Transfer of Sparse Coding Representations And Object Classifiers Across Heterogeneous Robots*

Zsolt Kira[1]

Abstract— **This paper examines the problem of transfer learning in the context of object recognition in a heterogeneous robot team. We specifically look at the case where robots individually learn object classifiers and must then transfer the resulting learned knowledge to another robot. Recent trends in computer vision and robotics have moved towards feature representation learning, where the underlying feature representation used in classification is learned in a data-driven way. This poses a problem to knowledge transfer, as the underlying representations learned by different robots will differ significantly. In this paper, we present several hypotheses with regard to knowledge transfer in such a scenario, specifically that 1) the transfer of knowledge will be most effective if it involves not just the classifier itself, but the learned feature representations themselves, 2) this is not a problem because given similar scenes and objects, some methods such as sparse coding are able to learn representations that can be successfully used by another robot, and 3) a codebook encoding scheme such as Fisher vectors will result in a smaller reduction in accuracy after transfer even if the receiving robot uses its own learned feature representation. Finally, we contribute an alignment procedure and demonstrate that it can serve to facilitate knowledge transfer even when the underlying feature representations are independently learned by each robot and codebook methods are not used. We test all three of the hypotheses and the alignment procedure on a real-world dataset consisting of two robots viewing the same 12 objects using cameras with differing characteristics.**

I. INTRODUCTION

There has been a great deal of progress made in the last decade in terms of perception for robotics. Some of the problems that have been successfully addressed recently include the detection of pedestrians [3][13] and other objects [16], SLAM and mapping, and obstacle avoidance. One of the most successful application that requires many of these techniques is in self-driving cars [6]. Despite these successes in single-robot systems, however, there has been little research into distributed perception systems, where individual robots learn, interact with each other and with humans, and share knowledge. As successes in single-robot perception filter through to larger multi-robot and swarm systems, it is unlikely that the current methods of learning where a single classifier is learned *a-priori* and deployed across a group of robots will be realistic. Rather, each individual robot will learn using its own sensor data, and any knowledge sharing would involve the transfer of learned information.

*This research has been supported by the Georgia Tech Research Institute Robotics strategic initiative.

[1]Zsolt Kira is a research scientist with the Georgia Tech Research Institute, Atlanta, GA 30318 USA. zkira@gatech.edu

While this latter scenario is realistic, there are several challenges to transferring such classifiers across heterogeneous robots. If hand-designed features such as HOG [2] or SIFT [17] are used as the underlying feature representation, there are methods for transferring the resulting classifiers [23] and potentially adapting them as well. However, while the design of these features over decades of research has made them robust, even in these cases there may be a significant loss of performance for some object categories when the receiving robot uses transferred classifiers [11]. An even larger complication that has arisen is the recent trend in computer vision and robotics of learning of underlying feature representations themselves in a data-driven way. This includes unsupervised feature learning through techniques such as sparse coding and filtering [15][19], or more sophisticated deep learning methods that build hierarchical feature representations [9]. While such techniques have recently yielded state-of-the-art recognition results [14][5], the transfer of learned classifiers across different robots is hampered by the fact that even the underlying feature representation will be different.

A key open question is therefore whether object models learned using these techniques can be transferred between robots with different sensing characteristics. In this paper, we focus on this question when using a particular type of unsupervised feature learning, namely sparse coding [15]. We hypothesize that classifier transfer will exhibit similar characteristics as the hand-designed ones under certain circumstances. Specifically, we investigate the transfer of different pieces of information, including the learned feature representation, the clusters of the feature vectors if a codebook Bag of Words method is used [22], and the learned classifier itself.

We investigate several hypotheses with regard to this knowledge transfer, namely: 1) that the transfer of knowledge will be most successful if it includes not just the classifier itself, but the learned feature representations themselves, 2) that this is not a problem because given similar scenes and objects, filters learned on one robot will be able to be successfully used by another robot, and 3) that an encoding scheme such as Bag of Words will result in a smaller reduction in accuracy if the resulting clusters are also transferred. These hypotheses are informed by our previous research [10][12][11], although in those works we did not use a learned feature representation or state of the art computer vision techniques. We test all three of these hypotheses on a real-world dataset consisting of two robots viewing the same 12 objects using cameras with differing characteristics.

Given these results, we then develop a method to align

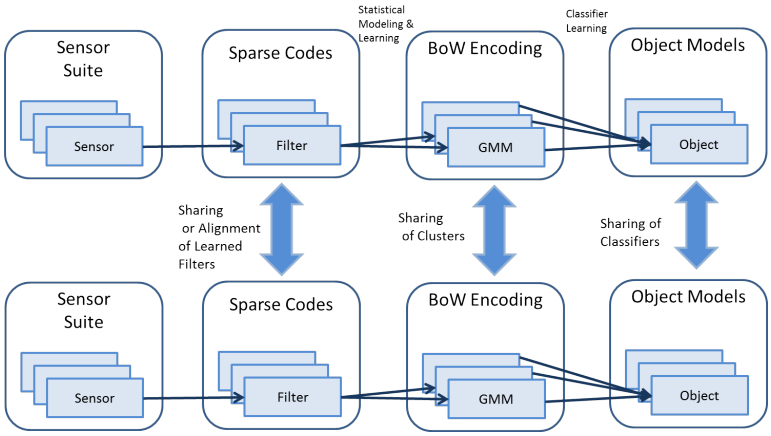

Fig. 1. Overview of object recognition pipeline and multiple levels of knowledge transfer possible. Raw sensor data is filtered through sparse codes, which represent kernels that are learned using unsupervised data. The resulting filters are convolved with the image patches, resulting in a set of output maps. These output maps are down-scale and form a set of feature vectors. In order to summarize this set of vectors, a codebook technique is used to statistically represent the set of vectors based on distances to learned clusters. The resulting feature vector is then fed to a classifier. Knowledge sharing can occur at all three levels, including the sharing of filters or kernels, the sharing of clusters learned in the Bag of Words method, and/or sharing of the final classifier.

the feature representations across robots and show that this alignment procedure is able to facilitate knowledge transfer even when the underlying feature representations are learned by each robot individually and a codebook method is not used. This alignment procedure enables the scenario described above, where each robot learns its own representation and classifiers but can still leverage learning performed by other robots without having to learn one global feature representation (which would be suboptimal), to adopt feature representations learned by another robot, or to transfer clusters that are part of the codebook techniques. We show that the resulting alignment technique significantly improves accuracy after transfer of the learned classifiers.

Section II will describe related and previous work in the areas of transfer learning, domain adaptation, and knowledge sharing across heterogeneous robots. We will then describe in Section III the learning process used for object recognition, including unsupervised learning to derive sparse coding filters that can then be applied to test images and used for classification. We will also describe the different levels of sharing that can occur in this framework. Section IV will outline the robot experiments we perform and their results, and we conclude in Section V with a discussion and description of future work.

II. RELATED WORK

While an exhaustive summary of work within object recognition is not possible, we first highlight standard and state-of-the-art techniques that have recently been developed and specifically those techniques related to the ideas in this paper. Successful object classification pipelines have typically included the extraction of hand-designed features (such as HOG [2], SIFT[17], etc.), an intermediate step such as Bag of Words that statistically models the distribution of these features using clustering [8], and a final classifier step. Recently, however, a trend of automatically learning feature representations has become popular for numerous reasons [7][26]. Besides the fact that they have been able to beat existing methods to achieve state-of-the-art performance [27], feature learning is attractive due to its applicability to many different modalities. Instead of having to hand-design features for every modality, feature learning does so in a data-driven manner. This has been demonstrated, for example, in pedestrian detection using both images and stereo disparity maps [13].

While transfer learning has not been significantly studied in the context of feature learning techniques used for object recognition, it has received significant attention in the past several years in the machine learning [21], reinforcement learning [24], and computer vision communities (e.g. [25][4]). One example is zero or one-shot learning that can be achieved through the transfer of priors and mid-level attributes while learning new object categories [4]. In these cases, however, knowledge transfer occurs across different object categories as opposed to different sources of data in robotics. [20] provides some investigation into domain adaptation in the context of feature learning, however the main focus there is to adapt the dictionaries learned on top of features, not features that differ themselves.

Few examples exist in the case of transfer learning applied to perception-related robotics problems. Some computer vision results, such as those of [18], have used robots with multiple cameras and transferred SVM classifiers as in our work. Another recent work looks at transfer learning for place categorization [1], focusing on the problem of deciding what to transfer. However, these works focus either on homogeneous robot teams with the same sensors or transfer of perceptual information across categories, hence differing from our focus on heterogeneity. This paper focuses specifically on the interplay between feature representation learning and resulting implications for transfer learning across heterogeneous robots.

III. LEARNING AND TRANSFERRING OBJECT CLASSIFIERS USING SPARSE REPRESENTATIONS

A. Learning a Feature Representation

In this section, we will describe the architecture used to both learn and transfer object classifiers. Figure 1 shows the overall pipeline. Raw sensor data, in our case images, are fed into an unsupervised sparse coding optimization to learn the underlying feature representation. The representation itself is a set of basis kernels, learned through sparse coding, that are convolved with the images to produce a set of output images. These images are subsampled to reduce their size, vectorized, and concatenated to produce a single feature vector. We then either classify the resulting feature vector, or use an extended Bag of Words approach (Fisher vectors [22]) to produce a Gaussian Mixture Model codebook and produce a new encoded feature vector. These two steps (feature learning and codebook methods) are critical in the sharing of classifiers, as we have shown that mid-level features that abstract pixel-level information are critical to enable the transfer of learned classifiers across heterogeneous robots [10][12][11]. We then frame the problem within a supervised learning framework, where classifiers are learned mapping the resulting feature vector to provided object labels.

The first step of the pipeline produces a basis set of vectors through sparse coding [15]. Specifically, we take a large sample of unlabeled image patches and optimize a basis set that minimizes the reconstruction error while maintaining sparsity in the weight vectors representing the contribution of the basis vectors to reconstruct each patch. Suppose that there is a basis set of vectors $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_k\} \in \mathbb{R}^d$ and a weight vector $\mathbf{s} \in \mathbb{R}^k$ such that the basis set is overcomplete ($k > d$). Given an input image patch that is vectorized (i.e. rows are concatenated to form a single vector) $I \in \mathbb{R}^k$, the goal is to reconstruct the image patch using a linear weighted combination of the basis set, i.e. $I \approx \sum \mathbf{b_j} s_j$. In order to learn such a basis set of vectors using unsupervised data, an optimization is performed such that a set of unlabeled image patches $\mathbf{I} = \{I_1, ..., I_n\}$ are reconstructed in such a manner that each of the weight vectors $\mathbf{S} = \{\mathbf{s}_1, ..., \mathbf{s}_n\}$ corresponding to each image patch is sparse. Specifically, the objective function is [15]:

$$minimize_{B,S} \quad \frac{1}{2\sigma^2}\|\mathbf{I} - \mathbf{BS}\|^{\mathbf{2}}_{\mathbf{F}} + \beta \sum \phi(\mathbf{S_{i,j}})$$

$$subject\,to \quad \sum B_{i,j}^2 \leq c, \forall_j 1, ..., k$$

Here, the $\frac{1}{2\sigma^2}$ term comes from the assumption of a zero-mean Gaussian reconstruction error, β is a constant, and ϕ is a penalty function on the weight vectors which enforces sparsity. This can be, for example, an L_1 penalty on the norm of the vectors. This latter objective term ensures that most of the values in the weight vectors are close to zero with only a few key non-zeros, ensuring regularization, and is motivated by neurological findings. The minimization constraint (second line) enforces a norm constraint using a constant c to prevent to prevent bases whose weight vectors approach all zeros during the optimization. Note that both the bases and the weight vector must be simultaneously optimized, and efficient methods have been developed which alternate between these two problems [15]. The resulting bases have been shown to produce strong results for tasks such as object detection.

B. Learning a Classifier using the Learned Features

Given the bases learned in the previous section, there are several ways to use the resulting representation to classify objects. One method is to hold the bases $\mathbf{B}$ constant for a given test image patch and perform one round of the optimization to derive the weight vector $\mathbf{s}$ [15]. This weight vector represents the strength of contributing bases, and can be used to describe objects. However, this requires an iteration of optimization for each image, resulting in slow run-times.

Instead of the approach above, we instead take the learned bases and use them as filters that are convolved with the image patch. This is possible because, as described elsewhere and as will be shown in the results section, the learned filters typically resemble Gabor-like edges as well as other patterns that can be used as kernels for a convolution operation with an image. We convolve each kernel with the image patch, resulting in *k* output maps. These outputs maps are then subsampled and vectorized to create a high-dimensional feature vector that can be fed to a classifier.

While it is possible to stop there and classify objects, it is hypothesized that transferring the resulting classifier across heterogeneous robots will result in diminished accuracy. This is because even if the same kernels are used by both robots, the convolution operation may result in output maps that are spatially or distributionally different for each robot. Since the output maps are simply vectorized, differences in a few output maps will result in a large difference in the resulting feature vector. As a result, we instead encode the feature vector using Fisher vectors, a soft clustering Bag of Words method [22]. This creates a new feature vector representing the distribution of output maps over the clustered codebook, and is hypothesized to be more robust to transfer.

C. Aligning Sparse Codes across Robots

While we hypothesize that the best performance can be obtained by transferring both the feature representation (filters) as well as the classifiers, there are some drawbacks to this approach. Namely, the receiving robot must perform convolution using additional sets of filters which may become prohibitive as the system scales with more robots, filters, and object categories. As a result, we propose an alternate approach. We observe that the sparse codes learned by different robots share some similarity since their characteristics are partially determined by the scenes that they are trained on. Based on this observation, we perform an alignment procedure that maps the two sets of sparse codes. We do this by computing the correlation coefficient for each pair of codes (one from each robot), and obtain the best match for each filter on the sending robot. Specifically,

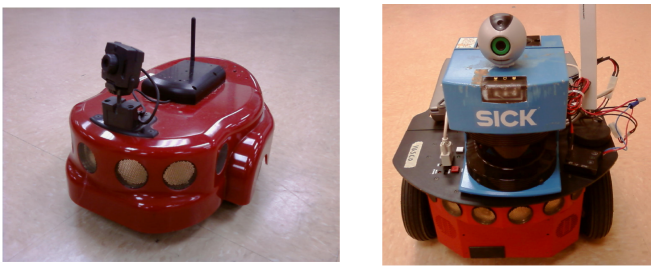

Fig. 2. Robots used in the experiments: A Mobile Robots Amigobot with a wireless camera and a Pioneer 2DX robot with a Quickcam Express web camera. The image characteristics of these two cameras differ significantly, in addition to varying in resolution.

Fig. 3. The twelve real-world objects classified, representing a wide variety of shapes, textures, and colors.

we compute the correlation function r, where A and B are matrices of the same size representing the two codes:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

The resulting feature vectors are then realigned based on this mapping. No other calculations are necessary and hence the codes used by the robot sending its classifiers do not need to be communicated or convolved with images by the receiving robot.

IV. ROBOT EXPERIMENTS

A. Platforms and Hypotheses

In this section, we conduct real-world experiments using data from two heterogeneous robots to test our hypotheses. Specifically, we used a Mobile Robots Amigobot with a wireless camera and a Pioneer 2DX robot with a Quickcam Express web camera (shown in Figure 2). The Amigobot used 640x480 resolution images while the Pioneer used 320x240 resolution images, another source of heterogeneity. We perform object classification using these robots for twelve different objects, which are shown in Figure 3. The three hypotheses that were tested are:

1) **Hypothesis 1**: The transfer of knowledge should include not just the classifier itself, but the learned feature representations themselves
2) **Hypothesis 2:** Given similar scenes and objects, sparse coding methods are able to learn filters that can be successfully used by another robot
3) **Hypothesis 3**: An encoding scheme such as Fisher vectors will result in a smaller reduction in accuracy if the resulting clusters are also transferred

B. Baseline Learning Results

One baseline condition for these experiments is the usual learning pipeline where each individual robot trains and tests using its own data. In order to classify the twelve objects, we obtained approximately 100 instances of each object and divided the data into a training and testing set along a 90/10 split. To learn the sparse codes, we also extracted 10k random 14x14 patches from the set of training images (which included both background and potentially the objects themselves). We used these patches to learn 64 bases in 100 iterations of the optimization, where each iteration used a random 1k subset of the unlabeled patches. Figure 4 shows the resulting sparse codes learned for both robots. As expected, the codes are different for each robot, as they are learned in a data-driven way using images from each robot. However, note that there are some commonalities and filter pairs across robots that are similar, lending credence to our hypothesis that the scene characteristics determine a large part of these filters and hence using one set of these over another across robots may not make a difference in terms of accuracy. This observation informed our alignment method described in Section III-C which enables robots to share classifiers without having to adopt each other's feature representations.

Given these filters, we then treat each one as a kernel for a convolution and stride it across an image patch containing the object to derive an output map, representing the response of the filter on the image patch. These response maps are then down-sampled to 14x14 size and vectorized to create a 12544-dimensional vector (there are 64 14x14 output maps). This vector can be used directly with a classifier, in our case a linear SVM. However, one of our hypotheses is that while the resulting classifier can be transferred to obtain better-than-random performance, there will be a significant degradation in performance when doing so. This is because the high-dimensional vector uses the convolution outputs directly to classify the objects, and the statistics of the output maps will vary across robots.

As a result, we instead encode the 64 feature vectors in a distributional manner, specifically using Fisher vectors [22]. This representation can be seen as a Gaussian Mixture Model (GMM) variant of Bag of Words methods, where instead of using a fixed determination of which cluster a particular vector belongs to we use distances to Gaussian clusters learned in the GMM. In this paper we use thirty clusters, resulting in a 11760-dimensional encoded feature vector.

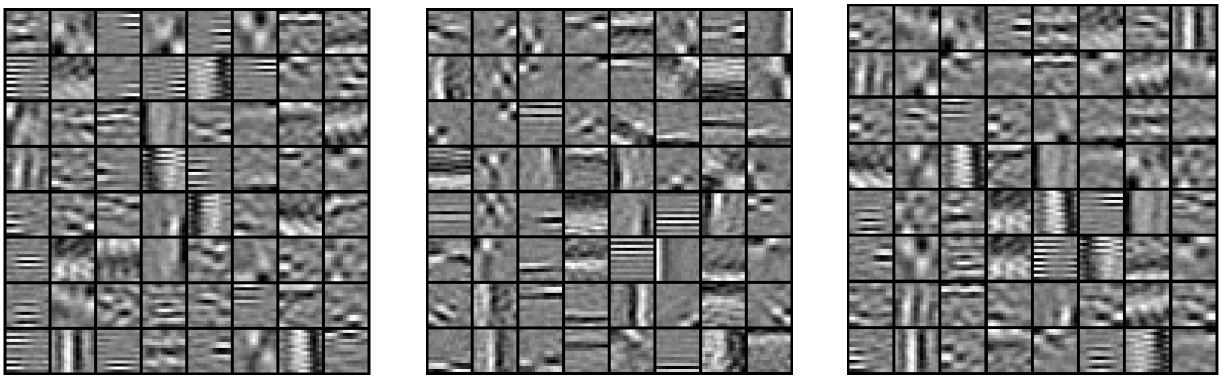

Fig. 4. Set of 64 sparse codes (or bases) learned during an unsupervised optimization process, given data from the Amigobot (left) and Pioneer 2DX (middle). These codes are then used as kernels for a convolution operation to derive output feature maps on image patches. The rightmost subfigure shows the Amigobot filters after the alignment process, described in Section III-C, showing that there is a great deal of similarity in the Pioneer and Amigobot filters after alignment.

C. Experimental Conditions and Results

Given the baseline above, we then vary the condition in order to test our hypotheses. We tested the first hypothesis by comparing the object recognition accuracy on the Amigobot data when using training data from the same robot versus when directly obtaining classifiers from the Pioneer robot. We test the transfer condition both when the Amigobot uses its own sparse codes (learned on data from the Amigobot camera), and when using sparse codes obtained from the Pioneer. Table I shows mean and standard deviation of the overall classification accuracy, calculated as the ratio of true positives plus true negatives and the total number of instances. The columns differentiate between using sparse codes learned by the testing robot itself ("Own Filters") versus obtaining sparse codes from another robot ("Transferred Filters"). The rows differentiate whether the robot learned the classifier using its own data ("Own Classifier") versus receiving a classifier from the other robot ("Transferred Classifier"), and if it was transferred whether alignment was used ("Transferred Classifier with Alignment"). These results did not use Fisher vectors: i.e. no codebook was used and the feature output maps were fed to the classifier directly.

	Own Filters		**Transferred Filters**	
	Mean	**Stdev**	**Mean**	**Stdev**
Own Classifier	90.8	6.5	91.9	5.4
Transferred Classifier	51.0	3.6	83.7	11.3
Transferred Classifier with Alignment	71.9	18.7	-	-

TABLE I

AVERAGE AND STANDARD DEVIATION CLASSIFICATION ACCURACY OVER THE TWELVE OBJECTS UNDER SEVERAL CONDITIONS.

As can be seen, if the robot uses its own filters ("Own Filters" column), the transfer of classifiers from the other robot is not successful and the performance is close to chance. However, if the receiving robot adopts the filters used by the transferring robot, the accuracy of classification is significantly better than chance. However, there is approximately a 8.9% degradation in accuracy. These results confirm our first two hypotheses. Namely, if the filters are transferred as well, there can be a bootstrapping that is achieved by transferring classifiers. The second hypothesis is confirmed as well, showing that if the receiving robot uses filters it received, its classification accuracy is not degraded significantly, i.e. "Own Filters" performance is close to the "Transferred Filters" performance in the non-transfer case.

Finally, the last row shows the result with the receiving robot using its own filters, but uses the transferred classifier after the alignment process. Figure 4 (right-most filter set) shows the Amigobot filters after alignment, matching nicely with the Pioneer filters (middle) after the alignment. As can be seen from the table of results, while there is a significant degradation across some objects, the average accuracy is significantly better than chance and represents a bootstrapping process. Note that the last two columns of this row are blank because no alignment is needed if the receiving robot uses transferred filters as well.

While promising, the above results show that similar to transferring SIFT-based classifiers (where the feature representation is not learned) [11], a significant degradation of performance occurs. More importantly, when the receiving robot uses its own learned representation performance is at chance levels, and even with alignment it is lower than desired. In our third hypothesis, we address this by showing that codebook-based methods that encode feature vectors can be used both to significantly increase overall accuracy across all conditions, as well as alleviate some of the reduction in accuracy when classifiers are transferred. As described in Section III-B, we create a 30-cluster GMM and encode the feature vectors as a new Fisher vector that is then used

	Own Filters		Transferred Filters	
	Mean	**Stdev**	**Mean**	**Stdev**
Own Classifier	98.3	1.6	99.8	0.7
Transferred Classifier	79.3	16.7	87.9	11.2
Transferred Classifier with Alignment	74.7	15.8	-	-

TABLE II

AVERAGE AND STANDARD DEVIATION CLASSIFICATION ACCURACY OVER THE TWELVE OBJECTS UNDER SEVERAL CONDITIONS, WHEN USING FISHER VECTORS TO CODEBOOK THE SPARSE CODING-BASED FEATURE VECTOR.

in classification. Table II shows the accuracy results. As can be seen, this method significantly increases the overall accuracy of recognition across all conditions. Furthermore, as before, transferred filters can be used by the receiving robot successfully using its own classifier ("Transferred Filters" column, "Own Classifier" row) as well as when receiving a classifier ("Transferred Filters" column, "Transferred Classifier" row). Furthermore, our final hypothesis is proven correct as evidenced by the fact that there is significantly less degradation in the "Own Filters" and "Transferred Classifier" cell. Without codebook methods, this accuracy was approximately chance but now it can achieve an average accuracy of approximately 79.3%, with very high performance for some object classes. One way to explain this is that the codebook method works over the set of output maps and describes their distribution in terms of membership to learned clusters, thereby removing dependency on the actual order of the filters. In other words, unlike before, the alignment procedure is not necessary and there is no statistical difference between the alignment condition and no alignment condition. Note that the Fisher vectors technique did not reduce the amount of degradation when the receiving uses transferred filters when comparing the "Own Classifier" and "Transferred Classifier" conditions (last column, first two rows). There is a degradation of 11.9% in the average accuracy (from 99.8% to 87.9%).

V. DISCUSSION AND CONCLUSIONS

In this paper, we have explored the interplay between recent advances in computer vision and robotics, namely unsupervised feature learning, and the transfer of perceptual knowledge across heterogeneous robots. Using a state of the art pipeline that leverages sparse codes to classify objects, we have validated three hypotheses. First, we have shown that as long as the underlying feature representation is shared in addition to the classifier, that the resulting classifiers can be successfully transferred from one robot to another despite perceptual differences. Second, we have shown that the features learned in an unsupervised manner can still be successfully used even if they are learned on a different robot. This is because they capture essential shapes or filters that describe the general scene and objects, and as long as this is shared they can still be used. We have shown that more sophisticated codebook methods can even further leverage transferred knowledge and results not just in increased accuracy across the board, but also more successful knowledge transfer as demonstrated by a smaller amount of degradation in accuracy when the receiving robot uses its own filters. Finally, leveraging the observation that the filters learned across robots share similarity, we proposed an alignment process that can enable the robots to share classifiers without having to communicate or compute new filters beyond the ones learned by each robot. We demonstrated that this alignment can significantly increase the success of sharing when codebook methods are not used.

The results presented here demonstrate the first investigation of these methods in a real-robot setting involving multiple sensors with different characteristics. Several key questions remain, however, and will be investigated in future work. First, it is not clear whether better techniques or alignment algorithms can be used to overcome the decrease in accuracy for some objects that occurred when using our proposed alignment procedure or Fisher vectors. For example, techniques such as domain adaptation [23] may be applicable in order to adapt classifiers learned on one source distribution to a new target distribution. One limitation of existing approaches in that field is that data from both source and target domains are needed *a-priori*, meaning that annotated instances from particular objects must be provided from each robot's data. While we have previously leveraged such a shared context to align other fixed feature representations [12], the need for a shared context is limiting in real-world applications. However, there may be additional solutions that only require unlabeled data from the robots in the team, thereby relaxing this constraint. Such methods offer great potential for future work, with the ultimate goal of allowing each robot to individually learn while also incorporating information and learned knowledge from its peers when available.

REFERENCES

[1] Gabriele Costante, Thomas A. Ciarfuglia, Paolo Valigi, and Elisa Ricci. A transfer learning approach for multi-cue semantic place recognition. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, page 2122–2129. IEEE, 2013.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. volume 1, page 886–893, 2005.

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (99):1–1, 2011.

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006.

[5] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. *arXiv preprint arXiv:1303.5778*, 2013.

[6] Erico Guizzo. How google's self-driving car works. *IEEE Spectrum Online, October*, 18, 2011.

[7] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, page 2146–2153. IEEE, 2009.

[8] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 1, pages 604–610 Vol. 1, October 2005.

[9] Koray Kavukcuoglu, Pierre Sermanet, Y.-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning convolutional feature hierarchies for visual recognition. *Advances in neural information processing systems*, 23:14, 2010.

[10] Zsolt Kira. Transferring embodied concepts between perceptually heterogeneous robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, page 4650–4656, 2009.

[11] Zsolt Kira. *Communication and alignment of grounded symbolic knowledge among heterogeneous robots*. PhD thesis, Georgia Institute of Technology, 2010.

[12] Zsolt Kira. Inter-robot transfer learning for perceptual classification. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, page 13–20, 2010.

[13] Zsolt Kira, Raia Hadsell, Garbis Salgian, and Supun Samarasekera. Long-range pedestrian detection using stereo and a cascade of convolutional network classifiers. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, page 2396–2403, 2012.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, page 1106–1114, 2012.

[15] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, page 801–808, 2006.

[16] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1683–1698, 2008.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, page 1150–1157, 1999.

[18] Jie Luo, Andrzej Pronobis, and Barbara Caputo. Svm-based transfer of visual knowledge across robotic platforms. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS'07)*. Applied Computer Science Group, Bielefeld University, Germany, 2007.

[19] Jiquan Ngiam, Zhenghao Chen, Sonia A. Bhaskar, Pang W. Koh, and Andrew Ng. Sparse filtering. In *Advances in Neural Information Processing Systems*, page 1125–1133, 2011.

[20] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 692–699. IEEE, 2013.

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[22] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, page 1–8, 2007.

[23] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, number 6314 in Lecture Notes in Computer Science, pages 213–226. Springer Berlin Heidelberg, January 2010.

[24] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10:1633–1685, 2009.

[25] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, 2007.

[26] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, page 1794–1801. IEEE, 2009.

[27] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013.