Handwritten Digit Recognition

Ravinshu Makkar

Lovely Professional University, Punjab India

mravinshu@gmail.com

Abstract

OCR Optical character recognition or optical character reader is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image. Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

Early versions needed to be trained with images of each character and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs. Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

Introduction

Handwritten numeral recognition is in general a benchmark problem of Pattern Recognition and Artificial Intelligence. Compared to the problem of printed numeral recognition, the problem of handwritten numeral recognition is compounded due to variations in shapes and sizes of handwritten characters. Considering all these, the problem of handwritten numeral recognition is addressed under the present work in respect to handwritten digit recognition.

Handwritten digit

A normal digit is written in many different shapes and size by different people and can be understood by human brin as it has seen millions of variations rather than a machine which needs to combine firstly all the data of previous handwritten digits and then start to recognize a digit and then further the problem arise for many numbers written in same image. An image could be blur or unorganizable by us or the machines so there arise a new issue to verify it. It's hard to build a database with perfect output available. Overcoming all the problems the next is to build a perfect model.

An Artificial Neural Network (ANN) is a system that imitate the biological neural network in the brain of a human being. It's a machine learning algorithm, which means it uses data to learn how to respond to various inputs. The ANN can be thought of as a box that accepts one or more inputs and produces a single output. There are multiple connecting connections inside the box. The data is entered into the software, which then passes through the ANN's layers and nodes to produce an output using a transfer function.

For OCR, Artificial Neural Networks are employed, and they have shown to have a very high accuracy rate. The ANN would "recognize a character based on its topological properties, such as shape, symmetry, closed or open areas, and number of pixels" in this scenario. ("School of Science Engineering HANDWRITTEN DIGITS ...") This type of algorithm's excellent accuracy is due to its capacity to learn from a training set of characters with comparable characteristics.

Support Vector Machine:

SVM stands for Support Vector Machine and is a machine learning algorithm. Pattern classifiers with good performance are known as SVMs. SVMs strive to minimize the "upper bound of the generalization error" whereas Neural Networks aim to minimize the training error. This technique's learning method is based on classification and regression analysis. This type of classifier has shown to be particularly effective in the recognition of very complicated characters, such as those found in the Khmer language.

Feature Extraction

Feature extraction is a pattern recognition approach. The basic concept behind feature extraction is to analyses photos and generate some characteristics from them that may be used to identify each individual element. Curvatures, holes, and edges, for example, are examples of these properties. The holes inside the digits (for example, the eight, the six, and maybe the two) as well as the angles between some straight lines (for example, the one, the four, and the seven) are examples of these features in digit recognition. When an unknown image needs to be recognized, its characteristics are compared to these in order to classify it.

Image Correction

The technique of image correlation is used to recognize characters in images. In order to analyses the photos, this method, also known as Matrix Matching, employs mathematical algorithms. The images are read as matrices with this methodology, with each element representing a pixel, making it easier to modify them using mathematical methods. The image that needs to be identified is loaded as a matrix and compared to the reference set's images. The test image is superimposed on each image in the reference set to evaluate how it compares to each one and determine which one best depicts it. The judgement can be determined by looking at the pixels that match and the ones that are missing from either image.

More about OCR:

It is easy for the naked eye to recognize a character when spotted in any document; however, computers cannot identify the characters from an image or scanned document. In order to make this possible, a lot of research has been done, which resulted in the development of several algorithms that made this possible. One of the fields that specialize in character recognition under the light of Image Processing is Optical Character Recognition (OCR). In Optical Character Recognition, a scanned document or an image is read and segmented in order to be able to decipher the characters it contains. The images are taken and are preprocessed to get rid of the noise and have unified colors and shades, then the characters are segmented and recognized one by one, to finally end up with a file containing encoded text containing these characters, which can be easily read by computers. ("Handwritten Digit Recognition for Banking System - IJERT") Optical Character Recognition dates to the early 1900s, as it was developed in the United States in some reading aids for the blind. In 1914, Emanuel Goldberg was able to implement a machine able to convert characters into "standard telegraph code". ("School of Science Engineering HANDWRITTEN DIGITS ...") In the 1950s, David Shepard, who was at that time an engineer at the Department of Defense, developed a machine that he named Gismo, which can read characters and translate them into machine language. In 1974, Ray Kurzweil decided to develop a machine that would read text for blind and visually impaired people under his company, Kurzweil 4 Computer Products. There are several software and programs, nowadays, which use OCR in several different applications. In 1996, the United States Postal Services were able to develop a mechanism, HWAI, which recognizes handwritten mail addresses. ("School of Science Engineering HANDWRITTEN DIGITS ...")

1) Octave:

Octave is a free and open-source software that uses a high-level programming language. It is compatible with MATLAB and has the same functionality. It provides a very easy and appropriate interface for performing some mathematical calculations. It includes certain tools for solving math issues, such as some popular linear algebra problems.

When it comes to these operations, it is also quite efficient in terms of resource usage, such as time and memory. It is also quite simple to use when dealing with matrices, as it includes many functions and operations that make manipulating them less expensive. We will be dealing with photos as matrices in this project, with each element representing a pixel, so selecting a tool that will make our computations easier and more efficient in terms of time and memory resources is critical. MATLAB and Octave are both simple to learn and use, and they provide an ideal environment for this type of research.

2) MNIST Database

The MNIST database (Modified National Institute of Standards and Technology database) is a massive dataset with hundreds of thousands of handwritten numbers. Because the NIST set was divided into those written by high school students and those written by Census Bureau workers, this dataset was created by combining different sets within the original National Institute of Standards and Technology (NIST) sets to create a training set containing a variety of types and shapes of handwritten digits.

It is required to have a good dataset with a big number of handwritten digits in order to develop our recognizer and test its performance. This dataset should enable us to identify the obstacles and limitations of the image correlation technique, as well as encourage us to seek out ways and rules to improve it and measure its correctness. We chose this dataset to test our program since it has proven to be extremely reliable and important in the field.

3) Feasibility Study:

From a technical standpoint, because this project involves a lot of numerical computations, utilizing Octave is a good idea because it will make the application run faster. This software will also offer us with libraries for reading and manipulating photos, making the implementation process much easier.

The MNIST Database was chosen as the dataset to use in the project's testing. Thousands of handwritten numbers have been used in the creation of programs with a similar goal in this database. This dataset is free to use by the general public. It is also incredibly useful for our project, since it will save us time by allowing us to use it as a test set without having to make one ourselves.

Data Set Format:

The MNIST dataset I downloaded contains 60,000 photographs of handwritten numbers ranging from zero to nine, all clustered together in one file. Each graphic is 28 by 28 pixels in size and represents a digit. I've noticed that the photographs in the file aren't grouped in any pattern or sequence. The images are represented as matrices, with the pixels represented by the elements. Each image also has a label that identifies the digit it represents. This label came in handy later when it came time to create the test set. Furthermore, the data was used without preprocessing because there was no noise or serious issues to deal with.

Reference Set:

To be able to recognize the digit represented by a given image, it must be compared to other images containing recognized digits. To do so, you'll need to construct a reference set that includes all these photographs.

That is to say, each image we would want to recognize is to be compared to the images in the reference set. The image that best portrays the correct number is the one with the highest match. Because handwritten digits vary from person to person, the reference collection must include digits in a variety of fonts. As a result, we used the online image editor pixlr.com to make six photos of each digit, each with a different font. Images in the reference set are the same size as those in the MNIST dataset, i.e., 28 by 28 pixels.

Test Set:

To analyses its performance and quantify its success rate, the program to be developed must be tested against some images containing handwritten digits. As a result, it's critical to construct a test set. The test set is an example of an image comprising handwritten digits that must be compared to the photos in the reference set in order to be identified.

The file from the MNIST database was used to create this set. There were 60,000 photos in the original file, each representing a distinct digit. This made it impossible to find each number using the label during the program's testing. We opted to keep several photos from each digit in a different file to make it easier to access each digit we desire. As a result, we have 20 photographs of each digit stored in 10 distinct files. That is, the final test set consisted of ten files, each of which represented a number and contained 20 photographs of that digit. Octave was used to extract these photos from the original file by reading them and their labels.

We had to make some changes to the elements of all the matrices representing the test set in order to make manipulating the matrices/images easier. The black pixels were left alone because they were initially represented as zeros. The white ones, on the other hand, each had a separate non-zero number, thus we all became ones.