Ge'ez Handwritten Character Recognition System with Machine Learning

A PROJECT REPORT

Submitted by

Name	ID
Eba Alemayehu	TER_0075_09
Kerebish Genet	TER_0083_09
Nardos Sharifo	TER_0088_09

in partial fulfilment for the award of the degree of

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

Under the guidance of

Mr. Baye Atinafu

\-

ADVISOR SIGNATURE

DEPARTMENT OF SOFTWARE ENGINEERING

INSTITUTE OF TECHNOLOGY

DEBRE MARKOS UNIVERSITY

DEBRE MARKOS

February 2020

Abstract

The project tries to build a comprehensive handwritten Ge'ez characters dataset. In the project we try to find the best deep learning model witch best fit to learn handwritten Ge'ez characters. A couple of sub systems are also built as a part of the project.

Acknowledgment

We sincerely want to thank our adviser who were helping us in this journey. We also want to thank our department for providing us an office support we have needed for our project. Last but not the list we want to thank all volunteers who participated in filling the quaternary.

Table of Contents

List of table

List of figures

1. Introduction

1.1. Background of the project

Ge'ez is liturgical language of the Ethiopian church. Ge'ez is a Semitic language of the Southern Peripheral group, to which also belong the South Arabic dialects and Amharic, one of the principal languages of Ethiopia ((1)). Ge'ez has its own writing style and alphabet. Both Ge'ez and the related languages of Ethiopia are written and read from left to right, in contrast to the other Semitic languages. Extinct as a vernacular language, Ge'ez is the ancestor of the modern Tigrinya and Tigré languages of Eritrea and Ethiopia. The oldest known inscription in the language dates from the 3rd or 4th century and is written in a script that does not indicate vowels.

Most of Ethiopian history and documentations were written in Ge'ez characters. Either with Ge'ez language itself or other descendent languages of Ge'ez like Amharic. Before the era of computers and automation the government and other organizations used handwritten documents. These documents contain accumulated wisdom of our forefathers, the history of our country, the history of gov't records like police records etc... Therefore, we need some automated way to help us digitize these documents in order to make storage and distribution of these documents match easier.

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ		ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ	
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ	ሏ	ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ	
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ	ሗ	ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ	ዟ
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ	ሟ	ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ	ዧ
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ	ሧ	የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ	
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ	ሯ	ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ	ዷ
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ	ሷ	ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ	ጇ
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ	ሿ	ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ	ጧ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ	ቧ	ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ	ጯ
ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ	ቯ	ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ	ጷ
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ	ቷ	ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ	ጿ
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ	ቿ	ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ	
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ	ፏ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ	ኗ	ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ	ፗ
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ	ኟ	።	፡	፣	፤	፥	፩	፪	፫
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ	ኧ	፬	፭	፮	፯	፰	፱	፲	፳

ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኳ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ	ዀ

ዅ	ዂ	ዃ	ዄ	ዅ	ዅ	ዅ	ዅ

Figure 1: Amharic scripts

1.2. Statement of the problem

Typing Ge'ez characters into computer is relatively harder because the standard QWERTY keyboard which most of us use is not designed for Amharic language. It also requires a lot of labor. It is a very time-consuming task. These days people tend to use mobile or tablet devices rather than the conventional desktop computers. These devices have no convenient way of writing a long text. Especially writing Amharic text is much more difficult because though there exist few applications which allow as to write Ge'ez there is no standard for the key layout.

On the other hand, artificial intelligence and fully intelligent systems are growing. These systems are expected to dominate the world. This kind of systems incorporate artificial general intelligence which means they almost mimic a human mind. One of the issues in the part of this big general system call machine learning is the issue of diversity. Machine learning is an algorithm which takes data and tries to learn from that data. So that if it will not be trained with our language we will be left over in the digital divide. Which negatively affect us and our language. Therefore, we need to develop AI systems which can mimic our culture and our language. Reading or recognizing character are one of the skills of language we need to train machines.

Humans can easily recognize characters one they have learned them in spite of how distorted they are but identifying handwritten characters is not an easy computer vision task in a traditional way of programming, because it would be impossible to be able to write rules about how each character are represented. Humans write characters in unpredictable way; style of writing differs from person to person. Therefore, we need to use another approach which is machine learning. This pause a big computer since challenge but recently machine learning had become good at this kind of computer vision tasks because know we have good enough computational power to teach computers to detect this unpredictable writing of characters.

1.3. Literature review

Though the growth of computer technology people still use handwriting as it is still easy and convenient way of writing. For computers it is very easy to process encoded text in other hand humans have a very good visual capability to process handwritten text. Researches has been trying to recognize both hand written and computer printed characters from a photo graph for a long time.

There are two peridium's in handwritten recognition and segmentation-based and holistic segmentation base approach segment word image in to constituent character where are holistic approach tries to recognize the whole word ignoring character segmentation. Holistic approach also extracts repetitive features of the whole word and it is more pragmatic in case of cursive handwriting where characters are physically connected each other by segmentation turns out to be impractical (Yaregal Assabie a, 2019). Ge'ez writing doesn't have a cursive writing characteristic which makes it easy for segmentation. There for segmentation-based recognition would be feasible. This research also uses this segmentation-based recognition (Birhanu, 2010).

Offline recognition of Latin, Chinese, Japanese, Indian and Arabic hand written has long area of active research. However Ethiopic handwriting recognition in general and Amharic word recognition in particular, is one of the least investigated problems (Yaregal Assabie a, 2019).

A couple of studies were made on Amharic, Ge'ez or in general Ethiopic character recognition. Some of them used machine learning techniques where as others used different tadeonal algorithms. As far as our knowledge one of the pioneers are the study by [John and Fiaz Hussain]. This paper describes the Amharic script and discusses the difficulties of applying conventional structural and syntactic recognition processes. Two statistical algorithms for identifying Amharic characters are described. In both, the characters are normalised for both size and orientation. The first compares the character against a series of templates. The secondderives a characteristic signature from the character and compares this against a set of signature templates (Cowell, 2003). Even though this algorthm prformed relatively well on computer printed characers with a certain font. It has avery poor performance in handwritten characater recogntion. Other studies have pusshed forward the research area with different techniques like HMMs, support vector machine (Yaregal Assabie a, 2019) (Gauri Katiyar1, 2017).

One of difficulties in recognizing amharic charaters are the number of unique characates and the simmilarty between them (Yaregal Assabie a, 2019). CNN have shown a remarkabel performance in recognizing such a commplex patterns (Sen Maitra, 2015). Recent remarkable research used estate of the art deep nural network and convolutional nural networks (Mesay Samuel Gondere, 2019). This research used CNN with the techinque called Multi task learning. The dataset used by this research were from the work of (Assabie & Bigun, 2008). It uses 12 unique handwritten characers and it use different kinds of agumantaion to increase the number of data to 450, 800 and 400 Traing, validation and test set. Finally the result of the reserch shows. Different accurecy lavels in average bettwen 50% and 75%.

1.4. The Objective of project

1.4.1. General objective

To build a computer system that can recognize any handwritten Ge'ez characters.

1.4.2. Specific objective

The specific objective of this project is:
- Preparing Ge'ez characters dataset which is publicly available and anyone who want to experiment on it can try out.
- To find the best learning algorithm and neural network architecture
- To train a model which can classify Amharic characters
- To make application software which make use of the trained model

1.5. Scope of the project

This project consists of:
- Ge'ez character dataset collection
- Designing the learning algorithm and architecture
- Training the model with the collected dataset and Designed architecture
- Building an application on the top of the trained model
- Different small programs that help to automate some tasks on the process for instance data collection.

This project does not include:
- Any natural language processing. Our system does not understand the meaning of a text.
- No semantic analysis or data mining on text is done.
- Word or sentence-based recognition. Our system is character-based recognition.

1.6. Significance of the project

The significance of the project can be seen in different dimensions. On one side the system we are going to build an application which can solve some problems. It includes API that developers we amazing idea can build applications. On the other side when we see the big picture it could be one step forward for next AI projects that can be done with our juniors.

1.7. Tools and methodology

1.7.1. Data Collection methodology

To train our model we need a lot of data. We are planning two ways to collect data.
  I. By distributing questionnaire paper to different people to get there handwriting.
 II. Building an android app which help us collect characters data. The app will have a canvas to enable draw characters on screen.

1.7.2. Technologies to be used

I. Programming languages
- Python
- JavaScript
- Java, swift or dart (optional)
- Html and CSS

II. Tools and technologies
- OpenCV
- Tensor flow
- Flutter
- Koras
- NumPy
- Matplotlib
- Django or flask

1.7.3. System requirements hardware and software

2. Operating system
- Linux: will used as the development and training operating system
- Windows: for documentation and some drawing
- MacOS: will be used for compiling iOS apps
3. Software
- Android studio for android development
- Visual studio code as a text editor
- Apache or nginx serves
- Google chrome for testing and debugging JavaScript
4. Hardware
- Two computers one for training the model one for a development. The specification for these two computers are listed as follows.
  a) Training server: core i7 processor, 16GB RAM 1 TB storage with GPU capability.
  b) Development pc: core i7 processor, 8GB RAM 1 TB storage.
- Android and iPhone devices for testing
- Printed paper for data collection
- Other office apparatus for different purposes

1.7.4. System modeling tools

- Microsoft Visio and project
- StarUML as modeling tool

1.8. Feasibility study

1.8.4. Technical

Thanks to the big improvements that deep learning brings into the computer vision world, we think the system is technically feasible with the resources we have now.

1.8.5. Operational

Through neural networks are relatively expensive in terms of computational resources we think it is feasible to operate our system in a computational power we have today

1.8.6. Economical

As we have mentioned earlier on the problem statement typing characters are time and resource consuming this project will allow our users to save a lot of time and resource so it is economically feasible.

2. System Analysis

2.7. Overview of existing system

Whenever we want to change a handwritten text in to computer understandable (editable text). We have to type the text again in to a computer system. People usually higher typists to do this work. In press industry usually, authors like writing in handwriting. When they are done with writing they take their work to the typists to type done to a computer. Also, media reporters take notes one some scenarios and type text to make news. We can mention a lot of areas where typing is used to change handwritten text to a computer.

Though there exist some character recognition system and a very few researches on handwritten Amharic character recognition systems, we don't think there still exist any Amharic handwritten character recognition system yet. Especially systems that are available for developers to work on more application systems which are based on this character recognition systems.

2.8. System Requirement Specification

2.8.4. Functional Requirements

- ✓ FR1: The system should allow data collection for Ge'ez handwritten characters
- ✓ FR2: The system should allow data labeling by authorized users
- ✓ FR3: The system should segment characters from the questioner scanned image
- ✓ FR4: The system should be able to classify handwritten characters at minimum of >75% accuracy
- ✓ FR5: The system should detect, recognize and localize handwritten characters from scanned image or digital input

2.8.5. Non-Functional Requirements

1. Response Time:
   - ✓ TR1: The system should have a response time less than 10 seconds for recognition, detection and localization
2. Capacity
   - ✓ TR2: The system is expected to handle 100 recognition simultaneously
3. User interface:
   - ✓ TR3: The system should have a standard user interface that is easy to use
4. Authorization:
   - ✓ TR 4: only authorized users should label training datasets
5. Working hours:
   - ✓ TR 5: The system should be available at minimum 95%
6. Device and hardware:
   - ✓ TR 6: The system should run on mobile devices well as computer devices as a client
   - ✓ TR 7: The system should run on a system which has a GPU or TPU for faster training and neural network propagation at backend.
7. Operating systems:
   - ✓ TR 8: The system should run on android, IOS, Linux and windows Operating systems
   - ✓ TR 9: The system should run on any server operating system but Linux server is recommended.
8. Browser:
   - ✓ TR 10: the system should support Google chrome, chromium based operating systems, Mozilla Firefox, opera, safari and edge.

2.8.6. Overview of proposed system

The proposed system is a trained machine learning model which can recognize any handwritten character recognition. The system uses neural networks in order to achieve this task. The network will be a classifier network.

Ones we could train a model with a satisfactory accuracy level we will make different applications and interfaces which uses this model. This includes a web app, mobile app, desktop app and an API for developers. There are a lot of applications which can be built on this model. Some of them are:

- Amharic writing learning app which teaches how to write Amharic characters
- Amharic road signs reading and translation app
- Amharic optical character recognition systems

3.   Dataset Collection

Machine learning algorithms inherently require a lot of data. The more the data the better the learning accuracy of the machine learning algorithm will be. More data also help as to easily overcome the bias variance trade of problem.

In our research we couldn't find any comprehensive dataset for Ge'ez handwritten character recognition. Though some researches tried to prepare some dataset, their dataset is either not complete or the dataset is not available. There for we have prepared our own dataset which we will use in our model training in the next phase of the project. We will also make this dataset public and opensource so that any one interested in the Ge'ez handwritten character recognition use it.

We have also developed a system which help us automate the data collection and labeling task. Though human intervention is required, the system allow to upload scanned images of questioner and slice the handwritten characters. It also allows labeling of the characters.

3.7.   Data gathering methodology

We have used questioner data collection method. We have prepared a questioner for people to fill in for us. Our questioner has 3 parts. The first part of our questioner instructs the person to rewrite an Amharic poem given on the questioner. We have prepared 6 different poems. Each volunteer participant will get one of the 6 poems. This part of the questioner allows us to test our trained model on consecutively written words than independently written characters. This will give us more real-world scenario. The second part of our questioner instruct the user to write 291 characters on the table provided. This are characters we are going to segment and train our model with. The third section will have the demographic data of the volunteer participant which include age, gender and educational level.

3.8.   Sampling mechanism

Random sampling method was used to select our sample population. We have tried to collect the data from any volunteers because we need a lot of data. We could collect 485 unique data from 209 volunteer participants. In general, we have managed to collect 141,135 unique characters. This data collection will also continue one phase to of our project.

3.9.   Data analysis and preprocessing

After we have collected the questioner, we have developed image processing algorithm which segment characters on the section two of questioner. The algorithm

will slice each character using OpenCV image processing library and save each character as separate JPEG image. Hear is how the algorithm works:

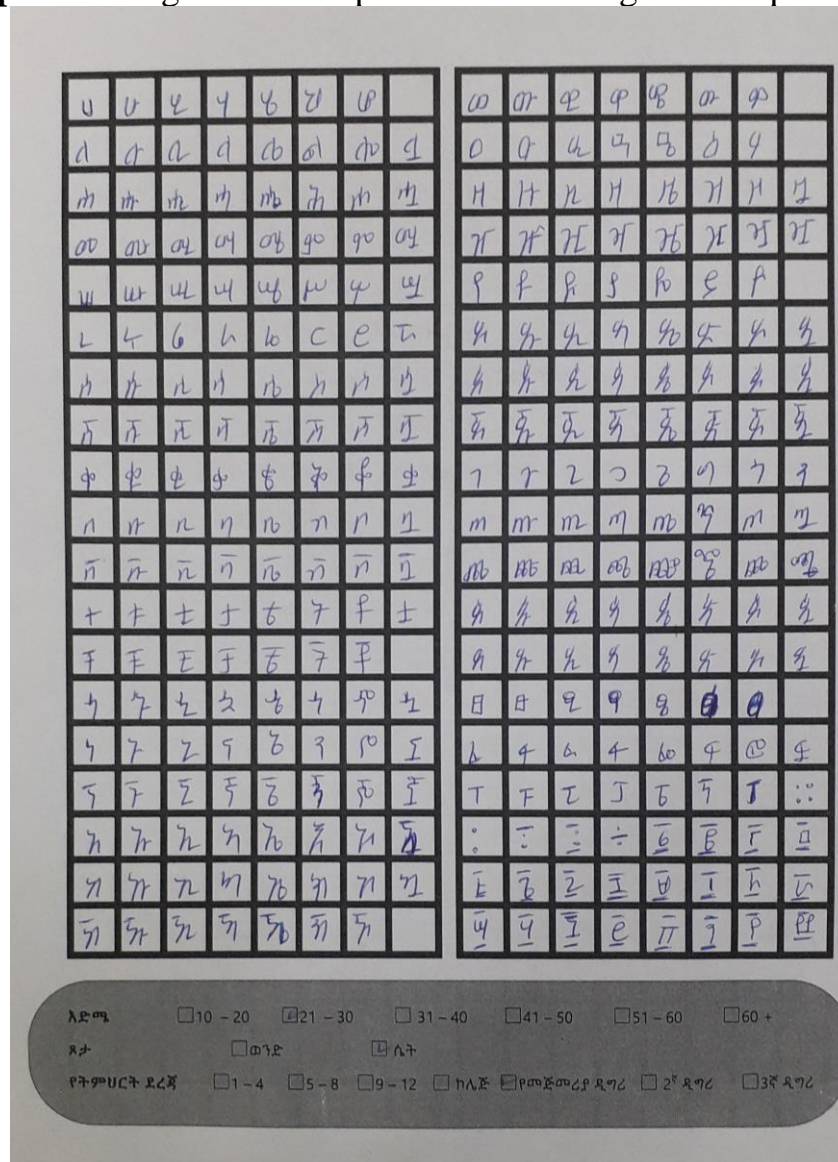**Step 1:** The algorithm accepts a scanned image of the questioner

Figure 2: Original scanned image

Step 2: The input image will be resizing to a size of 1200px keeping the proportion of the image. Also, since we don't need the color the algorithm converts the image in to a gray scale.
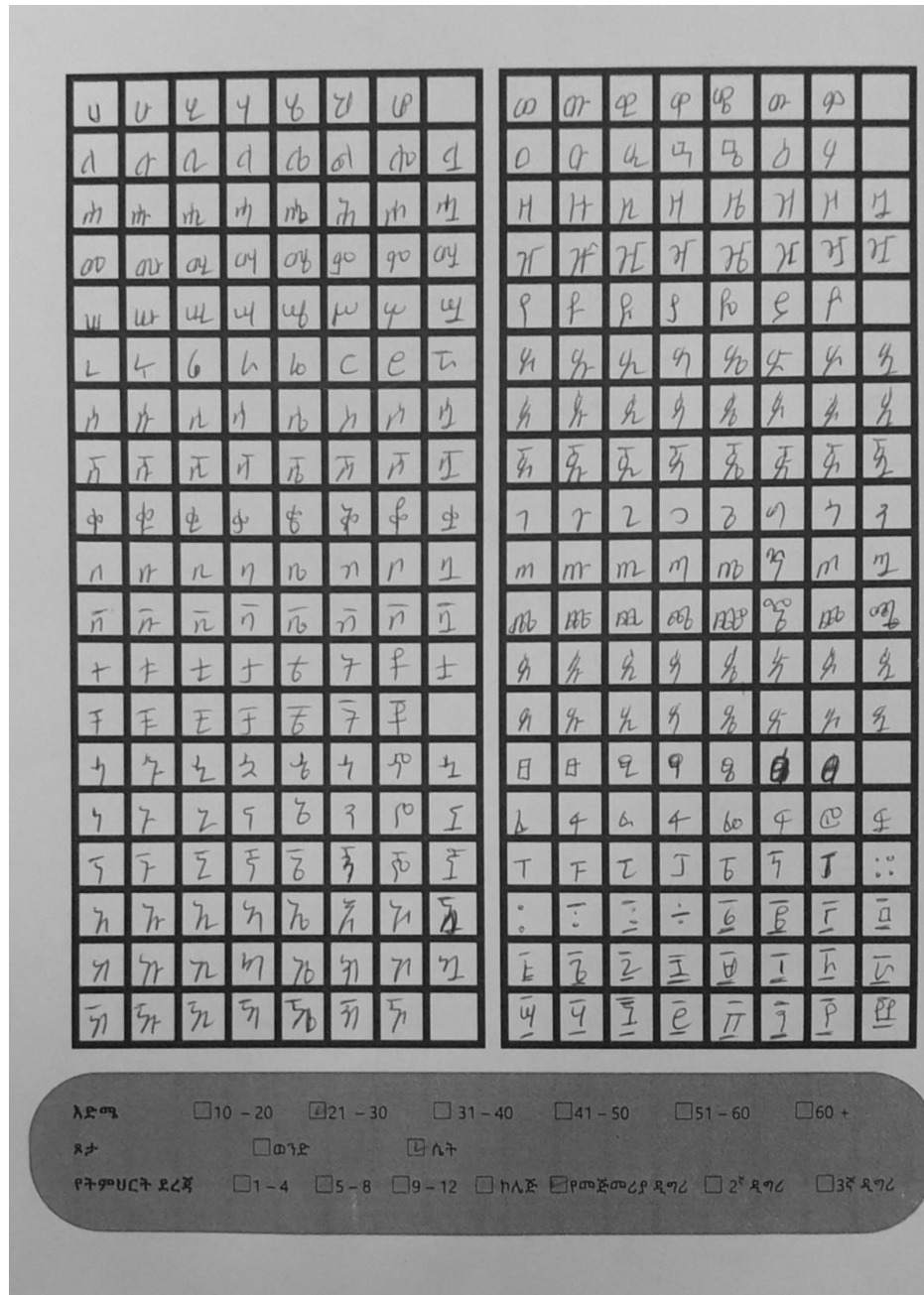
Figure 3: Resized gray scale image

Step 3: Next we will try to remove noise from the picture by using gaussian blur algorithm then thresh hold the image buy using OTSU thresh algorithm and invert the image. This means the final image will be a binary image 0 and 1 with black background and white text.
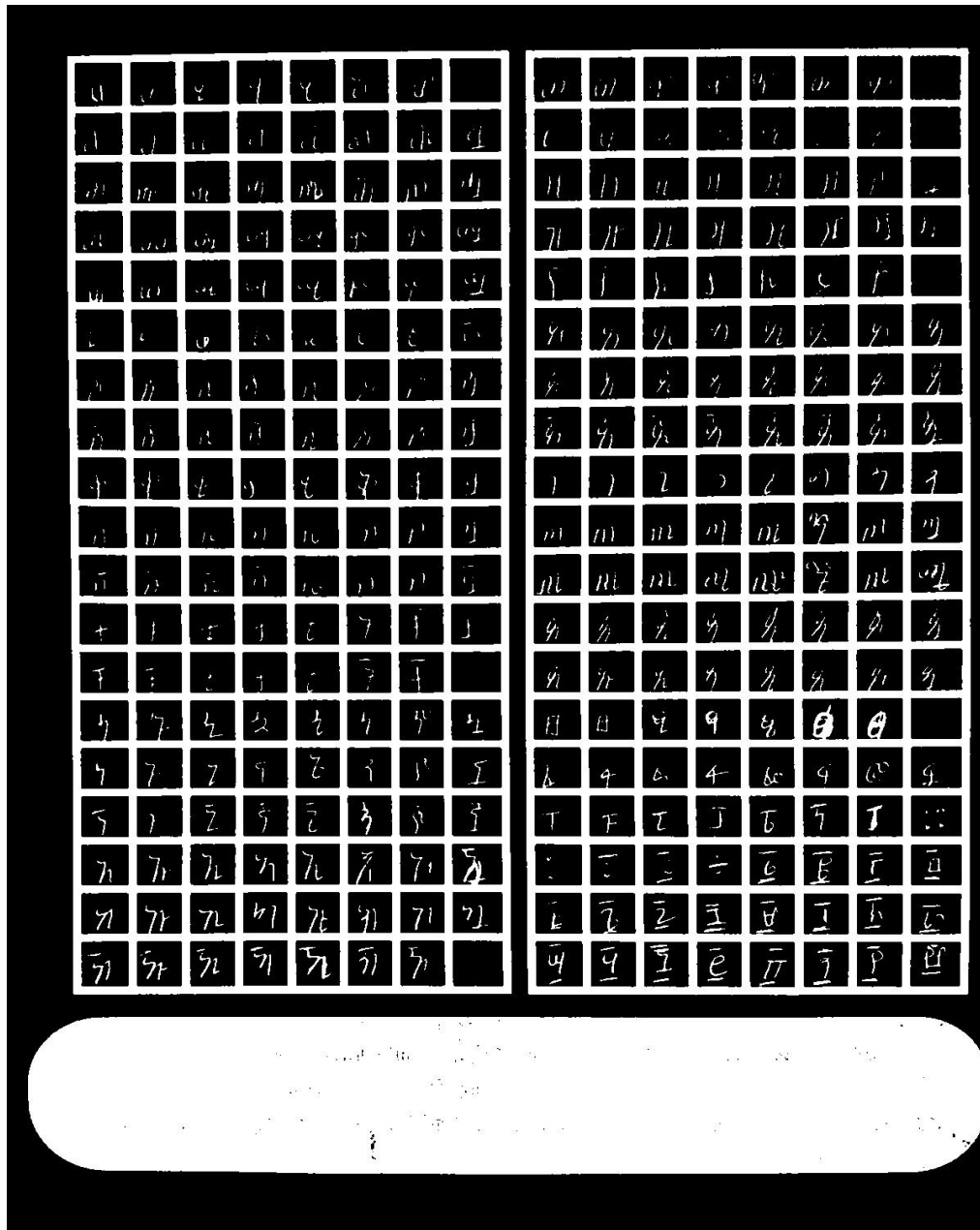
Figure 4: Filtering and OTSU threshold applied to the image

As you can see, we have managed to clear out the image and make it is for farther processing.

Step 4: the algorithm makes some morphological adjustment. This process helps us to clear some damaged edges of the lines during the previous processes.
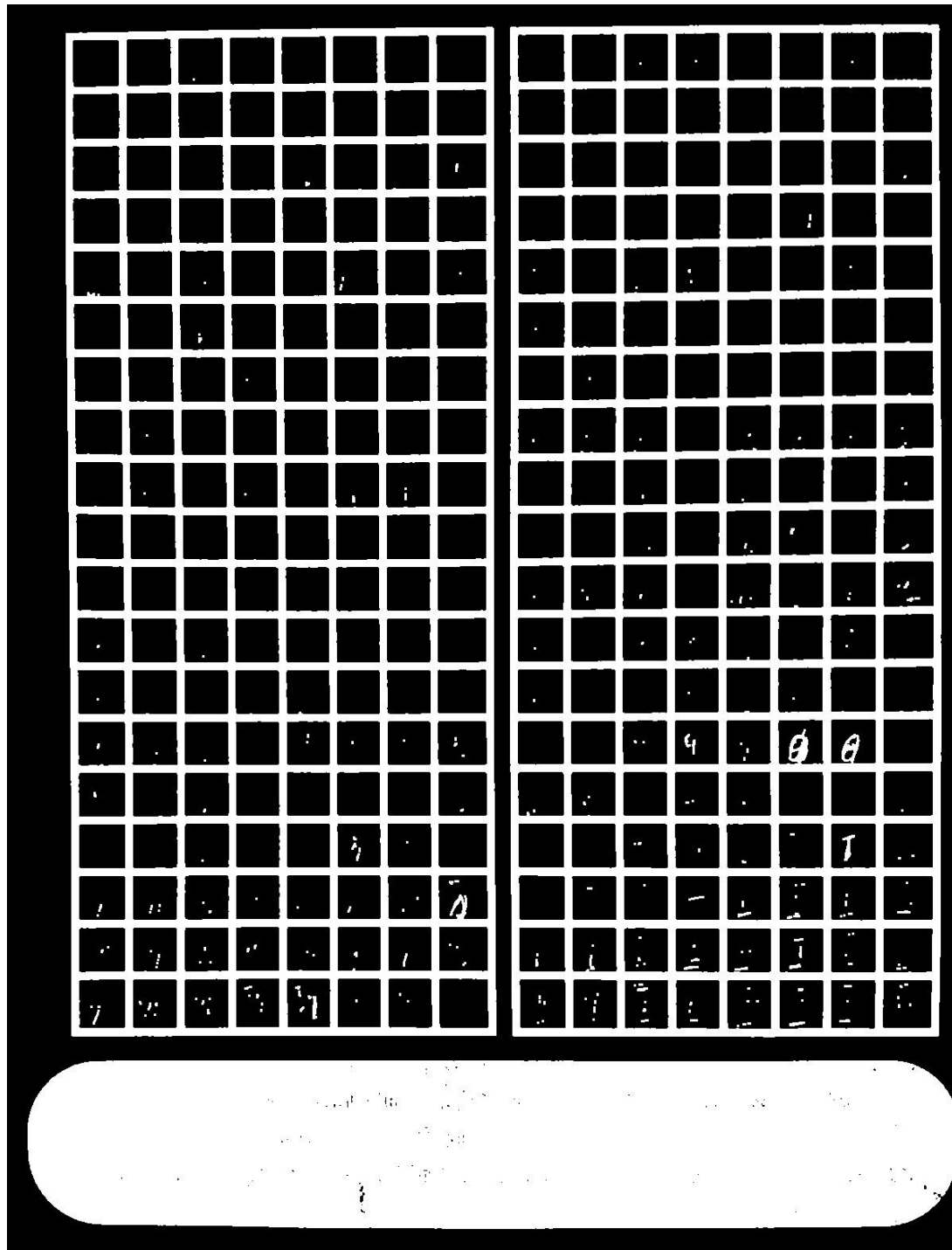
Figure 5: Morphological adjustment is made on the image

Step 5: Now the algorithm tries to detect only horizontal and vertical lines from the image using horizontal and vertical kernels.
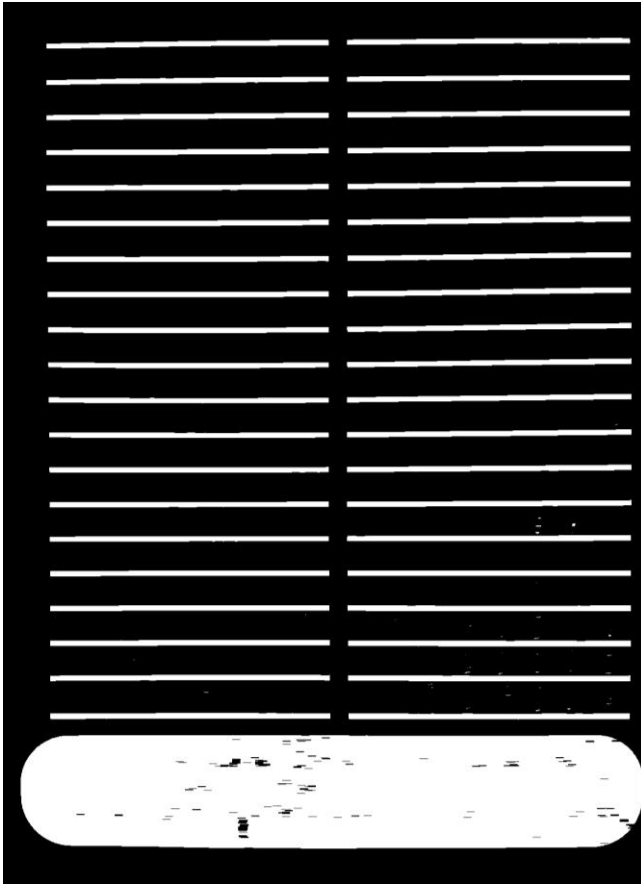
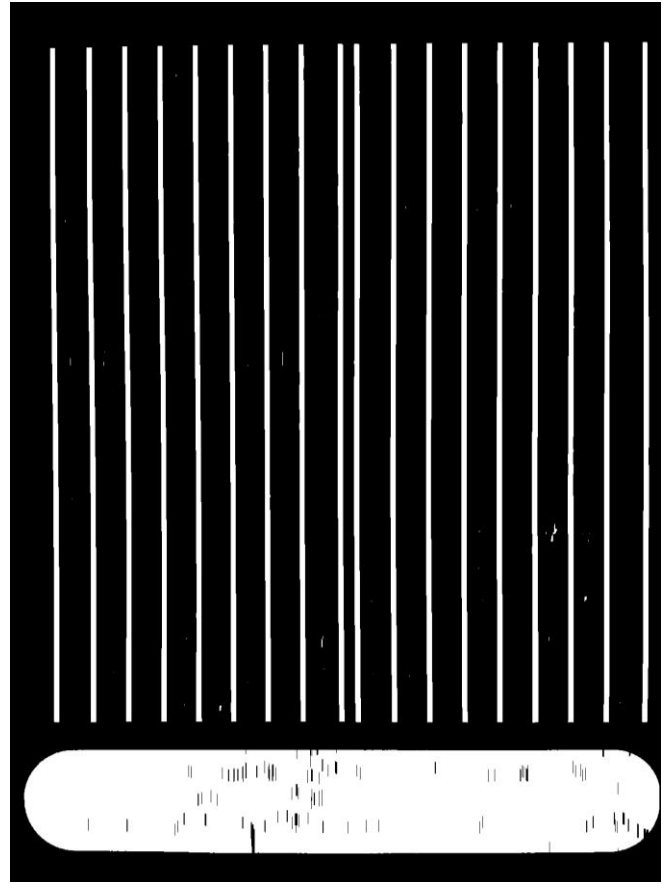**Figure 6:** Horizontal lines are extracted from image

Figure 7: Vertical lines are extracted from image

Step 6: Now the algorithm do bitwise or operation on two images to merge the two images together so that we can get an image which with only the lines of the table.
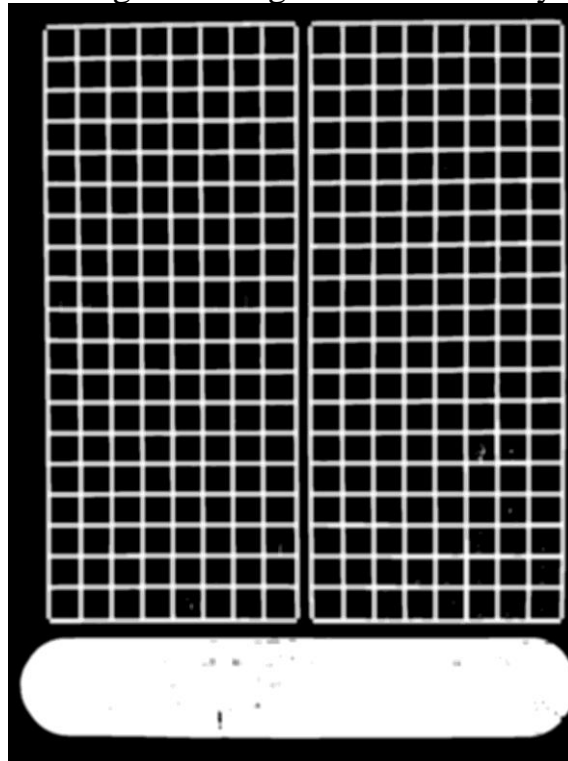

Figure 8: Bitwise or are applied on extracted horizontal lines and extracted vertical lines

Step 7: after we have detected the table, we will use canny edge detection algorithm to get separate boundaries of each box in the table.
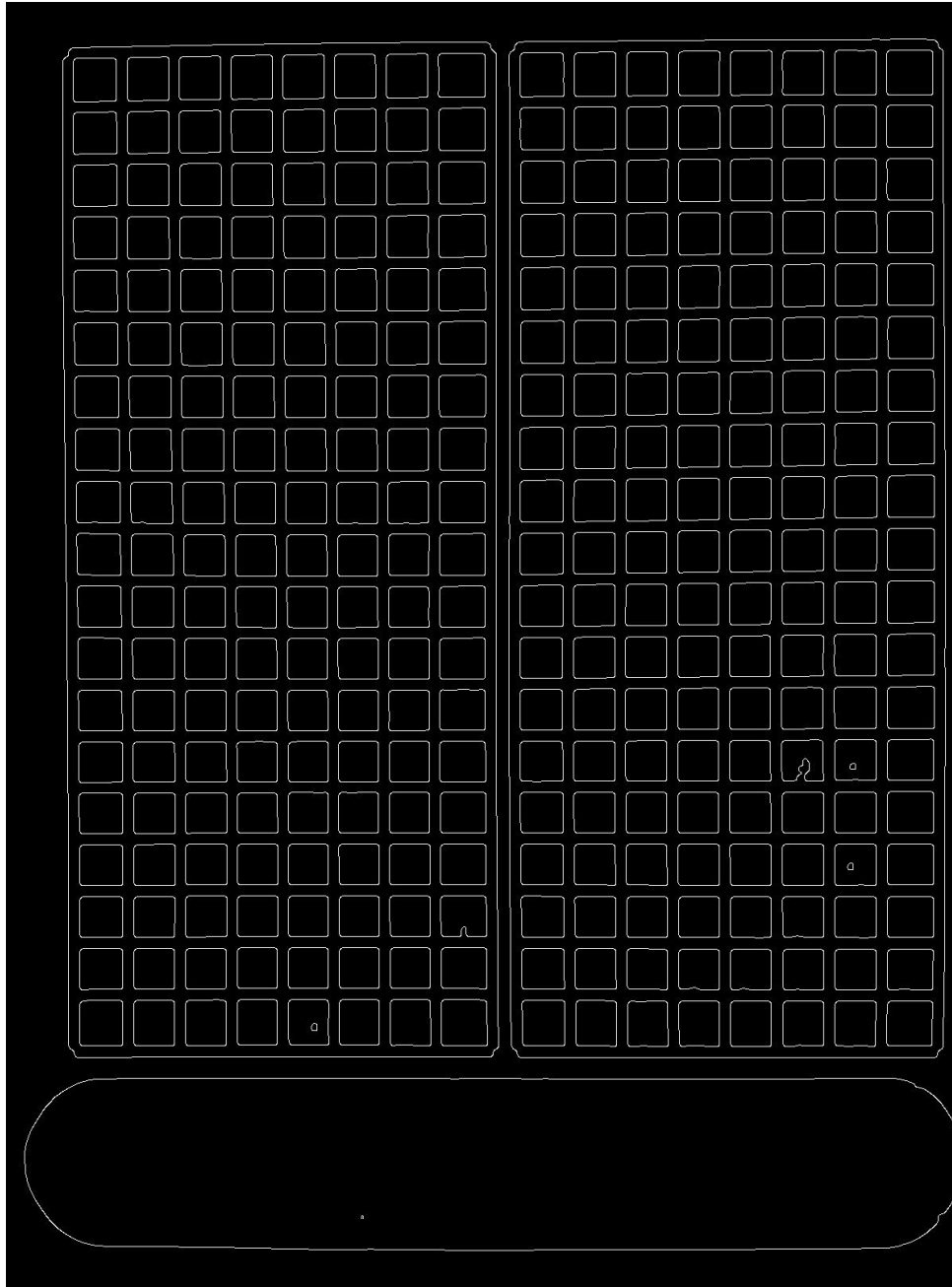

Figure 9: Canny edge detection is applied on the image on figure 7

Step 8: Now, the algorithm has got the location of each box one the image and also the area of the box. In order to prevent form detecting some noise smaller or larger boxes the algorithm only takes boxes with area between the mean area of all boxes and 1000. The algorithm saves the points in array.

Step 9: We have developed a sorting algorithm which uses one kernel which has the width of the image. This kernel scans the image from top to bottom and sort points in same row.

Step 10: Finally, the algorithm has sorted the points. Now it can crop each box containing character and save as a jpg file.

Figure 10: Sample segmented images

In this way the algorithm helps us segment each character in to individual images in automated way.

- **Note**: Sort coordinates algorithm will help as automatically label the images by their position
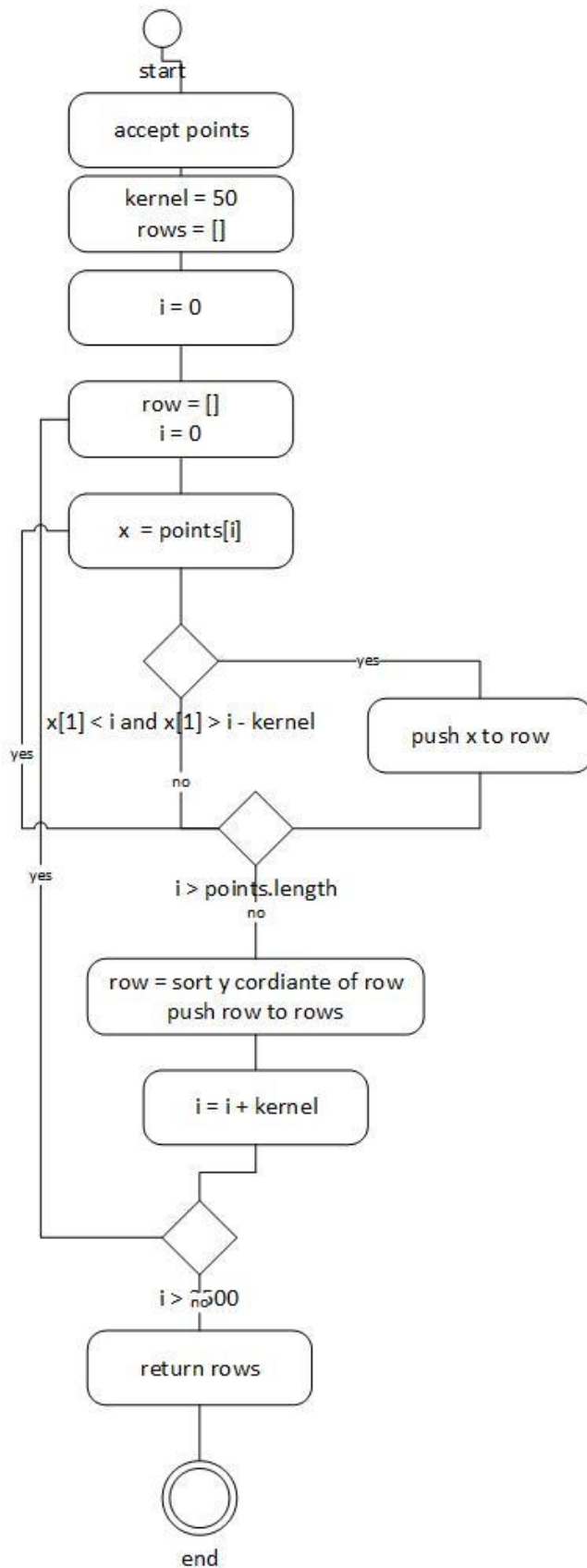
Figure 11 (left flowchart)

start

accept points

kernel = 50
rows = []

i = 0

row = []
i = 0

x = points[i]

x[1] < i and x[1] > i - kernel

yes → push x to row

no

i > points.length

no

row = sort y cordiante of row
push row to rows

i = i + kernel

i > 500

no

return rows

end

Figure 12 (right flowchart)

Accept scanned image

Resize image 1200px keep ratio

Noise removal with gaussian blur

Apply inverse OTSU thresh

Morphological adjustment

Vertial and horizontal line detetction

Mearg horizontal and vertial lines wit bitwise or

Detect edges with canny edge detection

avg_area = get avarage area of canny edges
cordinates = []

loc = Get location of bounding box

area = get area of bounding box

area < loc < 1000

push loc to cordinates

End of canny edges

sort cordinates

crop and save images

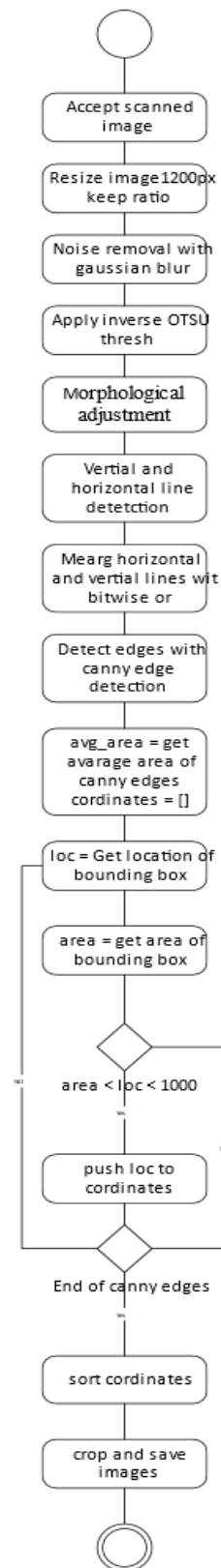**Figure 11:** Algorithm for sorting bounding box coordinates questioner   **Figure 12:** Algorithm for segmenting images in the

3.10. Software Requirement specification for data collection app

3.10.4. Functional requirement

- FR 6: Authenticate and authorize data collector
- FR 7: Allow the user to upload questioner
- FR 8: Allow the user to label images
- FR 9: Download a dataset generated form the images

3.10.5. Non-functional requirement

1. Response Time:
   - TR11: The system should have a response time less than 30 seconds for recognition, detection and localization
2. Capacity
   - TR12: The system is expected to handle up to 100 users simultaneously
3. User interface:
   - TR13: The system should have a standard user interface that is easy to use
4. Authorization:
   - TR1 4: only authorized users should label training datasets
   - TR 15: only authorize users should be able to upload scanned images
5. Working hours:
   - TR 16: The system should be available at minimum 95%
6. Device and hardware:
   - TR 17: The system should work on desktop environment
7. Operating systems:
   - TR 18: The system should run on Linux and windows Operating systems
   - TR 19: The system should run on any server operating system but Linux server is recommended.
8. Browser:
   - TR 20: the system should support Google chrome, chromium based operating systems, Mozilla Firefox, opera, safari and edge.

3.10.6. Overview of Proposed system

The proposed system is a web application which automate the data collection and labeling process. The application will have the following futures.

- Upload scanned images
- Segment images
- Allow agents to label images

3.10.7. Business rules

- BR1: Only authorized agents by system admin can upload images
- BR2: Any one can download the dataset
- BR3: System admin create account for agents

3.11. System requirement analysis for data collection application

3.11.1 Actor and use case identification

Actor name	Description
System admin	The administrator of the system
Agent	A person who is assigned to collect and upload data to the system

Table 1: Actors and their description

3.11.2. Use cases

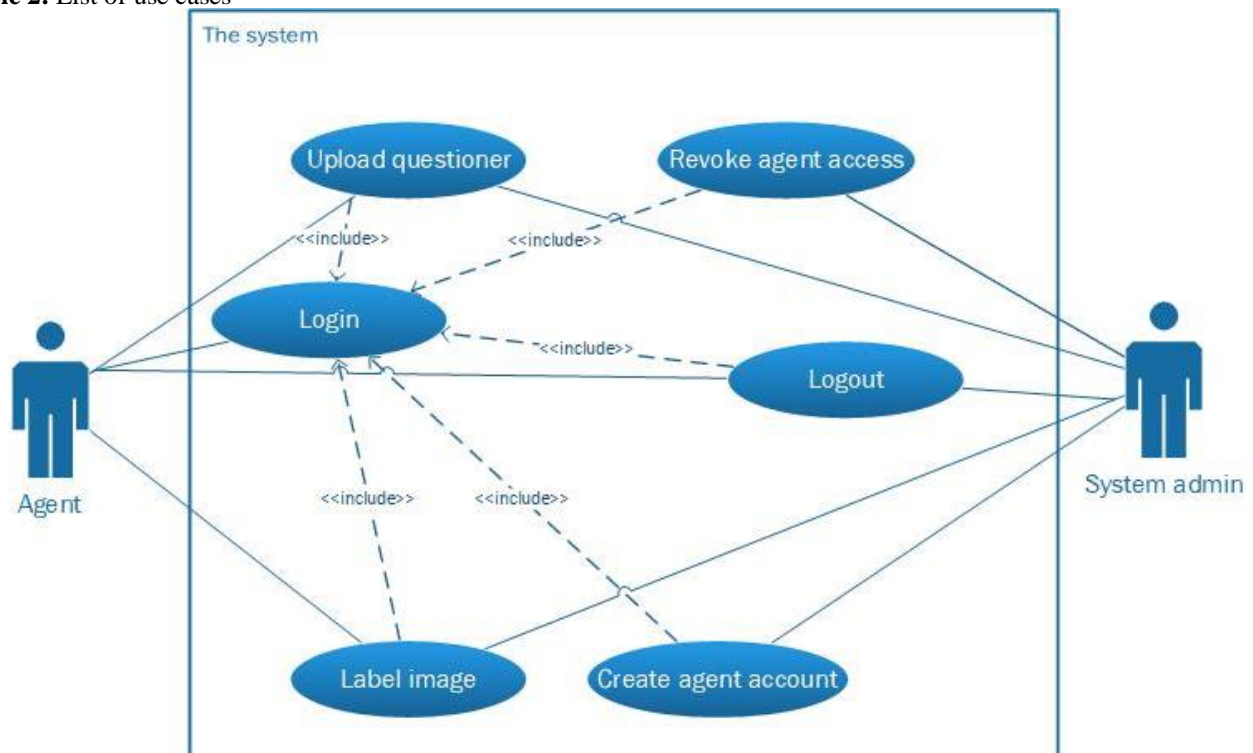| Use case id	Use case name	Include	Extends
UC01	Login		
UC02	Logout	Login	
UC03	Upload questioner	Login	
UC04	Label images	Login	
UC05	Create agent account	Login	
UC06	Revoke agent access	Login	

Table 2: List of use cases

Figure 13: Use case diagram for data collection application

3.11.2.1. Use case description

Use case name	Login

ID	UC01
Actor	System admin, agent
Description	The user of the system enter credential to access the system
Precondition	The user should have proper credentials
Posttension	The system admin or agent login into the system
Basic course of action	User action	System response
	1. The user open login page	
		2. The system shows login form
	3. The user enters credentials and press login	
		4. The system checks the credentials 5. Show home page 6. Use case end
An alternative course of action	A: If step 4 failed show error message A1: The system displays and error message A2: Start from step 3	

Table 3: Login use case description

Use case name	Logout
ID	UC02
Actor	System admin, agent
Description	The user logs out of the system
Precondition	The user should be logged in
Posttension	The system admin or agent out of the system
Basic course of action	User action	System response
	1. The user sends logout request	
		2. The system expires authentication key 3. Respond success message 4. Redirect the user to login page
An alternative course of action	A: If step 2 failed show error message	

Table 4: Logout use case description

Use case name	Upload questioner	
ID	UC03	
Actor	System admin, agent	
Description	The user fills the questioner form and upload questioner scanned images	
Precondition	The user should already be logged in to the system	
Posttension	The questioner image will be uploaded and new questioner is created	
Basic course of action	User action	System response
	1. The user navigates to questioner form	
		2. The system shows questioner form
	3. The user fills the questioner form and upload scanned images and submit the form	
		4. The system creates new questioner, segment and save uploaded images. 5. Show success message 6. Use case end
An alternative course of action	A: If step 4 failed show error message	

Table 5: Upload questioner use case description

Use case name	Label images	
ID	UC04	
Actor	System admin, agent	
Description	The user label uploaded images	
Precondition	• The user should already be logged in to the system • A questioner has to already been uploaded	
Posttension	The system labels the image	
Basic course of action	User action	System response
	1. The user selects the questioner and image to label.	

	2. The user navigates to labeling page	
		3. The system shows images and labeling form
	4. The user enters label for each image or verify if it was already been labeled	
		5. The system saves the label of the image 6. Show next image set to label 7. Use case end
An alternative course of action	A: If step 5 failed show error message	

Table 6: Label images use case description

Use case name	Create agent account	
ID	UC05	
Actor	System admin	
Description	The system admin creates agent account	
Precondition	The system admin should be logged in as system admin role	
Posttension	New agent is created	
Basic course of action	User action	System response
	1. Navigate to create agent form	
		2. The system shows create agent form
	3. Fill the create agent form login	
		4. The system creates new agent 5. Show list of agent page 6. Use case end
An alternative course of action	A: If step 4 failed show error message	

Table 7: Create agent account use case description

Use case name	Revoke agent access	
ID	UC06	
Actor	System admin	
Description	The system admin revoke access from the agent	
Precondition	The system admin should be logged in as system admin role	
Posttension	The agent can no longer access the system	
Basic course of action	User action	System response
	1. Navigate to agents list and press revoke button	
		2. The system shows confirmation message
	3. Confirm revoking access of the agent	
		4. The system revokes the access of the user 5. Show list of agent page 6. Use case end
An alternative course of action	A: If step 4 failed show error message A1: If the user cancel confirmation on step 3 the revoke process will be canceled and go back to step 1	

Table 8: Revoke agent access use case description

3.11.3. Sequence Diagram

Login Sequence diagram
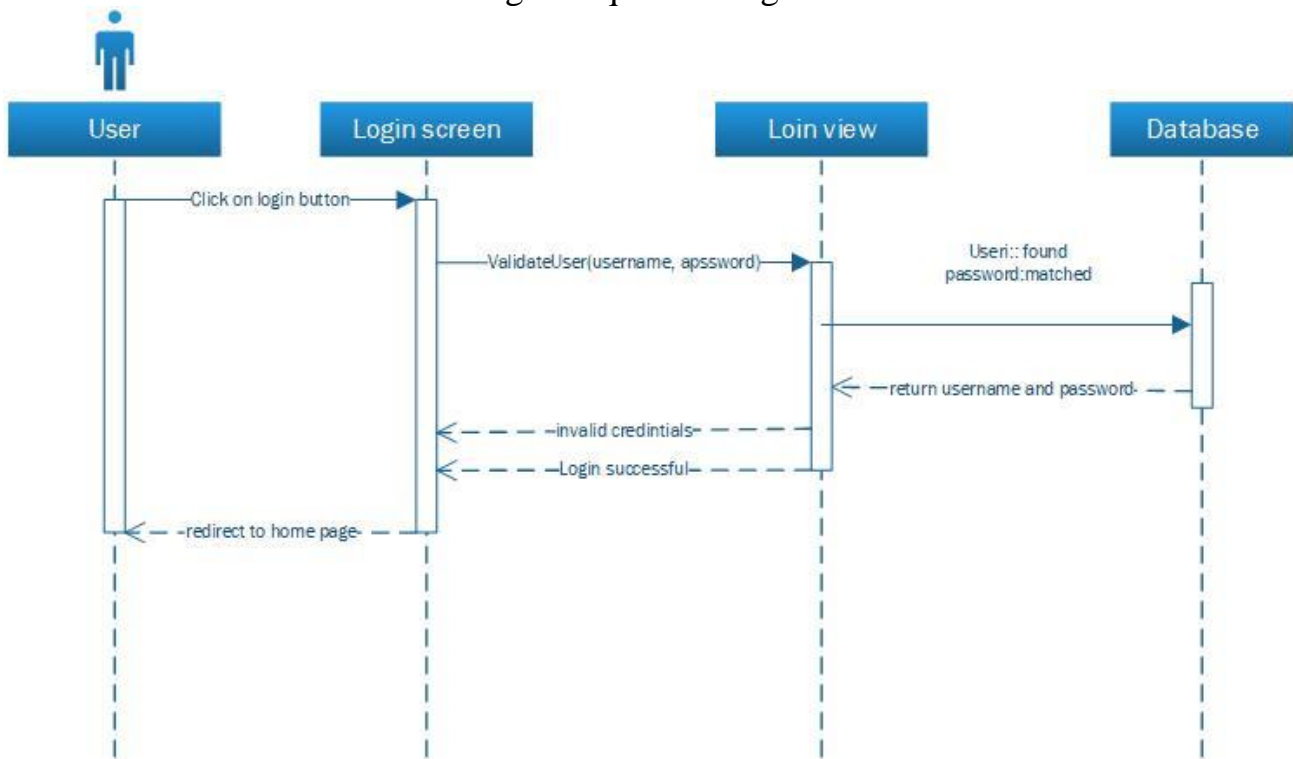
Figure 14: Sequence diagram for login

Logout Sequence diagram
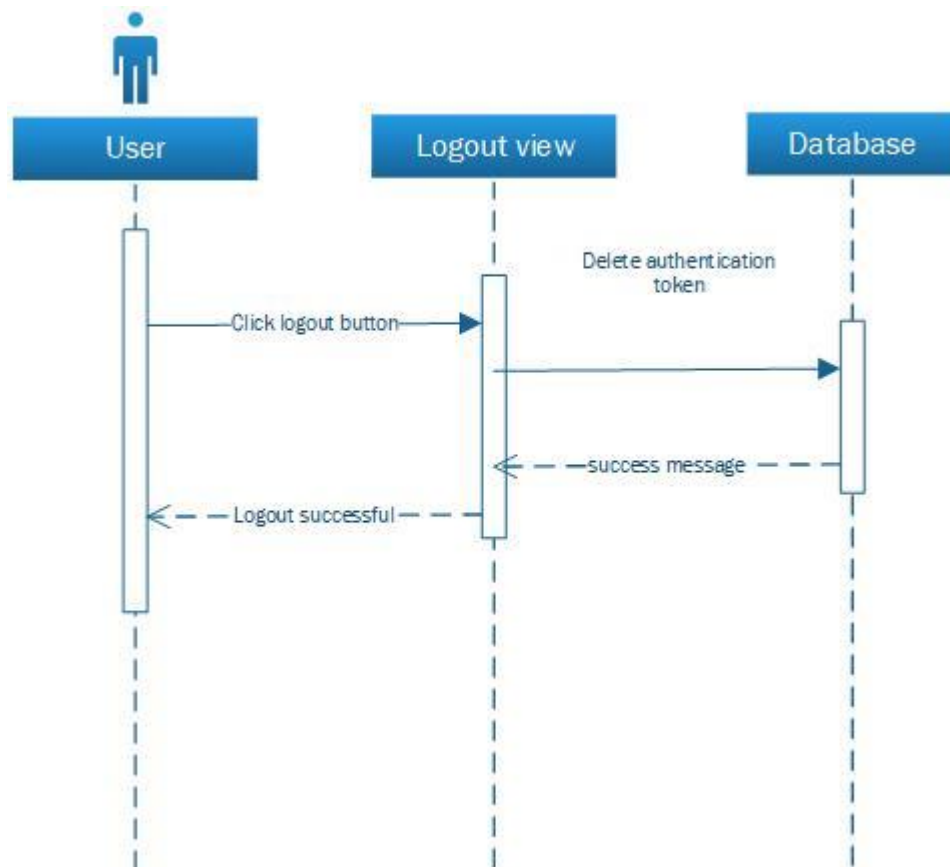
Figure 15: Logout sequence diagram
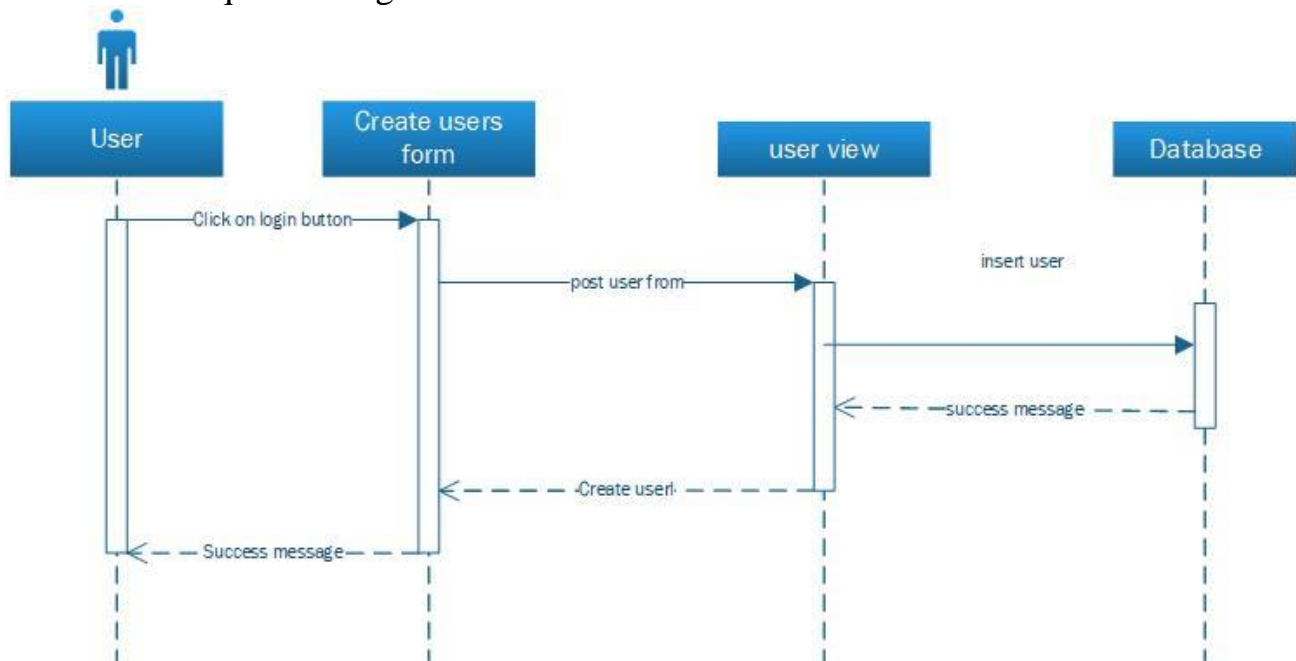
Create user sequence diagram

Figure 16: Create user sequence diagram
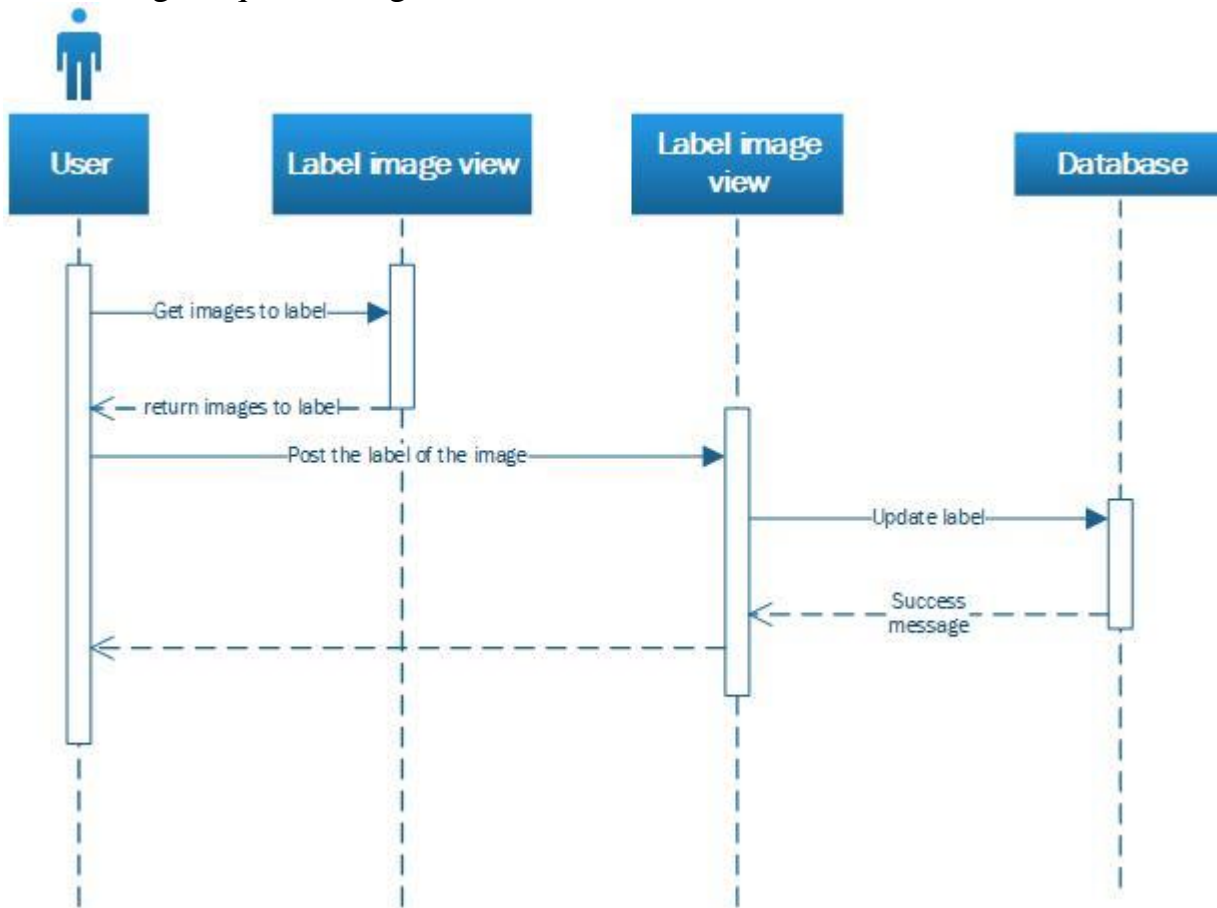
Label image sequence diagram

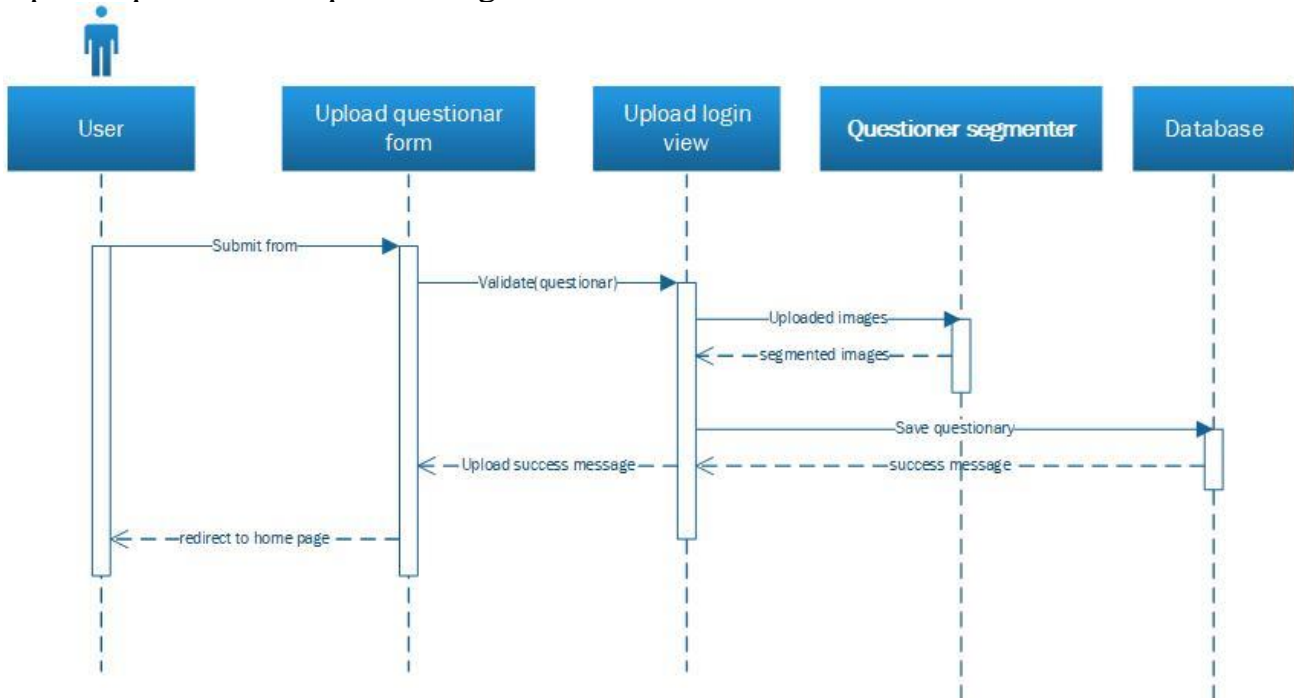**Figure 17:** Label images sequence diagram

Upload questioner sequence diagram

Figure 18: Upload questioner sequence diagram

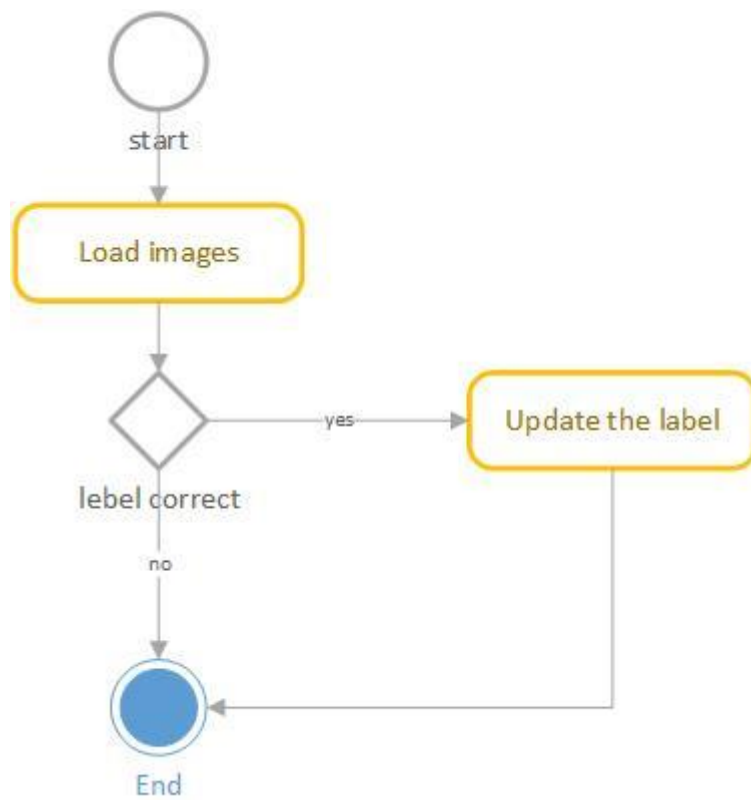### 3.11.4. Activity Diagram

The following activity diagram shows how agents label images.

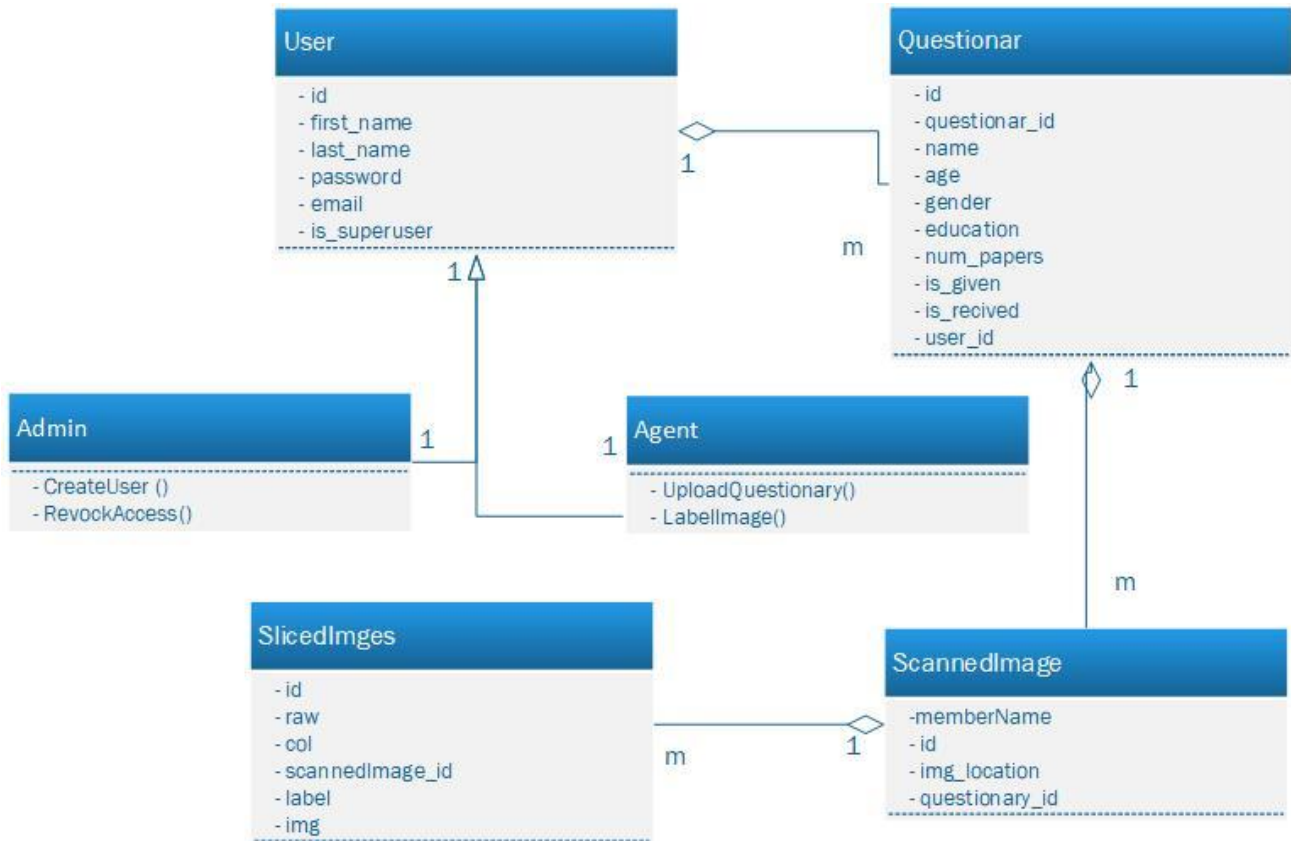**Figure 19:** Figure 16: Label image sequence diagram

3.11.5. Analysis Class Diagram

Figure 20: Analysis class diagram
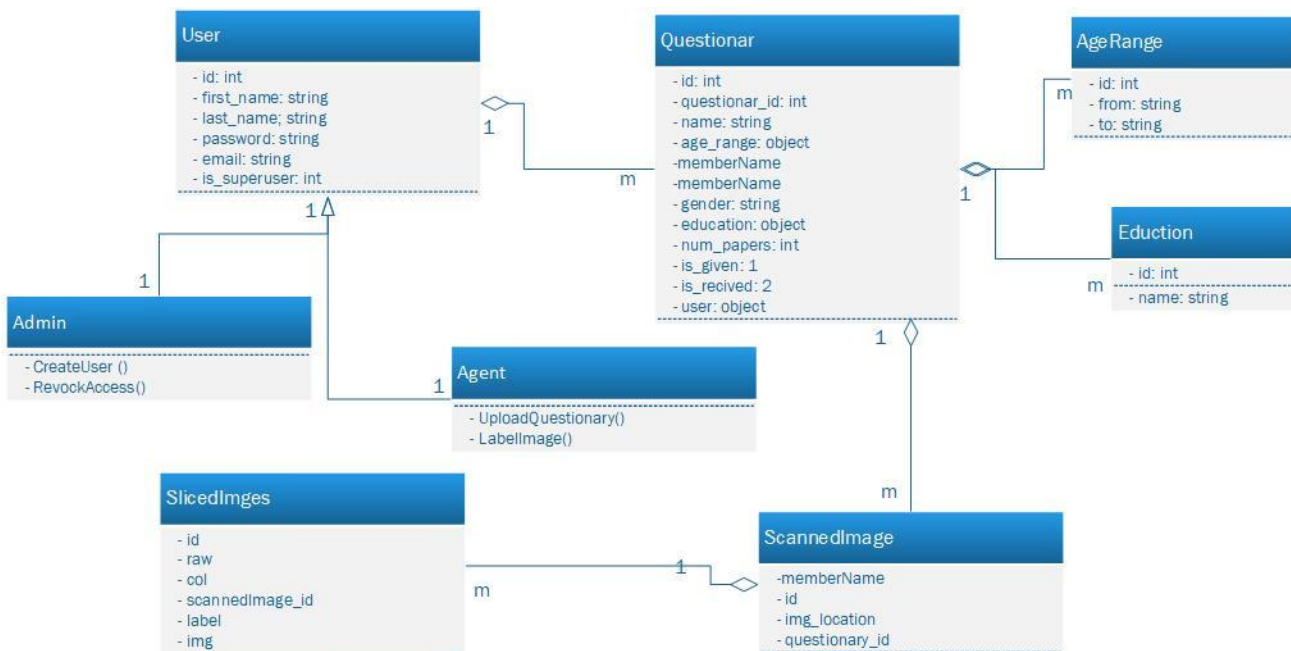
3.12. System Design

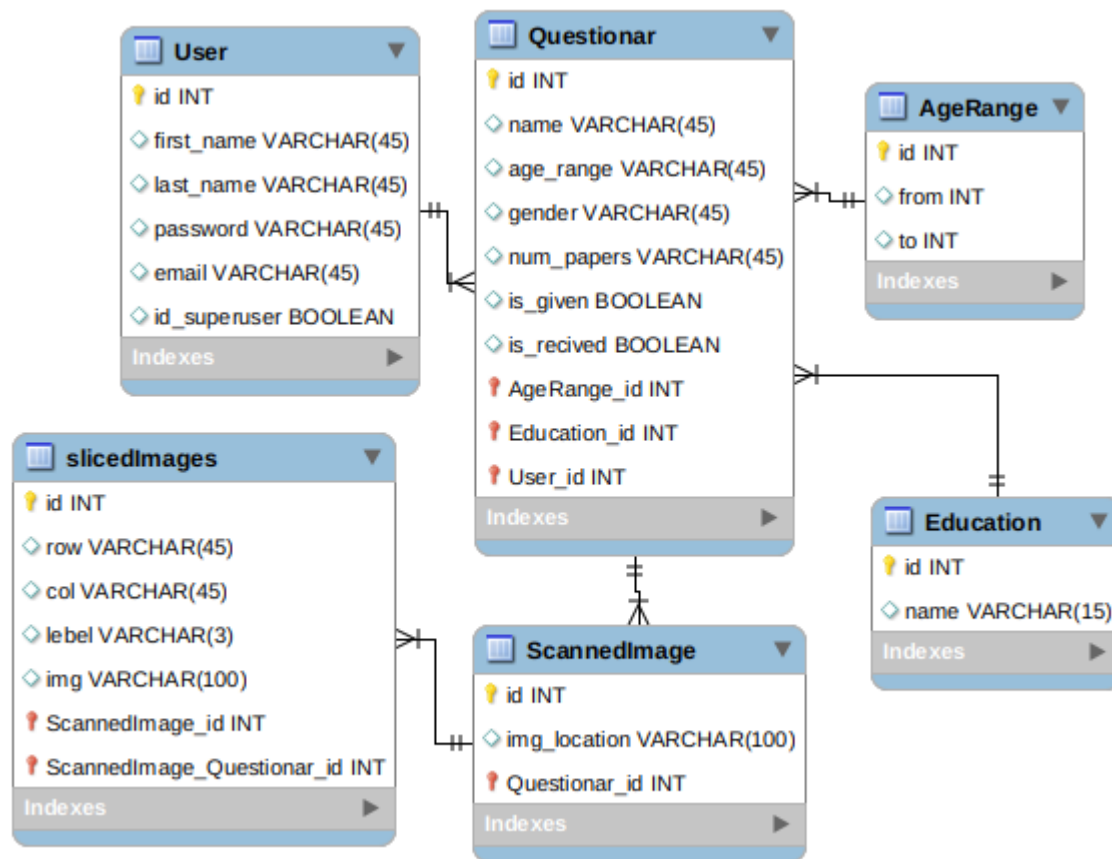### 3.12.1. Design Class Diagram

3.12.2. Physical Data Model

Figure 22: Physical data model

3.12.3. User Interface Design

Figure 23: Login interface

Upload questionar

Name

Age range   Gender   Education

ID   Number of papers

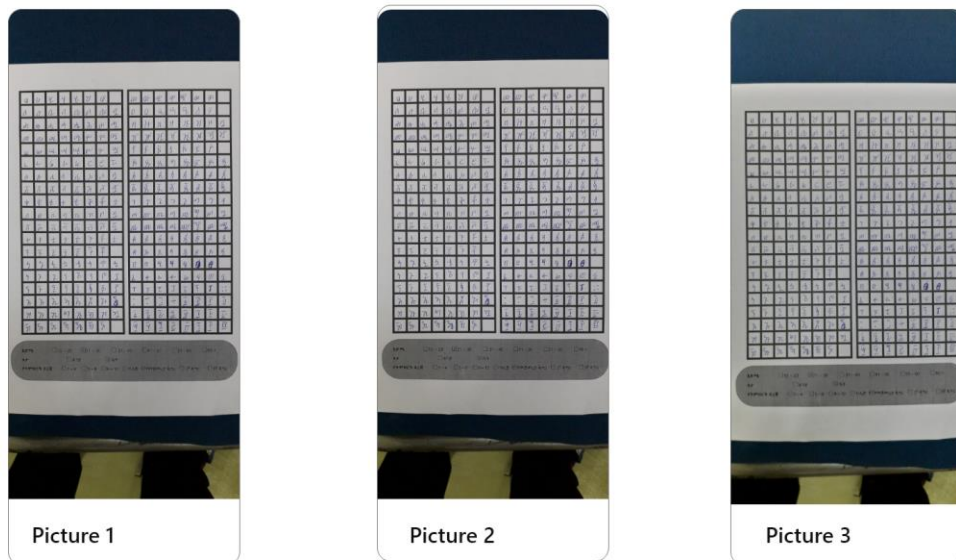Dag & Drop Images

Upload

Figure 24: Upload questioner form

Picture 1   Picture 2   Picture 3

Figure 25: Questioner images list

Figure 26: Questionnaires list interface design

3.12.4. Deployment Design
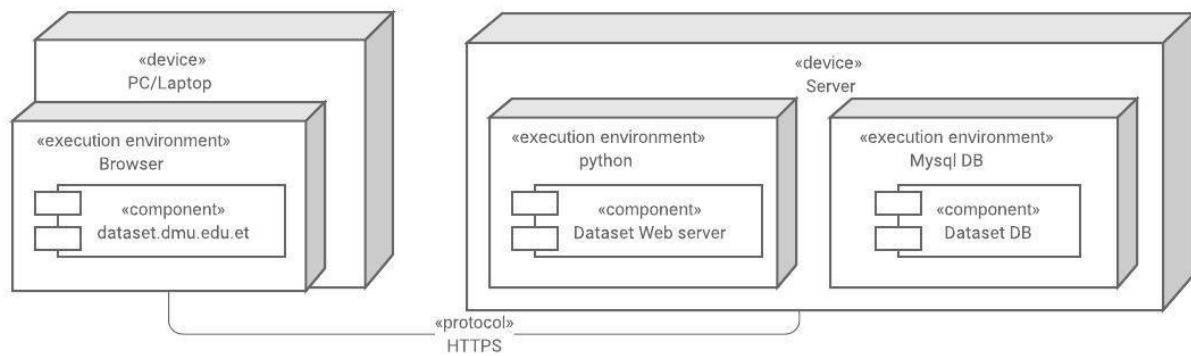
Figure 27: Display design diagram

Appendices

የካህሳ አንታፎ

ይህ መጠይቅ ለምርጃሙራት ዳግም ፕሮግራም ማጠየቃ ፕሮጀክት የተዘጋጀዉ። እባክ በትዛዙ መሰረት በጥንቃቄ ይሞላሉት።

1. የተሰጠዉን ጽሁፍ ከታች ባለዉ ሳጥን ዉስጥ ደግመዉ ይጻፉ።   006

በሄዉት መንንደ ላይ ፤ በማያት ዉጥንጥ
በማማን በማጣት ፤ በዚሁ ሄደን ዉሰጥ

ላይዋፉ ይሞኳል ፤ መዋርን ሳዋቀን
በለተፉጣ ም5ም ፤ ለይፈልጉት

እሺ ካለ5 ጋ ፤ ደከኳን አትንፈ5
ባለዋር የቆየዉ ነ5 ፤ ዋሬ እንደ ማት አርገን

[boxed handwritten text:]
ለበፉን መነጋጋ ለ0 ፤ ለመሄድቀ ዋ5ዋ5
በሰ7ለ ፤ ለ0ገ01 ፤ በ56 ሄደ5 ዋ55
ጓ54ርሩ ጓማዋ ፤ መ51ሩ5 ጓዋቀን
ጊለፉ5ሩ ም5ም ፤ ገዘ ፈለ55
ጄ5 ጓል7 5 75 ፤ ዙ5ኳ 55555
ጓደ0 ዳ5ዋ ን5 ፤ ዋ0ላ 55ዋ5 ጸ011

2. የአማርኛ ፈደላት በተሰጠት ሳጥኖች ዉስጥ በተሰጠዉ ምሳሌ መሰረት በእጅጽ ሁፉ ይጻፉ። ፈደላቱ በሳጥኑ
መሃል ላይ ጥን5 ሳይነኩ ይጻፉ።

[Amharic fidel chart / handwritten grid of Ge'ez/Amharic syllabary characters]

Figure 28: Questioner first page

31 | P a g e

Figure 29: Questioner page 2 a

Figure 30: Questioner page 2b

Figure 31: Questioner page 2 c

እድሜ   □10 – 20   ☑21 – 30   □31 – 40   □41 – 50   □51 – 60   □60 +

ጾታ   ☑ወንድ   □ሴት

የትምህርት ደረጃ   □1 – 4   □5 – 8   □9 – 12   □ከላይ ☑የመጀመሪያ ዲግሪ   □2ኛ ዲግሪ   □3ኛ ዲግሪ

References

1. *Amharic character recognition using a fast signature based algorithm.* **Cowell, John and Fiaz Hussain.** s.l. : Proceedings on Seventh International Conference on Information Visualization, 2003. IV 2003., 2003.

2. *Offline handwritten Amharic word recognition.* **Yaregal Assabie a, Josef Bigun b.** 2019.

3. *Convolutional Neural Network Model for Arabic Handwritten Characters Recognition.* **Murtada Khalafallah Elbashir, Mohamed Elhafiz Mustafa.** 2018.

4. *Ethiopic Character Recognition Using Direction Field Tensor.* **Bigun, Yaregal Assabie and Josef.** 2006.

5. *Handwritten Amharic Character Recognition Using a Convolutional Neural Network.* **Mesay Samuel Gondere, Lars Schmidt, Abiot Sinamo Boltena, Hadi Samer Jomaa.** 2019.

6. *International Journal of Advanced Research in Computer Science and Software Engineering.* **Shaout, Omar Balola Ali and Adnan.**

7. *Off-Line Handwritten Character Recognition System Using Support Vector Machine.* **Gauri Katiyar1, Ankita Katiyar2, Shabana Mehfuz3.** 2017.

8. *Amharic Character Recognition using a Fast Signature Based Algorithm.* **Dr JOHN COWELL, Dr FIAZ HUSSAIN.** 2003.

9. *Building Fast and Compact Convolutional Neural Networks for Offline Handwritten Chinese Character Recognition.* **Xuefeng Xiaoa, Lianwen Jina,∗, Yafeng Yanga , Weixin Yanga, Jun Sunb , Tianhai Changa.** 2017.

10. *CNN based common approach to handwritten character recognition of multiple scripts.* **Sen Maitra, Durjoy, Bhattacharya, Ujjwal.** 2015.

11. *Amharic Character Recognition System For Printed Real-Life Documents.* **Birhanu, Abay Teshager.** 2010.