

PORTABLE CAMERA BASED LABEL

READER FOR BLIND PEOPLE

Submitted in partial fulfilment of the requirements for the award of bachelor
of Engineering degree in

Electronics and Communication Engineering

BY



Chennavaram Irfan -38130040

T.Uma Venkatesh-38130220

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATIONENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
BONAFIDE CERTIFICATE

This is to certify that Project Report is the bonafide work of **Chennavaram Irfan and T.Uma Venkatesh** who carried out the Project entitled
“ PORTABLE CAMERA BASED LABEL READER FOR BLIND PEOPLE ”. Under the guidance of
V.VIJAYAKUMAR, Associate Professor, ECE during the period DEC
2021 to
MAY 2022

INTERNAL GUIDE

1.1 V.VIJAYA KUMAR, Associate Professor, ECE

HEAD OF THE DEPARTMENT

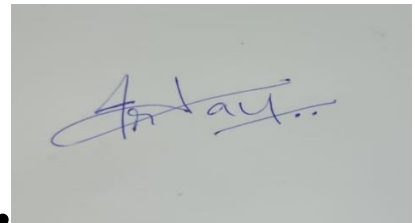
Dr. T. RAVI, M.E, PHD

Submitted for Viva Voice held on:-06/05/2022

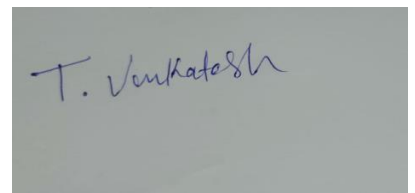
DECLARATION

We, **Chennavaram Irfan and T.Uma Venkatesh**, hereby declare that the Project Report entitled **Portable Camera Based Label Reader For Blind People** is done under the guidance of **Dr.V.VIJAYA KUMAR,Associate Professor,ECE**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

1.

A rectangular box containing a handwritten signature in blue ink, which appears to be 'Irfan'.

2.

A rectangular box containing a handwritten signature in blue ink, which appears to be 'T. Venkatesh'.

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my profound gratitude **to Dr. N.M. NANDHITHA, M.E., Ph.D., Prof. & Dean, School of Electrical and Electronics and Dr. T. RAVI, M.E., Ph.D., Head of the Department of Electronics and Communication Engineering, Sathyabama University** for having been a constant source of support and encouragement for the completion of the project.

I consider my self-most fortunate and immensely privileged to have worked under the guidance of **Dr.V.VIJAYA KUMAR,Associate Professor , Department of Electronics and Communication Engineering** to complete this project successfully.

I express our heartfelt thanks to our family members and all our friends who helped me to complete our project successfully & finally, I thank the exciting Super Natural Power, the Almighty GOD.

Raspberry Pi and Open Source OCR based Automatic Book Reader

ABSTRACT:

An OCR (Optical Character Recognition) system which is a branch of computer vision and in turn a sub-class of Artificial Intelligence. Optical character recognition is the translation of optically scanned bitmaps of printed or hand written text into audio output by using of Raspberry Pi. OCRs developed for many world languages are already under efficient use. This method extracts moving object region by a mixture-of-Gaussians-based background subtraction method. A text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object, a text localization and Tesseract algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Ada boost model. Text characters in the localized text regions are then binaries and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech. Performance of the proposed text localization algorithm. As the recognition process is completed, the character codes in the text file are processed using Raspberry pi device on which recognize character using Tesseract algorithm and python programming, the audio output is listened.

CHAPTER 1

1. INTRODUCTION

In the running world, there is growing demand for the software systems to recognize characters in computer system when information is scanned through paper documents as we know that we have number of newspapers and books which are in printed format related to different subjects. These days there is a huge demand in “storing the information available in these paper documents in to a computer storage disk and then later reusing this information by searching process”. One simple way to store information in these paper documents in to computer system is to first scan the documents and then store them as IMAGES. But to reuse this information it is very difficult to read the individual contents and searching the contents form these documents line-by-line and word-by-word. The reason for this difficulty is the font characteristics of the characters in paper documents are different to font of the characters in computer system. As a result, computer is unable to recognize the characters while reading them. This concept of storing the contents of paper documents in computer storage place and then reading and searching the content is called DOCUMENT PROCESSING. Sometimes in this document processing we need to process the information that is related to languages other than the English in the world. For this document processing we need a software system called **CHARACTER RECOGNITION SYSTEM**. This process is also called DOCUMENT IMAGE ANALYSIS (DIA).

Thus our need is to develop character recognition software system to perform Document Image Analysis which transforms documents in paper format to electronic format. For this process there are various techniques in the world. Among all those techniques we have chosen Optical Character Recognition as main fundamental technique to recognize characters. The conversion of paper documents in to electronic format is an on-going task in many of the organizations particularly in Research and Development (R&D) area, in large business enterprises, in government institutions, so on. From our problem statement we can introduce the necessity of Optical Character Recognition in mobile electronic devices such as cell phones, digital cameras to acquire images and recognize them as a part of face recognition and validation.

To effectively use Optical Character Recognition for character recognition in-order to perform Document Image Analysis (DIA), we are using the information in Grid format. . This system is thus effective and useful in *Virtual Digital Library’s* design and construction.

1.1 PURPOSE

The main purpose of **Optical Character Recognition (OCR)** system based on a grid infrastructure is to perform Document Image Analysis, document processing of electronic document formats converted from paper formats more effectively and efficiently. This improves the accuracy of recognizing the characters during document processing compared to various existing available character recognition methods. Here OCR technique derives the meaning of the characters, their font properties from their bit-mapped images.

- The primary objective is to speed up the process of character recognition in document processing. As a result the system can process huge number of documents with-in less time and hence saves the time.
- Since our character recognition is based on a grid infrastructure, it aims to recognize multiple heterogeneous characters that belong to different universal languages with different font properties and alignments.

1.2 PROJECT SCOPE

The scope of our product Optical Character Recognition on a grid infrastructure is to provide an efficient and enhanced software tool for the users to perform Document Image Analysis, document processing by reading and recognizing the characters in research, academic, governmental and business organizations that are having large pool of documented, scanned images. Irrespective of the size of documents and the type of characters in documents, the product is recognizing them, searching them and processing them faster according to the needs of the environment.

1.3 EXISTING SYSTEM

In the running world there is a growing demand for the users to convert the printed documents in to electronic documents for maintaining the security of their data. Hence the basic OCR system was invented to convert the data available on papers in to computer process able documents, So that the documents can be editable and reusable. The existing system/the previous system of OCR on a grid infrastructure is just OCR without grid functionality. That is the existing system deals with the homogeneous character recognition or character recognition of single languages.

1.4 DRAWBACK OF EXISTING SYSTEM

The drawback in the early OCR systems is that they only have the capability to convert and recognize only the documents of English or a specific language only. That is, the older OCR system is uni-lingual.

1.5 PROPOSED SYSTEM

Our proposed system is OCR on a grid infrastructure which is a character recognition system that supports recognition of the characters of multiple languages. This feature is what we call grid infrastructure which eliminates the problem of heterogeneous character recognition and supports multiple functionalities to be performed on the document. The multiple functionalities include editing and searching too where as the existing system supports only editing of the document. In this context, Grid infrastructure means the infrastructure that supports group of specific set of languages. Thus OCR on a grid infrastructure is multi-lingual.

1.6 BENEFIT OF PROPOSED SYSTEM

The benefit of proposed system that overcomes the drawback of the existing system is that it supports multiple functionalities such as editing and searching. It also adds benefit by providing heterogeneous characters recognition.

1.7 ARCHITECTURE OF THE PROPOSED SYSTEM

The Architecture of the optical character recognition system on a grid infrastructure consists of the three main components. They are:-

- Scanner
- OCR Hardware or Software
- Output Interface

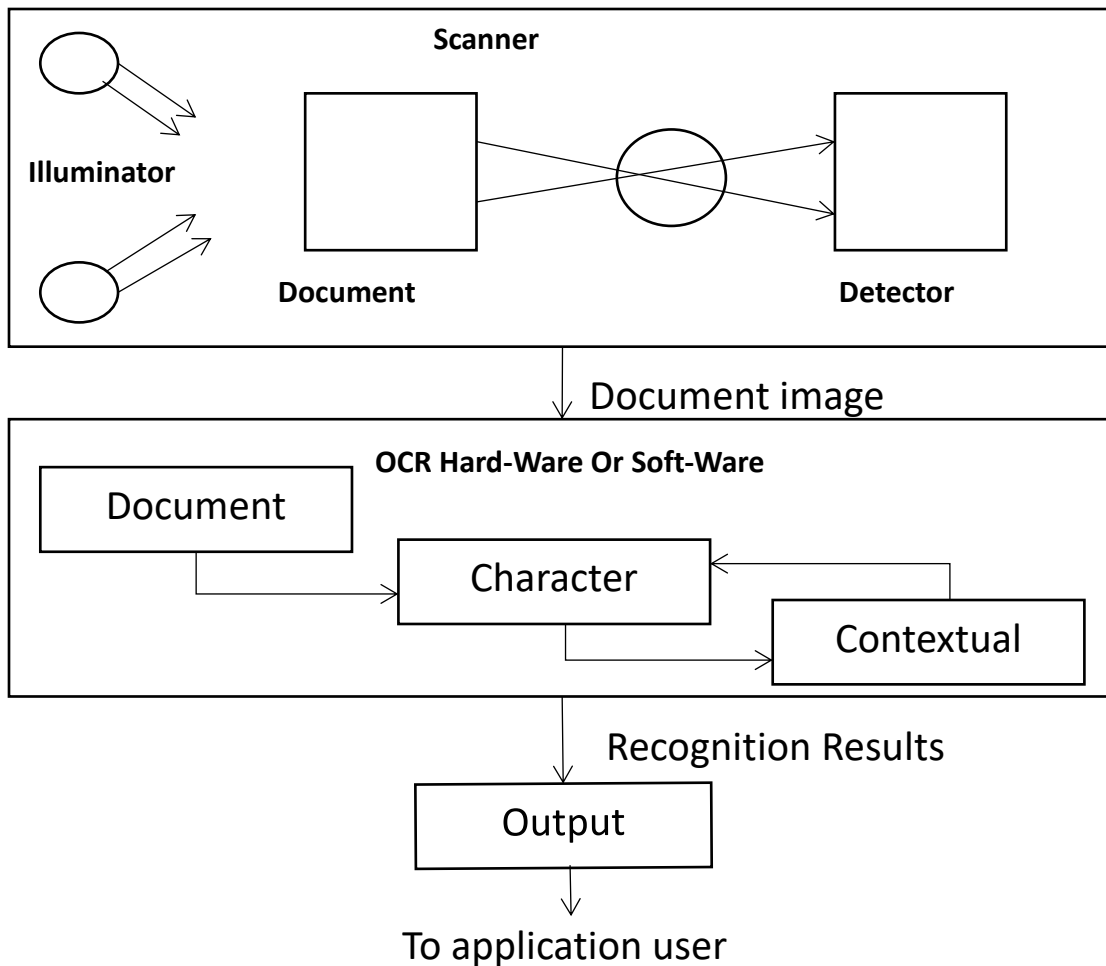


Figure.1: OCR Architecture

1.8 INTENDED AUDIENCE AND READING SUGGESTIONS

In this section, we identify the audience who are interested with the product and are involved in the implementation of the product either directly or indirectly. As from our research, the OCR system is mainly useful in R&D at various scientific organizations, in governmental institutes and in large business organizations, we identify the following as various interested audience in implementing OCR system:-

- The scientists, the research scholars and the research fellows in telecommunication institutions are interested in using OCR system for processing the word document that contains base paper for their research.

- The Librarian to manage the information contents of the older books in building virtual digital library requires use of OCR system.
- Various sites that vendor e-books have a huge requirement of this OCR system in-order to scan all the books in to electronic format and thus make money. The Amazon book world is largely using this concept to build their digital libraries.

Now we present the reading suggestions for the users or clients through which the user can better understand the various phases of the product. These suggestions may be effective and useful for the beginners of the product rather than the regular users such as research scholars, librarians and administrators of various web-sites. With these suggestions, the user need not waste his time in scrolling the documents up and down, browsing through the web, visiting libraries in search of different books and ... The following are the various reading suggestions that the user can follow in-order to completely understand about our product and to save time:-

- It would help you if you start with Wikipedia.com. It lets you know the basic concept of every keyword you require. First learn from it what is OCR? And how does it work based on a Grid infrastructure?
- Now you can proceed your further reading with the introduction of our product we provided in our documentation. From these two steps you completely get an in-depth idea of the use of our product and several processes involved in it.
- The more you need is the implementation of the product. For this you can visit FreeOCR.com where you can view how the sample OCR works and you can try it.

1.9 PROBLEM STATEMENT:

- Traditional methods like Braille exist using which the blind people have to trace and read text, which is very slow and not very practical.
- Existing OCR systems are not automatic and require full-fledged computers to run and hence are not effective.
- Reader Mobile runs on a cell phone and allows the user to read mail, receipts, fliers, and many other documents

2. LITERATURE REVIEW

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

2.1 TECHNICAL FEASIBILITY

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Find out whether the organization currently possesses the required technologies. Is the required technology available with the organization?.

2.2 OPERATIONAL FEASIBILITY

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

- Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
- Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
- Have the user been involved in the planning and development of the project?
- Early involvement reduces the chances of resistance to the system and in general and increases

the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

2.3 ECONOMIC FEASIBILITY

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

2.4 TRAINING

Training is a very important process of working with a neural network. As seen from neural networks, there are two forms of training that can be employed with a neural network. They are namely:-

1. Un-Supervised Training
2. Supervised Training

Supervised training provides the neural network with training sets and the anticipated output. Unsupervised training supplies the neural network with training sets, but there is no anticipated output provided.

2.4.1 UNSUPERVISED TRAINING

Unsupervised training is a very common training technique for Kohonen neural networks. We will discuss how to construct a Kohonen neural network and the general process for training without supervision.

What is meant by training without supervision is that the neural network is provided with training sets, which are collections of defined input values. But the unsupervised neural network is not provided with anticipated outputs.

Unsupervised training is usually used in a classification neural network. A classification neural network takes input patterns, which are presented to the input neurons. These input patterns are then processed, and one single neuron on the output layer fires. This firing neuron can be thought of as the classification of which group the neural input pattern belonged to. Handwriting recognition is a good application of a classification neural network.

The input patterns presented to the Kohonen neural network are the dot image of the character that was hand written. We may then have 26 output neurons, which correspond to the 26 letters of the English alphabet. The Kohonen neural network should classify the input pattern into one of the 26 input patterns.

During the training process the Kohonen neural network in handwritten recognition is presented with 26 input patterns. The network is configured to also have 26 output patterns. As the Kohonen neural network is trained the weights should be adjusted so that the input patterns are classified into the 26 output neurons. This technique results in a relatively effective method for character recognition.

Another common application for unsupervised training is data mining. In this case you have a large amount of data, but you do not often know exactly what you are looking for. You want the neural network to classify this data into several groups. You do not want to dictate, ahead of time, to the neural network which input pattern should be classified to which group. As the neural network trains the input patterns will fall into similar groups. This will allow you to see which input patterns were in common groups.

2.4.2 SUPERVISED TRAINING

The supervised training method is similar to the unsupervised training method in that training sets are provided. Just as with unsupervised training these training sets specify input signals to the neural network.

The primary difference between supervised and unsupervised training is that in supervised training the expected outputs are provided. This allows the supervised training algorithm to adjust the weight matrix based on the difference between the anticipated output of the neural network, and the actual output.

There are several popular training algorithms that make use of supervised training. One of the most common is the back-propagation algorithm. It is also possible to use an algorithm such as simulated annealing or a genetic algorithm to implement supervised training

2.5 INTRODUCING KOHONEN NEURAL NETWORK

The Kohonen neural network differs considerably from the feed-forward back propagation neural network. The Kohonen neural network differs both in how it is trained and how it recalls a pattern. The Kohonen neural network does not use any sort of activation function. Further, the Kohonen neural network does not use any sort of a bias weight.

Output from the Kohonen neural network does not consist of the output of several neurons. When a pattern is presented to a Kohonen network one of the output neurons is selected as a "winner". This "winning" neuron is the output from the Kohonen network. Often these "winning" neurons represent groups in the data that is presented to the Kohonen network. For example, in an OCR program that uses 26 output neurons, the 26 output neurons map the input patterns into the 26 letters of the Latin alphabet.

The most significant difference between the Kohonen neural network and the feed forward back propagation neural network is that the Kohonen network trained in an unsupervised mode. This means that the Kohonen network is presented with data, but the correct output that corresponds to

that data is not specified. Using the Kohonen network this data can be classified into groups. We will begin our review of the Kohonen network by examining the training process.

It is also important to understand the limitations of the Kohonen neural network. Neural networks with only two layers can only be applied to linearly separable problems. This is the case with the Kohonen neural network. Kohonen neural networks are used because they are a relatively simple network to construct that can be trained very rapidly.

A "feed forward" neural network is similar to the types of neural networks that we are ready examined. Just like many other neural network types the feed forward neural network begins with an input layer. This input layer must be connected to a hidden layer. This hidden layer can then be connected to another hidden layer or directly to the output layer. There can be any number of hidden layers so long as at least one hidden layer is provided. In common use most neural networks will have only one hidden layer. It is very rare for a neural network to have more than two hidden layers. We will now examine, in detail, the structure of a "feed forward neural network".

The Structure of a Feed Forward Neural Network

A "feed forward" neural network differs from the neural networks previously examined. **Figure 2.1 shows a typical feed forward neural network with a single hidden layer.**

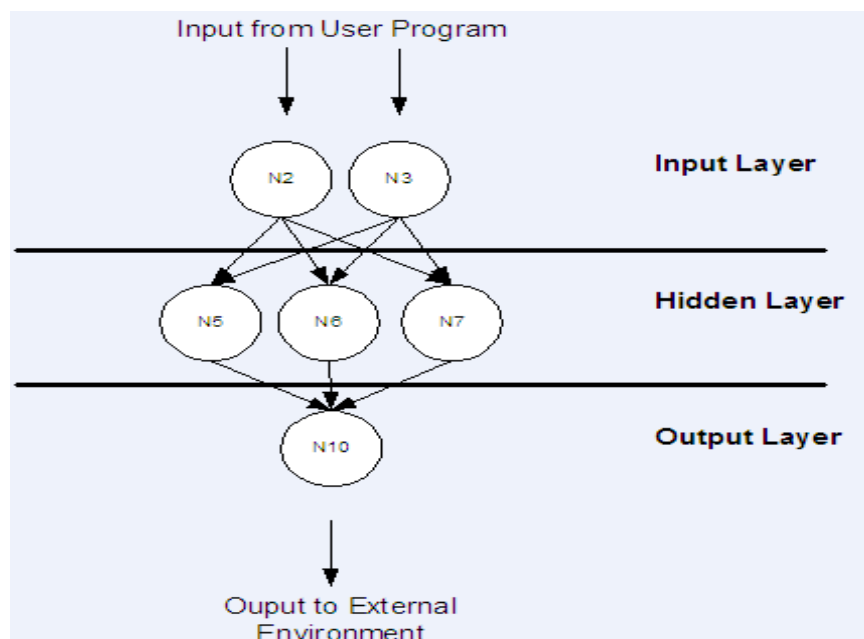


Figure 2 Feed Forward Neural Network

The Input Layer

The input layer to the neural network is the conduct through which the external environment presents a pattern to the neural network. Once a pattern is presented to the input layer of the neural network the output layer will produce another pattern. In essence this is all the neural network does. The input layer should represent the condition for which we are training the neural network for. Every input neuron should represent some independent variable that has an influence over the output of the neural network.

It is important to remember that the inputs to the neural network are floating point numbers. These values are expressed as the primitive Java data type "double". This is not to say that you can only process numeric data with the neural network. If you wish to process a form of data that is non-numeric you must develop a process that normalizes this data to a numeric representation.

The Output Layer

The output layer of the neural network is what actually presents a pattern to the external environment. Whatever pattern is presented by the output layer can be directly traced back to the input layer. The number of output neurons should be directly related to the type of work that the neural network is to perform.

To consider the number of neurons to use in your output layer you must consider the intended use of the neural network. If the neural network is to be used to classify items into groups, then it is often preferable to have one output neuron for each group that the item is to be assigned into. If the neural network is to perform noise reduction on a signal then it is likely that the number of input neurons will match the number of output neurons. In this sort of neural network you would one day want the patterns to leave the neural network in the same format as they entered.

For a specific example of how to choose the numbers of input and output neurons consider a program that is used for optical character recognition, or OCR. To determine the number of neurons used for the OCR example we will first consider the input layer. The number of input neurons that we will use is the number of pixels that might represent any given character.

Characters processed by this program are normalized to universal size that is represented by a 5x7 grid. A 5x7 grid contains a total of 35 pixels. The optical character recognition program therefore has 35 input neurons.

The number of output neurons used by the OCR program will vary depending on how many characters the program has been trained for. The default training file that is provided with the optical character recognition program is trained to recognize 26 characters. As a result using this file the neural network would have 26 output neurons. Presenting a pattern to the input neurons will fire the appropriate output neuron that corresponds to the letter that the input pattern corresponds to.

2.6. VISION BASED ASSISTIVE SYSTEM FOR LABEL DETECTION WITH VOICE OUTPUT

A camera based assistive text reading framework to help blind persons read text labels and product packaging from hand-held object in their daily resides is proposed. To isolate the object from cluttered backgrounds or other surroundings objects in the camera view, we propose an efficient and effective motion based method to define a region of interest (ROI) in the video by asking the user to shake the object. In the extracted ROI, text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object ROI, we propose a novel text localization algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Adaboost model. Text characters in the localized text regions are then binarized and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech

I. INTRODUCTION

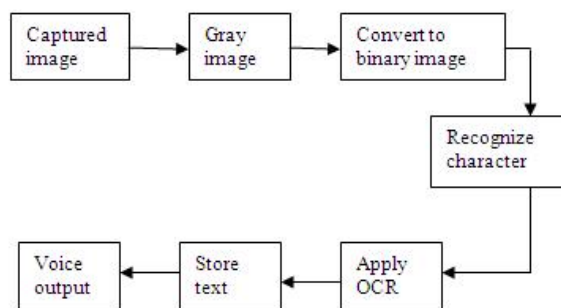
Of the 314 million visually impaired people worldwide, 45 million are blind. Recent developments in computer vision, digital cameras and portable computers make it feasible to assist these individuals by developing camera based products that combine computer vision technology with other existing commercial products such optical character recognition (OCR) systems.

Reading is obviously essential in today's society. Printed text is everywhere in the form of reports, receipts, bank statements, restaurant menus, classroom handouts, product packages, instructions on medicine bottles, etc.

The ability of people who are blind or have significant visual impairments to read printed labels and product packages will enhance independent living and foster economic and social self-sufficiency. Today, there are already a few systems that have some promise for portable use, but they cannot handle product labeling. Selectively extract the image of the object held by the blind user from the cluttered background or other neutral objects in the camera view; and 2) text localization to obtain image regions containing text, and text recognition to transform image-based text information into readable codes. We use a mini laptop as the processing device in our current prototype system. The audio output component is to inform the blind user of recognized text codes.

III. IMAGE CAPTURING AND PRE-PROCESSING

The video is captured by using web-cam and the frames from the video is segregated and undergone to the pre-processing. First, get the objects continuously from the camera and adapted to process. Once the object of interest is extracted from the camera image and it converted into gray image. Use haar cascade classifier for recognizing the character from the object. The work with a cascade classifier includes two major stages: training and detection. For training need a set of samples. There are two types of samples: positive and negative.



To extract the hand-held object of interest from other objects in the camera view, ask users to shake the hand-held objects containing the text they wish to identify and then employ a motion-based method to localize objects from cluttered background.

IV. AUTOMATIC TEXT EXTRACTION

In order to handle complex backgrounds, two novel feature maps to extract text features based on stroke orientations and edge distributions, respectively. Here, stroke is defined as a uniform region with bounded width and significant extent. These feature maps are combined to build an Adaboost based text classifier.

V. TEXT REGION LOCALIZATION

Text localization is then performed on the camera based image. The Cascade-Adaboost classifier confirms the existence of text information in an image patch but it cannot the whole images, so heuristic layout analysis is performed to extract candidate image patches prepared for text classification. Text information in the image usually appears in the form of horizontal text strings containing no less than three character members.

VI. TEXT RECOGNITION AND AUDIO OUTPUT

Text recognition is performed by off-the-shelf OCR prior to output of informative words from the localized text regions. A text region labels the minimum rectangular area for the accommodation of characters inside it, so the border of the text region contacts the edge boundary of the text characters. However, this experiment show that OCR generates better performance text regions are first assigned proper margin areas and binaries to segments text characters from background

The recognized text codes are recorded in script files. Then, employ the Microsoft Speech Software Development Kit to load these files and display the audio output of text information. Blind users can adjust speech rate, volume and tone according to their preferences. Are designed to easily interface with dedicated computer systems by using the same USB technology that is found on most computers. Static random-access memory (SRAM) is a type of a semiconductor memory that uses bi-stable latching circuitry to store each bit.

ARM11 features are, Supports 4-64k cache sizes, Powerful ARMV6 instruction set architecture, SIMD (Single Instruction Multiple Data) media processing extensions deliver up to 2x performance for video processing, and High-performance 64-bit memory system speeds data access for media processing and networking applications. LAN9512/LAN9512i contains an

integrated USB 2.0 hub, two integrated downstream USB 2.0 PHYs, an integrated upstream USB 2.0 PHY, a 10/100 Ethernet PHY, a 10/100 Ethernet Controller, a TAP Controller and EEPROM Controller. Flash memory is an electronic non-volatile compute storage medium that can be an electrically erased and reprogrammed. Earphones either have wires for connection to a signal source such as an audio amplifier, radio, CD player, portable media player or have a wireless receiver, which is used to pick up signal without using a cable

The proposed system ensures to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, a motion-based method to detect the object of interest is projected, while the blind user simply shakes the object for a couple of seconds. This method can effectively distinguish the object of interest from background or other objects in the camera view. An Adaboost learning model is employed to localize text in camera-based images. Off the shelf OCR is used to perform word recognition on the localized text regions and transform into audio output for blind users.

CHAPTER 3

PROPOSED SYSTEM

3.1 INTRODUCTION

Of the 314 million visually impaired people worldwide, 45 million are blind. Recent developments in computer vision, digital cameras and portable computers make it feasible to assist these individuals by developing camera based products that combine computer vision technology with other existing commercial products such as optical character recognition (OCR) systems. Reading is obviously essential in today's society. Printed text is everywhere in the form of reports, receipts, bank statements, restaurant menus, classroom handouts, product packages, instructions on medicine bottles, etc.

The ability of people who are blind or have significant visual impairments to read printed labels and product packages will enhance independent living and foster economic and social self-sufficiency. Today, there are already a few systems that have some promise for portable use, but they cannot handle product labeling. For example, portable bar code readers designed to help blind people identify different products in an extensive product database can enable users who are blind to access information about these products through speech and Braille.

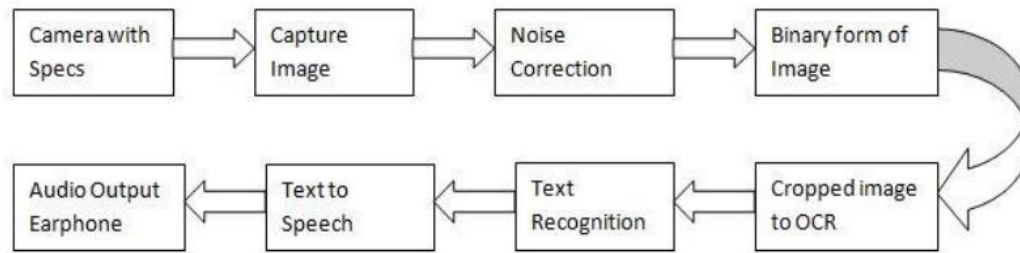
3.2 PROJECT DESCRIPTION

But a big limitation is that it is very hard for blind users to find the position of the bar code and to correctly point the bar code reader at the bar code. Some reading assistive systems such as pen scanner might be employed in these and similar situations. Although a number of reading assistants have been designed specifically for the visually impaired, to our knowledge, no existing reading assistant can read text from the kinds of challenging patterns and backgrounds found on many everyday commercial products.

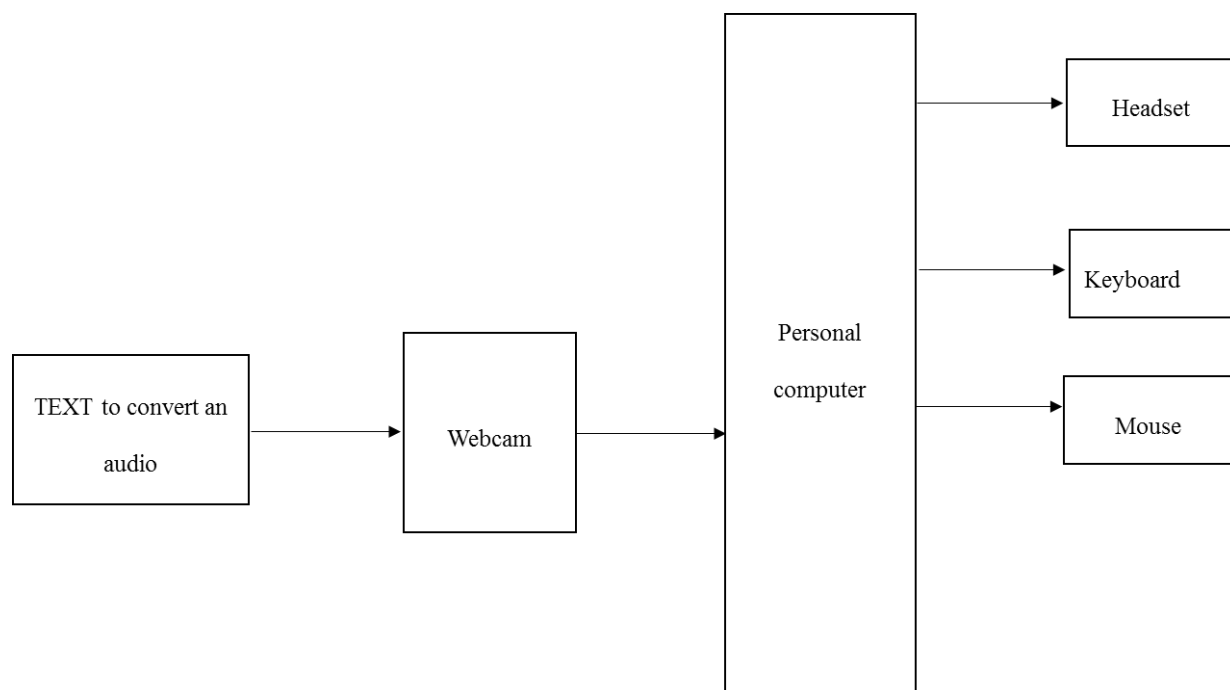
To assist blind persons to read text from these kinds of hand-held objects, we have conceived of a camera based assistive text reading framework to track the object of interest within the camera view and extract printed text information from the object. Our proposed algorithm can effectively

handle complex background and multiple patterns, and extract text information from both hand-held objects and nearby signage.

FLOW DIAGRAM:



3.3 BLOCK DIAGRAM:



We have described a prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have

proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. The automatic ROI detection and text localization algorithms were independently evaluated as unit tests to ensure effectiveness and robustness of the whole system. We subsequently evaluated this prototype system of assistive text reading using images of hand-held objects captured by ten blind users in person. Two calibrations were applied to prepare for the system test. First, we instructed blind users to place hand-held object within the camera view. Since it is difficult for blind users to aim their held objects, we employed a camera with a reasonably wide angle.

In future systems, we will add finger point detection and tracking to adaptively instruct blind users to aim the object. Second, in an applicable blind-assistive system, a text localization algorithm might prefer higher recall by sacrificing some precision

When our application starts running it first check all the devices and resources which it needs are available or not. After that it checks the connection with the devices and gives control to the user. The GUI for the user has the following options. An optional label is used for displaying the image taken from the camera. A status box is for representing the detected data from the image. The capture button is to detect the data from the image. The detect button is to detect the human from the video streaming in front of the camera. The audio jack port is the output port here. The Raspberry board comes with integrated peripherals like USB, ADC and Serial etc. On this board we are installing Linux operating system with necessary drivers for all peripheral devices.

3.4 WORKING PRINCIPLE

When capture button is clicked this system captures the product image placed in front of the web camera which is connected to ARM microcontroller through USB .After selecting the process button the captured label image undergoes Optical Character Recognition(OCR) Technology. OCR technology allows the conversion of scanned images of printed text or symbols (such as a page from a book) into text or information that can be understood or edited using a computer program.

The most familiar example is the ability to scan a paper document into a computer where it can then be edited in popular word processors such as Microsoft Word. However, there are many other

uses for OCR technology, including as a component of larger systems which require recognition capability, such as the number plate recognition systems, or as tools involved in creating resources for SALT development from print based texts. In our system for OCR technology we are using TESSERACT library. Using Flite library the data will be converted to audio.

Camera acts as main vision in detecting the label image of the product or board then image is processed internally and separates label from image by using open CV library and finally identifies the product and identified product name is pronounced through voice.

Now it identifies received label image is converted to text by using tesseract library. Once the identified label name is converted to text and converted text is displayed on display unit connected to controller. Now converted text should be converted to voice to hear label name as voice through ear phones connected to audio jack port using flite library.

CHAPTER 4

OCR

Optical character recognition is the translation of optically scanned bitmaps of printed or written text into digitally editable data files. OCRs developed for many world languages are already under efficient use.

4.1.1 Computer Vision

Human intelligence is described as the capability to make decisions based on information which is incomplete and noisy. This is the ability which makes human being the most superior amongst all living creatures in the world.

There are five senses that provide information to humans for making everyday decisions and out of these the senses of hearing and vision are the sharpest of all. The auditory sense helps us recognize sounds and classify them. It is this sense which tells us that the person on the phone is a friend because his voice is *recognizable*. We can differentiate between an endless variety of sounds, voices, utterances and put them in exactly the slots they belong to, animal sounds, musical notes, wind swishing, the footsteps of a family member, all are within recognition range of a person with a normal sense of hearing.

The other one and the one more profound is the human vision which allows us to identify a known person in a crowd of unknowns merely by casting a cursory glance at him. It allows us to pick an object that belongs to us from a number of those looking exactly as ours, and by being able to recognize a miss-spelt word in a sentence and unconsciously correct it. The fact is that the human mind is capable of identifying an image on features spontaneously determined and not predefined or predetermined.

With the development of technology these human processes are imitated to create intelligent machines, so much for the immense growth of robotics and intelligent decision making systems and yet the work done so far is not comparable to any natural involuntary human action or process. The hindrance however is that it is not practically possible to imitate all

the functions of the human mind and make computer vision as efficient and accurate as the human eye but even though such a possibility may be remote, efforts are consistently being made to bring them as close to it as possible.

Artificial Intelligence is a broad field of computer science taking into account a number of other disciplines that form the bulk of its study. One of the most popular definitions of Artificial Intelligence (AI) was given by Elaine Rich and Kevin Knight as, “Artificial Intelligence (AI) is the study of how to make computers do things which, at the moment, people do better” [20]. One important branch of Artificial intelligence is Computer vision, as mentioned in Figure 1.1 that shows few sub-branches of Computer Vision, the area of research which aims to imitate human vision and forms the basis of all image acquisition, its processing, document understanding and recognition. Computer vision relies on a solid understanding of the physical process of image formation to obtain simple inference from individual pixel values like shape of the object and to recognize objects using geometric information or probabilistic techniques

In its own turn document understanding is a vast and difficult area for the focus of research today lies in being able to make content based searches which hope to allow machines to look beyond the key words, headings or merely topics to find a piece of information. A far more streamlined field of Document Recognition and understanding is Optical Character Recognition which attempts to identify a single character from an optically read text image as a part of a word that can be then used to process further information on. The area gains rising significance as more and more information each day needs to be stored, processed and retrieved rather than being keyed in from an already present printed or handwritten source.

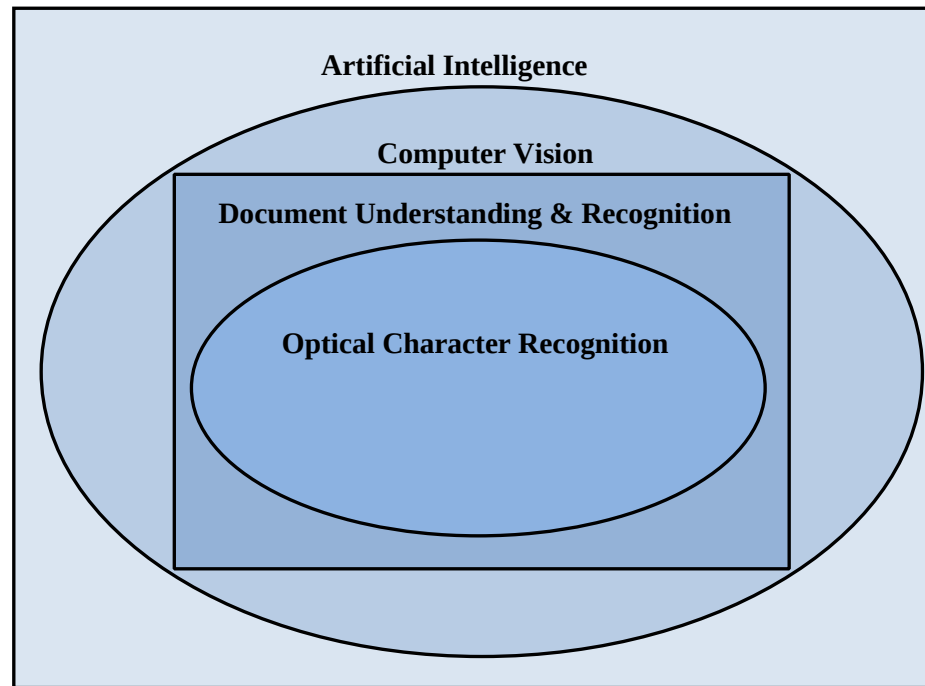


Figure 1.1: Sub-fields of Artificial Intelligence

4.1.2 Character Recognition

Character recognition is a sub-field of pattern recognition in which images of characters from a text image are recognized and as a result of recognition respective character codes are returned, these when rendered give the text in the image.

The problem of character recognition is the problem of automatic recognition of raster images as being letters, digits or some other symbol and it is like any other problem in computer vision

Character recognition is further classified into two types according to the manner in which input is provided to the recognition engine. Considering figure 1.2 which shows the classification hierarchy of character recognition

the two types of character recognition are:

- a. On-line character recognition
- b. Off-line character Recognition

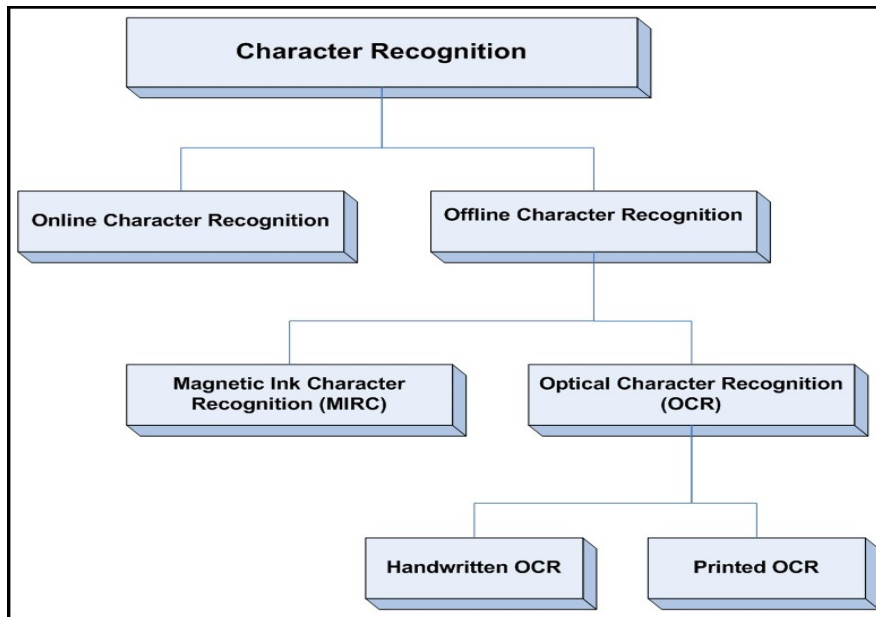


Figure 1.2: Classification of Character Recognition

c. **1.2.1 Online Character Recognition**

Online character recognition systems deal with character recognition in real time. The process involves a dynamic procedure using special sensor based equipment that can capture input from a transducer while text is being written on a pressure sensitive, electrostatic or electromagnetic digitizing tablet. The input text is automatically converted with the help of a recognition algorithm to a series of electronic signals which can be stored for further processing in the form of letter codes. The recognition system functions on the basis of the x and y coordinates generated in a temporal sequence by the pen tip movements as they create recognizable patterns on a special digitizer as the text is written.

4.1.3 Offline Character Recognition

There is a major difference in the input system of off-line and on-line character recognition which influences the design, architecture and methodologies employed to develop recognition systems for the two. In online recognition the input data is available in the form of a temporal sequence or real time text generated on a sensory device thus providing time sequence contextual information. On the contrary in an off line system the actual recognition begins after the data has been written down as it does not require real time contextual information.

Offline character recognition is further classified in two types according to the input provided to the system for recognition of characters. These are,

- i. Magnetic ink Character Recognition (MICR)
- ii. Optical Character Recognition (OCR)

4.1.4 Magnetic ink Character Recognition (MICR)

MICR is a unique technology that relies on recognizing text which has been printed in special fonts with magnetic ink usually containing iron oxide. As the machine prepares to read the code the printed characters become magnetized on the paper with the North Pole on the right of each MICR character creating recognizable waveforms and patterns that are captured and used for further processing. The reading device is comparable to a tape recorder head that recognizes the wave patterns of sound recorded on the magnetic tape. The system has been in efficient use for a long time in banks around the world to process checks as results give high accuracy rates with relatively low chances of error.

There are special fonts for MICR, the most common fonts being E-13B and CMC-7.

4.1.5 Optical Character Recognition (OCR)

Optical Character Recognition or OCR is the text recognition system that allows hard copies of written or printed text to be rendered into editable, soft copy versions. It is the translation of optically scanned bitmaps of printed or written text into digitally editable data files. An OCR facilitates the conversion of geometric source object into a digitally representable character in ASCII or Unicode scheme of digital character representation [37].

Many a times we want to have an editable copy of the text which we have in the form of a hard copy like a fax or pages from a book or a magazine. The system employs the use of an optical input device usually a digital camera or a scanner which pass the captured images to a recognition system that after passing it through a number of processes convert it to a soft copy like an MS Word document.

When we scan a sheet of paper we reformat it from hard copy to a soft copy, which we save as an image. The image can be handled as a whole but its text cannot be manipulated separately. In order to be able to do so, we need to ask the computer to recognize the text as such and to let us manipulate it as if it was a text in a word document. The OCR application does that; it recognizes the characters and makes the text editable and searchable, which is what we need. The technology has also enabled such materials to be stored using much less storage space than the hard copy materials. OCR technology has made a huge impact on the way information is stored, shared and communicated.

OCRs are of two types,

- i. OC Rs for recognizing printed characters
- ii. OC Rs for recognizing hand-written text.

OCRs meant for printed text recognition are generally more accurate and reliable because the characters belong to standard font files and it is relatively easier to match images with the ones present in the existing library. As far as hand writing recognition is concerned the vast variety of human writing styles and customs make the recognition task more challenging. Today we have OC Rs for printed text in Latin script as an everyday tool in offices while an OCR for hand writing is still in the research and development stage to have more result accuracy.

Optical Character Recognition (OCR) is one of the most common and useful applications of machine vision, which is a sub-class of artificial intelligence, and has long been a topic of research, recently gaining even more popularity with the development of prototype digital libraries which imply the electronic rendering of paper or film based documents through an imaging process.

4.1.6 History of OCR

The history of OCR dates back to the early 1950s with the invention of *Gismo* a machine that could translate printed messages into machine codes for computer processing. The product had been a combined effort of David Shepard, a cryptanalyst at (Armed Forces Security Agency) AFSA and Harvey Cook. The successful achievement was then followed by the construction of the world's first OCR system, also by David Shepard under his Intelligent Machines Research Corporation. Shepard's customers included The Readers' Digest, Standard Oil Company of California for making credit card imprints, Ohio Bell Telephone Company for a bill stub reader and The U.S. Air force for reading and transmitting teletype messages.

Another success came in the area of postal number recognition, which although crude in its inception stage became more refined with advancement in technology. By 1965 Jacob Rainbow an American inventor had patented an OCR for sorting mails which was then used by the U.S. Postal Service. The OCR has since become an interesting field of research posing numerous complexities and offering unique possibilities in the areas of Artificial Intelligence and Computer Vision. As advancements were made and new challenges were undertaken it became more and more clear that the scope of such an undertaking, though attractive would be daunting and adventurous.

4.1.7 OCR Processes

The OCR process begins with the scanning and subsequent digital reproduction of the text in the image. It involves the following discrete sub-processes, as shown in figure 1.3.

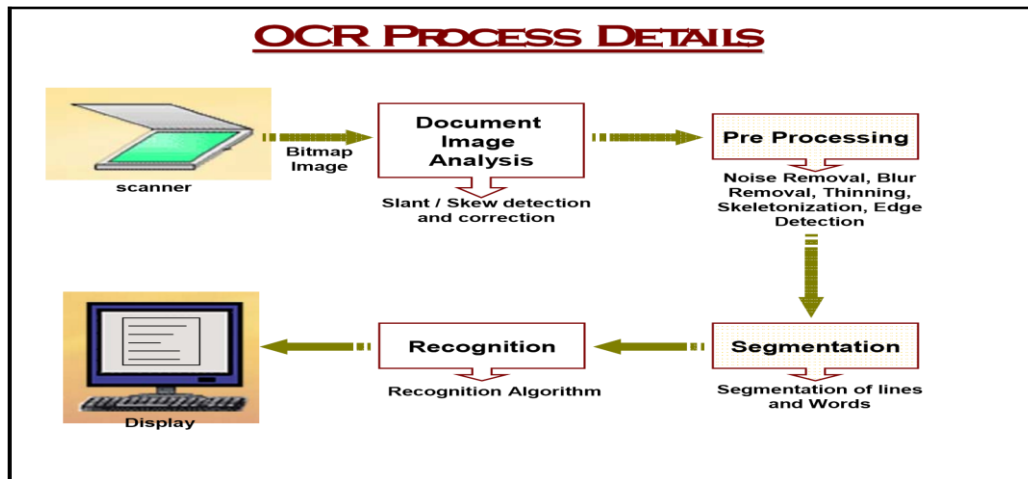


Figure 1.3: OCR Processes

4.1.8 Scanning

A flat-bed scanner is usually used at 300dpi which converts the printed material on the page being scanned into a bitmap image.

4.1.9 Document Image Analysis

The bitmap image of the text is analyzed for the presence of skew or slant and consequently these are removed. Quite a lot of printed literature has combinations of text and tables, graphs and other forms of illustrations. It is therefore important that the text area is identified separately from the other images and could be localized and extracted.

4.1.10 Pre-processing

In this phase several processes are applied to the text image like noise and blur removal, banalization, thinning, skeletonization, edge detection and some morphological processes, so as to get an OCR ready image of the text region which is free from noise and blur.

4.1.11 Segmentation

If the whole image consists of text only, the image is first segmented into separate lines of text. These lines are then segmented into words and finally words into individual letters. Once the individual letters are identified, localized and segmented

out in a text image it becomes a matter of choice of recognition algorithm to get the text in the image into a text processor.

4.1.12 Recognition

This is the most vital phase in which recognition algorithm is applied to the images present in the text image segmented at the character level. As a result of recognition character code corresponding to its image is returned by the system which is then passed to a word processor to be displayed on the screen where it can be edited, modified and saved in a new file format.

Tesseract - an open-source OCR engine that has gained popularity among OCR developers. Even though it can be painful to implement and modify sometimes, there weren't too many free and powerful OCR alternatives on the market for the longest time. Tesseract began as a Ph.D. research project in HP Labs, Bristol. It gained popularity and was developed by HP between 1984 and 1994. In 2005 HP released Tesseract as an open-source software.

OCROPUS - OCROPUS is an open-source OCR system allowing easy evaluation and reuse of the OCR components by both researchers and companies. A collection of document analysis programs, not a turn-key OCR system. To apply it to your documents, you may need to do some image preprocessing, and possibly also train new models. In addition to the recognition scripts themselves, there are several scripts for ground truth editing and correction, measuring error rates, determining confusion matrices that are easy to use and edit.

Ocular - Ocular works best on documents printed using a hand press, including those written in multiple languages. It operates using the command line. It is a state-of-the-art historical OCR system. Its primary features are:

Unsupervised learning of unknown fonts: requires only document images and a corpus of text.

Ability to handle noisy documents: inconsistent inking, spacing, vertical alignment

Support for multilingual documents, including those that have considerable word-level code-switching.

Unsupervised learning of orthographic variation patterns including archaic spellings and printer shorthand.

Simultaneous, joint transcription into both diplomatic (literal) and normalized forms.

SwiftOCR - I will also mention the OCR engine written in Swift since there is huge development being made into advancing the use of the Swift as the development programming language used for deep learning. Check out blog to find out more why. SwiftOCR is a fast and simple OCR library that uses neural networks for image recognition. SwiftOCR claims that their engine outperforms well known Tesseract library.

In this blog post, we will put focus on Tesseract OCR and find out more about how it works and how it is used.

Tesseract OCR

Tesseract is an open source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly, or (for programmers) using an API to extract printed text from images. It supports a wide variety of languages. Tesseract doesn't have a built-in GUI, but there are several available from the 3rdParty page. Tesseract is compatible with many programming languages and frameworks through wrappers that can be found here. It can be used with the existing layout analysis to recognize text within a large document, or it can be used in conjunction with an external text detector to recognize text from an image of a single text line.

OCR Process Flow to build API with Tesseract from a blog post

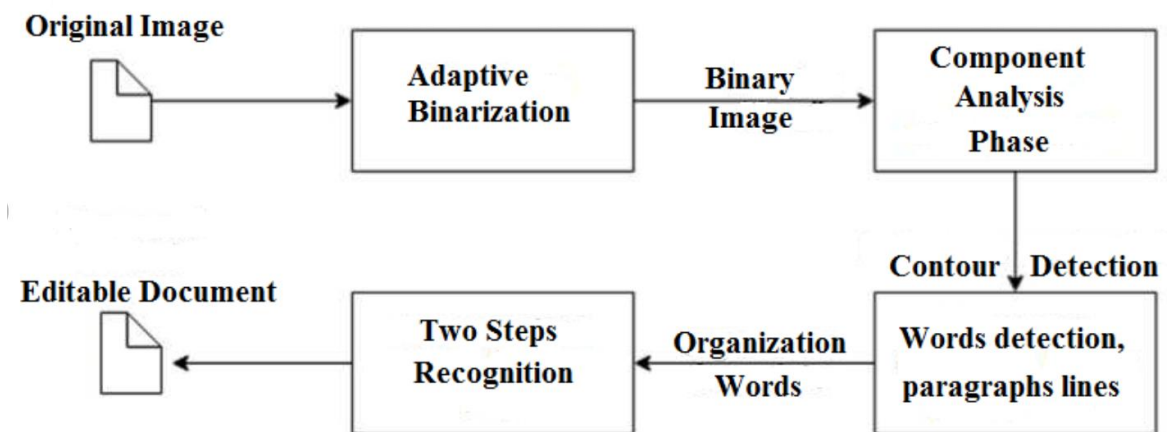
Tesseract 4.00 includes a new neural network subsystem configured as a text line recognizer.

It has its origins in OCRopus' Python-based LSTM implementation but has been redesigned for Tesseract in C++. The neural network system in Tesseract pre-dates TensorFlow but is compatible with it, as there is a network description language called Variable Graph Specification Language (VGSL), that is also available for TensorFlow.

To recognize an image containing a single character, we typically use a Convolutional Neural Network (CNN). Text of arbitrary length is a sequence of characters, and such problems are solved using RNNs and LSTM is a popular form of RNN. Read this post to learn more about LSTM.

Technology - How it works

LSTMs are great at learning sequences but slow down a lot when the number of states is too large. There are empirical results that suggest it is better to ask an LSTM to learn a long sequence than a short sequence of many classes. Tesseract developed from OCRopus model in Python which was a fork of a LSMT in C++, called CLSTM. CLSTM is an implementation of the LSTM recurrent neural network model in C++, using the Eigen library for numerical computations.



Tesseract 3 OCR process from paper

Legacy Tesseract 3.x was dependant on the multi-stage process where we can differentiate steps:

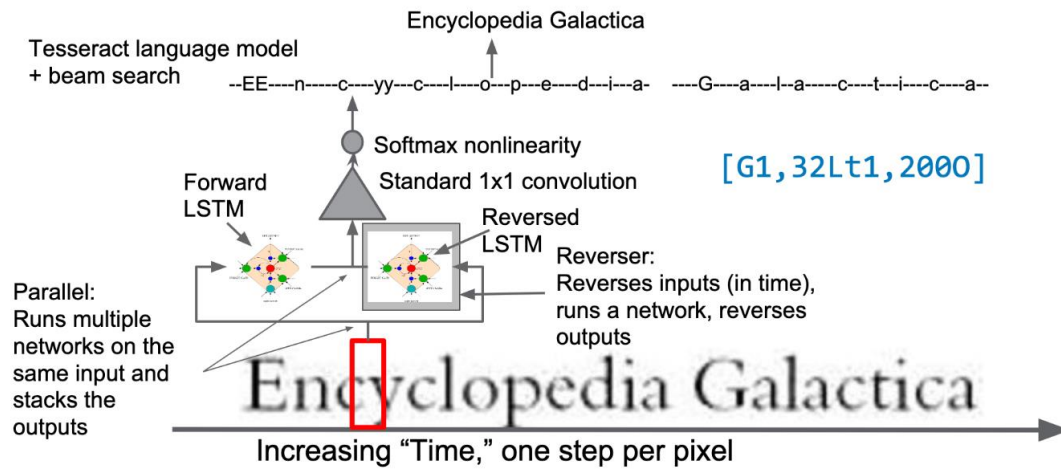
Word finding

Line finding

Character classification

Word finding was done by organizing text lines into blobs, and the lines and regions are analyzed for fixed pitch or proportional text. Text lines are broken into words differently according to the kind of character spacing. Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page.

How Tesseract uses LSTMs...



Google

Tesseract Blends Old and New OCR Technology - DAS2016 Tutorial - Santorini - Greece

Modernization of the Tesseract tool was an effort on code cleaning and adding a new LSTM model. The input image is processed in boxes (rectangle) line by line feeding into the LSTM model and giving output. In the image below we can visualize how it works.

How Tesseract uses LSTM model presentation

After adding a new training tool and training the model with a lot of data and fonts, Tesseract achieves better performance. Still, not good enough to work on handwritten text and weird fonts. It is possible to fine-tune or retrain top layers for experimentation.

Installing Tesseract

Installing tesseract on Windows is easy with the precompiled binaries found here. Do not forget to edit "path" environment variable and add tesseract path. For Linux or Mac installation it is installed with few commands.

After the installation verify that everything is working by typing command in the terminal or cmd:

```
$ tesseract --version
```

And you will see the output similar to:

```
tesseract 4.0.0
```

```
leptonica-1.76.0
```

```
libjpeg 9c : libpng 1.6.34 : libtiff 4.0.9 : zlib 1.2.8
```

```
Found AVX2
```

```
Found AVX
```

```
Found SSE
```

You can install the python wrapper for tesseract after this using pip.

```
$ pip install pytesseract
```

Tesseract library is shipped with a handy command-line tool called tesseract. We can use this tool to perform OCR on images and the output is stored in a text file. If we want to integrate Tesseract in our C++ or Python code, we will use Tesseract's API.

Running Tesseract with CLI

Call the Tesseract engine on the image with `image_path` and convert image to text, written line by line in the command prompt by typing the following:

```
$ tesseract image_path stdout
```

To write the output text in a file:

```
$ tesseract image_path text_result.txt
```

To specify the language model name, write language shortcut after -l flag, by default it takes English language:

```
$ tesseract image_path text_result.txt -l eng
```

By default, Tesseract expects a page of text when it segments an image. If you're just seeking to OCR a small region, try a different segmentation mode, using the --psm argument. There are 14 modes available which can be found [here](#). By default, Tesseract fully automates the page segmentation but does not perform orientation and script detection. To specify the parameter, type the following:

```
$ tesseract image_path text_result.txt -l eng --psm 6
```

There is also one more important argument, OCR engine mode (oem). Tesseract 4 has two OCR engines — Legacy Tesseract engine and LSTM engine. There are four modes of operation chosen using the --oem option.

- 0 Legacy engine only.
- 1 Neural nets LSTM engine only.
- 2 Legacy + LSTM engines.
- 3 Default, based on what is available.

Result of the Tesseract OCR engine

OCR with Pytesseract and OpenCV

Pytesseract is a wrapper for Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. More info about Python approach read [here](#). The code for this tutorial can be found in this repository.

```
import cv2
```

```
import pytesseract
```

```
img = cv2.imread('image.jpg')
```

```
# Adding custom options
```

```
custom_config = r'--oem 3 --psm 6'
```

```
pytesseract.image_to_string(img, config=custom_config)
```

Preprocessing for Tesseract

To avoid all the ways your tesseract output accuracy can drop, you need to make sure the image is appropriately pre-processed.

This includes rescaling, binarization, noise removal, deskewing, etc.

To preprocess image for OCR, use any of the following python functions or follow the OpenCV documentation.

Chapter 5

Artificial Intelligence

We can all agree that Artificial Intelligence has created a huge impact on the world's economy and will continue to do so since we're aiding its growth by producing an immeasurable amount of data. Thanks to the advancement in Artificial Intelligence Algorithms we can deal with such humungous data. In this blog post, you will understand the different Artificial Intelligence Algorithms and how they can be used to solve real-world problems.

To get in-depth knowledge of Artificial Intelligence and Machine Learning, you can enroll for live **Machine Learning Engineer Master Program** by Edureka with 24/7 support and lifetime access.

What Is Artificial Intelligence?

To simply put it, Artificial Intelligence is the science of getting machines to think and make decisions like human beings do.

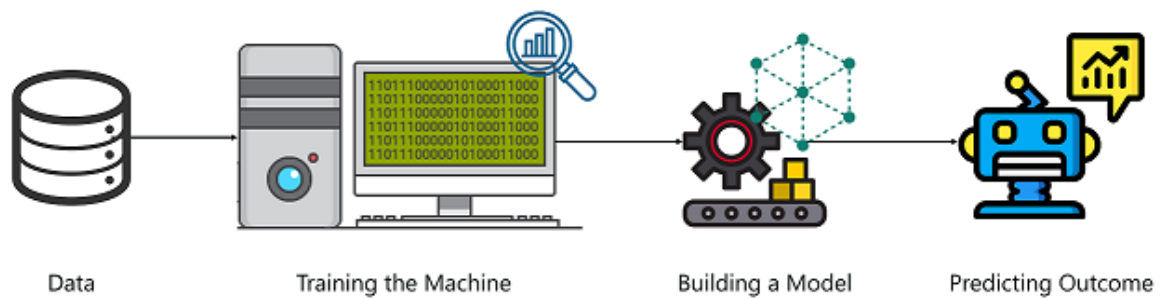
Since the development of complex Artificial Intelligence Algorithms, it has been able to accomplish this by creating machines and robots that are applied in a wide range of fields including agriculture, healthcare, robotics, marketing, business analytics and many more.

Before we move any further let's try to understand what Machine Learning is and how does it is related to AI.

What Is Machine Learning?

Generally, an algorithm takes some input and uses mathematics and logic to produce the output. In stark contrast, an Artificial Intelligence Algorithm takes a combination of both – inputs and outputs simultaneously in order to “learn” the data and produce outputs when given new inputs.

This process of making machines learn from data is what we call Machine Learning.



Machine Learning is a sub-field of Artificial Intelligence, where we try to bring AI into the equation by learning the input data.

If you're curious to learn more about Machine Learning, give the following blogs a read:

[Introduction To Machine Learning: All You Need To Know About Machine Learning](#)

[Machine Learning Tutorial for Beginners](#)

[Machine Learning Algorithms](#)

Machines can follow different approaches to learn depending on the data set and the problem that is being solved. In the below section we'll understand the different ways in which machines can learn.

Types of Machine Learning

Machine Learning can be done in the following ways:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Ensemble Learning

Let's briefly understand the idea behind each type of Machine Learning.

What Is Supervised Learning?

In **Supervised Learning**, as the name rightly suggests, it involves making the algorithm learn the data while providing the correct answers or the labels to the data. This essentially means that the classes or the values to be predicted are known and well defined for the algorithm from the very beginning.

What Is Unsupervised Learning?

The other class falls under **Unsupervised Learning**, where, unlike supervised methods the algorithm doesn't have correct answers or any answers at all, it is up to the algorithms discretion to bring together similar data and understand it.

What Is Reinforcement Learning?

Along with these two prominent classes, we also have a third class, called **Reinforcement Learning**. Just as children are generally “reinforced” certain ideas, principles by either rewarding them when doing the right thing or punishing upon doing something wrong, in Reinforcement Learning, there are rewards given to the algorithm upon every correct prediction thus driving the accuracy higher up.

Here's a short video recorded by our Machine Learning experts. This will help you understand the difference between Supervised, Unsupervised and Reinforcement learning.

What Is Ensemble Learning?

While the above three classes cover most fields comprehensively, we sometimes still land into the issue of having to bump up the performance of our model. In such cases it might make sense, to use **ensemble methods** (explained later) to get the accuracy higher up.

Now let's understand how Artificial Intelligence algorithms can be used to solve different types of problems.

Also, enroll in AI and ML courses to become proficient in this Artificial Intelligence and ML.

Types of Problems Solved Using Artificial Intelligence Algorithms

Algorithms in each category, in essence, perform the same task of predicting outputs given unknown inputs, however, here data is the key driver when it comes to picking the right algorithm.

What follows is an outline of categories of Machine Learning problems with a brief overview of the same:

- Classification
- Regression
- Clustering

Here's a table that effectively differentiates each of these categories of problems.

Regression	Classification	Clustering
• Supervised Learning • Output is a continuous quantity • Main aim is to forecast or predict • Eg: Predict stock market price • Algorithm: Linear Regression	• Supervised Learning • Output is a categorical quantity • Main aim is to compute the category of the data • Eg: Classify emails as spam or non-spam • Algorithm: Logistic Regression	• Unsupervised Learning • Assigns data points into clusters • Main aim is to group similar items clusters • Eg: Find all transactions which are fraudulent in nature • Algorithm: K-means

Type of Problems Solved Using AI – Artificial Intelligence Algorithms

For each category of tasks, we can use specific algorithms. In the below section you'll understand how a category of algorithms can be used as a solution to complex problems.

Artificial Intelligence Algorithms

As mentioned above, different Artificial Intelligence algorithms can be used to solve a category of problems. In the below section we'll see the different types of algorithms that fall under Classification, Regression and Clustering problems.

Also, enroll in AI ML Courses to become proficient in Artificial Intelligence and Machine Learning.

Classification Algorithms

Classification, as the name suggests is the act of dividing the dependent variable (the one we try to predict) into classes and then predict a class for a given input. It falls into the category of Supervised Machine Learning, where the data set needs to have the classes, to begin with. Thus, classification comes into play at any place where we need to predict an outcome, from a set number of fixed, predefined outcomes.

Classification uses an array of algorithms, a few of them listed below

1. Naive Bayes
2. Decision Tree
3. Random Forest
4. Logistic Regression
5. Support Vector Machines
6. K Nearest Neighbours

Let us break them down and see where they fit in when it comes to application.

Naive Bayes

Naive Bayes algorithm follows the Bayes theorem, which unlike all the other algorithms in this list, follows a probabilistic approach. This essentially means, that instead of jumping straight into the data, the algorithm has a set of prior probabilities set for each of the classes for your target.

Once you feed in the data, the algorithm updates these prior probabilities to form something known as the posterior probability. Hence this can be extremely useful in cases where you need to predict whether your input belongs to either a given list of n classes or does it not belong to any of them. This can be possible using a probabilistic approach mainly because the probabilities thrown for all the n classes will be quite low.

Let us try to understand this with an example, of a person playing golf, depending on factors like the weather outside.

- We first try to generate the frequencies with which certain events occur, in this case, we try to find frequencies of the person playing golf if it's sunny, rainy, etc outside.
- Using these frequencies we generate our apriori or initial probabilities (eg, the probability of overcast is 0.29 while the general probability of playing is 0.64)
- Next up, we generate the posterior probabilities, where we try to answer questions like “what would be the probability of it being sunny outside and the person would play golf?”

We use the Bayesian formula here,

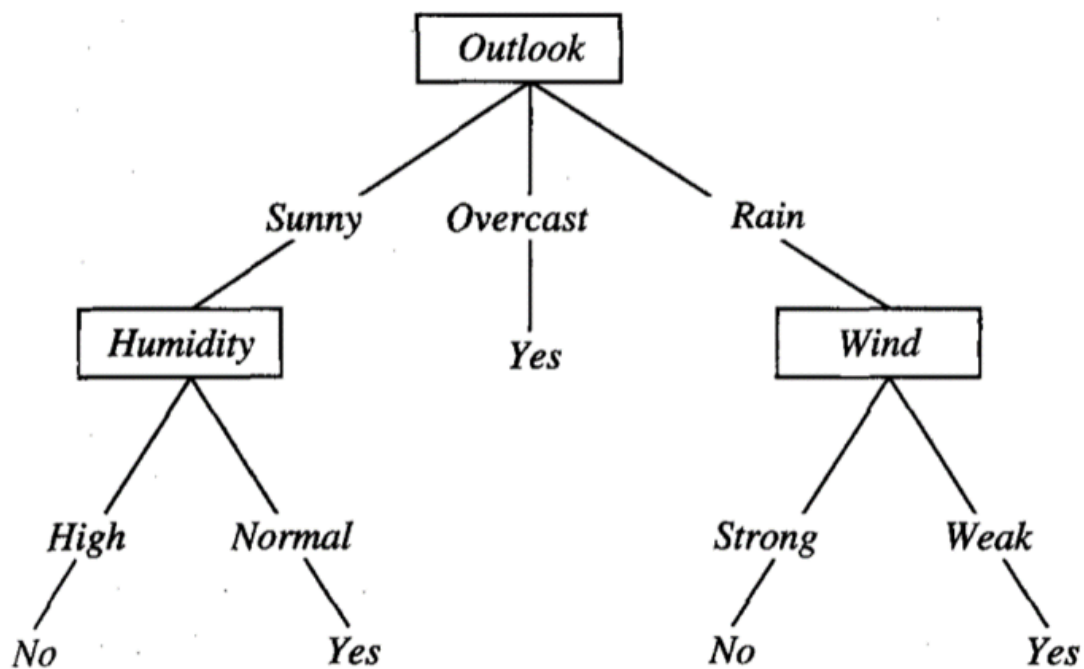
$$P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

You can go through this [A Comprehensive Guide To Naive Bayes](#) blog to help you understand the math behind Naive Bayes.

Decision Tree

The Decision Tree can essentially be summarized as a flowchart-like tree structure where each external node denotes a test on an attribute and each branch represents the outcome of that test. The leaf nodes contain the actual predicted labels. We start from the root of the tree and keep comparing attribute values until we reach a leaf node.



Decision Trees – Artificial Intelligence Algorithms

We use this classifier when handling high dimensional data and when little time has been spent behind data preparation. However, a word of caution – they tend to overfit and are prone to change drastically even with slight nuances in the training data.

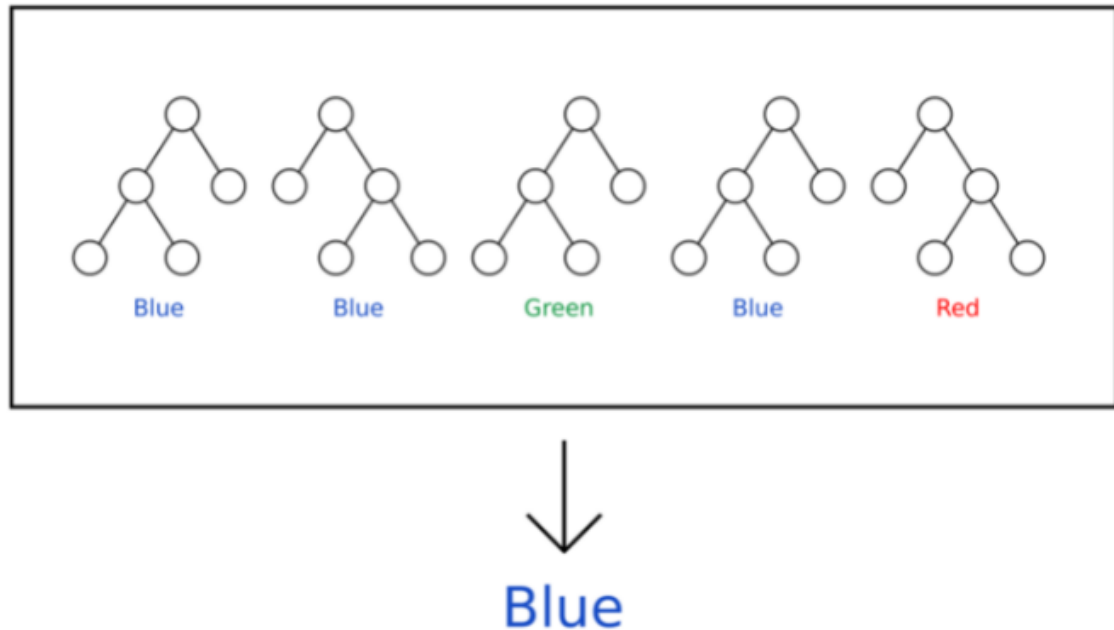
You can through these blogs to learn more about Decision Trees:

- [A Complete Guide On Decision Tree Algorithm](#)
- [Decision Tree: How To Create A Perfect Decision Tree?](#)

Random Forest

Think of this as a committee of Decision Trees, where each decision tree has been fed a subset of the attributes of data and predicts on the basis of that subset. The average of the votes of all decision trees are taken into account and the answer is given.

An advantage of using Random Forest is that it alleviates the problem of overfitting which was present in a standalone decision tree, leading to a much more robust and accurate classifier.



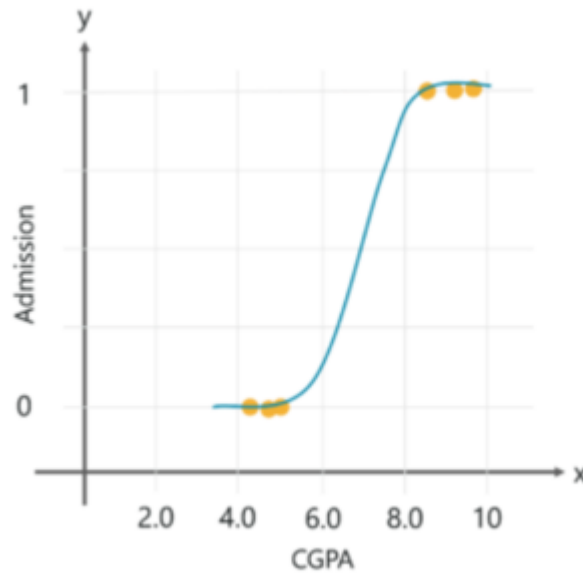
Random Forest – Artificial Intelligence Algorithms

As we can see in the above image, we have 5 decision trees trying to classify a color. Here 3 of these 5 decision trees predict blue and two have different outputs, namely green and red. In this case, we take the average of all the outputs, which gives blue as the highest weightage.

Here's a [blog](#) on Random Forest Classifier that will help you understand the working of Random forest algorithm and how it can be used to solve real-world problems.

Logistic Regression

It's a go-to method mainly for binary classification tasks. The term 'logistic' comes from the legit function that is used in this method of classification. The logistic function, also called as the sigmoid function is an S-shaped curve that can take any real-valued number and map it between 0 and 1 but never exactly at those limits.



Logistic Regression – Artificial Intelligence Algorithms

Let's assume that your little brother is trying to get into grad school, and you want to predict whether he'll get admitted in his dream establishment. So, based on his CGPA and the past data, you can use Logistic Regression to foresee the outcome.

Logistic Regression allows you to analyze a set of variables and predict a categorical outcome. Since here we need to predict whether he will get into the school or not, which is a classification problem, logistic regression would be ideal.

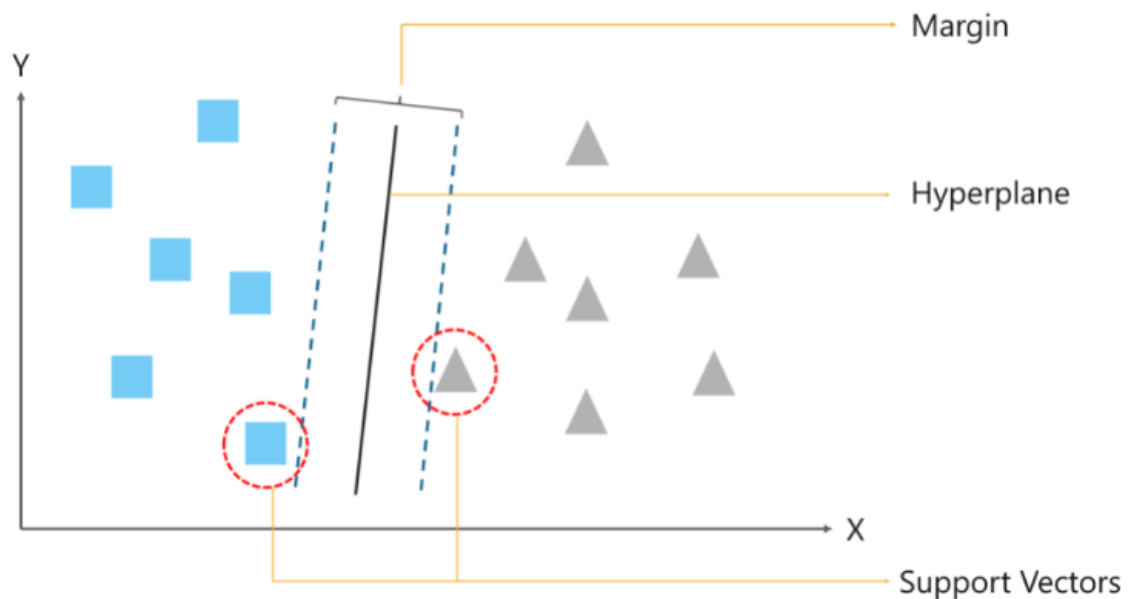
Logistic Regression is used to predict house values, customer lifetime value in the insurance sector, etc.

Support Vector Machine

An SVM is unique, in the sense that it tries to sort the data with the margins between two classes as far apart as possible. This is called maximum margin separation.

Another thing to take note of here is the fact that SVM's take into account only the support vectors while plotting the hyper plane, unlike linear regression which uses the entire dataset for that purpose. This makes SVM's quite useful in situations when data is in high dimensions.

Let's try to understand this with an example. In the below figure we have to classify data points into two different classes (squares and triangles).



Support Vector Machine – Artificial Intelligence Algorithms

So, you start off by drawing a random hyperplane and then you check the distance between the hyperplane and the closest data points from each class. These closest data points to the hyperplane are known as **Support vectors**. And that's where the name comes from, Support Vector Machine.

The hyper plane is drawn based on these support vectors and an *optimum hyper plane will have a maximum distance from each of the support vectors*. And this distance between the hyper plane and the support vectors is known as the **margin**.

To sum it up, SVM is used to classify data by using a hyper plane, such that the distance between the hyper plane and the support vectors is maximum.

To learn more about SVM, you can go through this, [Using SVM To Predict Heart Diseases](#) blog.

K Nearest Neighbors

KNN is a non-parametric (here non-parametric is just a fancy term which essentially means that KNN does not make any assumptions on the underlying data distribution), lazy learning algorithm (here lazy means that the “training” phase is fairly short).

Its purpose is to use a whole bunch of data points separated into several classes to predict the classification of a new sample point.

The following points serve as an overview of the general working of the algorithm:

- A positive integer N is specified, along with a new sample
- We select the N entries in our database which are closest to the new sample
- We find the most common classification of these entries
- This is the classification we give to the new sample

However, there are some downsides to using KNN. These downsides mainly revolve around the fact that KNN works on storing the entire dataset and comparing new points to existing ones. This means that the storage space increases as our training set increases. This also means that the estimation time increases in proportion to the number of training points.

The following blogs will help you understand how the KNN algorithm works in depth:

- [A Practical Implementation Of KNN Algorithm In R](#)
- [K-Nearest Neighbors Algorithm Using Python](#)

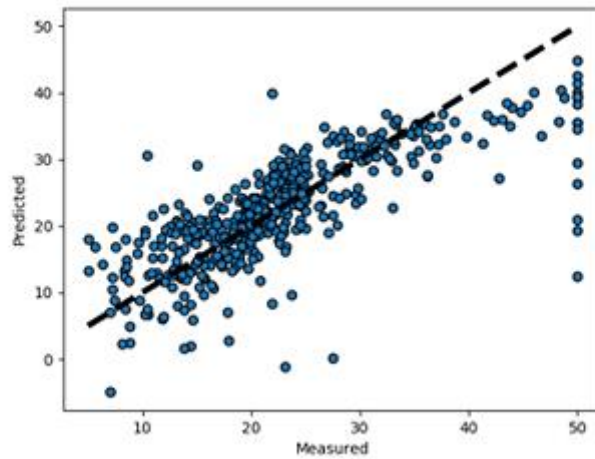
Now let's understand how regression problems can be solved by using regression algorithms.

Regression Algorithms

In the case of regression problems, the output is a *continuous quantity*. Meaning that we can use regression algorithms in cases where the target variable is a continuous variable. It falls into the category of Supervised Machine Learning, where the data set needs to have the labels, to begin with.

Linear Regression

Linear Regression is the most simple and effective regression algorithm. It is utilized to gauge genuine qualities (cost of houses, number of calls, all out deals and so forth.) in view of the consistent variable(s). Here, we build up a connection between free and ward factors by fitting the best line. This best fit line is known as regression line and spoken to by a direct condition $Y = a * X + b$.



Linear Regression – Artificial Intelligence Algorithms

Let us take a simple example here to understand linear regression.

Consider that you are given the challenge to estimate an unknown person's weight by just looking at them. With no other values in hand, this might look like a fairly difficult task, however using your past experience you know that generally speaking the taller someone is, the heavier they are compared to a shorter person of the same build. This is linear regression, in actuality!

However, linear regression is best used in approaches involving a low number of dimensions. Also, not every problem is linearly separable.

Some of the most popular applications of Linear regression are in financial portfolio prediction, salary forecasting, real estate predictions and in traffic in arriving at ETAs

Now let's discuss how clustering problems can be solved by using the K-means algorithm. Before that, let's understand what clustering is.

Clustering Algorithms

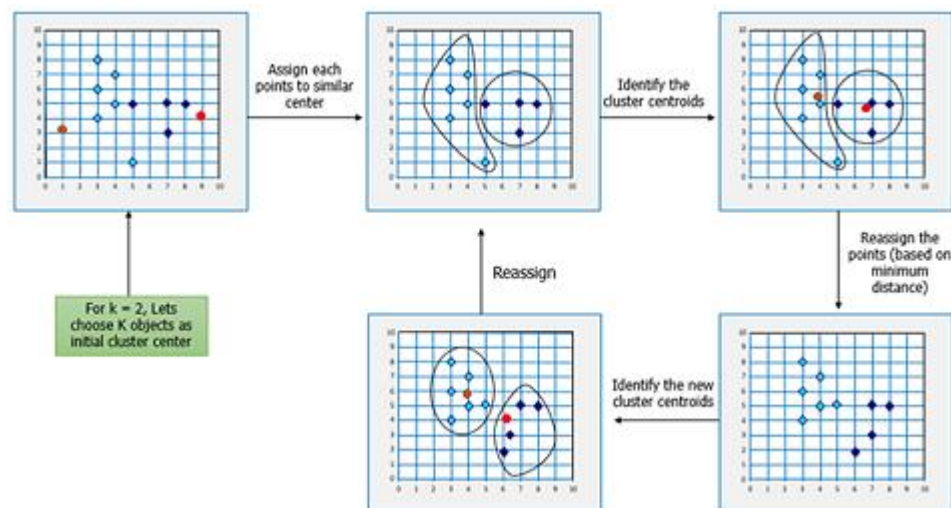
The basic idea behind clustering is to assign the input into two or more clusters based on feature similarity. It falls into the category of Unsupervised Machine Learning, where the algorithm learns the patterns and useful insights from data without any guidance (labeled data set).

For example, clustering viewers into similar groups based on their interests, age, geography, etc can be done by using Unsupervised Learning algorithms like K-Means Clustering.

K-Means Clustering

K-means is probably the simplest unsupervised learning approach. The idea here is to gather similar data points together and bind them together in the form of a cluster. It does this by calculating the centroid of the group of data points.

To carry out effective clustering, k-means evaluates the distance between each point from the centroid of the cluster. Depending on the distance between the data point and the centroid, the data is assigned to the closest cluster. The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data.



K-means – Artificial Intelligence Algorithms

The 'K' in K-means stands for the number of clusters formed. The number of clusters (basically the number of classes in which your new instances of data can fall into) is determined by the user.

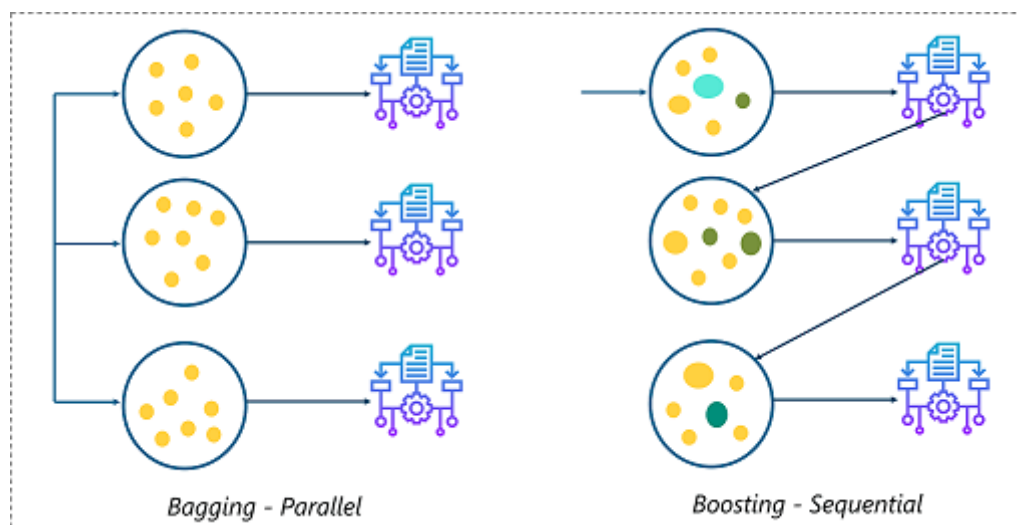
K-means is used majorly in cases where the data set has points which are distinct and well separated from each other, otherwise, the clusters won't be far apart, rendering them inaccurate. Also, K-means should be avoided in cases where the data set contains a high amount of outliers or the data set is non-linear.

So that was a brief about K-means algorithm, to learn more you can go through this content recorded by our Machine Learning experts.

Ensemble Learning Algorithms

In cases where data is of abundance and prediction precision is of high value, boosting algorithms come into the picture.

Consider the scenario, you have a decision tree trained on a data set along with a whole bunch of hyperparameter tuning already performed, however, the final accuracy is still slightly off than you'd like. In this case, while it might seem that you have run out of possible things to try, ensemble learning comes to the rescue.



Ensemble Learning – Artificial Intelligence Algorithms

You have two different ways in which you can use ensemble learning, in this case, to bump up your accuracy. Let us say your decision tree was failing on a set of input test values, what you do now is, to train a new decision tree model and give a higher weighting to those input test values that your previous model struggled with. This is also called as Boosting, where our initial tree can be formally stated as a weak learner, and the mistakes caused by that model pave way for a better and stronger model.

Another way to approach this is by simply training a whole bunch of trees at once (this can be done fairly quickly and in a parallel fashion) and then taking outputs from each tree and averaging them out. So this way, if after training 10 trees, let's say 6 trees reply positive to input and 4 trees reply negative, the output you consider is positive. This is formally known as Bagging.

They are used to reduce the bias and variance in supervised learning techniques. There are a host of boosting algorithms available, a few of them discussed below:

Gradient Boosting

Gradient Boosting is a boosting algorithm used when we deal with plenty of data to make a prediction with high prediction power. It combines multiple weak or average predictors to build strong predictor. These boosting algorithms are heavily used to refine the models in data science competitions.

Here, we consider an “optimal” or best model, so essentially our model is at some distance from that “optimal” model. What we now do is, use gradient mathematics and try to get our model closer to the optimal space.

XGBoost

Pertaining to its extremely high predictive power, XGBoost is one of the go-to algorithms when it comes to increasing accuracy as it contains both linear & tree learning algorithms making it 10 times faster than most boosting techniques.

It is the holy grail algorithm when it comes to hackathons, it is no wonder CERN used it in the model for classification of signals from the Large Hadron Collider.

If you want to learn more about Boosting Machine Learning, you can go through this, [Comprehensive Guide To Boosting Machine Learning Algorithms](#) blog.

So with this, we come to an end of this Artificial Intelligence Algorithms blog. If you wish to learn more about Artificial Intelligence, you can give these blogs a read:

- [Artificial Intelligence – What It Is And How Is It Useful?](#)
- [A Comprehensive Guide To Artificial Intelligence With Python](#)

- Artificial Intelligence Tutorial: All you need to know about AI
- AI vs Machine Learning vs Deep Learning
- AI Applications: Top 10 Real World Artificial Intelligence Applications

If you wish to enroll for a complete course on Artificial Intelligence and Machine Learning, Edureka has a specially curated Machine Learning Engineer Master Program that will make you proficient in techniques like Supervised Learning, Unsupervised Learning, and Natural Language Processing. It includes training on the latest advancements and technical approaches in Artificial Intelligence & Machine Learning such as Deep Learning, Graphical Models and Reinforcement Learning

Project Picture

OutPut Picture

CHAPTER 6

RESULT AND DISCUSSION:

The existing project consists of Raspberry-Pi, PC and a speaker. Python language is used for coding the Raspberry-pi. There are number of ways though which the input text can be converted into speech like string matching, frequency matching etc. Initially, the GPIO pins should be initialized in order to use them as input or output. This can be coded in python by importing GPIO module. At first the coding is done for matching the English alphabets with the user defined alphabets and playing the corresponding audio file. Audio files can be database or voice recordings. Python coding provides one more logic called frequency matching. In this method the frequencies on the basis of repetition of alphabets for each word should be written and then as per the coding logic all the words having matched frequencies should be called and corresponding audio for that word should be played.

Another method for converting the text into speech can be through the ASCII values of English letters. By using this method the coding length can be decreased. There are many Text to Speech converters are there but there performance depends on the fact that the output voice is how much close to the human natural voice. For example, consider a name pretty, it can be a name of a person as well as it can be defined as beautiful. Thus it depends on how the words are pronounced. Many text to speech engines does not give the proper pronunciation for such words thus combining some voice recordings can give more accurate result. The TTS system converts an English text into a speech signal with prosodic attributes that improve its naturalness. There are many systems which include prosodic processing and generation of synthesized control Parameters. The proposed system provides good quality of synthesized speech. The text processing component provides reliable grammatical classification and phonetic transcription

Prototype implementation of the proposed system is done. Fig. 5 shows the image of the Text to speech (TTS) conversion. The input string or word is given by the user to Raspberry-pi which further uses python coding and some programming logic, calls the corresponding audio file of that input text. Thus, Text to Speech conversion is done successfully for the set of words by using some

algorithm. Fig. 6 shows the hardware setup which includes Raspberrypi connected to the computer. The work can be extended for using the video characteristics of Raspberry-pi to make the proposed system more effective. The further application can also be extended to controlling the electronic devices like LED through speech signal.

The proposed Text to speech (TTS) system was designed in order to produce an equivalent acoustic signal which goes in synchronization with the text which is provided as an input to the Raspberry-Pi system. The programming strategy involved the GPIO lines, the database connectivity and the audio amplifiers. This experimental validation can be extended to conjoined sounds and words of higher complexity also. The scope of the Raspberry-Pi in speech synthesis and feature extraction is also quite boundless.

CHAPTER 7

CONCLUSIONS:

In this paper, we have described a prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. This method can effectively distinguish the object of interest from background or other objects in the camera view. To extract text regions from complex backgrounds, we have proposed a novel text localization algorithm based on models of stroke orientation and edge distributions. The corresponding feature maps estimate the global structural feature of text at every pixel. Block patterns project the proposed feature maps of an image patch into a feature vector. Adjacent character grouping is performed to calculate candidates of text patches prepared for text classification. An Adaboost learning model is employed to localize text in camera-based images. Off-the-shelf OCR is used to perform word recognition on the localized text regions and transform into audio output for blind users. Our future work will extend our localization algorithm to process text strings with characters fewer than three and to design more robust block patterns for text feature extraction. We will also extend our algorithm to handle non horizontal text strings. Furthermore, we will address the significant human interface issues associated with reading text by blind users. It has been developed by integrating features of all the hardware components and software used. In this paper, the camera acts as input for the paper. As the Raspberry Pi board is powered the camera starts streaming. The streaming data will be displayed on the screen using GUI application. When the object for label reading is placed in front of the camera then the capture button is clicked to provide image to the board. Using Tesseract library the image will be converted into data and the data detected from the image will be shown on the status bar. The obtained data will be pronounced through the ear phones using Flite library. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. In this by using highly advanced ARM11 board this paper has been implemented.

REFERENCES:

- [1] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in Proc. Comput. Vision Pattern Recognit., 2004, vol. 2, pp. II-366–II-373.
- [2] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, "Text extraction and document image segmentation using matched wavelets and MRF model," IEEE Trans Image Process., vol. 16, no. 8, pp. 2117– 2128, Aug. 2007.
- [3] K. Kim, K. Jung, and J. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12, pp. 1631–1639, Dec. 2003.
- [4] N. Giudice and G. Legge, "Blind navigation and the role of technology," in The Engineering Handbook of Smart Technology for Aging, Disability, and Independence, A. A. Helal, M. Mokhtari, and B. Abdulrazak, Eds. Hoboken, NJ, USA: Wiley, 2008.
- [5] World Health Organization. (2009). 10 facts about blindness and visual impairment.
- [6] Advance Data Reports from the National Health Interview Survey (2008).
- [7] International Workshop on Camera-Based Document Analysis and Recognition (CBDAR 2005, 2007, 2009, 2011).
- [8] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," IEEE Trans. Image Process., vol. 13, no. 1, pp. 87–99, Jan. 2004.
- [9] D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: A survey," IEEE Trans. Syst., Man, Cybern., vol. 40, no. 1, pp. 25–35, Jan. 2010.
- [10] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in Proc. Comput. Vision Pattern Recognit., 2010, pp. 2963–2970.

[11] Y. Freund and R. Schapire, “Experiments with a new boosting algorithm,” in Proc. Int. Conf. Machine Learning, 1996, pp. 148–156.

[13] An overview of the Tesseract OCR (optical character recognition) engine, and its possible enhancement for use in Wales in a pre-competitive research stage Prepared by the Language Technologies Unit (Canol-fan Bedwyr), Bangor University April 2008.

[14] A. Shahab, F. Shafait, and A. Dengel, “ICDAR 2011 robust reading competition: ICDAR Robust Reading Competition Challenge 2: Reading text in scene images,” in Proc. Int. Conf. Document Anal. Recognit., 2011, pp. 1491–1496.

[15] KReader Mobile User Guide, knfb Reading Technology Inc. (2008).

[16] S. M. Lucas, “ICDAR 2005 text locating competition results,” in Proc. Int. Conf. Document Anal. Recognit., 2005, vol. 1, pp. 80–84.

