

HAND WRITTEN MEDICAL PRESCRIPTION RECOGNITION USING MODIFIED CNN ALGORITHM

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

VALLABHANENI SAI KRISHNA
(Reg. No. 37110802)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work
VALLABHANENI SAI KRISHNA (Reg. No. 37110802) who carried
out the project entitled “ **HAND WRITTEN MEDICAL**
PRESCRIPTION RECOGNITION USING
MODIFIED CNN ALGORITHM” under my supervision from **November 2020** to
March 2020.

Internal Guide

Mr.MURARI D KAMALESH.,

Head of the Department
Dr.S. Vigneshwari, M.E., Ph.D.,
Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voice Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **VALLABHANENI SAI KRISHNA (37110802)** hereby declare that the Project Report entitled "**HAND WRITTEN MEDICAL PRESCRIPTION RECOGNITION USING MODIFIED CNN ALGORITHM**" is done by me under the guidance of **Mr.Murari D Kamalesh.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

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 thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. Murari D Kamalesh**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

We propose a camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily lives. Raspberry Pi is a revolutionary credit card sized embedded computer using which many projects that were earlier not possible or were not practical can be implemented. An Automatic book reader will be implemented, using a combination of various technologies like embedded linux, Image processing and Optical character recognition (OCR). This project, 'Handwritten Character Recognition' is a software algorithm project to recognize any hand written character efficiently on computer with input is either an old optical image or currently provided through touch input, mouse or pen. Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. In today's world advancement in sophisticated scientific techniques is pushing further the limits of human outreach in various fields of technology. One such field is the field of character recognition commonly known as OCR (Optical Character Recognition). In this fast paced world there is an immense urge for the digitalization of printed documents and documentation of information directly in digital form. And there is still some gap in this area even today. OCR techniques and their continuous improvisation from time to time is trying to fill this gap. This project is about devising an algorithm for recognition of hand written characters also known as HCR (Handwritten Character Recognition) leaving aside types of OCR that deals with recognition of computer or typewriter printed characters.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	V
	LIST OF FIGURES	VIII
1	INTRODUCTION	1
	1.1 HAND-WRITTEN CHARACTER RECOGNITION	2
2	LITERATURE SURVEY	3
3	AIM AND SCOPE	4
	3.1 AIM	4
	3.1.1 OBJECTIVES	4
	3.2 SCOPE	4
	3.2.1 ADVANTAGES	5
4	SYSTEM DESIGN & METHODOLOGY	6
	4.1 EXISTING SYSTEM	6
	4.2 SOFTWARE REQUIRED	6
	4.3 INTENDED AUDIENCE AND READING SUGGESTIONS	10
	4.4 REQUIREMENT SPECIFICATION	11
	4.5 MODULES	16
	4.6 OCR	18
	4.6.1 CHARACTER RECOGNITION	18
	4.6.2 ONLINE CHARACTER RECOGNITION	19
	4.6.2 OFFLINE CHARACTER RECOGNITION	20
	4.7 ARCHITECTURE OF THE PROPOSED SYSTEM	23
	4.8 OCR PROCESS	26
5	RESULTS AND DISCUSSION	32
6	CONCLUSION AND FUTURE WORK	34
	REFERENCES	35
	APPENDIX	37
	A. SOURCE CODE	37
	B. SCREEN SHOTS	39
	C.PAPER AND PLAGARISM	44

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.6	BLOCK DIAGRAM	18
4.6.1	CLASSIFICATION OF CHARACTER RECOGNITION	19
4.7	ARCHITECTURE OF PROPOSED SYSTEM	23
5	DETAIL RECOGNITION PERFORMANCE OF SVM	32
5	GRAPH REPRESENTATION OF ACCURACY OF SVM	33

CHAPTER 1

INTRODUCTION

N

1. INTRODUCTION

This project, 'Handwritten Character Recognition' is a software algorithm project to recognize any hand written character efficiently on computer with input is either an old optical image or currently provided through touch input, mouse or pen.

Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. It is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on character recognition has shifted to implementation of proven techniques. Optical character recognition is a scheme which enables a computer to learn, understand, improvise and interpret the written or printed character in their own language, but present correspondingly as specified by the user. Optical Character Recognition uses the image processing technique to identify any character computer/typewriter printed or hand written. A lot of work has been done in this field. But a continuous improvisation of OCR techniques is being done based on the fact that algorithm must have higher accuracy of recognition, higher persistency in number of times of correct prediction and increased execution time. The idea is to device efficient algorithms which get input in digital image format. After that it processes the image for better comparison. Then after the processed image is compared with already available set of font images. The last step gives a prediction of the character in percentage accuracy.

1.2 HAND-WRITTEN CHARACTER RECOGNITION

In today's world advancement in sophisticated scientific techniques is pushing further the limits of human outreach in various fields of technology. One such field is the field of character recognition commonly known as OCR (Optical Character Recognition). In this fast paced world there is an immense urge for the digitalization of printed documents and documentation of information directly in digital form. And there is still some gap in this area even today. OCR techniques and their continuous improvisation from time to time is trying to fill this gap. This project is about devising an algorithm for recognition of hand written characters also known as HCR (Handwritten Character Recognition) leaving aside types of OCR that deals with recognition of computer or typewriter printed characters. A novel technique is proposed for recognition English language characters using Artificial Neural Network including the schemes of feature extraction of the characters and implemented. The persistency in recognition of characters by the AN network was found to be more than 90% of times.

CHAPTER 2

LITERATURE SURVEY

Due to increased usage of digital technologies in all sectors and in almost all day to day activities to store and pass information, Handwriting character recognition has become a popular subject of research. Handwriting remains relevant, but people still want to have Handwriting copies converted into electronic copies that can be communicated and stored electronically. Handwriting character recognition refers to the computer's ability to detect and interpret intelligible Handwriting input from Handwriting sources such as touch screens, photographs, paper documents, and other sources. Handwriting characters remain complex since different individuals have different handwriting styles. This paper aims to report the development of a Handwriting character recognition system that will be used to read students and lectures Handwriting notes. The development is based on an artificial neural network, which is a field of study in artificial intelligence. Different techniques and methods are used to develop a Handwriting character recognition system. However, few of them focus on neural networks. The use of neural networks for recognizing Handwriting characters is more efficient and robust compared with other computing techniques. The paper also outlines the methodology, design, and architecture of the Handwriting character recognition system and testing and results of the system development. The aim is to demonstrate the effectiveness of neural networks for Handwriting character recognition.

CHAPTER 3

AIM AND SCOPE

3.1 AIM

The primary aim of this proposed system i.e “Handwritten Character Recognition” is a software algorithm project to recognize any hand written character efficiently on computer with input is either an old optical image or currently provided through touch input, mouse or pen. Character recognition, usually abbreviated to optical character recognition or shortened OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text.

3.1.1 OBJECTIVES

We presents a camera-based label reader to help blind persons to read names of labels on the products. 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.

3.2 SCOPE

This model will help us to recognize any hand written character efficiently on computer with input is either an old optical image or currently provided through touch input, mouse or pen.

3.2.1 ADVANTAGES

- It is effective
- It is low cost
- Automatic character recognition

CHAPTER 4

SYSTEM DESIGN & METHODOLOGY

4.1 EXISTING SYSTEM

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. K Reader Mobile runs on a cell phone and allows the user to read mail, receipts, fliers, and many other documents.

4.2 Software required:

Linux Ubuntu

Python

Open cv

Linux Ubuntu:

Ubuntu is a Linux distribution based on Debian and composed mostly of free and open-source software. Ubuntu is officially released in three editions: Desktop, Server, and Core for Internet of things devices and robots. All the editions can run on the computer alone, or in a virtual machine. Ubuntu is a popular operating system for cloud computing, with support for OpenStack. Ubuntu's default desktop has been GNOME, since version 17.10. Ubuntu is released every six months, with long-term support (LTS) releases every two years. As of 22 October 2020, the most recent long-term support release is 20.04 ("Focal Fossa"), which is supported until 2025 under public support and until 2030 as a paid option. The latest standard release is 20.10 ("Groovy Gorilla"), which is supported for nine months. Ubuntu is developed by Canonical, and a community of other developers, under a meritocratic governance model. Canonical provides security updates and support for each Ubuntu release, starting from the release date and until the release reaches its

designated end-of-life (EOL) date. Canonical generates revenue through the sale of premium services related to Ubuntu. Ubuntu is named after the Nguni philosophy of ubuntu, which Canonical indicates means "humanity to others" with a connotation of "I am what I am because of who we all are".

Ubuntu 	
Ubuntu 20.10 "Groovy Gorilla"	
Developer	Canonical Ltd.
OS family	Linux
Working state	Current
Source model	Open-source ^{[1][2]}
Initial release	20 October 2004
Latest release	Ubuntu 20.10 / 22 October 2020 ^[3]
Latest preview	Ubuntu 21.04 (Hirsute Hippo) Daily Build ^[4]
Repository	code.launchpad.net /ubuntu (https://code.launchpad.net/ubuntu)
Marketing target	Cloud computing, personal computers, servers, supercomputers, IoT
Available in	More than 55 languages by LoCos
Update method	Software Updater
Package manager	GNOME Software, APT, dpkg, Snappy, flatpak
Platforms	IA-32, x86-64

FEATURES:

A default installation of Ubuntu contains a wide range of software that includes Libre Office, Firefox, Thunderbird, Transmission, and several lightweight games such as Sudoku and chess. Many additional software packages are accessible from the built in Ubuntu Software (previously Ubuntu Software Center) as well as any other APT-based package management tools. Many additional software packages that are no longer installed by default, such as Evolution, GIMP, Pidgin, and Synaptic, are still accessible in the repositories and installable by the main tool or by any other APT-

based package management tool. Cross distribution snap packages and flat paks are also available, that both allow installing software, such as some of Microsoft's software, in most of the major Linux operating systems (such as any currently supported Ubuntu version and in Fedora). The default file manager is GNOME Files, formerly called Nautilus. All of the application software installed by default is free software. In addition, Ubuntu redistributes some hardware drivers that are available only in binary format, but such packages are clearly marked in the restricted component.

Security:

Ubuntu aims to be secure by default. User programs run with low privileges and cannot corrupt the operating system or other users' files. For increased security, the sudo tool is used to assign temporary privileges for performing administrative tasks, which allows the root account to remain locked and helps prevent inexperienced users from inadvertently making catastrophic system changes or opening security holes. Polkit is also being widely implemented into the desktop. Most network ports are closed by default to prevent hacking. A built-in firewall allows end-users who install network servers to control access. A GUI (GUI for Uncomplicated Firewall) is available to configure it. Ubuntu compiles its packages using GCC features such as PIE and buffer overflow protection to harden its software. These extra features greatly increase security at the performance expense of 1% in 32-bit and 0.01% in 64-bit. Ubuntu also supports full disk encryption as well as encryption of the home and Private directories.

Installation:

The system requirements vary among Ubuntu products. For the Ubuntu desktop release 20.04 LTS, a PC with at least 2 GHz dual-core processor, 4 GB of RAM and 25 GB of free disk space is recommended. For less powerful computers, there are other Ubuntu distributions such as Lubuntu and Xubuntu. Ubuntu supports the ARM architecture. It is also available on Power ISA, while older PowerPC architecture

was at one point unofficially supported, and now newer Power ISA CPUs (POWER8) are supported. X86 64 a.k.a. AMD64 architecture is also officially supported. Live images are the typical way for users to assess and subsequently install Ubuntu. These can be downloaded as a disk image (.iso) and subsequently burnt to a DVD and booted. Other methods include running the live version via UNetbootin, or Startup Disk Creator (a pre-installed tool on Ubuntu, available on machines already running the OS) directly from a USB drive (making, respectively, a live DVD or live USB medium). Running Ubuntu in this way is slower than running it from a hard drive, but does not alter the computer unless specifically instructed by the user. If the user chooses to boot the live image rather than execute an installer at boot time, there is still the option to then use an installer called Ubiquity to install Ubuntu once booted into the live environment. Disk images of all current and past versions are available for download at the Ubuntu web site. Various third-party programs such as Reconstructor are available to create customized copies of the Ubuntu Live DVDs (or CDs). "Minimal CDs" are available (for server use) that fit on a CD. Additionally, USB flash drive installations can be used to boot Ubuntu and Kubuntu in a way that allows permanent saving of user settings and portability of the USB-installed system between physical machines (however, the computers' BIOS must support booting from USB). In newer versions of Ubuntu, the Ubuntu Live USB creator can be used to install Ubuntu on a USB drive (with or without a live CD or DVD). Creating a bootable USB drive with persistence is as simple as dragging a slider to determine how much space to reserve for persistence; for this, Ubuntu employs casper. The desktop edition can also be installed using the Netboot image (a.k.a. netboot tarball) which uses the debian-installer and allows certain specialist installations of Ubuntu: setting up automated deployments, upgrading from older installations without network access, LVM or RAID partitioning, installs on systems with less than about 256 MB of RAM (although low-memory systems may not be able to run a full desktop environment reasonably).

Package Archives

A Personal Package Archive (PPA) is a software repository for uploading source

packages to be built and published as an Advanced Packaging Tool (APT) repository by Launchpad. While the term is used exclusively within Ubuntu, Launchpad's host, Canonical, envisions adoption beyond the Ubuntu community.

Third-party software:

Some third-party software that does not limit distribution is included in Ubuntu's multiverse component. The package `ubuntu-restricted-extras` additionally contains software that may be legally restricted, including support for MP3 and DVD playback, Microsoft TrueType core fonts, Sun's Java runtime environment, Adobe's Flash Player plugin, many common audio/video codecs, and `unrar`, an unarchiver for files compressed in the RAR file format. Additionally, third-party application suites are available for purchase through Ubuntu Software, including many games such as *Braid* and *Oil Rush*, software for DVD playback and media codecs.

4.3 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.

4.4 REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the customer's machines already. So the main need is to introduce appropriate equipment for the product. The requirements are split into two categories, namely:

Hardware Requirements

The basic hardware required to run the program are:

1. Hard disk of 20 GB
2. System memory (RAM) of 512 MB
3. I3 processor-based computer or higher.

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. 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.

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?.

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.

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.

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.

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.

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

trainin

g

4.5 MODULES:

Pre-processing

Pre-processing is a common name for operations with images at the lowest level of abstraction of both input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortion. Some of the point processing techniques include: contrast stretching, global thresholding, histogram equalization, log transformations and power law transformations. Some mask processing techniques include averaging filters, sharpening filters, local thresholding... etc.

Different techniques:

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. or enhances some image features important for further processing.

Image Processing

→ Image Processing is a technique to enhance raw images received from cameras/sensors placed on satellites, space probes and aircrafts or pictures taken in normal day-today life for various applications.

→ Various techniques have been developed in Image Processing during the last four to five decades. Most of the techniques are developed for enhancing images obtained from unmanned spacecraft's, space probes and military reconnaissance flights. Image Processing systems are becoming popular due to easy availability of powerful personnel.

→ computers, large size graphics software's etc. Image Processing is used in various Techniques.

Edge detection

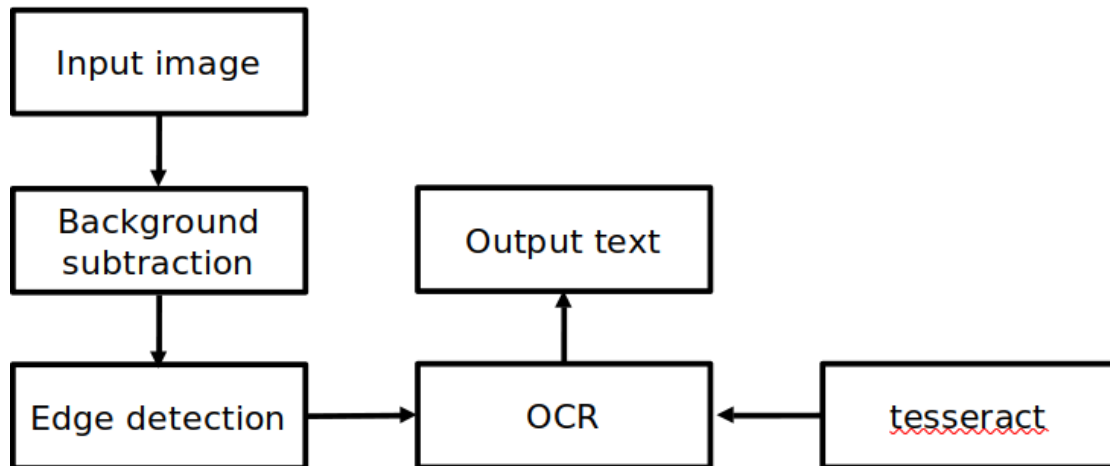
Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.

Tesseract

An optical character recognition (OCR) engine Tesseract is an OCR engine with support for unicode and the ability to recognize more than 100 languages out of the box. It can be trained to recognize other languages.

SOFTWARE DEVELOPMENT METHODOLOGY

4.6 OCR



BLOCK DIAGRAM

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.

4.6.1 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

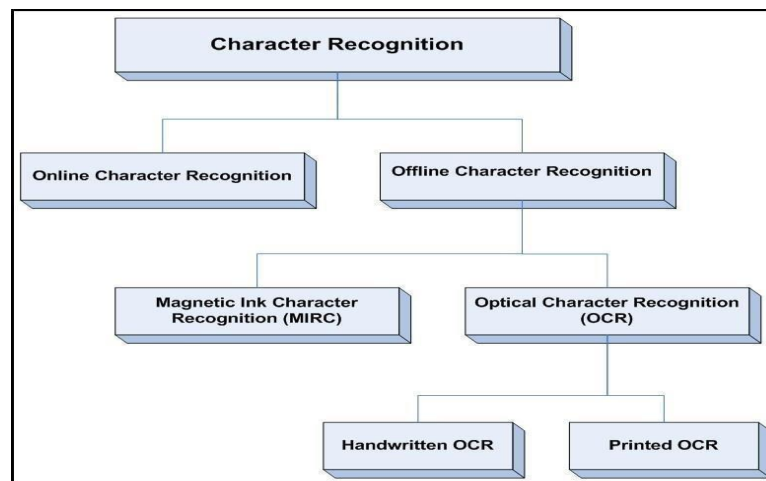


Figure 4.6.1: Classification of Character Recognition

4.6.2 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.

4.6.2 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)

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.

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. OCrs for recognizing printed characters
- ii. OCrs 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 OCrs for printed text in Latin script as an everyday tool in offices while an OCR for hand writing is still in the

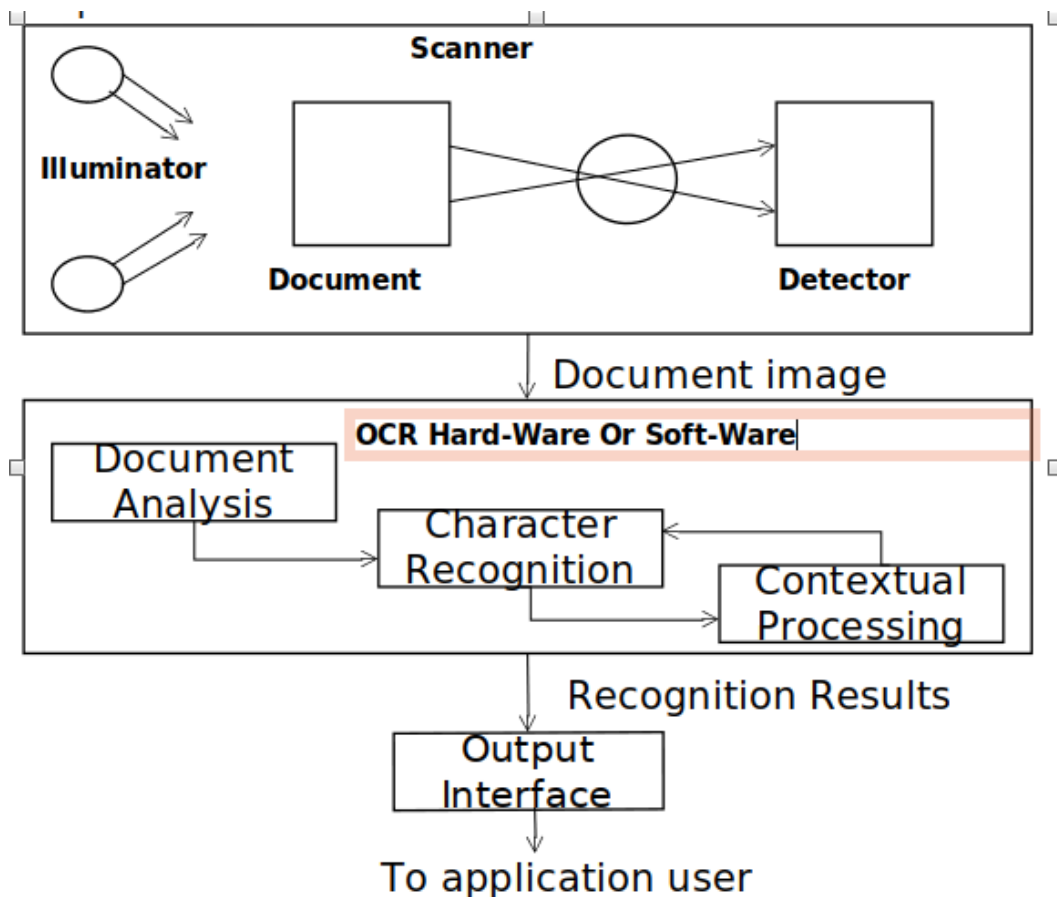
research and development stage to have more result accuracy.

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.

4.7 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:-

- A. Scanner
- B. OCR Hardware or Software
- C. Output Interface

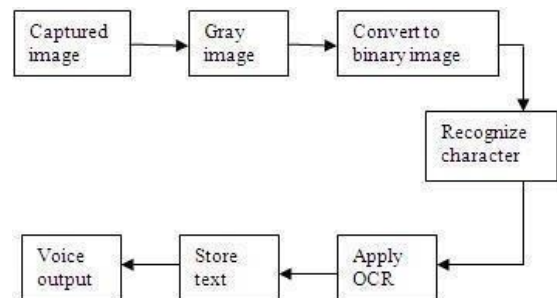


VISION BASED ASSISTIVE SYSTEM FOR LABEL DETECTION WITH VOICE OUTPUT

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.

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.

AUTOMATIC TEXT EXTRACTION

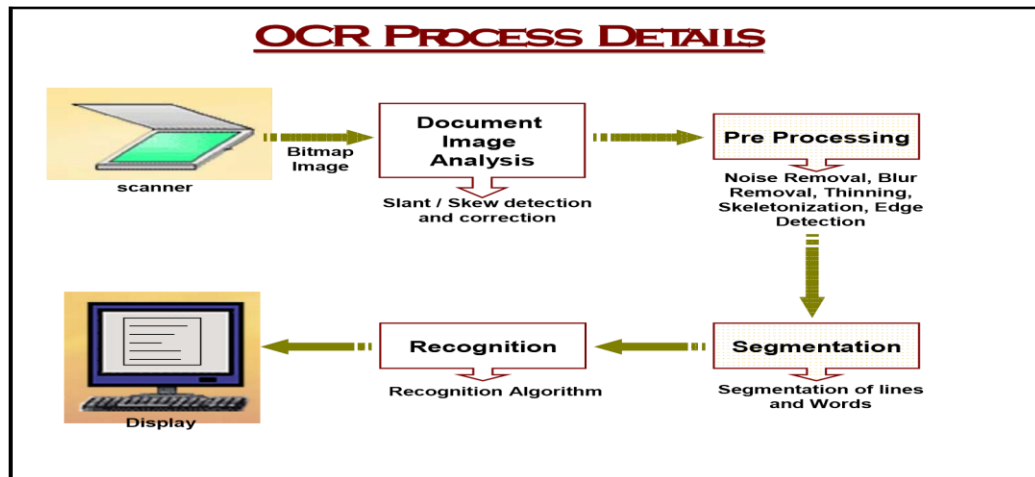
In order to handle complex backgrounds, two novel feature maps to extract text features based on stroke orientations and edge distributions, respectively. Here, stroke is defined as a uniform region with bounded width and significant extent. These feature maps are combined to build an Adaboost based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

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.

4.8 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



Scanning

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

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.

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.

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.

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. Offline Handwritten Text Recognition (HTR) systems transcribe text contained in scanned images into digital text, an example is shown in Fig. 1. We will build a Neural Network (NN) which is trained on word- images from the IAM dataset. As the input layer (and therefore also all the other layers) can be kept small for word-images, NN-training is feasible on the CPU (of course, a GPU would be better). This implementation is the bare minimum that is needed for HTR using TF.

Model Overview

We use a NN for our task. It consists of convolutional NN (CNN) layers, recurrent NN (RNN) layers and a final Connectionist Temporal Classification (CTC) layer.

We can also view the NN in a more formal way as a function (see Eq. 1) which maps an image (or matrix) M of size $W \times H$ to a character sequence $(c_1, c_2, \dots)$ with a length between 0 and L . As you can see, the text is recognized on character-level, therefore words or texts not contained in the training data can be recognized too (as long as the individual characters get correctly classified).

$$\text{NN: } M \rightarrow (c_1, c_2, \dots, c_n)$$

$W \times H$
 $0 \leq n \leq L$

Eq. 1: The NN written as a mathematical function which maps an image M to a character sequence $(c_1, c_2, \dots)$.

Operations

CNN: the input image is fed into the CNN layers. These layers are trained to extract relevant features from the image. Each layer consists of three operation. First, the convolution operation, which applies a filter kernel of size 5×5 in the first two layers and 3×3 in the last three layers to the input. Then, the non-linear RELU function is

applied. Finally, a pooling layer summarizes image regions and outputs a downsized version of the input. While the image height is downsized by 2 in each layer, feature maps (channels) are added, so that the output feature map (or sequence) has a size of 32×256 .

RNN: the feature sequence contains 256 features per time-step, the RNN propagates relevant information through this sequence. The popular Long Short- Term Memory (LSTM) implementation of RNNs is used, as it is able to propagate information through longer distances and provides more robust training- characteristics than vanilla RNN. The RNN output sequence is mapped to a matrix of size 32×80 . The IAM dataset consists of 79 different characters, further one additional character is needed for the CTC operation (CTC blank label), therefore there are 80 entries for each of the 32 time-steps.

CTC: while training the NN, the CTC is given the RNN output matrix and the ground truth text and it computes the **loss value**. While inferring, the CTC is only given the matrix and it decodes it into the **final text**. Both the ground truth text and the recognized text can be at most 32 characters long.

Data

Input: it is a gray-value image of size 128×32 . Usually, the images from the dataset do not have exactly this size, therefore we resize it (without distortion) until it either has a width of 128 or a height of 32. Then, we copy the image into a (white) target image of size 128×32 . This process is shown in Fig. 3. Finally, we normalize the gray-values of the image which simplifies the task for the NN. Data augmentation can easily be integrated by copying the image to random positions instead of aligning it to the left or by randomly resizing the image.

Fig. 3: Left: an image from the dataset with an arbitrary size. It is scaled to fit the target image of size 128×32 , the empty part of the target image is filled with white color.

CNN output:

Fig. 4 shows the output of the CNN layers which is a sequence of length 32. Each entry contains 256 features. Of course, these features are further processed by the RNN layers, however, some features already show a high correlation with certain high-level properties of the input image: there are features which have a high correlation with characters (e.g. “e”), or with duplicate characters (e.g. “tt”), or with character-properties such as loops (as contained in handwritten “l”s or “e”s).



Fig. 4: Top: 256 feature per time-step are computed by the CNN layers. Middle: input image. Bottom: plot of the 32nd feature, which has a high correlation with the occurrence of the character “e” in the image.

RNN output:

Fig. 5 shows a visualization of the RNN output matrix for an image containing the text “little”. The matrix shown in the top-most graph contains the scores for the characters including the CTC blank label as its last (80th) entry. It can be seen that most of the time, the characters are predicted exactly at the position they appear in the image (e.g. compare the position of the “i” in the image and in the graph). Only the last character “e” is not aligned. But this is OK, as the CTC operation is segmentation-free and does not care about absolute positions. From the bottom-most graph showing the scores for the characters “l”, “i”, “t”, “e” and the CTC blank label, the text can easily be decoded: we just take the most probable character from each time-step, this forms the so called best path, then we throw away repeated characters and finally all blanks: “l---i--t-t-l-...-e” → “l---i--t-t-l-...-e” → “little”.



Fig. 5: Top: output matrix of the RNN layers. Middle: input image. Bottom: Probabilities for the characters “l”, “i”, “t”, “e” and the CTC blank label.

CHAPTER 5

RESULTS AND DISCUSSION

Experimental Results

Test application Analysis:

The test application accompanying the source code can perform the recognition of handwritten digits. To do so, open the application (preferably outside Visual Studio, for better performance). Click on the menu File and select Open. This will load some entries from the Optdigits dataset into the application. To perform the analysis, click the Run Analysis button. Please be aware that it may take some time. After the analysis is complete, the other tabs in the sample application will be populated with the analysis' information. The level of importance Experiments were performed on different samples having mixed scripting languages on numerals using single hidden layer.

Data set	Training Set Size	Testing Set Size	Validation Set Size	Training Set Accuracy	Test Set Accuracy	Validation Set Accuracy
Digit	1778	6270	5430	96	97	96

Table: Detail Recognition performance of SVM

It is observed that recognition rate using SVM is higher than other model, i.e. Hidden Markov Model. However, free parameter storage for SVM model is significantly higher. The memory space required for SVM will be the number of support vectors multiply by the number of feature values. This is significantly large compared to HMM which only need to store the weight. HMM needs less space due to the weight-sharing scheme. However, in SVM, space saving can be achieved by storing only the original online signals and the penup/ pen-down status in a compact manner. During recognition, the model will be expanded dynamically as required. SVM clearly outperforms in all three isolated character cases. The result for the isolated character cases above indicates that the recognition rate for the hybrid word recognizer could be improved by using SVM instead of HMM.

Results:

After the analysis has been completed and validated, we can use it to classify the new digits drawn directly in the application. We can see the analysis also performs rather well on completely new and previously unseen data.

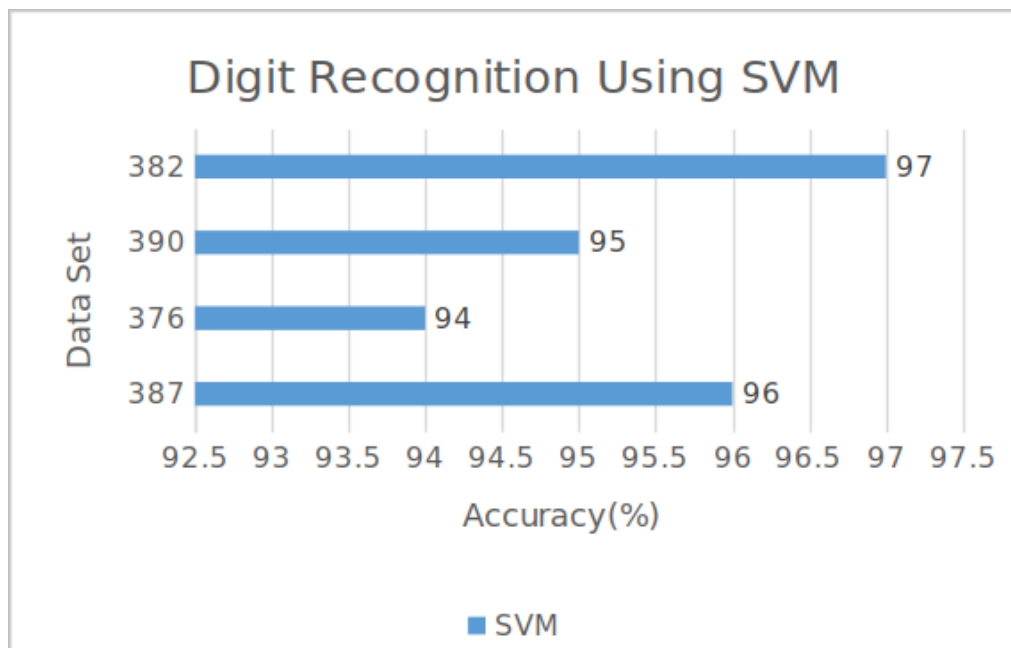


Figure: Graph representation of accuracy of SVM

CHAPTER 6

CONCLUSION AND FUTURE WORK

CONCLUSION

In this, we detailed and explored how (Kernel) Support Vector Machines could be applied in the problem of handwritten digit recognition with satisfying results. The suggested approach does not suffer from the same limitations of Kernel Discriminant Analysis, and also achieves a better recognition rate. Unlike KDA, the SVM solutions are sparse, meaning only a generally small subset of the training set will be needed during model evaluation. This also means the complexity during the evaluation phase will be greatly reduced since it will depend only on the number of vectors retained during training. SVM model requires the most space since each support vector (SV) consist of many feature values . However, space saving can be achieved by storing only the original online signals and the pen-up/pen-down status corresponding to the SV in a compact manner. During recognition, the model will be expanded dynamically as required. Experiments using SVMs with probabilistic output were also performed on the same datasets for comparison. In many experiments, the results have shown that at character level, SVM recognition rates are significantly better due to structural risk minimization implemented by maximizing margin of separation in the decision function.

FUTURE WORK

Future works on the database includes extending the character class collection by including all the presently used valid orthographic shapes for specific language script and creating word, line and page level collection of document images so that the researchers can focus on other stages of document recognition system as well. It has been shown that Support Vector Machines (SVMs) can be applied to image and handwritten character recognition . However, SVMs don't perform well in large datasets as the training time becomes cubic in the size of the dataset. This could be an issue as bigger datasets dataset containing thousand of samples which is quite large. To deal with this issue, a technique can be proposed ,which is to train a support vector machine on the collection of nearest neighbours in a solution they called "SVM-KNN" .

REFERENCES

1. X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in Proc. Comput. Vision Pattern Recognition, 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.
12. 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.
13. 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.
14. KReader Mobile User Guide, knfb Reading Technology Inc. (2008).
15. S. M. Lucas, "ICDAR 2005 text locating competition results," in Proc. Int. Conf. Document Anal. Recognit., 2005, vol. 1, pp. 80–84.

APPENDIX

A. Source code

```
import cv2
from time import sleep
import os
import requests
import io
import json
import os
import random
import pyttsx3
resim = "images.jpg"
img =
cv2.imread(resim)
print("Picture is Detected")

api = img

# Ocr
url_api = "https://api.ocr.space/parse/image"
_, compressedimage = cv2.imencode(".jpg", api, [1, 90])
file_bytes = io.BytesIO(compressedimage)

result = requests.post(url_api,
                        files={resim: file_bytes},
                        data={"apikey": "helloworld",
                            "language": "eng"})

result = result.content.decode()
print(result)
result = json.loads(result)

parsed_results = result.get("ParsedResults")[0]
text_detected = parsed_results.get("ParsedText")
print(text_detected)
print("Text is writing to file...")
f = open("text_detected.txt", "a+")
f.write(text_detected)
f.close()
print("Operation is successful")
engine = pyttsx3.init()
engine.say(text_detected)
engine.runAndWait()
```

```
cv2.imshow("roi", api)
```

```
#cv2.imshow("Img", img)
cv2.waitKey(0)
os.remove(resim)
```

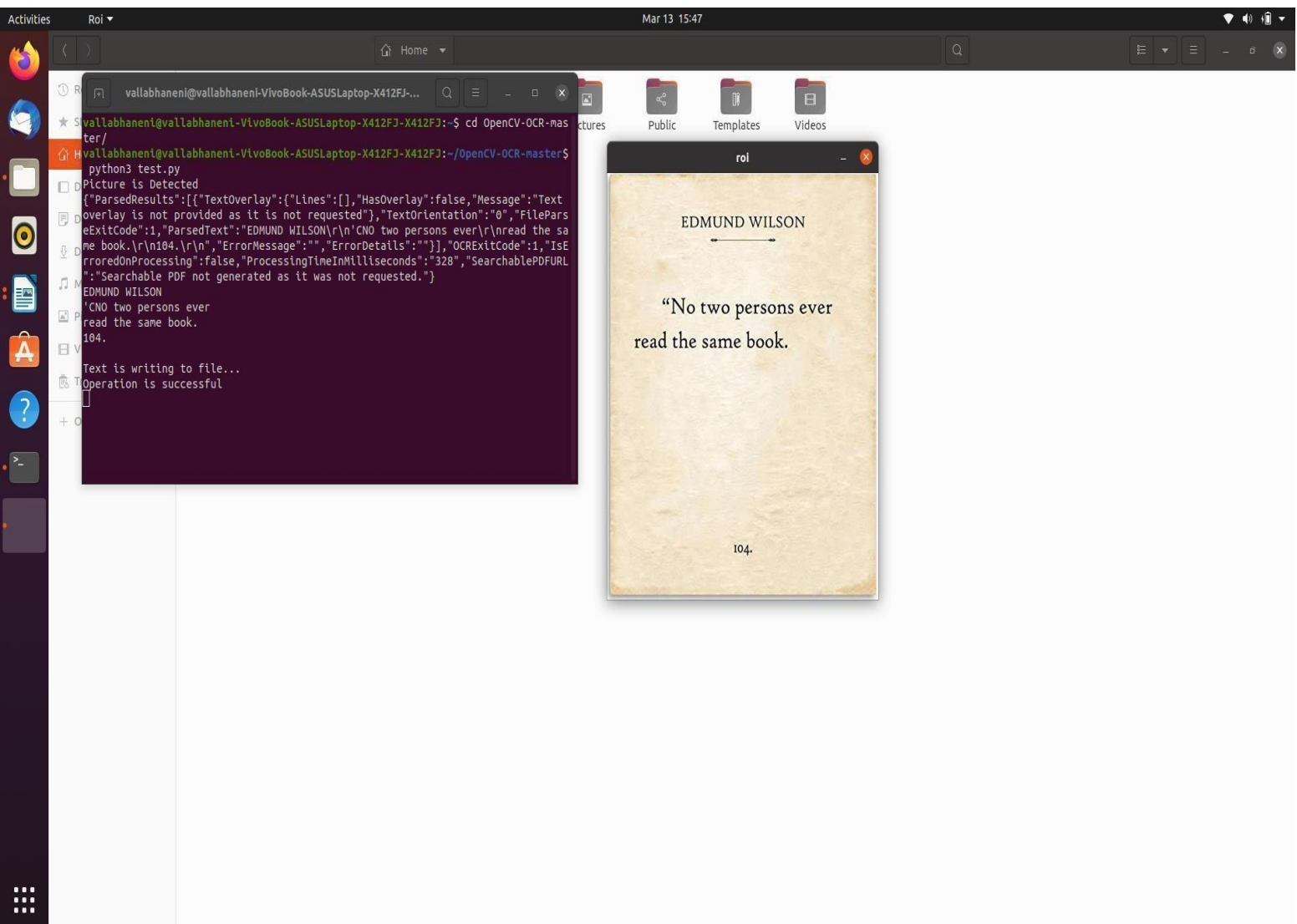
B. screen shots

Importing the image



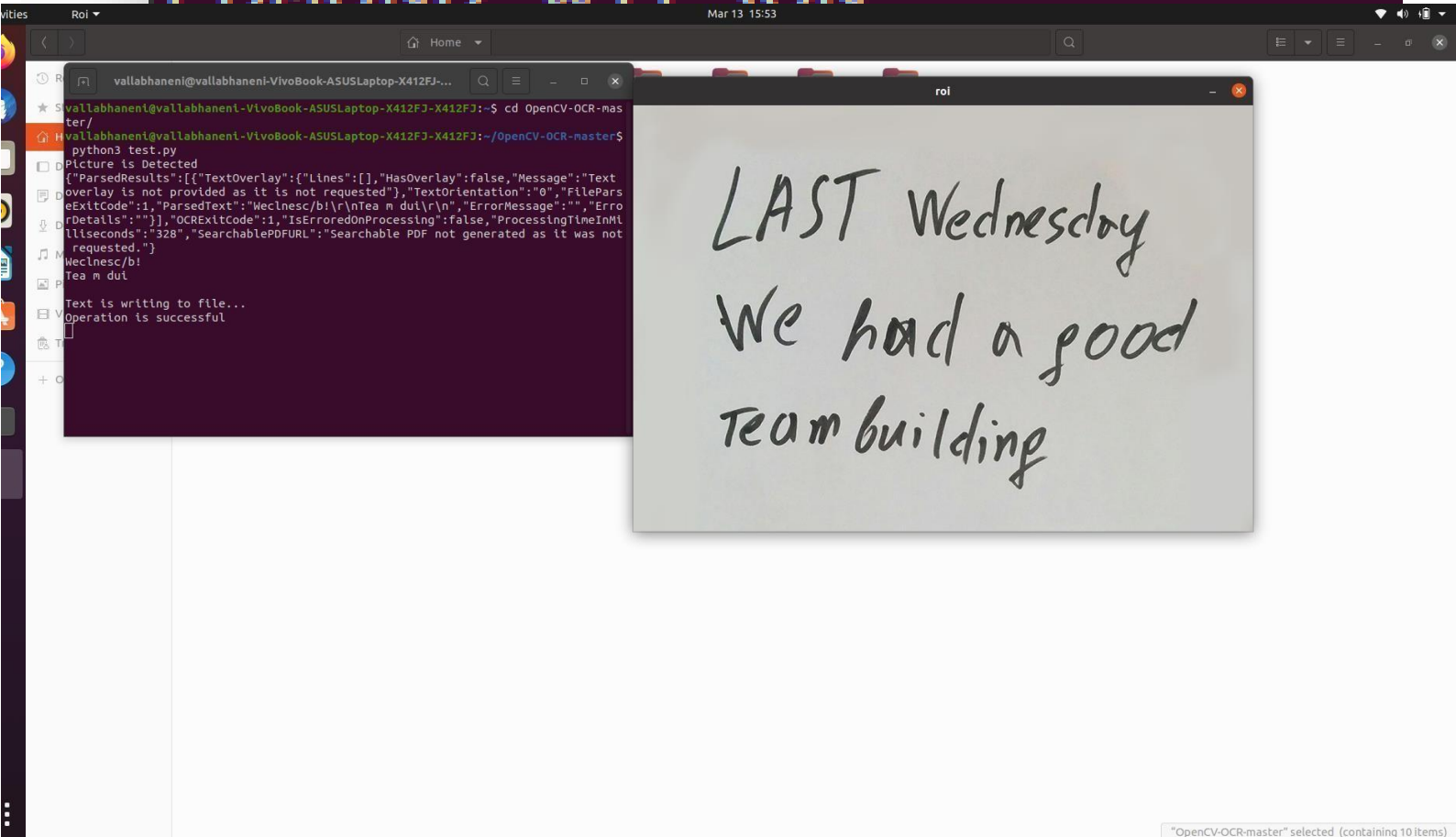
```
vallabhaneni@vallabhaneni-VivoBook-ASUSLaptop-X412FJ-...  
vallabhaneni@vallabhaneni-VivoBook-ASUSLaptop-X412FJ-X412FJ:~$ cd OpenCV-OCR-master/  
vallabhaneni@vallabhaneni-VivoBook-ASUSLaptop-X412FJ-X412FJ:~/OpenCV-OCR-master$ python3 test.py
```

Image converted into text



converted text

": "Searchable PDF not g
EDMUND WILSON



"OpenCV-OCR-master" selected (containing 10 items)

C.PAPER AND PLAGARISM

HAND WRITTEN MEDICAL PRESCRIPTION RECOGNITION USING MODIFIED CNN ALGORITHM

PROJECT SUPERVISOR :- Mr.Murari D kamalesh.

TEAM MEMBERS :-

- 1 .Saikrishna-37110802;**
- 2. Kishore- 37110803.**

Submission date: 28-Feb-2021 10:03AM (UTC-0500)

Submission ID: 1520166956

File name: AL_PRESCRIPTION_RECOGNITION_USING_MODIFIED_CNN_ALGORITHM_1.docx (38.77K)

Word count: 804

Character count: 4704

HAND WRITTEN MEDICAL PRESCRIPTION RECOGNITION USING MODIFIED CNN ALGORITHM

ABSTRACT

- Introduce camera-based labels to assist daze individuals with learning the label names in items. The camera is a good thought to discover the item name picture or wood board. The picture is handled inside and the name is isolated from the picture utilizing the open CV library . At long last, the item is perceived and the brand name is articulated by voice.
 - Now, it can perceive the utilization of tesseract library to change over the Not picture of the name into text. At the point when the predetermined mark name is chanced over to message and the chanced over content is show n on the man-machine interface associated with the regulator. Presently, the chanced over content ought to he changed over to discourse so the label name can be heard as discourse through earphones associated with the sound jack port utilizing the flite library.
-

OBJECTIVE

- We propose a camera-supported content refining asset to help discouraged individuals read text checks and heaps of hand-held articles in regular day to day existence.
- **Raspberry Pi** is an inserted PC comparable to a Visa used to make numerous undertakings unthinkable or incomprehensible previously.
- It will coordinate an assortment of advances like installed Linux, picture preparing and Optical character recognition (OCR).

EXISTING SYSTEM

- Blind individuals should utilize customary braille-like innovation to track and understand text, which is slow and powerful.
- The current OCR framework isn't computerized and requires the utilization of a total PC, so it doesn't work.
- K Reader Mobile works on the telephone and permits clients to understand messages, receipts, handouts and numerous different archives.

PROPOSED SYSTEM

- A minimal effort robotized perusing framework will be accessible, which can change over printed course readings into computerized text, yet additionally read them as sound yield.
- Our proposed calculation can successfully oversee complex spaces and different examples and concentrate text data from close by items and images.

ADVANTAGES

- It is Effective.
- It is Low cost.
- **Automatic** character recognition.

LITERATURE SURVEY

- **A System-Prototype Representing 3D Space via Alternative Sensing for Visually Impaired Navigation:** This article proposes how to speak with the closest 3-D space for the outwardly debilitated in free driving is a vital objective, including a few significant difficulties.
- **A System for Identification of Dangerous Spots and POIs for the Blind and Visually Impaired:** This article diagrams the way toward building up an arrangement that thinks about the territories of danger to the outwardly debilitated and

outwardly disabled worries in metropolitan zones.

- A survey on wearable devices used to assist the visual impaired

US42R FIBYiQ8tiOlt iFl OUtdOOfi

environments: This article leads a top to bottom investigation of wearable projects to help outwardly hindered clients use them in their day by day lives.

BLOCK DIAC'.RAM:

SOFTWARE REQUIREMENTS

- Linux Ubuntu
- Python
- Open cs'

CONCLUSION

In this work, to improve the presentation of advanced penmanship acknowledgment, we investigated an assortment of conv'olucional neural organizations, which dodge complex pre-preparing, exorbitant component

Input image

subtraction

Edge detection

extraction, and complex joining techniques (classifier blend) of conventional

acknowledgment frameworks. With broad investigation utilizing the MNIST information base, the current work has exhibited the part of different hyperparameters. We additionally affirmed t hat appropriate hyperparameter change is fundamental to improve CNN's presentation. Utilizing the Adam enhancer of the MNIST information base, we acquired a 99.89%/c acknowledgment rate, which is superior to any recentl y revealed results. The examinations obviously show the impact of expanding the quantit y of convolutional layers in the CN N structure in the exhibition of computerized

acknowledgment physically. The advancement of this work is that it completely explores all the requirements of C NN development, which gives the best precision of MNIST information choice. Shared experts utilizing unadulterated CNN models can't echindmate this exactness. A few scientists are utilizing CN N's coordinated organization arrangements for a similar arrangement information to improve its exactness, yet because of extra PC costs and the intricac y of cutting-edge testing, the precision has arrived at this

Output text

ocr

tesseract



degree of exploration. Later on,
we can examine CN N's

various half breeds, to be specific the CNN mixture, specifically CNN-RNN and CNN-HMM models, just as in usefulness programs in a specific field. You can test transformative calculations to improve CNN learning boundaries, i.e., number of convolution channel layers, understanding rate, and partition size.

REFERENCES

- [1] E. Barnes, "Image recognition for shipping container tracking and
1. D., "Adv. Imag., vol. 10, no. 1, pp. 61—62, 1995.
- [2] M. S. Brown and W. B. Seales, "Document restoration using 3D shape: A general deskewing algorithm for arbitrarily warped documents," in Proc. IC CV, vol. 2, 2001, pp. 367-374.
- [3] Y. Cui and Q. Huang, "Character extraction of license plates from video," in Proc. CVPR, 1997, pp. 502-507.
- [4] D. Deng, K. P. Chan, and Y. Yu, "Handwritten Chinese character recognition using spatial Gabor filters and self-organizing feature maps," in Proc. ICIP, vol. 3, 1994, pp. 940-944.

[1] R. C. Gonzalez and R. E. Woods, Digital Image Processing. Reading, MA: Addison-Wesley, 1993.

HAND WRITTEN MEDICAL PRESCRIPTION RECOGNITION USING MODIFIED CNN ALGORITHM (1).docx

ORIGINALITY REPORT

5%

5%

INTERNET SOURCES

5%

PUBLICATIONS

1%

STUDENT PAPER

SIMILARITY INDEX

PRIMARY SOURCES

1

www.matec-conferences.org

Internet Source

2%

2

www.miviclub.org

Internet Source

2%

3

researchr.org

Internet Source

1%

Exclude quotes bibliography	On Exclude On	Exclude matches	Off
--------------------------------	------------------	-----------------	-----