Implementation of Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks

Hari S R, Atul Parthasarathy

Abstract—**As a part of our Machine Learning Course (BITS F464), we were given the task of recognizing arbitrary multi-digit numbers from Google Street View imagery. In this paper we have taken the approach as implemented in [1]. We have used a deep convolutional neural network that operates directly on the image pixels. We have used TensorFlow as the primary library for the implementation. Our best model with six hidden layers had obtained an accuracy of 83% on the cross validation set in recognizing complete street numbers.**

Keywords—TensorFlow, ReLU, tanh, softmax, Convolution, Max Pool

I. INTRODUCTION

Recognizing multi-digit numbers in photographs captured at street level is an important component of modern-day map making. A classic example of a corpus of such street level photographs is Google's Street View imagery comprised of hundreds of millions of geo-located 360 degree panoramic images. The ability to automatically transcribe an address number from a geo-located patch of pixels and associate the transcribed number with a known street address helps pinpoint, with a high degree of accuracy, the location of the building it represents.

In this paper, we focus on recognizing multi-digit numbers from Street View panoramas using the approach suggested in [1]. Traditional approaches to solve this problem typically separate out the localization, segmentation, and recognition steps. But the approach suggested in the [1] as that of a unified approach that that integrates these three steps via the use of a deep convolutional neural network that operates directly on the image pixels. The neural network with the best configuration we have come up with 6 convolutional layers and 2 fully connected layers. We have evaluated this approach on the publicly available Street View House Numbers (SVHN) dataset and achieve close to 83% accuracy in recognizing street numbers on the cross-validation set. Our results show the validity of the claim of that performance of the approach increases with the depth of the convolutional network as stated in [1].

II. PROBLEM DESCRIPTION

A. Basic Idea

The given problem is a Computer Vision problem that involves Classification and Localization. We need to localize each digit in the image and then classify to which class it belongs to, here the classes being the digits 0 to 9. There is also the problem of identifying the sequence of the numbers as in the image.

B. Evaluation of the Predictions

When determining the accuracy of a digit transcriber, we compute the proportion of the input images for which the length $\boldsymbol{n}$ of the sequence and every elements of the sequence is predicted correctly. There is no "partial credit" for getting individual digits of the sequence correct. One special property of the dataset is that length of the sequence is bounded, and in [1] they have limited with $\boldsymbol{n}$=5, where we have taken $\boldsymbol{n}$=6.

III. PROPOSED METHOD

A. Preprocess

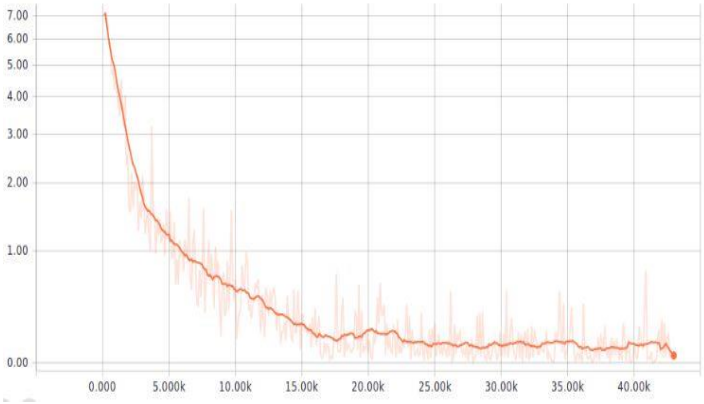
We have preprocessed the data as suggested in [1]. First we found first find the bounding box of the entire sequence with the given data. We then expand this bounding box by 15% along the height and the width. After that we crop the image to 64×64 pixels, from which we take a random crop of size 54×54 pixels.

B. Model

- The random crop is passed as an input to the Deep Convolutional Network, which consists of 6 hidden layers of convolutions and max pooling, each using filters of different sizes. The activation used in these layers is ReLU. We have also done batch normalization. After these 6 layers, the output is of the order of 4×4×256, where 4×4 is the image size and 256 is the number of filters. This passed to 2 fully connected layers where the activation used is ReLU and tanh respectively.
- The output from this layer is given to 7 softmax function, 1 for the Length and 6 for the individual digits.
- Predictions are made by taking the argmax of the output of the each softmax. The loss we have used calculates the sum of cross entropy losses for each the softmax function.

IV. Training

We have trained the deep convolutional neural network by taking random samples in a batch size of 32 from that training dataset. We then infer on this batch by using model and calculate the loss. The optimizer we have used is Adam with a learning rate of 10^{-4}. We train the model by calculation the loss after 100 batches. After 1000 batches approx. 1 epoch we calculate the accuracy on the validation set. This process is in an infinite loop. We have incorporated an idea of patience in the training. Patience is an indicator of when the training should stop i.e. the number of epochs to wait before early stop if no progress on the validation set. If the accuracy of the model on the cross validation set decreases after one epoch, the Patience decreases by one suggesting that the training time should be reduced. We have set patience to 10 i.e. the training ends only if the accuracy decreases for 10 epochs in a row.



V. Results

We have trained the 4 different models by changing various hyper parameters to obtain better accuracy on the cross validation set. We initially started of the model suggested in the TensorFlow MNIST tutorial, a 4 layer Deep Convolutional Neural Network which contains 2 convolutional layers and 2 fully connected layers. This model was not capable of finding any correct sequence. We then trained a 6 layer Deep Convolutional Neural Network which contains 4 convolutional layers and 2 fully connected layers. The accuracy obtained as 78%. Lastly we trained an 8 layer Model. Initial Accuracy with the proposed model as 83%. We decided to change the activation in the hidden layers from ReLU to eLU, but the accuracy of the model went down to 76%.

A. *Figures and Tables*

TABLE I. Accuracies of the Different Models

Model	**Properties of the Model**			
	Layers	***Convolutional***	***Fully Connected***	***Accuracy***
1	4	2	2	0
2	6	4	2	78%
3	8	6	2	83%
4	8	6	2	76%

Acknowledgment

This project was supported by Ashwin Srinivasan, IC, BITS 464. We thank our TA Rajat Agarwal who provided insight and expertise that greatly assisted the project, although they may not agree with all of the interpretations/conclusions of this paper.

References

[1] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet, "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks"