iLab-20M: A large-scale controlled object dataset to investigate deep learning

Ali Borji\textsuperscript{1} Saeed Izadi\textsuperscript{2} Laurent Itti\textsuperscript{3}

\textsuperscript{1}Center for Research in Computer Vision, University of Central Florida
\textsuperscript{2}Amirkabir University of Technology, \textsuperscript{3}University of Southern California
aborji@crcv.ucf.edu, sizadi@aut.ac.ir, itti@usc.edu

Abstract

Tolerance to image variations (e.g., translation, scale, pose, illumination, background) is an important desired property of any object recognition system, be it human or machine. Moving towards increasingly bigger datasets has been trending in computer vision especially with the emergence of highly popular deep learning models. While being very useful for learning invariance to object inter- and intra-class shape variability, these large-scale wild datasets are not very useful for learning invariance to other parameters urging researchers to resort to other tricks for training models. In this work, we introduce a large-scale synthetic dataset, which is freely and publicly available, and use it to answer several fundamental questions regarding selectivity and invariance properties of convolutional neural networks. Our dataset contains two parts: a) objects shot on a turntable: 15 categories, 8 rotation angles, 11 cameras on a semi-circular arch, 5 lighting conditions, 3 focus levels, variety of backgrounds (23.4 per instance) generating 1320 images per instance (about 22 million images in total), and b) scenes: in which a robotic arm takes pictures of objects on a 1:160 scale scene. We study: 1) invariance and selectivity of different CNN layers, 2) knowledge transfer from one object category to another, 3) systematic or random sampling of images to build a train set, 4) domain adaptation from synthetic to natural scenes, and 5) order of knowledge delivery to CNNs. We also discuss how our analyses can lead the field to develop more efficient deep learning methods.

1. Introduction

Object and scene recognition is arguably the most important problem in computer vision and while humans do it quickly and almost effortlessly, machines still lag behind humans. In some cases, where variability is relatively low (e.g., frontal face recognition) machines outperform humans but they do not perform quite as well when variety is high. Hence, the crux of the object recognition problem is tolerance to intra- and inter-class variability, lighting, scale, in-plane and in-depth rotation, background clutter, etc [9]. Thanks to big data and deep neural networks, computer vision has recently enjoyed a rapid progress, witnessed by high accuracies over the ImageNet dataset (top-5 error rate between 3-10% over 1,000 object categories). Recent models (e.g., Alexnet [31], VGG [54], Overfeat [50], GoogLeNet [57], and ResNet [23]) have surpassed previous scores in several benchmarks such as generic object and scene recognition [31, 54], object detection [50, 20], semantic scene segmentation [6, 29], face detection and recognition [60], texture recognition [7], fine-grained recognition [39], multi-view 3D shape recognition [56], activity recognition [33, 28], and saliency prediction [32].

One chief concern regarding the wild large-scale benchmarks and datasets, however, is the lack of control over data collection procedures and deep comprehension of stimulus variety. While existing large-scale datasets are very rich in terms of inter- and intra-class variability, they fail to probe the ability of a model to solve the general invariance problem. In order words, natural image datasets (e.g., ImageNet [8], SUN [64], PASCAL VOC [14], LabelMe [48], Tiny [61], and MS COCO [38]) are inherently biased in the sense that they do not offer all object variations [60]. To remedy this, some works (e.g., [45, 35, 41]) have resorted to synthetic datasets where several object parameters exist.

Ideally, we would like models to be tolerant to identity-preserving image variations (e.g., variation in position, scale, pose, illumination, occlusion). To probe this, some researchers have used synthetic home-brewed datasets either by taking pictures of objects on a turntable (e.g., NORB [35], COIL [41], SOIL-47 [29], ALOI [19], GRAZ [42], BigBIRD [55]) or by constructing 3D graphic models and rendering textures to them (e.g., Pinto et al.[45], Peng et al.[43]). While proven to be beneficial in the past, these datasets are very small for training deep neural networks with millions of parameters. Further, they usually have small number of classes, instances per class, background variability, in plane and in-depth rotation, illuminations, scale, and total number of images. Here, to remedy these shortcomings, we introduce a large-scale controlled object dataset with rich variety and a larger set of images.
2. Related work

Several controlled datasets have been introduced in the past which have dramatically helped progress in computer vision (Table 1). Two famous examples are FERET face [44] and MNIST digit [34] datasets. Nowadays, we have face and digit recognition systems that perform either at the level of humans (e.g., [58]) or superior (perhaps not as robust due to variations and noise). Similar datasets are available for generic object recognition but lack characteristics of a large-scale representative dataset covering many sorts of invariance (e.g., background clutter, shape, occlusion, size). For example, the COIL dataset [41], which also used a turntable to film 100 objects under various lightings and poses, contains one object instance per category (e.g., one telephone, one mug). The larger ALOI dataset [19] contains 1,000 objects but few instances per category. The NORB dataset [35] has 50 small toy objects (10 instances in each of 5 categories). Almost all available turntable datasets are small scale and not very rich in terms of variations.

Previous research using controlled datasets, such as turntables images, has been mainly focused on inspecting models or brewing concepts and ideas. Some recent works have attempted to show that there is a real benefit of these datasets in transferring knowledge to large-scale natural scene datasets [26, 67]. This has been studied under the names of domain adaptation, task transfer, or multi-task learning. The idea here is that knowledge gained from a controlled dataset (or task), via turntables or graphic models, can be transferred to real-world naturalistic datasets with even different statistics (e.g., texture). For example, Peng et al. [43] trained models on an augment of synthetically generated images (using a 3D graphics object model) and natural scenes (from ImageNet and PASCAL) and reported an improvement in accuracy over the latter datasets. They, however, did not probe whether the improvement was due to learning better invariance or instance level variety and richness. Some other works have also advocated similar directions [21, 49, 11, 16].

Another motivation for utilizing controlled datasets comes from neuroscience and cognitive vision literature. CNNs were initially inspired by the hierarchical structure of the visual ventral stream [18]. They were later used to explain some physiological and behavioral data of humans and monkeys (e.g., [46, 52, 65, 51]). It has been asserted that humans learn invariance with few exemplars a.k.a. zero- or one-shot learning. This is the opposite of the way that CNNs currently learn recognition. These models need an enormous amount of labeled data. In this work, we explore the ways a large-scale controlled dataset, containing rich information regarding various object parameters, can be utilized to improve object recognition performance. It is important to be aware of human performance to gauge the progress [4]. Just recently, He et al. [24] reported a top-5 error rate of 4.9% on ImageNet which is lower than 5.1% human error rate on this dataset [47]. This raises some questions: Have models surpassed humans? If yes, in what aspects? Is it theoretically possible to achieve a better performance than humans on these problems? etc.

Another related area to our work, which naturally fits well to turntable setups, is the manifold embedding and di-
mensionality reduction. These techniques try to preserve and leverage the underlying low dimensional manifold in data in supervised or unsupervised manners (e.g., [69, 59]). For instance, Weston et al. [63] introduced an embedding-based regularizer to impose the same labels for the neighboring training samples to benefit from the structure in the data. They used gradient descent to optimize the regularizer and adopted it for CNNs. Another classic example is Siamese Networks [5] which are two identical copies of the same network, with the same weights, fed into a ‘distance measuring’ layer to compute whether the two examples are similar or not. Given the labeled data, the network encourages similar examples to be close, and dissimilar ones to have a certain minimum distance from each other. While these techniques have been applied to controlled datasets, their usefulness over large-scale controlled datasets still remains to be explored. Our proposed dataset can be helpful in this direction as it combines the best of the two worlds: instance-level variety of large-scale datasets and rich parametrization of controlled synthetic images which are precious to study probing the behavior of CNNs.

3. The iLab-20M dataset

Many image datasets have been proposed to assist machine vision algorithm development and testing (Table 1). Those datasets which have provided large collections of training exemplars per well-defined object category have been useful in advancing the state of the art. Excellent examples include FERET for face recognition [44], with 14,051 images of 1,199 individuals in one class (human faces), or MNIST for handwritten digits [34], with 60,000 images in 10 classes from 500 writers. Today, recognizing faces or handwritten digits is considered a reasonably well solved problem, although of course improving tolerance to noise and other nuisance parameters is always possible.

In other domains, including recognition of objects from generic categories, most efforts have focused on providing very useful test sets and performance challenges (e.g., ImageNet [8]), but these often lack in the sheer volume of training exemplars provided within each object category and for each object instance, lack pose information, and often contain occlusions. This limits their usefulness for training. For example, the ‘calculator’ category of Caltech-256 [22] contains 100 images of what appears to be 100 different calculators with no pose data. While this is highly appropriate for testing, we hypothesize that training can be greatly improved by using many different views of different instances of objects in a number of categories, shot in many different environments, and with pose information explicitly known. Indeed, biological systems can rely on object persistence and active vision to obtain many different views of a new physical object. In humans and monkeys, this is believed to be exploited by the neural representations [37], though the exact mechanisms remain poorly understood. Although adult humans can learn new object instances from a single view, one should not forget that this ability might only emerge at the culmination of a long evolutionary process plus 20-some years of individual training.

Popular datasets fall short in at least one dimension, be it the number of classes, objects per class, number of backgrounds/environments, or views per object, as shown in Table 1. Particularly relevant to our effort are: 1) COIL [41], which also used a turntable to film 100 objects under various lighting and poses; however, COIL only contains one object instance per category and only black backgrounds (similar to the larger ALOI dataset with 1,000 objects and a few per category [19]), and 2) NORB [35] with 50 small toy objects similar to the ones we used (10 instances in each of 5 categories); however, all objects were painted uniformly and shot in grayscale on black backgrounds (different backgrounds were later composited digitally).
Table 2. Summary statistics of iLab-20M dataset. There are 21,798,480 images in total from 16 categories (one used for calibration purposes only) with 25 to 160 instances per category. Five parameters include: 11 cameras on an arch, 4 lighting sources on 4 corners (5 conditions), 8 horizontal rotations, 132 backgrounds (7 solid color) and 3 focus values. Average number of backgrounds per object instance is 23.39. There are 46 unique backgrounds per category (average backgrounds per object 145.76 with std = 162.62; min = 25, max = 731). Total size of the dataset with resolution 960 × 720 is 17.65TB. The cropped version of the images (256 × 256 pixels) is also available with 2.2TB in size. Total number of images per category is rounded to save space.

### 3.1. Turntable setup
The turntable consists of a 14”-diameter circular plate actuated by a robotic servo mechanism. A CNC-machined semi-circular arch (radius 8.5”) holds eleven Logitech C910 USB webcams which capture color images of the objects placed on the turntable (Fig. 1.a). A micro-controller system actuates the rotation servo mechanism and switches on and off four LED lightbulbs (Ecosmart ECS 16 WW FL, 295 lumens each, color rendering index 87, correlated color temperature 3000K). Lights are controlled independently, in 5 conditions: all lights on, or one of the four lights on.

Cameras were connected to a Linux computer (6-core AMD Phenom CPU, 16GB RAM) with 11 independent USB controllers. Camera settings were as follows, using the Linux V4L2 driver: resolution 960 × 720, color mode YUYV, brightness 128 (default for these cameras), contrast 32 (default), saturation 32 (default), gain 30, manual white balance off, manual white balance temperature 3100K, sharpness 72 (default), auto white balance off, manual color temperature 3000K. Lights are controlled independently, in 5 conditions: all lights on, or one of the four lights on.

In total, 1,320 images were captured for each object and background combination: 11 azimuth angles (from the 11 cameras), 8 turntable rotation angles, 5 lighting conditions, and 3 focus values (-3, 0, and +3 from the default focus of each camera). Each image was saved with lossless PNG compression (~1 MB per image). The complete dataset hence consists of 704 objects, each shot on 14 or more backgrounds, with 1,320 images per object/background combination, or almost 22M images (See Table 2). The dataset is freely available and distributed on 3 8TB hard drives.

### 3.2. Robotics-assisted model scenes
In addition, we created robotics-assisted model scenes to record broader scenes where objects were placed in variable contexts. The long-term motivation for this larger scenery is to collect many images which can be used to test algorithms both on their ability to first locate and then to recognize objects, and on their possible ability to exploit larger scene contexts to aid recognition (see, for example [12, 25]).

The robotics-assisted scenes (Fig. 1.b) consist of a 40"×29" table onto which 1:160 poster prints of satellite images (e.g., Google maps) are placed (corresponding to a real-world area of 195m×118m). One 8-axis robot arm holds a camera (Microsoft LifeCam Cinema, 1280×720, YUYV) which can be placed and oriented at any location and pose reachable by the arm. A second arm holds a light source (Jingsam LED 7W, 437 lumens, 3000K).

The robots are programmed in two ways: 1) pseudo-random motion, generating flybys, 2) point to specific locations on the table and shoot objects from different viewpoints and distances. An interactive user interface assists in configuring a scene for robotics-assisted filming.

### 4. Experiments and results
To start exercising the dataset, we tested it on small subsets of the available data. To understand generalization across image variations (object shape, object viewpoint, lighting, etc), CNNs are evaluated by taking slices of the dataset. We utilize pre-trained Alexnet [31] (on ImageNet) and fine-tune it on iLab-20M. The behavior of off-the-shelf features is investigated in our analyses as well. We use 7 object categories (out of 16) and avoid data augmentation as we have flipped versions of the objects from the turntable. The label layer contains several units depending on the task.
(2, 4 or 7 for object categorization; variable number of units for parameter prediction). We report average accuracies and standard deviations where there is randomness in the experimental procedure. Experiments are performed using the publicly available Caffe toolkit [27] ran on a Nvidia Titan X GPU and Ubuntu 14.04 OS.

We aim to answer these questions: Can a pre-trained CNN model predict the setting parameters such as lighting source, degree of azimuthal rotation, degree of camera elevation, etc? Can it transfer the learned knowledge from one object category to another? Which parameters are more important in the transfer? How much knowledge can a model transfer from iLab-20M to the ImageNet? Which one is a better strategy to make an object dataset: random or systematic image harvesting? How the order of learning parameter invariance influences overall network parameter tolerance and accuracy? Some of these questions have been addressed in the past to some extent [1, 68, 70, 10, 40].

4.1. Selectivity and invariance

Humans are very good at predicting the category of an object and also telling about its parameters. Human visual system is selective to object category and invariant to parameters and variations. In this experiment, we aim to systematically investigate this competition for two layers of the Alexnet: pool5 and fc7. We probe the expressive power of these layers for object category and parameter prediction.

Four categories from iLab-20M (out of 16) were chosen for this analysis including boat, bus, tank and ufo. Images were lumped to train a SVM classifier. All features were normalized to have zero-mean before feeding to the classifier. The dimensionality was reduced to N-dimensions using SVD, where N refers to the number of instances in the training set. The reported results are average accuracy over random 5-fold cross validation test sets, each of size 2K. We trained two SVMs, one for category prediction and another for parameter prediction. Results are shown in Fig. 2.

As expected, we see that fc7 features result in a high classification accuracy, however, the surprising salient result is the shoulder-to-shoulder performance of pool5 and fc7 layers. Relying on this outcome, it seems that both fc7 and pool5 representations convey useful discriminative information for object recognition. Comparing the performance over parameter prediction, one can notice the superiority of pool5 layer over fc7. This is consistent with the work by Bakry et al. [2] where they analytically found that fully connected layers make effort to collapse the low-dimensional intrinsic parameter manifolds to achieve invariant representations. However, only view manifold was taken into consideration in Bakry et al.’s work, while here we analyze the behavior of more common parameters.

In brief, our results suggest that the feature space spanned by pool5 layer contains more information than fc7 layer for parameter prediction. At the same time, the very representation forces different categories to be highly apart from each other (thus keeping the structure of manifolds as linearly-separable as possible for different categories). The representation by fc7 sensibly discards parameter information to become invariant while keeping the categories as separable as possible. We observe that the layer just before fully connected layers provides better compromise between categorization and parameter estimation.

Parameter prediction accuracies for lighting (5 classes), turntable rotation (4 classes), and camera view (6 classes) in order are 100%, 62%, and 77%. These figures suggest that camera view (considering the normalized-to-chance accuracy) has the most complex structure for parameter prediction whereas the lighting is simpler. This is somewhat sensible since changing camera view leads to geometric shape variations, and ports the prediction task into a much more difficult problem to address. In contrast, lighting variations do not alter the shape of the object, and are thus easier to capture. Note that this result is on our data and may not necessarily scale to natural scenes.

We use the t-SNE dimensionality reduction method in [62] to visualize the learned representations over seven

![Figure 3. t-SNE representation of the Alexnet layers. The fc7 representation works remarkably well at recognizing objects as they are mutually linearly separable after fine-tuning. Further, pool5 representation does not contain discriminative information compared to fc7. This figure also demonstrates the effect of fine-tuning. Distribution of samples for different categories tend to become very compact after fine-tuning. Fine-tuning does not seem to add more discriminative power to the pool5 representation.](image-url)
categories of iLab-20M along with variation parameters (See Figure 3). The fc7 representation works remarkably well at recognizing objects as they are mutually linearly separable after fine-tuning. Further, pool5 representation does not contain discriminative information compared to fc7. Please see also the supplement for more details.

4.2. Knowledge transfer

Humans are very efficient at estimating and transferring parameters of a seen object to another unseen object in complicated scenarios. For example, they can reliably estimate the lighting source direction of an object and tell whether another object has been subject to the same lighting exposure. Complementary to our previous analysis, in this experiment, we aim to assess the power of CNNs in transferring the learned parameter over one object category to another. We focus on pool5 layer here since as we discussed above, fc7 layer is invariant to parameters and is thus less useful for discriminating between different parameters.

All parameters were fixed except one (i.e., slicing the dataset along only one parameter). We included instances from four categories (boat, bus, tank, ufo) in the training set, and tested the learned knowledge on instances from an unseen category (F1Car) as well as 4 seen categories (but different instances). We utilized the pool5 representation and reduced the dimensionality to \( N \), where \( N \) refers to number of samples. The 5-fold cross validation average accuracy for parameter prediction is shown in Fig. 4.

Results show a decent degree of knowledge transfer. As Fig. 4 exhibits, the lighting parameter is relatively easier to be transferred to unseen categories. It has a head-to-head accuracy across seen and unseen categories. On the other hand, knowledge transfer for rotation and camera view parameters is accompanied with sensible degradation in performance. In summary, we see that the knowledge is promisingly transferable across seen and unseen categories. The degradation in rotation and camera prediction is intuitively justifiable as these parameters are highly dependent on the 3D properties of the object shape (See also [36]).

4.3. Systematic and random sampling

Large-scale datasets have been so far constructed by harvesting images randomly from the web. The major reasoning for doing so is to include as much variability (mainly intra- and inter-class variation) as possible in the dataset. It has not yet been systematically studied whether this is a good strategy compared to controlled strategies conducted in turntable setups. In this analysis, we consider two strategies to find the answer: i) Random strategy where \( n \) samples (across all parameters and instances) are chosen randomly and are used to train an SVM to predict the object category, and ii) Systematic (or exhaustive) strategy, in which an object instance is chosen randomly and then other images from that object are added to our training set, by scanning all parameters, until \( n \) samples are reached. We assume availability of a fixed limited budget (time or cost) enough for processing only \( n \) samples.

We addressed a 4 class problem (boat, bus, tank, ufo) by increasing \( n \) starting from 12 up to 10,000 samples. In each experiment, \( n/4 \) samples were chosen randomly from all 4 categories across all parameters, and were fed into the Alexnet to get the fc7 (or pool5) representation. Then, we trained a linear SVM classifier on this data. A fixed test set of size 500 was randomly selected from all categories with all parameters and was kept fixed during the analysis. We measured category prediction at fc7 and parameter prediction at pool5, reducing the dimensionality to 2,500 for all values of \( n \) in the latter. Results are shown in Fig. 5.

We observe that random sampling strategy performs better in category prediction. This makes sense since randomly choosing images offers more instance level variety (better than systematic) leading to better recognition. Interestingly, and counter-intuitively, we see that random strategy works better in parameter prediction as well. We believe that the parameter prediction is somewhat dependent on the 3D properties of object shape, and since in the systematic strategy, the learner is not faced with sufficient instances, it fails to predict parameters compared to random strategy. Overall, what we learn is that instance level variation is of
high importance for both category and parameter prediction and this is perhaps why the systematic sampling strategy is hindered. Thus, in dataset creation, it is vitally advantageous to have as much instance level variation as possible.

### 4.4. Domain adaptation

Currently, there is a gap in relating results learned over synthetic datasets to results learned on large-scale datasets. We train models on iLab-20M and apply them to natural scenes (and vice versa) to see how much knowledge they can transfer from one dataset (source domain) to another (target domain). This way, we can also discover along which dimension(s) a dataset varies the most and whether it offers sufficient variability for learning invariance. In other words, we can somehow indirectly measure dataset bias \([60]\). Ultimately, it is desirable to generalize what is learned from synthetic datasets to natural scene datasets.

We consider two scenarios: \(i\) a binary classification problem \(\text{boat vs. tank}\), and \(ii\) a 4-class problem including \(\text{boat, tank, bus and train}\). In each scenario, we train a SVM (using \(fc7\) representation) from either natural scenes (selected from ImageNet) or iLab-20M and apply it to the other dataset. We also merge images from the two datasets and measure the accuracy on each individual dataset. We consider both off-the-shelf features of the Alexnet (pre-trained over ImageNet) and fine-tuned features over iLab-20M.

**Augmenting data along all parameters:** Here, we choose images along all parameters. Results in Table 3 show that training on each type of image, expectedly works the best on the same type of test image (95% from ImageNet to ImageNet and 97% from iLab-20M to iLab-20M). Cross application of models results in lower (but above 50% chance) accuracy. We observe that fine tuning the Alexnet on iLab-20M boosts the performance on iLab-20M to 100% while hindering the accuracy over ImageNet as CNN features are now tailored (and are hence selective) to our images.

Table 4 shows domain adaptation results over 4 classes. Results align with accuracies over 2 classes, although accuracies are lower here. Here, again combining images from datasets hinders performance over each individual dataset due to contamination of features. The reason why performance is low when applying a model trained on iLab-20M to ImageNet is mainly because objects in these two datasets have different textures and statistics which demand more sophisticated ways of domain adaptation.

### Table 4. Domain adaptation over a 4-class problem (boat, tank, bus, and train). Numbers in parentheses are standard deviations.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Without fine tuning</th>
<th>With fine tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>95 (0.5)</td>
<td>95.56 (0.6)</td>
</tr>
<tr>
<td>iLab-20M [2000]</td>
<td>66.92 (3.2)</td>
<td>65.22 (1.4)</td>
</tr>
<tr>
<td>iLab-20M [1000]</td>
<td>94.42 (0.8)</td>
<td>92.52 (0.2)</td>
</tr>
<tr>
<td>Natural + iLab-20M [1000]</td>
<td>94.9 (0.8)</td>
<td>92.5 (0.2)</td>
</tr>
</tbody>
</table>

Figure 6. Confusion matrices of Alexnet over seven categories of the iLab-20M dataset without (left) and with fine tuning (right).

Please see the supplementary material for t-SNE visualization \([62]\) of without- and with fine-tuned \(fc7\) and \(pool5\) features.
least effective parameter in our analysis. The absence of camera-view drops the recognition accuracy more than the other two parameters (the right side bars in Fig. 7).

### 4.5. Analysis of parameter learning order

In this analysis, we study whether/how the order of knowledge delivery to CNNs matters. First, we prepare two datasets (training with 40K images, validation with 10K) from four categories (boat, bus, tank and train) and annotate them with rotation labels. Alexnet is fine tuned on the training set. We set the learning rate for all the layers to 0.001, except fc8 layer which is set to 0.01. All other parameters are set to their default values. Next, we prepare a new training set including 40K images from the same four object categories and annotate them with camera view labels. A validation set of size 10K is also constructed. Obtained weights from the first step are loaded to the network and are treated as a promising initialization point for another round of fine tuning over the new data.

We assess the performance of the network for camera view and rotation prediction using the pool5 representation. As fine tuning with low learning rate slightly changes weights within the network, we are interested to see which order of changes in weights (before fully connected layers) gives the superior performance in our desired task. To hunt what we are looking for, prepared datasets are delivered to the network in reverse order (i.e., camera first, rotation next). We denote the two orderings as follows: 1) rotation-camera, and 2) camera-rotation for simpler reference. In the evaluation phase, 2,000 samples are randomly selected from four categories, and pool5 features are extracted. After mean subtraction and dimensionality reduction, 5-fold cross validation accuracies of models are reported in Table 5.

Counter-intuitively, we find that order of data delivery is very important to the network such that when the network is fed with samples with rotation labels prior to camera labels, it ostensibly performs better in parameter prediction. We also find that when the network is firstly fine tuned on rotation, the second stage (i.e., fine tuning on camera labels) does not impair the weights for rotation prediction. In contrast, when the camera labels are seen first, rotation prediction accuracy is expectedly better than the previous ordering. This boost, however, causes dramatic degradation in camera prediction performance.

As in the previous experiments, camera view variation is a more ill-structured parameter to predict. When the network sees the camera labels in the second stage, the adapted weights are more biased towards learning this parameter. This bias does also try to keep the pre-seen knowledge for rotation unchanged. We thus conclude that when there is the option for stage-wise training, it would be better to learn parameters following a simple to complex order. This way, the last steps are devoted to manage the difficulties in complex parameters, while imposing less damage to weights adapted for simpler parameters (thus maintaining the structure).

### 5. Discussion

We challenged the solitary use of uncontrolled natural image datasets in guiding the object recognition progress and introduced a large-scale controlled object dataset of over 20M images with a rich parameter variety. By cutting slices through our dataset, we systematically studied the invariance and generalization properties of CNNs by independently varying the choice of object instances, viewpoints, lighting conditions, or backgrounds between training and test sets. Progressively extending these results on increasingly larger subsets of our dataset may help gain new insights on how the algorithms can be modified to show greater invariance and generalization capabilities.

In summary, we learn that: 

1. The representation learned in pool5 layer is selective to parameters while fc7 layer is not. 
2. The knowledge obtained from some parameters is easier to be transferred to unseen object categories. 
3. Random sampling strategy leads to better generalization since more instance level variations can be captured. 
4. Simple cross application of one dataset to another results in above chance accuracy but does not improve performance, and 
5. It would be advantageous to feed the network with data that has been sorted according to complexities of different dimensions. This can lead to layer-wise training of CNNs for learning different invariances in different layers.

In the future, we will attempt to evaluate the accuracy of recent deep learning architectures on our dataset. In particular, we will consider techniques such as feature embedding and loss regularization [63, 5] and joint prediction of camera parameters and object categories [13, 53].

**Acknowledgments:** This work was supported by the National Science Foundation (grant numbers CCF-1317433 and CNS-1545089), the Army Research Office (W911NF-12-1-0433), and the Office of Naval Research (N00014-13-1-0563). The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof. We also wish to thank NVIDIA for their generous donation of the GPU used in this study.

---

**Table 5. Influence of data delivery order on parameter prediction.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Order 1 [rotation-camera]</th>
<th>Order 2 [camera-rotation]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>89.20% (1.47)</td>
<td>77.05% (1.18)</td>
</tr>
<tr>
<td>Rotation</td>
<td>93.75% (1.66)</td>
<td>95.30% (1.00)</td>
</tr>
</tbody>
</table>

---

**Figure 7. Domain adaptation with a single parameter change.**

![Figure 7](image_url)
References


