Visual Tasks beyond Categorization for Training Convolutional Neural Networks

by

Hyodong Lee

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY June 2016

© Massachusetts Institute of Technology 2016. All rights reserved.

Signature redacted

Author .................................................................

Department of Electrical Engineering and Computer Science

May 20, 2016

Signature redacted

Certified by ..............................................................

James J. DiCarlo
Professor of Neuroscience; Head, Department of Brain and Cognitive Sciences
Thesis Supervisor

Signature redacted

Accepted by ............................................................

Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses
Visual Tasks beyond Categorization for Training
Convolutional Neural Networks

by

Hyodong Lee

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Humans can perceive a variety of visual properties of objects besides their category. In this paper, we explore whether convolutional neural networks (CNNs) can also learn object-related variables. The models are trained for object position, size and pose, respectively, from synthetic images and tested on unseen held-out objects. First, we show that some object properties come “for free” from learning others, and pose-optimized model can generalize to both categorical and non-categorical variables. Second, we demonstrate that pre-training the model with pose facilitates learning object categories from both synthetic and realistic images.

Thesis Supervisor: James J. DiCarlo
Title: Professor of Neuroscience; Head, Department of Brain and Cognitive Sciences
Acknowledgments

I would like to start with deepest gratitude to my advisor, Professor James J. DiCarlo, for all of his support. His invaluable advice has always guided me towards right directions and his encouragement has helped me to dive into the challenging problems. I am also indebted to Daniel L. K. Yamins for all of his assistance and intuition in this work. His technical advice and novel suggestions have enriched my work in numerous ways. Without their guidance, I would not have been able to conduct this research.

I appreciate all the group members of DiCarlo lab, especially Darren Seibert and Rishi Rajalingham, for the useful conversations. I would like to express special thanks to Ha Hong, who has provided both technical and mental support since I joined the group.

Additionally, I appreciate Professor Jose Martinez, my undergraduate advisor at Cornell, and also Saugata Ghose and Kai-Yuan Chen for encouraging me to pursue a research career in graduate school.

I owe thanks to Samsung Scholarship for financial support for my graduate study.

To my beloved mom and dad, I am more than grateful for your infinite love and absolute belief in my capability. For my little brother, I'd like to thank for your invisible yet strong mental support.
Contents

1 Introduction .................................................. 7
  1.1 Motivation ................................................. 7
  1.2 Contributions ............................................. 9
  1.3 Thesis Outline ............................................ 9

2 Methods ....................................................... 10
  2.1 Dataset .................................................... 10
  2.2 Pose Training ............................................. 11
  2.3 Model Architecture ....................................... 12
  2.4 Training details ........................................... 13

3 Results ......................................................... 15
  3.1 Object variable generalization tasks ....................... 15
  3.2 Transfer learning to category ................................ 16
      3.2.1 Learning category from synthetic dataset ................ 17
      3.2.2 Learning category from ImageNet dataset ................ 17

4 Discussion ................................................... 19
List of Figures

2-1 Sample images of the training dataset. .............................................. 11

2-2 Model architecture for category, position and size (blue-boxed) and pose. For pose training, the blue-boxed region represents a single column of the duplicates. ................................................................. 13

3-1 Performance of object variable-optimized models on held-out test images. Each row corresponds to the model optimized for each variable: category, position, size, or pose. The first row represents a model whose filter weights are randomly initialized. Each column shows the performance on the corresponding tasks as absolute values (A) or column-wise normalized values (B). ................................................................. 16

3-2 Transfer learning to category of unseen objects from synthetic dataset (A and B) and ImageNet database (C). The y-axis represents softmax error on training images (A and C) or top-1 categorization accuracy on test images (B). The x-axis corresponds to each training epoch. . . . 17
List of Tables

2.1 Architectural parameters of the models. .......................... 12
Chapter 1

Introduction

In this thesis, we explore the performance of convolutional neural networks on different object-related visual tasks. For each model, we investigate whether it can predict object-related properties that the model was not trained for. In this chapter, we introduce the motivation for this research, describe the main contributions, and provide an overview of the thesis’ organization.

1.1 Motivation

The primate visual system allows animals to rapidly assimilate visual information from retinal images. Past studies have shown that IT cortex, at the top of the hierarchy of the visual ventral stream, is responsible for object identification and categorization [6, 14, 18]. Yet the perception of objects in a scene requires an understanding of a variety of object-related properties, not limited to its category. Recently, Hong et al. [5] found that information for category-orthogonal object properties, such as size and position, increases along the ventral stream as well. This finding suggests that a complete model of the ventral stream should have an ability to estimate not only categorical but also non-categorical properties of the objects.

Moreover, in real world, certain non-categorical properties, such as position, pose, and surface normal, are accessible almost freely. Since these variables could be internally estimated from self-motion, they can be supplied almost unlimitedly and
learning them does not require external supervision. Given the abundance of such mid-level features, it is more biologically plausible and probably more efficient to develop visual perception by utilizing those properties as opposed to training only with labels.

Neural networks are biologically inspired models of visual systems, well known for efficiently solving object categorization problems. Stochastic gradient descent (SGD) allows efficient training of deep convolutional neural networks (CNNs) via back-propagation [13]. Recent architectural advances, such as [19] and [20], have significantly improved categorization performance. Combined with large-scale image sources like ImageNet [1] and efficient implementation of mini-batch normalization [8], the recently proposed deep CNNs have achieved high categorization accuracy with reduced training time.

Recent studies have found that such category-trained CNNs have a solid generic feature representation that can solve different visual recognition tasks [15, 2, 16], semantic problems [4, 10] and object detection [4]. Girshick et al. [4] showed that supervised pre-training for an auxiliary task, followed by fine-tuning on the target, can significantly improve performance. Until recently, it has not been well studied whether the generic CNN features can estimate mid-level object properties, mainly because those variables, such as position and pose, were accepted as invariant to the object category. Yet, in addition to [4], Hong et al.[5] has shown that the category-trained CNN features can generalize to category-orthogonal properties, such as object scale and 3D rotation. These results suggest that the visual tasks are indeed interrelated such that a common feature representation may support the different tasks.

Observing that category-driven features can solve non-categorical problems, we tries to see if the opposite is true. If learning object-related properties is interconnected, training for those variables can also drive a feature representation useful for estimating catgorical and non-categorical variables. In particular, it will be interesting to see if we can find a non-categorical variable that gives as powerful features as the category does when it is trained for. Many recent studies have trained CNNs for estimating object position [20, 4] and human pose [21, 22], but their generalizability
tests are limited to categorical or semantic problems.

In this work, we trained a deep CNN for non-categorical object-related properties (position, three-dimensional scale (referred to as size), and 3D rotation (referred to as pose)) and investigate whether the learned features can generalize to object property estimation tasks. Unlike previous studies [20, 4, 21, 22], each variable is learned through direct regression on the values. Further, we investigate whether learning those properties drives generic features that solve categorization task via transfer learning.

1.2 Contributions

This thesis investigates three main problems. First, we demonstrate that neural networks can learn object-related visual tasks besides categorization. Second, we investigate which object properties can be learned “for free” from training for each variable. The models are optimized for object category, position, size and pose, respectively, and their performance on estimating those variables is analyzed. Lastly, we finetune a model on categorization tasks by taking the learned weights of the models pre-trained for size, position, and pose. We show that pre-training with pose facilitates learning object category.

1.3 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 explains how our dataset and models are constructed and provides training methodology in details. In Chapter 3, we present the test results on which variable comes “for free” and on whether pre-training with non-categorical variables expedites learning categories. Chapter 4 summarizes the thesis and suggests future research.
Chapter 2

Methods

2.1 Dataset

For latent variable training, it is important that the object variables are well defined in the dataset. Yet, it is not trivial to retrieve object properties from natural images of 3D objects. To avoid this problem, we generated a new dataset from 3D synthetic objects and backgrounds by defining the object position, size and pose in each image at the time of generation. This also allows an even distribution of each variable in the 3D space and thus prevents the trained models from becoming distorted spatial representations.

Our dataset consists of about 6.3 million training images, 632,832 validation images and 5,760 test images (see Figure 2-1). The training and validation images are drawn from 68 categories (1,049 objects). The test images are constructed from 64 held-out objects. The background and object geometric transformation parameters are randomly chosen from a given pool. The object position is represented by x- and y-coordinates of the image, where each variable is in the range of (-1, 1) and (0, 0) corresponds to the image center. The object size is represented as a three-dimensional scale factor. The object pose is defined as the three-dimensional rotation of an object (details are described in Section 2.2).
2.2 Pose Training

One needs to be careful about the representation of 3D object rotation. One simple way to define object pose is to determine a "front pose" or "canonical pose", which defines zero rotation in all planes, and to compute the pose difference with respect to it. Yet, the limitation of this approach is that such front poses vary by objects. During the training period, the model will be given the rotated images and rotation angles without knowing the front pose of the objects. Then, the model needs to figure out the definition of zero rotation and the correct value of rotated angles at the same time. This can be quite a challenging task, especially as the front pose is defined with object-level variations.

To avoid this problem, we set the learning objective as a pose difference between two images containing the same object. The input to the model is a pair of images. The first images come from the same dataset described in Section 2.1, and the paired images contain the same object with a different pose. We limited object rotation in the second image, so that it can represent the front pose of the corresponding object. This mimics the way humans perceive object rotation, where object pose is computed with respect to the perceived front pose. In this way, the model does not need to explicitly know the canonical angles of each object.

We use a four-dimensional unit quaternion vector to represent each rotation.
Quaternion representation does not suffer from the problem of ambiguous representation, while different combinations of Euler angles represent the same rotation under different orders of rotation. To compute the error between the predicted and true quaternion vectors, we use the distance function, shown in Equation 2.1:

\[
dist(q, \hat{q}) = \cos^{-1}(|q \cdot \hat{q}|)
\]  

(2.1)

This quaternion loss function is preferable to the Euclidean distance, which does not truly reflect the distance of two rotations, as described in [7].

### 2.3 Model Architecture

Our model is based on the Alexnet [12] architecture with slight modifications. The model consists of five ReLU convolutional layers, three of which are followed by response-normalization layers and/or max-pooling layers. Two fully connected layers and dropout layers follow after the last pooling layer. The main difference from the original model is an increase in the number of filters at intermediate layers. Details are shown in Table 2.1.

<table>
<thead>
<tr>
<th>type</th>
<th>filter size</th>
<th># filters</th>
<th>stride</th>
<th>output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>convolution</td>
<td>7</td>
<td>96</td>
<td>4</td>
<td>56x56x96</td>
</tr>
<tr>
<td>local response normalization</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max pool</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>28x28x96</td>
</tr>
<tr>
<td>convolution</td>
<td>5</td>
<td>256</td>
<td>1</td>
<td>28x28x256</td>
</tr>
<tr>
<td>local response normalization</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max pool</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>14x14x256</td>
</tr>
<tr>
<td>convolution</td>
<td>3</td>
<td>512</td>
<td>1</td>
<td>14x14x512</td>
</tr>
<tr>
<td>convolution</td>
<td>3</td>
<td>1024</td>
<td>1</td>
<td>14x14x1024</td>
</tr>
<tr>
<td>convolution</td>
<td>3</td>
<td>512</td>
<td>1</td>
<td>14x14x512</td>
</tr>
<tr>
<td>max pool</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>7x7x512</td>
</tr>
<tr>
<td>fully connected</td>
<td></td>
<td></td>
<td></td>
<td>1x1x4096</td>
</tr>
<tr>
<td>fully connected</td>
<td></td>
<td></td>
<td></td>
<td>1x1x4096</td>
</tr>
</tbody>
</table>

Table 2.1: Architectural parameters of the models.

Four models were optimized for object category, position, size and pose, respectively. At the top of each model, a fully connected layer predicts the object variable,
and its output is fed into a loss layer to compute the error. For category training, a 68-way softmax classifier was used to compute the loss. For other variables, the regression loss was computed as the Euclidean distance between predicted and actual values of length one (size) or two (position).

For pose optimization, the model architecture is modified to have two duplicate columns of layers, each of which takes one image from the pairs (see Figure 2-2). Each convolutional or fully connected layer in one column shares the filter weights with the duplicate in the other column. This allows the two columns to represent one single model, while taking different images. A fully connected layer takes the difference of the two columns’ outputs and is followed by another fully connected layer to predict the pose difference as a four-dimensional quaternion vector. The regression loss was computed as a quaternion distance, as described in Section 2.2.

Figure 2-2: Model architecture for category, position and size (blue-boxed) and pose. For pose training, the blue-boxed region represents a single column of the duplicates.

2.4 Training details

Our network implementation is derived from the Alexnet model [12], which is implemented in the Caffe framework [9]. A single NVIDIA Tesla K20m GPU was used for training each of the models for category, position, and size, respectively. Due
to the GPU memory limit, the model for pose was trained on two GPUs via data parallelism, as described in [11].

The input images were randomly cropped from 256x256 into 224x224 patches. This data augmentation scheme reduces overfitting to the training images. During the test phase, the input images were center-cropped. Horizontal flipping was not used, as it affects the position and pose of the objects in the images.

We trained the models using stochastic gradient descent with a batch size of 128 images, a momentum of 0.9, and a weight decay of 0.0005. The initial weights were filled randomly from the zero-mean Gaussian distribution for all convolutional and fully connected layers. The biases were initialized as 0 for the first and third convolutional layers and 1 for others. The learning rates of weights and biases were initialized as 0.001 and 0.002 for all layers. Throughout the training, we decreased the learning rate by a factor of 10 if the validation error stops decreasing. The entire training period was 12 epochs.
Chapter 3

Results

3.1 Object variable generalization tasks

In this section, we determine whether the features of the model trained on one variable are generalizable to other object properties. Using the same methods described in [5], we computed the predicted outputs of the intermediate layers of the trained networks in response to the test images, and tested simple, cross-validated linear classifiers (support vector machine for class labels and Ridge regression for other labels) based on these features. The test set consists of 64 unseen objects from 8 categories, and the tasks are to predict category, horizontal or vertical position, three-dimensional scale, or the x, y, or z-axis rotation of the given object. We define the categorization performance as the features' discriminability for different categories. For other properties, the task is to estimate the value of each variable, and performance is defined as the Pearson correlation of the features' response with the actual property value. Figure 3-1 shows the results.

All models except the size-optimized one performed best predicting the variables it was trained for (blue boxes in Figure 3-1B). Compared to size and position, the pose-optimized model (red boxes in Figure 3-1) performed well in predicting all other variables, yielding well above the performance of the randomly initialized filters. This suggests that training for object pose accompanies learning other object information better than position or size.
Figure 3-1: Performance of object variable-optimized models on held-out test images. Each row corresponds to the model optimized for each variable: category, position, size, or pose. The first row represents a model whose filter weights are randomly initialized. Each column shows the performance on the corresponding tasks as absolute values (A) or column-wise normalized values (B).

In Figure 3-1A, we observe that the absolute performance of category- and pose-optimized models on position and size estimation is similar to that of the directly trained models. This indicates that position and size information come for free from learning category or pose.

It is notable that the categorization performance of the pose-optimized model was sufficiently high, yielding 75.8% of the category-optimized model. This is evidence that the representation of category information naturally emerges from learning the lower level object variable. The categorization performance of the position and size-optimized models was around 50% of direct training. This suggests that position or size optimization also helps the model to develop the generic features relevant to object category, although not as much as the pose-optimized model does. Overall, the results show that learning the object property is not just about learning how to predict the target value, but rather a process of grasping the generic visual representations with regard to the objects.

3.2 Transfer learning to category

In this section, we investigate whether explicit learning of object position, size or pose expedites training for categories of unseen objects. Models were initialized with
the learned filter weights of the position-, size-, or pose-optimized model, and further fine-tuned for the categorization task. We also trained a model with same architecture from scratch to compare with the fine-tuned models. The results are shown in Figure 3-2.

![Figure 3-2: Transfer learning to category of unseen objects from synthetic dataset (A and B) and ImageNet database (C). The y-axis represents softmax error on training images (A and C) or top-1 categorization accuracy on test images (B). The x-axis corresponds to each training epoch.](image)

### 3.2.1 Learning category from synthetic dataset

Figure 3-2A and B show the training and testing results of category transfer learning from our synthetic test dataset. We observed that pre-training with pose (green lines) allows the model to converge faster than training from scratch (blue lines). This indicates that pose optimization drives generic features that can be used to learn categories of unseen objects by direct training. Pre-training with size (orange lines) or position (purple lines) did not provide any gain to category training. This matches with our generalization test results, where either size- or position-optimized model did not perform well on categorization task.

### 3.2.2 Learning category from ImageNet dataset

ImageNet is a dataset built upon the hierarchical structure of WordNet [3], containing over 15 million labeled full resolution images in approximately 22,000 categories. It contains enough images to well represent natural images in the real world, and thus
widely used for training and testing CNNs on object recognition tasks [12]. Our models were tested on a subset of the ILSVRC-2012 dataset, containing 999 object classes, which can be downloaded from the webpage [17]. The softmax loss layer of the model was modified from 68-way to 999-way to cover all object classes in ImageNet dataset.

As shown in Figure 3-2C, the model learns category from ImageNet images faster when pre-trained for pose (green line) than trained from scratch (blue line). This demonstrates that the features learned by training synthetic object pose is generalizable not only onto different variable tasks but also onto different image statistics, even more naturalistic images like ImageNet.
Chapter 4

Discussion

In this work, we found that deep CNNs trained for object-related properties can generalize to different variables the models were not trained for. Along with recent findings of [4] and [5], this result shows that learning different visual properties are interconnected. This suggests that training for multiple variables simultaneously may not significantly degrade the performance as opposed to single variable training. Based on our findings, it is probable that such multi-task training will reduce the number of training images and overall training time, while yielding better interleaved features for multiple visual tasks. Yet, this is an open question and needs to be verified by further investigation.

We observed that the representational power of the models depends on which latent variable is optimized for, i.e., the pose- or category-optimized models generalize better than position-optimized one. This suggests that certain object properties are more essential than others. Thus, it is important to choose the target properties carefully in order to build a powerful representation that can solve a broad range of visual tasks.

In pursuing this, future studies can explore different task regime that can build a rich representation of generic, semantic and 3D visual features as a whole. A good learning goal should be, first, easily accessible from every-day objects and scenes, and second, interrelated to the learning of different object-related properties that humans can predict well. Good candidates in the object-related property regime are surface
normal and depth cues, which may better encode shape information of an object as well as geometric properties than pose. Considering the every-day accessibility, self-orientation relative to objects is another good objective since it emerges naturally by definition. It is also more biologically plausible in the sense that humans can supply the target variables by themselves and thus easily estimate and correct a large amount of predictions in a very short period, as opposed to instance-specific properties like the category.

The limit of this work is that our dataset lacks realistic visual cues, such as lighting and 3D depth. Enriching the training set to represent more natural 3D scenes may result in richer visual representation. It will be also helpful to incorporate more complex scenarios, such as multi-object scenes with occlusion.

Lastly, our experiments demonstrated that pre-training with non-categorical visual tasks on synthetic dataset can improve learning categories of unseen objects in both synthetic and real-life images. Such fine-tuning can be particularly useful for category learning with limited resources, i.e. fine-grained classification with very limited labels.

This becomes more practical when a model can learn such properties in semi-supervised way, i.e. future pose prediction of a moving object. It will be interesting to see whether the learned features from semi-supervised tasks can solve different visual tasks better than supervised training.
Bibliography


