Analysis of Illumination Robustness in Long-Term Object Learning

Analysis of illumination robustness in long-term object learning

Ingo Keller MACS, Heriot-Watt University
Email: ijk1@hw.ac.uk

Katrin Solveig Lohan MACS, Heriot-Watt University
Email: k.lohan@hw.ac.uk

***Abstract*— In this article we evaluate the incremental object learning approach of the iCub humanoid robot which is directed towards long-term engagement. Affordable robot companion systems are currently entering the consumer market which highlights the importance in understanding environmental influences on robotic systems under real world conditions. If a robot is to be sent into the real world or different robots/sensors are to be used, we need our algorithms to be independent from both illumination and sensor influenced changes. In our work, we investigate the robustness of the interactive object learning to linear and non-linear lighting changes which can occur due to illumination changes throughout the day or the sensors used. Our results with the models we use suggest that the current method is susceptible to these changes. Therefore, we provide an adjustment to the current method to be able to cope with this problem.**

I. INTRODUCTION

With the recent advances in robotics, computer vision and machine learning, robot companion systems have started to appear in private homes such as the Jibo[1] or the Budy Robot[2]. Whilst this may lead to the availability of personal robotic assistants, the research into long-term engagement with these systems is still in its early stages. One major obstacle is the real world environment which is less controlable than a lab environment and therefore raises new challenges to established methods.

Computer vision for object recognition is one of the most important skills that robotic companions need to have. It is not only required to be able to recognize the robot's working environment such as people or places, but also necessary to support other abilities of the robotic system such as object manipulation [6] and navigation [2]. Thus, it is important to understand the capabilities and constraints that come with state-of-the-art visual recognition systems.

For our analysis, we had a closer look at the iCub's incremental object learning approach. This approach utilizes a Convolutional Neural Network (CNN) in combination with a Multiclass Support Vector Machine (SVM) for the classification of objects that were shown to the iCub [8]. An exhaustive evaluation on the general performance of the combined networks can be found in [10]. The iCub's implementation provides a viable method for long-term object recognition due to its incremental training on given input images which are acquired by interaction with the robot over time. The vision system is able to retrain its classifiers in real-time and therefore is usable in real world scenarios in which novel objects can appear at anytime.

[1] https://www.jibo.com/
[2] http://www.bluefrogrobotics.com/en/home/

TABLE I
IMAGES PER OBJECT

Class \ Object Number	1	2	3	4
Cup	1373	1380	1368	1481
Dishwasher Detergent	1389	1384	1495	1384
Laundry Detergent	1334	1329	1725	1443
Plate	1369	1646	1436	1537
Soap	1331	1365	1625	1353
Sponge	1313	1352	1353	1475
Sprayer	1365	1332	1383	1373

While in theory every robot could be trained individualy to learn the objects of the world, it is not desirable to be forced to do it for each machine. Pretrained learning systems that are able to expand their knowledge on-the-fly are required for practical use. Therefore, we are interested in the reusablity of provided datasets and trained classification models. If learned classifiers could be reused on different robots or even different robotic plattforms it would dramatically reduce the amount of computation required to fully train such systems. However, before we are able to do so, we need to understand the constraints that state-of-the-art systems are bound to. In particular, we are interested to see if current algorithms are able to deal with real world environments. Furthermore, if by using, e.g. illumination changes for learning, we can support these algorithms.

II. EXPERIMENTAL SETUP

In this section we describe the setup of our experiment and the models we are using.

To compare with existing results, we conducted our evaluation on the ICUBWORLD28 dataset[8]. While a practical experiment could be conducted to show the method's behaviour under different illumination conditions, we chose to modify the sample images of the given dataset for the purpose of repetability. This way, we are able to separate linear and non-linear changes as well while this is much more difficult to achive in an experimental setup. In a practical experiment both types of changes might happen at the same time in a way that is not easy to control. One such example is the Auto Gain Control (AGC) on consumer camera sensors that influence the resulting image in a non-linear mannor when the surrounding illumination is changed [3]. Due to the integration on the current CCD sensors themselve, it is sometimes impossible to deactivate these helper systems.

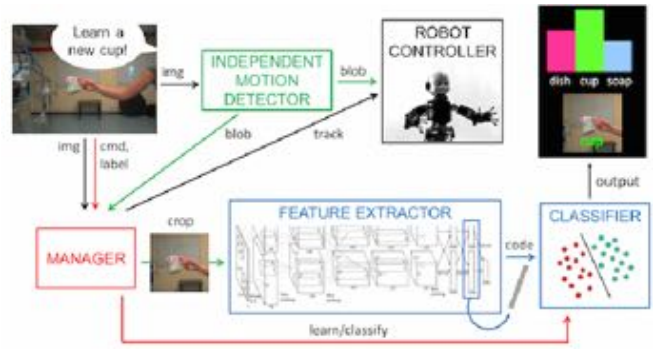

Fig. 1. The visual recognition system currently used on the iCub. [9]

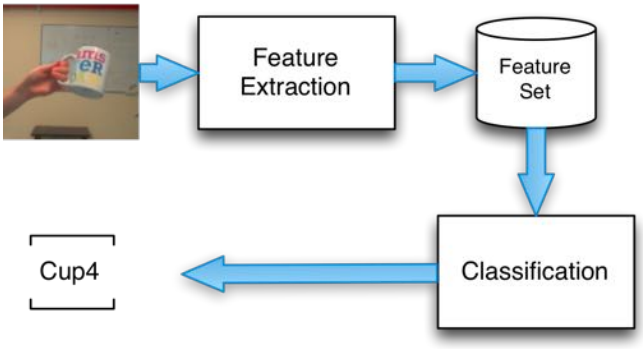

Fig. 2. Analysis Workflow.

A. Dataset

For our analysis we used the ICUBWORLD28 dataset from Pasquale et al.[8]. It represents the visual experience of the iCub humanoid and was created during a four day interactive learning session. It consists of nearly 40.000 images of 28 objects from 7 object classes with more than 1300 images per object (see Table I).

The dataset comes separated by day and is split into a training and test set per day. Since we were interested in the overall performance of the approach we merged all images from one object into one set. The merged dataset was then resplit into test and training data; with different numbers of images for training in each experiment and the remainder used as test images. Whilst the number of training samples is fixed for each object the number of test samples dependent on the available images.

B. Visual Recognition System

The iCub's interactive object learning system is part of the open source software stack that is provided by the iCub community[3]. Here we concentrated on the Feature Extractor and Classifier (see Figure 1).

The Feature Extractor consists of a pre-trained CNN based on the *BLVC Reference CaffeNet* which is provided by the Caffe library[4]. This network was trained on the ImageNet dataset [5]. The module receives cropped images from earlier stages of the overall system. In our case the images are provided by the ICUBWORLD28 dataset which are saved from the 4 days experiment as mentioned earlier. From these the Feature Extractor generates a feature vector. This vector consists of 4096 double values which are characteristic for the input image under a given CNN. The feature vector corresponds to the vector representation of the highest convolutional layer as shown in Figure 1 (see [9]).

The Classifier uses a Multiclass SVM as a linear classifier on the encoded features that are produced by the Feature Extractor. The SVM is trained with a 1-vs-all strategy for 1000 epochs and thus provides a classification predictor for each of the 28 objects. The training time not only increases with the number of samples but also with the number of classes for which it needs to train the individual classifier.

[3]https://github.com/robotology

In order to use the given implementation without the robot involved, we had to separate the recognition workflow from the rest of the system to use the pipeline in a standalone manor. This was necessary to provide an analysis environment that resembles the original system as closely as possible (see [9]). Since the provided solution is a highly integrated system we had to rearrange the workflow to fit our needs thus we also could decrease the processing time for the analysis (see Figure 2). Since we did not require any interaction we first generated all feature vectors for the images and stored them in a feature set cache.

C. Illumination change models

To account for linear and non-linear light changes, we derived new images from the original ICUBWORLD28 dataset based on two different models. As an example of a linear modification we chose the value modification from the HSV colorspace. HSV stands for *hue*, *saturation*, and *value*, where *value* accounts for brightness. The second model is given by the gamma transformation which serves us as a model for non-linear changes. The transformation was done using the OpenCV library[1]. Figure 3 shows examples for both modifications. While these changes seem to be not that difficult for the human eye it has an impact on the recognition as we will see later.

To realise the linear model, the images where transformed into the HSV colorspace for the modification. This allows for changes to the luminance without interfering with the colors and is one of the common colorspaces for classical visual recognition. It is defined as $V_{out} = V_{in} \pm (V_{max} * V_c)$. After the colorspace transformation V_c was changed to 5%, 10%, 15% and 25% each for lighter and darker appearance.

Gamma correction is a non-linear transformation that is often used to enhance the visual appearance of images that are under- or overexposed. It is defined as $V_{out} = V_{in}^{1/\gamma}$. For the gamma value we chose 0.4, 0.5, 0.7 for darker and 1.5, 2.0, 2.5 for brightness images.

III. EXPERIMENTS AND RESULTS

For the analysis, we first generated the transformed images and their feature vectors as mentioned above and stored them in the feature set cache. For the training we chose the first

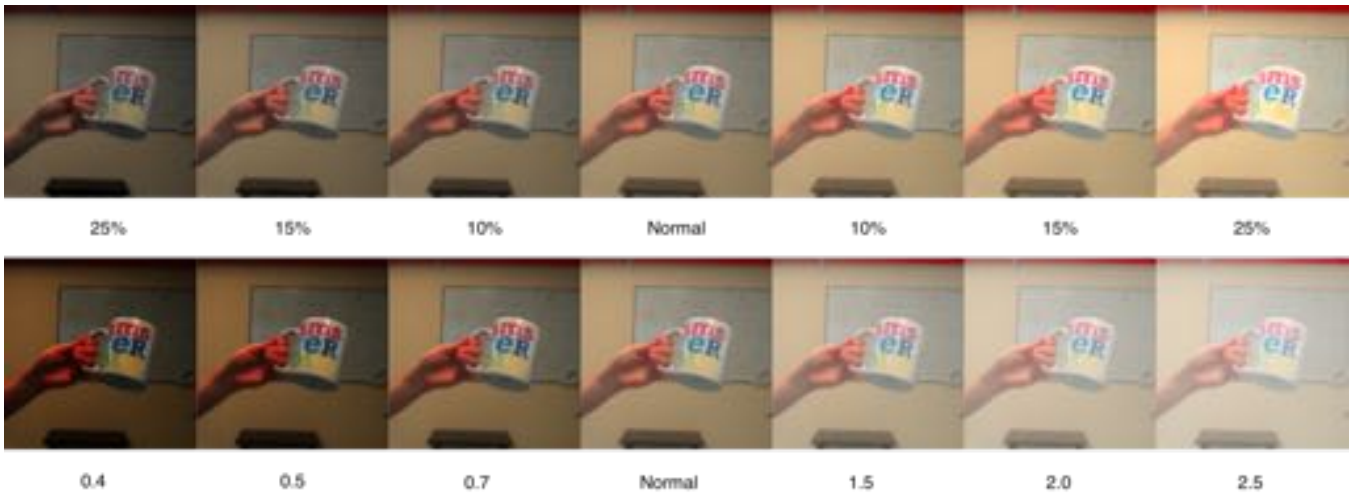

Fig. 3. A sample and its transformations: (top) changing the value in the HSV colorspace, (bottom) changing the gamma value in RGB colorspace

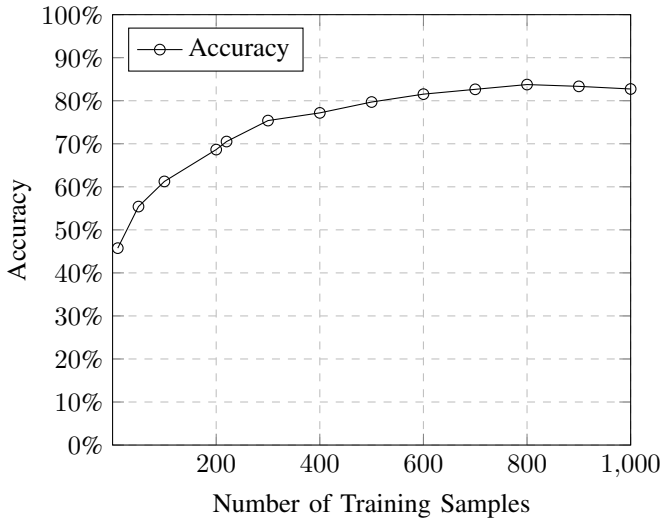

Fig. 4. Accuracy - Sample Size

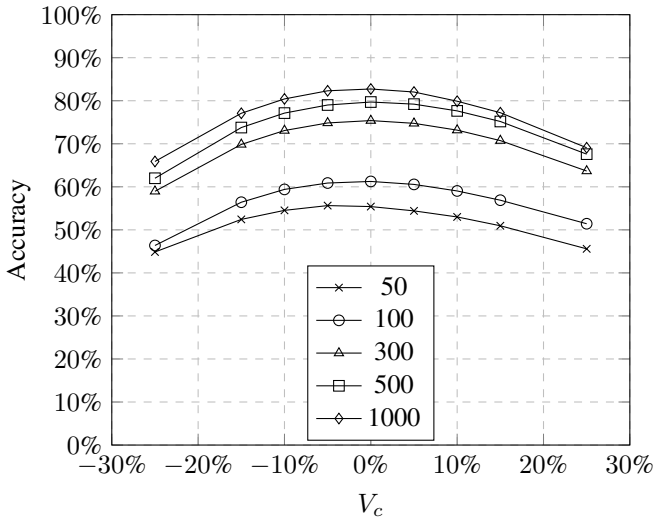

Fig. 5. Linear modified test samples

i images from each feature set and used the rest for testing the recognition. In the following, accuracy is defined as the percentage of correctly predicted images with respect to the whole test set.

For the training of the SVM we chose a wide range in sample size to see the impact in respect to the number of images for training. While this was limited to 220 samples (from 20 seconds of interaction) for training in [8], we wanted to see how the system performs with the availability of more training data.

Figure 4 shows the accuracy for different training set sizes. At 100 samples per object it already shows an accuracy above 60% which increases up to 84% for 800 samples. We also can confirm the reported 70% accuracy at 220 samples, showing that our workflow is inline with the experiment from Pasquale et al. [9].

In the next step, we assessed the performance of the linear classifier with respect to the modified images. In Figure 5 and Figure 6 we show the results for the linear and non-linear transformed test samples. In both conditions the SVM shows a decrease in accuracy towards the more extrem conditions (see Table II). The accuracy for the modified images decreases faster with the increase of training samples, indicating that the robustness in recognition comes at the cost of flexibility to illumination changes. For example, the

TABLE II
SELECTED ACCURACY VALUES

Linear condition	-25%	not modified	+25%
50 Training Samples	44.86%	55.41%	45.59%
500 Training Samples	61.99%	79.70%	67.59%
Non-linear condition	0.4	not modified	2.5
50 Training Samples	34.96%	55.41%	35.47%
500 Training Samples	52.48%	79.70%	51.67%

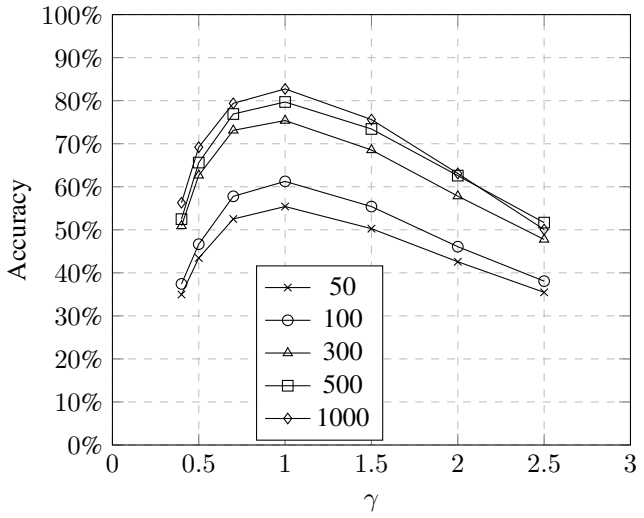

Fig. 6. Non-linear modified test set

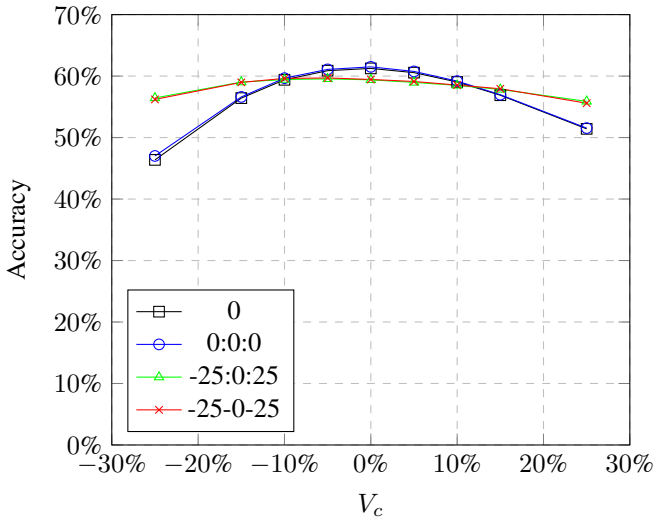

Fig. 7. Linear modified training and test samples

accuracy at 50 training samples decreases by around 10% points whereas with 500 training samples the decrease is around 15% points for the linear condition. The differences in the extrema in both conditions are different partially due to the different strength of the changes. The extreme condition in the non-linear case appears more dark/light than in the linearly changed case. One of the next steps could be to find a metric that makes these changes comparable in respect to brightness to identify the influence of the type of illumination change on the recognition accuracy.

TABLE III
MODIFIED TRAINING SETS

Linear condition	
0	$V = 0$, this corresponds to the values of Figure 4
0:0:0	$V \in \{0, 0, 0\}$ to account for the increased number of samples on the next set
-25:1:25	$V \in \{-25\%, 0, 25\%\}$
-25-1-25	$V \in \{-25\%, -15\%, 0, 15\%, 25\%\}$
Non-linear condition	
1	$\gamma = 1$, this corresponds to the values of Figure 4
1:1:1	$\gamma \in \{1, 1, 1\}$ to account for the increased number of samples on the next set
0.4:1:2.5	$\gamma \in \{0.4, 1, 2.5\}$
0.4-1-2.5	$\gamma \in \{0.4, 0.5, 0.7, 1, 1.5, 2, 2.5\}$

Finally we conducted an analysis using modified images as training sets. We compared the predictions for both conditions using training sets that are composed of 100 training samples of each modification. The modifications and corresponding training sets are shown in Table III. Adding more images to the recognition process has an impact on the accuracy as shown above. Thus, we included a training set that contains the same sample size as in one of the modified versions. For the linear condition we added the 0:0:0 set as a control group for the -25:0:25 set and for the non-linear condition we added the 1:1:1 set as a control group for the 0.4:1:2.5 set. Additionaly, we were interested to see how adding more modified images to the training set influences the performance of the learning.

TABLE IV
SELECTED ACCURACY VALUES - MODIFIED TRAINING SETS

Linear condition	-25%	not modified	+25%
0	46.37%	61.25%	51.42%
0:0:0	47.01%	61.50%	51.54%
-25:0:25	56.44%	59.39%	55.98%
-25-0-25	56.21%	59.45%	55.59%
Non-linear condition	0.4	not modified	2.5
1	37.43%	61.25%	38.07%
1:1:1	37.64%	61.50%	38.14%
0.4:1:2.5	50.13%	55.76%	53.91%
0.4-1-2.5	49.54%	55.71%	53.32%

As shown in Figure 7 and Figure 8, in both conditions the modified training sets outperform the standard training method under modified test samples but at the cost of accuracy for the unmodified test samples. This means the resulting classifier from our training methods are not as specific as the standard method but are more flexible in respect to changes in light conditions for linear and non-linear illumination changes.

Table IV also shows that the use of replicated images does not change the systems accuracy. But that was expected since a SVM cannot draw more information from a sample repetition. However, it is important to note that using more of the modified images did not help the recognition either. That indicates that it is sufficient to feed the the most extrem modified images to the linear classifier in order to increase the performance within that range. Additionally, a relationship between the decrease in specificity and the increase in flexibility is indicated by the given plots. That means the more one wins in flexibility the more one loses in specificity. However, this needs further investigations.

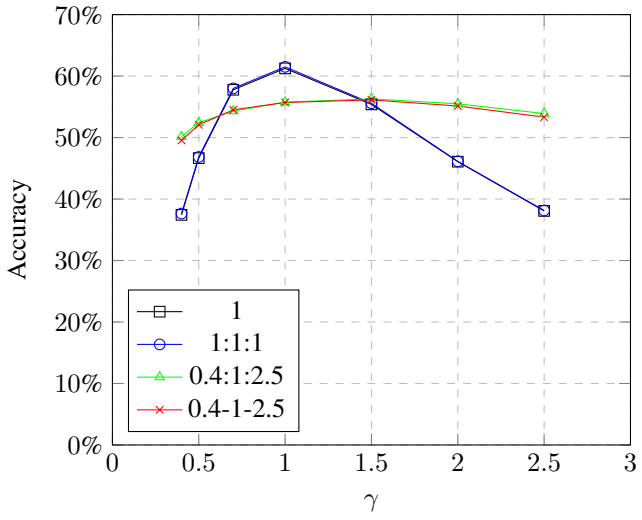

Fig. 8. Non-linear modified training and test samples

IV. DISCUSSION

The results show that the visual recognition approach is able to cope with minimal changes in the illumination settings. However, if the difference in illumination between trained and tested images becomes too large, such as the 25% in the linear model, the systems accuracy drops. In the non-linear case the observed reduction in accuracy is even larger. This indicates that color takes a dominate role in the feature generation step. This is in line with the literature [7] which suggests that CNN-based approaches can be fooled by images that resemble color patterns but not shape or size. Our results show that training with modified samples can positivly influence overall performance across different lighting conditions at the cost of a small decrease in performance under normal lighting conditions.

The impact in a real-world test scenario needs further investigations, but we can assume that changes in illumination from the environment will have an impact on recognition as will using different sensors being used to acquire the images. The non-linear case needs to be especially mentioned as current vision sensors, such as webcams, tend to incorporate automatic non-linear adjustments to increase the visual appearance of the image for the human eye. In particular automatic gain control mechanisms need to be accounted for (see [3]).

V. CONCLUSIONS

From our findings we draw the conclusion that the state-of-the-art visual recognition approach using the combination of Convolutional Neural Network and Support Vector Machine is susceptible to changes in illumination. These changes can occur due to changing environmental light or due to the use of different sensors. Thus making the deployment of datasets for training or pre-trained visual recognition models for robotic systems difficult to achive. Additionally, we provided a method to compensate for illumination change problem. Currently, we are planning to evaluate the object recognition approach on different datasets and additional transformation models. Furthermore, we want to investigate the relationship between specificity and flexibility.

ACKNOWLEDGMENT

The authors would like to thank Eli Shepard for his help. We would also like to thank the iCub community for providing their software stack as open source. Finally, we like to thank the Heriot-Watt University for the financial support.

REFERENCES

[1] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] Christian Dondrup, Christina Lichtenthler, and Marc Hanheide. Hesitation signals in human-robot head-on encounters: a pilot study. pages 154–155. ACM Press, 2014. 00004.

[3] K. R. Fowler. Automatic gain control for image-intensified camera. *IEEE Transactions on Instrumentation and Measurement*, 53(4):1057–1064, August 2004.

[4] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 04004.

[6] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4373–4378, May 2012. 00048.

[7] Anh Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, June 2015.

[8] Giulia Pasquale, Carlo Ciliberto, Francesca Odone, Lorenzo Rosasco, and Lorenzo Natale. Real-world Object Recognition with Off-the-shelf Deep Conv Nets: How Many Objects can iCub Learn? April 2015.

[9] Giulia Pasquale, Carlo Ciliberto, Francesca Odone, Lorenzo Rosasco, Lorenzo Natale, and Ingegneria dei Sistemi. Teaching iCub to recognize objects using deep Convolutional Neural Networks. In *Proceedings of The 4th Workshop on Machine Learning for Interactive Systems*, pages 21–25, 2015.

[10] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. pages 806–813, 2014. 00006.