footprint and still function as good.

Additional data We studied the effect of additional data from various points of view and our experiments suggest that additional data almost always helps.
In our analysis, we found a consistent correlation between the similarity of source/target tasks and the effect of those factors. This intuition can be useful for the use of ConvNet representations in other computer vision tasks. The summary of analysis is found in table 1.

4 Summary

Our studies over a diverse family of visual recognition tasks solidified that indeed, ConvNets trained on large datasets learn better-performing representations than the one based on handcrafted features descriptors. Also, ConvNets encode both high-level and low-level visual semantics in their representations and these semantics are disentangled and linearly accessible. We also provided insight on how to more effectively use ConvNet representation by defining factors that control the transferability of such representations. In chapter 4, we explain our contributions in more details.
Chapter 2

Background

In the book "Computer Vision" [77], Prince defines a vision problem as “take a visual data and use them to infer the state of the world”. He further describes a model as “a family of possible relationships between the data and the state of the world”. The pipeline of mapping between the input and the desired output or state of the world has gone through changes over time (See figure 1). In the rule-based era, no learning was involved in the pipeline but as the time went by, more and more machine learning was included in visual recognition pipelines. Machine learning components generally perform better than intuitive and hand-crafted components but they require more data and processing power. What all the pipelines have in common is that they repeatedly try to transform an image and represent it in a form that is easier or more intuitive to make an inference on. A classical visual recognition pipeline first tries to describe an image in terms of certain semantics that are believed to be informative like edges or colors. Then those descriptors are represented by aggregating them into a fixed length vector and the inference is made on those representations. A deep ConvNet does not make any explicit assumption on what information should be kept or discarded during each transformation. This chapter briefly describes some of the important components in visual recognition pipelines before the deep learning triumph in 2012 [58, 85] including some of the most commonly used hand-crafted features in section 1 and their associated encodings in section 2. Then we explain some of the important components of Convolutional Networks in section 3 and finally try to make a connection between the two in section 4.

1 Descriptors

In this section, we brief four descriptors (SIFT [65], HOG [23], GIST [70] and SCD [9]). These descriptors have been the most frequently used hand-crafted descriptors in the domain they are designed for. SIFT, HOG and SCD are histogram-based descriptors. Their performance are generally on par with other families of descriptors (e.g., Local Descriptor Learning [90]) while their simplicity and intuitiveness made them favorable to other descriptors. These descriptors aggregate the response of a certain filter over a large region
in the image by simply computing the histogram of the response. For cases where the response is continuous, some quantization has to be computed.

**SIFT** [65] or scale invariant feature descriptor and its variations like SURF [7], ORB [83] and BRIEF [15] are widely used in variety of computer vision task from classification to regression, retrieval and registration.

SIFT is the most commonly used hand-crafted feature descriptor. SIFT descriptor computes the histogram of image orientation for the $16 \times 16$ pixel areas around the interest points. The orientations (from the range $0^\circ$ to $360^\circ$) are then quantized into 8 bins. Finally, a histogram for every non-overlapping grid of $4 \times 4$ is computed. At the end, a $16 \times 16$ pixel region is described in a $4 \times 4$ grid of 8 dimensional histogram which can be viewed as a 128-D vector per region of interest. To find the interest points, SIFT computes the response of the image over $K$ difference of Gaussian kernels with increasing scale and picks the extrema in this $X \times Y \times K$ volume.

SIFT algorithm estimates these points by approximating a local quadratic function to the subvoxel around the extrema. This way, interest points will have a sub-pixel resolution. Although SIFT-based models have recently been outperformed by ConvNet-based pipelines for certain tasks like classification [80, 17, 71, 26, 106] and regression...
1. DESCRIPTORS

Figure 2: **Left:** HOG descriptor for a pedestrian image with overlapping grid of 2×2 as displayed in green and blue squares. HOG algorithm extracts gradient histogram from each cell as described in section 1. **Middle:** The visualization of HOG descriptors for the image. In each cell, the brightness of the segment lines correlates with the strength of the gradient in the same direction. The images on the left and middle are taken from [39]. **Right:** Visualization of SIFT descriptors. SIFT descriptors divide the area around the interest points into a grid of 4×4 and extract the histogram of orientation as described in section 1. SIFT visualization image is taken from [95].

[79, 27, 42, 63, 107, 105, 62], their sub-pixel resolution interest point gives them advantages on many other tasks e.g., registration [109, 65].

**HOG [23]** or **Histogram of Oriented Gradient**, is also a collection of normalized histogram and is generally viewed as a global descriptor. HOG computes the histogram of oriented gradient for small overlapping cells of 6×6 and returns a descriptor for each cell. These gradients are then quantized into 9 bins of 0° to 180°. The final descriptor is the product of concatenating all the cell descriptors. There are many extensions to HOG as well like **PHOG** [12], **HOG-3D** [57] and **Histogram of Sparse Coding** [82].

HOG descriptor in nature is similar to SIFT [65] but it has higher resolution and performs better localized normalizations. HOG is more suited for detection tasks where the spatial structure of object should be preserved (e.g., pedestrian detection).¹

**GIST [70]** is another gradient based global descriptor. GIST divides the image into grids of 4×4 and computes a 32D feature for each cell by convolving Gabor filters at 4 scales.

¹There are different implementations of SIFT and HOG with different parameters. We took the details from [77].
CHAPTER 2. BACKGROUND

and 8 orientations. These 16 vectors are then concatenated into a single 512 vector. This compact descriptor collects statistics about the gradients in each region of the image while preserving some spatial structure. This property is useful for categorizing different scenes.

**SCD [9]** or **Shape Context Descriptor** has been built upon the idea that for many classes, an object can be described with its silhouette. Shape Context descriptors, similar to SIFT and HOG, collect local information into a histogram but unlike those, the histogram is collected in a polar coordinate system as opposed to the Cartesian one. Also, instead of gradient, Shape Context models the distribution of the relative position of the interest point to other points that are sampled along the contours of the object.

## 2 Encoding Methods

Encoding methods try to embed the feature descriptors into a representation space by a non-linear transformation. In this chapter, we brief two families of encoding methods that have appeared frequently in computer vision. We briefly describe dictionary based methods and deformable part-based models.

**Dictionary-Based Encodings** In dictionary based encodings, a set of bases \( C \subset \mathbb{R}^d \) called codebooks or dictionary is learned (\( d \) is the dimension of the descriptors) so that each descriptor \( x \) can be approximated by these codebooks: 
\[
x \approx \sum_{c \in C} \alpha_c c.
\]
\( \alpha \) coefficients are usually non-negative and add up to 1. In histogram encoding [92] each descriptor is assigned to its nearest codebook and a histogram of those assignments will be collected for each image. The representation of a descriptor with its nearest codebook loses information. To address this problem, the size of dictionary is usually considered to be big.

**VLAD** [53] or Vector of Locally Aggregated Descriptors divides the descriptor space into Voronoi cells based on its codebook and tries to preserve the information in each Voronoi cell. To do so, VLAD aggregate the difference of assignments between descriptors and the assigned dictionary. **Fisher Vectors (FV)** [51] try to model both first and second order information about the distribution of descriptors in each cluster. Many different normalization and pooling methods have been proposed for these encodings. FV have proven to be the most successful of these family of encodings for large-scale image classification.

**Deformable Part Models** [38] are a family of the detector and have been built around the idea that an object can be recognized by detecting its parts and their spatial relations with each other. The work of Felzenszwalb et al. [37] is one of the most important class of algorithms in this family that uses HOG pyramid as an input. DPMs try to learn the most discriminative parts among different categories. In other word, it implicitly and simultaneously encodes features into parts and infers about the content of the image based on those parts.
3. CONVOLUTIONAL NETWORKS

Convolutional Network or ConvNet is an important class of supervised pattern recognition models. It has been developed in the field of Artificial Neural Network and after 2012, has reemerged as a crucial component in various computer vision tasks including classification [58, 91, 89, 47, 71, 26, 55, 106, 42], regression [79, 27, 42, 63, 107, 105, 62], retrieval [81, 80, 2, 6, 99, 104, 5, 94, 1] and multi-model alignments [102, 18, 54, 67].

ConvNet [60, 61] was developed in late 1980s in Bell labs to address the task of digit recognition. The architecture of the proposed model (LeNet) was influenced by the work of Neocognitron [41, 40], a hierarchical network for pattern recognition. Neocognitron itself was inspired by the psychological model of visual system by Nobel prize winners Hubel and Wiesel [49]. Neocognitron is composed of two types of neurons, Simple (S) neurons with small field of view, transform the input of the previous layer into a new representation space.

Figure 3: Left: Hierarchical network structure of the Neocognitron as displayed in Fukushima et al. (1983) [41]. S stands for simple layer and C for complex layer. Middle: LeNet-5 architecture as displayed in Lecun et al. (1998) [61]. LeNet substituted Simple layers with Convolution operation and Sub-sampling instead of Complex Layer. FC stands for fully connected layer. Right: AlexNet architecture as displayed in Krizhevsky et al. (2012) [58]. The structure of AlexNet is very similar to LeNet. Except AlexNet has more layers and trained with different regularizer on bigger and more training samples.
CHAPTER 2. BACKGROUND

Every S layer then is followed by a Complex (C) layer whose primary function is to "condense" the previous layers. The architecture of Neocognitron is displayed in figure 3.

LeNet, elevated the structure of Neocognitron with at least two major changes: LeNet replaced the Complex neurons with max pooling function. By doing so, it eliminated all the parameters between S and C layers. LeNet also imposed parameter sharing among neurons of a layer and reduced the number of parameters between C and S layers drastically (see figure 3).

ConvNets are mostly trained according to error back-propagation algorithm [84]. In back-propagation algorithm, the prediction of the model for the data batch is defined as \( y = f(x; \theta) \). In this definition, \( f \) is the model that maps input \( x \) to output \( y \) using parameters \( \theta \). Then, the Loss function \( L \) measures error of prediction \( (E) \) w.r.t. the ground truth \( \tau \) according to equation 2.1.

\[
E = L(\tau, y) = L(\tau, f(x; \theta)) \tag{2.1}
\]

The gradient descent algorithm updates the parameters of model iteratively based on equation 2.2. This algorithm repeatedly updates the \( t \) dimensional vector of \( \theta \) by adding a vector proportional to the gradient of the loss function \( \nabla_\theta L \) but in the opposite direction. Parameter \( \eta \) controls the size of this residual.

\[
\theta_{new} = \theta_{old} - \eta \nabla_\theta L(\tau, f(x; \theta)) \tag{2.2}
\]

A feed forward Neural Network can be written recursively as

\[
f(x; \theta) = f^L(f^{L-1}(\ldots f^1(x; \theta^1) \ldots ; \theta^1) \ldots ; \theta^{L-1}); \theta^L)\tag{2.3}
\]

where \( l \) is the index of a layer ranging from 1 to \( L \) and \( f^l \) is usually a linear function followed by a nonlinearity. Error Back-propagation algorithm applies chain rule to update the parameters according to equation 2.4.

\[
\nabla_\theta L = \frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial f^L} \frac{\partial f^L}{\partial f^{L-1}} \ldots \frac{\partial f^{l+1}}{\partial f^l} \frac{\partial f^l}{\partial \theta_l} \tag{2.4}
\]

As reflected in equation 2.4, \( \delta^l \) can be computed recursively according to equation 2.5.

\[
\delta^l = \delta^{l+1} \frac{\partial f^{l+1}}{\partial f^l} \tag{2.5}
\]

\[
\delta^L = \frac{\partial L}{\partial f^L} = \nabla_y L(\tau, y)
\]

It can be seen that the back-propagation algorithm is linear w.r.t. the size of dataset. In ConvNet model, \( f \) is defined as a convolution operation: \( f(x; \theta) = [x \ast \theta]_+ \) where \( [.]_+ \) is a nonlinear transformation usually in the form of \( [.]_+ = \max(., 0) \).
One advantage of the convolution operation is that it can operate on a big image size with few parameters. For example the first layer of AlexNet [58] transforms images from a 150k (=224×224×3) dimensional input space into a 145k (=50×50×48) dimensional hidden representation space with only 17k parameters that are highly correlated to one another. ConvNets benefit greatly from large and diverse datasets. This combination of massive data and small parameters regulates the model to a high degree.\footnote{The winner of 2014 ImageNet Competition GoogleNet [93] had roughly 7M parameters and was trained on 1M images of 256×256 pixels each. In other word, 1 parameter for every 9k input pixels.} Beside shared parameters and large datasets, ConvNets are usually further regularized by other techniques like ℓ2 regularizations [58], Dropout [58, 22] or batch normalizations [50].

4 Comparison

After the ImageNet competition in 2012 [58], Gradient-based Neural Networks in general and ConvNets in particular reemerged into computer vision pipelines. Collobert and Weston (2008) [20] and Bengio et al. (2011) [10] speculated that the magic of deep models are that their representations are generic. Ciregan et al.[19], Oquab et al. (2013) [71] and Donahue and Jia et al. (2013) [26] showed that ConvNets “pre-training on one different set greatly improves performance on quite different sets.” [88]. Razavian et al. (2014) [80] showed that Deep Convolutional Network trained on large datasets “can extract useful features from quite diverse off-training-set images, yielding better results than traditional, widely used features such as SIFT [65] on many vision tasks”[88].

There has been much speculation on why these models perform so well. First layer’s transformation seems to capture edges, textures and colors similar to hand-crafted features. The work of [97] showed that HOG features are noisy and they can map many different image patches to the same point in the HOG feature space. This is not the case for ConvNets where an image can be reconstructed almost perfectly from the early layers’ response map of a ConvNet [66]. In fact, the empirical results in [97] showed that given only HOG features, humans and DPMs perform almost as good.

Another interesting comparison is between DPMs and ConvNets [44]. In this work, Girshick et al. suggested that DPMs can be viewed as a special class of ConvNets with a certain generalization of the pooling layers. This implies that if HOG pyramids were not noisy, a DPM on HOG can be viewed as a shallow ConvNet with different learning procedure.

Dictionary based encodings usually learn their bases with an unsupervised density estimation method (e.g., K-means or GMM) while ConvNet in each transformation realigns the represented data manifold according to its objective.

It has been assumed [11, 45] that ConvNets simultaneously flatten and align the data manifold. This explanation is justified by many experimental results in the field. For example, deeper ConvNets with their piecewise linear manifold functions can provide better approximations to this process. Also in this explanations, the representation space should be smooth [11] or in other word, similar images should have similar representations [79, 81].
CHAPTER 2. BACKGROUND

Figure 4: The size of dataset annotations over time. It is interesting to see that Pietro Perona of Caltech and his student Fei-Fei Li (later professor in Stanford) were five years ahead of the rest of the vision community in grasping the importance of annotated data in computer vision and investing on it.

Closing discussion  A major drawback of ConvNets over well engineered models was that ConvNets require large datasets and immense processing power for training. The finding of [4] suggests that both density and diversity of datasets affect the quality of the representation space. While deep learning community had MNIST dataset [61], a massively annotated dataset of natural images was not around before 2009 (see figure 4). With the advancements in the processor industry, in 2012 deep learning reemerged in visual recognition. In 2013 and 2014, the works of [71, 26, 17, 80] showed that ConvNet representation outperforms baselines on small datasets with the help of a large dataset (ImageNet [25]). Those findings eliminated the last obstacle for ConvNet-based pipelines to take over the visual recognition tasks.
Chapter 3

Summary of papers

A CNN Features off-the-shelf: an Astounding Baseline for Recognition

Studies suggest that ConvNets are powerful tools to learn the representation [20, 10, 19, 71, 26, 17]. In this paper, we tried to put those findings in practice and showed that in fact, this is the case. In this work, we studied the performance of ConvNet representations on a set of fourteen different visual recognition tasks.

Our simple pipeline is as follows. First we trained a Convolutional Network (Overfeat [89]) on a large dataset (ImageNet [85]). Then for every image in a visual recognition task, we fed the image forward to the network and took the $\ell_2$ normalized vector of neural activities of a certain layer as the image representations. For every task (except for visual instance retrieval tasks) we trained a linear SVM [21] on the training set and reported the performance on the test set (see figure 1). For visual instance retrieval, we measured the performance based on the $\ell_2$ distance of representation vectors.

This simple pipeline proved to be effective and competitive to highly-tuned pipelines.

Figure 1: Comparison of our pipeline (top) vs an example of a highly tuned and well-engineered pipeline (bottom) for the task of fine-grained classification. Our simple pipeline proved to be effective and competitive to every other pipeline on a diverse set of tasks.
CHAPTER 3. SUMMARY OF PAPERS

Figure 2: Augmented ConvNet representation constantly outperforms s.o.a. on a variety of tasks. (Specialized CNN refers to other forms of pooling ConvNet features)

based on hand-crafted features. We later on enhanced our pipeline by data augmentation. By extracting multiple patches from a single image and compute the representation for those patches we drastically improved the performance of our model to a point that ConvNet representations outperformed almost every other pipelines (see figure 2.).
B. PERSISTENT EVIDENCE OF LOCAL IMAGE PROPERTIES IN GENERIC
CONVNETS

B Persistent Evidence of Local Image Properties in Generic
ConvNets

Although a generic ConvNet has never been explicitly trained to preserve spatial semantics, previous studies suggest that generic ConvNet representation trained for the task of classification is still sensitive to spatial semantics (i.e., the position of the object of interest in the image) [72]. In this work, we tried to measure the reliability of spatial semantics in a generic ConvNet representation.

To do so, we evaluated the ConvNet representation performance on four families of tasks that require local spatial semantics. We again chose our simplistic pipeline (i.e., one representation per image and a linear mapping from representation to output) to tackle these tasks. The families of tasks that we studied are 1) 2D landmark estimation, 2) 2D keypoint prediction, 3) RGB reconstruction and 4) Semantic Segmentation.

The qualitative and quantitative results on the aforementioned tasks suggest that ConvNet representation retain significant spatial semantics (see figure 3 for an example on task 1). In order to enhance the quality of our estimations, we employed the simple technique of data augmentation (extracting multiple patches per image instead of one) combined with a bounding box regression.

Our studies confirm that both low-level spatial semantics and high-level class semantics co-exist in ConvNet representation and they are accessible almost independently. We further expanded our experiments and tried to move image representations in the representation space along the directions that have meaningful interpretations for us.

![Figure 3: Left: ground truth landmarks. Middle: estimated landmarks from a single image representation. The prediction is reliable enough to apply bounding box regression for further enhancements. Right: fine-tuned landmarks after bounding box regression. Our simple augmentation technique improves to the performance of landmarks to a point where it is comparable to the highly tuned baselines based on hand-crafted features.](image-url)
To be precise, given an image, we extracted the image representation first, then moved the representation along the direction that correlates the most with "gender" or "pose" and visualize the modified representation. Our observation suggests that changing a high-level representation according to a high-level semantic can be translated to the pixel-level change that carries the same meaning to human subjects. Our findings can be viewed as evidence for a theory that ConvNets may disentangle different underlying factors that generate the image.

Figure 4: **Center column**: We extract the ConvNet representation of an image (A face in this example) and reconstruct the image from the representation. We can also move the representation vector in the direction of high-level semantics and revisualizes it (Other columns). By moving the image representation along the gender direction, we see that the corresponding visualization changes accordingly. **Top row**: Masculinity in ConvNet representation space correlates with facial hair, triangular faces, and thick eyebrows. Femininity is visualized with rounder faces, more makeups, and thinner eyebrows. **Bottom row**: Visualization of the in-plane rotation of the face. Modifying the image according to spatial semantics also corresponds to the expected changes in the reconstructed image.
C. FACTORS OF TRANSFERABILITY FOR A GENERIC CONVNET REPRESENTATION

C Factors of Transferability for a Generic ConvNet Representation

The use of Artificial Neural Networks (ANN) in general for transfer learning is not new. But before 2012, Neural Networks in visual recognition were generally considered to be prone to over-fitting and therefore, few works had been done to the study of factors that affect the transferability in visual recognition.

Recent studies suggest that ConvNet (a special family of Neural Networks) based representation should be considered as the primary choice for visual recognition [80]. Yet, the performance of these models depends heavily on the factors that upon them, the ConvNet is trained or used.

In this work, we follow the pipeline proposed in [80] where the ConvNet is trained on a source task, then the activity of a certain layer of the ConvNet will be treated as the vector representation of the image in the target task. This work tries to corner out several factors that affect this transferability of the representation space from the source task to the target task. We divided the factors into two groups: 1) Learning factors involving the training of the ConvNet on the source task including: ConvNet architecture (depth and width), Source data and early stopping. 2) Post-learning factors that exploit source ConvNet for the target task including: The choice of layer, spatial pooling, dimensionality reduction and the use of extra data in different ways on both source and target task (see figure 5).

![Diagram of factors affecting transferability](image)

Figure 5: Factors of transferability for a generic ConvNet from the source task to the target task. **Top:** learning factors, **Bottom:** post-learning factors.
In our study, we observed a consistent pattern in the relation between the effect of these factors on the performance of the target tasks and the similarity between the source and target task. We performed our experiments over a diverse set of visual recognition tasks, datasets, network architectures and models and showed that just by choosing the optimal settings on these factors, the relative error on the target task can be improved by up to 50% (see figure 6).
D. VISUAL INSTANCE RETRIEVAL WITH DEEP CONVOLUTIONAL NETWORKS

Visual Instance Retrieval with Deep Convolutional Networks

In this paper, we tackled the task of visual instance retrieval specifically. In this task, a reference set is provided and the model should be able to sort this reference set based on their relevance to a query image. The relevancy is usually defined as the presence of a particular visual semantic of the query image in the reference image set (e.g., the presence of a particular object or scene or shape of interest). In another word, the model should learn an embedding space where the representation of the relevant images ends up close to one another. A major challenge in this task is that the item of interest can appear in different viewpoints, lightings, scales etc.

In this work, using a multi-scale scheme, we proposed a pipeline that tries to preserve certain spatial consistency while being less variant toward the change in position and scale of the item of interest. We used five datasets to test our pipeline and showed that “generic ConvNet image representations can outperform other state-of-the-art methods if they are extracted appropriately.”

This work, as well as the parallel work of Babenko et al. [6, 5], were among the first to use ConvNet representations for the task of retrieval.

Figure 7: The items of interest in visual instance retrieval can vary in many aspects, namely: viewpoints, lightings, scales and locations. A good model should be able to handle those variations.
Chapter 4

Conclusion

Over the last years, the part of computer vision community focused on recognition have shifted their focus from hand-crafted features to the deep ConvNet representations. The publications in this thesis were written to address some of the important questions in this path. We addressed series of tasks, but the most significant contribution of this thesis is that we showed a deep ConvNet trained on ImageNet learns a generic image representation that can be transferred to a wide variety of visual recognition domains. Based on this, we concluded that:

“From now on, deep learning with Convolutional Networks has to be considered as the primary candidate in essentially any visual recognition task.”

In more details, the contributions of this thesis are listed below:

• In paper A, we systematically evaluated the performance of ConvNet representations on a wide variety of visual recognition tasks and showed that simple pipelines with ConvNet representations can outperform highly-tuned baselines with hand-crafted features. Our diverse experiments solidify and justify the superiority of ConvNet representations over hand-crafted ones for visual recognition tasks.

• In paper B, we measured the spatial information in generic ConvNet representations. It was shown that ConvNet representation trained for classifications preserves some spatial information [72, 64]. We evaluated the quality of this low-level visual semantics and observed that ConvNets retain rich spatial semantics that are well disentangled from the high-level ones.

• In paper C, we studied the factors involved in training and using ConvNets that affect the transferability of ConvNet representations. By defining and studying a diverse set of factors over a large number of target tasks, this work provides intuitions on how one can take advantage of these factors to optimize the performance of a visual recognition model.
• In paper D, we focused on the task of visual instance retrieval. We provided a single pipeline that outperforms every other pipeline based on hand-crafted features on all the datasets. Our work was among the first to successfully use the ConvNet representation for the task of instance retrieval.

• Last but not least, our vision of a unified pipeline based on an off-the-shelf representation that outperforms a wide variety of baselines over different datasets and objective has permeated the computer vision field (e.g., [56] in natural language processing).

Future work. Computer vision will probably remain a challenging problem in the coming years. Although ConvNets are sitting unchallenged at the core of visual recognition models, there is a large body of ongoing work trying to improve the performance of ConvNets which only means they are not perfect yet. Despite all the great works that have been done recently in embedding structures into ConvNets, still it is safe to say that a ConvNet is only as good as the dataset it is trained on and many semantics are hard to annotate. Some of this semantics are low-level ones that are simply left behind. For example, to the best of our knowledge, there has not been a large dataset that addresses the problem of specular/transparent surfaces yet. Or, some other semantics are too abstract to annotate. A visual model may be able to reliably detect most of the objects in an image but we probably have to wait for a model that can tell us why “Starry Night” of Van Gogh feels so gloomy yet soothing. There are efforts in computer vision community to address these challenges by creating synthetic datasets or harvesting abstract semantics from other modes of data (texts for example). But to see how well those efforts would pay off, we just have to wait.
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


