# House Numbers Identification from Images with CNN

Fangling Zhang

## Abstract

Convolutional Neural Networks (CNN) are proved to be effective in digits identification such as MNIST dataset. However, multi-character text recognition in photographs is still highly challenging with its complexity and difficulty. In this paper, we focus on recognizing multi-digit numbers from Street View. I set up a unique multiple layers networks (3 convolutional layers in hidden layers) to complete the task and experiment some parameters, methods latter. With our best configuration, achieve over 91% accuracy in recognizing street numbers in the publicly available Street View House Numbers (SVHN) dataset.

## 1. Introduction

Identifying street view house numbers is an important part of modern-day map making. More broadly, recognizing numbers in photographs is a problem of interest to the optical character recognition community. While OCR on constrained domains like document processing is well studied, arbitrary multi-character text recognition in photographs is still highly challenging.

Digits identification here is different from standard identification of numbers. There are two main difficulty to identify the digits: I need to identify the digits from real world images, and some images include a sequence of numbers. These difficulty arise due to the wide variability in the visual appearance of text in the wild on account of a large range of fonts, colors, styles, orientations, and character arrangements. The recognition problem is further complicated by environmental factors such as lighting, shadows and occlusions as well as by image acquisition factors such as resolution, motion, and focus blurs.

In this paper we propose a unified model via the use of a deep convolutional neural network to complete task. This model is configured with multiple hidden layers (our best configuration had eleven layers, but our experiments suggest deeper architectures may obtain better accuracy, with diminishing returns).

### 1.1 research contribution

The key contributions of this paper are: (a) a unified model to localize, segment, and recognize multi-digit numbers from street level photographs; (b) the model use 5 parallel softmax layers before the output layer; (c) empirical results that show this model performing best with a deep architecture (d) results of applying proposed model on the SVHN images to achieve 91.62% test accuracy.

## 2. Related Work

Convolutional neural networks have previously been used mostly for applications such as recognition of single objects in the input image. In some cases they have been used as components of systems that solve more complicated tasks. Girshick et al. (2013) use convolutional neural networksas feature extractors for a system that performs object detection and localization. Many papers show that MNIST identification can get high accuracy with Convolutional Neural Networks (CNN). However, the system as a whole is larger than the neural network portion trained with backprop, and has specialmcode for handling much of the mechanics such as proposing candidate object regions. Szegedy et al.(2013) showed that a neural network could learn to output a heatmap that could be post-processed to solve the object localization problem. In our work, we take a similar approach, but with less post-processing and with the additional requirement that the output be an ordered sequence rather than an unordered list of detected objects. Alsharif and Pineau (2013) use convolutional maxout networks (Goodfellow et al., 2013) to provide many of the conditional probability distributions used in a larger model using HMMs to transcribe text from images. In this work, we can use

similar method to identify SVHN dataset. However, as SVHN dataset include more complex images and number sequence in single image compared to MNIST dataset, we need to adjust the structure and parameters we used for MNIST identification.

## 3. Proposed Method

CNN preserve the spatial relationship amongst pixels, learning internal feature representations where they are used across the whole image. Hence a digit, for instance, can be blurred, shifted to the left or right, or even distorted and the CNN can still detect the digit.
Currently, CNN are used for object recognition in images and videos with state-of-the-art results. In this project, I use 6-layers network (3 convolutional layers in hidden layers) to identify SVHN dataset.
I build deep CNN with Python and TensorFlow, Other substantial packages that are needed to run the code are shown in the following figure.

```
import os
import sys
import tarfile
import tarfile
import tensorflow as tf
import numpy as np, h5py
from __future__ import print_function
from six.moves import cPickle as pickle
from six.moves import range
from six.moves.urllib.request import urlretrieve
from scipy import ndimage
from PIL import Image
from numpy import random
```

In this first trial, I started with a simple model comprising the following layers.

**Table 1:** *CNN Topology: Trial 1*

| Layer |
|---|
| Input Layer |
| Convolution 1 |
| ReLU |
| Max Pooling |
| Convolution 2 |
| ReLU |
| Max Pooling |
| Convolution 3 |
| ReLU |
| Dropout |
| Fully Connected Layer (FC) |
| Softmax Layer |

I use parallel softmax layers after the fully connected layer. This idea is inspired by the recommendation on tackling multiple digits in the paper “Reading digits in natural images with unsupervised feature learning”. In this paper, they used 6 parallel softmax layers instead of5 as they included the length of the digits. The specific description of each layer in the above figure are as follows:

**Table 2:** *CNN Description of each layer: Trial 1*

| Description |
|---|
| Numpy array of size (230070, 32, 32, 1) |
| Filters: 16 \| Receptive Field: 5 x 5 \| Stride: 2 \| Padding: Valid |
| Rectified Linear Unit Activation |
| Receptive Field: 2 x 2 \| Stride: 2 \| Padding: Valid |
| Filters: 32 \| Receptive Field: 5 x 5 \| Stride: 2 \| Padding: Valid |
| Rectified Linear Unit Activation |
| Receptive Field: 2 x 2 \| Stride: 2 \| Padding: Valid |
| Filters: 96 \| Receptive Field: 5 x 5 \| Stride: 2 \| Padding: Valid |
| Rectified Linear Unit Activation |
| Keep probability of 0.5 |
| Nodes: 64 |
| 5 softmax (readout) activation layers for 5 digits |

Moreover, the weights are initialized with special functions I created so it is easier to expand the convolutional networks and have access to TensorBoard’s visualizations. The weights are initialized with a random normal distribution with a standard deviation of 0.01.
For my optimizer, I used Stochastic Gradient Descent (SGD) with a step-wise decay with a starting learning rate of 0.05. In the equation as shown in Equation 2, I used step-wise decay, so global-step/decay-steps would return an integer.
Importantly, unlike running a ConvNet on the MNIST dataset or majority of the datasets, our loss function is a combination of the parallel softmax (readout) activation layers.

## 4. Experiment

I tried some parameters or methods with the above 6 layers networks. Here, I just give one leads to highest test accuracy.
First, I changed my dropout probability to 0.8 before fully connected layer.
Second, I changed my optimizer from SGD to AdaGrad. This is because our data is quick sparse for digits with many numbers and AdaGrad is well-suited for such a

dataset because it adapts the learning rate to the parameters performing larger updates for infrequent parameters and smaller updates for frequent parameters.

Third, I changed the initialization of the weights using Xavier Initialization. It is a sampling of Gaussian distribution where the variance is a function of the number of neurons as shown in the following custom functions I made where the first function is for initializing weights for the convolution layers and the second function is for the initialization of weights for the FC layer after all the convolutions.

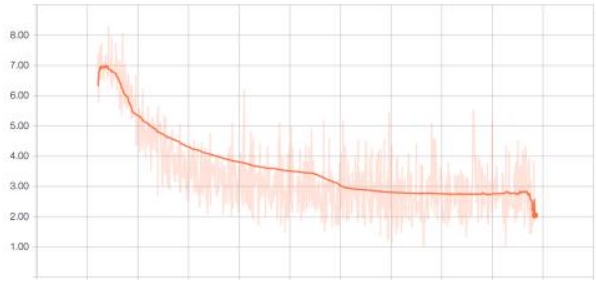
**Table 3:** *CNN Topology: Trial 2*

| Layer | Description |
|---|---|
| Input Layer | Numpy array of size (230070, 32, 32, 1) |
| Convolution 1 | Filters: 16 \| Receptive Field: 5 x 5 \| Stride: 2 \| Padding: Valid |
| ReLU | Rectified Linear Unit Activation |
| Max Pooling | Receptive Field: 2 x 2 \| Stride: 2 \| Padding: Valid |
| Convolution 2 | Filters: 32 \| Receptive Field: 5 x 5 \| Stride: 2 \| Padding: Valid |
| ReLU | Rectified Linear Unit Activation |
| Max Pooling | Receptive Field: 2 x 2 \| Stride: 2 \| Padding: Valid |
| Convolution 3 | Filters: 96 \| Receptive Field: 5 x 5 \| Stride: 2 \| Padding: Valid |
| ReLU | Rectified Linear Unit Activation |
| Dropout | Keep probability of 0.8 |
| Fully Connected Layer (FC) | Nodes: 64 |
| Softmax Layer | 5 softmax (readout) activation layers for 5 digits |

## 5. Results

With the model CNN trail 1, the loss decrease to almost 2.0 and it is interesting to note how the loss seems to plateau and they are very erratic as seen in Figure1.

**Figure 1:** *Loss of trail 1 Tensor Board graph*



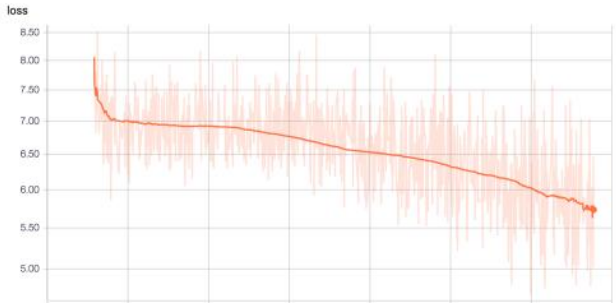

In the first trial, the test accuracy is 87.64% and the training and validation accuracies are shown below.

```
Minibatch accuracy: 90.0%
Validation accuracy: 87.2202674173%
Test accuracy: 87.6400367309%
```

In the second trial, I managed to get a test accuracy of 91.62% with the following test and validation accuracies.

```
Minibatch accuracy: 96.25%
Validation accuracy: 90.5594651654%
Test accuracy: 91.6176920722%
```

**Figure 2:** *Loss of trail 2 Tensor Board graph*




Interestingly, unlike previously, the loss function is not plateauing and there seem to be more room for learning as shown on Figure 2.

## 6. Discussion

I just complete several experiments and our empirical results suggest that models with a deep architecture can get better modern perform. So in the future, we may try deeper networks to get higher test accuracy on the numbers identification with SVHN dataset.

## 7. Conclusions and Future Work

Our paper set up a unified model to localize, segment, and recognize multi-digit numbers from street level photographs. The model use 3 convolutional layers and 5 parallel softmax layers before the output layer. Our empirical results that show this model performing best with a deep architecture. Our results of applying proposed model on the SVHN images to achieve 91.62% test accuracy.

The accuracy of the second model can be further improved with more training as it has yet to converge. Also, more investigation can be done to look at the large variation in our loss function.

I did not use any renowned topology highlighted in the introduction such as VGGNet or ResNet to keep this project focused on how to classify multiple digits. This particular classification of multiple objects in a single image is lacking in examples online hence it provides a base for people to explore how we can use proven topologies such as ResNet on this problem while maintaining the parallel readout layers to classify multiple digits.

More importantly, there is room for improvement by using existing weights trained on the ImageNet dataset using ResNet or other winning topologies and change the readout layer to the parallel readout layers. This

may give a substantial boost to our test accuracy and reduce our training time. This method is recently coined as transfer learning.

## References


Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 12 2015.

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.





Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 09 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. 09 2014.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 11 2013