International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 11 • Issue 1 pp. 33-37 Sept 2019 - March 2020 www.csjournals.com

Design and Implementation of Optical Character Recognition System

Abhilash Singh[1], Dr. Sarvottam Dixit[2]
Student[1], Professor[2]
Department of Computer Science and Engineering, Mewar University, Chittorgarh, Rajasthan, India

Abstract: Suppose you wanted to digitize a magazine article or a printed contract. You could spend hours retyping after which correcting misprints. Or you could convert all of the required materials into digital format in numerous minutes by the usage of a scanner (or a digicam) and Optical Character Recognition software. Optical Character Recognition (OCR) is a technology that permits you to convert different kinds of documents, which include scanned paper documents, PDF files or pictures captured through a digital camera into editable and searchable data.

Imagine you were given a paper report - for example, magazine article, brochure, or PDF settlement your partner dispatched to you by means of email. Obviously, a scanner is not enough to make this information available to be had for editing, say in Microsoft Word. All a scanner can do is create an photograph or a snapshot of the file that is nothing more than a group of black and white or color dots, called a raster photo. In order to extract and repurpose statistics from scanned documents, camera pictures or picture-best PDFs, you need an OCR software that would unmarry out letters on the image, placed them into words after which - words into sentences, hence permitting you to get admission to and edit the content of the original record.

This venture goals at the creation of a comprehensive application, which may be used in company environments. The OCR software is as easy as feasible so that it can be configured even through a non-technical person. The major objective of this assignment is to scan the pictures uploaded to research the textual content in it. With this, consumer can effortlessly retrieve the text content material and use it as needed.

Keywords: Machine Learning, Optical Character Recognition, Segmentation, Normalisation, Classification and Algorithms.

Introduction

Machine Learning is the field of study that offers computers the aptitude to find out while not being expressly programmed. It is clear from the name that it offers the computer the ability which makes it a lot like humans: the power to find out. Machine learning is actively being employed these days, maybe in more places than one would expect.

The process of learning begins with observations or knowledge, like examples, direct expertise, or instruction, so as to seem for patterns in knowledge and build higher selections within the future based on the examples that we provide. The first aim is to permit the computers to learn mechanically without any human intervention or help and regulate actions consequently.

Basic Information

The process to categorize photographs of handwritten characters by the letters represented is termed as Optical Character Recognition (OCR).

OCR translates photographs of handwritten or typewritten text (commonly captured via a scanner) into system-editable text, or photographs of characters into a widespread encoding scheme representing them.

Early techniques exploited the regularity of the spatial patterns. Techniques like template matching which used the form of single-font characters to discover them in textual snap shots. More recent techniques that are used to recognise handwritten text do no longer rely solely at the spatial patterns but rather they characterise the shape of characters primarily based on the strokes used to generate them. These newer techniques at the moment are also implemented to the recognition of device printed characters, enhancing recognition prices and increasing the robustness of the techniques against image defects together with textual content alignment, imaging noise, and comparison deficiencies.

The major OCR programs are able to take benefit of the fact that the textual content is supplied on a uniform history that has sufficiently excessive assessment between text and history. When the heritage isn't always uniform OCR reputation quotes are significantly lower, and at present, aren't commercially viable. Although modern OCR algorithms exhibit full-size variety inside the techniques employed, the principal stages inside the OCR system can nevertheless be elaborated.

No one algorithm uses all the strategies, nor all of the ranges, and different algorithms use the ranges in exclusive orders. However, all algorithms do need to confront the issues raised in every of the ranges.

Proposed System

A. Extraction of Character Regions from an a picture

The extraction of character regions from a picture is predicated on exploitation ofancillary information known about the image to pick out an image property (or properties) that's (are) sufficiently totally different for the text regions and the background regions to be a basis of separating one from the other one.

B. Segmentation of the Image into Text and Background

While some OCR algorithms work with grey-scale pictures, several convert their input into binary pictures throughout the first stages of process. Though this is the case most try and extract non-textual regions, like graphics before they phase text from the background. Provided we've got image regions that contain text, whether or not single word regions or whole slabs of text, the goal of this stage is to spot image pixels that belong to text and those that belong to the background. The foremost common technique used here is thresholding of the grey-level image. The brink worth is also chosen exploitation auxiliary data regarding the image using applied mathematics techniques to pick out a world threshold that divides the image into 2 categories, or by exploitation measures calculated within the neighborhood of every pixel to see the 'best' native threshold. In practice, native adaptive thresholds appear to supply the best segmentation results.

C. Conditioning of the Image

Whatever techniques are used to extract text from its background, it's inevitable that the resultant image segments can contain some pixels known as happiness to the incorrect cluster. Conditioning the image refers to the techniques used to 'clean up and processing' the image, to delete noise. Morphological operators and neighborhood operations are the foremost standard for distinctive noise and deleting it. Usually, isolated pixels are simply taken away whereas regions of noise adjacent to text or background are tougher to spot and therefore remove.

D. Segmentation of Characters

Historically, OCR algorithms segmented their input image into regions that contained individual characters. While this is often possible for machine-printed text, it is more difficult for handwritten material and is perhaps the major task encountered in reading cursive scripts. Many newer algorithms, even for machine-printed text, avoid this stage and move into character recognition without prior character segmentation. In fact character recognition occurs before character segmentation. Morphology operations, Connected Component Analysis, and vertical projections are used to segment characters. Most of the techniques used to extract character regions from an image can be used to segment characters. However, most of the algorithms that employ this stage assume that some joined characters will be segmented as one character and some characters will be segmented into more than one piece. Later stages of processing may need to attempt to split a region or join one or more to form a single character. Character segmentation may be alternated with character recognition, recognition of easy characters being used to segment the remaining space into character regions.

E. Normalization of Character Size

After the image is segmented into characters, it is usual to adjust the size of the character region so that the following stages can assume a 'standard' character size. Of course, characters that have the same height vary in width, so the aspect ratio of the character region is important in the normalisation of character size. It is important to note that size normalisation is usually only required when the techniques in the following stage depend on character size. Some character features such as topological ones are independent of size and consequently size normalisation is not a pre-requisite in such features.

F. Feature Detection

The stages preceding Feature Detection are often described as preliminary processing. Feature detection and classification are the heart of OCR. Over the history of OCR many different feature detection techniques have been used. Initially template matching was used to find the whole character as a feature, while later, sub features of the character were sought. Algorithm found the boundary outlines, the character skeleton or medial axis, the Fourier or Wavelet coefficients of the spatial pattern, various moments both spatial and grey-level, and topological properties such as the number of holes in a pattern. All have been used as features for classification of the character regions.

G. Classification

The role of classification is to assign a personality region to the character whose properties best match the properties keep within the feature vector of the region. Initially, OCR classifiers cared-for be structural classifiers, that is, the designer devises a collection of tests supported whether or not explicit options exist in specific positions at intervals the character region. For instance, one may take a look at for a pointy corner within the lower left of a personality region as the way of characteristic between a 'B' and an '8'. The tests used were based on the designer's understanding of character formation and were a product of his coaching. The structural approach has a lot of success with machine written text than it will with written text wherever careful abstraction options are less characteristic of the text than in machine printed kind.

H. Verification

The final stage in OCR is verification. In this stage, knowledge about the expected result is used to check if the recognized text is consistent with what is expected. Such verification may include confirming that recognized words are indeed words found in a dictionary, or that vehicle licence plates are listed in the database of issued plates.

Verification cannot guarantee that the recognised text is correct but often it can detect errors. Errors can be handled by redoing the OCR but this time getting the second best classification, and repeating the verification. Alternatively, the classifier might provide an ordered list of possible classifications when the certainty of correct classification is below some threshold. The verifier can use its information to determine the final classification. Verification, while an important stage, is very much application dependent and often implemented independently of the OCR engine.

Methodology

This module work as Optical Character Recognizer. To analyse the results of the research, we can look at a number of things: a general comparison of the algorithms used, analysis of the results of each specific algorithm across the different datasets, as well as a closer look at some of the most frequent errors and confusion points.

Acknowledgement

I would like to express my special thanks of gratitude to our Guide Prof. Dr. Sarvottam Dixit.

Result and Analysis

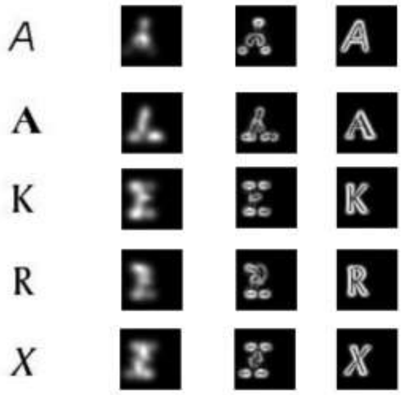
Datasets: The unique dataset consisted of one 40x40 pixel photo of each character (62 in total) in 20-factor length from over a thousand True Type fonts, all extracted the use of a simple PHP script. Due to huge memory necessities of this dataset, the dataset used for this research was decreased - we look handiest at capital letters and have notably reduced the range of fonts used to forty seven fonts for the first series of experiments and 238 fonts for the subsequent experiment. To extract the features from this dataset, a small .NET application become written, that accesses the raster of the photo and finds the needed data. A very modifiable photo manipulation application, ImageJ (additionally written in Java), changed into used to use various function extraction algorithms to the dataset, the usage of the ImageScience and FeatureJ plugins.

This pre-processing (the usage of ImageJ and FeatureJ) was applied to three distinctive units of pictures:

Dataset 1: Plain, unaltered images. Elements of the sparse array were black pixels.

Dataset 2: The identical set of photographs, which become modified by using calculating the structure tensor for each detail and then constructing a new image from the lowest eigenvalues of the eigenvectors of this transformation.

Dataset 3: The very last set of pics changed into constituted of the smallest eigenvalues of the Hessian of the authentic pictures. The Hessian transformation is used for discriminating domestically among plate-like, line-like, and blob-like photo structures.



Results: When the classification method was finished, the formula that gave the simplest results was the support vector classifier - SMO, with 70% properly classified instances because the worst result and a high of 88% correctly classified instances. The worst classifier, by far, was complement naïve Bayes, with 40%-70% properly classified. Naïve Bayes and also the J48 formula for coaching a cropped call C4.5 tree stayed within the middle with 60-70% properly classified instances for each dataset.

A Detailed check at the Performance of the Algorithms:

A. Complement Naïve Bayes

Complement naïve Thomas Bayes showed the weakest results overall, its kappa-statistic showing insignificant correlation between classified results and real teams altogether 3 datasets wherever it had been used. The common F-measure per category was never larger than 0.65, with values as low as 0.041 for a few categories. Precision and recall failed to show higher results, with averages from 0.5 to 0.7.

B. J48

J48 conjointly showed unsatisfactory leads to most datasets, with the exclusion of the Dataset one, large. Unlike the naïve Thomas Bayes algorithms, J48 failed to show any important grouping of classification errors, spreading them out instead. For the smaller datasets, the F-measure averages were between 0.4 and 0.65, with a coffee of 0.208, with similar results for exactitude and recall. Once applied to the larger dataset, the results modified considerably, with F-measure, recall and exactitude all larger than 0.785. Some common classification errors for this algorithmic rule are classifications of “P” as “F”, “F” as “P”, “V” as “Y”, “Y” as “V” and “C” as “G” and “G” as “C”, most of that are solely visible on the larger dataset.

C. Naïve Bayes

Naïve Thomas Bayes took the center ground between the weaker CNB and J48 and also the abundant stronger SMO. just like the 2 previous algorithms, it showed insignificant correlation once accustomed, to classify the smaller datasets, with the exception of Dataset one, wherever the correlation was statistically important ($\kappa = 0.7004$). For the smaller datasets, it achieved F-measures between 0.5 (Hessian) and 0.7 (plain raster), with recall and exactitude moving among similar bounds. For the larger dataset, it showed slightly weaker results than J48, with all 3 measures between 0.75 and 0.77.

D. SMO

Of all the classification algorithms tested, SMO gave the best results, significantly outclassing all the other algorithms, with the weakest percentage of correctly classified instances of 70% and a maximum of 88.96%. Due to the large number of correctly classified instances, there were no visible groupings of errors, with the exception of the “F” as “P” and “P” as “F” misclassifications. The lowest average F-measure was 0.704, while the highest was 0.89 (a full 0.114 better than the best achieved by any other algorithm). The highest average values for recall and precision were, similarly, just below 0.9.

Of all the classification algorithms tested, SMO gave the optimum results, considerably outclassing all the opposite algorithms, with the weakest proportion of properly classified instances of 70% and a most of

88.96%. Thanks to the massive variety of properly classified instances, there have been no visible groupings of errors, with the exception of the "F" as "P" and "P" as "F" misclassifications. Rock bottom average F-measure was 0.704, whereas the very best was 0.89 (a full 0.114 higher than the simplest achieved by the other algorithm). The very best average values for recall and exactitude were, similarly, just under 0.9.

References

1. Ian H. Witten and Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques, second edition", Morgan Kaufmann Publishers, Elsevier, San Francisco, CA, 2005.
2. Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, San Mateo, CA, 1993.
3. N. Sebe, Ira Cohen, Ashutosh Garg, Thomas S.Huang, "Machine Learning in Computer Vision", Springer, Netherlands, Dordrecht, 2005.
4. M. Sonka, V. Hlavac, R. Boyle, "Image Processing, Analysis, and Machine Vision, 2nd ed.", PWS Publishing, Pacific Grove, CA, 1999.
5. G. Smith, "Optical Character Recognition", CSIRO Manufacturing and Infrastructure Technology, Australia, 2003.
6. https://en.wikipedia.org/wiki/Optical_character_recognition
7. https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/