Efficient Aspect Object Models Using Pre-trained Convolutional Neural Networks

Eric Wilkinson and Takeshi Takahashi[1]

***Abstract*— We study the problem of object recognition on robotic platforms where large image collections of target objects are unavailable and where new models of previously unseen objects must be added dynamically. This situation is common in robotics, where task related objects can require recognition over multiple viewpoints and training examples are sparse. The proposed framework uses pre-trained convolutional neural network layers to support aspect object models while emphasizing a minimal computational footprint. In this paper, we maintain an object model database consisting of aspect and class descriptors computed from images of target objects at varying view points. By querying the model database we show how to recognize objects with respect to previously seen exemplars. We investigate the effectiveness of different dimensionality reduction techniques for key generation on query efficiency and accuracy. We also demonstrate a working system with a small collection of objects including classes that do not appear in the network's pre-training data set.**

I. INTRODUCTION

The ability to leverage past experience is a critical component of any robotic system that can interact with and solve problems in previously unseen, yet similar environments. For visual systems, this means efficiently finding and retrieving previously stored information based on the current visual cues. While there are many overlapping ways to store object information, when a system wants to interact with the environment the scheme chosen can have a large impact on the types of methods available as well as performance.

We adopt the view that objects can be modeled as a collection of aspects which describe the visual appearance of the object. We define an aspect as a collection of features consistently observed together on an object. For example, the descriptor "*the front of a car*" does not describe an exact viewpoint or pose of the vehicle but rather alludes to a collection of features commonly associated which describe a car's front. Object aspects can be connected together in a graph with edges describing transitions as a result of time, actions, object behaviors or more. As a result, while related to pose, aspects allow for object modeling through both space and time. Although all aspects that appear in this paper consist of visual features, non-visual features could be considered as components of aspects as well.

Aspect objects models can be constructed online through object observation and interaction. For example, consider a scenario where a robot encounters a traffic light. As the traffic light changes state between red, yellow, and green these transitions can be observed by the robot and stored as aspects under a traffic light aspect object model. Associated information such as the predicted time between state changes can be stored in the edges between aspects. Note that this information cannot be captured by pose information, object mesh models, or classifiers alone.

This work proposes an aspect object modeling system which address two important challenges to this kind of system: feature selection and scaling to large collections of objects where many descriptors will be stored. While previous work has examined using features selected by the programmer, we propose using the feature layers from a pre-trained convolutional neural network. Features from this kind of network have been demonstrated to maintain high performance when transferring to new domains where test classes do not appear in the original training data set [2]. For this reason, features from convolutional neural networks serve as ideal candidates for aspect descriptors in the system since target object classes cannot always be known a priori.

When encountering a target object, the aspect modeling system must be able to simultaneously identify if the object belongs to its existing model database or if it represents a novel object. Evaluating a possibly unknown object over all known object aspects quickly becomes infeasible using brute force methods when time is a constraint and the collection of known objects grows large.

Our contributions are as follows:

- We provide a multilevel framework for recognition of object class, object instance, and closest aspect by querying a database of object models.
- We compare different dimensionality reduction techniques for key generation to reduce the query time and storage space while minimizing the effect on retrieval performance from the database.
- We demonstrate the system on a small collection of objects including classes that do not appear in the pre-training data set of the convolutional network.

II. RELATED WORK

The strong performance of convolutional networks for image classification tasks, along with their ability to autonomously construct generic features, has led to a flurry of research surrounding these kinds of vision processing systems. Large convolutional networks such as *AlexNet* train on labeled data sets to learn millions of free parameters [7]. Model complexity at this scale is managed by organizing the network architecturally into convolutional layers that find statistical patterns useful for classification in the preceding layer

[1]The authors are with the Laboratory for Perceptual Robotics at the College of Information and Computer Science, University of Massachusetts Amherst, MA 01003, USA. `ewilkinson@cs.umass.edu`, `ttakahas@cs.umass.edu`

resulting a layer by layer compression of the original image into feature space of increasing sparsity and abstraction [17].

To the best of our knowledge, Babenko et al. are the first group to investigate these CNN models in conjunction with image retrieval from a database [3]. They use PCA compression along with discriminative dimensionality reduction to generate very short codes from a single layer in the CNN and investigate the effect off shrinking code length on retrieval performance. Razavian et al. perform a similar study with great rigor comparing image retrieval performance to 17 other state of the art feature generating methods for images and using a spatial pooling method before PCA to generate their retrieval code [10]. Both groups demonstrate state of the art performance for image retrieval on a number of standard data sets including INRIA Holidays and Oxford5k. Wan et al. investigate the retrieval performance of the first three convolutional layers on a number of data sets using distance metric learning [16]. The proposed framework differs from these more traditional image retrieval applications in that we are focused on identifying object models for interactive robotic systems and perform a multistage query for identification.

Our work shares many conceptual similarities with the classification system of Torresani et al. [15] and the recognition via association work of Maliziewicz et al. [8]. In Torresani et al., an image descriptor is generated using the output of a collection of weakly trained object category classifiers, called classemes, which describe new categories in terms of more generic labels. However, each classeme is a separate trained classifier which must be run independently and the category labels must be chosen ahead of time by the designer. In contrast, we depend on the convolutional network to generate its own generic labels in the intermediate feature layers and rely on the wide range of classes in the pre-training data set such as ILSVRC [11] to diversify labels sufficiently to adequately describe novel objects. Maliziewicz et al. also consider recognition as a problem of association and use distance learning methods over simple hand selected feature sets to recognize image segments in the LabelMe data set. We do not rely on distance learning and focus on identifying different aspects from the same object rather than many different object instances of the same class.

In Schwarz et al. [12] a pre-trained convolutional network is used to generate features from a RGB-D image for object recognition and pose estimation using SVMs. Our work differs from theirs in several key respects. We do not use SVMs for prediction nor do we use features from a depth channel. Instead we focus on time efficiency and design a system that supports aspect models over exact pose predictions. In Sen et al. [13], aspects in an object model are connected together in a graph through actions afforded by the robot and are used to support manipulation planning over a belief space. Their work shows how aspect object models can be used for robotic planning and our system supports these methods.

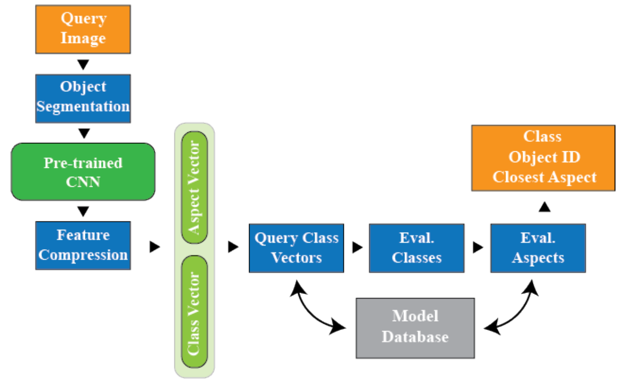

Fig. 1. Overview of the proposed system.

III. Method Overview

An overview of the proposed system is provided in Fig. 1. The input into the system consists of an image containing the target object where the object has been segmented via bounding box from the background. The system does not require a specific segmentation algorithm, only that the target object be reasonably separated from the scene. The image is then scaled to the size accepted by the convolutional network and is passed through the network for feature extraction. We extract two feature vectors from the network, using an intermediate convolutional feature layer as the aspect vector $\vec{\Lambda}$ and the final fully-connected layer as the class vector $\vec{\Upsilon}$. Since $\vec{\Upsilon}$ is used as the database key, we use a compression algorithm to reduce it to a more compact a representation so that queries can be performed efficiently.

The system contains a model database of previously seen objects. The model database is made up of tuples $\{\vec{\Upsilon}_i, \vec{\Lambda}_i, O_k, T_j\}$ where O_k is an object identifier and T_j is a class identifier. The class and object identifiers can either be assigned manually or autonomously deduced by the system. For each unique class identifier T_j in the database, we precompute the statistical distribution of all $\vec{\Upsilon}_i$ stored with that class by fitting a Gaussian model G_{T_j}. This model is used to score the likelihood of input $\vec{\Upsilon}$ belonging to class T_j.

In order to identify the target object using the model database, the input class vector $\vec{\Upsilon}$ is queried against the database so that the k closest tuples are returned as the result set, $\mathbf{R}$. For each unique T_j appearing in $\mathbf{R}$ the system evaluates the likelihood of the input $\vec{\Upsilon}$ belonging to T_j using G_{T_j}. For any T_j in $\mathbf{R}$ whose likelihood is above a user defined threshold, the system compares all $\vec{\Lambda}_i$ from those tuples to the input $\vec{\Lambda}$ to find the most highly correlated aspect. The tuple in $\mathbf{R}$ containing the highest correlated $\vec{\Lambda}_i$ is returned as the output of the system containing the class and object identifier along with the closest aspect.

A. Identifying Novelty

There are two cases in which the system can autonomously identify novelty. The system can identify that the target object belongs to a novel class if the input $\vec{\Upsilon}$ is not likely to belong to any G_{T_j} in $\mathbf{R}$. In this case the input can be stored as a tuple in model database with a new T_j and O_k. The system can also identify if the target object is a novel object belonging to an existing class if the input is likely to belong to a G_{T_j} but no $\vec{\Lambda}_i$ in $\mathbf{R}$ is sufficiently correlated. This time, the system can store the input as a tuple with the same T_j as the most likely class but assign a new object identifier O_k.

IV. Resolving Queries

Given an query image of an target object producing $\{\vec{\Upsilon}^*, \vec{\Lambda}^*\}$, we perform a query of the database using only $\vec{\Upsilon}^*$ and retrieve the k closest class vectors using the L2 norm as the result set $\mathbf{R}$. For each unique class identifier T_j in $\mathbf{R}$, the query is scored using the log-likelihood of the class Gaussian model G_{T_j}. Since the log-likelihood of a Gaussian model increases in magnitude as the number of dimensions increases, we normalize the log-likelihood with dimensionality of $\vec{\Upsilon}$ so that a threshold can be chosen independent of the number of dimensions. Our scoring function is therefore defined as the following.

$$\mathcal{L}_j(\vec{\Upsilon}^*) = -\frac{(\vec{\Upsilon}^* - \bar{\Upsilon}_j)\Sigma_j(\vec{\Upsilon}^* - \bar{\Upsilon}_j)}{|\vec{\Upsilon}|} \tag{1}$$

For each T_j in $\mathbf{R}$ whose likelihood is above a threshold $\mathcal{L}_{thresh}$, we retrieve $\{\vec{\Lambda}_i, O_k\}$ for each result appearing in $\mathbf{R}$ with class T_j. Since only aspects from likely object classes will be evaluated, we can cast a wide net with the $\vec{\Upsilon}$-query while keeping the number of $\vec{\Lambda}$ comparisons relatively low. If there exists no $\mathcal{L}_j(\vec{\Upsilon}^*) > \mathcal{L}_{thresh}$ from $\mathbf{R}$ then we treat the query as a new, unknown class and do not evaluate any aspects.

A. Aspect Comparison

For each $\{\vec{\Lambda}_i, O_k\}$ emerging from the $\vec{\Upsilon}$-query we compare against $\vec{\Lambda}^*$ to find the closest aspect. Since each dimension of $\vec{\Lambda}$ comes from a convolutional layer it has semantic meaning with respect to the bank of convolutional filters contributing to the final response. Due to this, we compare two aspect vectors by Pearson's correlation coefficient [18]

$$\rho(x,y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{2}$$

where N is the dimensionality of $\vec{\Lambda}$. We take the highest ranking $\vec{\Lambda}$ above a threshold ρ_{thresh} as the current aspect. If all correlations are below the threshold then we treat the observation as belonging to a new object of the most likely class T_j.

V. Experiment

For our study we design two separate but related experimental setups. In the first experiment, we seek to find a compression algorithm that preserves class information about images while reducing the storage space and query times. Because the database needs to be large and cover a wide variety of object classes, we sub-sample images from the ILSVRC data set [11] for compression evaluation. In the second experiment, we use the best performing compression algorithm from the first and evaluate accuracy and timing over a small model database.

Both experiential setups use the *AlexNet* [7] implementation available through Caffe [6] as the pre-trained convolutional neural network which was selected because of the wide variety of supporting literature dedicated to this network. The 'fc7' layer from *AlexNet* is used to construct the class descriptors $\vec{\Upsilon}$ since it is the closest layer to the final classification output. The 'pool5' layer is used for aspect descriptors $\vec{\Lambda}$ since it is the highest level convolutional layer. We used an Intel E5-1620 v2 3.70 GHz processor, 32GB RAM, and a GTX 780Ti GPU to obtain timing results.

A. Compression Experimental Setup

Due to the time constraints present in robotic systems, we examine a number of compression techniques to obtain the class vector $\vec{\Upsilon}$. We look for schemes that satisfy the following criteria:

- Queries from a given class should return class vectors belonging to the same class.
- As the model space grows larger, query times over large collections of images must remain low.

We examine three compression algorithms: Iterative PCA [1], Linear Discriminant Analysis (LDA) [5], and Locally Sensitive Hashing (LSH) [14]. For all compression methods we measure distance using the L2 norm, with the exception of LSH where we use the hamming distance. To reduce query times and handle scalability we use a Ball Tree index structure that efficiently performs nearest neighbor searches even for high dimensional data [9].

We sub-sample 50,000 images spanning 1,000 object classes from the ILSVRC training data set, with 50 examples per classes, for the comparison of the compression algorithms. We sub-sample another 250,000 images to investigate the scalability to a much larger database size to measure the performance of the best performing compression methods from the smaller sized database. As a test set we use 5,000 randomly sampled images from the ILSVRC validation set.

B. Compression Comparison

Given query image I with class label T^* we measure the image retrieval performance of a compression algorithm whose result set is $\mathbf{R}$ over all images in the test set.

Since the time required for compression will vary according to both the technique used and the dimensionality of the target space we consider the compression time together with the time for retrieval. The success rate (SR) of the algorithm is measured by the average number of times the query class T^* appeared at least once in the $\mathbf{R}$ where we retrieved the 5 closest vectors. The results for the 50,000 and 250,000 image cases can be found in Table I and Table II respectively. Based on these results, we conclude that Incremental PCA compression with 128 dimensions presents the best compromise between time and success rate for the system and use it in all further results.

TABLE I
PERFORMANCE COMPARISON ON A DATABASE OF 50,000 IMAGES

Algorithm	Dimensions	Time (s)	SR
Inc. PCA	32	0.006	0.516
Inc. PCA	64	0.012	0.549
Inc. PCA	128	0.017	0.556
Inc. PCA	256	0.019	0.555
LDA	32	0.009	0.381
LDA	64	0.014	0.458
LDA	128	0.019	0.511
LDA	256	0.022	0.539
LSH	128	0.032	0.438
LSH	256	0.033	0.434
LSH	512	0.034	0.440
LSH	1024	0.034	0.423
Uncompressed	4096	0.212	0.509

TABLE II
PERFORMANCE COMPARISON ON A DATABASE OF 250,000 IMAGES.

Algorithm	Dimensions	Time (s)	SR
Inc. PCA	128	**0.084**	0.598
LDA	256	0.106	0.560
LSH	512	0.560	**0.622**

C. Accuracy Experimental Setup

To demonstrate that the proposed system is accurate and simple to implement we construct our own data set that selects objects which benefit from aspect models, such as objects with visually distinct sides, and evaluate on object classes both similar and very different from those that appear in the ILSVRC pre-training set. The purpose is not to set a performance benchmark but rather provide illustrative examples of the strengths and weaknesses of the approach.

We created aspect models of 25 different objects over 20 classes consisting of 8 aspects each for a total of 200 images. Each object is entered into the model database by taking a sequence of images at approximately 45 degree rotations of the target object and providing each of the resulting aspects with a label. Note that because of visual symmetries in some objects, aspect labels may not be unique. We include 8 object classes that appear in the ILSVRC 2012 training data set as well as 12 novel classes unseen by the network during training. For the test set, we select 4 rotations randomly for each object in the database and 4 rotations of an additional 10 objects of unknown classes for a total of 140 test images. Sample aspects from each object are provided in Fig. 7. [1]

D. Distribution of Class Vectors

Because we are using novel classes not appearing in the pre-training set it is important to ensure that there exists sufficiently large separation between distinct classes as well as tight inter-class clustering for all T_j. In Fig. 2 we use LDA to embed the $\vec{\Upsilon}$ stored in the database into a two dimensional space showing that novel classes are not scattered and have closely distributed $\vec{\Upsilon}_i$ similar to those that appear in the pre-training set.

[1] Some objects were obtained from the YCB object set [4].

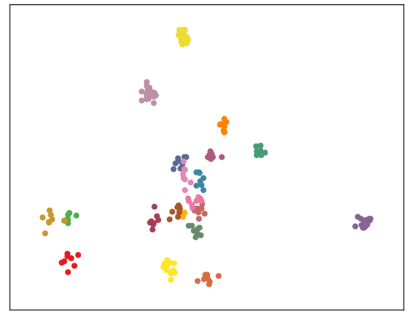

Fig. 2. A two dimensional embedding using LDA of all $\vec{\Upsilon}$ used to populate the model database. Classes are represented by color. Note large separation between distinct classes as well as tight inter-class clustering for all categories in the experiment including those that do not appear in the ILSVRC pre-training set.

E. Query Evaluation

Sample queries are provided in Fig. 3 and Fig. 4 which show results that highlight the strengths of the system. The thresholds selected for evaluation were $\mathcal{L}_{thresh} = -2.8$ (equivalent to probability threshold $P_{thresh} = 0.061$) and $\rho_{thresh} = 0.45$ which were chosen empirically. In Fig. 5 we show an example query involving a humanoid robot, uBot-6, identified as the same class as its previous generation cousin uBot-5. In this case, the system identifies that the uBot-6 aspect does not belong to the uBot-5 object allowing the robot to be entered into the database dynamically as a new object sharing the same class identifier.

We evaluate the accuracy of the system on the test set and provide the success rate in Table III and the average timings in Table IV. The success rate for a $\vec{\Upsilon}$-query is measured by how often the query class is in $\mathbf{R}$ and above $\mathcal{L}_{thresh}$. The success rate for aspect comparison is measured by how often the query aspect label is the highest correlated aspect from the model database above ρ_{thresh} from likely classes in $\mathbf{R}$. We also provide the success rate when considering the top-2 above threshold aspects. Finally, the success rate of correctly identifying the object ID is counted as the number of times the highest ranking aspect is from the correct object regardless of whether the aspect labels match.

Table III shows that the system is effective on the test set while Table IV demonstrates efficiency in time. The difference between the top-1 and top-2 $\vec{\Lambda}$-comparisons is a result of the query aspect being between two aspects in the model database which produces two vectors with high correlation in the result set. This suggests that models should be entered into the database at positions where $\vec{\Lambda}_i$ are sufficiently uncorrelated from previous object aspects already entered rather than at regular rotational intervals. In Fig. 6 we show some example failures from the test set.

VI. CONCLUSION

We present a novel, object modeling system that uses codes from a pre-trained convolutional neural network to

TABLE III

Measure	Performance
$\vec{\Upsilon}$-query	0.971
Object ID	0.960
$\vec{\Lambda}$-comparison	0.787
Top-2 $\vec{\Lambda}$-comparison	0.858

TABLE IV

Component	Time (ms)
AlexNet	52.1
128 Inc. PCA	0.4
$\vec{\Upsilon}$-query	0.3
$\vec{\Lambda}$-comparison	0.6
Total	**53.4**

support recognition of class, object instance, and closest aspect. We have investigated a number of compression schemes to find a low dimensional representation that remains accurate while reducing query times when searching over object classes. We explore the efficacy of the system over a small, noisy data set even for classes that do not appear in the pre-training set as well as providing a number of illustrative examples for the strengths and weaknesses of the approach. Although we use *AlexNet* for our evaluation, we believe the basic design of the proposed system should work for other convolutional neural network architectures trained over a similarly large data set and by selecting appropriate feature layers for $\vec{\Upsilon}$ and $\vec{\Lambda}$.

Our work would be of interest to experimental setups where training individual classifiers / pose regression models for each object in the task is impractical and novel objects must be treated equally. In future work, we plan on investigating the use of aspect object models for high level task planning using information stored in the edges between aspects.

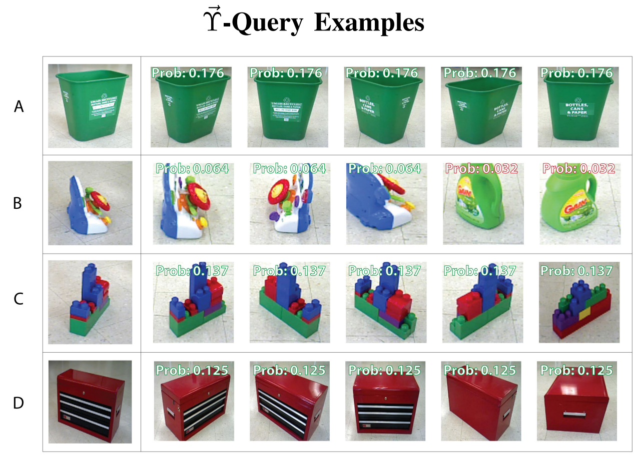

Fig. 3. A sample of illustrative $\vec{\Upsilon}$-query results. The query image is shown on the left and the closest resulting $\vec{\Upsilon}$ vectors appear in sorted order to the right with the class identifier probability displayed in each image. Green highlighted text indicates values above the threshold and red indicates below. Query A shows the typical test case and result set. Query B shows a toy whose features change dramatically under rotation resulting in objects from another class appearing in the $\vec{\Upsilon}$-query. The model G_{T_j} for the detergent bottle scores the query image as unlikely and is not considered further. In Query C, another object under the same class identifier appears in the result set. Query D shows that the closest $\vec{\Upsilon}$ in the result set is not necessarily the closest aspect and demonstrate robustness under changes in illumination.

References

[1] Matej Artač, Matjaž Jogan, and Aleš Leonardis. "Incremental PCA for on-line visual learning and recognition". In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 3. IEEE. 2002, pp. 781–784.

[2] Hossein Azizpour et al. "From generic to specific deep representations for visual recognition". In: *arXiv preprint arXiv:1406.5774* (2014).

[3] Artem Babenko et al. "Neural codes for image retrieval". In: *Computer Vision–ECCV 2014*. Springer, 2014, pp. 584–599.

[4] Berk Çalli et al. "Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols". In: *CoRR* abs/1502.03143 (2015). URL: `http://arxiv.org/abs/1502.03143`.

[5] Alan Julian Izenman. "Linear discriminant analysis". In: *Modern Multivariate Statistical Techniques*. Springer, 2008, pp. 237–280.

[6] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[8] Tomasz Malisiewicz, Alexei Efros, et al. "Recognition by association via learning per-exemplar distances". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[9] Stephen Malvern Omohundro. *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

[10] Ali Sharif Razavian et al. "Visual Instance Retrieval with Deep Convolutional Networks". In: *arXiv preprint arXiv:1412.6574* (2014).

[11] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* (2015). DOI: `10.1007/s11263-015-0816-y`.

[12] Max Schwarz, Hannes Schulz, and Sven Behnke. "RGB-D Object Recognition and Pose Estimation based on Pre-trained Convolutional Neural Network Features". In: *Proc. IEEE International Conference on Robotics and Automation (ICRA2015)*. 2015.

[13] Shiraj Sen and Roderic Grupen. "Manipulation planning using model-based belief dynamics". In: *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS*

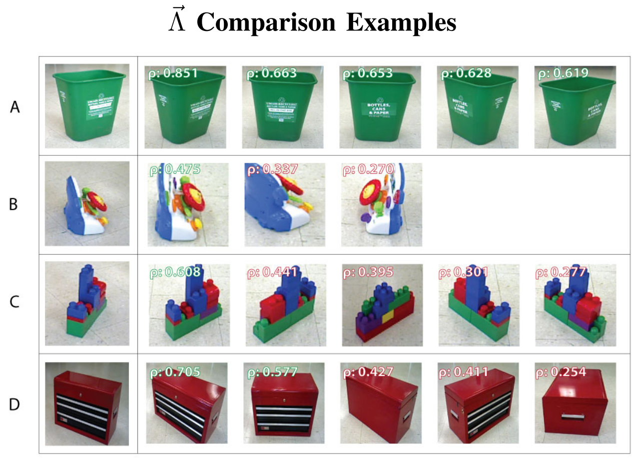

Fig. 4. A sample of illustrative aspect comparisons resulting from the $\vec{\Upsilon}$-query in Fig. 3. The query image is shown on the left and resulting aspect comparisons are shown in sorted order to the right. Green highlighted text indicates values above the threshold and red indicates below. Query A shows the system is able to correctly distinguish between the two sides of the recycle bin from the different textual patterns. In query B, only three aspects are considered since the detergent bottle was below $\mathcal{L}_{thresh}$. Query C shows that the correct toy block is identified between two choices. In query D, we show that the closest $\vec{\Upsilon}$ may not be the closest $\vec{\Lambda}$.

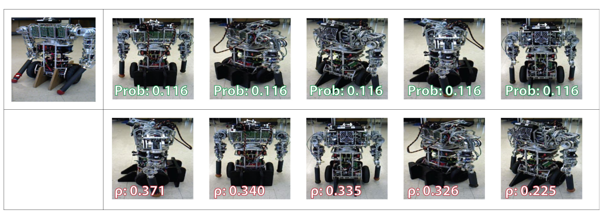

Fig. 5. A sample query involving a the humanoid robot uBot-6 which does not exist in the model database. The results set includes entries from the similar, previous generation uBot-5. The model G_{T_j} for the uBot-5 class scores the query $\vec{\Upsilon}^*$ above threshold but no aspect comparison is above the aspect threshold. In this case, the system identifies the query as a new object belonging to the same class as uBot-5.

International Conference on. IEEE. 2013, pp. 403–409.

[14] Malcolm Slaney and Michael Casey. "Locality-sensitive hashing for finding nearest neighbors [lecture notes]". In: *Signal Processing Magazine, IEEE* 25.2 (2008), pp. 128–131.

[15] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. "Efficient object category recognition using classemes". In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 776–789.

[16] Ji Wan et al. "Deep Learning for Content-Based Image Retrieval: A Comprehensive Study". In: *Proceedings of the ACM International Conference on Multimedia*. ACM. 2014, pp. 157–166.

[17] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014*. Springer, 2014, pp. 818–833.

Fig. 6. Sample queries from the experimental test set that resulted in failure. In Query A, we show an aspect comparison failure with an ARcube where the correct box orientation is returned but the alphanumeric labels are incorrect. This is likely because distinguishing between characters did not help on the ILSVRC data set and therefore individual characters are not encoded in $\vec{\Lambda}$. Query B shows an aspect comparison failure case involving a deformable object, where the backpack has been emptied and the straps appear in different orientations. Query C shows one of the few class identification failures where a jar of peanut butter previous unseen by the model database is incorrectly identified as a class of cup.

Fig. 7. Sample aspects from each object entered into the model database for the experimental setup. Note that objects are not cleanly separated from the background, are subject to changes in illumination and resolution, and cover a wide diversity of object classes.

[18] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. CRC Press, 1999.