R.V. COLLEGE OF ENGINEERING,

BANGALORE-560059

(Autonomous Institution Affiliated to VTU, Belgaum)

REPORT ON

LICENSE PLATE RECOGNITION USING OCR

Submitted by

J M RAAGINI	**1RV15CS064**
GANESH BIRADAR	**1RV15CS054**
VINDHYA N	**1RV15CS183**

Under the guidance of

Prof. Rekha B S

Submitted to

INFORMATION SCIENCE AND ENGINEERING DEPARTMENT

R.V. College of Engineering, Bangalore-59

R.V. COLLEGE OF ENGINEERING, BANGALORE - 560059

(Autonomous Institution Affiliated to VTU, Belgaum)

TABLE OF CONTENTS

Topics **Page No.**

INTRODUCTION

Optical character recognition (also optical character reader, OCR) is the mechanical or electronic 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 (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast).

License Plate Recognition Systems use the concept of optical character recognition to read the characters on a vehicle license plate. In other words, LPR takes the image of a vehicle as the input and outputs the characters written on its license plate.This technology is gaining popularity in security and traffic installations. The technology concept assumes that all vehicles already have the identity displayed (the plate!) so no additional transmitter or responder is required to be installed on the car.

The system uses illumination (such as Infra-red) and a camera to take the image of the front or rear of the vehicle, then an image-processing software analyzes the images and extracts the plate information. This data is used for enforcement, data collection, and (as in the access-control system featured above) can be used to open a gate if the car is authorized or keep a time record on the entry or exit for automatic payment calculations.

The LPR system significant advantage is that the system can keep an image record of the vehicle which is useful in order to fight crime and fraud ("an image is worth a thousand words"). An additional camera can focus on the driver face and save the image for security reasons. Additionally, this technology does not need any installation per car (such as in all the other technologies that require a transmitter added on each car or carried by the driver).

METHODOLOGY

The basic stages involved in the implementation of OCR are explained below:

1. **Preprocessing Phase:**
   - Image being fed cannot be directly be fed to the model due to non-relevant components.
   - So, the image has to be preprocessed in order to improve the image data to suppress unwanted distortions or enhances some image feature important for further processing.
   - The output of this phase will be a cropped image containing the license plate.
2. **Training Phase:**
   - Printing out the characters which it is expected to recognize.
   - Scanning those characters into an image and crop it down so that it includes only the training characters.
   - Using the resulting training image to specify which characters the image contains.
3. **Character Recognition Phase:**
   - Load the scanned image of the document to be converted to text.
   - Preprocess the scanned image.
   - Break the input document into characters, based on several factors.
   - For each character, determine the most closely matching character from the training images and append that to the output text.

We have used License Plate Recognition(LPR) as the use case for the above implementation.

LPR also called ALPR (Automatic License Plate Recognition) has 3 major stages.

1. ***License Plate Detection:*** This is the first and probably the most important stage of the system. It is at this stage that the position of the license plate is determined. The input at this stage is an image of the vehicle and the output is the license plate.
2. ***Character Segmentation*:** It's at this stage the characters on the license plate are mapped out and segmented into individual images.

3. ***Character Recognition:*** This is where we wrap things up. The characters earlier segmented are identified here. We'll be using machine learning for this.

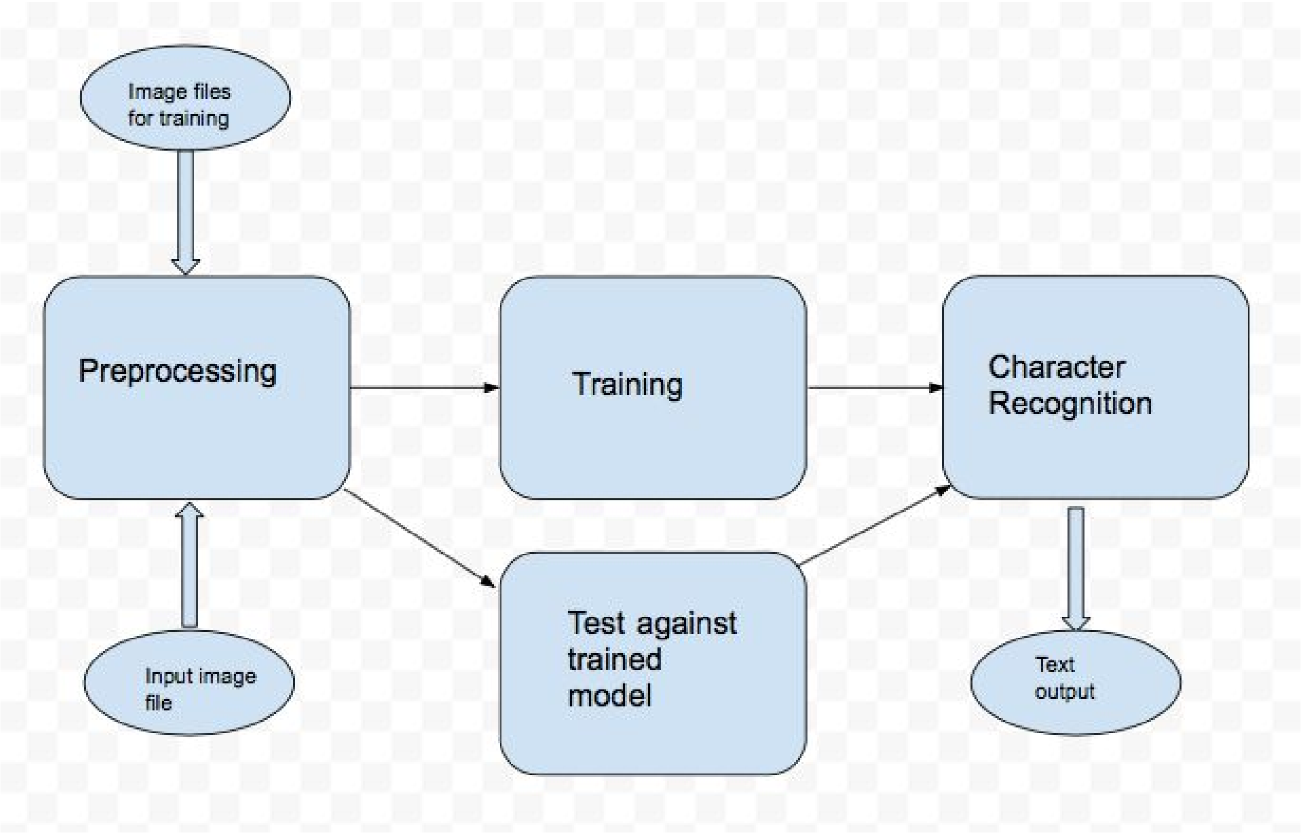

Fig 1: Structure chart for methodology

TECHNOLOGIES USED

Tesseract for OCR:

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0, and development has been sponsored by Google since 2006. Tesseract can process right-to-left text such as Arabic or Hebrew, many Indic scripts as well as CJK quite well.

Accuracy rates are shown in this presentation for Tesseract tutorial at DAS 2016, Santorini by Ray Smith. Tesseract is suitable for use as a backend and can be used for more complicated OCR tasks including layout analysis by using a frontend such as OCRopus.

Asprise for OCR:

Asprise OCR is a commercial optical character recognition and barcode recognition SDK library that provides an API to recognize text as well as barcodes from images and output in formats like plain text, xml and searchable PDF. Asprise OCR has been in active development since 1997.

License plate haar classifiers for object detection:

Haar-like features are digital image features used in object recognition. The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of integral images, a Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature).A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image.

Opencv for image preprocessing:

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.[1] Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel[2]). The library is cross-platform and free for use under the open-source BSD license.

JAVA CONCEPTS USED

- Data Abstraction : OO Abstraction occurs during class level design, with the objective of hiding the implementation complexity of how the the features offered by an API / design / system were implemented, in a sense simplifying the 'interface' to access the underlying implementation.
- Encapsulation : Encapsulation is a process of binding or wrapping the data and the codes that operates on the data into a single entity. This keeps the data safe from outside interface and misuse. One way to think about encapsulation is as a protective wrapper that prevents code and data from being arbitrarily accessed by other code defined outside the wrapper.
- Inheritance: In object-oriented programming, inheritance is the mechanism of basing an object or class upon another object (prototypical inheritance) or class (class-based inheritance), retaining similar implementation.
- Polymorphism : Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object. Any Java object that can pass more than one IS-A test is considered to be polymorphic.
- String Handling: Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects.The Java platform provides the String class to create and manipulate strings.

RESULTS AND ANALYSIS

Reasonable accuracy was achieved by using Haar classifier to extract the license plate and then using the OCR to extract the number from the plate. Also, the speed of the implemented algorithm was comparable to state of the art OCR's and can be used in real time.

CONCLUSION

Although we got a decent accuracy, the model can be still improved, mainly by giving it more data. It is a rule of thumb in Machine Learning that the more data the model has, the better it can learn and generalize. Apart from this, newer implementations can be tried which might give better results.

This is a generic implementation of OCR. The engine is designed to digitize a wide range of documents (or specific sections of documents). Normally, documents come in different formats (such as PDF, Word, Excel, PDF, images). They can also be in the form of paper-based or handwritten invoices and receipts which can be printed with a handful of known fonts. The engine can be used for training and classifying text, in order to minimize the error. Hence, this can be used to get greater accuracies than standard off-the-shelf OCR implementations.

REFERENCES

[1]. OpenCV C interface: http://docs.opencv.org

[2]. D.S. Chan, Theory and implementation of multidimensional discrete systems for signal processing, doctoral diss., Massachusetts Institute of Technology, Cambridge, MA, 1978.

[3]. G.S. Lehal and Chandan Singh, A Gurmukhi Script Recognition System, Proceedings of the International Conference on Pattern Recognition (ICPR'00), 1051-4651/00, 2000.

[4]. Anil K. Jain and Sushil Bhattacharjee, Text Segmentation Using Gabor Filters for Automatic Document Processing*, Pattern Recognition and Image Processing Laboratory. Michigan State University. E. Lansing,MI 48824-1027. USA.

[5]. AJ Palkovic, Improving Optical Character Recognition, Villanova University, United States.

[6]. Gama, J.; Brazdil, P. (2000). "Cascade Generalization". Machine Learning.

[7]. Lienhart, R. and Maydt, J., "An extended set of Haar-like features for rapid object detection

[8]. Dalal, Navneet (2005). "Histograms of oriented gradients for human detection"

[9]. Draghici, Sorin (1997). "A neural network based artificial vision system for license plate recognition"

[10]. "License Plate Recognition, Tribal Casinos, and Banned Persons"

[11]. "Automatic Number Plate Recognition". Cctv-information.co.uk.

[12]. "Arabic Licence Plate Recognition". Sourceforge.