OCR Document

Ahmed Sherif, Omar Anas, Omar Khaled

May 2019

Contents

1 Introduction

- Our Project is **OCR** or **Optical Character Recognition**, specifically using English Alphabets and Numbers. The Main idea of **OCR** is to convert handwritten or printed text into machine-encoded text, whether it's coming from a scanned document, or a printed photo. **OCR** can be used in a wide variety of Activities, like: Voucher Code Scanning, QR code reading and Scanning Foreign Languages.

2 Project Overview

- Our Project manages to implement and execute **OCR** in its very own unique way.
- The Program first takes the external data-set as an Image, and converts it into a **GRAY-SCALE** image.
- The program afterwards takes the another external Image that we want to Read or Recognize.
- The Program then starts dividing (Segmentation) the image horizontally and Vertically by rows and columns.
- The program starts Inserting the divided image into two arrays, and starts the process of Training and Testing.
- After Training and Testing has been successfully completed, we assign a number to each letter and number of the Data-Set.
- At the end of the Program we use the (**KNN**) Classifier to try and find the nearest neighbors to each letter and number.
- Finally the program prints the values of the inserted image into the console, after using the Dictionary to translate each number to it's corresponding Letter or Number.

3 Libraries

- The Following Libraries have been used and imported inside our project, in order for us to be able to use some predefined functions and variables.

3.1 Numpy

- NUMPY is a fundamental package in python, that we used because it provides multidimensional array objects, along with the ability to perform fast operations on arrays, including mathematical, logical, shape manipulation and sorting.

3.2 Cv2

- OPENCV (Cv2) is also a fundamental library in Python designed to solve computer vision problems. we used this library in order to be able to read external Images and convert them into Gray-Scale Images.

3.3 Sklearn

- sklearn.metrics (Confusion Matrix) is mainly used in python to evaluate the accuracy of a classification. We used this particular library to be able to generate a matrix that shows how many times a Number or Letter has been successfully identified and recognized, and how many times it was not identified correctly.

4 Functions

- The Following functions has been used in implementing our **OCR** program :

1- **imread**() - Read Image from External Source.
2- **cvtColor**() - Convert Image color (Gray-Scale - Color).
3- **hsplit**() - Horizontally Split the Image.
4- **vsplit**() - Vertically Split the Image.
5- **reshape**() - Give a new shape to the array without changing its data.
6- **arange**() - Assign Values to other objects.
7- **repeat**() - Repeat elements of an array.
8- **KNearest-create**() - Finding the nearest neighbour.
9- **confusion-matrix**() - Create or Construct a matrix, after receiving both the Actual and the Predicted values as parameters after classification happens.

5 Classifier

- The Classifier we used in our project is the **KNN** classifier. The **k-nearest neighbors** algorithm is mainly used for classification, so it stores all available cases and classifies new cases based on a similarity between them.

6 Project Sequence

6.1 Acquire Data-set

- Data-set required is acquired in an image form, and them converted to a Gray-Scale Image.

6.2 Acquire To-Be-Detected Image

- The Image to be converted is read, and also converted to a Gray-Scale Image.

6.3 Segmentation

- Both Images are divided Horizontally and Vertically, to be able to extract the unique feature of each Letter and Number in the Data-set and the External Image.

6.4 Dividing To Arrays

- The Segmented parts are then added into two arrays.

6.5 Training and Testing

- The Arrays are used for Training and Testing, normally training should be more than testing, for example a 80:20 ratio.

6.6 Assigning Values

- Values from 0 to 35 are assigned to the Numbers (0-9) and Letters (A-z).

6.7 Classification

- The KNN Classifier works to try and find the closest match and the nearest neighbor to each number and letter.

6.8 Printing Output

- The Outut of the External Image is then printed in the console.

6.9 Confusion Matrix

- At the End a Confusion-Matrix has been constructed, so we could observe the success and failure of our project, by counting how many times a Number or Letter has been successfully recognized after classification.

6.10 Dictionary

- To be able to get the correct values of each number and letter, after assigning values from 0 to 35 to them, a small dictionary has been created, that matches each value to it's corresponding Number or Letter.

7 References

1- Website - Applications of OCR You Haven't Thought Of, https://medium.com/swlh/applications-of-ocr-you-havent-thought-of-69a6a559874b.
2- Website - Optical character recognition, https://en.wikipedia.org/wiki/Optical-character-recognition .
3- Website - Open Source Computer Vision, https://docs.opencv.org/trunk/d0/de3/tutorial-py-intro.html .
4- Book - opencv python tutorials
5- Website - https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion-matrix.html