Digit Recognizer

ABSTRACT: This project uses computer vision and machine learning techniques to recognizer to digit. A modest amount of training and test data (typeset digits) is available on the dataset. The files named “ones” “two” “threes” each contain many instances of the same digit, making them convenient for classifier training. If you would like more training and test data you have the following options. Use a scanner to obtain further images. Or to create more challenging images, use a camera to photograph scenes that contain digits such as house numbers or license plates. Locating digits in natural scenes images is a challenging image-processing problem; you can omit this step by instead manually delineating each region that contains a digit.

PROBLEM DESCRIPTION:

In this project you implement and test a classifier for recognizing digits 0..9 in a scanned document image. Complete part 1 of the project in the first half of the term: write code to find digit regions and measure features of these regions. Complete part 2 in the second half of term: implement and test a digit recognizer that classifies regions based on their measured features. Your final project report should describe your feature set and classifier design (what type of classifier did you choose, and why), your training and test data, the implementation of your classifier (what code did you write, or how did you configure classification code provided by a toolbox), and classifier performance (what is P(error) and how accurate is this estimate). Also include discussion and analysis: describe strong and weak points of your classification approach, avenues for improving the classifier in future work. Digit recognition has important practical applications such as reading zip codes for automated mail sorting. (USA zip codes contain only digits only, in contrast to the mix of letters and digits in Canadian postal codes.) Years of research and development have led to highper formance noise-tolerant digit classifiers that perform well on a large variety of fonts and handwritten digits. That level of performance is far beyond what you are expected to achieve in this course project.

NOTEBOOK USED - GOOGLE COLAB:

Collaboratory,or“Colab”forshort,isaproductfromGoogleResearch.Colaballowsanybodyto write and execute arbitrary python code through the browser and is especially well suited to machine learning, data analysis, and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

MODEL:

Lanet-5: The network has 5 layers with learnable parameters and hence named Lenet-5. It has three sets of convolution layers with a combination of average pooling. After the convolution and average pooling layers, we have two fully connected layers. At last, a Softmax classifier which classifies the images into respective class. The input to this model is a 32 X 32 grayscale image hence the number of channels is one. We then apply the first convolution operation with the filter size 5X5 and we have 6 such filters. As a result, we get a feature map of size 28X28X6. Here the number of channels is equal to the number of filters applied. After the first pooling operation, we apply the average pooling and the size of the feature map is reduced by half. Note that, the number of channels is intact. we have a convolution layer with sixteen filters of size 5X5. Again the feature map changed it is 10X10X16. The output size is calculated in a similar manner. After this, we again applied an average pooling or subsampling layer, which again reduce the size of the feature map by half i.e 5X5X16. Then we have a final convolution layer of size 5X5 with 120 filters. As shown in the above image. Leaving the feature map size 1X1X120. After which flatten result is 120 values.

After these convolution layers, we have a fully connected layer with eighty-four neurons. At last, we have an output layer with ten neurons since the data have ten classes.

Dataset Name: MNIST Dataset for Digit Recognizer

Dataset Location link: https://www.kaggle.com/c/digit-recognizer/data

Github link: https://github.com/ruchigambhava/project