TEXT DETECTION IN IMAGES

SHAHANA.V.M

DATE:

EMAIL: ayshashahana006@gmail.com

CONTENT

* Abstract

* Objective

* Technology Used

* Libraries And Packages Used

* Algorithm

* Data Collection

* Result

* Conclusion

ABSTRACT

Extracting text from images is a very popular task in the operations units of the business (extracting information from invoices and receipts) as well as in other areas. Text recognition in images is an active research area which attempts to develop a computer application with the ability to automatically read the text from images. Nowadays there is a huge demand of storing the information available on paper documents in to a computer readable form for later use. In this paper, we have reviewed and analysed a method for text recognition from images. Detecting text from an image is an important prerequisite for the content based image analysis process. To understand the contents of an image or the valuable information, there is need of analysing the text appears in it. Various methods have been proposed over past years for text detection and extraction from different types of images, like scene image, born digital image and document image. In this paper, we describe the existing methods of text detection. OCR (Optical Character Recognition) is an electronic computer-based approach to convert images of text i machine-encoded text, which can then be extracted and used in text format.

OBJECTIVE

The objective of this project is to create a machine learning model to detect text from images. Extracting text from images has found numerous applications. Some of the applications are Passport recognition, automatic number plate recognition, converting handwritten texts to digital text, converting typed text to digital text, etc.

TECHNOLOGY USED

PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects. Python is open-source, so it is free to use, modify and distribute the python source code. Python is a high-level programming language that has English-like syntax making it easier to read and understand the code also the standard library of python is very big, that any and all function needed for a project can be found minimizing the use of external libraries. Different python packages like numpy, pandas, matplotlib, opencv will used in the project.

MACHINE LEARNING

Machine learning is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to identify without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications where it is difficult to develop conventional algorithms to perform the needed tasks. The model which detects text from images will be built using machine learning algorithm

LIBRARIES AND PACKAGES USED

NUMPY

numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high -level mathematical functions to operate on these arrays. It is an open-source software and has many contributors.

MATPLOTLIB

matplotlib is an amazing visualization library in Python for 2D plots of arrays. It is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader scipy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. It consists of several plots like line, bar, scatter, histogram etc.

OPENCV

opencv(Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. It helps to process an image and apply various functions like resizing image, pixel manipulations, object detection, etc.

TESSERACT

tesseract is an open source optical character recognition (OCR) platform for python that also serves as a wrapper for the Tesseract-OCR Engine. It can read and recognize text in images and is commonly used in python ocr image to text use cases.

ALGORITHM

I used **Optical Character Recognition (OCR)** to create the text detection model. It is the process that converts an image of text into a machine-readable text format. A scanner reads documents and converts them to binary data. The OCR software analyses the scanned image and classifies the light areas as background and the dark areas as text.

The OCR software first cleans the image and removes errors to prepare it for reading. These are some of its cleaning techniques:

- Tilting the scanned document slightly to fix alignment issues during the scan.
- Despeckling or removing any digital image spots or smoothing the edges of text images.
- Cleaning up boxes and lines in the image.
- Script recognition for multi-language OCR technology.

OpenCV's text detector implementation of EAST is quite robust, capable of localizing text even when it's blurred, reflective, or partially obscured. We call the algorithm "EAST" because it's an: Efficient and Accurate Scene Text detection pipeline.

EAST is a very robust method for text detection. It is worth mentioning as it is only a text detection method. It can find horizontal and rotated bounding boxes. It can be used in combination with any text recognition method. The text detection pipeline has excluded redundant and intermediate steps and only has two stages. One utilizes the fully convolutional network to directly produce word or text-line level prediction. The produced predictions which could be rotated rectangles or quadrangles are further processed through the non-maximum-suppression step to yield the final output.

OCR process consists of several steps, starting with importing images and ending with exporting the detected results.
Here, I am working with essential packages. OpenCV package uses the EAST model for text detection. The tesseract package is for recognizing text in the bounding box detected for the text.

1.Image Processing

After loading the packages, we have to give the location of the image to be read. Then save the original image and shape. Calculate the ratio between original and new image for both height and weight. This ratio will be used to translate bounding box location on the original image. Resize the original image to new dimensions. Construct a blob from the image to forward pass it to EAST model.

2.Loading pre-trained EAST model and defining output layers

To detect text, we have to load the pre-trained EAST model for text detection. I would like to get two outputs from the EAST model. They are

a) Probability scores for the region whether that contains text or not.
b) Geometry of the text - - Coordinates of the bounding box detecting a text.

3.Forward pass the image through EAST model

To get the desired output layers, we have to forward pass the blob from the image. For that it sets the new value for the layer output blob. Then it will give numpy ndarray as output which you can use it to plot box on the given input image.

4.Function to decode bounding box from EAST model prediction

It returns a bounding box and probability score if it is more than minimum confidence. Then it loop over rows and number of columns. Extracting the rotation angle for the prediction and computing the sine and cosine. Using the geometric volume to get the dimensions of the bounding box. Then it compute start and end for the text prediction box and return bounding boxes and associated confidence value.

5.Getting final bounding boxes after non max suppression

Non Maximum Suppression is a class of algorithms to select one entity (e.g., bounding boxes) out of many overlapping entities. We can choose the selection criteria to arrive at the desired results. The criteria are most commonly some form of probability number and some form of overlap measure.

6.Generating list with bounding box coordinates and recognized text in the boxes

Initialize the list of results. Loop over the bounding boxes to find the coordinate of bounding boxes. Scale the coordinates based on the respective ratios in order to reflect bounding box on the original image. Extract the region of interest. Hence configuration setting to convert image to string. This will recognize the text from the image of bounding box. It append box coordinate and associated text to the list of results.

7.Display image with bounding box and recognized text

Moving over the results and display on the image. Display the text detected by Tesseract. Then it will display the text and shows the original image.

DATA COLLECTION

Data is mainly collected from internet. Relevant data can be obtained from websites like kaggle. Data is also collected from Google trends and Goggle forms.

RESULT

CONCLUSION