OFF-LINE HANDWRITTEN DIGITS RECOGNITION USING MACHINE LEARNING

Graduation Project

Project report submitted in partial fulfillment of the requirement for the award of the Degree of Bachelor of Science in Computer Science/Information Technology

By

Abdulrahman Yousef AlQunaibit (341101866)

Nemer Mahmoud AlBalawi (341106053)

Raaed Dogaiem AlMohlsi (351100197)

Huthyfh Abdullah AlShahrani (342107075)

Under the Supervision of

Dr. Ahmed A. Mohamed

Department of Computer Science/Information Technology

College of Computer and Information Sciences

Majmaah University

Al Majmaah

Kingdom of Saudi Arabia

Spring 2018

CERTIFICATE

This is to certify that the project report entitled **Off-Line Handwritten Digits Recognition Using Machine Learning** being submitted by

Abdulrahman Yousef ALQunaibit (341101866)

Nemer Mahmoud Albalawi (341106053)

Raaed Dogaiem ALMohlsi (351100197)

Huthyfh Al-shahrani (342107075)

in partial fulfillment for the award of the Degree of Bachelor of Science in Computer Science/Information Technology to the Majmaah University, Al Majmaah, Kingdom of Saudi Arabia during the academic year 2017-2018 is a record of authentic work carried out under my supervision and guidance and it has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or another similar title to any candidate of any University.

Signature of Supervisor

Name: Dr. Ahmed A. Mohamed

Countersigned by

Head of the Department

Seal of the College

DECLARATION

We, Abdulrahman AL-Qunaibit, Nemer AL-Balawi, Raaed AL-Mohlsi and Huthyfh Al-Shahrani hereby declare that the project report entitled **Off-Line Handwritten Digits Recognition Using Machine Learning** submitted to Majmaah University, Al Majmaah, Kingdom of Saudi Arabia, in partial fulfillment for the award of the Degree of Bachelor of Science in Computer Science/Information Technology during the academic year 2017-2018 is a record of authentic work carried out under the supervision and guidance of Dr. Ahmed A. Mohamed and it has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or another similar title to any candidate of any University.

Signature of Student(s)

ACKNOWLEDGEMENT

The team would like to express their gratitude and appreciation to College of Computer and Information Sciences, Majmaah University for providing the permission and various hardware/software resources for completing the project successfully.

Special thanks to our honorable rector **Dr. Khalid Bin Saad Al Meqrin**, Majmaah University for his constant help and for providing us with all facilities needed for this project work.

Our sincere gratitude to **Dr. Mohammed Al Shehri**, Dean, College of Computer and Information Sciences, Majmaah University for supporting and encouraging the project work in every phase.

Sincere thanks to **Dr. Sultan Al Shehri, Vice Dean (Academic Affairs)** College of Computer and Information Sciences, Majmaah University for his continued support in completing this project successfully.

Our special thanks to our supervisor **Dr.** Ahmed A. Mohamed, College of Computer and Information Sciences, Majmaah University. His inspiring guidance and valuable advice and continuous encouragement at every stage in the progress of this project work enabled us to complete the project on time.

Table of Contents

Abstract

Handwriting digits are part of our daily life style. phone numbers dates ZIP Codes simple mathematics are examples of what we encounter every day.

Recognizing handwritten digits sometimes is a challenge for people who must deal with vast Number of customers from various cultures or ages daily.

Recognizing a handwritten ZIP Code written by an old man with Parkinson's disease on letter to his son my not be easy. A pharmacist my find it very tricky trying to recognize the exact dose of a prescription written by a doctor with shaky hands single mistake may cause someone's life in this case.

In this project, we are using artificial intelligent classifier namely K Nearest Neighbors to recognize handwritten digits in a scanned document.

Introduction

The artificial intelligence is the ability of a machine to do things that people say require intelligence.

The artificial intelligence contains two categories:

1- Supervised learning

   The supervised learning is where you have input data and output data and you use an algorithm to map from the input data to the output data. The goal of Supervised learning is when you have new input data you can predict the output from the data. The name supervised learning comes from the following process. the learning algorithm is trained on a dataset I can be thought of as a teacher supervising the process

2- Unsupervised learning

   The unsupervised learning is when you have only the input data without the suitable output data. The goal of unsupervised learning is to divide the import data based on structure or distribution of the data in order to learn from them. It was named unsupervised because there is no correct answer or teacher to guide the process.

- This project will use supervised learning where the software will have the dataset of handwritten digits to train on it and try to learn to recognize the digits.

Purpose and Scope of this Specification:

- The project main purpose is to recognize handwritten digits and convert it to digital form with the ability to display it.

- The project is made of several steps which are taking the image of the Handwritten digits as an input and enhancing the quality of the image then it will select the digits specifically based on multiple factors to run it through a classifier to determine the exact digit and display it.

- The project can not Recognize uncompleted digit also mixed digits will not be recognized at the same time it will not give you an error message even if the image does not have a digit let's say it has a character it will Match it to the closest number like it will read the letter "S" as a "5" or the letter "g" as "9".

- The project is helpful for anyone deals with digits like the post office to detect the exact address of litter or the bank's employees to minimize the error of reading a check.

Problem statement

The challenges are to do the processing that ensures all those millions of digits in any document. When it comes to processing the image, it is a hard task for a computer to do because we must "communicate" to them through relative devices such as keyboards and mice so, they can figure out what we want them to do.

we have chosen the handwritten recognition as a solution to these problems. But, to implement this task we also face problems that are:

a- HDR is a challenging problem since there is a variation of the same digit due to the change of fonts and sizes.

b- The differences in font types and sizes make the recognition task difficult and resulting the recognition of digit process become not accurate.

1. What is the Purpose and Need for the work?

The purpose of this project is to share the written information with others digitally, so it will be readable for everyone.

2. What are the Goals and Objectives of the work?

a- Saves the time for writers.
b- Increase the knowledge about K-Nearest Neighbors (KNN).
c- Saves the money.

3. What evaluation measures will be used?

The accuracy of converting the handwritten sheets to digital form.

4. What types of useable information and tools are available and practical?

A program that converts handwritten images to textual form.

A motivation for the project

Many reasons pushed us to choose this particular topic for our graduation project here are the top two:

- Students write and a lot of papers inside classes some of them wants to share the documents with each other in the process of sharing takes a lot of time to re-type again due to the different and handwriting between students the retyping make it readable by all the students.

- The Artificial intelligent is a new hot Field in the world right now and we find it very interesting, so we thought the best way to into this field is to try to build the project and after researching and reading we find out that one of the best ways to get into machine learning is K-Nearest Neighbors (KNN) which is one of the key element in artificial intelligent.

Expected outcomes

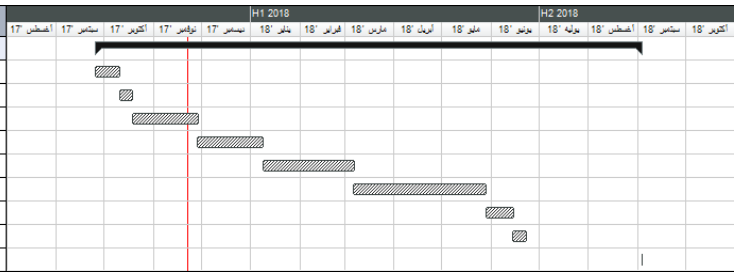
- Building a software from scratch
- Operate on Web application
- Operate on Desktop application
- Mobile friendly
- A software can recognize handwriting digits
- A software can convert handwriting digits to digital form

Identified tasks and a tentative work plan

- **Gantt chart**

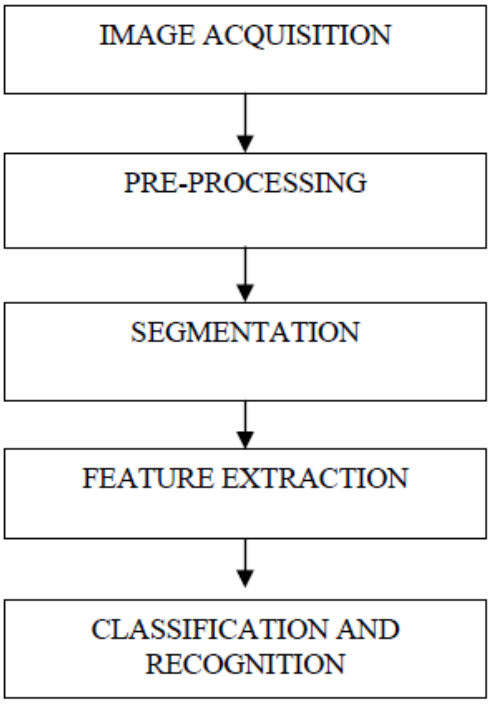
The below Gantt Chart describes the duration of our work and how much it will take from us to complete it. the longest time we will take on the designing our project (project planning, Requirements, Analysis and designing the interface)

	Task Name	Duration	Start	End
1	**Graduation Project**	**246 days**	**9/25/17**	**9/3/18**
2	Idea of the project	12 days	9/25/17	10/10/17
3	Project Planning	7 days	10/10/17	10/18/17
4	Gathering requirement	30 days	10/18/17	11/28/17
5	Analysis	30 days	11/28/17	1/8/18
6	Project design and Development	42 days	1/8/18	3/6/18
7	Implementation	60 days	3/6/18	5/28/18
8	Testing	14 days	5/28/18	6/14/18
9	Deployment	7 days	6/14/18	6/22/18
10	Maintenance	1 day	9/3/18	9/3/18



Background of the project

After we are discussing what the project we will choose, we prefer to choose **The Handwritten Digits Recognition Project**. In Our project, we will display how will acquisition the image and what the process of pre-processing phase. After that, when we finished the pre-processing phase will go to segment the image and this is the most important process. After the segmentation operation, we need the feature extraction to retrieve the important data of characters to help us for a good recognition accuracy. Finally, we will display how the classification and recognition operation by K-Nearest Neighbors (KNN).



Handwriting recognition is comprised of two techniques offline technique that is used to read a digits using an image capture device such as phone camera which means to convert Handwritten digit into a machine form (digital form) (Srihari, 2000)

IMAGE ACQUISITION: it is an image we import it from phone's camera and this image consist of handwritten digits and we will compare it with the image digits after the testing

IMAGE PRE-PROCESSING: first we find the size of the image that we will be using (for example 54 pixels – 34 pixels), there are three steps in the pre-processing operation …

The first step is reading the image into the compiler we will be using for example (visual studio environment).

The second step is converting the image into RGB colors (red, green, blue), note that the image will be stored in the form of an array of 0's and 1's.

The third step is cropping the image into the size we have chosen to remove the extra spaces.

Finally, we will resize the image to 54 by 34 pixels. This image we will be using for testing and before applying the pre-processing steps we will separate more than one digits into individual digits (Rozenberg, 2007)

Noise Reduction: when we scanned the image, and the image has some noise above it. This is will reduce the quality of the image and affect the next phase that is called Segmentation operation. As we know, the pre-processing is important to improve the quality of the image before moving to the next step. Based on noise, we can cut line segment, large gaps between the lines.

Noise reduction techniques can be categorized into two major groups:

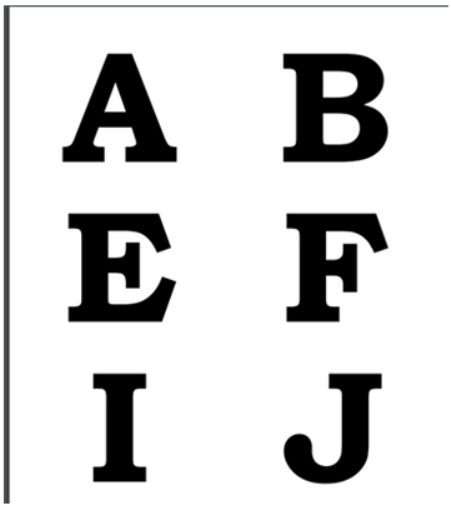
- **Filtering**: it used for reducing the noise and minimize the bogus points. Various spatial and
  frequency domain filters can be designed for this purpose.

- **Morphological Operations**: it is a tool used in image processing to appear image components. Also, it uses to remove the noise on the image to increase the quality of paper and ink, as well as erratic hand movement. (Rozenberg, 2007)

Binarization: it is an exclusive step that uses to facilitate segmentation and recognition of the digits. Binarization process converts a grayscale image into a binary image. (Rozenberg, 2007)

Edge Detection: it is a mechanism used to reduce the useless image details and catches the important details in the image scanned.

Before



After

Thresholding:

In order to reduce storage requirements and to increase processing speed, it is often desirable to represent grey scale or color images as binary images by picking some threshold value for everything above that value is set to 1 and everything below is set to 0.
(Rozenberg, 2007)

Skew Detection: usually there is a skewness in the document scanning process. There are some methods used for detecting skew in the page. The skew must remove because it reduces the accuracy of the image. (Rozenberg, 2007)

Slant Estimation and Normalization: Slant normalization is used to normalize all digits to a standard form. The most common method for slant estimation is the calculation of the average angle of near vertical elements. (Rozenberg, 2007)

SEGMENTATION:

Segmentation wherein each digit is separated from its neighbor digit, it is important because it makes the separation between the individual digit on the image.

There are two Types of SEGMENTATION:

- **External Segmentation**: is the isolation of different writing units such as paragraph and words, it is an important part of document analysis. The page layout analysis consists of two stages the first stage known as the structural analysis witch it segments the image into blocks of documents components such as paragraph, raw, word. The second stage is known as functional analysis that uses a location, size, and different layout roles to identify the functional content of the document components (Acton, 2007)

- **Internal Segmentation**: it is decomposing an image of a sequence of character into sub-images of individual characters symbols (Rozenberg, 2007)

FEATURE EXTRACTION:

It is a process to retrieve the most important data from the exact data have been used, in the feature extraction stage each digit is represented as a feature vector. it maximizes the recognition rate at least of an amount of element (Taxt, 1996)

the feature extraction methods are based on three types

- **Statistical**

They provide high speed (Extraction, Scanning), low complexity and take care of style variations.

The major of statistical features:

1. Zoning:

The character image is divided into NxM zones. From each zone, features are extracted to form the feature vector. The goal of zoning is to obtain local attributes instead of the global attributes. (Taxt, 1996)

2. Characteristics loci:

For every white point in the background of the digit, vertical and horizontal vectors are generated.

The number of times that the line segments intersected by these vectors are used as features.

With handwritten and printed numeral recognition based on an improved version of the loci characteristic method for appearing the numeral features after a preprocessing operation of the numeral image, the method divides the image into four equal parts and applies the traditional CL to each of the parts. On the constrained

samples, recognition rates exceeding 98 percent were achieved using the simpler algorithm. (Taxt, 1996)

3. Crossing and Distance:

Crossings calculate the number of transitions from background to foreground pixels according to vertical and horizontal lines through the character image.

Distances calculate the distances of the first image pixel detected from the upper and lower boundaries, of the image, along vertical lines and from the left and right boundaries along horizontal lines. (Taxt, 1996)

- **Structural Features**

It represents the characters with high tolerance to distortion and style variations. Structural Features are generated on topological and geometrical properties of the digit such as inflection between two points, cross points and horizontal curves at top or bottom.

- **Global transformations and moments**

It includes Fourier Transform, Gabor Transforms. Common transform and series expansion methods used in the CR field are:

Gabor Transform:

The Gabor transform is a type of short time Fourier transform. It is used to find the sinusoidal frequency. it is also used to find the phase content of local sections of a signal as it changes over time.

Fourier Transforms:

The general procedure is to choose magnitude spectrum of the measurement vector as the features in an n-dimensional Euclidean space. One of the most attractive properties of the Fourier transform is the ability to recognize the position-shifted digits.

CLASSIFICATION AND RECOGNITION:

The classification and recognition process are the decision-making part of a recognition system and it uses the components that take from the feature extraction. it uses the classification methods like K-Nearest Neighbors, Statistical, and kernel method.

There are four approaches to pattern recognition: Template Matching, Statistical Techniques, Structural Techniques and K-Nearest Neighbors (KNN).

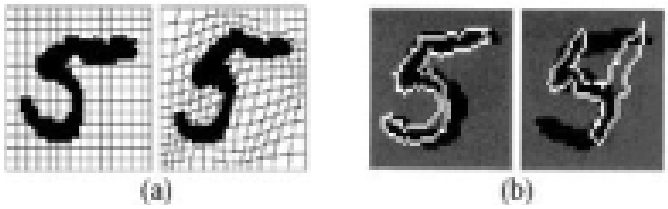
Template Matching

Template Matching is used by optical character recognition and it is a system where can use it to recognize the digits by comparing two images. Template Matching is a process to find the position of a sub-image that is called a template inside an image. also, the template matching must select the similarity between a template that used and windows of the same size in an image and selecting that extraction the highest similar measure

In Template Matching, there are two classes

- **Deformable Templates and Elastic Matching:**

Deformable templates are used in many object recognition applications. The alternative method is the use of Deformable templates when the image deformation is used to relate an unknown image against a database of known images.



(a) (b)

- **Direct Matching**

The direct matching method has a solid mathematical background and its recognition is very sensitive to noise.

Statistical methods

statistical can be classified based on the Bayes decision rule and it is divided into parametric recognition and non-parametric recognition:

a) Parametric Recognition

parametric recognition has information about the data. it can find a parametric model for each digit. the Consideration of the model is based on some probabilities and has the digits are categorizing according to a decision rule like Baye's method.

b) Non-parametric Recognition

A method that can classify non-parametric is the Nearest Neighbor and it is mostly used. An incoming pattern is categorized using the cluster that exists in the minimum distance. An incoming pattern does not contain information about the data.

K-Nearest Neighbors (KNN)

It is a classifier that works well on basic recognition problems, and it can be slow in prediction if there are large numbers of training examples and it is not robust to noisy data.

Advantage of K-Nearest Neighbors (KNN)

The main advantage, it is a very simple classifier that works well on basic recognition problems.

Disadvantage of K-Nearest Neighbors (KNN)

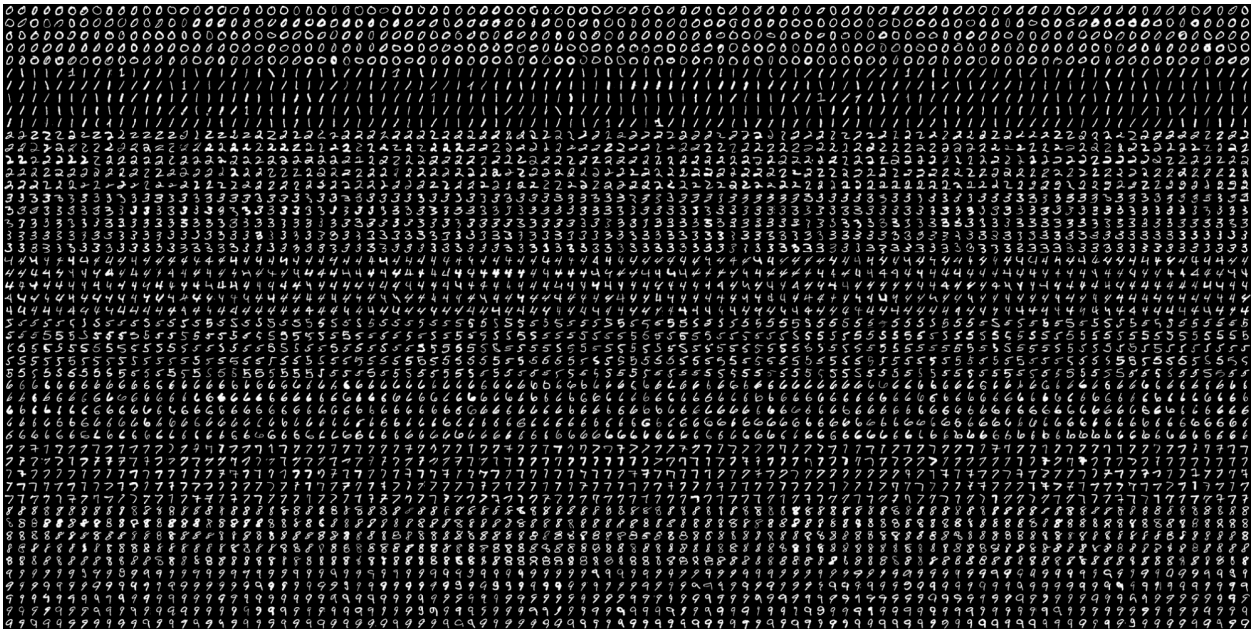
- The KNN algorithm does not learn anything from the training data, which can result in the algorithm not generalizing well and also not being robust to noisy data. Also, changing *K* can change the resulting predicted class label.

- The algorithm must compute the distance and sort all the training data at each prediction, which can be slow if there are large numbers of training examples.

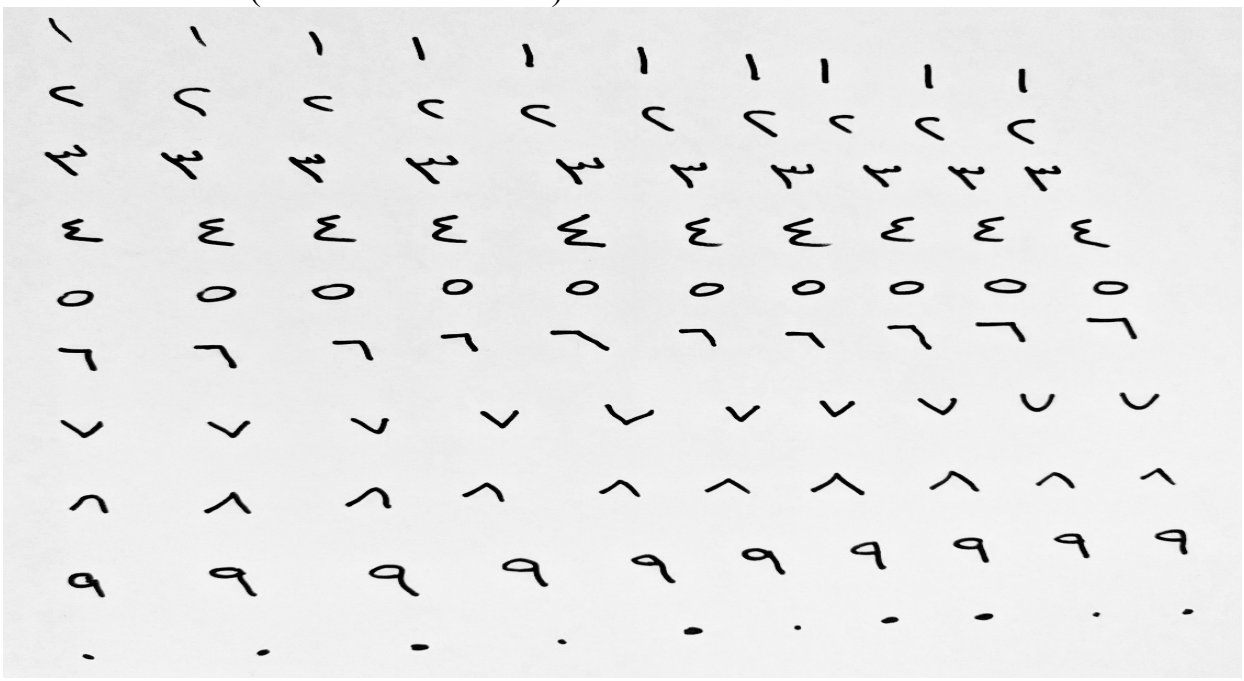
- The KNN algorithm is a *lazy learner*, that means it does not learn anything from the training data and simply uses the training data itself for classification.

Datasets for Recognition have been used:

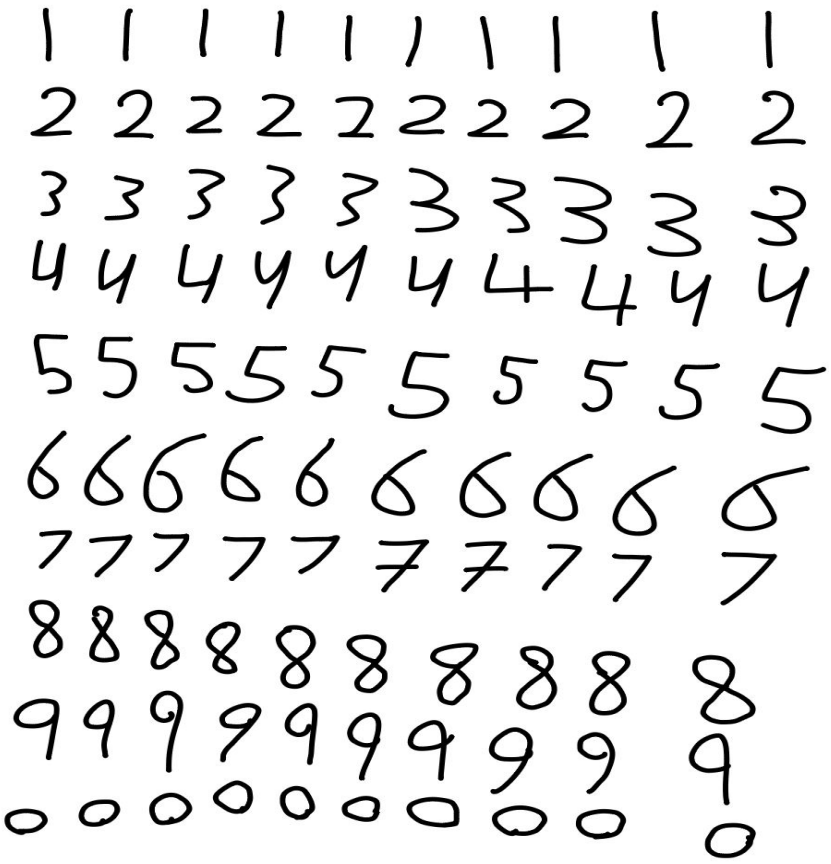
- This is **MNIST** handwritten digits database that have been used



- A dataset (Indian Handwritten) has been used

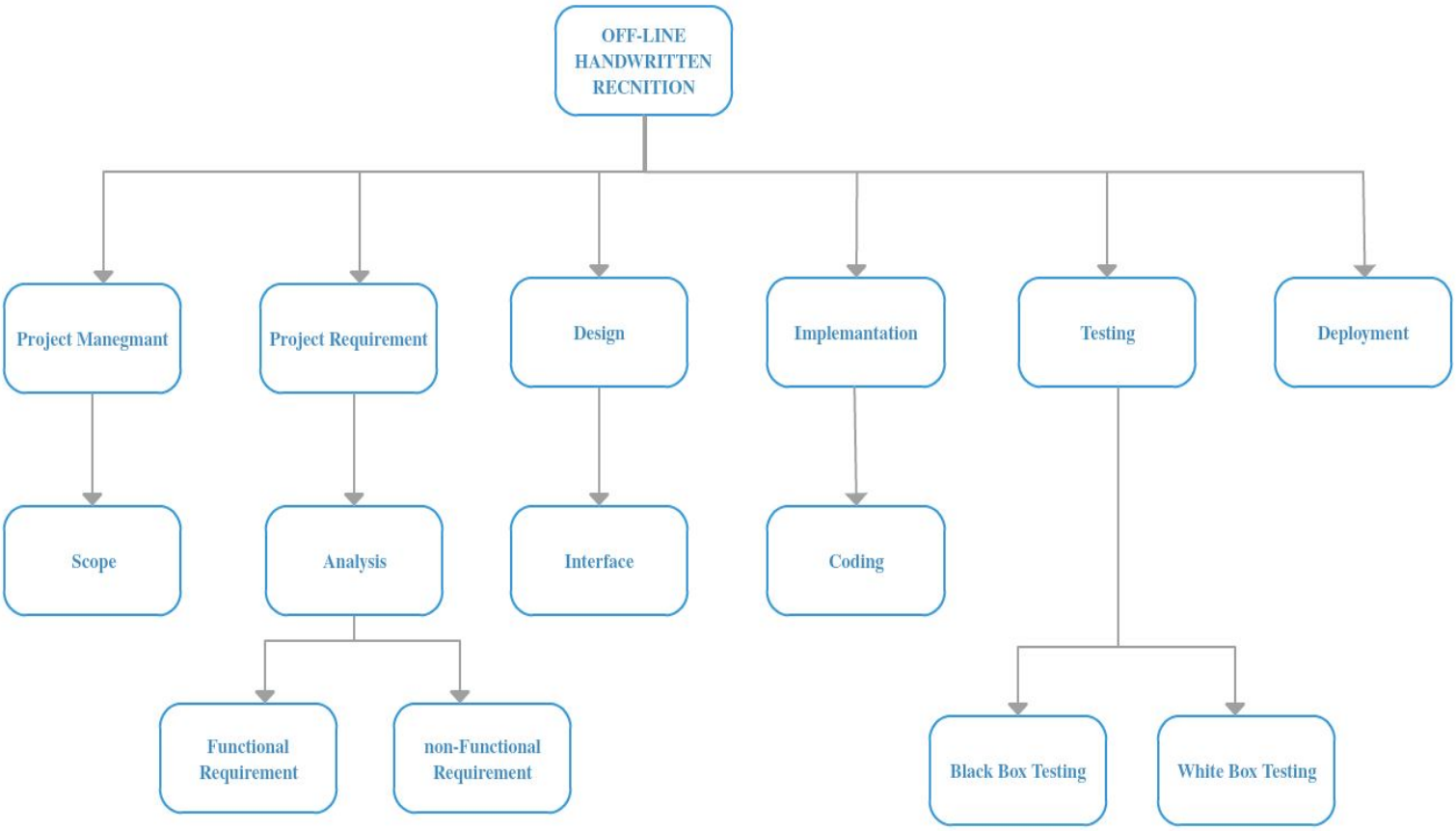


- Another Dataset (Arabic Handwritten) has done by us

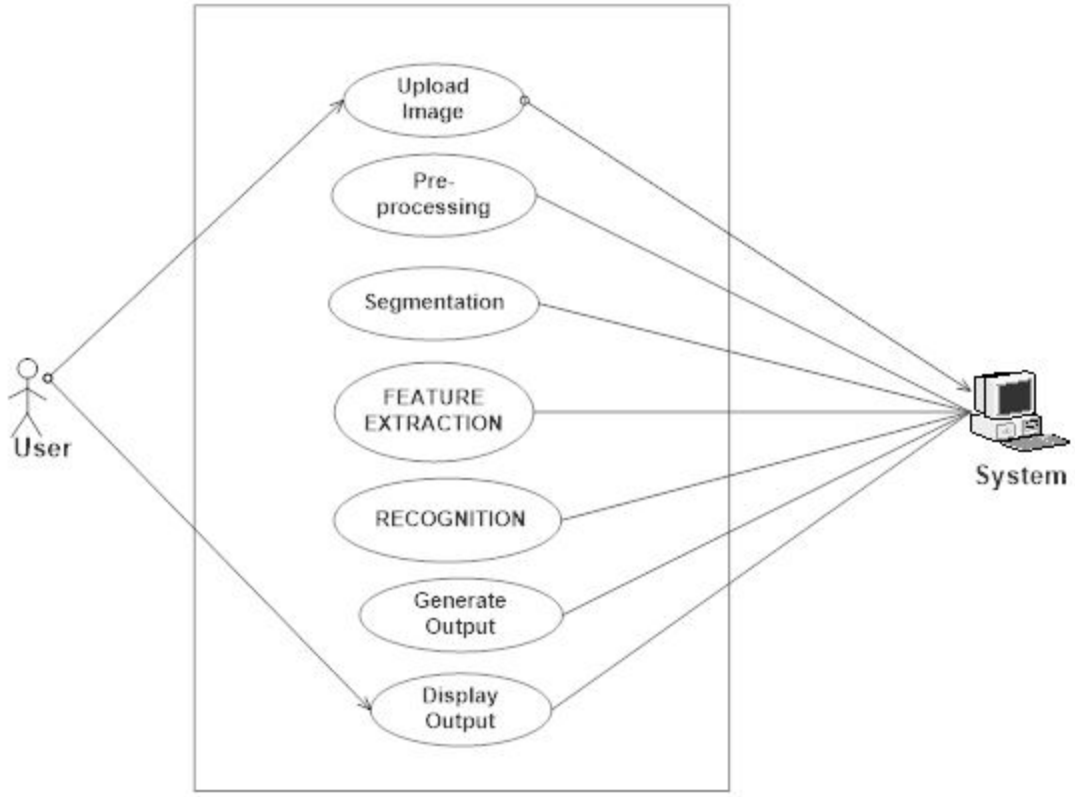


Project Architectural Design

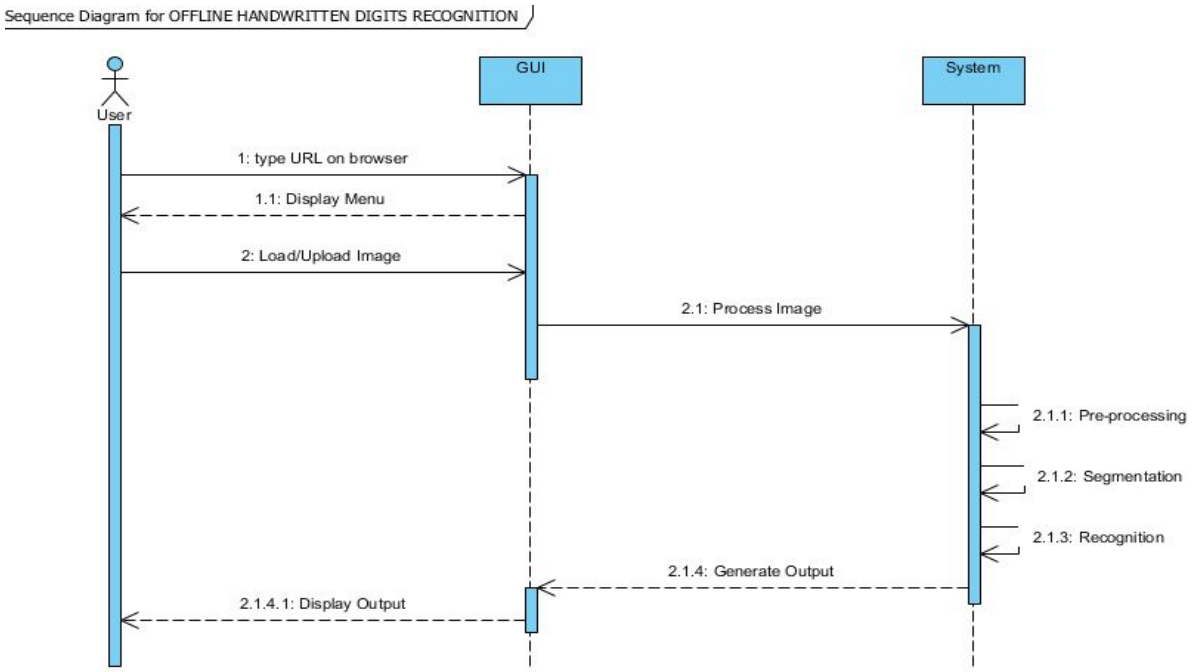
- Work Breakdown Structure



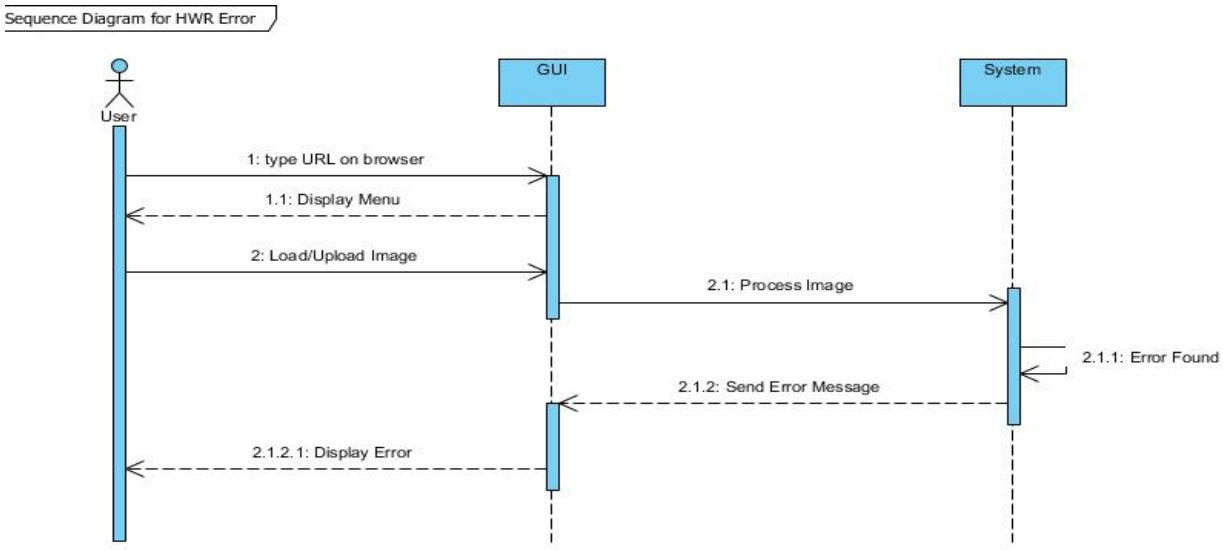
- Use Case Diagram



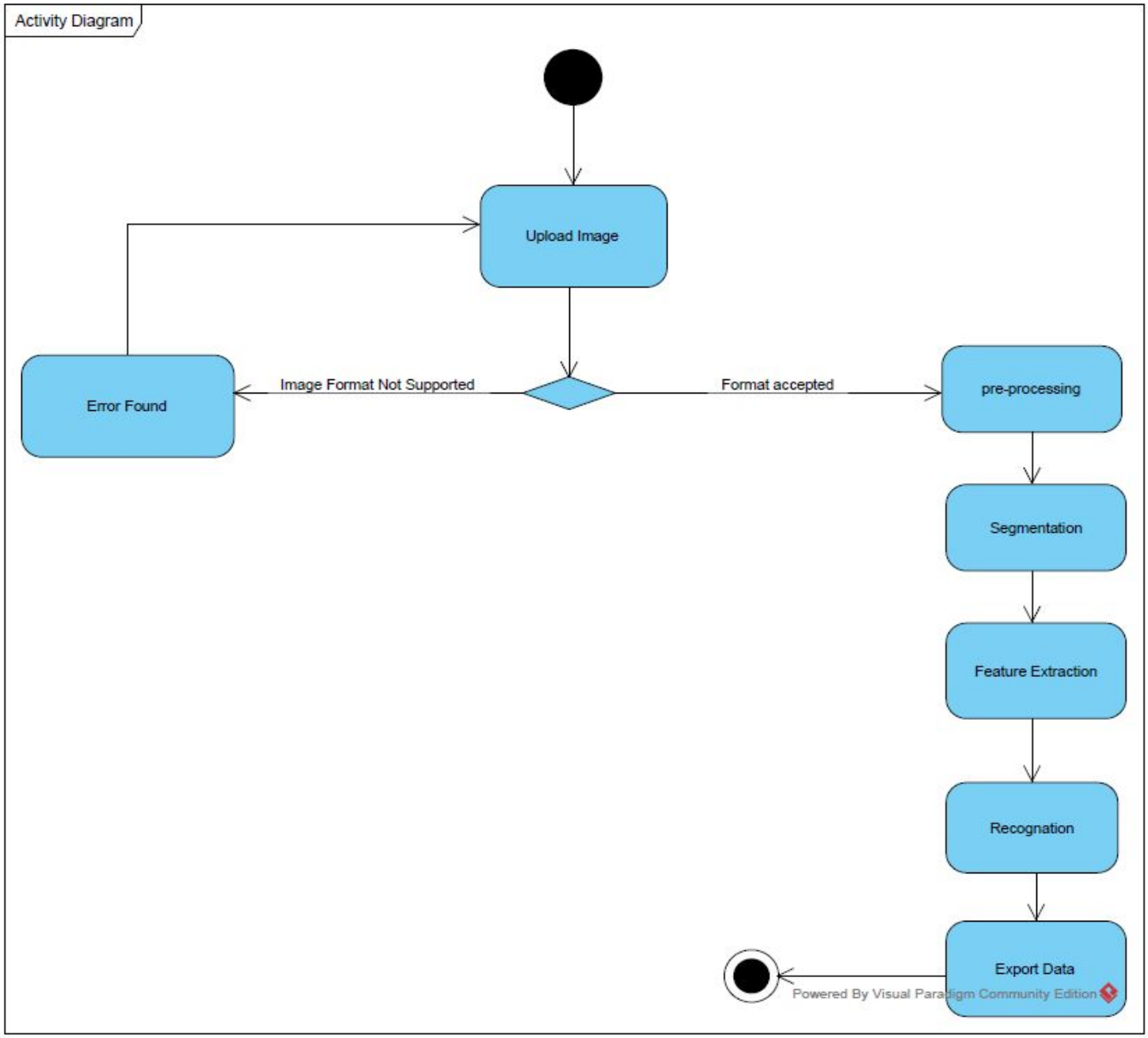
- Sequence Diagram



- Sequence Diagram for HWR Error



- Activity Diagram



Project Requirements

1. Project baseline requirements:

The software main point is to convert handwriting characters to optimal characters in order to do this it has to do the following:

1.1 The system shall be able to scan or capture an image.

1.2 The system shall be able to enhance the image quality that has been captured.

1.3 The system shall be able to dedicate the characters in the image.

1.4 The system shall be able to separate each character in the image.

1.5 The system shall be able to recognize each character in the image.

1.6 The system shall be able to rewrite the recognized characters.

1.7 The system shall be able to reform the text in the image into optimal writing.

1.8 The system shall be able to export the content of the image into PDF file.

Functional Requirement:

2.1 Acquisition

2.1.1 A user should be able to upload an image into the system

2.1.2 The font on the image shall bold

2.2 Pre-possessing

2.2.1 The system shall go through the preprocessing stages.

2.2.2 The system shall convert the imaging to RGB Colors.

2.2.3 The system shall store the image on a form of array 0's and 1's.

2.2.4 The System must save the original copy for the comparing after the pre-processing is done.

2.2.5 System shall filter the image by minimizing but points.

2.2.6 The system shall use morphological operation to improve the image and the quality of paper and ink.

2.2.7 The system shall convert grey scale image into a binary image.

2.2.8 The system shall reduce useless Image details and catches the important details by using edge detection.

2.2.9 The System shall detect skew and remove the skew in the page by using skew Direction.

2.2.10 The system shall use slant normalization to normalize all character's standard form.

2.3 Segmentation

2.3.1 The system shall separate individual characters of an image

2.4 Feature extraction

2.4.1 The system shall present each character as a feature vector.

2.4.2 The System shall generate for every white point in the background of a character vertical and horizontal vectors.

2.4.3 The system shall divide the image into four equal parts and applies the traditional CL to each part.

2.4.4 The Systems will calculate the number of transitions from background to foreground pixels according to a vertical and horizontal line.

2.4.5 The system shall calculate the distance of the first image picture detected from the upper and lower boundaries.

2.4.6 The system shall find the sinusoidal frequency and. Find the face content of a logical section of a single as it changes over time.

2.4.7 The system shall be able to recognize the position shifted characters.

2.5 Classification and recognition

2.5.1 The system shall use template matching to Recognize digits and characters.

2.5.2 The system shall use deformable templates to relate unrecognized images from a database of no images.

2.5.3 The system shall use statistical message, so it can classify on the bases decision rule and it's divided into parametric recognition and non-parametric recognition.

2.5.4 The system shall implement an K-Nearest Neighbors (KNN) to implement classification and recognition.

2.6 Interface Requirements

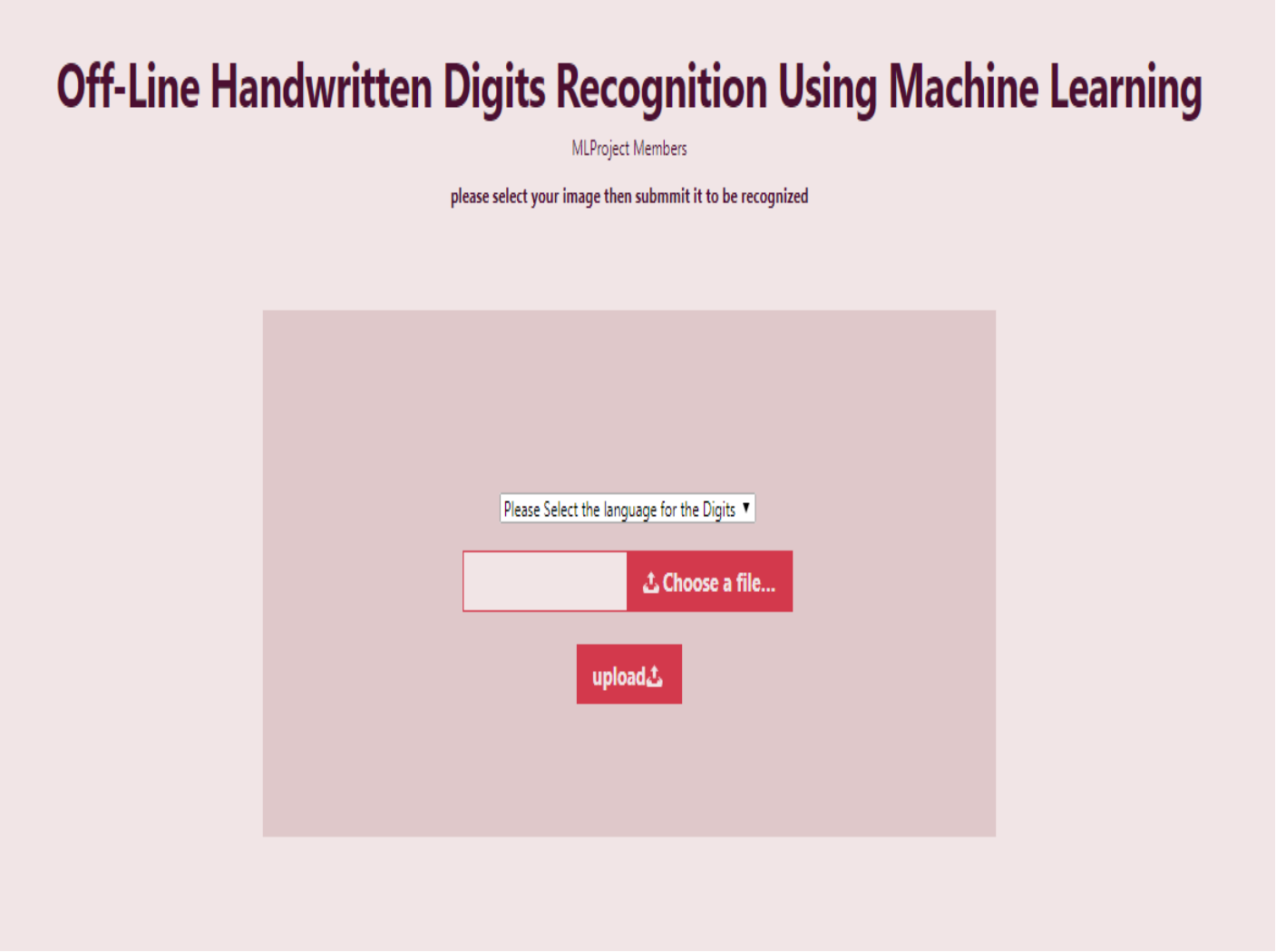
2.6.1 It will describe the user interfaces that are to be implemented by the system.

2.7 Hardware Interfaces

2.7.1 The system will support hardware interfaces such as a console machine that can run the program and a scanner to read the image.

2.8 Software Interfaces

2.8.1 Our handwritten recognition program needs a simple interface, so it can be ease of use for all the types of people such as expert users, normal users, and children.



Non-Functional Requirements

3.1 Performance

3.1.1 Performance is one of major issues for our system. The system should have a high performance such that the user should be able to use the system without any disturb such as hang

3.2 Accuracy

3.2.1 Accuracy of the system is important for the user. The system should have higher accuracy, meaning that it should recognize all digits correctly.

3.3 Simplicity

3.3.1 Relies on user interface, an important design goal is keeping it simple. User interface should not be so complicated for users.

3.4 Availability

3.4.1 System will be available as long as our server is running.

3.5 Maintainability

3.5.1 Our system is specified for one task. There will be more tasks in the future.

3.6 Usability

3.6.1 The system should be easy to use by users.

4. Operational Requirements

We need operational requirements to run the system and make some actions such as scanning, importing and exporting. Also, the system needs to communicate with operations personnel.

5. Security and Privacy

Our system needs security and privacy to increase the reliability and attracts the users and it will reject disclosure of sensitive images and prevent malware inside the images. Any image must be decrypted so it can be hard to access it from any anonymous users

6. Reliability

The system must be available 24/7 and provide a backup system when it fails or goes down, and it must have repaired fast, and it must provide high performance

7. **General Performance**

Our system provides a fast response time and maintains any error or failure in the system, Also, it can scan several of images concurrently

8. **Capacity**

It can accomplish a high amount of storage for the users.

9. **Error Handling**

The system can avoid error mistakes by the user such as uploading double images or half image

10.Software Requirement:

10.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

10.2 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

10.3 Visual Studio

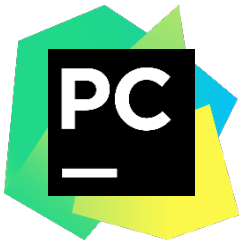
Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge.

10.4 pycharm



PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition released under a proprietary license - this has extra features.

Features

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: including rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with changelists and merge

10.5 Amazon Web Services

Amazon Web Services (AWS) is a comprehensive, evolving cloud computing platform provided by Amazon. It provides a mix of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

AWS launched in 2006 from the internal infrastructure that Amazon.com built to handle its online retail operations. AWS was one of the first companies to introduce a pay-as-you-go cloud computing model that scales to provide users with compute, storage or throughput as needed.

Amazon Web Services provides services from dozens of data centers spread across availability zones (AZs) in regions across the world. An AZ represents a location that typically contains multiple physical data centers, while a region is a collection of AZs in geographic proximity connected by low-latency network links. An AWS customer can spin up virtual machines (VMs) and replicate data in different AZs to achieve a highly reliable infrastructure that is resistant to failures of individual servers or an entire data center.

More than 100 services comprise the Amazon Web Services portfolio, including those for compute, databases, infrastructure management, application development and security.

Code overview

In the implementation phase we have decided to use python language and here some of the implementations:

```
import numpy as np
import cv2
```

These are the libraries that we used.

NumPy: is the fundamental package for scientific computing with Python

CV2: is OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.

KNN: it is the library that helps us to train and test the dataset

```
image = cv2.imread('images/digits.png')
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
small = image # to center the image
cv2.imshow('Digits Image', small) # display the image

# Split the image to 5000 cells, each 20x20 size
# This gives us a 4-dim array: 50 x 100 x 20 x 20
cells = [np.hsplit(row,100) for row in np.vsplit(gray,50)] #

# Convert the List data type to Numpy Array of shape (50,100,20,20)
x = np.array(cells)
```

In this block of code, we take the dataset which is in form of an image and we split it into cells and store it into an array

```
# Split the full data set into two segments
# One will be used fro Training the model, the other as a test data set
train = x[:,:90].reshape(-1,400).astype(np.float32) # Size = (3500,400)
test = x[:,90:100].reshape(-1,400).astype(np.float32) # Size = (1500,400)

# Create labels for train and test data
k = [0,1,2,3,4,5,6,7,8,9]
train_labels = np.repeat(k,450)[:,np.newaxis]
test_labels = np.repeat(k,50)[:,np.newaxis]

# Initiate kNN, train the data, then test it with test data for k=3
knn = cv2.KNearest()
knn.train(train, train_labels)
ret, result, neighbors, distance = knn.find_nearest(test, k=3)

# Now we check the accuracy of classification
# For that, compare the result with test_labels and check which are wrong
matches = result == test_labels
correct = np.count_nonzero(matches)
accuracy = correct * (100.0 / result.size)
```

In here the software will split the dataset into two segments one for training and another one testing then it will create labels for each number and assigned it to the correct sells.

Then it will initiate the KNN algorithm to train the data and test it for number 3, then compare the result with the labels to check and display the accuracy

```
#Acquisition
image = cv2.imread('uploaded/testimage.jpeg')
cv2.imshow('test image', image)
#cv2.waitKey(0)
```

here we apply the Acquisition phase and give the user the option to upload an image of his/her choice

```
#Preprocessing : binarization
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
cv2.imshow('gray test image', gray)
#cv2.waitKey(0)

#Preprocessing : Filtering
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
cv2.imshow("blurred", blurred)
#cv2.waitKey(0)

#Preprocessing : Edge Detection
edgeDetection = cv2.Canny(blurred, 30, 150)
cv2.imshow("edged", edgeDetection)
#cv2.waitKey(0)
```

In this block of code, we apply the pre-processing phase which includes the Binarization where the image will convert it into two colors black and white. Also, include Filtering which effects the image by blurring it. Finally, the edge detection

```
def x_cord_contour(contour):
    # This function take a contour from findContours
    # it then outputs the x centroid coordinates

    if cv2.contourArea(contour) > 10:
        M = cv2.moments(contour)
        return (int(M['m10'] / M['m00']))
```

This function's job is to determine the area of the digit

```
def makeSquare(not_square):
    # This function takes an image and makes the dimenions square
    # It adds black pixels as the padding where needed

    BLACK = [0, 0, 0]
    img_dim = not_square.shape
    height = img_dim[0]
    width = img_dim[1]
    if (height == width):
        square = not_square
        return square
    else:
        doublesize = cv2.resize(not_square, (2 * width, 2 * height), interpolation=cv2.INTER_CUBIC)
        height = height * 2
        width = width * 2
        if (height > width):
            pad = (height - width) / 2
            doublesize_square = cv2.copyMakeBorder(doublesize, 0, 0, pad, pad, cv2.BORDER_CONSTANT, value=BLACK)
        else:
            pad = (width - height) / 2
            # print("Padding = ", pad)
            doublesize_square = cv2.copyMakeBorder(doublesize, pad, pad, 0, 0, cv2.BORDER_CONSTANT, value=BLACK)
    doublesize_square_dim = doublesize_square.shape
    return doublesize_square
```

The makeSquare function will make the borders around the digit

```
def resize_to_pixel(dimensions, image):


    buffer_pix = 4
    dimensions = dimensions - buffer_pix
    squared = image
    r = float(dimensions) / squared.shape[1]
    dim = (dimensions, int(squared.shape[0] * r))
    resized = cv2.resize(image, dim, interpolation=cv2.INTER_AREA)
    img_dim2 = resized.shape
    height_r = img_dim2[0]
    width_r = img_dim2[1]
    BLACK = [0, 0, 0]
    if (height_r > width_r):
        resized = cv2.copyMakeBorder(resized, 0, 0, 0, 1, cv2.BORDER_CONSTANT, value=BLACK)
    if (height_r < width_r):
        resized = cv2.copyMakeBorder(resized, 1, 0, 0, 0, cv2.BORDER_CONSTANT, value=BLACK)
    p = 2
    ReSizedImg = cv2.copyMakeBorder(resized, p, p, p, p, cv2.BORDER_CONSTANT, value=BLACK)
    img_dim = ReSizedImg.shape
    height = img_dim[0]
    width = img_dim[1]
    return ReSizedImg
```

This function then re-sizes an image to the specified dimensions

```
contours, _ = cv2.findContours(edgeDetection.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

This particle line is what the pass three functions deepened on

```
for c in contours:

    (x, y, w, h) = cv2.boundingRect(c)

    if w >= 5 and h >= 25:
        roi = blurred[y:y + h, x:x + w]
        ret, roi = cv2.threshold(roi, 127, 255, cv2.THRESH_BINARY_INV)
        squared = makeSquare(roi)
        final = resize_to_pixel(20, squared)
        cv2.imshow("final", final)
        final_array = final.reshape((1, 400))
        final_array = final_array.astype(np.float32)
        ret, result, neighbours, dist = knn.find_nearest(final_array, k=1)
        number = str(int(float(result[0])))
        full_number.append(number)


        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
        cv2.putText(image, number, (x, y + 155),
                    cv2.FONT_HERSHEY_COMPLEX, 2, (255, 0, 0), 2)
        cv2.imshow("image", image)
```

This is a loop over the contours it will draw a rectangle around the digit, and show what the digit was classified as

```
cv2.destroyAllWindows()
file = open('output.text', 'w')
file.write(" " + ''.join(full_number))
file.close()
cv2.imwrite("outputs/testimage.jpeg", image)
```

Here where it will give the user the output of the process which is the written digits and display the segmented image

Outputs

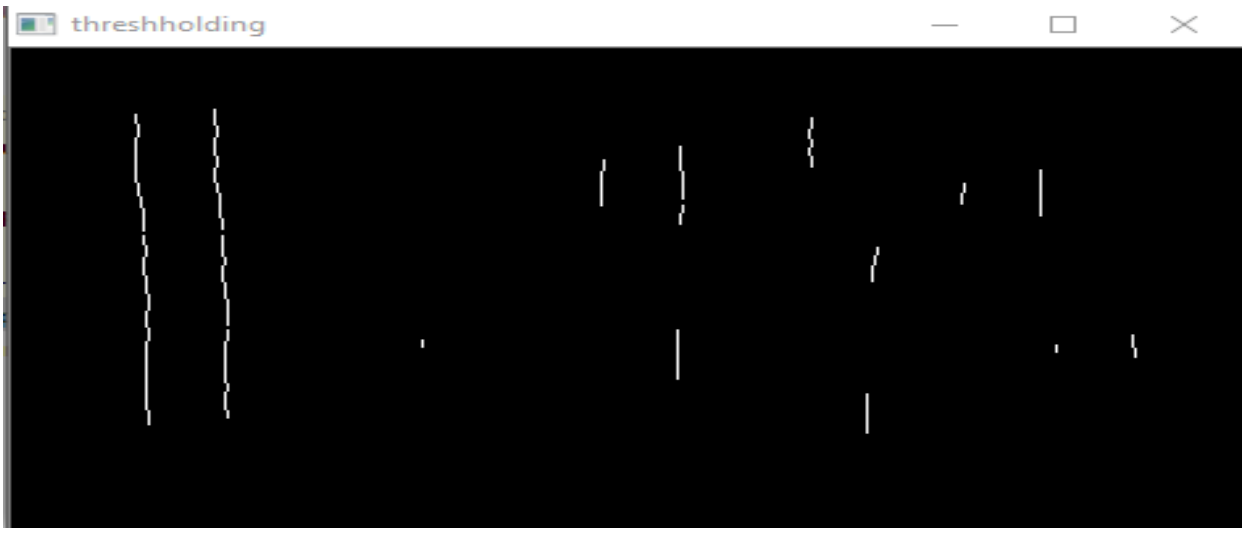
- Original

- Binarization

- Noise Reduction

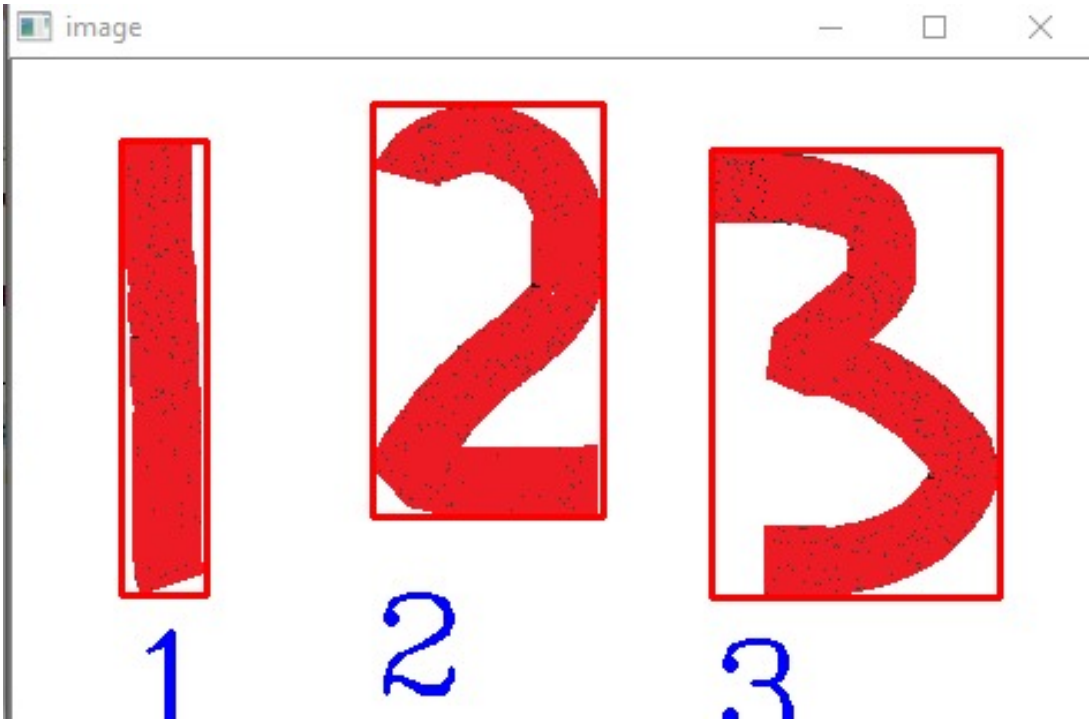
- Edge Detection

- Thresholding



- SEGMENTATION

- Final result



Future Plan

First version always sat far away from perfection, we mean by the perfection word here is that the form which the software has the ability to meet the user needs in real world because we know that "Users are complex" and one of the software engineering principles we should always keep in mind is "The requirements will change". Our project lacking some features we didn't include them - for now –. The current form of this project able only to recognize digits Our plan is to make it better.

Here, is a list of features we think it can achieve the goal:

- Better interface
- Character recognition in English
- Character recognition in Arabic
- symbols recognition (ex: math symbols)
- Signature recognition

conclusion

The name of this project is self-explanatory it will make it easy to recognize numbers in two major languages Arabic and English. It is a solution to any one deals with handwritten digits. In the team eyes it is not complicated until it covers the future plan and Beyond. it's a shy step for now it will need more effort more collaboration more enhancements to make it ready to deal with any documents and notebooks.

References

(n.d.). Retrieved from openCV.org: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html

(n.d.). Retrieved from OpenCV.org: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

(n.d.). Retrieved from OpenCV.org: https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html

Acton, B. L. (2007). Active Contour External Force Using Vector Field Convolution for Image Segmentation. *IEEE Transactions on Image Processing*, 2096-2106.

Rozenberg, W. B. (2007). Image Preprocessing for Improving OCR Accuracy. *2007 International Conference on Perspective Technologies and Methods in MEMS Design*, 75-80.

Schmidhuber, r. (2015). Deep Learning in Neural Networks: An Overview. *Neural networks: the official journal of the International Neural Network Society*, 85-117.

Srihari, j. P. (2000). On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 63-84.

Taxt, O. D. (1996). Feature extraction methods for character Recognition-A survey. *Pattern Recognition*, 641-662.