Detecting Anomalous Objects on Mobile Platforms

Wallace Lawson Laura Hiatt Keith Sullivan
Naval Research Laboratory
Washington, DC USA
{ed.lawson, laura.hiatt, keith.sullivan}@nrl.navy.mil

Abstract

We present an approach where a robot patrols a fixed path through an environment, autonomously locating suspicious or anomalous objects. To learn, the robot patrols this environment building a dictionary describing what is present. The dictionary is built by clustering features from a deep neural network. The objects present vary depending on the scene, which means that an object that is anomalous in one scene may be completely normal in another. To reason about this, the robot uses a computational cognitive model to learn the dictionary elements that are typically found in each scene. Once the dictionary and model has been built, the robot can patrol the environment matching objects against the dictionary, and querying the model to find the most likely objects present and to determine which objects (if any) are anomalous. We demonstrate our approach by patrolling two indoor and one outdoor environments.

1. Introduction

Surveillance systems are a common way of providing security in a variety of environments. A typical surveillance system consists of multiple cameras providing visual coverage of an environment to a human operator monitoring video feeds. The goal of the system is to identify anomalous objects: the appearance of low-probability objects with respect to a model of normality for the environment (see Figure 1 for examples of anomalous objects). For example, seeing a toaster in the kitchen is not unusual, but seeing the same toaster in the hallway is. These systems require that the human sustains vigilance on the surveillance task over long periods of time, which has been shown to lead to avoidable errors and oversights [27, 12]. Automated surveillance systems mitigate this problem by providing automatic detection and tracking of unusual objects and people, and then alerting the human operator [10, 29].

We approach automated surveillance using a mobile platform (i.e., a mobile robot patrolling an environment). Although our work is applicable for either a static or mobile platform, a mobile platform provides more flexibility for the areas that we wish observe. Our mobile robot learns

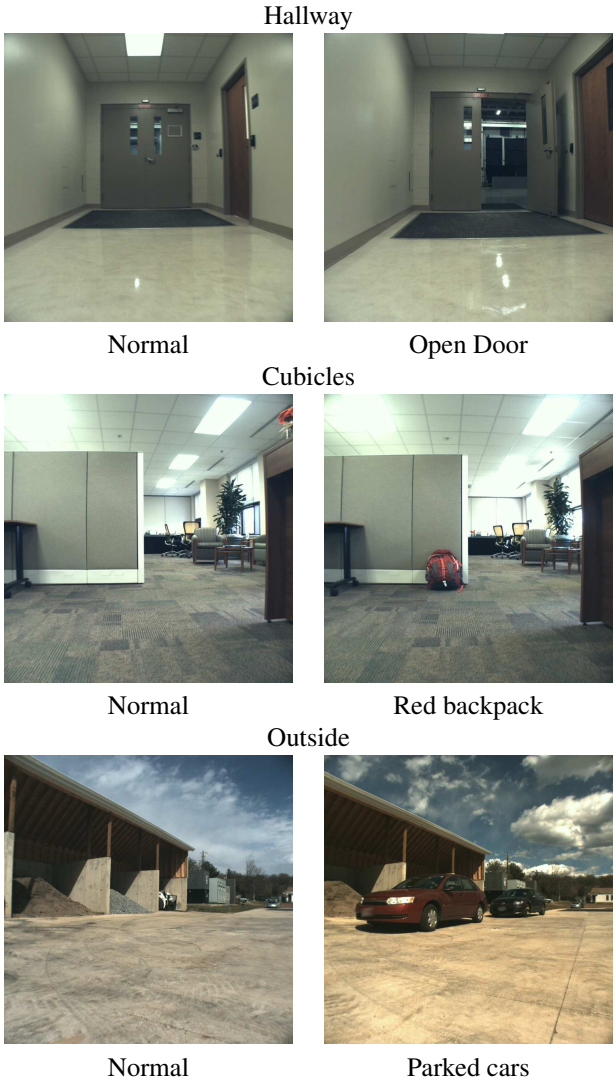

Figure 1. Example of normal environment (left) and anomalies (right).

about about the environments in an unsupervised manner, simply by observing what is present. This approach allows the robot to learn what is normal and what is not normal for environments, without a human expert having to specify such distinctions a priori.

Our mobile robot analyzes the environment using image patches extracted using a fixed sized grid, which has been shifted by a small amount in order to ensure that there are small overlaps. Each patch is processed by a deep network, with the sole purpose of using the features from the network (i.e., fully connected layer) to describe the patch. This representation incorporates both color and shape. Further, it has been trained in a way that makes it robust to changes in illumination, scaling, translation, and object size. Additionally, the features are a compact representation of the observed region, making this more feasible to be used on a computationally limited robot feasible.

During initial training, the robot patrols the environment in order to build up a dictionary of the things that normally appear in each environment. In practice, this can be an enormous amount of data: the robot typically captures at least 10,000 images during each patrolling session, resulting in over 1.5 million patches. We seek a clustering approach that does not require setting the number of clusters beforehand, and that can also append to an existing dictionary when new training data is acquired. We accomplish this using a streaming variant to k-means clustering, where a new cluster is formed whenever it exceeds a predefined distance to existing clusters.

In parallel, we query the PlacesCNN [32] to get the appropriate label for the robot's current location. Then, given the robot's area and the dictionary of features, we build up a model of those features are common to the different areas. Our model of normality uses context from the computational cognitive architecture, ACT-R/E [30]. Context in ACT-R/E takes the form of associations between concepts (here, locations within the environment and the dictionary of features). During training, as the robot learns about new areas in an environment, or sees new objects in a known area, the strengths of the associations between the involved areas and features are created and strengthened: if a feature is very typical for an area, then the association between them will be strongly weighted; if the feature is atypical for an area, then the association between them will be absent or weakly weighted.

Once we have constructed models of normality for various locations, the system is ready for anomaly detection. During runtime, the robot matches the deep features observed with the dictionary, and queries the PlacesCNN to determine its current scene. Then, it checks the cognitive model to determine the strengths between the observed features and its current location. A strong association indicates an in-context, non-anomalous object, and a weak or absent association indicates an out-of-context, anomalous object.

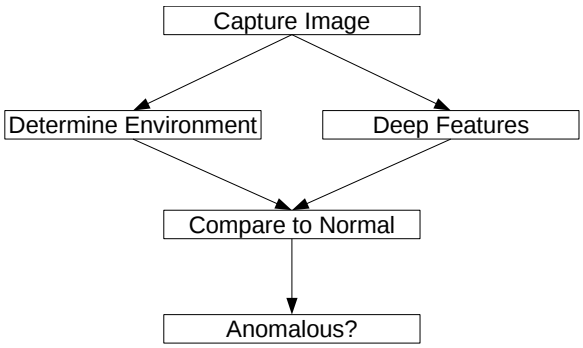

Figure 2. Overview of our approach

In our experiments, we patrol through 3 different types of indoor and outdoor environments training over 3 different days. A fourth day is collected to evaluate, and during this evaluation , we see a true positive rate of 91.43% and a false positive rate of 8.7% on anomalies that were inserted into the environment. Anomalies generally are things that were intentionally changed by adding objects to the environment or significantly modifying one of the objects in the environment. We are able to automatically detect the vast majority of the changes, while keeping the false positive rate low, which can provide a useful cue to help maintain the vigilance of a human operator. Also of note is that one of the strengths of this approach is that it is agnostic to a particular location, provided that it has been given a sufficient amount of time, and that it works equally well with static or mobile cameras.

2. Related Work

Related work on automated surveillance can be broadly split into techniques that target static versus mobile cameras, as well as techniques that learn in a supervised vs. unsupervised fashion.

2.1. Surveillance on Static Platforms

Most of the work on automated surveillance comes from the computer vision community and uses supervised learning on static cameras. Such approaches typically either focus on detecting anomalies at the pixel-level, and or at the region-level. Due to its algorithmic simplicity, image differencing is a common method for anomaly detection at the pixel-level [5, 7]. The procedure computes the pixel-by-pixel difference between a reference image and the current image, and then compares the difference to a threshold to determine if the pixel has changed. Image differencing depends on a data-dependent threshold that is sensitive to lighting and perspective changes, making it difficult and error-prone when implemented on a mobile platform.

Background modeling expands upon this work by build-

ing a model across multiple images rather than the pair-wise comparison of image differencing. The most common technique is to use a Gaussian mixture model to model changes in intensity across individual pixels [17, 33]. These approaches, however, still typically fail with moving cameras (such as those mounted on a robot) and environments with complicated patterns.

Region-based techniques, in contrast, are more robust to noise since they model changes across regions in an image sequence. Li et al. [18] look for anomalies using dynamic textures[8], using this approach to look for event-level anomalies in crowded pedestrian scenes. The authors define anomalies in terms of both appearance and motion. One of the strengths of this approach is the definition of anomalies, which as the authors state, should be considered in the context of the immediate surrounding environment. In this approach, they specifically target anomalies learned from objects moving in a scene from a single, unchanging viewpoint. We build on this approach by considering anomalies from fixed objects, and we do so using an approach that will also permit a camera to move through different scenes.

A large body of work has looked at anomaly detection from the perspective of motion: an anomaly is defined as motion that is different from the expected motion for the scene. Researchers either look at tracking motion anomalies in time [4, 26, 31], or using spatio-temporal gradients such as optical flow [1] and Markov Random Fields [15]. Mahadevan et al. [20] take a different approach by focusing on representation rather than using a global statistical representation. Minematsu et al. [21] develop a motion model of the sensor and use the model to find regions not described by the model. Our approach does not currently consider motion; however, we could do so in a straight-forward manner by using Minematsu et al.'s techniques to extract motion features and incorporate them into our approach.

Object level anomalies have been considered in the past, in the context of objects that are atypical for a certain object class (e.g., a car made of wood or a chain that is shaped like a boat) [24]. They reason that attributes are the most sensible way to describe abnormalities, since it is difficult to build a model for unexpected. They describe objects using shape and color attributes, then locate anomalous objects using a support vector machine. There are several key distinctions between our works. First, they do not take context into account, where this is one of the key features of our approach (i.e., a car parked in the front yard would be unusual). Second, they examine the image as a whole, whereas we only analyze parts of the image in an effort to find regions that are anomalous. Finally, supervised learning of anomalies in such a complicated environment would be infeasible.

In addition to the supervised techniques, there are several recent approaches to surveillance using static cameras that use unsupervised learning techniques [2]. Unsupervised learning offers a way for an automated surveillance system to adapt to changes in environment or to change the state of objects with an environment [25].

2.2. Surveillance on Mobile Platforms

Robotics researchers have looked at anomaly detection in the context of patrolling robots performing surveillance. For training, the robot is tele-operated through the environment, collecting data at regular intervals. During patrol, the robot determines the closest data in the training set to is current data, and then uses a statistical technique to detect any changes. While most of the robotics literature focuses on detecting anomalies in 3D point clouds (e.g., [13, 3]), some work focuses on a pure-vision solution. Kato et al. use GIST features to detect anomalies [14], while Chakravarty et al. use stereo correspondences to detect anomalies and a particle filter to track anomalies in time [6]. Both these approaches store the training images, and then use a localization technique to determine which training image is closest to the current view. Anomalies are determined by comparing the training image from the database with the current image. Due to the storage of training images, neither approach scales to large environments. Soibam et al. [28] perform a quantitative comparison of anomaly detection algorithms running on a patrol bot. They manually spatio-temporally align images from the training run with images from the testing run before running anomaly detection. Clearly, the manual alignment step introduces a significant overhead and precludes autonomous operation.

Most similar to our approach, Neto and Nehmzow construct a model of a normal environment, and then compare images to the model to determine if anomalies are present [22]. They construct a neural network using salient regions of the image described as local color statistics. The dependence solely on RGB color statistics, however, suggests their technique is not scalable beyond the presented engineered experimental environment, and would fail in real-world environments with real-world anomalies.

3. Methodology

We assume that a robot is continuously patrolling an environment, roughly following the same path through the environment. During the patrol, the robot is processing images, looking for anything that is different for the current environment. Our approach starts by constructing a model of normality for different environments. We define normal using a dictionary of deep features building using an unsupervised learning technique referred to as streaming clustering. Next, a cognitive context model associates the elements of this dictionary with different scenes. An anomaly is detected when the features are not strongly associated with the

normal model for a given scene.

We start by tele-operating the robot through various environments, and then constructing the individual normality models off-line. However, our approach is also able to learn this model incrementally and on-line, having the robot conterminously construct/update models of normality. The following discussion assumes construction of the models are off-line, but a similar approach would work for on-line construction.

3.1. Building a Dictionary of Deep Features

Given a single image, the robot first extracts patches from the image using a grid of size NxN, with a step size is $N/2$ to ensure a small overlap between grid cells. The appropriate size of N typically depends on the environment and the types of anomalies that will be detected. We found, however, that our approach worked well for a range of values of N since we typically see the objects and anomalies at various scales during normal interaction in the environment.

The appearance of each patch is represented with deep features, using the "AlexNet" architecture [16], which was fully trained using ImageNet data. The representation of each patch comes from the last fully connected layer ($fc7$) from AlexNet, which has 4096 features, typically quite sparse. AlexNet has shown a great ability to generalize to other datasets, including the well-known and challenging Pascal VOC dataset [19]. These features also generalize well to other domains, such as has been shown in [23].

The robot will be given a number of images at a rapid frame rate, which would make traditional clustering techniques infeasible to organize the data. Instead, we rely on an online version of clustering, which has been previously referred to as streaming clustering [9]. This applies in situations when clustering is needed, but due to time, space, or other limitations, the data can only be seen once. In this case, streaming clustering builds clusters as it processes the data. When a new patch (v) is seen, it is compared to the clusters representing known shapes (c). If it is sufficiently closer to an existing cluster, it is labeled appropriately. If it is not similar to an existing cluster (using Euclidean distance), a new cluster is created.

Using this approach, the dictionary is built incrementally as the robot interacts with the environment. New deep features are added to each dictionary as they are seen. Some examples of clusters created in this manner are in figures 4 and 5. It interesting to note that when possible, streaming clustering re-uses symbols from indoor scenes to outdoor. Some examples here are potted plants, carpets, lighting and wall. While it's quite obvious why a plant would be re-used, most of the others were likely kept due to to their ability to describe something in the outdoor scene as well. For example, in the case of the solid colors, it also matches the appearance of the walls outside of the building

3.2. Learning Scene Context

The contextual learning component of our approach takes place within the computational cognitive architecture ACT-R/E [30]. As part of the architecture, ACT-R/E learns rich associations between concepts in memory that are learned incrementally over time. Here, context takes the form of associations between related concepts that are learned incrementally over time. The strengths of the associations are Bayesian-esque: as concepts are thought about with one another, their association is strengthened; if the concepts are thought about without the other, however, then their association is weakened. The exact equations for the associations' strengths can be found in [11].

We use these associations, and their strengths, to represent what one is typically expected to see in an environment. During training, the context model "saw" the deep features of each patch of each image, and was provided with the location (a) from the Places CNN for that image. With each datapoint (k), the context model incrementally learned what normal was for each scene type by updating its associations (c_{ka}) as described above. Ultimately, after training, features that are typical for a scene have very strong associations with that scene; features that are atypical for a scene have absent, or very weak, associations with it.

3.3. Detecting Anomalies

During evaluation, the robot must match up the patches observed with the scene. The spareness of the data can lead to an issue where a data point may be close to several different dictionary elements. In this case, the correct assignment may be difficult to make without any further scene information. Scene information gives us prior information on the types of clusters that we will see (e.g., a kitchen may have a 'coffee pot' cluster; an office may have a 'telephone' cluster). Therefore, each observed patch is weighted by the association strengths ($c_k a$) for dictionary element k in scene a.

$$p_k = \exp(-d_k/\sigma) \tag{1}$$

$$k(a) = argmin_k(p_k c_{ka}) \tag{2}$$

Where d_k is the distance to cluster K, and σ is a parameter that can be estimated from the expected distribution of the data. The robot detects an anomaly when $k(a)$ falls below a threshold when given scene a from the PlacesCNN.

4. Experimental Results

Our experiments sought to show how our technique applies to a wide variety of environments. To that end, we drove our robot through three unique environments: hallway, cubicles, and outdoors. Figure 3 shows example images from each environment. The hallway environment is a

Environment	Number of Images
Hallway	13206
Cubicles	8827
Outside	11669

Table 1. Number of images collected during training for each environment.

typical office style hallway with two side corridors and multiple doors, both open and closed. The cubicle environment consists of multiple cubicles, chairs, and tables along with multiple reception areas, a kitchenette, and a copy room. The outdoor environment contains mainly concrete and the outside of a building.

We mounted a Carnegie Robotics S7 camera at 1024x1024 resolution at 15 FPS atop a Pioneer 3AT mobile robot. The S7 camera projects a circular image on a square field; so we crop a centered middle square of 750x750 pixels. Data collection was performed by tele-operating the robot multiple times through each environment. Training data consisted of six runs through each environment spread over three days. Table 1 shows the number of images collected per environment. Learning continues over all 3 days due to natural variations in the environment, as well as some variation in the path taken by the robot. Interestingly, the robot continued to learn over all 3 days, although the amount learned decreased over time. The PlacesCNN identified 47 different scenes which build up 11,203 associations between the scenes and the deep features dictionary.

The computational cognitive model learned about 47 different environments during training, building associations between all of the learned objects. Figure 4 shows some of the patches from an indoor environment, and figure 5 shows some of the patches from an outdoor environment. Interestingly, there is overlap between the patches most strongly associated with indoor and outdoor environments. In some cases, the computational cognitive model is learning basic concepts like vertical lines, horizontal lines, and center-surround. In other cases, (such as the wall color), the outside of the building is also lightly colored. The strength of the cognitive model comes from the amount of weight is placed on each of these patches, which in this figure is represented by the order of in which they appear in the figure (top to bottom, left to right).

To evaluate accuracy of anomaly detection, the goal was to determine how well our approach handles different types of anomalies. Some anomalies are obvious (e.g., a backpack in the hallway, a car parked illegally) while others are subtle (e.g., a moved trashcan, additional poster on the wall). Our testing data consists of two runs through each environment.

In the experimental results, we extract features from 100x100 grid cells, using a step-size of 50 pixels, resulting in a total of 169 evaluated patches per image. An anomaly is

Environment	FPR	TPR
Office 1	11.82%	100%
Office 2	13.25%	87.5%
Hallway 1	4.27%	87.5%
Hallway 2	4.35%	88.9%
Outside 1	8.42%	93.33%
Outside 2	12.28%	88.24%
	8.7%	91.43%

Table 2. Accuracy of the approach at a selected threshold.

considered to be correctly detected if it has at least one grid cell that exceeds the given threshold for an anomaly. Likewise, a false positive is whenever a grid cell activates when it is not on an anomaly. Such an experiment can be difficult to perform in an active office environment, since there are always small, subtle changes. In some cases, anomalies were detected, but since they were not one of the inserted anomalies, they are considered nuisances and are false positives. Some examples are office plants that were moved, chairs that were at a different orientation, trash bags that were placed differently, and paper towels that were left on a counter.

Figure 6 shows the ROC curve, and Figure 7 shows some common detection and failure modes. Results at a selected threshold are shown in table 4. The algorithm performs well on true positives, but not surprisingly struggles on non-rigid objects like sweatshirts. It is possible that evaluating this from different angles and perhaps different scales would increase the performance. Many of the false positives, particularly in the office environment, were at a distance. This was because of the increasing complexity when considering a lot of objects together, it is possible that we can eliminate this by looking only at the immediately surrounding region. Finally, some of the biggest issues were related to lighting, due the auto-gain and auto-white balance of the camera. When near windows, or moving into and out of shadows, the picture changed dramatically. In all of the cases, the performance on the environment with constant light (hallway) is the highest.

5. Discussion

Detecting anomalies in practical environments can be highly challenging for a number of reasons. It is difficult to define normal, especially since normality changes from environment to environment. Also, it is difficult to train hand crafted object detectors to find anomalies, since by their very nature they are not typical for the environment.

The proposed approach simultaneously solves both of these problems using a computational cognitive model to learn normality, and unsupervised learning of deep features to build a model of the objects that appear in each environment.

Figure 3. Example images from each experimental environment. The top row is hallway, the middle row is cubicles, and the bottom row is outdoors.

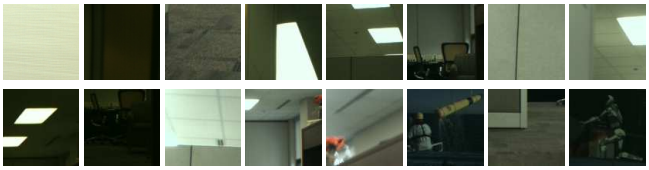

Figure 4. The patches from the deep features dictionary most strongly related to to an indoor scene. These are in order of decreasing association strength from left to right, top to bottom.

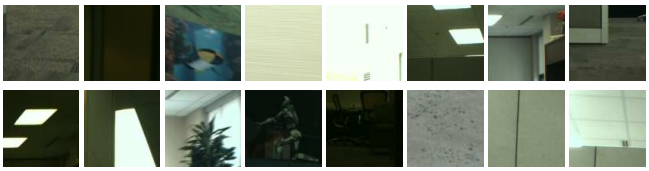

Figure 5. The patches from the deep features dictionary most strongly related to an outdoor scene. These are in order of decreasing association strength from left to right, top to bottom.

The strength of this approach is the ability to learn about a lot of different objects and environments. In our results, we did this over a 3 day period, where the approach learned about almost 1600 items in 47 different environments. This model was queried during the evaluation mode, both to identify the most likely objects in the environment, as well as to find anomalies. Through repeated exposure of the same environment, it is likely that the accuracy would continue to increase. Although we did not incorporate a map into our approach, it is possible that this could further improve accuracy.

Detecting anomalous objects can be difficult. Some of the limitations in the model were objects that moved on a regular basis which were identified as anomalies. Some examples include chairs, trash cans, and plants moved for watering. Although they could be classified as an anomaly, most operators would identify this as more of a nuisance. Higher level reasoning is needed in such cases to identify the likely cause of the change and to rule out such nuisances.

For the purposes of this work, we considered a fixed sized grid with the intention of moving the robot closer and farther from the environment, which eliminates the need for multiple scales. In practice, it may be useful to consider bounding boxes of different sizes, possibly by incorporat-

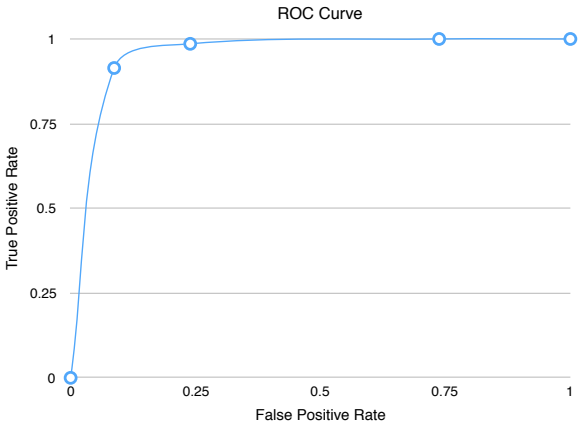

Figure 6. ROC curve showing the performance of the algorithm in the test environment.

ing superpixels or selective search.

Acknowledgements

Wallace Lawson was supported by the Office of Naval Research, and Keith Sullivan was supported by the Naval Research Laboratory under a Karles Fellowship. Laura Hiatt was supported by the Office of Naval Research and the Office of the Secretary of Defense.

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, March 2008. 3

[2] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, volume 1, pages 175–178, 2006. 3

[3] H. Andreasson, M. Magnusson, and A. Lilienthal. Has somethong changed here? autonomous difference detection for security patrol robots. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3429–3435, Oct 2007. 3

[4] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008. 3

Figure 7. Anomalies detected in the environments. In the figure, green rectangles show correctly detected anomalies; red rectangles show false positives. Any detected faces are blurred to preserve privacy.

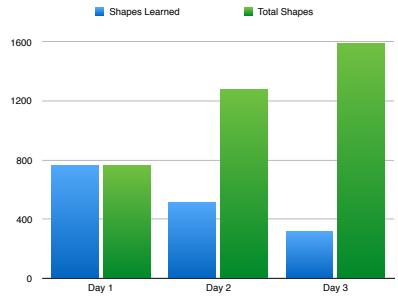

Figure 8. Learning about environment during the training sessions.

[5] R. A. Bindschadler, T. A. Scambos, H. Choi, and T. M. Haran. Ice sheet change detection by satellite image differenc-

ing. *Remote Sensing of Environment*, 114(7):1353 – 1362, 2010. 2

[6] P. Chakravarty, A. M. Zhang, R. Jarvis, and L. Kleeman. Anomaly detection and tracking for a patrolling robot. In *Proceedings of Australasian Conference on Robotics and Automation*, 2007. 3

[7] C.-S. Chan, C.-C. Chang, and Y.-C. Hu. Color image hiding scheme using image differencing. *Optical Engineering*, 44(1):017003–017003–9, 2005. 2

[8] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003. 3

[9] Z. Duric, W. E. Lawson, and D. Richards. Streaming clustering algorithms for foreground detection in color videos. In *VISAPP (2)*, pages 486–491, 2007. 4

[10] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, Aug 2000. 1

[11] L. M. Hiatt, W. E. Lawson, A. M. Harrison, and J. G. Trafton. Enhancing object recognition with dynamic cognitive context. In *AAAI Workshop on Symbiotic Cognitive Systemks*, 2016. 4

[12] R. Hockey. *The psychology of fatigue: work, effort and control*. Cambridge University Press, 2013. 1

[13] P. D. Jr., P. Núñez, R. P. Rocha, M. Campos, and J. Dias. Novelty detection and segmentation based on gaussian mixture models: A case study in 3d robotic laser mapping. *Robotics and Autonomous Systems*, 61(12):1696 – 1709, 2013. 3

[14] H. Kato, T. Harada, and Y. Kuniyoshi. Visual anomaly detection from small samples for mobile robots. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3171–3178, Oct 2012. 3

[15] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2928, June 2009. 3

[16] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of Neural Information Processing Conference (NIPS)*, 2012. 4

[17] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, May 2005. 3

[18] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014. 3

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 4

[20] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 0:1975–1981, 2010. 3

[21] T. Minematsu, H. Uchiyama, A. Shimada, H. Nagahara, and R. i. Taniguchi. Evaluation of foreground detection methodology for a moving camera. In *Proceedings of Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 1–4, Jan 2015. 3

[22] H. V. Neto and U. Nehmzow. Real-time automated visual inspection using mobile robots. *Journal of Intelligent and Robotic Systems*, 49(3):293–307, 2007. 3

[23] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 4

[24] B. Saleh, A. Farhadi, and A. Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 787–794, 2013. 3

[25] T. Sandhan, A. Sethi, T. Srivastava, and J. Y. Choi. Unsupervised learning approach for abnormal event detection in surveillance video by revealing infrequent patterns. In *Proceedings of International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2013. 3

[26] N. T. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of European Conference on Computer Vision (ECCV)*, 2002. 3

[27] G. J. Smith. Behind the screens: Examining constructions of deviance and informal practices among CCTV control room operators in the UK. *Surveillance & Society*, 2(2/3), 2002. 1

[28] B. Soibam, S. K. Shah, A. Chaudhry, and J. Eledath. Quantitative comparison of metrics for change detection in video patrolling applications. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 601–608, Sept 2009. 3

[29] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, Aug. 2000. 1

[30] J. G. Trafton, L. M. Hiatt, A. M. Harrison, F. P. Tamborello, II, S. S. Khemlani, and A. C. Schultz. ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1):30–55, 2013. 2, 4

[31] T. Zhang, H. Lu, and S. Z. Li. Learning semantic scene models by object classification and trajectory clustering. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1940–1947, June 2009. 3

[32] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2

[33] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773 – 780, 2006. 3