Robust Single-View Instance Recognition

David Held, Sebastian Thrun, Silvio Savarese

***Abstract*— Some robots must repeatedly interact with a fixed set of objects in their environment. To operate correctly, it is helpful for the robot to be able to recognize the object instances that it repeatedly encounters. However, current methods for recognizing object instances require that, during training, many pictures are taken of each object from a large number of viewing angles. This procedure is slow and requires much manual effort before the robot can begin to operate in a new environment. We have developed a novel procedure for training a neural network to recognize a set of objects from just a single training image per object, allowing a user to easily train a robot to operate in a new environment. To obtain robustness to changes in viewpoint, we take advantage of a supplementary dataset in which we observe a separate (non-overlapping) set of objects from multiple viewpoints. After pre-training the network in a novel multi-stage fashion, the network can robustly recognize new object instances given just a single training image of each object. If more images of each object are available, the performance improves. We perform a thorough analysis comparing our novel training procedure to traditional neural network pre-training techniques, as well as previous state-of-the-art approaches including keypoint-matching, template-matching, and sparse coding, and we demonstrate that our method significantly outperforms these previous approaches. Our method can thus be used to easily teach a robot to recognize a new set of object instances in order to operate in a fixed environment.**

I. INTRODUCTION

There are many robotics applications in which a robot must interact with a fixed set of objects. For example, a robot in a factory would need to recognize the objects on the conveyer belt. A lab assistant robot would need to recognize the lab equipment, such as beakers and test tubes. A cooking robot would need to recognize the cooking equipment. In these and other robotic settings, a user can tell the robot in advance which objects it needs to recognize.

Such applications have motivated the robotics community to work on the problem of object instance recognition [1], [2], [3], [4], [5], [6]. For this task, a user pre-defines a set of objects that the robot must recognize and trains a perception system to recognize these objects in the environment. Previous attempts to solve this problem require that the user rotate the object on a turn-table while recording a 3D scan. These 3D scans can then be combined to create a 3D object model [1], [2], [3] or used to train a classifier [4], [5]. However, this process requires special equipment to turn the object in a controlled fashion, and the process requires a fair amount of time from the user. We would like to enable users to quickly train robotic perception systems to recognize objects from just a small number of training images, or even from a single image.

D. Held, S. Thrun, and S. Savarese are with the Computer Science Department, Stanford University, Stanford, California 94305 USA {davheld, thrun, ssilvio}@cs.stanford.edu

Fig. 1. Given only a single image of an object, we want to recognize this object from novel viewpoints. Traditional neural networks pre-train on ImageNet alone. We perform a multi-stage training procedure, in which we first pre-train on a large class-level dataset (left), followed by pre-training on an auxiliary multi-view dataset (middle), which trains our network to be robust to viewpoint changes. Finally, we train on the objects we wish to recognize from just a single image (right).

Further, for some real-world applications, only a small number of images of the target objects may be available. For example, a user may wish to train a robot to recognize objects from a product databases available on Amazon, Safeway, or another website. Unfortunately, for many of these product databases, only a small number of images are available for each product. If a perception system could robustly recognize objects given just a small number of training images, a large number of robotics applications would be available that make use of such product databases.

We introduce a new approach to training neural networks to recognize objects from just a single training image, using a general-to-specific training procedure. We start by pre-training our network in the traditional fashion, using a large class-level dataset. Our insight is that, while this procedure teaches the network to be robust to intra-class variation, the network has not yet learned to be robust changes in viewpoint. Thus, we continue pre-training our network using a smaller multi-view dataset in which we observe a set of objects from multiple viewpoints. Finally, we train our network on a separate dataset in which only a single image is available for each object instance. This procedure is depicted in Figure 1.

By training our network in this general-to-specific manner, our network learns the invariances that it needs to perform the final task. Our network initially learns general visual properties about the world. It then learns object invariances, enabling the network to be robust to rotations and changes in viewpoint. Finally, our network learns to recognize a specific set of objects from just a single training image per object.

Using this novel multi-stage training procedure, our network learns to robustly recognize objects from new view-

points. To our knowledge, this is the first work that uses deep learning to recognize specific object instances from a single image. We perform an extensive evaluation and show that multi-view pre-training outperforms previous state-of-the-art approaches for recognizing both textured and untextured objects from novel viewpoints. If more than one training image is available, our performance will improve, and we continue to outperform all baseline approaches for any given number of training images.

II. Related Work

Instance recognition has traditionally been achieved by matching either 2D or 3D keypoints across images [7], [8], [9], [10]. Keypoints can be filtered using different criteria [7] and validated using RANSAC or Hough Voting to ensure geometric consistency [11]. Although keypoint-based approaches have shown some success for instance recognition, 2D keypoints are unreliable for recognizing untextured objects or non-planar objects when the viewpoint is changed by more than 25 degrees [12]. 3D keypoints are often not sufficiently discriminative to recognize a wide range of objects.

Template matching has also been used for instance recognition [13]. Much work has recently been done to make template matching scalable, efficient, and robust to occlusions [14], [15]. However, viewpoint invariance is usually achieved by recording many templates during training from different viewing angles. If only a small number of images are available from each object during training, template matching methods will not robustly detect the target object, as we will demonstrate.

Others have approached this problem by building a 3D model of each object and then fitting the 3D model to the scene [2], [3]. However, the process required to create these models is slow and often requires a specialized setup to carefully scan the object from different views. Our approach outperforms these methods without requiring any special equipment, and we demonstrate robustness when training from just a single image or a small number of images, thus significantly easing the burden on the user.

Another approach that can be used for recognizing objects is to use machine learning methods to train an object classifier [5]. One example of such a classifier that has shown great success in recent years is a convolutional neural network [16], [17], [18]. However, statistical methods such as neural networks typically require many training examples to perform well. For example, for the ImageNet challenge, participants train their methods on 1.2 million training examples [19]. We will show that our approach can be used to recognize object instances from new viewpoints given only 1 training example per object, and we significantly outperform previous approaches.

One-shot learning has also been explored for classifying objects at the category-level [20], [21], [22] or for recognizing handwritten characters [23]. In contrast, we focus on recognizing object instances from new viewpoints, and we compare our approach to state of the art techniques for object instance recognition.

Our method makes use of a separate multi-view dataset to improve performance on the task of instance recognition from a single training image. Our idea of using a supplemental multi-view dataset is related to previous efforts to improve recognition performance by using a video sequence [24], [25]. Another related effort is to use unlabeled video for unsupervised feature learning [26], [27]. These methods typically enforce the consistency of features between subsequent video frames. We instead use multi-view objects in a classification setting to improve our performance for recognizing single-view objects, and we do not treat the multi-view dataset as a linear video sequence.

III. Method

A. Problem Setup

Suppose that we are given an image x_i of an object instance that we want to recognize. We assume that we have a "single-view" database of K_S different objects, and that the object in our image x_i is one of the K_S objects in our single-view database. We also assume that each of the objects in our database has only one image taken of it. Given that our image x_i is likely to be taken from a novel viewpoint relative to the images in our database, how can we robustly identify the instance label for this object?

In order to robustly perform this task, we suppose that we also have a separate "multi-view" set of K_M objects for which we have recorded images from many viewpoints. Because we have observed each of these separate objects from many viewing angles, we can use these images to teach our method to be invariant to viewpoint changes. Then, given a novel viewpoint of an object from the single-view dataset, we can use this learned invariance to correctly recognize the target object.

Note that the multi-view objects are chosen so that there is no overlap between the K_M multi-view objects and the K_S single-view objects. Thus, any invariances that we learn from the multi-view dataset must be general to be able to transfer over to a new set of objects. Our final goal is to identify an image x_i as belonging to one of the K_S single-view objects; the multi-view dataset is helpful only in teaching our method to be invariant to viewpoint changes.

B. Multi-View Pre-Training

We consider instance recognition as a classification problem, and we will explore the use of neural networks to perform this task. Because neural networks represent a non-convex decision boundary, the initialization of the network is important. One common approach for training a neural network with a limited amount of data is to initialize the network by pre-training on a larger dataset [28] (e.g. ImageNet [19]). These initial weights are then fine-tuned using a smaller dataset for the relevant task. This training procedure allows the network to find a better local optimum.

However, the ability to transfer information from the larger dataset to the smaller dataset, via network initialization,

depends on the similarity between the datasets. If the datasets are not very similar, then this initialization will be poor [29]. As we will show, pre-training the network for class-level recognition (e.g. using ImageNet) is not ideal for training these networks to be viewpoint invariant with respect to specific object instances.

For the original ImageNet classification task, the goal of the network is to recognize 1000 different object classes. Each class represents an object category, such as "restaurant" or "mask," and the appearance of objects within the class can vary dramatically; different restaurants can have a very different appearance. Because the network must recognize generic object classes, the computational effort of the network is spent attempting to handle all of the different aspects of intra-class variability. On the other hand, if our goal is to perform object instance recognition, then we can focus our network's computational effort on being robust to rotations, leading to better performance at this task.

We will show that, although pre-training our network on ImageNet provides a decent initialization for our network, we can obtain better performance through a multi-stage training procedure, as follows:

1) Train our network on a large class-level dataset.
2) Train our network on an instance-level dataset with many views per object instance.
3) Train our network to recognize a new set of object instances from a single image per object.

This setup is illustrated in Figure 1. In more detail, we initially pre-train our network on a large class-level dataset, e.g. ImageNet, which allows our network to learn general image statistics. We then train our network on a smaller dataset in which we observe a set of objects from multiple viewpoints, and we learn to recognize these objects instances. This stage allows our network to learn to be robust to changes in viewpoint. Finally, we train our network on a separate dataset in which only a single image is available for each object. We show that adding an intermediate multi-view pre-training step (step 2 above) gives better performance than pre-training only on a class-level dataset. Adding multi-view pre-training increases the robustness of our network and enables us to recognize novel objects from new viewpoints.

We would also like to be able to recognize objects in real scenes against random backgrounds. To make our network robust to different backgrounds, during multi-view pre-training (step 2) we synthetically place the objects against random background scenes which do not contain any of the test objects. Although the single-view objects that we wish to recognize are placed against a fixed background for training (in step 3), we will show that pre-training with separate multi-view objects against random backgrounds in step 2 allows our method to learn to be robust to new backgrounds.

One can view our approach as an extension of data augmentation techniques for neural networks. It is common when training neural networks to perform multiple image transformations on each training example to synthetically generate more training examples. Common transformations include crops, horizontal flips, and synthetic relighting [18].

These data augmentation methods are an attempt to train the network to be robust to translations or changes in lighting. However, it is more difficult to construct an image transformation that would simulate an out-of-plane rotation. As an alternative, we propose multi-view pre-training, in which our intermediate training stage involves classifying a separate set of objects from multiple viewpoints. Multi-view pre-training allows our network to learn new kinds of invariances, such as out-of-plane rotations, that would be hard to simulate using data augmentations.

C. Network Details

Our neural network uses the CaffeNet architecture [30], which is very similar to the architecture proposed by Krizhevsky et al [18]. The network is initially pre-trained on ImageNet [19]. We then fine-tune this network on the multi-view dataset as follows: we replace the final layer with a K_M class classifier, and we fine-tune the weights to classify the K_M multi-view objects. We call this step "multi-view pre-training" since we are training the network to recognize object instances given multiple views of each object. During multi-view pre-training, we hold the convolutional layers fixed and only fine-tune the fully-connected layers on top.

During multi-view pre-training, we use a learning rate of 0.001 for all layers except the final layer, which we set to a learning rate of 0.01. After 50,000 iterations, we reduce the learning rate by a factor of 10, and after 100,000 iterations we stop the multi-view training. These parameters were determined using a hold-out validation set, and other hyperparameters are taken from the default parameters for CaffeNet [30].

Finally, we fine-tune the network to classify the single-view objects. To do this, we replace the final layer with a K_S class classifier for the K_S single-view objects. Each object in this dataset has only 1 training example from a single viewpoint. We use the same parameters as before, except that the learning rates are reduced by a factor of 10, which was again determined using cross-validation on a hold-out set. The final classifier is used to classify these K_S objects from novel viewpoints. We call a classifier trained in this manner a "neural network with multi-view pre-training."

IV. Results

We perform a number of experiments to analyze the performance of different instance recognition methods. In Sections IV-A through IV-C, we use the RGB-D object dataset [4], in which we recognize objects that are placed on a turntable and recorded from different viewpoints. In this controlled setup, we can measure the object's angular difference between the training and test images, allowing us to compute how robust the different methods are to out-of-plane rotations. Finally, in Section IV-D we will evaluate the methods on recognition of objects in cluttered scenes.

We evaluate the performance on this dataset under three conditions:

1) Training from many examples (Section IV-A)

2) Training from a variable number of examples (Section IV-B)
3) Training from just a single example (Section IV-C)

In all three cases, we use the same test set, which is the RGB-D instance recognition test set [4].

We vary the number of examples available during training to show how well each method generalizes with a limited number of training examples. When multiple training examples are available, we compare the neural network-based approaches to other machine learning approaches. When only one training example is available, we also evaluate keypoint-matching and other approaches that are designed to match pairs of images. We find that neural networks have superior performance in all three cases, and we further show the advantage of multi-view pre-training in the case of training from just a single example.

Finally, in Section IV-D, we evaluate how robust the different classifiers are to handling occlusions and backgrounds. For this we use the RGB-D scenes dataset [4], in which the objects from the previous test set are placed in a cluttered scene. Our task now is to recognize the object given the object's bounding box. In an end-to-end system, the bounding box could be generated using one of the many region proposal methods that have been developed for this purpose [31], [32], [33], or the object can be segmented from the scene using depth information [34], [35]. Although we can no longer compute the angular difference between training and test images in this less controlled setting, this experiment allows us to determine how robust the different methods are to recognition against a cluttered background and under occlusions. For this task, we use the same training set as before, i.e. training from just a single example. We also evaluate the performance as a function of the noise in the bounding box location and show that our method is robust to such variations.

A. Instance recognition from many examples

We first evaluate our method using the RGB-D object dataset [4], and we measure the performance when many training examples are available. This dataset consists of 300 objects of different types and textures, ranging from apples to cereal boxes. Given an image of one of these objects taken from a novel viewpoint, our task is to identify which of the 300 objects this image is taken from. We treat this task as a 300-class classification problem, and we are thus able to apply tools from machine learning to perform this task. Objects are pre-segmented in the training and test sets using depth information [4]. In Section IV-D we will explore the performance of the different methods when objects are placed in a cluttered scene where segmentation is not as simple.

We initially evaluate our method using the "leave sequence out" training setup [4]. In this setup, we observe each object at a 30 degree and 60 degree elevation angle during training, and we observe the object at a 45 degree elevation angle at test time. During training, we observe the object from many views spaced 6 to 9 degrees apart in azimuth.

TABLE I
TRAINING FROM MANY VIEWS

Method	% Accuracy
SIFT + Texton + Color Hist [4]	60.7
SIFT + Texton + Color Hist + Spin Img + 3D BB [4]	74.8
Convolutional k-means descriptor [38]	90.4
HMP (Depth) [5]	51.7
HMP (RGB) [5]	92.1
HMP (RGB + Depth) [5]	92.8
Neural network (Ours)	**93.3**

The results for this setup when many training images are available can be seen in Table I. One method that we compare against combines a number of 2D and 3D features [4], including dense SIFT [7], texton histograms [36], a color histogram, spin images [37], and 3D bounding box size. The methods that learn feature descriptors, such as [38] and [5], perform significantly better, achieving accuracies between 90.4 and 92.1%. The results from [5] indicate that only small gains are achieved by adding depth information, after the initial depth-based segmentation.

We evaluate the performance of a neural network pre-trained only with ImageNet (with no multi-view pre-training). Using such a network, we are able to outperform all of these previous methods, obtaining an accuracy of 93.3%. This method uses depth information only for segmentation, although depth could likely be used to further improve recognition performance. However, in Section IV-B we show that all of these approaches have poor performance when the number of training examples per object is limited, thus motivating the use of multi-view pre-training for such cases.

B. Varying the number of training images

We note that the dense training setup [4] used in Section IV-A requires a fair amount of manual effort. Objects were placed on a turn-table and recorded at many different angles and elevations. A casual user will want to be able to train a robot to recognize an object after taking only a few pictures during training. Further, for training from a typical product database (e.g. Amazon or Safeway), only a few images of each object may be available.

We therefore create a new training setup to test the performance of these methods when only a limited number of training images are available. Each object is now viewed at training time at only a 30 degree elevation angle. We vary the number of azimuthal angles for which an object is observed in training from 69 viewpoints down to just a single viewpoint, and we show the performance as a function of the number of training images. For any given number of training images, we evenly sample from the available training images for each object, starting from the first image. We can thus use this setup to determine how the performance of different methods are affected by the number of training examples.

Figure 2 shows the performance as we vary the number of views available during training. We compare the performance of the best methods of Section IV-A: HMP and a neural network pre-trained only on ImageNet (without multi-view pre-training).

As can be seen in this figure, the neural network saturates performance after about 10 training images. On the other hand, HMP [5] requires 20-30 training examples to saturate performance, and the result is still worse than that of the neural network. However, both methods perform poorly when only a single training example is available for each object. Because this is a situation that occurs often in practice, we would like to focus our attention on this scenario, which we call "one-shot learning for instance recognition." We will show that, when we have only one training example per object, we can improve the performance of a neural network by performing multi-view pre-training.

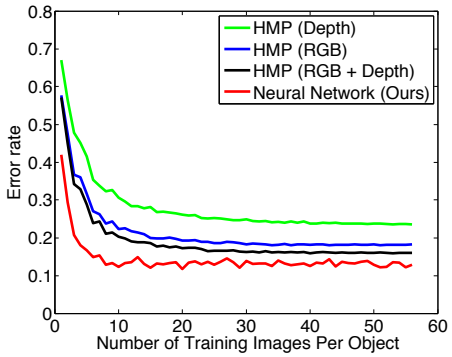

Fig. 2. We observe the effect on performance as we vary the number of training examples for a neural network as well as for the HMP baseline. The neural network that we evaluate here is pre-trained only on ImageNet, with no multi-view pre-training. The y-axis in this plot is the error rate (not accuracy). These results are not directly comparable to those of Table I because in this setup we are training on only a 30 degree elevation angle, whereas for Table I we trained on both 30 and 60 degree elevation angles.

C. One-shot Learning for Instance Recognition

1) Baseline Methods: In the next experimental setup, we are given only a single training example of each object. At test time, we would like to recognize each object from novel viewpoints. We use the same test set as in Section IV-A, making this a strictly harder (though more realistic) training scenario. For all objects, we train on only one training image at a 30 degree elevation angle, and we test on many different azimuthal viewpoints at a 45 degree elevation angle.

The results for this setup are shown in Table II. The keypoint-matching based methods perform poorly, ranging from 1.6% accuracy for BRISK [39] to 6.3% accuracy for SIFT [7].

As can be seen in Figure 3, HMP performs well when the test example is viewed from a similar angle as the training example. However, the performance drops off quickly as the angular difference between the training and test example increases. Note that, although we are varying the azimuthal angle difference from 0 to 180 degrees, all of the images have an additional 15 degree elevation angle difference between training and test. Given just a single training example, HMP is unable to find a good linear decision boundary that is viewpoint-invariant. As we discussed above, about 20-30 images are required for HMP to have good performance (see Figure 2).

2) Neural Networks: We first evaluate the performance of a neural network that is pre-trained in the traditional manner using ImageNet alone. After fine-tuning with just a single image per object, the network achieves an accuracy of 59.2%. Compared to the next-best method, this is an absolute improvement in accuracy of 16.9%, or a 29.3% drop in the number of errors.

We next experiment to see if we can gain an additional benefit from incorporating a separate multi-view dataset via multi-view pre-training. Note that the objects in the multi-view dataset are completely distinct from the 300 objects that we are trying to recognize. For this experiment, we use the multi-view BigBird dataset [1]. This dataset consists of 125 objects recorded from many different viewpoints, and to ensure that we have no overlap with the set of test objects, we remove the box of White Cheddar Cheez-it crackers, which also appears in the RGB-D object dataset [4]. We sample images from this dataset from 5 elevation angles and 20 azimuthal angles, for a total of 100 images per object. The multi-view dataset that we incorporate thus consists of a total of 12,400 images from 124 objects. Although this dataset took a fair amount of manual effort to construct, we will show that, once the network is pre-trained on this dataset, it can learn to recognize new objects from just a single image.

Our multi-view pre-training procedure follows the method described in Section III-C. After pre-training our network on the 1.2 million images from ImageNet, we further pre-train our network with the 124 objects from the multi-view dataset. Finally, we fine-tune the resulting network on the 300 objects from our single-view dataset, using just a single training example for each of the 300 objects.

Multi-view pre-training is especially impactful at improving the recognition of textured objects. By pre-training with a multi-view dataset, we obtain a 10.6% absolute improvement (or a 28.8% reduction in errors) on recognizing textured objects compared to the neural network pre-trained only on ImageNet. It is reasonable that multi-view pre-training gives a larger increase in performance on textured than untextured objects, since the appearance of textured objects changes more as a function of viewpoint compared to untextured objects. Thus, training our network to be invariant to rotations gives an especially large benefit for recognizing textured objects from novel viewpoints. Table II indicates that multi-view pre-training improves our performance for untextured objects as well.

Note that the multi-view dataset contains only 1% as many images as were used in the original ImageNet pre-training step. It is surprising that, given only 1% more images, we obtain a 10.6% improvement on the recognition of textured objects. Figure 4 shows some examples of objects that our method was able to correctly recognize that were incorrectly recognized by a neural network pre-trained on ImageNet alone.

D. Objects in a Scene

In the previous set of experiments, we used test objects placed on a turntable so we could measure the rotational

TABLE II

ONE-SHOT INSTANCE RECOGNITION

Method	% Accuracy		
	Overall	Textured	Untextured
Random guessing	0.3	0.3	0.3
BRISK [39]	1.6	2.6	1.3
ORB [40]	1.9	3.5	1.3
SURF [8]	3.4	5.3	2.6
BOLD [41]	5.2	5.9	4.9
SIFT [7]	6.3	12.6	3.9
Line-2D [14]	5.5	0.3	7.4
Color Histogram Intersection [42]	12.4	23.3	8.2
HMP (Depth) [5]	33.0	37.7	31.2
HMP (RGB) [5]	42.3	53.8	37.9
HMP (RGB + Depth) [5]	42.9	51.1	39.7
Neural Network (Ours)	59.2	63.2	57.6
Neural Network, MV + BG pre-train (Ours)	**63.9**	**73.8**	**60.0**

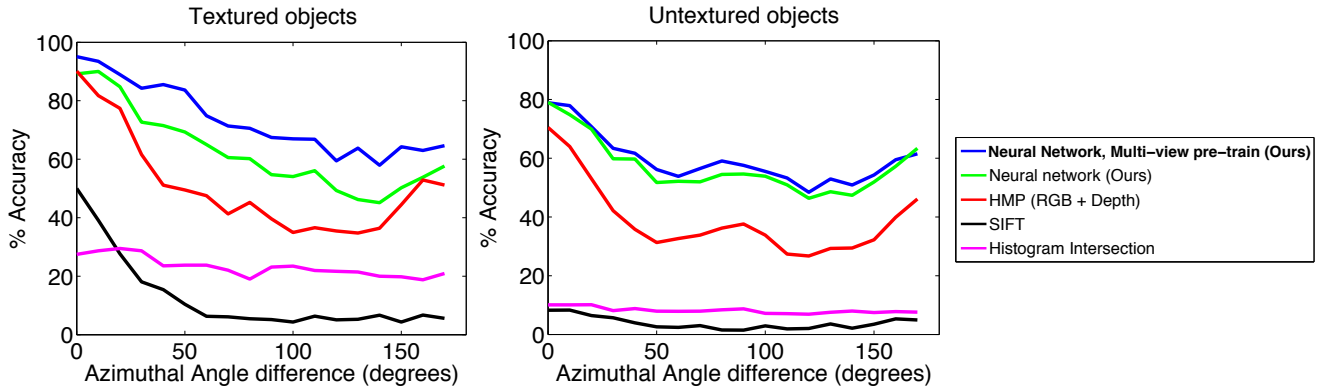

Fig. 3. Average accuracy as a function of the azimuthal angle difference between test examples and the corresponding training example. Note that in all cases there is a 15 degree elevation difference between training and test images. The machine learning methods have a small increase in performance near 180 degrees due to the rotational symmetry of some of the objects.

invariance of different methods in a controlled setting. However, for most applications we would want to be able to detect objects in a full scene, with a real background and occlusions. To measure whether our neural network with multi-view pre-training still gives the best performance in this more realistic setting, we used the RGB-D Scenes Dataset [4]. This dataset has per-frame bounding box annotations, which makes it suitable for our evaluation purposes. We crop the ground-truth bounding box from the scene and then classify the resulting image. The results can be found in Table III. As can be seen, multi-view pre-training improves performance even for objects placed in an indoor setting with a cluttered background and occlusions.

Note that the overall accuracies here are fairly low for all methods. Pre-training on the multi-view dataset teaches our network to be robust to rotations, but we are still not fully robust to lighting changes, occlusions, or other variations. However, our multi-stage pre-training approach is general, and with the proper auxiliary dataset, we should be able to learn these invariances as well. For example, to learn lighting invariance, we would perform multi-stage pre-training with videos of objects undergoing a lighting change.

To make our network robust to recognizing objects under novel backgrounds, the objects used for multi-view pre-training (BigBird [1]) were synthetically placed against random scenes taken from the background category of the RGB-D scenes dataset [4], as explained in Section III-B.

TABLE III

ONE-SHOT INSTANCE RECOGNITION IN A SCENE.

Method	% Accuracy
Random guessing	0.3
BRISK [39]	9.4
ORB [40]	6.6
SURF [8]	10.8
BOLD [41]	7.4
SIFT [7]	12.9
Line-2D [14]	0.9
Color Hist Intersection [42]	9.2
HMP [5]	25.4
NN (Ours)	41.0
NN + MV + BG (Ours)	**44.1**

In Table IV, we demonstrate the advantage of multi-view pre-training against a random background. "NN" is the baseline method with no pre-training. "NN + MV" is the method with multi-view pre-training, and "NN + MV + BG" is the method with multi-view pre-training on synthetic backgrounds. When the test images are part of a real scene, pre-training with a random background increases robustness, improving accuracy by 2.6%. This demonstrates that pre-training on random backgrounds teaches our network

Fig. 4. Examples that were classified correctly using multi-view pre-training but were incorrectly classified using a neural network pre-trained only on ImageNet. Left: Query image. Middle: Guess by neural network pre-trained only on ImageNet (incorrect). Right: Guess with multi-view pre-training (correct).

TABLE IV

DIFFERENT TYPES OF NEURAL NETWORK PRE-TRAINING FOR RECOGNIZING OBJECTS IN A SCENE.

Method	% Accuracy
NN	41.0
NN + MV (Ours)	41.5
NN + MV + BG (Ours)	**44.1**

to be robust to new backgrounds, even when the single-view objects being recognized are trained against a solid background.

We can also analyze the performance as a function of the noise in the bounding box location. To do this, for each bounding box we sampled a scaling factor s and a displacement Δx and Δy. These values are sampled from a distribution that varies with a noise parameter n:

$$s \sim |\mathcal{N}(1, 0.025n)| \quad (1)$$

$$\Delta x \sim \mathcal{N}(0, 2n) \quad (2)$$

$$\Delta y \sim \mathcal{N}(0, 2n) \quad (3)$$

The test crop locations are then scaled by the scaling factor and shifted by Δx and Δy pixels. Examples of noisy images can be seen in Figure 5. Figure 6 shows the accuracy as a function of the noise parameter $n \in [0, 10]$; as seen, our method is robust to noise in the bounding box location and still significantly outperforms the baseline methods. For this experiment, the HMP baseline method only uses RGB information, since the previous experiments of Sections IV-A and IV-C show that adding depth has only a minor effect on performance.

Fig. 5. Left: Crops from a scene used to test robustness to background and occlusions. Right: The same crops with maximum noise added, to test robustness to bounding box noise.

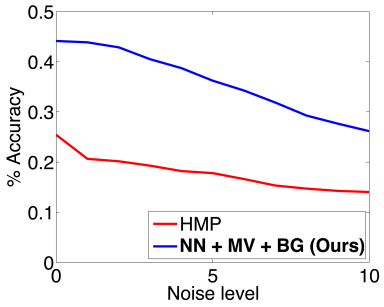

Fig. 6. Instance recognition accuracy as a function of the bounding box noise parameter n.

E. Multiview Pre-training analysis

We can analyze which layers of the neural network are benefiting most from multi-view pre-training. Recall that, for our experiments, we hold the convolutional layers fixed (Section III-C). Table V shows the effect of fixing different layers during multi-view pre-training, evaluated on the RGB-D objects dataset. If we hold the convolutional and both fully connected layers fixed, then we get the baseline performance (equivalent to not using multi-view pre-training). The biggest improvement seems to come from fine-tuning fc6. It seems that multi-view pre-training teaches the fully-connected layers the appropriate relationships between the convolutional features so that the network can be robust to viewpoint changes.

TABLE V

FIXING DIFFERENT LAYERS DURING MULTI-VIEW PRE-TRAINING.

Method	% Accuracy
Baseline (no fine-tuning)	59.2
Fine-tuning just fc7	60.9
Fine-tuning just fc6 + fc7	63.9
Fine-tuning all	65.1

V. Conclusion

We are able to train a neural network to recognize object instances from novel viewpoints given only a single training image of each object. By pre-training our network with multiple views of a separate set of objects, the network learns an increased robustness to viewpoint changes compared to pre-training only on class-level datasets. We show that neural networks with multi-view pre-training outperform previous state-of-the-art methods for instance recognition on both textured and untextured objects.

Thus, multi-view pre-training can make neural networks more robust to recognizing object instances under viewpoint changes. Such a method can be useful for a number of applications, enabling a user to train a robot to recognize a set of objects from just a single image per object. In the future, we hope to use our method to bootstrap a vision system, enabling the system to add more images to its training set over time in a semi-supervised fashion. Multi-view pre-training provides a useful initialization for such an approach, and the network's performance continues to improve as more images are added, thus enabling life-long learning and robust perception.

References

[1] A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *ICRA*, 2014.

[2] Z. Xie, A. Singh, J. Uang, K. S. Narayan, and P. Abbeel, "Multimodal blending for high-accuracy instance recognition," in *IROS*. IEEE, 2013, pp. 2214–2221.

[3] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *ICRA*. IEEE, 2012, pp. 3467–3474.

[4] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *ICRA*. IEEE, 2011, pp. 1817–1824.

[5] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics*. Springer, 2013, pp. 387–402.

[6] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/12, 2012.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*. Springer, 2006, pp. 404–417.

[9] A. Quadros, J. P. Underwood, and B. Douillard, "An occlusion-aware feature for range images," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4428–4435.

[10] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3d range scans taking into account object boundaries," in *ICRA*. IEEE, 2011, pp. 2601–2608.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *IJCV*, vol. 73, no. 3, pp. 263–284, 2007.

[13] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *PAMI*, vol. 15, no. 9, pp. 850–863, 1993.

[14] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *PAMI*, vol. 34, no. 5, pp. 876–888, 2012.

[15] E. Hsiao and M. Hebert, "Occlusion reasoning for object detection under arbitrary viewpoint," in *CVPR*. IEEE, 2012, pp. 3146–3153.

[16] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. IEEE, 2009, pp. 248–255.

[20] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *PAMI*, vol. 28, no. 4, pp. 594–611, 2006.

[21] T. Tommasi and B. Caputo, "The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories," in *BMVC*, no. LIDIAP-CONF-2009-049, 2009.

[22] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "One-shot learning with a hierarchical nonparametric bayesian model," 2010.

[23] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, vol. 172, 2011, p. 2.

[24] H. Mobahi, R. Collobert, and J. Weston, "Deep learning from temporal coherence in video," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 737–744.

[25] S. Becker, "Learning temporally persistent hierarchical representations," *NIPS*, pp. 824–830, 1997.

[26] C. Leistner, M. Godec, S. Schulter, A. Saffari, M. Werlberger, and H. Bischof, "Improving classifiers with unlabeled weakly-related videos," in *CVPR*. IEEE, 2011, pp. 2753–2760.

[27] L. Wiskott and T. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*. IEEE, 2014, pp. 580–587.

[29] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NIPS*, 2014, pp. 3320–3328.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[31] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

[32] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *ECCV*. Springer, 2014, pp. 725–739.

[33] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*. IEEE, 2014, pp. 3286–3293.

[34] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[35] A. K. Mishra and Y. Aloimonos, "Visual segmentation of simple objects for robots," *Robotics: Science and Systems VII*, pp. 1–8, 2012.

[36] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *IJCV*, vol. 43, no. 1, pp. 29–44, 2001.

[37] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 433–449, 1999.

[38] M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller, "A learned feature descriptor for object recognition in rgb-d data," in *ICRA*. IEEE, 2012, pp. 1298–1303.

[39] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*. IEEE, 2011, pp. 2548–2555.

[40] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *ICCV*. IEEE, 2011, pp. 2564–2571.

[41] F. Tombari, A. Franchi, and L. Di, "Bold features to detect texture-less objects," in *ICCV*. IEEE, 2013, pp. 1265–1272.

[42] M. J. Swain and D. H. Ballard, "Color indexing," *IJCV*, vol. 7, no. 1, pp. 11–32, 1991.