Viewpoint Selection for Visual Failure Detection

Akanksha Saran[1], Branka Lakic[2], Srinjoy Majumdar[3], Juergen Hess[4] and Scott Niekum[1]

***Abstract*— The visual difference between outcomes in many robotics tasks is often subtle, such as the tip of a screw being *near* a hole versus *in* the hole. Furthermore, these small differences are often only observable from certain viewpoints or may even require information from multiple viewpoints to fully verify. We introduce and compare three approaches to selecting viewpoints for verifying successful execution of tasks: (1) a random forest-based method that discovers highly informative fine-grained visual features, (2) SVM models trained on features extracted from pre-trained convolutional neural networks, and (3) an active, hybrid approach that uses the above methods for two-stage multi-viewpoint classification. These approaches are experimentally validated on an IKEA furniture assembly task and a quadrotor surveillance domain.**

I. INTRODUCTION

In recent years, great leaps have been made toward learning algorithms that produce robust robotic control policies that generalize across differing situations [1], [2], [3], [4]. However, while these methods provide good behavior in expectation, they typically do little to verify correct behavior or task completion in any given situation. Future robotics applications will require behaviors that are not only reliable in expectation, but also verifiable: safety-critical tasks must be completed with a high degree of assurance; robots that work with populations that rely on them, such as the disabled or elderly, must be dependable; manufacturing robots that chain together many behaviors, such as in an assembly task, must check their work at each step, or else face multiplicative error rates as the number of steps increase. Given the ubiquity of visual sensors in robotics, in this work we focus on determining the outcomes of tasks, or any latent visual concept, via image data.

Unfortunately, visual task outcome classification is difficult for several reasons. First, only a small segment of any given image may actually contain useful discriminative information. If there are only small differences between classes, the inter-class variance may be overwhelmed by the intra-class variance. Furthermore, these small differences may only be observable from certain viewpoints or may even require information from multiple viewpoints to fully verify. For example, a robot changing a tire, assembling a piece of furniture, or boiling a cup of tea must pay attention to small, specific parts of each object of interest; it should be able to precisely determine the alignment of the tire with the car, ensure that the furniture pieces have been properly attached to each other, or discern the subtle changes at the surface of the water that are associated with boiling. All of these features constitute only a small fraction of the available information in the field of view (Fig. 1) and their visibility can be highly viewpoint dependent (Fig. 2). These observations are illustrated in Fig. 1 and Fig. 2 for the furniture assembly task. In this paper, we propose to use viewpoint selection techniques based on fine-grained image classification to address both of these issues concurrently.

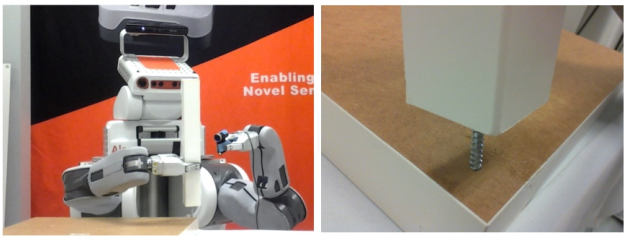


Fig. 1. The PR2 using a hand-held camera to view the table assembly task (left) and an example view of a successful attempt (right).

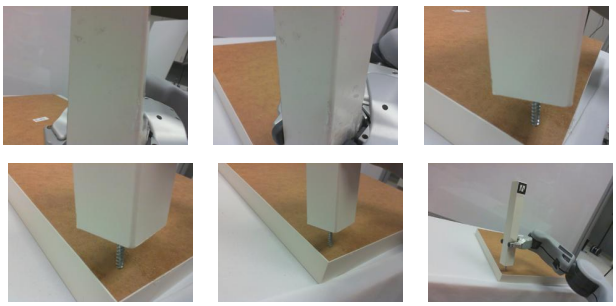


Fig. 2. Images captured from 6 of the 19 viewpoints after a successful attempt at table leg insertion. Bottom-right: the "generic" full-scene view. The images are captured at the end of the task execution by the robot.

Our contributions are threefold: First, we introduce a fine-grained viewpoint selection and failure detection algorithm based on discriminative random forests, and empirically show that semantically meaningful fine-grained details can be discovered from a small number of examples in diverse robotics tasks. However, this approach requires the rough alignment of images from each viewpoint. Second, we overcome this limitation by introducing an alternate viewpoint selection approach that uses SVMs trained on features from a pre-trained convolutional neural network (CNN). We demonstrate that, somewhat surprisingly, the SVM+CNN approach performs nearly as well as the random forest, despite the fact that it does not require alignment, the features are not

[1]Akanksha Saran and Scott Niekum are with the Department of Computer Science, University of Texas at Austin, Austin, TX 78712, USA. {asaran,sniekum}@cs.utexas.edu

[2]Branka Lakic is with the Department of Computer Science, Duke University, Durham, NC 27708, USA. blakic@cs.duke.edu

[3]Srinjoy Majumdar is with the Electrical and Computer Engineering Department, University of Texas at Austin, Austin, TX 78712, USA. srinjoy.majumdar@utexas.edu

[4]Juergen Hess is with the Bosch Research and Technology Center, Palo Alto, CA 94304, USA. Juergen.Hess2@us.bosch.com

optimized for the task, and that it is not an explicitly fine-grained method. Third, we propose a novel active, two-stage sequential viewpoint selection algorithm that can improve performance in domains in which the optimal viewpoint is context-dependent. We experimentally validate the proposed algorithms in two substantially different domains—screw insertion in a furniture assembly task and car break-in detection using a quadrotor.

II. RELATED WORK

A. Failure Detection

To the best of our knowledge, our presented method is the first that performs outcome classification (success or failure) for general robotics tasks from real world 2D image data. However, there is a rich history of failure detection methods in robotics that leverage domain knowledge for specific tasks such as navigation [5], [6] or that detect sensor errors rather than behavioral failures [7], [8].

In robotic manipulation, one straightforward approach simply looks for deviations outside of confidence bounds relative to a nominal trajectory [9]. Nearest-neighbor classifiers over state information have been used to sequence controllers [4], [10], which can be thought of as combined outcome classification and error recovery. Other approaches treat failure detection as a time-series classification problem. One data mining approach searches for discriminative sub-signals called *shapelets* [11], but is highly computationally intensive even for relatively small amounts of data. Spatio-Temporal Hierarchical Matching Pursuit [12] has been used to assess grasp stability and recognize objects from tactile data by learning sparse hierarchical features. Worcester et al. [13] use visual depth sensors for online error detection and correction during an assembly process by multiple mobile robots. Our work focuses on using monocular 2D image data to identify failures post task execution. Nguyen et al. [14] use simple image features, i.e. dimensionality reduction on concatenated image patches, from 2D images and 3D registered point cloud data to learn classifiers which determine where in 3D space manipulation behaviors for a task will succeed. Previous methods have also combined visual and force/torque sensing for error recovery during manipulation [15]. While these approaches work with simple visual features, such as surfaces of polyhedral objects, we work with real images and general-purpose image features to perform error detection.

B. Viewpoint Selection

Estimating optimal placement of cameras using machine vision has been proposed for tasks such as inspection [16], object recognition [17], human activity recognition [18], control and monitoring [19]. However, these methods typically require a significant amount of prior knowledge of objects to optimize the camera viewing angle, such as 3D object models and pre-determined areas of interest. Leifman et al. [20] proposed a viewpoint selection algorithm to detect regions of interest on surfaces i.e. parts of an object that would appeal to humans in general, whereas we look for regions of interest to discriminate between task outcomes. They focus on 3D mesh surfaces, while we consider a general task space which could contain multiple objects as part of the scene. The "next best view" problem [21], [22] has also been investigated to select informative viewpoints for performing 3D reconstruction from 2D images.

In the most relevant work to ours, Kootstra et al. [23] rotate a camera around an object to find stable keypoints that help to construct a model of an object. These keyframes are then leveraged to perform active viewpoint selection during the object recognition phase. Similarly, Govender et al. [24] use an active learning approach to collect images with unique, high-information features until an object is recognized with confidence above a set threshold. However, these approaches are designed for object recognition rather than task outcome classification and focus on objects with large inter-class differences, rather than fine-grained classification.

Visual servoing techniques have been used for manipulation and grasping [25], [26], [27], object tracking [27] and navigation tasks for miniature mobile robots [28]. These techniques use simple visual features, such as fixed patterns or position based features of objects, and can move the visual sensors anywhere in space to obtain these features. We sacrifice adaptability at the level of moving a camera anywhere in free space for gaining generality in the visual space. We care about the general information content obtained from real images of fixed viewpoints to detect task failures.

III. APPROACH

Given visual data to observe a task from multiple viewpoints, we propose to select the most suitable of those viewpoints to determine whether the task was carried out successfully or not. This viewpoint selection and subsequent failure detection occurs in a task independent manner. The only assumption however is that the viewpoints are at fixed locations with respect to the task setup and the viewpoint selection and subsequent failure detection occurs at the end of task execution. In this section, we discuss two image classification approaches for visually detecting task execution failures – a fine grained visual classification technique (Section III-A) and another simple image classification using deep neural network features (Section III-B). Section III-C describes how we extend these techniques to determine the most suitable viewpoint for failure detection. We use one of two approaches to select the optimal viewpoint- a static approach where the most suitable viewpoint is determined apriori during training, or an active approach where the optimal viewpoint depends on the visual information available from a particular execution of the task.

A. Fine-Grained Image Classification

To classify images, we use a fine-grained Random Forest (RF) approach first introduced by Yao et al. [29]. Traditionally, decision trees employ a weak classifier at each node that operates on a global set of features. To capture the fine-grained nature of this problem, Yao et al. use *discriminative decision trees*, which employ a strong classifier (a support

vector machine/SVM in this case) at each node that operates on features of an image sub-patch, also augmenting the state space at each node with the decision values of all the parents that it descended from.

However, this approach requires a search over a large, dense sampling space—all possible image patches of arbitrary width and height at all possible locations. To make this search tractable, the algorithm randomly selects a number of patches at each node and trains a classifier on each of them, finally selecting the best classification outcome using an information gain criterion [30].

One advantageous property of random forests is the ability to gain interpretable insight into classification results. Following Yao et al., we derive heat maps from the random forest, as shown in Figure 4, that visualize the relative importance of each pixel in classification. Each pixel weight can be calculated by summing the SVM class probability for a given class (the heatmaps for both classes tend to be nearly identical, so we only show one) for all image patches that include that pixel.

B. Image Classification with Pre-trained CNN Features

The recent success of convolutional neural networks (CNNs) has shown them to be a state-of-the art technique for image classification. CNNs provide an end-to-end learning framework from images to output labels without engineering features by hand, and have also been shown to work well for transfer learning, by pre-training on very large datasets and then fine-tuning them for smaller novel datasets [31]. If datasets are not large enough for fine-tuning (roughly 200 images per class), then features can be extracted from pre-trained networks (fully-connected layer activations before the soft-max classification, for example), without any fine-tuning. These features can then be used as input to classifiers such as SVMs to work with the smaller dataset. We leverage the richness of CNN feature layers in this manner by directly using the weights from existing models like AlexNet [32], VGGNet[33], GoogLeNet[34] and ResNet[35] on our small-sized datasets.

C. Optimal Viewpoint Selection

We can discriminate among task outcomes from any viewpoint using image classification techniques. However, our goal is to select a viewpoint that maximizes the discriminative power of the classifier. Rather than simply using a standard viewpoint (such as a robot's overhead view of a manipulation task), we introduce two methods for intelligently choosing optimized viewpoints for a given task.

For both methods, we assume that the robot is able to collect outcome-labeled training examples from n different viewpoints for each of m trials or executions. These viewpoints can be selected in any way, but ideally should be chosen in a manner that does not require prior knowledge about the task except only an approximate knowledge of the positions of any object(s) of interests. This data can then be used to train a classifier separately for each viewpoint, which can further be used to select an optimized viewpoint in two different ways—(1) a static strategy that chooses the best viewpoint, on average, for classification, and (2) a two-stage active strategy that chooses the best information-gathering viewpoint, which in turn allows the robot to predict which viewpoint will provide the most accurate classification.

Algorithm 1: COMPUTE_IOV finds the information optimizing view at training time, given multiple sets of corresponding training images from each viewpoint.

Input: n : number of viewpoints
m : number of trials or executions
$I = \{f_{ij}$ s.t. $i \in [1,m], j \in [1,n]\}$: set of image features
$L = \{l_{ij}$ s.t. $i \in [1,m], j \in [1,n]\}$: set of binary labels (correct/incorrect task classification by a failure detection algorithm)
$trainSVM(in, out)$: a function to train a linear SVM given input features (in) and output labels (out)
Output: iov : Index of information optimized view;
$S = \{S_{ij}$ s.t. $i \in [1,n], j \in [1,n]\}$: set of trained SVM models

```
for i ← 1 to n do
    x ← {f_1i, f_2i, ..., f_mi}
    y ← {l_1i, l_2i, ..., l_mi}
    errors(i) ← 0
    for j ← 1 to n do
        (model, error) ← trainSVM(x, y)
        S(i, j) ← model
        errors(i) ← errors(i) + error
    errors(i) ← errors(i)/n
iov ← arg min_i errors(i)
return iov, S
```

1) Static Viewpoint Selection: The simplest method for choosing a viewpoint for classification is to always pick the same viewpoint, regardless of the features of the test image—the viewpoint with the highest classification accuracy, on average. We train different types of classifiers for each viewpoint separately: a random forest classifier [36] and SVMs over different CNN features. Based on ablation studies, we use deep features from the activations of fc6/fc7 layers for AlexNet [32], fc6 layer for VGGNet[33], pool5 layer for GoogLeNet[34] and pool5 layer for ResNet[35] architectures pre-trained on ImageNet[37]. Then, k-fold cross-validation accuracy for each of these classifiers is computed. Finally, the viewpoints are ranked based on this cross-validated accuracy, and the top viewpoint is chosen for use in future test-cases.

2) Two-stage Active Viewpoint Selection: Rather than choosing the best static viewpoint on average, in many problems it may be beneficial to adaptively choose a viewpoint based on visual features of the particular trial. However, this still requires an initial image to be captured for analysis. Thus, we propose a two-stage active viewpoint selection method that first chooses a static information-optimized viewpoint (IOV) that is the best, on average, at predicting which optimal classification viewpoint (OCV) will most successfully classify any given execution. For example, in

Algorithm 2: COMPUTE_OUTCOME uses the information-optimizing viewpoint (IOV) to predict the best outcome classification viewpoint (OCV) at test time. OCV is used to classify the task outcome as success or failure.

Input: n : number of viewpoints
iov : index of information optimized viewpoint
$I = \{I_1, I_2, \ldots, I_n\}$: image features for a single execution corresponding to all viewpoints;
$S = \{S_{1,iov}, S_{2,iov}, \ldots, S_{n,iov}\}$: set of SVMs trained over accuracies of each viewpoint;
$testSVM(model, in)$: a function to test an SVM model over a given input (in). It returns distances to margins for the correctly ($m_{correct}$) and incorrectly ($m_{incorrect}$) classified class.
$failureDetection(v, f)$: a function to detect failures for a given viewpoint v and corresponding image features f for a trial.
Output: ocv : index of outcome classification view
$outcome$: predicted task outcome

```
for i ← 1 to n do
    [m_correct, m_incorrect] ← testSVM(S_{i,iov}, I_iov)
    score(i) ← m_correct
ocv ← arg max_i score(i)
[outcome] ← failureDetection(ocv, I_ocv)
return outcome
```

the quadrotor surveillance domain for car break-in detection, the IOV may try to get a view from the front of the car to determine which side of the car the person is on; then based on features of that image, an OCV can be chosen that matches the correct side of the car, in order to get the best possible view of the person's hand. This image can then be used to classify whether the person is trying to break into the car (i.e. their hand is on the handle) or if they are simply standing next to the car.

Since choosing an IOV is not an inherently fine-grained problem (i.e. global features may be useful in determining the OCV from the IOV), we obtain image features for each viewpoint by feeding each image into a deep neural network—AlexNet [32] pre-trained on Imagenet [37]—and record the activations of layer fc7. For each of the n candidate IOVs, these deep features are computed and then used as inputs to train n SVM classifiers [38], [39], leading to a set of n^2 classifiers overall. Essentially, each viewpoint is a candidate for being chosen as the IOV based on the training data, and whether a certain viewpoint does get selected as the IOV depends on how well the classification accuracy of all viewpoints can be predicted given training images from only that candidate IOV.

Given images from a candidate IOV (c), we use their features to train n SVMs ($S_{1,c}, S_{2,c}, \ldots S_{n,c}$), one associated with each viewpoint v. We train $S_{v,c}$ over the output labels of a failure detection algorithm for v. Thus, supervision is provided with binary labels that correspond to whether or not the failure detection algorithm's output matched ground truth labels. The candidate IOV c which provides the least error averaged over its corresponding n SVMs is finally chosen as the IOV iov. At test time, we only require an image from a single viewpoint—the IOV—which is used to predict how accurate task outcome prediction might be from other viewpoints. The viewpoint predicted to be the most accurate by the IOV image is then chosen as OCV. The image from OCV is then used to determine the task outcome through a failure detection algorithm. Note that for every image at test time, the OCV could be different based on the content of the image, allowing for adaptive selection based on information gathered from the IOV. While this requires images from two viewpoints to be captured (as opposed to one in the static case), we hypothesize that this adaptivity will significantly increase classification accuracy in certain domains. This approach of selecting the IOV and OCV is described in Algorithm 1 and Algorithm 2 respectively.

IV. EVALUATION AND DISCUSSION

A. Experimental Setup

Here we describe the details about our implementation and the experimental setup for our datasets. For the random forest, we use 100 trees, a maximum tree depth of 10, and a minimum of 11 patches per node. We use the linear SVM implementation made available as part of the Statistical and Machine Learning toolbox in Matlab [38], [39]. Deep features are extracted from neural network models (AlexNet [32], VGGNet [33], GoogLeNet [34], ResNet [35]) pre-trained on ImageNet in Caffe [40]. The runtime of all the methods is near real-time during testing, as it simply computes the output from trained classifiers or performs a forward pass for computing deep features for one or two viewpoints. Training time is considerably longer (order of hours) and varies according to the total number of viewpoints. However, training is assumed to be performed offline.

1) IKEA Table Assembly Task: The first domain is an IKEA table assembly task: the leg of the table has a screw protruding from one end, which a PR2 mobile manipulator attempts to insert into a pre-drilled hole in the table base, as shown in Fig. 1. A webcam is attached to the robot's left gripper, which can be moved to capture images from viewpoints of varying distances and angles relative to the table leg. Before capturing images in each trial, the PR2 attempts to insert the table leg via a hand-coded trajectory that moves the right gripper and table leg toward a hand-coded goal position. To provide variance in starting conditions, the gripper holding the table leg begins each trial at a different location.

An AR tag, a type of visual fiducial, is used to determine the rough position of the table leg. A set of 19 candidate viewpoints are then taken from three spheres with varying radii—one centered on the AR tag, one on the middle of the leg, and one on the bottom of the leg—in order to provide many different views with as little a priori domain knowledge as possible. Additionally a "generic view" was collected that contains an entire side-view of the table and leg; this can be used as a baseline to test whether a generic view

is sufficient for this task, or if viewpoint selection can be beneficial. We collected data on 39 executions of the task (39 images per viewpoint), hand-labeling successes and failures. The success and failure labels were nearly equally distributed (19 successes, 20 failures). Fig. 2 shows some of the images from different viewpoints in a single execution.

Finally, we preprocessed images for the fine grained RF by automatically aligning them pixel-wise in each view to account for small differences in inverse kinematics solutions and inaccurate servoing. As a result, all the images in each view were cropped to the intersection of their overlapping areas after alignment. This alignment and cropping was done automatically with intensity based image registration methods. However, for the SVM classifiers, deep features were computed from the raw unaligned camera images as the features are robust to translational variations.

2) IKEA Table Assembly with Obstacles: This task is similar to the task above, but we introduce obstacles in the task space obstructing the discriminating view of the screw and the pre-drilled hole on the table base from some of the viewpoints. The obstacles are placed at different locations blocking different viewpoints in different trials. We use two robots to perform this task, each with a Kinova Jaco 2 arm (6-DOF). One robot moves the table leg to the table base, while the other robot's arm captures images after the execution is complete. We capture images along 4 circles (two radii and two heights), each with 5 viewpoints equally spaced out, totalling to 20 viewpoints. The table leg insertion was attempted 40 times (40 images per viewpoint) with 22 successes and 18 failures. The obstacles placed in the scene were a subset of the YCB dataset [41].

3) Quadrotor Surveillance Task: In this experiment, a quadrotor equipped with a GoPro camera performs a surveillance task, in which it captures images at five locations around a car to detect whether a person is simply standing next to the car or trying to "break in" by pulling on the car door handle. One viewpoint was directly in front of the car, while the others were taken from the front corners and sides of the car, as shown in Fig. 3. In each iteration of the experiment, the person in the scene can either be standing on the left or the right side of the car.

We performed the experiment 68 times (68 images per viewpoint), with the person standing on each side of the car 34 times. In 17/34 images on both left and right, the person is opening the door and is just standing next to it in other 17 images. For this dataset, we cropped images manually to contain only the car with the person and some small amount of background to offset the large imprecision in the position of the quadrotor across different runs. This cropping could be performed automatically, but is not the focus of this work.

B. Results

1) IKEA Table Assembly Task: For the table leg dataset without obstacles, we first compare the cross-validated accuracy of the best static view and the generic view (view 19) that encompasses the larger scene, taken from distance. We use both random forest and SVM classifiers to determine how informative each viewpoint is to detect the class outcome. The best static view for each method detects failures better than the generic view. The results are shown in Table I.

Fig. 3. Five viewpoints in the quadrotor surveillance domain (View 1 starting from the top left image and going clockwise for views 2-5).

TABLE I

10-FOLD CROSS VALIDATION RESULTS FOR FAILURE DETECTION USING A STATIC VIEWPOINT ON THE IKEA TABLE ASSEMBLY DATASET.

View	Random Forest (%)	AlexNet (fc7) + SVM (%)	VGG (fc6) + SVM (%)	GoogLeNet (pool5) + SVM (%)	ResNet (pool5) + SVM (%)
1	63.9	57.3	65.4	64.0	64.6
2	66.7	65.0	67.9	60.4	62.6
3	69.4	65.0	64.9	61.0	67.6
4	66.7	56.9	69.7	56.9	62.8
5	66.7	61.9	60.0	59.3	44.9
6	69.4	62.5	60.6	60.1	55.3
7	61.1	64.2	57.6	61.8	63.0
8	66.7	75.2	65.6	62.3	66.6
9	69.4	57.7	59.2	72.5	67.2
10	72.2	68.8	76.4	79.3	70.7
11	66.7	61.5	71.8	74.5	71.6
12	63.9	65.6	65.3	61.8	66.6
13	91.7	70.0	76.0	85.4	80.8
14	72.2	58.1	66.1	61.8	64.1
15	80.6	69.8	72.4	78.6	72.8
16	66.7	64.2	63.8	57.6	53.7
17	**91.7**	70.0	81.9	**90.3**	75.7
18	77.8	74.2	77.2	80.8	78.3
19	63.9	59.4	55.1	47.5	63.5

We report results with 10-fold cross validation to retain a larger training data due to the small size of the dataset. 10-fold cross-validated accuracy for the best viewpoint (view 17) was 91.7% using the random forest classifier and 90.3% using GoogLeNet pool5 features. GoogLeNet's pool5 features perform nearly as well as the random forest classifier, despite not being tuned for the task and not having aligned images. This demonstrates the surprising effectiveness of these features in capturing fine-grained differences that has been recently observed in literature [42].

Examples of the best and worst view (apart from the

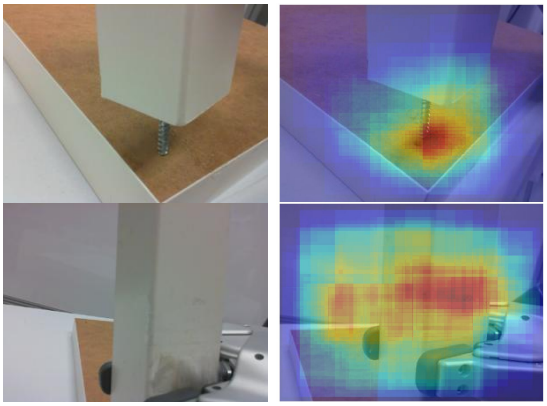
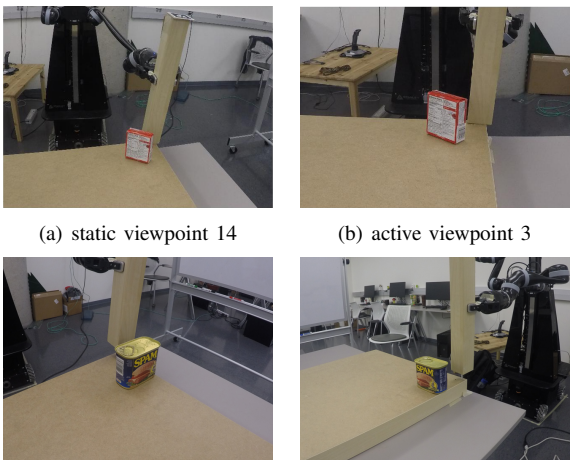


Fig. 4. Random forest classifier's best view (top row - view 17) and worst view excluding the generic view (bottom row - view 7) along with their overlaid heat maps for a successful execution.

generic view) for the random forest approach are presented in Fig. 4. Overall, the VGGNet fc6 features, AlexNet fc7 features and ResNet pool5 features do not perform as well as the RF or GoogLeNet features. GoogLeNet uses inception modules for dimensionality reduction in an embedding space to represent information in a dense, compressed form. Our results lends support towards the inception modules being more effective at retaining spatial relationships in a deeper and sparse network versus the loss of spatial specificity with pooling layers of VGGNet and AlexNet. The average accuracy over 18 views (excluding the generic view) is 71.3% (RF), 64.9% (AlexNet+SVM), 68.1% (VGGNet+SVM), 68.2% (GoogLeNet+SVM) and 65.9% (ResNet+SVM) which is still considerably lower than the respective accuracies from the best view. It should be noted that our approach has very little prior knowledge of the task and still manages to find a view with a high accuracy.



(a) static viewpoint 14 (b) active viewpoint 3

(c) static viewpoint 10 (d) active viewpoint 1

Fig. 5. Trials from the table assembly task with obstacles obstructing some viewpoints. (a), (b) are images from the same trial where the static viewpoint fails to detect the failure but the active viewpoint succeeds. (c), (d) provide another such example.

2) IKEA Table Assembly with obstacles: This is a more challenging dataset than the previous one, due to objects obstructing different viewpoints in different trials. The captured images also have more background clutter than the simple assembly dataset above. Consequently we see a drop in performance as can be seen in Table II. The highest accuracy among all methods for detecting failures goes down to 69.5% (GoogLeNet features for view 16), averaging over 10 runs of 10-fold cross validation. The random forest classifier gives the best static viewpoint which is different but has a comparable performance (69.0% for view 13). We see similar trends in comparison with Table I, where performance of CNN features and the random forest was comparable, with GoogleNet features doing the best. With ablative analysis, we found AlexNet fc6 features performing better than fc7 features for this dataset, whereas fc6 features of VGGNet still perform better than fc7 features. Random forest performs the best on average over all viewpoints (60.6%) compared to AlexNet+SVM (49.8%), VGGNet+SVM (50.7%), GoogleNet+SVM (53.3%) and ResNet+SVM (51.7%), demonstrating the the fine grained method's discriminative capability on real-world images.

We also test the active viewpoint selection algorithm for this dataset. We use GoogLeNet features as the image features and random forest classifier as the failure detection method for Algorithm 1 and 2. We choose the random forest classifier to select the OCV as it performs the best on average. Averaging across ten runs of 10-fold cross validation, we find that the active viewpoint selection approach gives a 3.5% improvement in detecting failures over the static approach. In Fig. 5, we show an example of two executions where the static viewpoint incorrectly classifies the task outcome, whereas the active viewpoint gives the correct classification. In Fig. 5(a), the best static viewpoint has an obstruction to the view of the hole and nail. Since the obstacle is not at a fixed position for every execution, the static viewpoint cannot account for dynamic elements such as the moving obstacle. However, the active viewpoint in Fig. 5(b) is one where the view is not obstructed by the obstacle. Similarly, on using GoogLeNet image features and GoogLeNet+SVM as the failure detection algorithm (which gave the highest performance for a static viewpoint), the active viewpoint improves performance by 11% over the static viewpoint, averaged across ten runs of 10-fold cross-validation. We show example views for this combination in Fig. 5 (c) and (d). This shows that GoogLeNet+SVM classifier is better able to determine task outcomes by latching on to discriminative regions of the image, in turn making active viewpoint selection work better. Active viewpoint selection shows promise in detecting failures even in the presence of obstacles and clutter that are part of real-world environments.

3) Quadrotor Surveillance Task: For this dataset, we test both static and active viewpoint selection approaches. For each of the 5 viewpoints shown in Fig. 3, we use 68 images. We use 5-fold cross validation to generate 5 partitions of the dataset (more image data available per viewpoint) and repeat this for several runs of the experiments. We report cross-validated results averaged across 10 runs of the experiment. The results for the static viewpoint selection approach using

TABLE II

10-FOLD CROSS VALIDATION FOR FAILURE DETECTION USING A STATIC VIEWPOINT ON THE TABLE ASSEMBLY DATASET WITH OBSTACLES.

View	Random Forest (%)	AlexNet (fc6) + SVM (%)	VGG (fc6) + SVM (%)	GoogLeNet (pool5) + SVM (%)	ResNet (pool5) + SVM (%)
1	52.0	44.3	46.8	63.8	40.8
2	67.0	52.3	44.8	61.0	60.8
3	65.5	48.0	49.0	45.3	39.8
4	62.8	58.5	41.3	43.5	49.5
5	61.8	**66.8**	51.3	45.0	58.0
6	54.3	44.5	43.3	51.3	**62.3**
7	60.5	50.3	45.5	53.0	51.3
8	65.5	55.0	44.5	51.0	47.0
9	55.5	43.5	63.5	61.3	50.5
10	45.8	54.8	47.3	64.0	50.5
11	56.8	45.3	46.3	56.8	61.0
12	68.5	62.3	52.5	53.3	50.5
13	**69.0**	45.8	**68.8**	44.0	43.5
14	66.5	46.3	53.8	55.0	54.0
15	63.0	47.0	58.3	50.0	51.0
16	53.5	43.3	46.3	**69.5**	49.8
17	57.8	45.3	50.5	47.0	61.3
18	67.3	45.5	63.0	43.8	59.8
19	67.3	47.8	54.0	54.3	48.5
20	51.0	50.5	43.5	58.3	45.3

both random forests and SVMs trained on deep features are shown in Table III. The highest accuracy with the RF method is 77.1% (view 2) which is matched by the performance of the GoogLeNet features (77.9%). Notably, the SVM+CNN approach (for all 4 architectures) performs better than chance for every viewpoint. This is a somewhat surprising result since many viewpoints appear to contain very little information about the classification of a given trial.

TABLE III

5-FOLD CROSS VALIDATION RESULTS FOR FAILURE DETECTION USING A STATIC VIEWPOINT ON THE QUADCOPTER DATASET.

View	Random Forest (%)	AlexNet (fc7) + SVM (%)	VGG (fc6) + SVM (%)	GoogLeNet (pool5) + SVM (%)	ResNet (pool5) + SVM (%)
1	56.3	65.2	65.5	54.6	49.75
2	**77.1**	55.9	63.1	**77.9**	62.87
3	51.3	64.5	63.0	58.8	74.66
4	56.3	53.9	70.9	58.1	70.74
5	51.8	57.1	48.0	57.5	64.53

We also evaluate two-stage active viewpoint selection. For one run of 5-fold cross validation, we find that the static viewpoint selection approach yields an outcome classification accuracy of 74.36% with the random forest classifier (best performing static view). Whereas, our proposed active viewpoint selection approach leads to an improvement of approximately 5% with an accuracy of 79.23%. We test active viewpoint selection with random forest as the failure detection algorithm and AlexNet features as the image features in Algorithm 1 and 2. We report these results across several runs of 5-fold cross validation on the entire dataset. Thus, on average the active two-stage viewpoint selection approach helps in providing additional information to determine the outcome more accurately. Both of these methods outperform selecting a random viewpoint to determine the outcome, which gives an average classification accuracy of 67.31%. This validates that selecting the most informative viewpoints is beneficial to determine the outcome of a task more accurately.

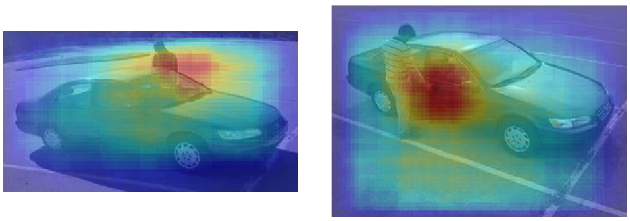


Fig. 6. Heat maps for various views in the quadrotor surveillance domain.

Again, heat maps were created for each viewpoint, and focused in on the intuitively informative area of the informative views, i.e. near the door handle (Fig. 6). It was interesting to observe that the heat maps of less informative views also focused on intuitively informative parts of the image, which are the parts where one can see the head of the person from over the car (Fig. 6).

V. CONCLUSIONS AND FUTURE WORK

We have demonstrated that viewpoint selection and fine-grained outcome classification are both tractable and useful for a general class of robotics problems. A task-agnostic approach was proposed to automatically select static viewpoints that result in high classification accuracy. The proposed approach was tested on different tasks and hardware setups and shown to work for small sized datasets. Additionally, a novel method was introduced that improves performance by discovering an information-optimized viewpoint that allows for active selection of an optimal classification viewpoint. We also demonstrated the potential of CNN features (extracted by using pre-trained networks) to perform surprisingly well on this problem, despite not being designed to extract fine-grained features. This adds to a growing body of evidence that deep learning is capable of performing fine-grained image analysis, even when not discriminatively trained for the task, without hand-crafting fine-grained strategies like a discriminative random forest and without requiring image alignment. We hypothesize that more specialized deep architectures focused on fine-grained features (such as [43], [44]), could improve performance in the future.

Our proposed approach can work in principle for any class that can be visually discriminated. Even though we only test for success or failure, our approach can be extended to work for multiple task outcomes in the future. Several other interesting directions are also available for future work. While our approach automatically selected viewpoints that are highly informative and discriminative, it may be beneficial to continuously adjust viewpoints over time during a task, or to use a viewpoint that was not seen in the original training set. One approach to do this may be to use interpretable data such as heat maps, along with RGB-D data, to identify important objects and relationships in the scene that can be tracked over time, or for which the viewpoint can be optimized. Also, rather than relying only on fine-grained visual details for outcome classification, future

efforts may integrate fine-grained information from several sources including vision, sound, and tactile data.

ACKNOWLEDGMENTS

This work has taken place in the Personal Autonomous Robotics Lab (PeARL), University of Texas at Austin and at Robert Bosch LLC, Palo Alto. PeARL research is supported in part by NSF (IIS-1638107, IIS-1617639).

REFERENCES

[1] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning (ICML), 2011*.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research, 2013*.

[3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *arXiv preprint arXiv:1504.00702, 2015*.

[4] S. Niekum, S. Chitta, B. Marthi, S. Osentoski, and A. G. Barto, "Incremental semantically grounded learning from demonstration," in *Robotics: Science and Systems (RSS), 2013*.

[5] C. Plagemann, D. Fox, and W. Burgard, "Efficient failure detection on mobile robots using particle filters with gaussian process proposals." in *International Joint Conference on Artificial Intelligence (IJCAI), 2007*.

[6] V. Verma, S. Thrun, and R. Simmons, "Variable resolution particle filter," in *International Joint Conference on Artificial Intelligence (IJCAI), 2003*.

[7] S. Roumeliotis, G. S. Sukhatme, G. A. Bekey, *et al.*, "Sensor fault detection and identification in a mobile robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1998*.

[8] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Robotic fault detection and fault tolerance: A survey," *Reliability Engineering & System Safety, 1994*.

[9] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *IEEE International Conference on Robotics & Automation (ICRA), 2011*.

[10] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2012*.

[11] A. Mueen, E. Keogh, and N. Young, "Logical-shapelets: an expressive primitive for time series classification," in *ACM International Conference on Knowledge Discovery and Data mining (SIGKDD), 2011*.

[12] M. Madry, L. Bo, D. Kragic, and D. Fox, "St-hmp: Unsupervised spatio-temporal feature learning for tactile data," in *IEEE International Conference on Robotics and Automation (ICRA), 2014*.

[13] J. Worcester, M. A. Hsieh, and R. Lakaemper, "Distributed assembly with online workload balancing and visual error detection and correction," *The International Journal of Robotics Research, 2014*.

[14] H. Nguyen and C. C. Kemp, "Autonomously learning to visually detect where manipulation will succeed," *Autonomous Robots*, 2014.

[15] G. Xue, T. Fukuda, and H. Asama, "Error recovery in the assembly of a self-organizing manipulator by using active visual and force sensing," *Autonomous Robots, 1995*.

[16] X. He, B. Benhabib, K. Smith, and R. Safaee-Rad, "Optimal camera placement for an active-vision system," in *IEEE International Conference on Systems, Man, and Cybernetics, 1991*.

[17] D. Stampfer, M. Lutz, and C. Schlegel, "Information driven sensor placement for robust active object recognition based on multiple views," in *IEEE International Conference on Technologies for Practical Robot Applications (TePRA), 2012*.

[18] R. Bodor, A. Drenner, M. Janssen, P. Schrater, and N. Papanikolopoulos, "Mobile camera positioning to optimize the observability of human activity recognition tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2005*.

[19] B. Triggs and C. Laugier, "Automatic camera placement for robot vision tasks," in *International Conference on Robotics and Automation (ICRA), 1995*.

[20] G. Leifman, E. Shtrom, and A. Tal, "Surface regions of interest for viewpoint selection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*.

[21] R. Pito, "A sensor-based solution to the next best view problem," in *International Conference on Pattern Recognition, 1996*.

[22] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, and J. Denzler, "An information theoretic approach for next best view planning in 3-d reconstruction," in *International Conference on Pattern Recognition, 2006*.

[23] G. Kootstra, J. Ypma, and B. De Boer, "Active exploration and keypoint clustering for object recognition," in *IEEE International Conference on Robotics and Automation (ICRA), 2008*.

[24] N. Govender, J. Claassens, F. Nicolls, and J. Warrell, "Active object recognition using vocabulary trees," in *IEEE Workshop on Robot Vision (WORV), 2013*.

[25] J. T. Feddema, C. G. Lee, and O. R. Mitchell, "Automatic selection of image features for visual servoing of a robot manipulator," in *IEEE International Conference on Robotics and Automation (ICRA), 1989.*

[26] H. Hashimoto, T. Kubota, M. Kudou, and F. Harashima, "Self-organizing visual servo system based on neural networks," *IEEE Control Systems*, 1992.

[27] C. W. Kennedy, T. Hu, and J. P. Desai, "Combining haptic and visual servoing for cardiothoracic surgery," in *IEEE International Conference on Robotics and Automation (ICRA), 2002.*

[28] P. Roßler, S. A. Stoeter, P. E. Rybski, M. Gini, and N. Papailikolopoulos, "Visual servoing of a miniature robot toward a marked target," in *International Conference on Digital Signal Processing, 2002.*

[29] B. Yao*, A. Khosla*, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*.

[30] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *IEEE International Conference on Computer Vision (ICCV), 2007*.

[31] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller, "Recognizing image style," *arXiv preprint arXiv:1311.3715, 2013*.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS), 2012*.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556, 2014*.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[36] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011*.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision, 2015*.

[38] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models, 2001*.

[39] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters, 1999*.

[40] 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*.

[41] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *International Conference on Advanced Robotics (ICAR), 2015*.

[42] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014*.

[43] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[44] S. Xie, T. Yang, X. Wang, and Y. Lin, "Hyper-class augmented and regularized deep learning for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.