UTM

UNIVERSITI TEKNOLOGI MALAYSIA

FUNDAMENTAL OF IMAGE PROCESSING

(SECV3213)

PROJECT REPORT

LECTURER NAME

Lecturer: DR. MD SAH SALAM

Group Members:

No	Name	Matric No
1.	IRMA ZAFIRAH BINTI MOHD IKRAM	A19EC0054
2.	NURUL SYAMIRA BINTI AMAT JIFRI	A19EC0145
3.	NOOR ARINIE BINTI NORHALIL	A19EC0121

TABLE OF CONTENTS

Introduction

In real life, datasets that are collected are in large amounts. To understand these datasets, preprocessing is needed. Preprocessing data is a common first step in deep learning to prepare raw data in a format that the network can accept. This project focuses on classifications of object patterns and preprocessing of the images for features extraction. There are various operations that can be used for image processing such as spatial domain filtering, frequency domain filtering, segmentation and morphology.

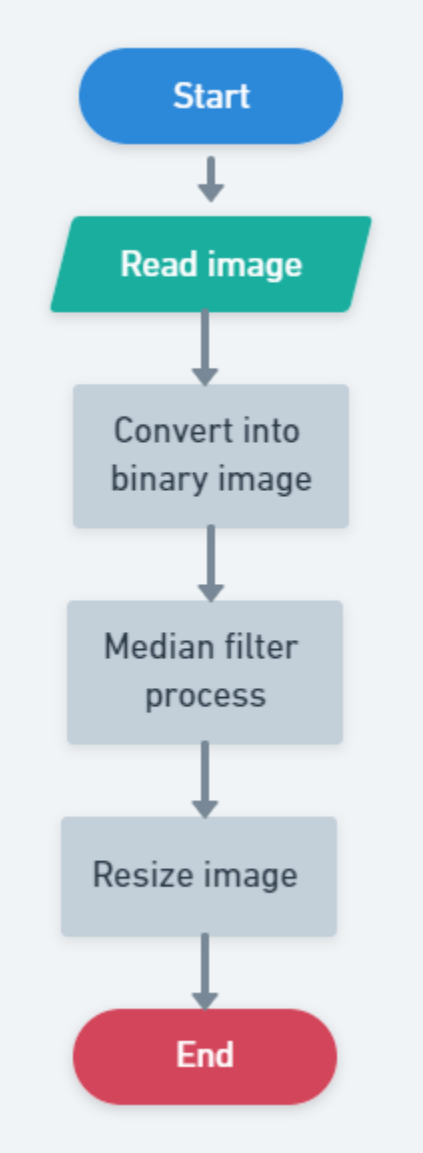
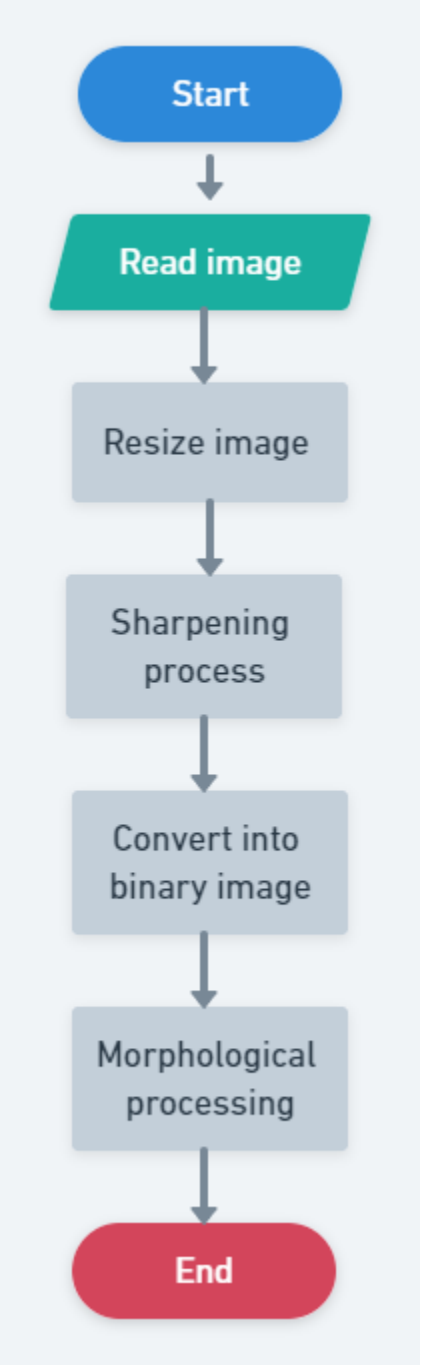
Experimental Data Used

Our group chose Arabic alphabet detection that will train and test six different patterns of experimental data which are alef (أ), beh (ب), dal (د), feh (ف), hah (ح) and waw(و). The total dataset for the test that we used is 3000 sets while for the train is 12000 sets. Dataset for alef (أ) is 308, beh (ب) is 480, dal (د) is 480, feh (ف) is 480, hah (ح) is 480 and waw(و) is 480.

Problem of the Data

Based on the dataset that we chose we found out that it has a few problems. Firstly, the size of the dataset is too small which is 32x32 pixels. This small size may result in loss of quality when proceeding to the next pre-processing so, we resized the dataset to 100x100 pixels. Secondly, the dataset is not sharp and looks blurry also some of the edges are missing. To make it sharper, we apply a laplacian filter. Thirdly, the problem is that the original dataset is not in binary image. With thresholding, we are able to convert the dataset into binary. After converting the dataset into a binary image, some parts of the image are not visible. Hence, morphology dilation is used.

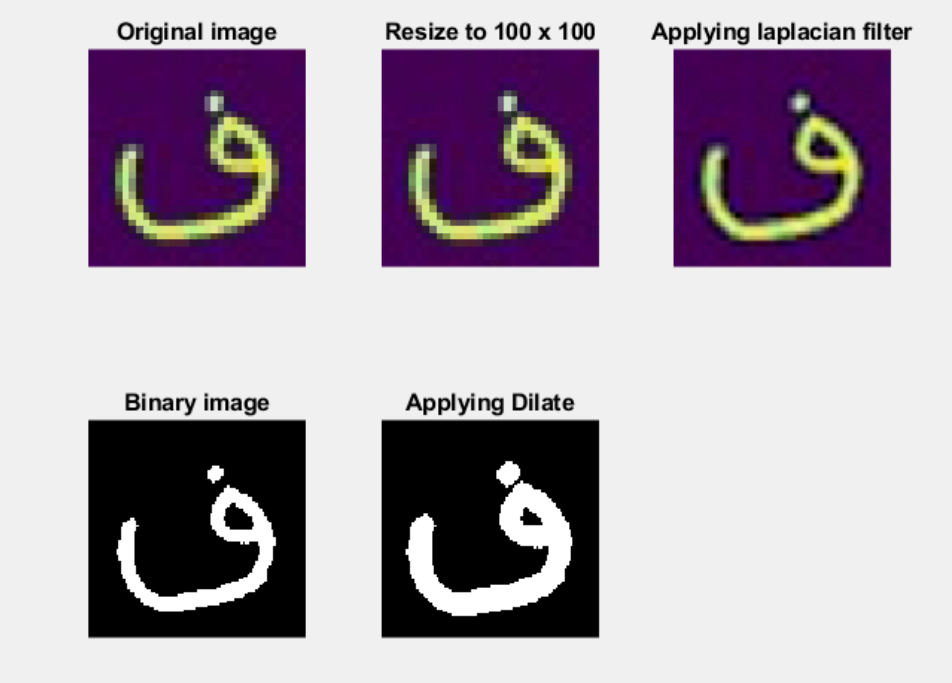
Pipeline of the Image Enhancement

Benchmark Pipeline of Image Enhancement	Improvement Pipeline of Image Enhancement
	

For the benchmark pipeline image enhancement it applies three image processing. The first step is converting the original image into binary because when we convert into binary it allows easy separation of an object from the background. Next, continue with applying the median filter process to remove noise from the images. Last step is to resize that image into 100x100 pixels.

For the improvement pipeline image enhancement we use different image processing and there are four processes that we applied which are resizing, sharpening, converting image into binary and morphological processing.

Qualitative Result



Example of one data being pre-processed.

Based on the figure above, we can see the difference made by each process. As stated before, the pre-processes included for the improvement are resizing, sharpening, thresholding into binary and morphology dilation.

The first pre-process used is resizing. After reading the image we change the original image with size of 32x32 pixels to 100x100 pixels. Then, we improve the image by doing a sharpening process using the laplacian filter formula. Laplacian filter is an edge-sharpening filter, which sharpens the edges of the image. Other than that, we convert the image into a binary image so that it is easy to separate an object from the background. Last but not least, we implement the morphological processing to expand the image pixels by using structuring elements i.e. se = strel('disk', 2). Strel function is to create a flat structuring element. Besides, morphological dilation makes objects more visible and fills in small holes in objects in the image.

Quantitative Result

Benchmark Result

```
cnt =

    67    55    70    60    53    55
```

Result of classes for training.

```
end of training

cnt =

     9    10     5    12     9    15
```

Result of classes for testing.

```
wrong rate :

ans =

    6.6667


D =

     9     0     0     0     0     0
     0    10     0     0     0     0
     0     0     5     0     0     0
     0     0     1    10     0     1
     1     0     0     0     8     0
     0     0     1     0     0    14


ans =

     6     6

recognition rate is :
   93.3333
```

The result of recognition rate.

Our result

```
cnt =

   300     0     0     0     0     0
```

Result of classes for training.

```
end of training

cnt =

   100     0     0     0     0     0
```

Result of classes for testing.

```
wrong rate :

ans =

    92


D =

     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
    18    21    19    10    24     8


ans =

     6     6

recognition rate is :
     8
```

The result of recognition rate.

Based on the results above, we can conclude that our recognition rate is lower than the benchmark. For the benchmark result, the dataset can be classified into their own class which is six classes for both training and testing. The recognition rate is also high, which is 93.33. The dataset being trained is 360 meanwhile for the testing is 60.

Our result, on the other hand, the recognition rate is only 8 and the wrong rate is 92. Hence, the classification of the dataset cannot be performed due to very low recognition rate. The dataset being trained is 300 meanwhile for the testing dataset is 100. We also have six classes to be trained and tested.

Conclusion

In conclusion, this project is about deep learning whereby we have a set of data and classify them into a few classes by using MATLAB. In order to execute the training and testing of the dataset, we first have to perform pre-processing upon our dataset. This allows the computer to recognize our dataset and classify them into specific classes easily. After performing pre-processing, training and testing, we can see the result quantitatively and qualitatively. All in all, even though our project cannot recognize the dataset and classify them into specific classes, we have managed to apply pre-processing operations that we learnt in class such as morphology, laplacian filter, binary and resizing.

Video Demo

Video link:
https://drive.google.com/file/d/1cSnZh4ehFLe16RtNWrX2kJEBNPbdkXmH/view?usp=sharing

Project Script

Link:
https://drive.google.com/file/d/11gefJXKm0SNNNzs6M_AzjkT3CMHxrFgC/view?usp=sharing