Artificial Dataset Generation and Convolutional Neural Network Regression for Object Localization in Laboratory Images

Benjamin Killeen
Undergraduate
Department of Computer Science
University of Chicago
killeen@uchicago.edu

Gordon Kindlmann
Associate Professor
Department of Computer Science
University of Chicago
glk@uchicago.edu

Machine learning has yielded promising results in the analysis of real-world images, prompting the question of whether similar methods may be applied to laboratory experiments. In scientific applications, known constraints and the desire for precision in a well-defined solution space constitute a very different problem than real-world image analysis. We explore the generation of artificial datasets for training, including the perturbations necessary to generate a precise network. We apply some of these methods to object localization, employing shallow convolutional neural networks with no pooling layers for the regression of image-space position. We show the effectiveness of translational perturbations in these experiments.

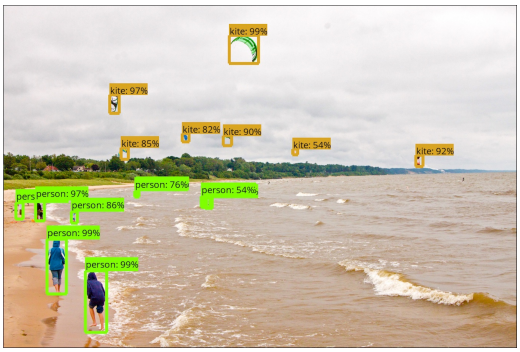
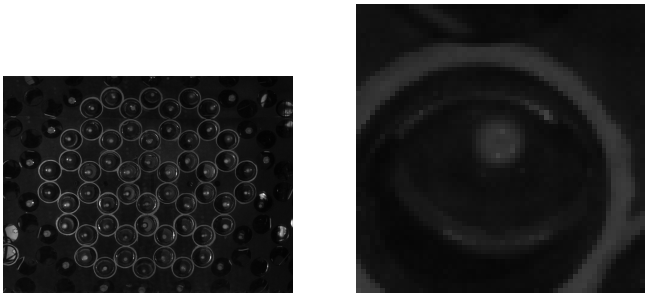


Fig. 1: Real-world object detection using bounding boxes, from [2]. Class labels and confidence scores accompany each box. Objects detected with $> 50\%$ confidence shown.

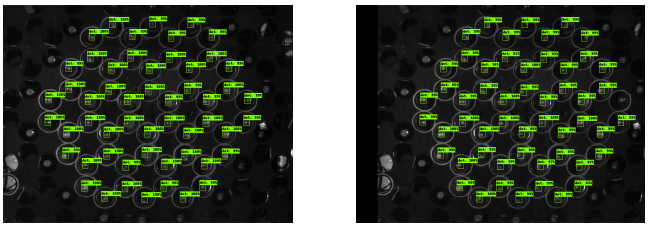
1 Introduction

Thanks to advances in neural network and computer processor design, machine learning methods have proven to be fast and effective for analysis of complicated images. Widely used datasets such as CIFAR and ImageNet are have been instrumental in this success, focusing on the classification of objects like "cat" or "dog" in a natural, or *real-world*, setting. More informative datasets, *e.g.* the Oxford-IIIT Pet Dataset in [1], include position information in the form of bounding boxes or segmentation masks, as well as class labels. Using such data, object detection networks can be trained to locate a range of objects in natural images. In particular, [2] discusses the tradeoffs between network accuracy and performance for these detections, as well as making available an "Object Detection API" through Tensorflow [3]. The API is capable of bounding box detection of multiple classes, as shown in Fig. 1.

The success of machine learning on real-world tasks suggests that similar techniques might be readily applied in laboratory conditions. Many experiments rely on detailed image analysis to obtain object location, which can require many man-hours for setup design or, as a last resort, pixel labeling. In this report, we offer a description of ongoing work which explores the application of Deep Neural Networks (DNNs) to laboratory image analysis, in an effort to reduce the necessary man-hours. In general, the primary obstacle here consists of obtaining training data with reliable ground truths. If these were easily obtained for a given experiment, then there would be no need for a machine learning approach.



(a) (b)



(c) (d)

Fig. 2: In (**a**), the full gyroscopic model for topological metamaterials from [4], where each individual gyro (**b**) is constrained to motion inside its circle. The center "dot" is the object being tracked. In (**c**), the network from [2] predicts bounding boxes for each gyro, using 16×16 boxes centered on each gyro's ground truth position for training. Although each box bounds its dot, it does not center itself around the dot, preventing location inference. The network is resilient to a translational shift (**d**) of 60 pixels. Note the lower confidence scores, with one gyro receiving $< 50\%$.

2 Experiment

In particular, we apply DNNs to detect the (x, y) image-space positions of 54 gyroscopes in the experimental setup from [4]. A single, 600×800 video was used, with 7742 frames extracted for the training set, 1000 for the test set. Fig. 2a shows a typical image from the full-image dataset, while Fig. 2b shows a cropped thumbnail of a single gyro, centered on its mean position. Each gyro is marked with a painted white dot and restricted to move within a constant circular area. This setup lends itself to traditional methods for object localization, *e.g.* gradient ascent or blob detection, which provided ground truth positions on which to train.

2.1 First Approach

Given the availability of box-detection schemes in [2], we first applied the same networks for gyro detection. We defined a training set with a single non-background class, "dot," and 16×16 boxes centered on the ground truth position of each gyro. As can be seen in Fig. 2c, the Object Detection API succeeds very well in bounding the gyros' positions, but it fails to predict boxes *centered* on those positions, making precise location inference impossible. Nevertheless, it is evident that the API, although designed for "real-world" tasks where precision is not expected, learned a valuable task for initial localization. Fig. 2d shows the networks' relative resilience to translational perturbations of the data, despite being trained on a non-perturbed dataset. From this point, more refined methods may be employed, inside each box, to infer precise gyro position. These include traditional methods, already mentioned, as well as the machine learning approach discussed below.

3 Regression using Convolutional Neural Networks on Small Thumbnails

3.1 Design

In order to recover pixel-level precision of the gyroscope positions, we applied a relatively shallow Convolutional Neural Network (CNN) [5] to several datasets of cropped images, or thumbnails, of a single gyro. The initial layer consisted of 32 different 5×5 kernels (over grayscale images), followed by four layers, each with 64 3×3 kernels. These were followed by two 2048 unit fully-connected layers and a 2 unit output layer. Unlike CNNs designed for classification tasks, which generally employ pooling layers to reduce the dimensionality of the image and to introduce translational invariance to the network, we retained full image dimension until the fully connected layer. Admittedly, this approach is computationally expensive for larger images, requiring the "first-shot" approximation that networks such as those in [2] provide. For this reason, we worked with the cropped datasets described below.

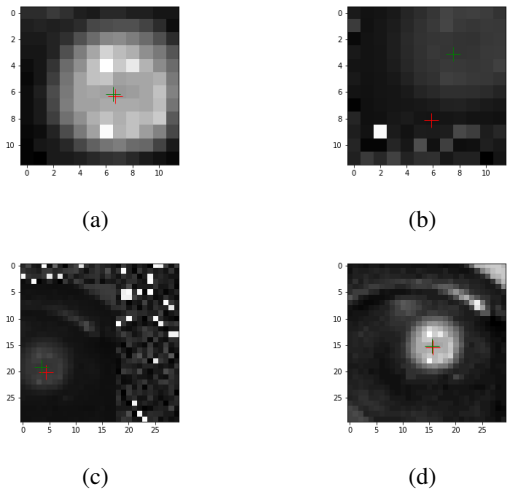
3.2 Datasets

In each dataset, we created one thumbnail from each image in the original video. Thus each training set consisted of 7742 training and 1000 test examples.

The first dataset, called the *centered* set, consisted of 12×12 thumbnails fixed on the mean ground-truth position of a single gyro in each. Since the gyro's dot is about 12 pixels in diameter, it nearly fills the image in every example, making the detection of its center a simple task. In the second dataset, we used the same thumbnails and imposed a random, uniformly distributed offset to create the 12×12 *perturbed* dataset. In each example, then, the ground-truth position of the gyro was off-center by up to 4 pixels in either direction. Additionally, any pixels outside of the original centered thumbnail were replaced with Gaussian noise, ensuring that the perturbations added no information not available to the centered dataset. Finally, we replicated this process for 30×30 thumbnails, both centered and perturbed. In the 30×30 perturbed set, the ground truth gyro position was off-center by up to 12 pixels in either coordinate.

3.3 Results

For both the 12×12 and the 30×30 datasets, we trained separate networks on the centered set and the perturbed set, then tested on both training sets. Fig. 3a shows the prediction of the network trained on the centered set. As can be seen, the network trained on centered data performs well on similarly centered data, with a mean error distance of 0.4 pixels across the test set (see Table 1). However, the network fails to learn from the desired features of the image, namely



(a) (b)

(c) (d)

Fig. 3: Isolated gyro images from the test set. Ground truth predictions are marked in green, inferred positions in red. The network trained on the 12×12 *centered* dataset (**a**) achieves sub-pixel accuracy relative to ground truths on the centered test set. The same network performs poorly (**b**) when tested on randomly offset thumbnails with random noise background. The 30×30 dataset with random perturbations and background noise (**c**) produces a resilient network that performs well on centered data, after learning on the perturbed set (**d**).

(a) (b)

Fig. 4: Test images from the perturbed dataset (**a**) show network performance for the gyro on which it was trained. That same network was tested on a perturbed test set from *a different gyro* (**b**).

the ability to recognize the gyro dot itself; it is incapable of recognizing the object far from the center of the image. The predictions on the perturbed test, as in Fig. 3b set have a mean error of 3.6 pixels, which is almost the maximum random offset in that test set. We hypothesize that this failure to learn the desired information results from a low variation in the training data; the ground truth positions had a variance of less than one pixel in either direction. As a result, the network learns to guess in a constrained region around the center of the image.

Fig. 3c, on the other hand, shows the results of training on the perturbed dataset, this time for the 30×30 dataset. The network performs with near-pixel precision on both the perturbed set, on which it was trained, and the centered set in Fig. 3d, which it has never seen before. (We say near-pixel, despite having sub-pixel average error, due to outliers.) In fact, it achieves better precision on both datasets than the networks trained and tested on 12×12 images (see Table 1). This suggests that the same network, rather than being inhibited by larger images, was able to learn more information from them.

Note, in particular, the precision of the network trained on the 30×30 perturbed set when tested on the centered set. In this case, the perturbations actually decreased the variance of predictions—from 0.2, when trained on centered data, to 0.1—despite having to generalize over a wider range of ground truths.

Finally, we tested the network trained on the 12×12 perturbed set on a *different* gyro, as shown in Fig. 4. Crucially, the network performs comparably on this new gyro, despite having never seen it during training. This suggests that a single network, trained on a dataset generated from one object, may be used to detect similar objects *without additional training*. The superior precision of the network trained on perturbed datasets suggests that the principles employed in generating this dataset significantly affect its performance. In this report, we explore the effects of translational perturbations, but we propose that other perturbations are essential for consistent precision in a laboratory experiment.

3.4 Ground Truth Precision

As can be seen in Figures 3 and 4, the ground truth positions of the gyroscopes (shown in green) are less than ideal. Intuitively, this can impede network generalization because it forces the network to learn special cases seen in the training set where the "true" position of the gyro is not its center. Interestingly, despite these limitations, we observed that the networks trained on the perturbed data tended to make errors more toward the apparent center of the blob, when tested on the centered data. That is, the predicted position, although offset from the provided ground-truth position of the gyro, was very often closer to the actual center of the gyro than the ground truth. This has potential implications for the training of networks with greater precision than the data they train on. However, it remains an area of exploration.

4 Artificial Dataset Generation

The usefulness of the machine learning approach in a laboratory setting is largely dependent on the availability of ground truths for training. In our data, from [4], careful construction of the experimental setup makes object positions available through the application of traditional methods, but in order to apply machine learning techniques to new experiments, well-representative ground truths must be available through less effort than traditional methods require. Here, we explore the generation of an artificial dataset from very few representative images, as they apply to the gyroscope setup in [4]. In this case, a single gyro could be precisely located in small, "seed" dataset, which would populate a larger dataset of generated images emulating the experimental setup. To the degree that objects being tracked are visually similar, one

Images	Training Set	Test Set	Mean Error Distance	Standard Deviation
12×12	centered	centered	0.3	0.2
	centered	perturbed	3.6	1.4
	perturbed	perturbed	0.8	0.4
	perturbed	centered	0.5	0.2
	perturbed	perturbed (new gyro)	0.8	0.4
30×30	centered	centered	0.4	0.2
	centered	perturbed	9.7	1.4
	perturbed	perturbed	0.6	0.3
	perturbed	centered	0.4	0.1

Table 1: Performance for various networks on individual cropped datasets, as in Figures 3 and 4. Networks trained on the perturbed dataset achieved sub-pixel accuracy.

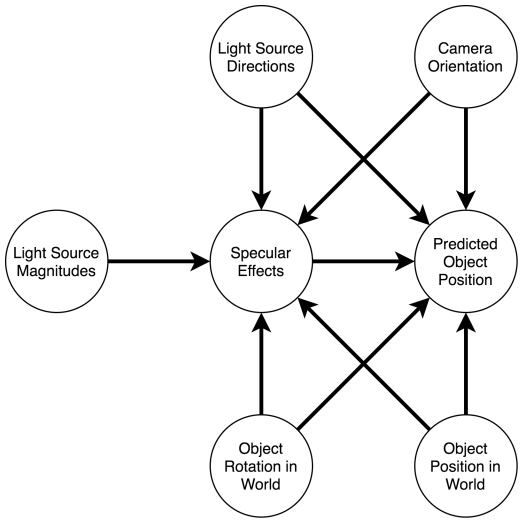


Fig. 5: A directed graph modeling dependencies in the experiment. We propose that the source nodes in such a graph determine which qualities should be perturbed in an artificial dataset.

seed set could be used to train many networks, each suited for a different experimental setup.

In Sec. 3.3, we explored the effects of translational perturbations on the dataset. This models one aspect of the experiment which we certainly want a network to be sensitive to, namely the actual position of the object in the image. However, the predicted position of an object depends on many factors in the experiment. In order to create a resilient network, these variables should be replicated in the artificial dataset. For example, changes in lighting can affect the precision of traditional blob detection; an artificial dataset, then, should include random perturbations to the image brightness in order to produce a resilient network. Other variables include bright reflections in the image (specular highlights), object rotation, and camera orientation. To the extent that these can be separated, we propose that an artificial dataset should include perturbations of each.

It is vital, then, that the variables in an experimental setup be well understood. Fig. 5 shows how different variables in the experiment depend on each other. The desired variable, "Predicted Object Position," has multiple dependencies, which are "sources" in the graph. The random translations in Sec. 3.2 amount to perturbations of "Object Position in World" in the 12×12 datasets, since that was roughly the size of the object. In the larger 30×30 datasets, translations more resembled a perturbation of "Camera Orientation," since the features surrounding the object were moved as well. A more comprehensive dataset would include both perturbations.

5 Conclusions

We explored the application of machine learning methods in the analysis of laboratory images from [4]. We acknowledged the difficulty involved in obtaining reliable ground truths for these methods and proposed, as a potential solution, the generation of artificial datasets through random perturbations of several variables. Despite imperfections in the ground truth positions, we showed in Table 1 that relatively shallow CNNs with no pooling layers can be remarkably effective for locating objects in small images. When combined with less precise bounding-box networks, such as those from [2], these CNNs can provide an end-to-end machine learning approach to position detection.

5.1 Future Work

Many aspects of laboratory image analysis with neural networks remains unexplored. We aim to more fully explore the possible perturbations of a seed dataset in order to determine which are most vital for the training of an effective network. Additionally, we acknowledge the problem of position verification for an experiment where, because a machine learning technique was applied with an artificial training set,

no ground truth positions are available. Here, we propose that the variance of pixel values may be obtained as function of distance from predicted position. A good prediction set, then, would have low variance close to the predicted position. This and similar methods may be employed to verify positional data once acquired.

Acknowledgements

We thank William Irvine for access to his laboratory and data, with special thanks to Noah Mitchell for continued support regarding the data. Finally, we thank Yali Amit for input and guidance.

References

[1] Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V., 2012. "Cats and Dogs". In IEEE Conference on Computer Vision and Pattern Recognition.

[2] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K., 2016. "Speed/accuracy trade-offs for modern convolutional object detectors". *arXiv:1611.10012 [cs]*, Nov. arXiv: 1611.10012.

[3] Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Man, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*.

[4] Nash, L. M., Kleckner, D., Read, A., Vitelli, V., Turner, A. M., and Irvine, W. T. M., 2015. "Topological mechanics of gyroscopic metamaterials". *Proceedings of the National Academy of Sciences,* **112**(47), pp. 14495–14500.

[5] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. "ImageNet Classification with Deep Convolutional Neural Networks". In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds. Curran Associates, Inc., pp. 1097–1105.