A Project Report

on

Handwriting Recognition System Using Machine Learning

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. G.Nagarajan

Submitted By

Abhishek Varshney
19021011994/19SCSE1010864
Shivam Srivastava
19SCSE1010206

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING /
DEPARTMENT OF COMPUTER APPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

DECEMBER 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"HANDWRITING RECOGNITION SYSTEM USING MACHINE LEARNING"** in partial fulfillment of the requirements for the award of the Name of Degree submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Mr. G.Nagarajan Guide, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Abhishek Varshney(19SCSE1010864)

Shivam Srivastava(19SCSE1010206)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. G.Nagarajan

CERTIFICATE

The Final Project Dissertation Viva-Voce examination of Abhishek Varshney(19SCSE1010864) and Shivam Srivastava(19SCSE1010864) has been held on _________________ and his/her work is recommended for the award of Name oF Degree.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: Dcecmber, 2021

Place: Greater Noida

Abstract

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.

Table of Contents

List of Table

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1

Introduction

1.1. Academic Question

a. How will the system work and what techniques, tools and technologies will it use to extract text from images?

b. Will the users need to log in for using this system?

c. What is the accuracy of your system? Will your system performance decrease if the image is provided with enough noise?

d. How will the users get benefited by this system?

1.2. Aims & Objectives

1.2.1. Aims

- To train the model on synthetic dataset.
- To extract handwritten word not just from paper or electronic documents but also from natural scene images.
- To help the people to store data digitally without copying from the documents

1.2.2. Objectives

- To research on the Internet, Books, Journals, Articles etc.

- To implement suitable classifiers and algorithm.
- To build the platform, web application for the people to perform Optical Character Recognition (OCR).

1.3. Brief Details of the Artifact Produced and Background to the project.

Data has become the most valuable assets in the world. People are storing data in both electronic and paper-based format. They need stored data in their daily lives to run their businesses. Rewriting those stored data is time consuming and unproductive. Traditionally, text recognition has been done on document images because of their well suited digitise planner paper based formats. But when it comes to natural scene images, the accuracy decreases drastically because of their highly variance in appearance and layout in the images. Additionally, natural images are suffered from noises, inconsistent light, occlusions, orientation etc which makes difficult for the classifier to detect and recognize the text in comparison to document images. In the recent years, the advancement came in the field of computer vision techniques and the large volume of datasets produced over the last decades has made possible to recognize the text form even natural scene images. In this project text spotting is done from natural images by implementing two techniques i.e. word detection followed by word recognition. This project does not perform character recognition instead it recognizes word through word spotting mechanism. The detector is built with Tesseract and OpenCV and recognition is done by Convolutional Neural Network (CNN). CNN is trained on synthetic datasets known as VGG synthetic word datasets. This project is based on flask web application where the users perform OCR by uploading images in the system.

Artefact (proposed) to be developed

Artefact 1

Image upload

Artefact 2

Word Detection

Artefact 3

Word Recognition

1.4. Potential Users

There are no specific users required to use this system. Everyone can utilize this system to perform OCR. Today, the corporates around the world upgraded to digital format. For instance, they store the corporate data, information etc. in electronic from. Moreover, the people from every field are recognizing the importance of OCR because they do not have to go through the hassle of copying the whole words from the hard documents. Since its development, it has been applied to many fields and still widening its horizon. Some of the fields of OCR are Handwriting recognition, Receipt Imaging, Legal Industry, Banking, HealthCare, Captcha, Automatic Number Plate Recognition, ATMA: android travel mate application etc. It seems everybody needed such systems in today's world where the data has become the valuable assets. So, application of OCR cannot be restricted to just some fields and some users.

1.5. Scope and Limitations of the project.

Text, being consider as the only tools for preserving and communicating information. Today's modern world is designed to interpret and communicate using text clues, labels, texts etc. found in the surroundings. So, text has been scattered through many images and videos for the communication purposes. Extracting such texts from the images and storing the information in digital format helps to secure from the damages done by the theft of hard documents. Sometimes we need to digitally replicate the text of the images. In such cases OCR can play an important role.

System is based on word recognition method instead of character recognition. Unlike the character recognition, which recognize the word by the recognition of letters, word recognition has to trained with the whole word as input.

So, the recognition of such model is constraint to the number of words in the dictionary because in such method we can cover all the words for recognition. Similarly, the accuracy of this method is low because the model is trained with small no of datasets. The reason behind small number of datasets is because of computational limitations. The other limitation of this system are it does not work offline, only recognize the English alphabetical words.

1.6. Report Structure

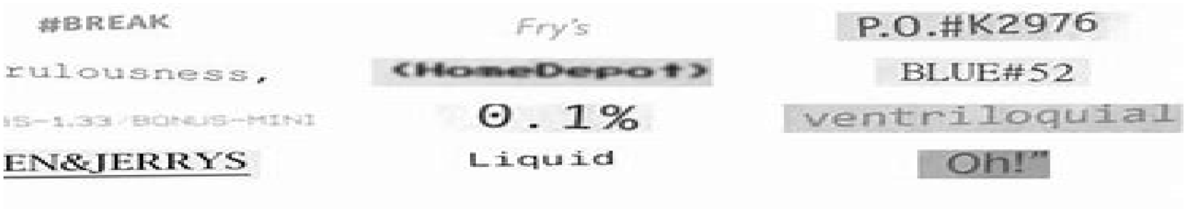
- Introduction: It provides the overall introduction of the project. It includes topic such as project aims, objectives, scope, limitations, academic question, and artifact.
- Literature Review: It includes the necessary information for the completion of the project such as background research, components, and similar system.
- Development: This section provides the information from project planning to its development. It includes all the planning's, designs, and testing.

- Answering Academic Question: This section provides the answers regarding the academic questions.
- Conclusion: It concludes the whole project with its future escalation.
- Critical Evaluation: It includes all the necessary evaluation towards the report, systems, and development process.

CHAPTER-2 Literature Review

2.1. Component

A. Data Exploration



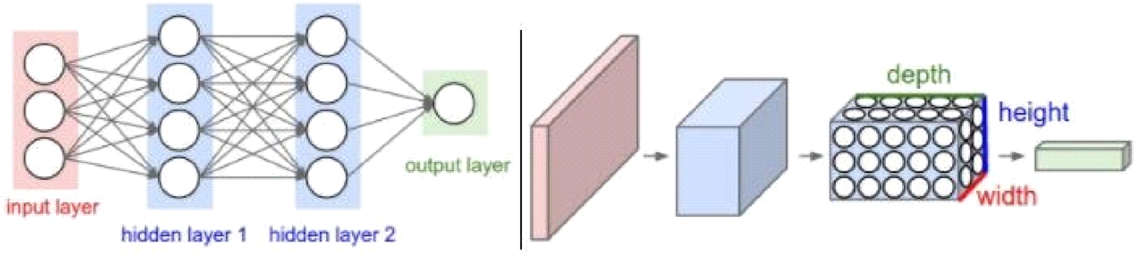
I use the synthetic dataset for this project. The dataset is imported from the Visual Geometry Group (VGG) of University of Oxford. The dataset has 9 million images of 90k word dictionary. The 90k dictionary consists of English words from the Hunspell, a popular open source spell checking system.

VGG generate 9 million 32×100 images synthetic datasets of 90k words provided by the Hunspell. We need dataset to train our model to recognize words from the images.

I trained my model using only some percentage of the total VGG word dataset due to computational limitations. So, my model could only train a 1000-word classifier using a fraction of total images i.e. one lakh images.

While choosing the 1000 words from 90k words, I took a reference form the google which had listed 1000 most common words.

B. Convolution Neural Network (CNN)



Convolutional Neural Network is a feed forward Artificial Neural Network (ANN) of which its mechanism is inspired by the neuron connectivity pattern of animal visual cortex. The CNN takes the input as multi-channeled image whereas in general, neural networks takes input in the form of vector. CNN does minimum preprocessing in comparison to the other algorithm. CNN consists of neurons with their learnable weights and biases. Each neuron takes input, perform some dot products which is followed by non -linearity. CNN consists of mainly three layers i.e. input layer, hidden layers, and output layer. The hidden layers include convolutional layers, ReLU layer i.e. activation function, pooling layers, fully connected layers, and normalization layers

a. Input Layer: It contains the actual raw pixel values of image.

b. Convolutional Layer: It is applied on the input image using a convolutional filter to extract a feature map. Filter is slide over the image which is known as stride.

c. ReLU Layer: It applies the activation function on the image data. The output is {F(x) = max(0,x)}

d. Pooling Layer: It performs a down-sampling operation reducing the height and width and depth intact. It reduces the number of parameters which shortens the training time and reduces the probability of overfitting.

e. Fully Connected Layer: It is the layer which classifies the results based on the score classes. The class with maximum scores will correspond to the input letters.

(Anuj Dutt, 2017) (Meer Zohra, 2019)

There are many standard CNN architectures available which all have common 4 layers i.e. Convolution Layer, Pooling Layer, ReLU Layer and Fully Connected Layer. For my model, I will use these all layers, but the number of each layer will depend on the types of data. The use of filters for convolution layer, max pooling layer will depend on the features that I want to extract. I will also use the dropout and activation function at each layer to optimize the model and reduces the chances of overfitting.

The CNN model proposed for this system consists of total eight weight layers. i.e. five convolutional layers with three fully connected layers. Each convolutional have both same and different parameters form each other. The convolutional layers have the parameters in following structure: (filter size, numbers of filters): (5,64), (5,128), (3,256), (3, 512), (3, 512). The batch normalization, activation and max pooling layers is added after each convolution layer. All convolutional layers have same batch normalization and ReLU as activation function for speeding the training the network with stability and solving the non-linearity problems, respectively. Similarly, pooling layer has also applied after each convolutional layer with pool -size (2,2) and strides of 2. It helps to reduces the computational cost by reducing the number of parameters to learn and provide basics translation invariance to the internal representation. The flatten layer has also included after fifth convolution layer to convert the data into a 1-dimesnional array so that a vector can be fed into a fully connected network classifier. The first two fully connected layers has ReLU as an activation function and 400 units or class whereas the final fully connected layer has the same units as in number of words in the dictionary i.e. 1k words. Dropout layer of 0.5 is added after each dense layer except final layer to prevent neural network from overfitting. The final classification layer has SoftMax normalization layer for converting its output into a probability distribution which finally checks with the words in the dictionary.

C. OpenCV

OpenCV is an opensource computer vision and machine learning software library developed in 1999 by Gary Bradsky at Intel and the first release came out in 2000. The aim of OpenCV was to fulfill the gap in the field of computer vision and machine learning by providing the infrastructures. OpenCV supports many algorithms and it has more than 2500 optimized algorithms including both classic and start-of-the art computer vision and machine learning algorithms. These algorithms have a wide range of applications in many fields such as identify objects, track moving objects, extract 3D models, classify human action videos etc. It has a larger supports community, more than 47000. It supports various languages such as C++, python, Java etc. and platforms such as Windows, Linux, Android etc.

While the OpenCV can processed image from preprocessing to recognizing the word from images, I had only applied this technique for word segmentation in my proposed system. There are mainly three types of common thresholding in the OpenCV i.e. Simple Thresholding: it segments the images with comparing the pixel value with a threshold value, Adaptive thresholding: the threshold is measured for each region of the image instead of using a global threshold value like Simple Thresholding. Lastly, the Otsu's Binarization which I had used for my proposed system.In this thresholding, it solves the problems of the bimodal image, where it occurs multiple times in my image datasets.

D. Tesseract

As there are many OCR engine/cloud service providers such as ABBYY cloud OCR SKD, Bing OCR etc. but most of them are commercial and closed source. Moreover, it is built by HP labs and extensively managed by Google at present. Tesseract in itself a complete classifier but I choose for detection purposes only. (Jin, 2014)

The working flow of Tesseract OCR engine are given below.

1. Thresholding:

It converts the image into a binary format and filters the noise on the images. There are two methods for thresholding images. They are

a. Adaptive Threshold: It is used for small regions of the images. It is used to account for variation within the images. (Kuguoglu, 2018)

b. Otsu’s Threshold: It is particularly used for bimodal images whose histogram has two peaks. It is also used for the small sized rectangular divisions of the images. (Kuguoglu, 2018)

2. Page Layout Analysis

Component outlines are gathered into blobs. Blobs are the region on the digital images that are detected to be from the surrounding region in color or brightness. By inspecting these outlines, it is easier to detect inverse text and recognize as black-white text. (Thunstrom, 2016)

3. Line and Word Finding

Blobs which are organized into text lines are broken into words. Lines and region are analyzed for pitch or proportional text. Character cells chopped the fixed pitch text immediately whereas the Proportional Text is broken into words using definite and fuzzy spaces. (Thunstrom, 2016) (Jin, 2014)

4. Recognize Word Pass 1

An attempt is made to recognize each word in turn in the first pass. Each word which have been recognize is transfer to an adaptive classifier as training helps in recognize the lower text down the page. (Thunstrom, 2016)

5. Recognize Word Pass 2

Because the first pass is more focus on the text down the page, a second pass is run over the page to gather missing words near the top of the page. (Thunstrom, 2016)

2.2. Similar System

From the very past, a lot of research and implementation has been done in the field of Computer Vision. Handwritten Word Recognition is one of the hot research applications in the computer vision sector. Different researchers from around the world have devoted their time to make the algorithms suitable for handwritten word recognition. Different implementation has been forwarded using existing and newly discover algorithms. The reason behind popularity of such applications in the world is due to the variance of languages around the world. At first, the priority was given to the English word as it was considered as a global language. But soon, the researchers realize that this implementation can be applied to every language after the successfully recognizing the English Word. Since then, many algorithms and techniques have been discovered and used for the language recognition around the world. In my projects, I have tried to build system that can recognize the English words and for that, various references have been taken from different past implementations and research. Here, some of the systems similar to my project are given below.

A. Arabic Handwritten Word Recognition Based on Neural Networks

They build the model based on neural networks for recognition of Arabic handwritten word. The Arabic Handwriting dataset (AHDB) written by many Arabic writers has been used to train and test the model. The implement their system in 4 steps:

1. Preprocessing: AHDB have some noises due to its lighting, formats, various writers etc. So, they preprocessed their datasets for preparing their images to the next steps of implementation. They used Fuzzy C means clustering (FCM) for the binarization of images. FCM produce some noises on the images which again removed by applying 3X3 median technique.

They also removed the black space form the written words by using boundary box. Image thinning concept has also applied to reduce the repetitive pixels of the images without the losing the important information of the image. Finally, they applied image normalization because the dataset has images in different formats and sizes, which also helps in faster recognition. The image was normalized into size 128*128.

2. Feature Extraction: They applied this technique for the recognition purpose. They categorize it into three main components i.e. structural features, statistical features, and global transformations.

a. Structural Features: Such features are the geometrical and topological features defined by their global and local properties. The structural features found in Arabic words are dots, zigzag, end points, loops, strikes and intersection points.

b. Statistical Features: It is extracted from the statistical distribution of pixels. They found two types of statistical features in their datasets:

(a) Connected Components: It is extracted because the Arabic words consists of different number of connected components pixels or segments. They are extracted by scanning the whole image from left to right. (b) Zoning Features: First, they divide the image into different zone and extract features from each zone.

c. Global Transformation: It is applied to reduce the dimensionality of features. They used two transform methods for this step. (a) Discrete Cosine Transform Features (DCT): It converts the image pixel value from their spatial domain into its elementary frequency. (b) Histogram of Oriented Gradient (HOG): This is applied to counts the occurrence of gradients orientation in the image of AHDB datasets.

B. Handwritten Word Recognition using MLP based Classifier: A Holistic Approach

The model built here is trained on the CMATERdb1.2.1 datasets which is a collection of mixed of Bangla-English words. The word images first binarized using thresholding technique. Then, the holistic features are computed from the word image. First, the features are computed from different orientations such as rows wise, column wise and diagonal wise. After the features are computed, word images are partitioned in a horizontal form which then divided into vertical form for extracting the local features. Hierarchical partitioning is implemented by considering the hypothetical line around the word image, producing two image segments. These segments are again partitioned in same way. In this way, the features are measured from the original images on each segment. The compute features are then fed to an MLP classifier for the identification of the image. The datasets consist of training and testing sets in the ratio 2:1. They trained the network examining different parameters and succeeded to achieve the accuracy rate of 83.24% (Ankush Acharyy, 2013)

C. An Efficient Way of Handwritten English Word Recognition

Figure 10: Sequential Steps for Handwritten Word Recognizer

They proposed the system for the offline English handwritten word recognizer. They used the handwritten datasets written by various people for the recognition purpose. These are the following steps implemented for their proposed system.

1. Pre-processing: Pre-processing technique is applied on the bases of the datasets noise. These are the methods applied for preprocessing.

a. Skew and Slant Correction: They calculated the skewness in every column of datasets by identifying the pixels with least black color. Moreover, Least-squares linear regression is applied for determining the skew of the lines. Slant technique is used to minimize the variations in different writing styles and thus, improves the word recognition accuracy. The applies the Affine transformation technique to improves the slant in the input image.

b. Baseline Estimation: For, the extraction of the baseline, they scan the whole images from top to bottom. They detect the baselines which is the maximum no of black pixels in the image. They calculated the upper and lower baseline using the upper and lower black pixels of each column, respectively.

c. Horizontal and Vertical Scaling: The scaling is done on the input word image using Normalization technique. Feature extraction becomes more robust when the normalization is applied and thus, the position of handwritten word is correlated with the corresponding characters. This is applied after finding the baselines.

d. Skeletonization: It is the process of removing the foreground images from the images resulted from binarization. First, they applied the 2-dimensional Gaussian filter to smoothen image. Subsequently, the strokes are reduced from the word images using iterative thinning algorithm.

2. Feature Extraction: First, they segment the words into different zone by using vertical slicing method. Then, features are extracted from each zone and saved into feature vector. These are the features extracted in the form of features vector.

a. Dots: Dots in many letters cannot be separated from each other in the ascender part of the letter. So, they applied the connected component algorithm to overcome this problem. They did by assigning unique connected components as strokes. Strokes with size between 2 and 16 will be consider as dot.

b. Horizontal and Vertical Lines: They applied the technique called Hough Transformation for detecting the line in the word image. It also helps to isolate disconnections, distortions, and noise.

c. Loops: Connected component analysis is applied here to detect the loops. This is implemented by isolating the background color from the surrounding not connected to the word region.

d. Endpoints: These are the points where strokes ended. They detect the points using morphological operation, which returns 1 if it is an endpoint else return 0.

e. Junction Points: These are the strokes meet point. The window of 3*3 is slides over the preprocessed image to find the connectivity. The point at which connectivity is found is consider as junction point.

3. Classification: They used Euclidean distance-based K-Nearest Neighbor (KNN) classifier for the recognition purpose. It utilizes the feature vector of the both training and testing datasets and produces the matched word based on minimum difference value. The datasets on which the above classifier trained on consists of total 300 images. The training is done of 87% of total datasets and testing is done on whole datasets i.e. 100%. The classification accuracy achieved by this model is 90%.

D. Offline Handwritten Word Recognition in Hindi

They collected the datasets of Hindi words and characters from 100 writers. They collected of total of 805 images, combination of 540 characters and 265 words. They applied many techniques before the recognition of Hindi words.

1. Preprocessing: In this step, they perform binarization and refitting to enhance the images for the recognition. The characters size was also refined here.

2. Features: They used the Directional Element Features (DEF) while training the datasets and segment the characters while recognizing the datasets after the preprocessing. DEF counts the various

directional element on the contours of binary characters.

3. Word Segmentation: They applied this technique by putting the boundary box around words. Word segmentation will help the recognition process faster and easier. This will allow to produce a higher score when features are matched with the prebuilt characters after extraction of features form images.

4. Word Recognition: They recognize the words using Two-pass dynamic programming-based algorithms. This algorithm works in two passes. In the first pass, the resulted matching scores between the super segments of input words and the words stored in the lexicon are computed and stored. In the second pass, it determines the optimum super segments sequence matched with the input word by applying a dynamic programming-based search algorithm on the matched scores. The word with the highest score will considered as the word recognized. They trained with the character datasets and tested with the word datasets. They used two methods for recognition of words. In method 1, they trained their datasets using K-means algorithm only while in the method 2, they had created the synthesized vectors with the K-means algorithms. The recognition

accuracy of both methods for different lexicons sizes.

(Sitaram Ramachandrula, 2012)

2.3. Analysis

Different system has been trained on different datasets. The classifier has chosen by each system based on the type of datasets and research purposes. Some has given more stress on preprocessing stage while other including me give more focus on postprocessing stage. But the preprocessing steps is depending on the type of datasets because some datasets are already stored in preprocessed form. Among the all systems, my system has lowest accuracy rate because the system has been trained on minimum datasets and the model has built to trained on larger datasets which contradicts the implementation process.

CHAPTER 3 DEVELOPMENT

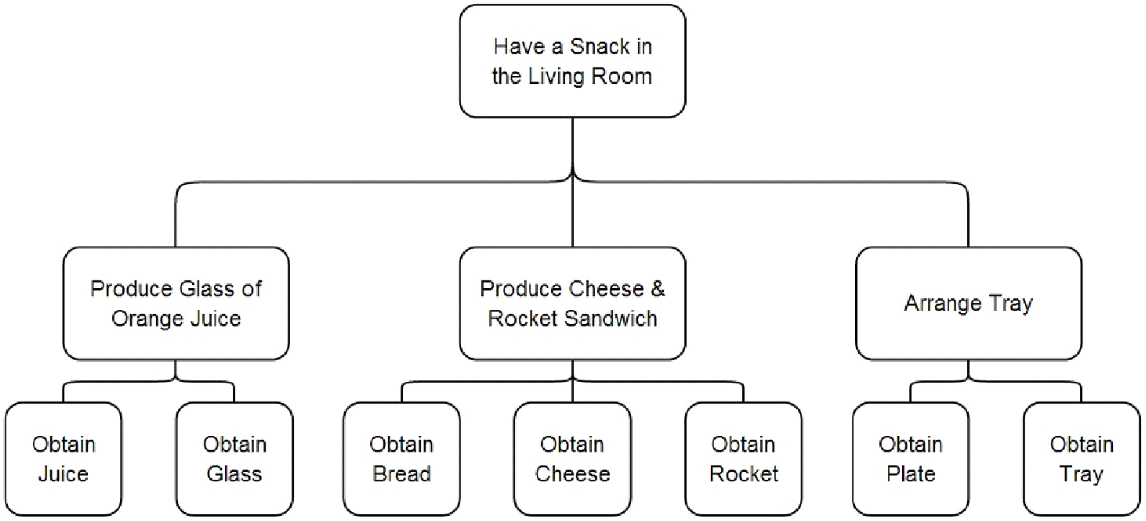
3.1. Initiation

3.1.1. Project Plan

I planned right after the proposal has been approved by the supervisor. While I had also planned before the proposal was submitted but the plans did not go with the proposal. There were many obstacles I was facing while doing my projects. For instance, the allocated time for some of the task was not sufficient while other had allocated with more time than they actually needed and other unavoidable circumstances. So, the completion of project was 1 month delayed.

3.1.1.1. Work Breakdown Structure (WBS)

First, the environment was setup for easing and speeding up the research and implementation. All the resources and plans were prepared to proceed the research part such as identifying stakeholders, requirements gatherings, PRF, Proposal, Prioritize requirements etc. After finishing the all initial research part, the developments stage is initiated for the building of the system. In development part, time and cost are allocated to the artifacts that were to be built. All the designs, development, unit test, and review part had been included here by following the agile methodology. Soon after each artifact developed, integrated testing was implemented to check readiness of their deployment. Integration testing was included in implementation part. The whole process is represented through WBS which is given below



3.1.1.2. Gantt Chart

a. Initial Gantt Chart

This is the initial Gantt chart mentioned in the proposal. Here, the project planned to finish on May 6, 2020 which was been postponed one month back because of the obstacles during the project development.

b. Final Gantt Chart

This is the Final Gantt chart extended from the time mentioned in the proposal. Here, the project is planned to finish on June 6, 2020 which had been successfully achieved.

3.1.2. Scope Identification

3.1.2.1. Fact Finding Techniques (FFT)

I use the fact-finding technique to find out the user's requirements i.e. Functional and Non-Functional requirements. There are mainly seven fact-finding techniques which I had implemented four.

a. Interview: Interview provides an opportunity to consult individually with the stakeholders. Interview can have both closed and open-ended questions. I did interview to the people who are accessible for me such as friends, family members, my seniors, and juniors. These are the questions I has asked them to identify the requirements.

- How will this system help you?
- Have you use any applications or website to perform OCR before? If yes, then what are the things that you like and

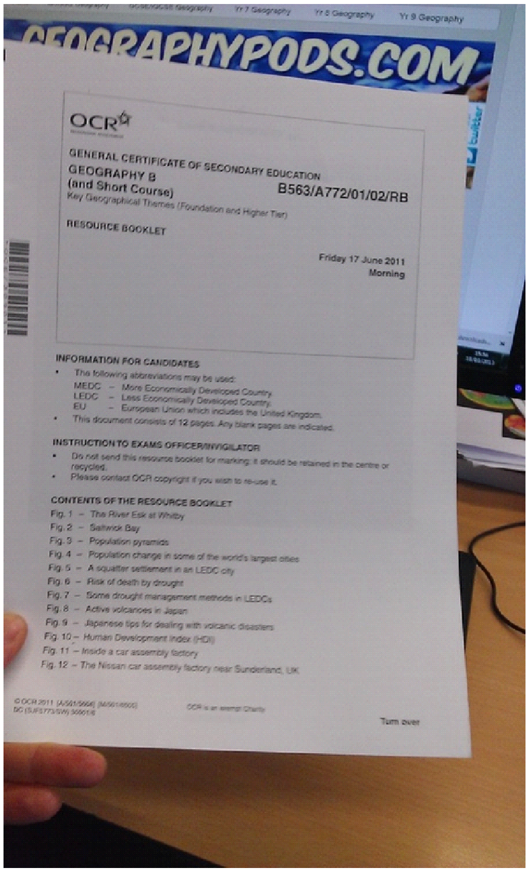
dislike most about that app or site?

- Will the user need to register/login for performing OCR? If yes, what kind of information needs to encrypt?

- Can you pay for using the system if it has subscription features?

- Do you need to see history of your performing OCR?

b. Document Review: The old system might have used papers and documents. By reviewing such papers, I can determine the working flow of the system and users' requirements. So, I did research on the Internet to find some papers related to the project. I collected some survey forms and questionnaires filled by the users.


Yr 7 Geography
Yr 8 Geography
Yr 9 Geography
GEOGRAPHYPODS.COM
twitter

OCR

GENERAL CERTIFICATE OF SECONDARY EDUCATION
GEOGRAPHY B
(and Short Course) B563/A772/01/02/RB
Key Geographical Themes (Foundation and Higher Tier)

RESOURCE BOOKLET

Friday 17 June 2011
Morning

INFORMATION FOR CANDIDATES

- The following abbreviations may be used:
  MEDC – More Economically Developed Country
  LEDC – Less Economically Developed Country
  EU – European Union which includes the United Kingdom.
- This document consists of 12 pages. Any blank pages are indicated.

INSTRUCTION TO EXAMS OFFICER/INVIGILATOR

- Do not send this resource booklet for marking; it should be retained in the centre or recycled.
- Please contact OCR copyright if you wish to re-use it.

CONTENTS OF THE RESOURCE BOOKLET

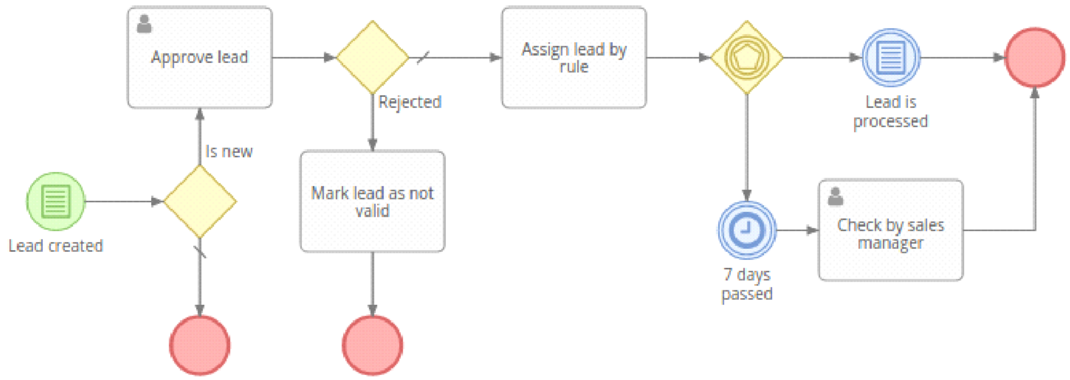
OCR is an exempt Charity

Turn over

c. Observation: This technique requires to visit onsite to know the business flow of the system. While I did not visit any organization physically, but I watch the videos online about different tech companies. This information had given me the idea of how they tackle the initial users' requirements along with the future changes' requirements.

3.1.2.2. Business Process Modelling (BPM)

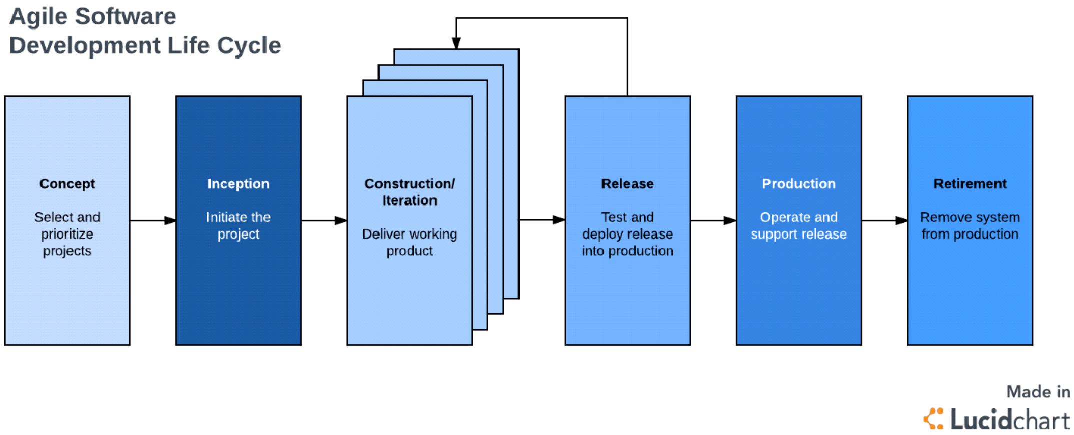
Business Process Modeling determines the working flow of business through graphical representation such as flowchart, data-flow diagram etc. The benefits provided by the BPM are improves efficiency, enforce best practices, process agility, transparency, beat the competition etc. (Amit, 2020). The BPM of the whole proposed system is presented below through flowchart.



3.1.2. Methodology

Among the different methodology, I choose the Agile Methodology for my project. This methodology combines both incremental and iterative process models focusing on customer satisfaction, process adaptability and rapid delivery of working software product. In agile, task is divided into small chunks to deliver features for a release. This practice requires continuous collaboration of each, and every individual involve in development process. Due to such flexibility, this can be applied in my final year project despite not a group project. I had planned to build each artifact by no means of order. So, having flexibility in methodology can help in achieving that goal. (tutorialspoint.com, n.d.)

3.1.2.1. Development Life Cycle Stages



a. Scope out and prioritize projects

It is the first step of the agile software development life cycle. The development team identify all the scopes and prioritizes the project. They also focus on the business opportunity, time, and work it will take to complete the project. (Lucidchart Content Team, 2017) Identifying and prioritizing the projects is also important for the individual project. So, I also had identified the requirements and priories it not on the bases of importance and demand by the users but on the bases of my easiness

b. Diagram requirements for the initial sprint

After identifying the scope, the team works with the stakeholders to determine requirements. The team will also focus on some initial designing part for introducing some new features.(Lucidchart Content Team, 2017)

I had not talk to the stakeholders, but I research the common problem facing by the people on the Internet and that helps to identify the requirements.

c. Construction/Iteration

Once the requirements have been identified, UX designer and the developers start to work on their first iteration with the aim of releasing the product at the end of sprint. (Lucidchart Content Team, 2017)

After, I list out the all requirements, I started to work on my first artifact.

d. Release the iteration into the production

After the development of first iteration, system is tested for the quality assurance, detecting bugs etc. Once they address any quality issues, they focus on the user documentation part. Lastly, they release the iteration into production.(Lucidchart Content Team, 2017)

When I develop my first artifact i.e. front end, I tested for the any issues and bugs. But I did not release the product because the product was built to showcase for my final year but not as a commercial product.

3.1.3.2. Why Choose Agile Methodology

Agile methodology has been practicing for all kinds of development – software, web, and mobile apps. The methodology has many advantages for projects initiated by both group and individual. These are the most common reason for choosing this methodology for my project.

a. Constant Communication between teams

It encourages constant communication between various teams which helps to remove the issues with the team members about the project. (Dixit, 2016)

I had also included brainstorm session each day for reviewing my works and making further plans.

b. Customize App Development

Continuous feedback from the stakeholders in time can helps to customize the app without compromising on the resources that were allocated. (Dixit, 2016)

As my project is individual based and it is not a commercial product, I get feedback by myself.

c. Seamless Project Management

Each team member has their own task allocated during the app development project. Allocating task to every individual helps to release a product in time and the highest amount of productivity is achieved. (Dixit, 2016)

This is the main reason why I choose for my project. This concept had helped in completing my project in time. I had breakdown the whole system in different components/artifact so that I can manage time and complete my project in time.

d. Improve Customer Experience

The customer itself involved in the process of developing app. So, the users can know where the project is going. This helps to improve the user experience by directly involving with the project.

(Dixit, 2016)

I had no users to collaborate with me but my experience on applying this methodology for my project is remarkable.

3.2. Implementation

3.2.1. Project Pipeline

My project consists of a model trained to extract words from natural scene images. When the users upload images, the detector will preprocess the image and detect the words in the images. After the detected words are cropped out and save it for classification. The classifier will then predict the word with highest confidence and return (*) if the word is not in the dictionary. These are the step by step implementation of my project.

a. Word Detection

Most of the word detection from natural scene image is done by two methods in the past i.e. character region and sliding window. Character region detect words first by segmenting each character and then group characters into words. Sliding window techniques slides over the character and grouped into words using a pictorial structure framework. Unlike those methods, I used word recognition method based on CNN which takes the whole word image as input to the network for recognition. In text detection, the three main tasks had been done and these are: (Max Jaderberg, 2015).

b. Word Recognition

The bounding box have been generated after performing segmentation and detection. Now, at this stage of pipeline, implementation is focused on recognizing the words inside bounding box. I use CNN to perform classification across a pre-defined dictionary of words which explicitly model natural language. The cropped images of each words inside the bounding box is passed as input to the CNN. CNN reshape the image and send it through its network for prediction. Each word classification will have a list of words and their resulting confidence or probability. CNN checks the probability distribution with all words available in the dictionary i.e. in our case 1000 words. The word with the highest probability or confidence will be recognize as the output for that image. But if the probability is below than 0.5, the system will return "(*)" as the word since the word is not in the dictionary. When all the words have been classified, the system will return the list of words to the web front to be displayed as a sentence of classified words. (Max Jaderberg, 2015)

3.2.2. Tools and Technology used

There are many tools and technique were used during the development of proposed system. These are the key tools and techniques with their uses are given below.

- HTML: HTML stands for HyperText Markup Language which is used to build the web pages and connect them with each other in a single website or between websites. It is used for building the front-end pages of proposed system i.e. Home page and Upload Page
- CSS: CSS stands for Cascading Style Sheets. It is used to describe how HTML elements are displayed in the web pages. It is used in proposed system to style the home and upload pages.
- Python: Python is a popular general-purpose programming language used for a wide variety of applications such as web development, software development, system scripting etc. The python is used in this project for different purposes such as building models, creating web application, data visualization and

analysis.

- Flask: Flask is a web framework written in python used for web applications and web services like SOAP, REST etc. It is used in the system because it allows to use python for both the model and web server, making it easier to import model into the application. It is used over Django because the proposed system does not require Django's numerous features for templating, routing, authentication, and database administration.
- OpenCV: OpenCV is a cross-platform library for the computer vision, machine learning and image processing and Realtime applications. It is used in this system for the preprocessing of datasets, for example, binarization using Otsu algorithm.
- Tesseract: Tesseract is an open-source engine for the character recognition. It implements artificial intelligence for searching and recognizing the text from images. It is used in this project for the segmentation and detection of words from images. Tesserocr, a python wrapper for tesseract, is used for the implementation of tesseract.
- NumPy: NumPy stands for Numerical Python. It is a python

library used for working with arrays. It also provides the functions for linear algebra, fourier transform, and matrices. It is used for the normalization of datasets, conversion of datasets and new images upload by the users into NumPy array.

- Pillow: Pillow is a free and open-source python imaging library used for opening, manipulating, and saving image file format. PIL is an older version of Pillow. It is used in this system to convert the images into pixel intensity arrays. It is applied while doing binarization and segmentation.
- TensorFlow: TensorFlow is an end to end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources. It is used for building the neural networks (CNN) as a backend for keras.
- Keras: Keras is an open-source neural network library written in Python. It runs on top of TensorFlow, Theano, or Cognitive Toolkit (CNTK). It is used in this proposed system for building neural network (CNN), runs on top of TensorFlow backend.

3.2.3. System Requirement Specification (SRS)

3.2.3.1. Functional Requirements

Functional requirements are the requirements or functions a software or systems must perform. These are the functional requirements for the proposed system.

- The system should allow users to upload images to perform OCR.
- The system should allow users to edit and copy extracted words.

3.2.3.2. Non-Functional Requirements

Non-Functional Requirements are the requirements which ensure the usability and effectiveness of the entire software system.

- The system should extract the words from images and displayed to the users when the upload button is clicked.
- Web pages should be simple and user friendly.

3.2.3.3. Usability Requirements

Usability Requirements are the requirements which determines the easiness of system while using it.

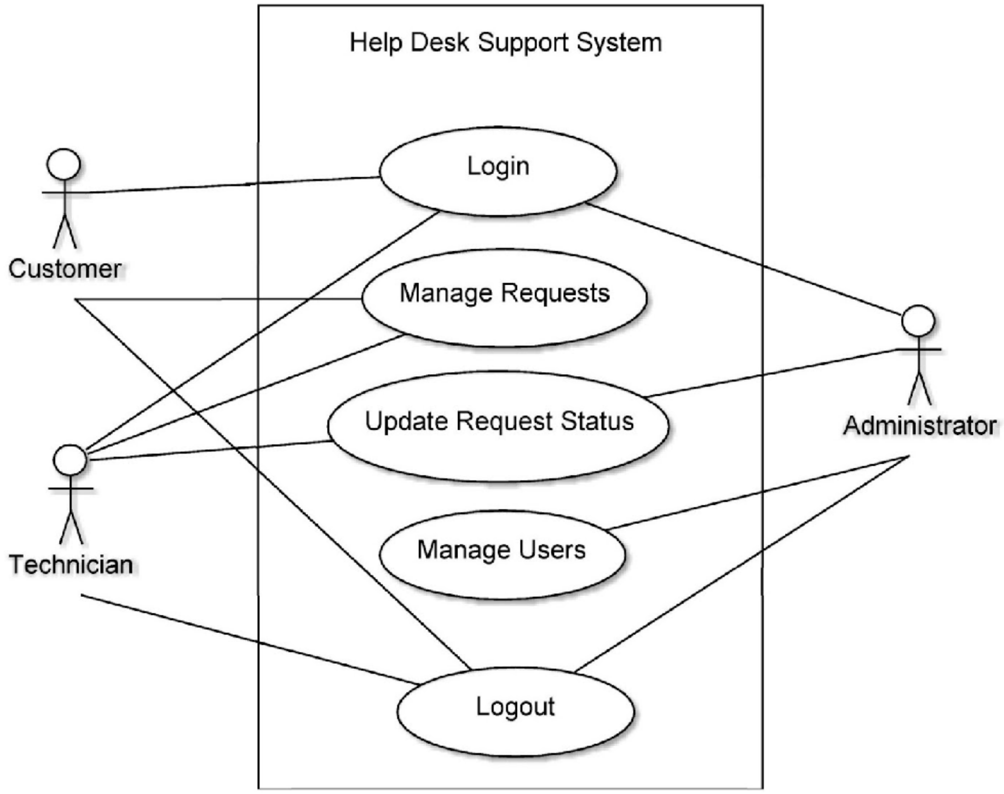
- The system should validate the form if the upload button is clicked without uploading the image.

3.2.4. Design

3.2.4.1. Process Design

a. Use Case Diagram

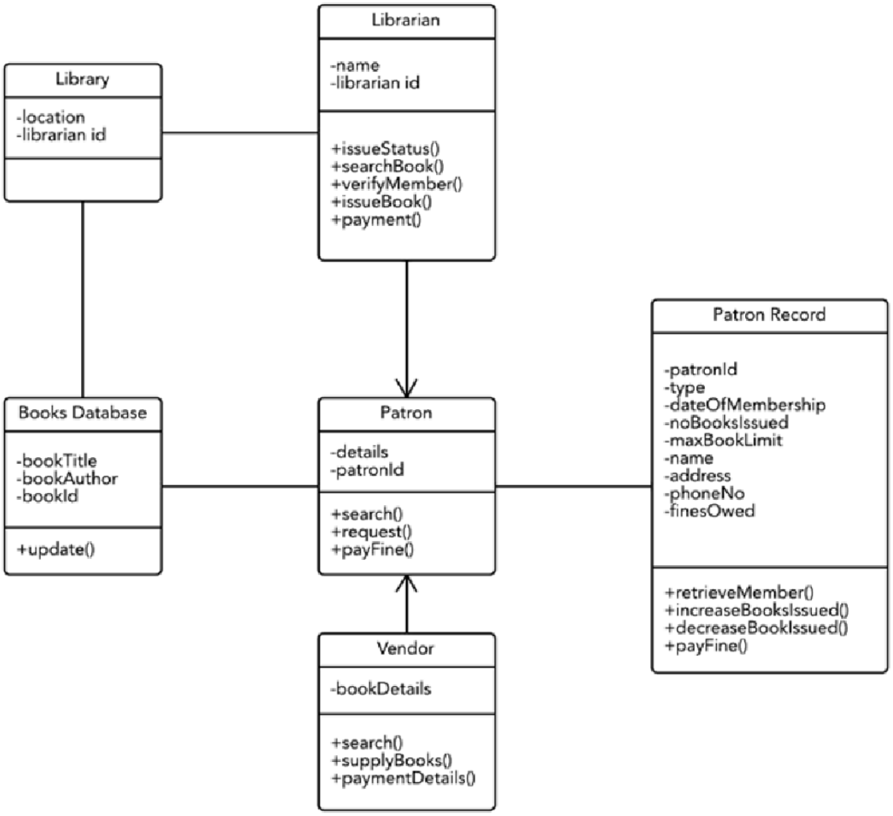
Use Case Diagram provide the details of system's users and their interaction with the system. The use case diagram for the proposed system is given below.



b. Class Diagram

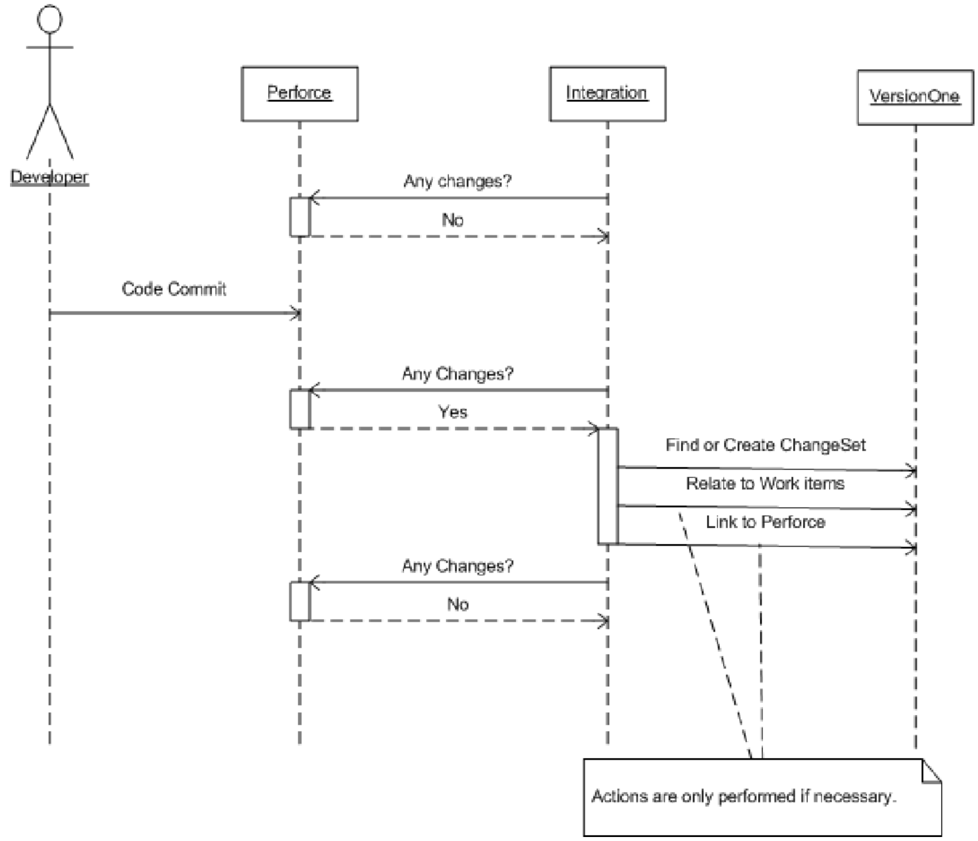
Class diagram shows the structure of the system by modeling its classes, attributes, methods, and the relationship among objects.

The class diagram for the proposed system is given below.



c. Sequence Diagram

Sequence Diagram provide the details how operations are carried out by the system. The sequence diagram for the proposed system is given below.



3.2.4.2. Database Design

The proposed system is designed for the simple demonstration of the model's capabilities. So, the database is not used in this project. But it can be implemented in future after adding some more features which requires database. For instance, login, registration, history saving etc.

3.2.4.3. User Interface Design

User Interface or UI design represents the visual layout of the elements of the system that a user interact with. UI design focuses not just on look or style but also on the friendliness of the site to the user.

3.2.4.4. System Architecture

The system architecture describes the major components with their relationships and how they interact with each other. The Architecture for the proposed system is given below.

3.3. Testing

Black Box Testing is the high level of testing which focuses on the behavior of the system. It does not have any information about the internal working of the software system. While white box testing is a testing technique which checks the internal functioning of the system. There is not much components in the proposed system. So, Testing is done manually.

3.3.1. Test Plan

Table 1: Test Plan

Test Cases Objectives

a. System should allow to upload two image files i.e. png and jpg to perform OCR.

b. Form must be validated to check whether the image is uploaded or not before performing OCR.

c. System should return a character i.e. “(*)” when it does not recognize the word. But it should return multiple asterisks when recognizing the sentences.

d. Word must be extracted in editable form.

3.3.2. Test Cases

a. System should allow to upload two image files i.e. png and jpg to perform OCR.

Table 2: Test Case a.

Steps Required Expected Result Actual Result

• User should able to select the images in jpg or png format. Image should upload Image is uploaded

b. Form must be validated to check whether the image is uploaded or not before performing OCR.

Table 3: Test Case b.

Steps Required Expected Result Actual Result

• User clicks the upload button before the image selected Prohibit from clicking the upload button Upload Button is not clicked.

c. System should return a character i.e. “(*)” when it does not recognize the word. But it should return multiple asterisks when recognizing the sentences.

Table 4: Test Case c.

Steps Required Expected Result Actual Result

- User select image
- User upload image

It should return “(*)” Returned asterisk (*)

4. Answering Academic Question

The answers regarding the academic questions of proposed system are given below:

a. How will the system work and what techniques, tools and technologies will it use to extract text from images?

The answer to this question is already included in the literature review and development section. The proposed system works by holistic method i.e. take the whole word as input and recognize it without detecting each character like in character recognition. Users upload the image into the website, the backend will save the image into its own local storage. With that image, the detector will grays cale, binarize it, and create a temporary copy of the image for word detection. For every word detected, the detector will crop the word out and save it for classification. After the word detection is complete, the server will remove the

temporary image. With the cropped word images, the classifier will create a list of word results. For each word image, the classifier will reshape the image and send it through its network for prediction. Each word classification will have a list of words and their resulting confidences. The classifier will extract the highest confidence and its respective word and append it to the word list. If the confidence level is below 50%, the system will return "(*)". When all of words have been classified, the system will return the list of words to the web front to be displayed as a sentence of classified words. This is done by using manytechnologies and libraries such as HTML, CSS for front end and flask, NumPy,
OpenCV, Tesseract, TensorFlow, Keras, Pillow for the backend.

b. Will the users need to log in for using this system?

In the proposed system, the users cannot login into the system and this is made so to make easier for the users while performing OCR. This is very annoying when the users have to login to the system every time while performing OCR. This feature can be included in the future if the system is diverged into the commercial form, but the proposed system was built only for the demonstration of my final tear project. So, login features haven't implemented in the system.

c. What is the accuracy of your system? Will your system performance decrease if the image is provided with enough noise?

After examining with different parameters of the model, the model achieves the accuracy rate of 68%. The system's aim is to extract words from the natural scene images rather than plain documents. The natural scene images have lots of noise in comparison to plain documents. So, model is trained to handle noises as well. But if the noises are in extreme level, then the system might perform poorly.

d. How will the users get benefited by this system?

The system was built for the convenience of the users. Before the concept of OCR came, people were compelled to store their data, information in hard documents. In corporate field, they had to copy the whole documents while storing the information. But with the OCR, people can store in the electronic from without going through the hassle of copying the whole information in the form of text. So, this proposed system will help them to store data by extracting word from images in editable form.

Conclusion

5.1. Summary

The project was initiated with the aim of building the platform for the users where they can perform OCR. The proposed project presents the OCR web application where user extract the words from images in edible form. Many components were integrated to build a complete system. First the datasets were collected and built the model and finally implemented the flask for the integration of front end and back end. It was done by using many technologies and algorithms such as Keras, TensorFlow, NumPy for building the CNN, Pillow, OpenCV and Tesseract for the preprocessing and postprocessing stages respectively, and HTML, CSS for front end. The accuracy rate achieved by the model was 68%. The entire system was built in agile methodology as it is very flexible and easily adaptable in any projects. The project was planned to breakdown the systems into many artefacts and allocated time for each task. Different methods were used for the project for gathering the requirements such as interviews, Observation, Document review etc. The requirements gathered from this method were successfully built in the system. Research had been done on similar systems for finding the information on the tools and techniques used on their system. Finally, the system was built in a way that it can answer the academic questions and fulfill the aims and objectives.

Future Escalation

Currently, this end-to-end system is a prototype that is only able to detect a subset of the English language. Below is a list of improvements that can be implement when given more time:

- Design and implement a custom detector that does not rely on Tesseract. This allows us to customize how we detect words and characters for the classifier.
- Train the neural network so that it does not classify by words but by characters. This allows the system to recognize and translate entire documents without relying on a dictionary of words. This reduces the complexity and training time for the network since the total size of the English alphabet is much smaller than the dictionary of English words.
- Implement a neural network ruleset that predicts words when given combinations of characters. This may increase the complexity of the network, and conflict with the second above point.

Critical Evaluation

The project was proposed to build the OCR web application with the implementation of AI components. Various machine learning algorithms are available for the word recognition but in the proposed systems, CNN was used because it is considered as the start-of-the-art algorithm for the image processing in the field of computer vision. The accuracy of the model was improved by changing its hyper parameters. First, the model was trained with the five layers i.e. 3 convolutional layer and 2 fully connected layers and the accuracy was very poor around 47% after 25 epochs. After 25, the model was started to overfitting. So, I extended the network up to 8 layers i.e. 5 convolutional layer and 3 fully connected layer. The accuracy was far better than the first architecture i.e. 60%. While the accuracy was improved, but the improve rate was very slow. It was taking average of 5 epochs to improve the accuracy by 1%. So, the model is trained again with changing some hyperparameters which then increased accuracy by 8% in just 25 epochs. The parameters were changes are dropout function and activation function. The optimizer was also changed from SGD to Adadelta while compiling the model. The datasets were preprocessed using OpenCV and tesseract was used for the detection part. While tesseract consists of algorithm which can recognize the words form the images but here, it was used only for the detection purposes as CNN was best for the 2D images. The project planning was first done with respect to the submission deadline of final year project. All the tasks were listed from initiation to closing in the form of Work Breakdown Structure (WBS). The time were allocated for each task and presented in the graphical representation from known as Gantt Chart. The time was allocated more on the development part in comparison to other parts. This has been all planned with respect to the first final submission deadline. Later, the deadline is extended further because of some obstacles and unavoidable circumstances. The Gantt chart was again updated according to the second final submission deadline.

References

Ahmed El-Sawy, M. L. H. E.-B., n.d. Arabic Handwritten Characters Recognition using
Convolutional Neural Network. [Online]
Available at:
http://www.wseas.org/multimedia/journals/computerresearch/2017/a045818-075.pdf
[Accessed 09 12 2019].

Alia Karim, M. S. K., 2018. Handwriting Word Recognition Based on Neural Networks.
International Journal of Applied Engineering Research, Volume 10, pp. 43120-43124 .

Amit, 2020. Business Process Modeling: Definition, Benefits and Techniques. [Online]
Available at: https://tallyfy.com/business-process
modeling/#:~:text=Business%20process%20modeling%20is%20the,data
flow%20diagram%2C%20etc.
[Accessed 20 5 2020].

Ankush Acharyy, S. R. R. S. S. B. M. N., 2013. Handwritten Word Recognition Using MLP

based Classifier: A Holistic Approach. IJCSI International Journal of Computer Science,

10(2), pp. 422-426.

Anuj Dutt, A., 2017. Handwritten Digit Recognition Using Deep Learning. International

Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 6(7).

Dixit, P., 2016. Why Choose Agile Methodology for Mobile App Development. [Online]

Available at:

https://www.tristatetechnology.com/blog/agile-methodology-for-mobile

application-development/

[Accessed 26 April 2019].

Fathma Siddique, S. S. M. A. B. S., n.d. Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance

for Various Hidden Layers, Dhaka: Department of EEE, International University of

Business Agriculture and Technology.

Flor, A., 2019. Text Segmentation. [Online]

Available at: https://medium.com/@arthurflor23/text-segmentation-b32503ef2613

[Accessed 28 5 2020].

Gregory Cohen, S. A. J. T. A. v. S., n.d. EMNIST: an extension of MNIST to

handwritten

letters, Australia: s.n.

Jabir Ali V, J. T. J., 2018. A Convolutional Neural Network based Approach for

Recognizing Malayalam Handwritten. International Journal of Scientific &

Engineering

Researc, 9(12), pp. 167-170.

Jin, J., 2014. A Flash Card Android Application Development Applied with OCR

technology, s.l.: Helsinki Metropolia University of Applied Sciences.

Kenton, W., 2019. Functional Decomposition. [Online]

Available at: https://www.investopedia.com/terms/f/functional-decomposition.asp

[Accessed 22 5 2020].

Kuguoglu, B. K., 2018. How to use image preprocessing to improve the accuracy

of

Tesseract. [Online]

Available at:

https://www.freecodecamp.org/news/getting-started-with-tesseract-part-ii

f7f9a0899b3f/

[Accessed 13 5 2020].

Lucidchart Content Team, 2017. The Stages of the Agile Software Development

Life

Cycle. [Online]

Available at:

https://www.lucidchart.com/blog/agile-software-development-life-cycle

[Accessed 2 April 2019].

M.S. Patel, S. L. R. A. J. N., 2015. An Efficient Way of Handwritten English Word

Recognition. Advances in Intelligent Systems and Computing 32, Volume 2, pp.

563-569.

Max Jaderberg, K. S. A. V. A. Z., 2015. Reading Text in theWild with

Convolutional Neural

Networks, New York: Springer Science+Business Media .

Meer Zohra, D. R., 2019. A Comprehensive Data Analysis on Handwritten Digit

Recognition using Machine Learning. International Journal of Innovative

Technology and

Exploring Engineering (IJITEE, 8(6).

OpenCV Computer Vision, 2020. Image thresholding. [Online] Available at: https://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html [Accessed 10 5 2020].

Prashanth Vijayaraghavan, M. S., n.d. Handwritten Tamil Recognition using a Convolution Neural Network, s.l.: MIT Media Lab.

Shize Shang, N. M. X. K., 2014. Detecting documents forged by printing andcopying. EURASIP Journal on Advances in Signal Processing.

Shubham Sanjay Mor, S. S. S. G. S. D. M. J. R. S. ., 2019. HANDWRITTEN TEXT RECOGNITION: with Deep Learning and Android. International Journal of Engineering and Advanced Technology (IJEAT, 8(3S), p. 820.

Sitaram Ramachandrula, S. J. H. R., 2012. Offline Handwritten Word Recognition in Hindi, Mumbai: s.n.

Thunstrom, J. B., 2016. Introduction on OCR. [Online] Available at: https://medium.com/@Johanni/introduction-on-ocr-242d5dbd3d01

[Accessed 13 5 2020].

Trung Quy Phan, P. S. S. T. C. L. T., 2013. Recognizing Text with Perspective

Distortion

in Natural Scenes. Singapore, IEEE International Conference on Computer Vision.

tutorialspoint.com, n.d. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) ,

s.l.: s.n