Research Proposal:
Revisiting Grocery Recognition using TensorFlow

Dilip Thiagarajan, Samuel C. Hoffman

June 20, 2016

1 Introduction

As the need for assistive technology for the visually impaired becomes more prominent and feasible with the use of machine learning, we revisit the problem of using pictures of objects taken in ideal conditions to recognize more common scenes of these same objects. This problem encompasses several possible applications, but in this study, we will specifically look at grocery products sold in Mattin's, a local café on the Cornell campus in Ithaca. In essence, we will revisit the study conducted by Merler, Galleguillos, and Belongie [7] but with a more modern approach involving the tools contained in TensorFlow [1].

The main problem we hope to address by conducting this study is paramount to maintaining the state of assistive technology: ideally, training data and testing data would be obtained from the same distribution of data. However, in real-world application, it is more convenient to obtain training data from more well-kept databases, such as the web or dedicated databases. As a result, using the tools offered in TensorFlow, we intend to design a portable system that can recognize the groceries in Mattin's, trained on images taken from both an ideal and realistic environment.

Specifically, the purpose of this study will be to build a database of images encompassing the inventory of Mattin's ranging from ideal to realistic shots, as well as to use various approaches to actually recognize the grocery products in Mattin's. This will include color histogram matching, SIFT matching, compared with training a neural network that takes image pairs as input (where the image pair consists of an ideal and realistic shot of a grocery object in Mattin's) for classification.

Our ultimate goal in beginning this study is to achieve real-time recognition and localization from a camera.

2 Data and Methodology

To begin, we will need to obtain the relevant types of data for input. As mentioned in the previous study, one database that can be used is the GroZi-120 database, which is a database of 120 products with images of objects, ranging over various attributes of the image, and where each product has two different representations: either *in situ*, i.e. in a realistic environment, or *in vitro*, i.e. in an idealistic environment. Another possible way to scrape data could be to use Google searches, and take a sample of the top image hits as training/test data.

Additionally, data would be obtained from Mattin's in person, and due to the changing inventory, images and videos can be taken periodically of each product in the whole inventory to better recognize the

inventory, given the varying nature of the products sold. To better train the network being used, photos and videos taken in person will vary in scale and lighting to simulate a more realistic environment.

Once the relevant data has been obtained, it will be used independently on each localization and recognition algorithm, as well as arranged into pairwise input to be used for training on a convolutional neural network built in TensorFlow. Specifically, we will build a sample of several neural networks built in TensorFlow (which will be chosen from the set of all possible networks by using distinctive heuristics) and see how they perform compared to each other on various samplings of training and testing data. Once we have selected the network that performs best, we will compare the results of the standard algorithms with the new accumulation of data to the results in the previous study, as well as comparing how the chosen network built in TensorFlow performs compared to other algorithms on the new accumulation of data.

3 Related Works

Given that the scope of computer vision has drastically changed over the past decade with the introduction of deep learning with neural networks, a lot of work has been done towards building tools and studying specific applications. With the ultimate goal of generalized, fine-grained image classification in mind, these tools and studies would prove useful in our own study.

3.1 Relevant Computer Vision Tools

As mentioned, deep learning has revolutionized the landscape of computer vision. Nonetheless, there were some frameworks which allowed for image classification, including things like OpenCV in Python, VLFeat, VLX libraries written in C++, and tools like WEKA, but with the advent of deep learning in the past few years, and with the progress they've made on well-known datasets, such as MNIST, CIFAR-10, and ImageNet, we will proceed with using these deep learning frameworks as opposed to the libraries that preceded them.

Several deep learning frameworks have been introduced in the past few years, including libraries such as Theano, Torch, Caffe, and TensorFlow. Our choice of using TensorFlow is somewhat arbitrary, although given it's recency, it does offer several advantages over the other frameworks. Lipton brings up several of these points in his analysis of TensorFlow against other deep learning frameworks.

While TensorFlow is not fundamentally different from Theano or Torch, and Caffe was essentially developed specifically for implementing convolutional neural networks, which is what will be used throughout this study, TensorFlow allows the user to write in Python, which is arguably much simpler to handle than Lua, C++, or Cuda. While Caffe also has Python bindings for running models, it's very inconvenient to define new models or different kinds of layers without adding a large portion of C++ code to their codebase [6].

In addition, because Python is the main language of use, TensorFlow allows for integration with any other workflow necessary in the grand scheme of the study, whether it be a web server to demonstrate the application, or the use of some open-source scientific computing tool (especially libraries like NumPy and Scikit-learn) [6]. Moreover, TensorFlow works seamlessly when running with multiple GPUs, and the code compiles in very little time.

3.2 Relevant Studies

Our work will build heavily on the study by Merler et al. (2007). Other attempts to improve upon the results of the GroZi baseline include a paper by George et al. (2015) who used a text-based approach for recognizing

classes [3]. We believe we can further improve upon this with a full deep learning architecture.

The use of deep convolutional neural networks (ConvNets) has revolutionized the field of object recognition and state-of-the-art networks have even surpassed human-level performance in some areas [4, 9]. We hope to leverage these breakthroughs to create a network which can discriminate between specific products (in this case, those sold at Mattin's). However, competitions such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which have pushed the limits of ConvNets, are usually focused on general recognition of more coarse-grained categories whereas our goal is much more fine-grained so we must find a way to adapt these broad classifiers to our needs.

One approach is to train a manageably-sized network from scratch on our data set. This allows us to learn deep features important to our classes while still being able to iterate quickly. Since the trend of the past couple years has tended toward larger and more complex networks which may take days to train, we could choose a slightly older but still powerful model such as AlexNet, the ILSVRC winner from 2012 [5].

Another approach is to fine-tune a state-of-the-art network by replacing the last layer and retraining with our data, starting with the fully-trained weights for the other layers. The early layers in most ConvNets have general functions such as blob detection which are useful for many different applications. If we start out with a network which can identify many higher-order features and then train it to learn class-specific features, we can achieve a very powerful model without spending so long training from scratch [2]. The drawback is that we may need to fine-tune multiple layers to get the best results which, for a state-of-the-art network, may still be a slow process.

We will be benchmarking both approaches and comparing them to the previous baselines mentioned above.

Another important aspect of this project is localization of detected objects in the video frame. Again, there have been great strides recently in this area toward real-time detection with localization particularly using Region Proposal Networks (RPNs). State-of-the-art systems have achieved near-real-time speeds of 5 fps with very deep networks which is fast enough for our purposes [8].

4 Milestones

Completed:

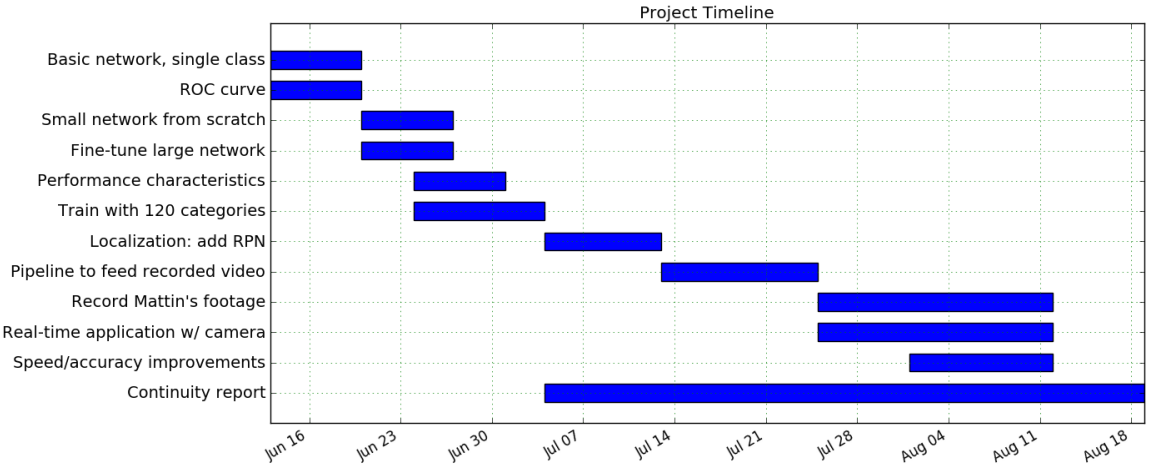
- Train basic network on a single class (Tide laundry detergent)
- Output basic performance characteristics (ROC curve, precision/recall)

To do:

- Train manageably-sized network (such as AlexNet) from scratch on a single class
  - *Dilip: June 20–June 27*
- Fine-tune a large network for a single class
  - *Sam: June 20–June 27*
- Produce framework for outputting performance characteristics for any underlying architecture
  - *Both: June 24–July 1*

- Generalize both approaches above to all 120 categories in GroZi-120
  - *Both: June 24–July 4*
- Incorporate a Region Proposal Network for localization of objects in a single, full video frame
  - *Both: July 4–July 13*
- Create a framework for applying the full workflow on recorded video of store shelves from the GroZi dataset
  - *Both: July 13–July 25*
- Record shelf footage in Mattin's Café and test system
  - *Both: Late July/Early August*
- Incorporate framework into an application using real-time camera data
  - *Both: July 25–August 12*
- Make improvements to recognition network architecture for speed or accuracy as necessary
  - *Both: August 1–August 12*
- Produce a continuity report for reference in future work
  - *Both: July 4–August 19*

Figure 1: Gantt chart depicting the tentative timeline



References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[3] M. George, D. Mircic, G. Sörös, C. Floerkemeier, and F. Mattern. Fine-grained product class recognition for assisted shopping. In *ICCV Workshops*, pages 546–554. IEEE, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[6] Z. Lipton. Tensorflow is terrific – a sober take on deep learning acceleration, 2015. Accessed 20-June-2016.

[7] M. Merler, C. Galleguillos, and S. J. Belongie. Recognizing groceries in situ using in vitro training data. In *CVPR*. IEEE Computer Society, 2007.

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, et al. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.