

Cyclops : A CNN based number recognition tool

I. INTRODUCTION

Identification of text in natural environments is a difficult problem. Convolution neural networks have shown promising trends towards prediction of text and this work will explore the prediction of digits of various lengths using the SVHN dataset. We will use TensorFlow as the backend for building the convolution neural network with Keras.

II. DATA

The dataset we use is The Street View House Numbers (SVHN) dataset. It consists of images with labeled digits and we have bounding box information for each digit.

A.Description

We have the following counts for numbers in the data set.

Digits	Number
1	5137
2	18130
3	8691
4	1434
5	10

B.Preprocessing images

The preprocessing involves stitching together individual bounding boxes and generating numbers of various lengths. We assume that the largest sequence in this data set is of length 5. We use vectors of length 6 as the output for each image with the first 5 elements denoting the digits and the last element denoting the length of sequence. All images are resized to 50x50. This was performed using the Python Image Library using a cubic spline interpolation.



C.One hot encoding of target

The target vectors are initially of dimension 6 with the first 5 elements representing the number and the 6th element representing the sequence length. '0' is represented by 10 and a 0 in the target vector represents that the digit is not present.

target = [4, 5, 4, 0, 0, 3]

The target vectors are one hot encoded and the vectors of length 6 are now converted into matrices of shape 6 x 11. Each digit is now represented by a 11 dimensional vector.

D.Data augmentation

Since the proportion of data was skewed, we generate more data for 3, 4 and 5 digit numbers for training



III. MODEL I: MINI

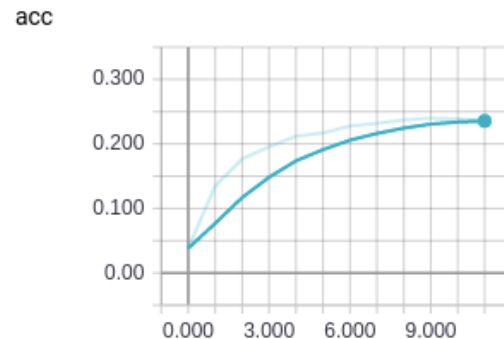
We use a simple model for the initial prediction where we input all the images and try to predict the digits and length of the sequence simultaneously.

Digits	Number
1	5137
2	18130
3	26821
4	10124
5	1434

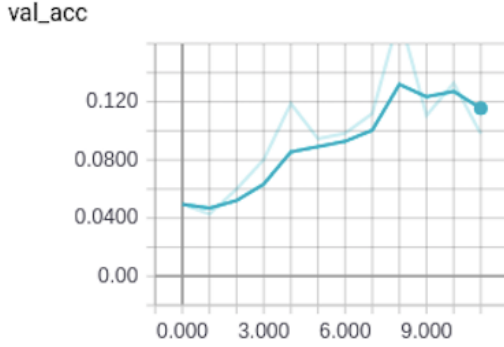
A.Architecture

2 Convolution layers
1 dense layer

B.Results



The validation accuracy plots show that the training is not smooth.

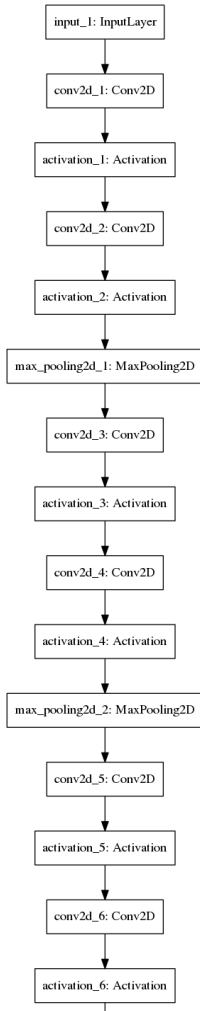


IV. MODEL II: FORK

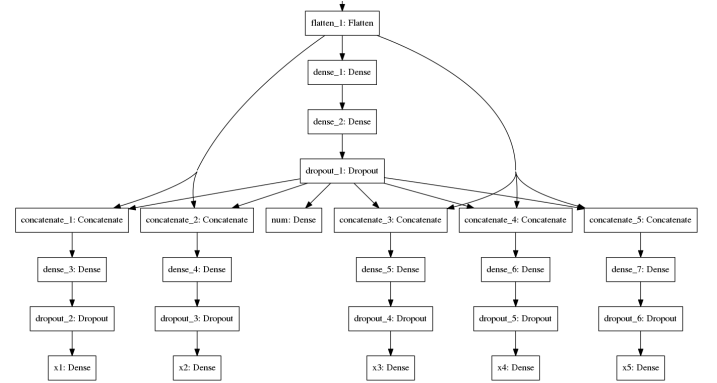
We now have 6 different softmax classifiers which learn feature from the same convolution neural network. One of the classifiers - numtower is for identifying the number of digits and the output of that is taken as one on the inputs by 5 different classifiers which predict digits at each position.

A. Architecture

The first part of the model has the following architecture:

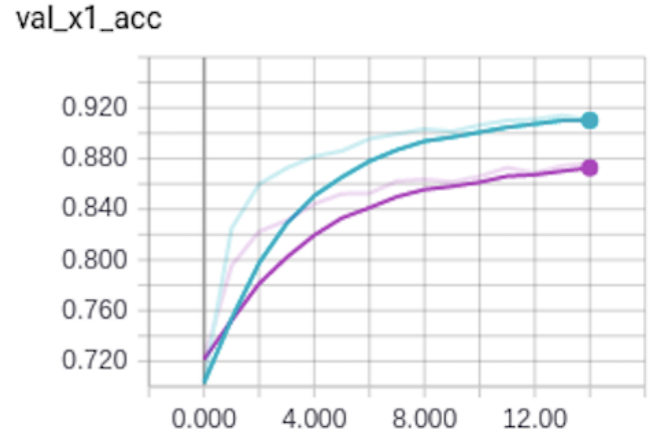


The second part of the model has the following architecture:



B. Tuning hyperparameters

We have 4, 6 and 8 convolution layers and compare the per-



formance in predicting the first digit.

The model with 6 convolution layer gives the best performance and we will select that for our model.

B. Results

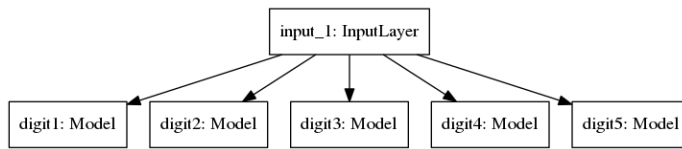
We get an accuracy of 95% for the number tower, which predicts the number of digits in the image. The individual accuracies for all digits are the following

Digits	Accuracy
1	
2	
3	
4	
5	

The final accuracy for the full prediction is

V. ION CANNON: INDEPENDENT TRAINING FOR DIGITS

proportion of images of each sequence length.



A.Architecture

A.Results

VI. THE COMPLETE PIPELINE

We use the model - fork for the final testing. It gives an accuracy of 81% on the validation set.

A.Image preprocessing

We obtain cropped images of various sizes which contain number of varying sequences. In order to pass it to our model we resize all images to 50x50 using a cubic spline interpolation.

B.Prediction

The resized images are then analysed and the model predicts the values of all 6 elements of the output vector - the digits and the sequence length.

C.Results

REFERENCES

1. Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet (2013, Dec). Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. Available: <https://arxiv.org/abs/1312.6082>