T.C DOĞUŞ ÜNİVERSİTESİ

ENGINEERING FACULTY
COMPUTER ENGINEERING

Optical Character Recognition Application for Arabic language Using image processing technology

COME 492 GRADUATION PROJECT

PREPARED BY

MOHAMAD KARBEJHA
20180301125

MARWAN ELSABIE
20180301093

IMAD AL KHAWAM
20180301093

ADVISOR

Asst.Prof. ARİF MURAT YAĞCI

İSTANBUL, JUNE 2022

PREFACE

First of all, we would like to thank our advisor Asst.Prof. ARİF MURAT YAĞCI for his assistance during the process of making this project and for his valuable guidance.

We would like also to thank our instructors of our department for the precious knowledge that they provided for us during the past last 4 years.

Finally, we would like to thank our families and friends for their support during our university years.

İstanbul, JUNE 2022

MOHAMAD KARBEJHA

MARWAN ELSABIE

IMAD KAWAHM

SUMMARY

With more than 313 Million speaker around the world and being the fifth most spoken language in the world, Arabic language indeed is popular, but unfortunately Arabic is still falling behind in the world of technology, therefore being native speakers of the language , we have decided to base our project mainly on developing an application which supports Arabic OCR , the application main idea is to capture Arabic words from images as input and transliterate to Latin characters as output, this shall be done using image processing techniques which will process the input image and segment each letter using a segmentation algorithm we designed after that we obtain individual sperate letters in the final step a neural network model is used to predict the letter and give the transliterated Latin character .

ÖZET

Dünyada en çok konuşulan 5. dil olan Arapçanın toplamda 313 milyondan fazla insan tarafından konuşuluyor,lakin oldukça popüler olmasına rağmen arapça dili teknoloji dünyasında geride kalıyor bu nedenle ana dili arapça olan öğrenciler olarak projemizi arapça dilini destekleyen bir OCR aplikasyonu yapmaya karar verdik.Aplikasyonun ana hedefi girdi olarak fotoğraf üzerinden arap alfabesi ile yazılan kelimeleri alıp latin alfabesine dönüştürmüş halini çıktı olarak vermek bunun için girdi olarak alınan görüntünden her harf için ayrı ayrı tasarlanmış olan segmentasyon algoritmaları ile segmente eden görüntü işleme teknolojilerini kullandık.Son adımda harfleri ayrı ayrı elde ettikten sonra harfleri tahmin etmesi ve transliterasyonlu latin harflerini çıktı vermesi için projemizde yapay sinir ağları kullandık.

Keywords:

- Recognition Rate
- Deep Learning
- Convolutional Neural Network
- Handwritten Digit
- Digit Recognition

CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS

ABBREVIATIONS

Py	PyCharm
OCR	Optical character recognition
Cv2	Open-Cv library
AR2LT	Arabic to Latin mobile application
NNM	Neural Network Module
API	Application Programming Interface

REQUIREMENTS ANALYSIS DOCUMENT

1. INTRODUCTION

1.1. Purpose of the system

The purpose of the system is to help the user obtain transliterated Latin characters from images containing Arabic words, the system will provide an accurate and fast transliteration and will only require from the user to give an input image which will be an Arabic word.

1.2. Scope of the system

The scope will include all the main functions and algorithms which process the input from the user, these functions are written using python and image processing libraries such as OpenCV, In addition to that the scope will include the mobile application AR2LT which will be built using Flutter programming language, The scope will not include the neural networks prediction model as we will use an externally built model.

1.3. Objectives and success criteria of the project

The objective of this project is to obtain correct and accurate transliteration of each character from the input image, moreover this will only be achieved if the input is preprocessed and segmented correctly , the project will be considered successful if the NNM can make a correct prediction on the letters and recognize each letter and its Latin equivlent

1.4. Definitions, acronyms, and abbreviations

Definitions, acronyms, and abbreviations are located at page ix.

1.5. References

References are located at section “REFERENCES” on page 9.

1.6. Overview

2. CURRENT SYSTEM

Normally system is working by converting the image containing Arabic letters to English letters through an external API (Mobile applications – Computer applications ... etc.). Google Translate is one application that uses OCR in all languages including Arabic but in some cases the accuracy of the OCR is not accurate enough which results in a wrongly translation.

The current system is using an external API for the OCR process in the beginning

3. PROPOSED SYSTEM

3.1. Overview

3.2. Functional requirements

- User upload or take an instant photo on the AR2LT application.
- Transform image into greyscale then thresholding will be applied to obtain a binary image.
- Segmenting the image using vertical histogram.
- Feature extraction on segmented image.
- Letter recognition by neural network.
- Printed letters in Latin will be shown for the user in the main AR2LT screen

3.3. Nonfunctional requirements

3.3.1. Usability

It is easy to use by user, since he will upload the desired image, and he will receive the result as a text.

3.3.2. Reliability

The application is reliable, no personal information will be provided from the user

3.3.3. Performance

We provide a clear and right letter from an image then its right equivalent in Lattin or English.

3.3.4. Supportability

Our project supports any android or iOS device .

3.3.5. Implementation

The system is coded using Python Open-CV library (On: PyCharm, Spyder, Google Collab).

For future mobile application Flutter will be used (On: Android studio).

3.3.6. Interface

User interface is simple and friendly where the user upload or take an image and receive the result as a text.

3.3.7. Packaging

Anyone can install and use the system, since there's no physical components then there will be no packaging required.

3.3.8. Legal

The source code is open for public use

3.4. System models

3.4.1. Scenarios

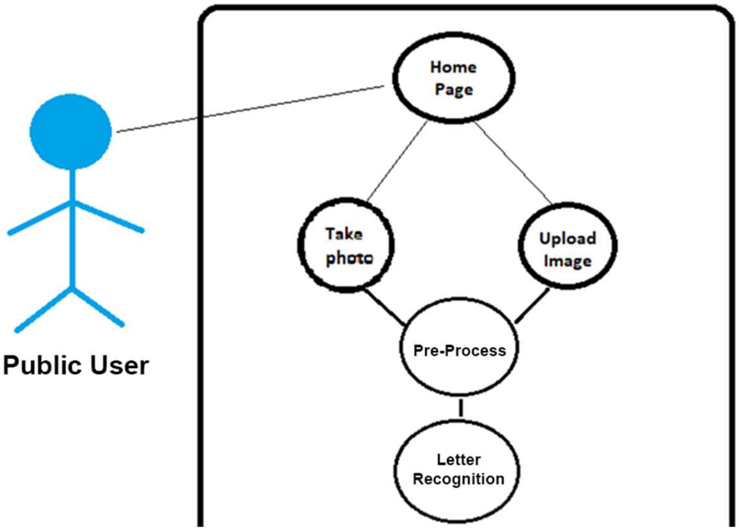
Description: User will access AR2LT mobile application then he will face two options:

1-Upload image

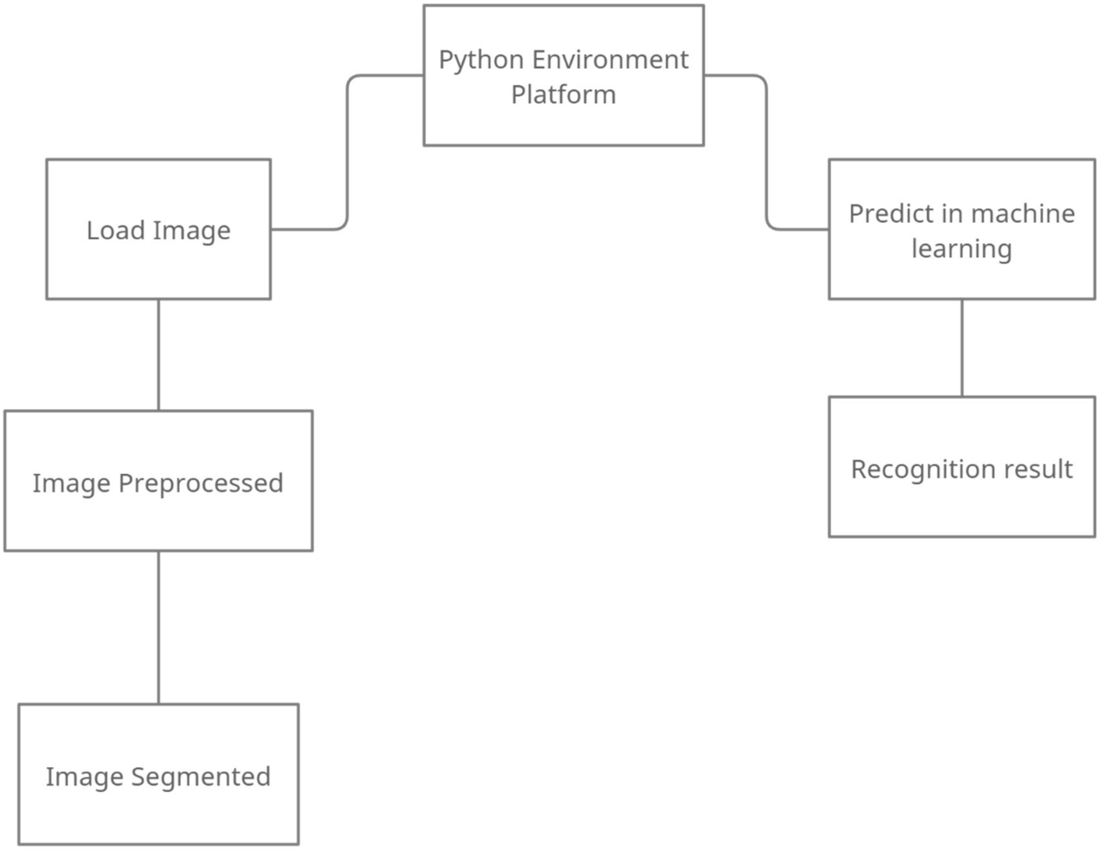
2- take photo

<table>
<tr><td colspan="2">Scenario 1: Upload image</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Action</td><td>• Select an image from mobile device.
• Preprocessing the selected image by applying some filters and prepare it to segmentation part.
• Segmenting the preprocessed image by vertical projection.
• Applying the segmented image to neural network which compares it with Arabic dataset of thousands of letters.
• Display the final result in few seconds.</td></tr>
<tr><td colspan="2">Scenario 2: Take photo</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Action</td><td>• Take an instant photo by mobile device (must be clear image as possible).
• Preprocessing the taken image by applying some filters and prepare it to segmentation part.
• Segmenting the preprocessed image by vertical projection.
• Applying the segmented image to neural network which compares it with Arabic dataset of thousands of letters.
• Display the final result in few seconds</td></tr>
</table>

3.4.2. Use_Case_model



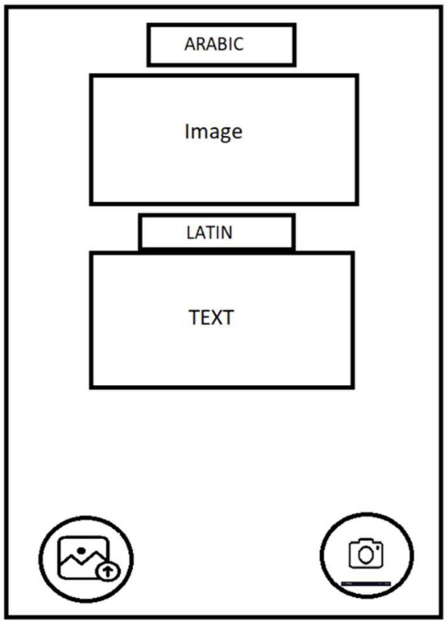
3.4.3. Object model

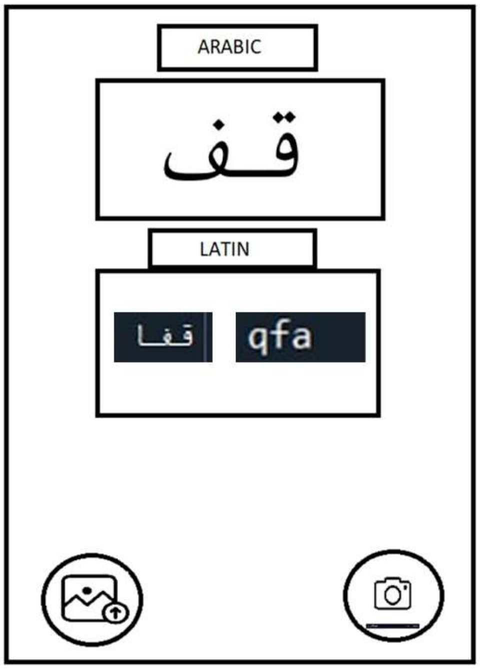


3.4.4. Dynamic model

3.4.5. Database model — ER diagram

3.4.6. User interface—navigational paths and screen mock-ups





4. GLOSSARY

1. **PyCharm – Spyder:**
   A python environment which contains a wide range of essential tools and libraries for python developers where they can use them to create applications, web etc... [1]
2. **Libraries:**
   OpenCV (Open-Source Computer Vision Library):
   Python library named as open-source computer vision library helps to provide common infrastructure for computer vision applications such as face recognition, radars, cameras [5]
3. **Preprocessing:**

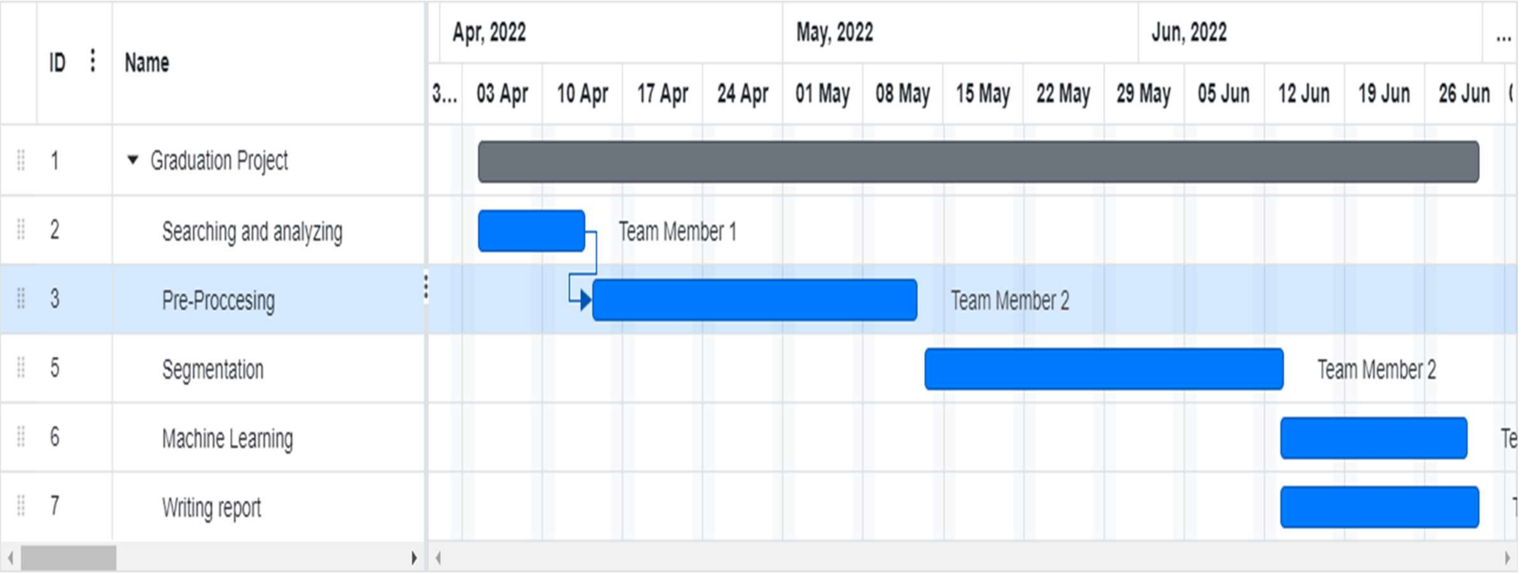
   Is the first step of the OCR application in preprocessing the image will be cleared from all the noise and corrected to be then read by the machine model

4. **Segmentation:**
   The process of splitting images into multiple layers, represented by a smart, pixel-wise mask is known as Image Segmentation. It involves merging, blocking, and separating an image from its integration level [6]
5. **Dataset:**
   A Dataset is a collection of relative data gathered and stored together [7]
6. **Neural Network:**
   A neural network is a set of nodes connected with each other to compute data and find relations between them, similar to human's brains [10]
7. **OCR (Optical Character Recognition):**
   It is a technology that recognizes text within a digital image. It is commonly used to recognize text in scanned documents and images. OCR software can be used to convert a physical paper document, or an image into an accessible electronic version with text. [9]
8. **CNN (Convolutional Neural Network):**
   a Deep Learning algorithm specially designed for working with Images and videos. It takes images as inputs, extracts and learns the features of the image, and classifies them based on the learned features [12]
9. **Contours:**
   A line that joins all the point along the boundary of an image that have the same intensity [13]

ROJECT BUDGET AND TIME PLAN

5. INTRODUCTION

5.1. Gantt Chart



ID	Name
1	Graduation Project
2	Searching and analyzing
3	Pre-Proccesing
5	Segmentation
6	Machine Learning
7	Writing report

5.2. Project Budget

	Estimated			Actual
Expenses	**Period** **(One time or Recurrent)**	**Cost per Period**	**Total Cost**	**Total Cost**
Human resource	3 months	100$	100$	100$
License for PyCharm	0 Time	Free software	0	0
Google play publishing fee	1 Time		25$	25$
Apple Store Publishing fee	1 Year		100$	100$
		TOTAL BUDGET	225$	225$

5.3. Project Phase Distribution

In this section, calculate the percentage from Gantt chart.

Table 5.1 Project phase distribution

Planning & Analysis	Design	Implementation	Testing
%10	%30	%50	%10

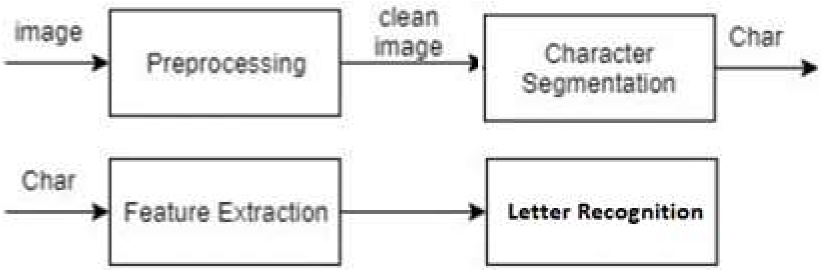
5.4. Other Tables

MEETING SUBJECT	ADVISOR NAME	DATE
Requirement Specification Meeting	Mustafa Zahid Gurbuz	25/03/2022
Progress meeting	Murat Arif Yagci	12/04/2022
Code Review	Murat Arif Yagci	10/06/2022

PROJECT SPECIFIC CONTENT

6. INTRODUCTION

6.1. Project's Pipeline



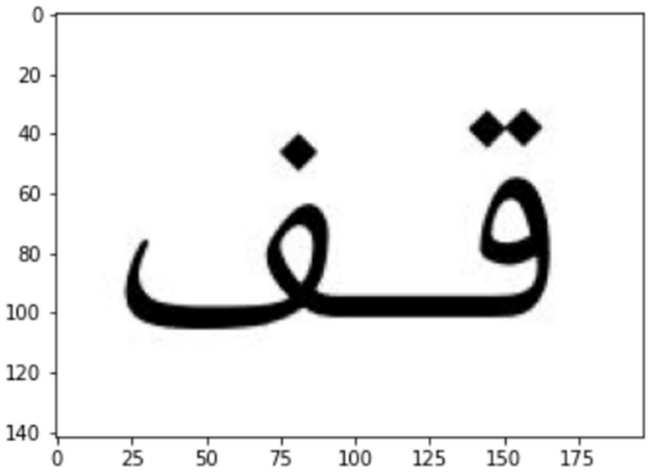
6.2. Technical Details:

The user can access to the service through Python environment (IDE)

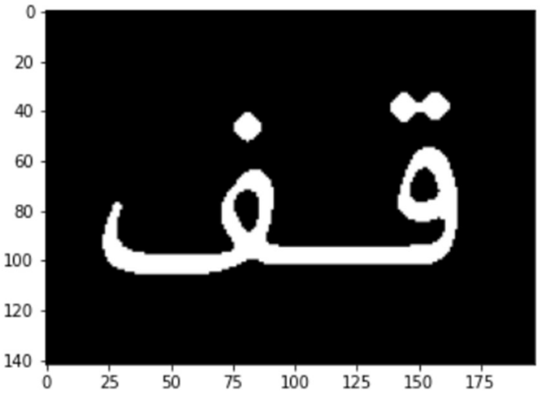
6.2.1. Preprocessing module:

Here we applying some filters to remove noise in the images and to prepare the image to be segmented correctly

- **Displaying original image:**

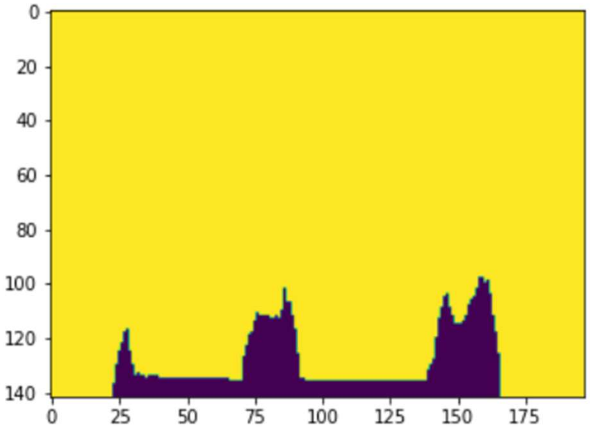


- we convert the image the image from RGB color space to **grayscale** space. It's a necessary step for the binarization which will be performed afterwards, transforming to grayscale is beneficial to represent the picture in black and white and make it easier for computation.
- **Converting to black and white image (Binarization**):

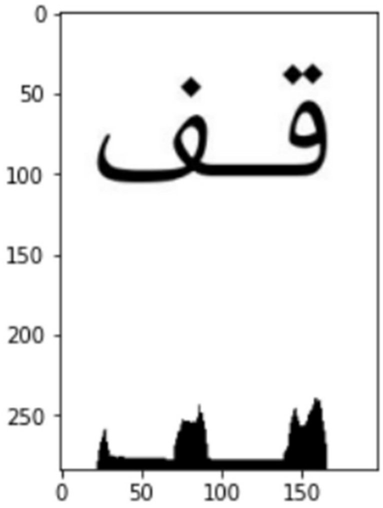


- After getting the grayscale image from the previous step , here a process known as binarization will be performed to obtain an image which contains only the foreground pixels and the background pixels , we are interested in foreground pixel as they represent the characters in which we will perform segmentation upon , to get the foreground pixel we perform an operation called thresholding , this works by setting a value for the threshold where pixel values are separated , pixel bigger or equal to the threshold will be set to 1 (white) and a pixel lower than this value will be set to zero (black) , final result will be an image which will be colored by black and white only.
- there exist many thresholding techniques like global thresholding and Otsu thresholding, these technique sets the same value for thresholding T upon all the values of the pixels in the image, this does not work in specific conditions where different lighting may affect the final result of the thresholding, adaptive thresholding will be used in our case for better results, adaptive thresholding works by computing the dynamic value T for thresholding by examining a segment of the image and finding the neighboring pixels value then calculating the threshold value in that area , this operation will be done on all the pixels in the image until the final binarized image is obtained. cv.adaptiveThreshold () is the function used to apply thresholding on the image. [11]

- **Histogram profiling:**

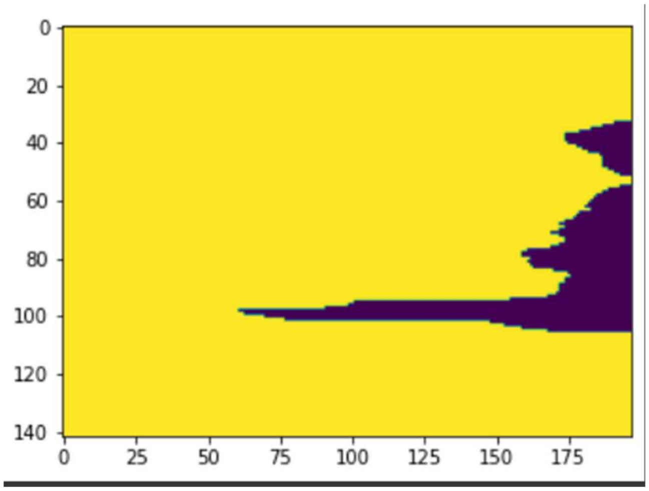


- Acquiring the histogram of the image is a critical step in the segmentation operation, it is the process of getting the sum of each pixel in the image in an individual column and row, in this step we get two histogram representations vertical and horizontal histograms each will help us to determine the points at which characters will be separated.[12]



- Vertical histogram with original image

Vertical axis represents the pixel intensity and horizontal axis represent the range of the pixels

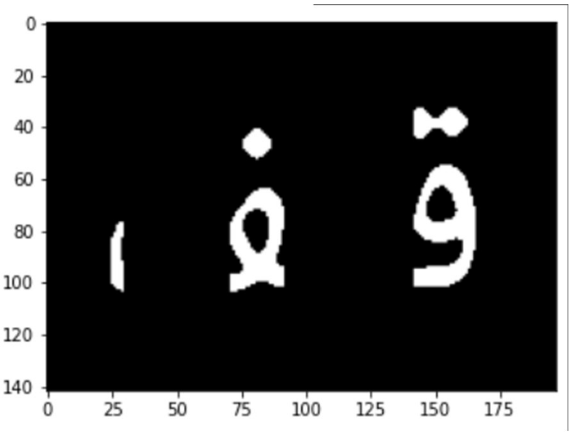


- Horizontal histogram

6.2.2. Segmentation:

In the segmentation procedure an algorithm will be performed on the binarized image

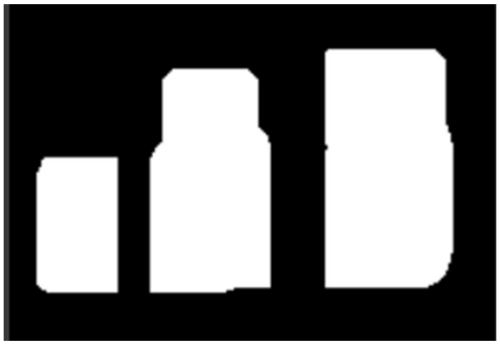
- Firstly the algorithm will scan the histogram vertical representation of the image the algorithm will scan each row of the variable Vertical_px using a for loop , this variable is an array of n number of rows and m number of columns , each row contains the sum of all the individual columns in the binarized image matrix array , after scanning and determining the minimum pixel value which are the areas where the letters should be separated , the algorithm will draw a line on the image at that point using a simple cv2.line() function arguments for this function will be the binarized image and the starting and ending point to represent the cutting points.[4]



```
1. For the sum of each column in the binarized image
2. If the value of the column <= Threshold value
3. Draw a line
4. Else
5. Repeat step 2
6. End if
```

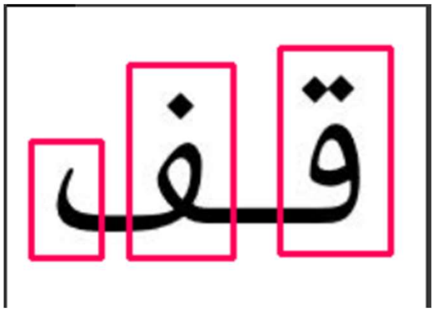
- Some drawbacks of this algorithm are that in some cases the individual letter will get segmented into two separate letters, since the algorithm only works by segmenting on the vertical projection the cases where the letter is written in a form where it has a horizontal line connecting the letter and this line's vertical value sum is less than threshold value, the algorithm will draw a line and cut , to result in two different letters.

6.2.3 Morphological operation:

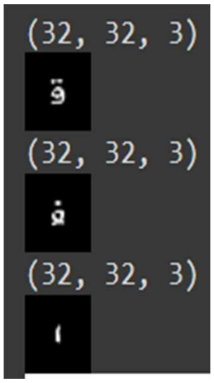


- In this step a dilation operation will be performed on the segmented letters to connect the neighboring pixels and to close the gaps between them thus making it easier for the system to recognize the clear structure of each letter, this is done by convolving a kernel of fixed sized five by five with the binarized image.

6.2.4 Drawing contours:



- In the last step of the image processing part a bounding box will be drawn over each individual letter, this is done by using OpenCV cv.findContours(), the argument for this function will be the dilated image we got in the previous step and the retravel mode and the approximation mode, for the approximation mode the cv2.CHAIN_APPROX_SIMPLE mode is used in our program, afterwards the coordinates of the contours where the points are connected will be sorted from right to left to match the Arabic way of writing via the sorted() function, after sorting we will pass the coordinates to a function called cv2.boundingRect(); this function will take the coordinates of the contours and draw a box around them which will result in multiple boxes drawn over the letters. [13]
- In the last step of Drawing contours, the program will loop through each box and resize it to (32*32) size and draw some border around it so it fits in our NNM then it will output each letter and save it to a file.

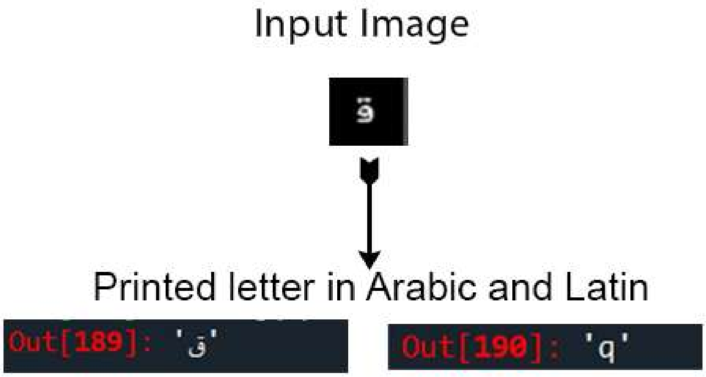


```
(32, 32, 3)

(32, 32, 3)

(32, 32, 3)
```

6.2.5 Feature extraction using Convolutional neural network (CNN):

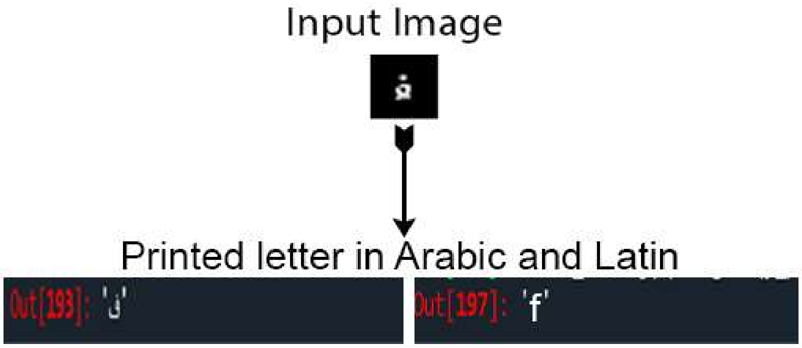
- In the final part of this project we have decided to use a premade neural network model from the internet which works on a Arabic handwritten letters dataset , the model is built on the concept of CNN , simply this neural network model is a popular model used for image classification problems , the model consists of multiple layers where in each layer a small segment of the input image is convolved with a different kernel each time to highlight the feature of an image, this processes of convolving will be performed many times over each letter in the dataset , in our case the model is trained on the dataset that we got , after training the model over the whole the 28 Arabic handwritten letters with a good accuracy , the model will be saved and used later to do predictions on the letters that we have obtained from the segmentation part , if the prediction is done correctly the letter and its Latin equivalent will be outputted and shown. [10]

- After making a prediction using keras predict () method the output of the model will be the letters that we provided as input and its Latin equivalent



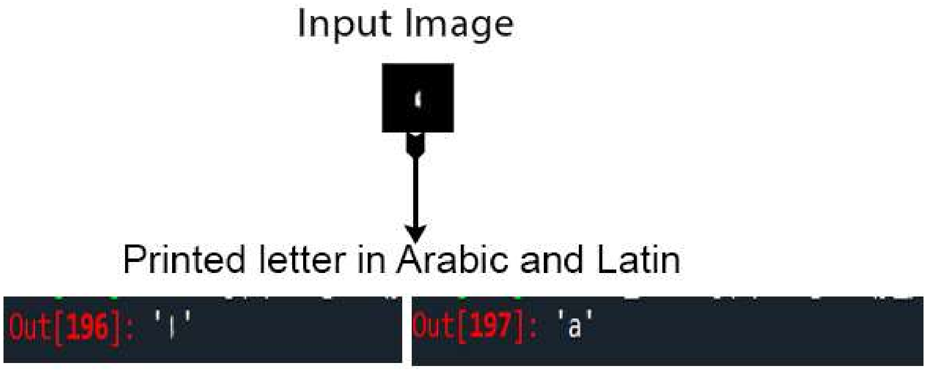
```
Out[189]: 'ق'
```

```
Out[190]: 'q'
```



```
Out[193]: 'ف'
```

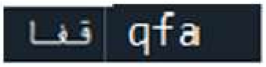
```
Out[197]: 'f'
```



```
Out[196]: 'ا'
```

```
Out[197]: 'a'
```

3- The word in Arabic and The equivlent in latin



```
قفا qfa
```

6.3.USER MANUAL

Here is the instruction of how to use this algorithm, the user is going to upload the desired image in the images folder then write its path in imread function.

Then after that, the program is going to do everything begins with preprocessing, segmenting, predicting from Arabic dataset set then finally display the Arabic letter and its equivalent to English letter.

6.4.TEST PLAN

The project has been tested in 3 different laptops, Future mobile application will be tested on android and ios devices to insure stability

6.5. MAINTENANCE PLAN

We planning to maintain our project by keep working on it and develop it after graduation to extend it to predict not just letters but also to predict sentences then paragraphs.

CONCLUSION AND DISCUSSION

The program is a simple OCR application where Arabic letters get extracted from an image and processed then a NNM is used to recognize the Arabic letter after it gets procced accordingly, the final result will be the printed Latin letters, although the implantation in this project is successful in general cases, there still exists some cases where the used algorithm is not applicable on all the letters, different shapes of Arabic letters require a specific segmentation algorithm, in future versions of this program each case where characters have a certain shape an algorithm written for that case will be used in the program.

The current final version of the program has a big room for future improvement, future versions include a more enhanced segmentation algorithm for specific cases where basic segmentation fails to get a good result, furthermore a designed machine model for auto correction for words could be implemented in the future versions of the application to help improve the accuracy of the output result.

Finally, we would like to thank Arif hoca for his guidance and help during the time which we worked on this project, this was indeed important project in our careers as students and Arif hoca made sure to guide us on the correct path to become successful engineers

Our future plans will be mainly in the field of data science and in the technology of OCR, we plan to explore this field and contribute to it with the knowledge we got from this project.

REFERENCES

1. **https://www.jetbrains.com/help/pycharm/quick-start-guide.html [Accessed on 03.07.2022]**

2. **https://www.kaggle.com/datasets/mloey1/ahcd1/code [Accessed on 12.06.2022]**

3. **https://opencv.org/about/ [Accessed on 2022.06.21]**

4. **Aziz Qaroush, Abdalkarim Awad, Mohammad Modallal, Malik Ziq, Segmentation-based, omnifont printed Arabic character recognition without font identification ,Journal of King Saud University,20 June 2022 https://www.sciencedirect.com/science/article/pii/S131915782030481X [2022.07.11]**

5. **https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/ [Accessed on 2022.06.15]**

6. **https://www.geeksforgeeks.org/image-segmentation-using-pythons-scikit-image-module/ [Accessed on 2022.06.30]**

7. **https://www.educba.com/dataset-in-python/ [Accessed on 2022.06.18]**

8. **https://realpython.com/python-ai-neural-network/ [Accessed on 2022.06.21]**

9. **https://nanonets.com/blog/ocr-with-tesseract/ [Accessed on 2022.05.25]**

10. **https://www.analyticsvidhya.com/blog/2021/08/beginners-guide-to-convolutional-neural-network-with-implementation-in-python/ [Accessed on 2022.06.30]**

11. **https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html [Accessed on 2022.06.10]**

12. **Abdelhay Zoizou, Arsalane Zarghili, Ilham Chaker,**

    A new hybrid method for Arabic multi-font text segmentation, and a reference corpus construction, Journal of King Saud University - Computer and Information Sciences,Volume 32, Issue 5,2020

    https://www.sciencedirect.com/science/article/pii/S1319157818301769#f0015

    [2022.07.04]

13. **https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/ [Accessed on 2022.06.30]**

APPENDIX

BIOGRAPHY

Photo	Name	Bio
	MOHAMAD KARBEJHA	*Computer Engineering student. *Working IT in full time. *29 years old * Syrian
	MARWAN ELSABIE	A computer engineering student 22 years old Egyptian
	IMAD AL KHAWAM	A 4th year computer engineering enthusiastic about data science and crypto currency Technolgies *24 years old