Object Detection for Semantic SLAM using Convolution Neural Networks

Saumitro Dasgupta
saumitro@cs.stanford.edu

1 Introduction

Conventional SLAM (Simultaneous Localization and Mapping) systems typically provide odometry estimates and point-cloud reconstructions of an unknown environment. While these outputs can be used for tasks such as autonomous navigation, they lack any semantic information. Our project implements a modular object detection framework that can be used in conjunction with a SLAM engine to generate semantic scene reconstructions.

A semantically-augmented reconstruction has many potential applications. Some examples include:

- Discriminating between pedestrians, cars, bicyclists, etc in an autonomous driving system.
- Loop-closure detection based on object-level descriptors.
- Smart household bots that can retrieve objects given a natural language command.

An object detection algorithm designed for these applications has a unique set of requirements and constraints. The algorithm needs to be reasonably fast - on the order of a few seconds at most. Since the camera is in motion, the detections must be consistent from multiple viewpoints. It needs to be robust to artifacts such as motion blur and rolling shutter. Currently, no existing object detection algorithm addresses all of these concerns. Therefore, our algorithm is designed with these requirements in mind.

In the past couple of years, convolutional neural networks have experienced a resurgence in popularity. They currently dominate the benchmarks for image classification and detection tasks [1]. This has motivated us to use it as the core of our detection framework.

2 Datasets

We used a number of datasets for developing our framework. While the primary dataset used in the final system was based on ImageNet, the rest were used for evaluating the "Network in Network" CNN architecture described in section 4.

2.1 ImageNet

The ImageNet dataset [2] is a collection of over 15 million labeled RGB images organized according to the nouns in the WordNet hierarchy. Currently, each node has on an average about 500 images. The associated ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been used for benchmarking image classification algorithms since 2010.

2.2 CIFAR-10

The CIFAR-10 dataset [3] is a labeled subsets of the 80 million tiny images dataset collected by Krizhevsky, Nair, and Hinton. It consists of 60,000 32x32 color images in 10 classes. There are 6,000 images per class. The first 50,000 images were used for training, while the remaining 10,000 comprised the validation set.

We pre-processed the data by performing ZCA whitening and global contrast normalization as described in [3].

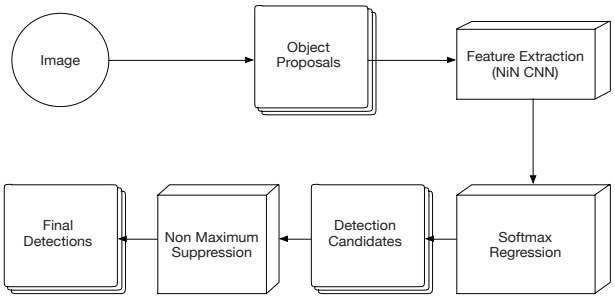

Figure 1: The object detection pipeline. The components shown above are described in section 3.

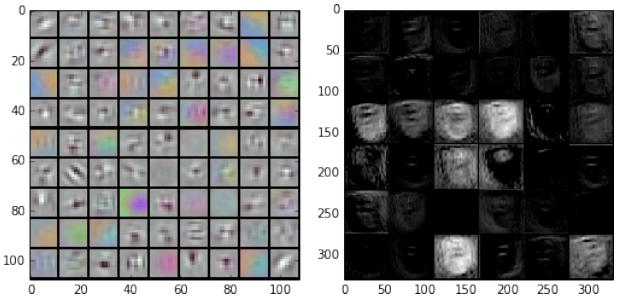

Figure 2: Visualization of the parameters from the first convolutional layer of the network, along with the corresponding (partial) feature map.

2.3 MNIST

MNIST [4] is a collection of 28x28 grayscale images of handwritten digits. It consists of 60,000 training and 10,000 test examples.

3 Framework Architecture

Figure 1 provides an architectural overview of our framework. Broadly, the components involved are:

1. **Object Proposals**: Since our network is trained on whole image classification, we need a way to re-purpose it for localized detections. One possibility would be to use a sliding-window approach at multiple scales. However, this would be too slow for our purposes. Therefore, we adopt the object proposal paradigm where an algorithm is used to generate bounding boxes for regions likely to contain an object. In particular, we use the recently published "Edge Box" proposal algorithm by Zitnick and Dollár [5]. It provides state-of-the-art level proposals while still being extremely fast.

2. **Feature Extraction:** For each proposal, we use a convolutional neural network trained on the ImageNet dataset to extract features from an RGB image. Figure 2 visualizes the parameters learned by the network along with the corresponding feature map. The model is described in greater detail in section 4.

3. **Classification:** The extracted features are fed to a softmax regression classifier to obtain the detection label.

4. **Non-Maximum Suppression:** Object proposals tend to yield multiple overlapping detections for a given object. We address this here by first finding bounding boxes with an IoU (intersection over union) score greater than a certain threshold (0.3 in our current implementation), and then retaining only the one with the highest score.

The resulting detections are then propagated to the SLAM engine for eventual localization in 3D. We exclude a discussion of the SLAM subsystem as it is beyond the scope of this report. For one potential approach, we refer interested readers to [6].

4 Model

A wide range of methods have been proposed for both object classification and detection. Up until recently, the dominant methods involved the use of hand-crafted feature descriptors such as SIFT and LBPs. However, in 2012, Krizhevsky et. al. [8] demonstrated that convolutional neural networks (CNNs) can be efficiently trained for achieving superior image classification results in the ImageNet Large Scale Visual Recognition Challenge. Since then, CNNs have dominated the image classification benchmarks. More recently, Girshick et. al [9] have shown state-of-the-

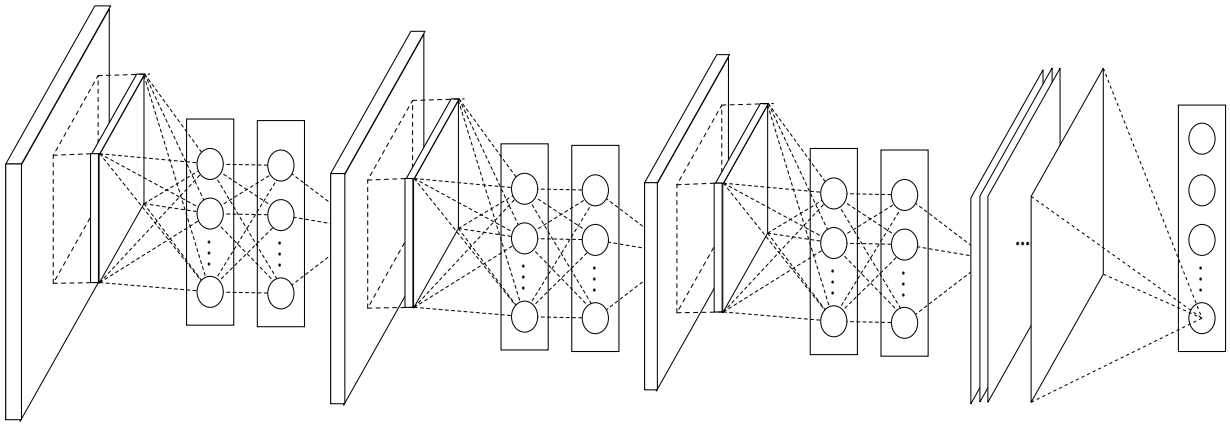

Figure 3: The "Network in Network" architecture, as described in [7]

art results for object detection using CNN features coupled with linear SVM classifiers. Therefore, we chose CNNs as the core of our detection framework.

Our implementation uses the novel "Network in Network" (NiN) architecture proposed by Lin et. al in [7]. The convolutional layer in the commonly used architecture described by Krizhevsky et. al. in [8] (often referred to as "AlexNet") uses a linear filter. The NiN model replaces this with a multi-layer perceptron (MLP) which slides over the input to produce the feature map. As MLPs are universal function approximators, this tweak results in greater abstraction capabilities over local patches. In addition, the fully connected layers present in AlexNet are replaced by global average pooling. This greatly reduces the number of parameters and makes it less prone to overfitting.

5 Results

5.1 Network in Network Benchmarks

Table 1 summarizes the test errors obtained by the "Network in Network" architecture on the datasets described in section 2. On the CIFAR-10 dataset it achieves state-of-the-art results, achieving a test error of 10.03 % without image augmentation. Interestingly, a NiN trained on ImageNet produces a much smaller

Dataset	Test Error
CIFAR-10	10.03%
MNIST	0.47%
ImageNet	59.36 %

Table 1: Benchmark results for the "Network in Network" architecture

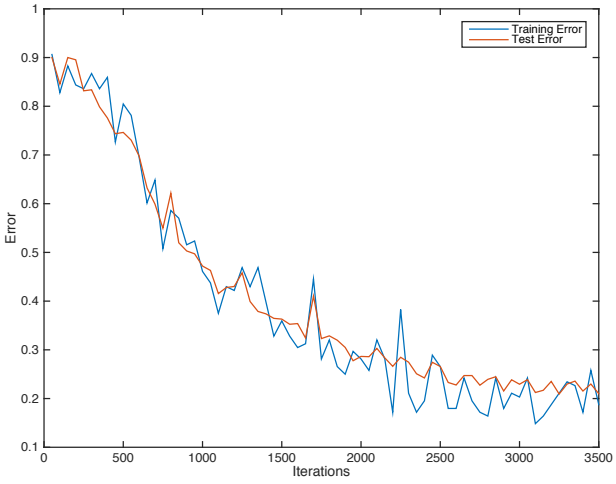

Figure 4: The training and test errors for the first 3500 stochastic gradient descent iterations on the CIFAR-10 dataset.

parameter set when compared to AlexNet (29MB vs 230MB when using Caffe's format [10]) while performing slightly better.

5.2 Object Detections

Figure 5 shows a series of frames taken from the TUM RGB-D SLAM dataset [11]. These frames capture roughly the same scene from multiple viewpoints. Each of them has been annotated with the bounding box of the detected object along with its label. The detector only considers 50 of the 1000 ImageNet classes for this sequence. Excluded classes include those that are unlikely to be stationary and/or encountered in an office environment (such as wildlife).

6 Discussion

6.1 Precision and Recall

The classification score threshold significantly affects precision and recall. An aggressive threshold reduces false positives but also suppresses true positives. On the other hand, a conservative threshold leads to improved recall, but reduced precision. In figure 5, the bottle in the first frame is an example of a false positive, whereas in the third frame, the power drill is missed. Our current implementation uses a hardcoded hand-tuned threshold for the TUM dataset. However, a dynamically-adjusted threshold would be a more robust solution.

Another significant hyper-parameter is the number of object proposals. We found that increasing the number of *Edge Box* proposals beyond 300 does not significantly influence the quality of the results. For comparison, R-CNN [9] generates 2000 proposals using the selective search algorithm.

6.2 Speed

The current implementation takes about 4.6 seconds to fully process a single frame. A component-wise breakdown is given in table 2. For comparison, R-CNN, the current state-of-the-art CNN based object detection algorithm [9] takes about 30 seconds on the same hardware. Furthermore, as our current implementation hasn't been optimized for speed yet, we expect its performance to improve.

7 Conclusion

In this report we described our implementation of an object detection framework that is suitable for use with a SLAM engine. We demonstrated that the Network in Network CNN model trained on the ImageNet dataset, coupled with Edge Box object proposals and non-maximum suppression provides fast and reasonably accurate results.

8 Future Work

The current implementation operates solely on 2D images. However, given that its designed to be used within a SLAM framework, it would be interesting to incorporate depth information into the detection process. Recent work along these lines have shown promising results [12].

We also plan to fine-tune tune our implementation using large datasets intended specifically for object detection, such as the ones published for the PASCAL VOC. We expect this to improve our detection performance, as well as provide an objective benchmark for evaluating our system.

Yet another interesting addition would be to incorporate bounding box regression, as described by Girshick et. al. in [9].

References

[1] Olga Russakovsky et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

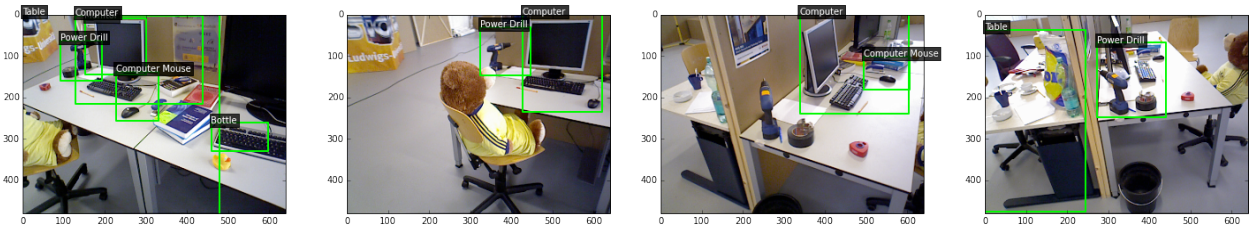

Figure 5: Frames from the TUM RGB-D SLAM dataset annotated with object detections.

Stage	Time (seconds)
Object Proposals	1.8
Feature Extraction + Classification	2.8
Non-Maximum Suppression	0.007
Total	r

Table 2: Time required by each component in the pipeline. The time quoted for "Object Proposals" include Edge Box proposal generation (which takes less than a second), as well as the overhead introduced by our implementation's (un-optimized) cropping algorithm.

[3] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 2009.

[4] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits, 1998.

[5] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.

[6] Jörg Stückler, Benedikt Waldvogel, Hannes Schulz, and Sven Behnke. Dense real-time mapping of object-class semantics from rgb-d video. *Journal of Real-Time Image Processing*, pages 1–11, 2014.

[7] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[10] Yangqing Jia et al. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[11] Jürgen Sturm, Stéphane Magnenat, Nikolas Engelhard, François Pomerleau, Francis Colas, W Burgard, D Cremers, and R Siegwart. Towards a benchmark for rgb-d slam evaluation. In *RSS*, volume 2, page 3, 2011.

[12] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.