**1. Definition**

**Large-scale Identification of Multiple Digits**

**From Real-World Images with Deep Convolutional Networks**

Ritchie Ng

National University of Singapore

ritchieng@u.nus.edu

**1.1 Overview**

This project explores how deep convolutional neural networks (ConvNets) can be used to effectively identify a series of digits from real-world images that are obtained from “The Street View House Numbers (SVHN) Dataset” (Netzer, Wang and Coates). ConvNets have evolved dramatically every year since the inception of the ImageNet Challenge in 2010.

A proverbial ConvNet structure is the “LeNet-5” that has relatively few layers of convolutions, poolings, and full connections (LeCun, Bottou and Bengio). Subsequently, with the advent of the ImageNet Challenge, we are experiencing a gradual trend towards deeper ConvNets with more layers and higher accuracy such as AlexNet (Krizhevsky, Sutskever and Hinton), ZFNet (Zeiler and Fergus), VGGNet (Simonyan and Zisserman), GoogLeNet (Szegedy, Liu and Jia), and the ResNet (He, Zhang and Ren) being the latest state-of-the-art implementation of ConvNets.

To this point, we used VGGNet’s (Szegedy, Liu and Jia) framework to determine our optimal model for identifying multiple digits from real-world images.

**1.2 Problem Statement**

We are attempting to predict a series of numbers given an image of house numbers from the SVHN dataset. An important thing to take note is that instead of the standard identification of numbers, as with the MNIST dataset, we now need to correctly detect the numbers and the sequence of numbers.

**1.3 Metrics**

Instead of the usual measure of accuracy where we divide the sum of true positives and true negatives over all our predicted classes, we used two other metrics that are more commonly used in the community today as they provide a better measure of a ConvNet’s performance compared to a simplistic metric like accuracy.

First, we used top-1 error to evaluate our deep ConvNet’s performance. It is similar to accuracy but checks if the top class, the one with the highest probability, matches the target label, i.e. this calculates the proportion of incorrectly classified images.

Second, we used top-5 error where it checks if the top five classes, the five classes with probabilities in descending order, match the target label, i.e. this calculates the proportion of incorrectly classified images in the top-5 category.

Moreover, we have three main sets of data: our training, validation and test sets. We calculated our top-1 and top-5 error using our test set and predictions.

**2. Analysis**

2.1 Data Exploration

2.2 Exploratory Visualization

2.3 Algorithms and Techniques

2.4 Benchmark

3. Methodology

3.1 Data Pre-processing

**3.2 Implementation**

Our programming language of choice is Python 2.7 and we will be using Keras with the backend as TensorFlow to build our deep ConvNets.

The models’ computations are accelerated with a CUDA-enabled GPU, NVDIA’s GeForce GT 750M with 2GB of GDDR5, on a local machine with 16GB of RAM. Hence, there was sufficient RAM to ensure there was no bottleneck in our data transfer from our main memory (RAM) to our GPU’s memory (VRAM) for numerical computation with the GPU’s 384 CUDA Cores.

3.3 Refinement

4. Results

4.1 Model Evaluation and Validation

4.2 Justification

5.0 Conclusion

5.1 Free-Form Visualization

5.2 Reflection

5.3 Improvement: ResNet