PROJECT REPORT

on

ROBUST BINARIZATION ALGORITHM FOR REMOVING UNEVEN LIGHTING OF IMAGES

Submitted in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENIGNEERING

by

SHREYAS DHULIYA
(Reg.no 1031010286)

Under the guidance of

Ms.R.Jeya
(Assistant Professor(OG), Department of Computer Science and Engineering)

SRM UNIVERSITY
(Under section 3 of UGC Act 1956)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
SRM UNIVERSITY
(Under section 3 of UGC Act, 1956)
SRM Nagar, Kattankulathur- 603203
Kancheepuram District

APRIL 2014

BONAFIDE CERTIFICATE

Certified that this project report titled **"ROBUST BINARIZATION ALGORITM FOR REMOVING UNEVEN LIGHTING IMAGES"** is the bonafide work of Shreyas Dhuliya (1031010286) who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion or any other candidate.

Signature of the Guide	**Signature of the HOD**
Ms.R.Jeya	**Dr. E. POOVAMMAL**
Asst. Professor(O.G)	Professor& Head
Department of Computer Science and Engineering	Department of Computer Science and Engineering
SRM University	SRM University
Kattankulathur- 603203	Kattankulathur- 603203

Submitted for Project Work Viva-voce Examination held on

____

Place :Kattankulathur

Date:

INTERNAL EXAMINER **EXTERNALEXAMINER**

ABSTRACT

The weight of the computation has shifted from client to the server side. Invasion of cloud computing has lead to machines that are dedicated to performing certain tasks. A system is proposed where image processing is done on the server instead of on the client's machine. The server will have a code which binarizes the image by separating the image into different regions called the text, near text and the background region. A local server is chosen to be created by using a personal computer. As Linux is the most secured operating system , it is chosen to be on the server side. The browser will give options to upload and download the image so that the client can use it to make notes.

ACKNOWELEDGEMENT

We wish to express our sincere gratitude to our project guide **Ms R.Jeya**, Assistant Professor(O.G), SRM University, for guidance and encouragement in carrying out this project work. He has been there to enlighten us every step of the way and hence we are much obliged to him for being the most integral part in the successful completion of our project.

We also wish to express our gratitude to each of the **penal members** involved in the reviewing of this project, who rendered their help during the period of the project work and for their kind co-operation in its completion. It is through their remarkable assessment skill and helpful disposition that the project has been polished and perfected.

We humbly state our gratitude to **Dr. E. Poovammal,** Professor & Head**,** CSE Department, for availing us of the opportunity to showcase our skill and prepare ourselves for our future in the technical world.We are also deeply obliged to **Mr.K.Senthil**, the project coordinator for ensuring that we are able to complete our project and for always being patient and understanding.

Our gratitude to our esteemed college **SRM University**, Kattankulathur, Chennai for providing us the chance to undergo such a learning experience and for imparting the chance to delve deep into the technical aspects of computer science so as to be able to practically implement the knowledge gained through this project.

Last but not least we wish to avail ourselves of this opportunity to express a sense of gratitude and love to our friends and our beloved parents for giving us support and strength to complete the project in its polished form.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATION

S.NO	ABBREVATION	EXPANSION
1.	BR	Background region
2.	NTR	Near text region
3.	TR	Text region

CHAPTER 1

INTRODUCTION

1.1General

The project aims to build a code which separates the text region from the background region and the near next region and binarize the image. The input to the code are images taken by the camera or mobile cameras of the books and the notes of the class. The output of the code will be a binarized image where the background is eliminated and only the text is displayed.

This code is saved in a local server which the client can use by uploading an image and downloading the resulting image of the code.

1.2 Motivation

As the technology Is advancing people try to find more shortcuts so that their work can be done in a short period of time. We students have to make notes and gather notes from various sources from books , teacher’s notes and many more. We often have to go to a Xerox machine to get our work done.

The money spent to take Xerox is although very less but when we need to collect extensive notes it is costly.We have to go and generally wait in a que to get our Xerox done and when the material is more than a hundred pages the shopkeeper asks us to collect he next day.

To save money and time I wanted to develop a code so that we can take pictures from the books or notes through our camera(mostly mobile phones) and get it binarized eliminating the background and the mid tone in the picture with just one click.

The code should eliminate hands , background and other unwanted pixels and display only the background as white and text as black.

1.3 Problem Description

Images of the books and notes taken by the camera has uneven lighting , strong mid tones , shadows at the edges and background such as bed sheet, table . The following code eliminates the uneven lighting condition and gives a perfect binarized image which is easily readable and can be used to make computer notes .

1.4 Main modules

The project is divided into two modules ,they are

1. The code which is written in opencv which will binarize the image by only taking the image as the parameter.
2. The local server which will provide pages to upload, process the image and upload the image for the client to download.

CHAPTER 2

LITERATURE SURVEY

There are many thresholding techniques to binarize an image. For each of this thresholding techniques there is an algorithm which selects a threshold, from which the grey scale value of each pixel is compared and assigned value black or white.Many algorithms are common for binarizing which are Otsu[5] and W.Niblack[6]. These algorithms deals with binarization of document image but doesn’t give perfect results for uneven lighted document images.

B.Su, S.Lu , and C. L. Tan[2] tells about binarizing historical images, as the historical images are degraded in various ways such as uneven light, un even mid tone, uneven text grey scale value, this gives an idea of working with degraded images taken by the camera to prepare computer notes form text books and class notes copy. These images have uneven lighting, very dark images uneven text grey scale values and some unwanted background behind the sheet while taking photo.

Siva Rama Sastry Gumma[10] and G. Leedham, C. Yan, K. Takru,J.Hadi, N. Tan ,and Limian[3] have discussed techniques to separate the document image into various regions which are text region,near text region and background region.

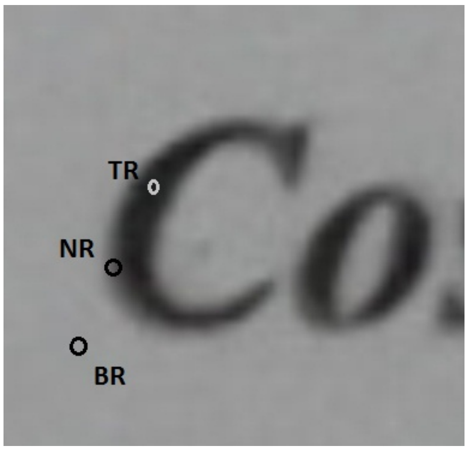


Figure 2.1: Regions

Wang Jianxin, Chen Songqiao, Jia Weijia, Pei Huiming[11],Wang Jianxin, Peng Bei, Jia Weijia[13] Keshou Wu, Wuhan, Lijuan[4] mention about DIPPS which is designed based on Browser/server mode.The clients in the form of Java Applets embedded in the brower. The server mainly includes a web server to run java beans components. With the help of this idea create a local server using php files .The server will have the code in it to process the image. The clients can use the local area network to connect with the server and uploaded the image using a form and download the processed image later.

CHAPTER 3

SYSTEM ANALYSIS

3.1Existing Algorithms

Comparing the results of the existing algorithms and the proposed algorithms.

3.1.1Otsu’s Algorithm

Otsu’s algorithm is used to perform clustering based image thresholding,this is an automated process or the reduction of a grayscale image into binary image. The algorithm assumes that the image to be thresholded contains two classes of pixels they are foreground and background. Then it calculates the optimum threshold separating those two classes so that their combined spread i.e intra class variance is minimal. Based on a very simple idea: Finds the threshold that minimizes the weighted within-class variance. Operates directly on the gray level histogram [e.g. 256 numbers, P(i)], so it’s fast once the histogram is completed. Histogram (and the image) are bimodal. There is no use of spatial coherence, nor any other notion of object structure. It Assumes stationary statistics, but can be modified to be locally adaptive. It Assumes uniform illumination implicitly, so the bimodal brightness behavior arises from object appearance differences only.The class variences are calculated as:

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=1}^{t} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Figure 3.1 Formula

Now, we could actually stop here. All we need to do is just run through the full range of *t* values [1,256] and pick the value that minimizes.

Disadvantages:

- The method breaks down when the two classes are very unequal.
- The method doesn’t work well for variable illumination

3.1.2Niblack’s Algorithm

Niblack’s algorithm calculates a pixel-wise threshold by sliding a rectangular window over the gray level image. The computation of threshold is based on the local mean m and the standard deviation s of all the pixels in the window and is given as:

$$T_{\text{Ni black}} = m + k*s$$

$$T_{\text{Ni black}} = m + k\sqrt{\frac{1}{NP}\sum (p_i - m)^2}$$

$$= m + k\sqrt{\frac{\sum p_i^2}{NP} - m^2} \quad = m + k\sqrt{B}$$

Figure 3.2 Threshold for Ni black

where NP is the number of pixels in the gray image, m is the average value of the pixels pi, and k is fixed to -0.2 by the authors.

Disadvantage:

- A lot of computation is needed.
- Binary noise in the resultant image is large

3.2Proposed algorithm

For the proposed algorithm the threshold is fixed between 0-30 and this threshold is compared by the difference between the difference between the maximum mean and minimum mean . The mean is found for different filter sizes at each pixel. The difference Vp=mean(max) – mean(min) when compared with the threshold Cv tells if the pixel is in background region or text and near text region.This algorithm tries to find the change when a text is starting to build up. The graph shows the various regions:

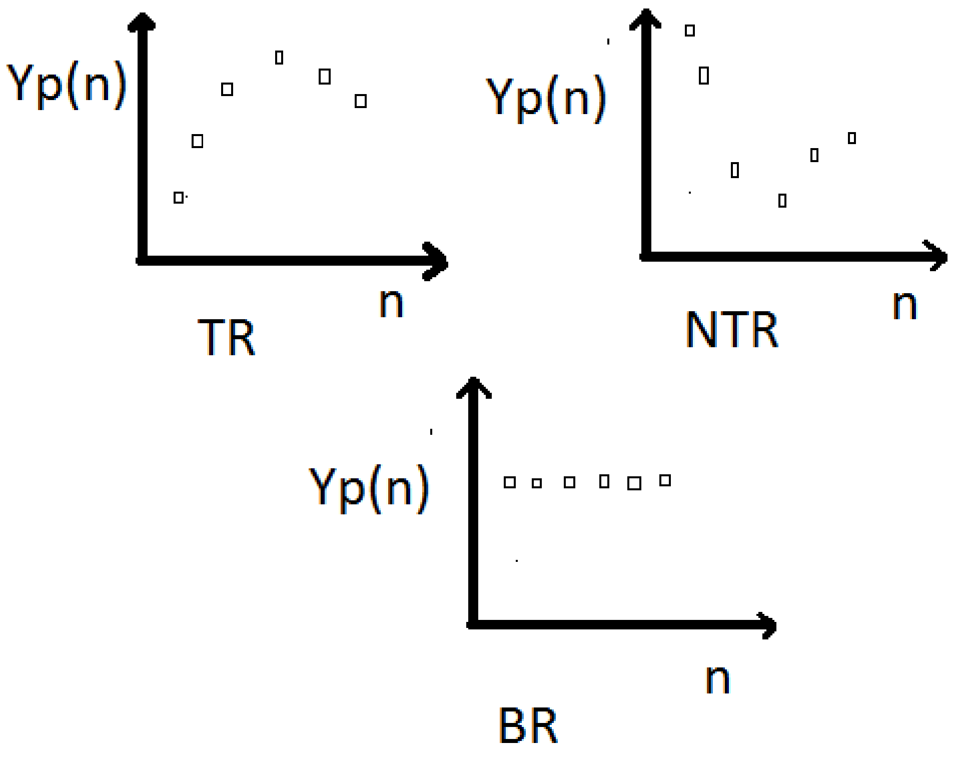


Figure 3.3 Graph Regions

Advantages:

- Threshold need not be calculated.
- Gives better result than Otsu's and Niblack algorithm in case of uneven lighting.

3.3 Server

To create a local server so that the clients in the given LAN connection can connect to the server which consists the proposed code. The clients can process the degraded image with the help of the browser and save it in their directory where they are keeping their notes.

Advantages:

- The code to process can be used by anyone in the LAN network.
- This server can be used within an institution.
- Students can make notes by using this code

CHAPTER 4
SYSYEM DESIGN

The system is broken down into two modules. The first modules deals with the code that deals with the processing of the image to get a binarized output and the second module deals with creating a server so that the client can upload and process the image using the code and then download back the resultant image to his computer.

4.1 Algorithm

Here the description of the code for binarizing the image is given step by step which is done with the help of Opencv libraries.

4.1.1 Creating a 3D matrix

The height and the width of the grayscale image is saved in an integer variable. Then we have created a 3D matrix to store the image. Each pixel is linking to an array to store information regarding that pixel.The 3D matrix is made using the malloc()

4.1.1.1 Coding

First we are saving the height and the width of the image in variables. A pointer logic points to an array of pointers of the size height. This array of pointers point to another array of pointers of size width. Hence this forms the two dimentional array representation of the picture of dimensions height and width. The pointers of size width points to an integer array of size four. This array stores information about the pixels in it. The first element is made to store the original pixel value and the other two are made to store the maximum and the minimum value from the mean filter which is discussed later.

4.1.1.2 Diagram

The diagram shows the 3D array created using the pointer.The ***logic points to the 3D array. **height is the pointer to *width and *width is the pointer to an integer array of size of four elements.

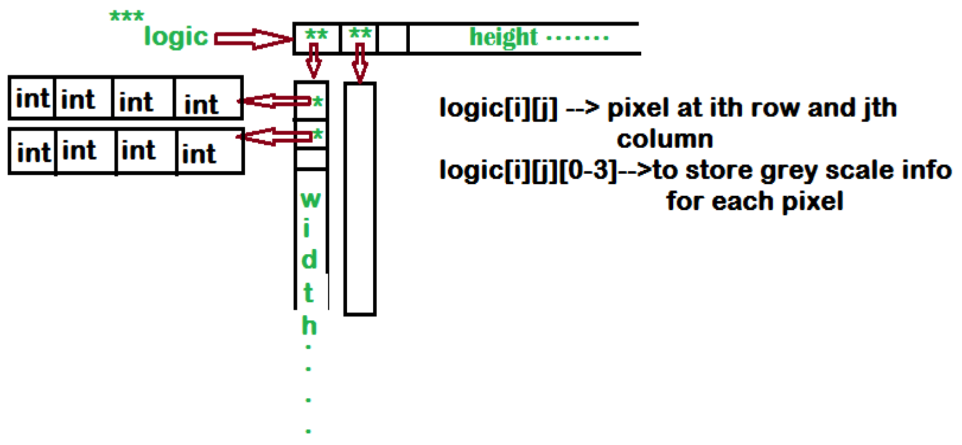

***
logic
**
**
height ....
int
int
int
int
int
int
int
int
*
*
width
logic[i][j] --> pixel at ith row and jth column
logic[i][j][0-3]-->to store grey scale info for each pixel

Figure 4.1 3D matrix

4.1.2 Creating New Image Variables

i. Tr_image:to store the text region of the image

ii. Ntr_image:to store the near next region of the image

iii. Br_image to store the Background region of the image

These variables are created to store the different regions.

4.1.3 Function For Finding Mean With Different Filter Size

Iterating 4 times to run mean filters of sizes 7x7,9x9,11x11,13x13 on each pixel of the image. The maximum mean and minimum mean is saved in the linked array elements arr[1] and arr [2] respectively.the original pixel value is stored in the first element [0] of the array.

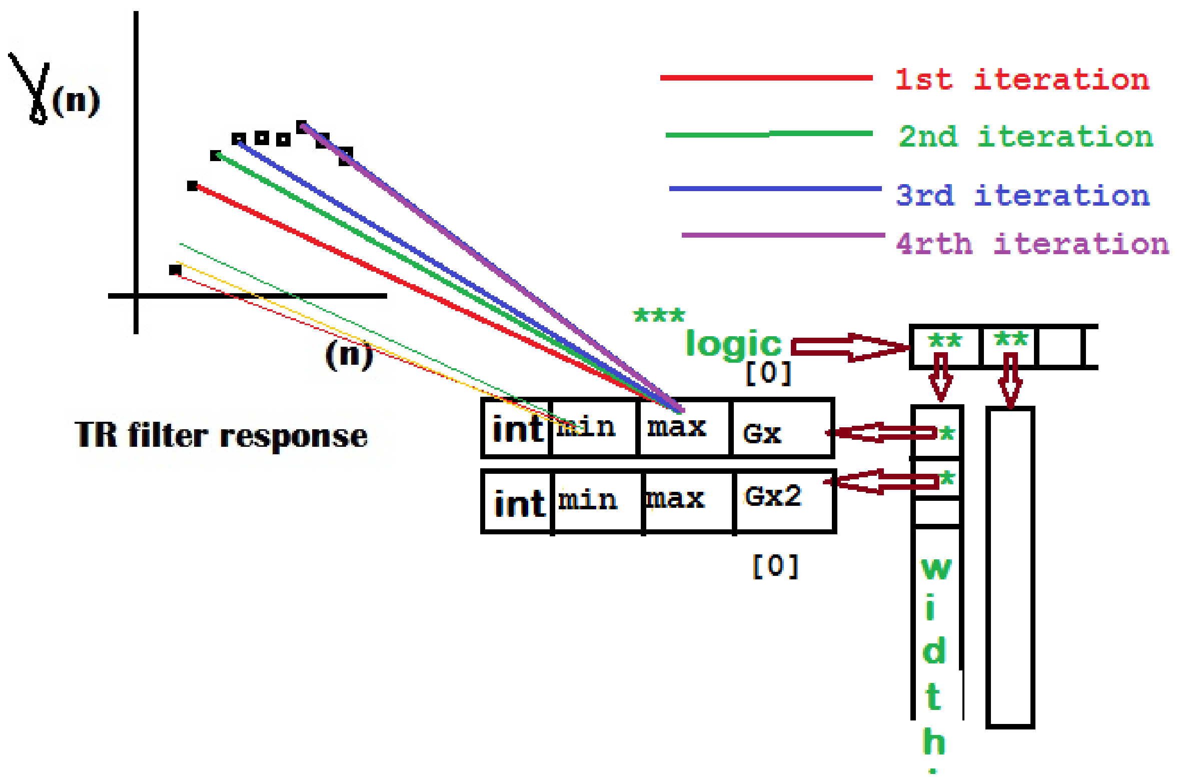


Figure 4.2 Iteration

4.1.4 Separating The Three Regions

Vp which is the difference between the mean maximum and mean minimum is calculated. The threshold for the mean difference is set as 11 represented as Cv. R is taken as CV/2.Using the following in if else saved the pixels in TR ,NTR and BR image variable separately by giving the pixel value 0 for TR and NTR and 255 for BR.

Region	Condition
BR	Vp<Cv
TR	Vp>Cv ;arr[0]-min mean(arr[1])<R
NTR	Vp>Cv ;max mean(arr[2])-arr[0]<=R

Table 4.1 different regions

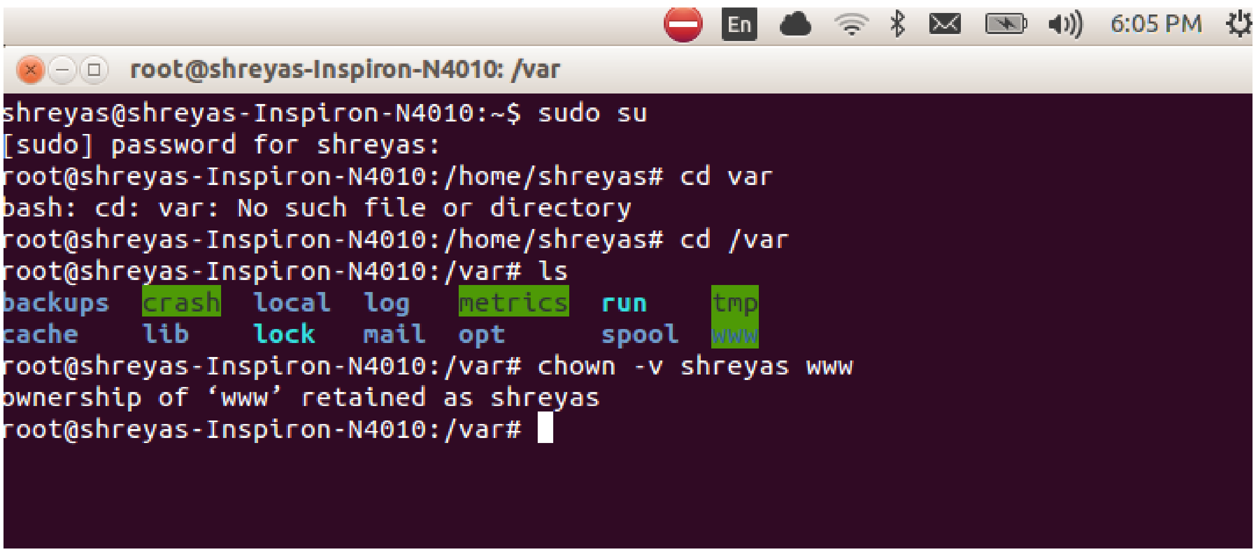
The image is written in the directory where the source image is stored with .jpg extension.

4.2 Creating Server

Downloaded XAMPP which is a free open source cross platform web server package , it consists of Apache2 HTTP Server, MySQL database and PHP language.

For my project I have used PHP for preparing the server-side scripting. XAMPP and linux provides a safe environment as most of the work needs root access hence the security is guaranteed.

After downloading XAMPP named it as localhost. As the operating system is Linux it has many permission issues. The php codes can not be created using notepad++ directory. It can only be created by terminal. The php code resides in the /var/www directory to function. www can initially be accessed only by using the terminal. We have changed the directory permission to the user "shreyas" from the root by typing the following lines in the command promt.



```
shreyas@shreyas-Inspiron-N4010:~$ sudo su
[sudo] password for shreyas:
root@shreyas-Inspiron-N4010:/home/shreyas# cd var
bash: cd: var: No such file or directory
root@shreyas-Inspiron-N4010:/home/shreyas# cd /var
root@shreyas-Inspiron-N4010:/var# ls
backups  crash  local  log   metrics  run    tmp
cache    lib    lock   mail  opt      spool  www
root@shreyas-Inspiron-N4010:/var# chown -v shreyas www
ownership of 'www' retained as shreyas
root@shreyas-Inspiron-N4010:/var# 
```

Figure 4.3 cmd

Sudo su is used to enter the super user mode

Cd /var is done to enter into var folder

ls will show the list of the directory "var" “chown” is used to change the ownership of a directory which is followed by “-v” name of the user we want to change the permission to “shreyas” and the name of the file/folder we want to change the permission of “www”.Now we can save notepad++ files into www and edit files by opening and saving again.

Before typing the php codes and executing them, we have to open the terminal and reach the php directory to open and edit the php.ini file. The “displaying error” = On which is initially set as Off so that the clients done get displayed error messages if any error occurs.

4.3 Working

1. Client uploads a picture using a form provided by the server.
2. The sever receivies the picture and downloads into a directory.
3. The server process the image using an image processing exe file.
4. The processed image is uploaded in form of thumbnail for the client to download the image
5. The client download s the processed image into its device.

The figure shows how client gets the processed image:

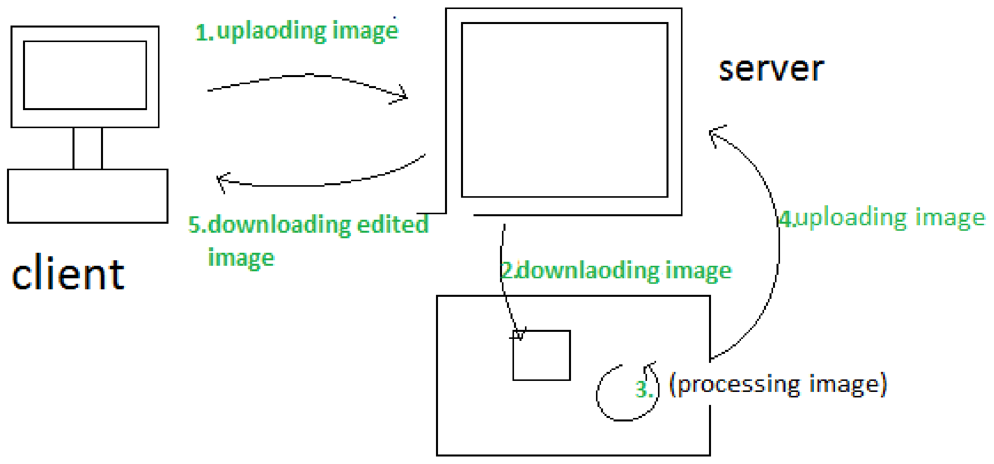


Figure 4.4 server and client

4.4 PHP Files

Made various php files to create a browser for client to upload the image ,process it and download the binarized image. The various php and html files are:

4.4.1 Index.html

The index.html will be the main page when the Ip address of the server is typed on the browser. The first page will have two frames. The frams are created by creating two rows by using
<frameset rows=”20%,80%”>
This will create two frames row wise. We set both the frames to a fresh php file namely link.php and link2.php by using the following code
<frame src=”link.php”>
<frame src=”link2.php”>
The title of the page is set as welcome

4.4.2 Link.php

This is used to create the heading for the page. The background is set as “teal” by writing the code as
<body bgcolor=”teal”>
Here teal has a hex value stored as variable. The heading is given as Welcome to Binarization by using font and bold tags

<font size=7 color=”#fffffff”>

#fffffff is used to generate white color againt the teal background which is one of the attributes of the font tag. The size for it is taken as 7 which is also an attribute of the tag font. The tag <dfn> adds boldness to the text.

4.4.3 Link2.php

The bottom part is again framed into two halves column wise. On th left side we can upload the image and on the right side we can download the image. The frames are set as 50%,50%.
<frameset cols=”50%,50%”>
<frame src=’link3.html’>

<frame src=’link4.php’>

4.4.4 Link3.html

Here we are displaying text asking the client if he wants to edit your photo if yes please click on the link bellow.

The link is provided by using the tag <a href=”file.php/html”> this tag will make the contents after it hypertext. This tag is closed by </a> tag.

The link is given to go to the page where form is provided for the client to choose and upload an image by submitting the image.

4.4.5 Link4.php

This is the other size of the frame in which the option to download is available. After process the image the user needs to come to this side of the frame to download the image to his specified directory. The background is given as “000080” which is light blue in color.
The hypertext is given by using the tag <a href=”link99.php”>download image</a>. ‘Download image’ becomes a hyper text to go to link99.php.

4.2.6 Link7.php

the form is created by using the tag <form>
<form action=”upload.php” method=”POST” enctype=”multipart/form-data”>

Attribute	Description
Action	Specifies where to send the form-data when a form is submitted.
Method	POST: Specifies the HTTP method to use when sending form-data
Enctype	Specifies how the form-data should be encoded when submitting it to the server

4.2 Attributes

The action is that the iamge is sent to upload.php

For this form we have to take an input of the type ‘file’ and of name ‘file’

<input type=’file’ name=’file’ />

We have to create a submit button so that we can submit the file

<input type=’submit’ value=’send file’>

The text “submit file” will appear on the button.

4.2.7 Upload.php:

After submitting the image in the previous page the image is stored in the temporary folder of the server. The name of the file can be saved in a variable $name by using the following command:

$name=$_FILES[‘file’][‘name’];

This is the exact name of the image that we have uploaded. When the image is uploaded on the server it resides in the temporary folder called tmp and has a temporary name to it. We can get the temporary name of the file as well as its path using the following code

$tmp_name=$_FILES[‘file’][‘tmp_name’];

We can also know the size and type of the file of the image by using $_FILES["file"]["size"] and
$_FILES["file"]["type"] respectively. If we want to view this we can directly echo this in the code.

Once this is done we have to save the image in the directory where the **project .exe** so that they remain in the same folder for the execution. This is done with the help of the php library function move_uploaded_image();

The parameters for this function are:

- Temporary name of the file $tmp_name.
- The path were the image will be stored.$_FILES[‘file’][‘name’]. instead of this we can directly use the variable $name.

After getting the name of the image we have to store the command to execute the program whose parmeter will be the image name.

Using the variable $cmd stored the code to execute the program.exe

```
$cmd=”cd uploads/ && ./project ”;
```

The image and the project.exe file is saved in the uploads folder in the www directory. To this we have concatenated the name of the image by using the code:

```
$cmd.=$name;
```

$cmd will store “cd uploads/ && ./project imagename.extension”. && is used to run to terminal codes,if the first one is successful only then the second line is executed. This execution line has to be stored into a file.txt. For this we have done the following line

```
$handle = fopen(“name.txt”,”w”);
fwrite($handle, $cmd);
fclose($handle);
```

handle is the handler of the file name.txt. this is open in the writing mode. Using the fwrite we have written the content of the variable $cmd into the text file name.txt. there can be various opening modes of a file they are:

W+	Open for reading and writing, place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
W	Open for writing only, place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
A	Opens for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.
A+	Opens for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.

4.3 opening modes

After this we have given a new link to display that the file is successfully uploaded and to continue click on process so that the picture can be edited. A hyperlink is given to the next page which will successfully process the image as “process”.

4.2.8 Process.php

Here the server waits for the image to download. This is done by executing the code sleep(5).Were 5 is for five seconds. And then the name.txt is read and the command is executed using the function exec(command line).

CHAPTER 5
CONCLUSION

The proposed algorithm is successfully eliminating the uneven lighting condition and the text are perfectly binarized separating the near text and the background region. The client can successfully download the image after processing on the server. The various advantages are that it gives better result than other algorithms, can be used to make digital notes and can be used to binarize any text image of any grey scale value. It can have various applications. It is used to make computer notes so that the processed image can be stored in a folder to keep a record of the photos taken to create the notes. The processed image is very clear with background eliminated.

APPENDIX 1

OPENCV CODE

```
#include “cv.h”
#include “highgui.h”
#include<math.h>
#include <iostream>

#define N 7

using namespace cv;
using namespace std;

void compute_logic ( Mat , int ***);
//function to perform to store maximum and minimum mean


int main(int argc, char* argv[])
{
   if(argc<2)
         {
                  printf(“ please specify the input file name \n execute like <executable>
<inputfile> \n “);
                  exit(0);

         }


   Mat im_rgb;// takes RBG image by the user
   Mat im_gray;// for storing the converted RBG into grey scale
   Mat im_smooth;// to


   //storing image in im_rbg given by the user
   im_rgb = imread(argv[1]);
   //converting to grey scale and saving in im_gray
   cvtColor(im_rgb,im_gray,CV_RGB2GRAY);

   int height = im_rgb.rows;//stores the height of the image
   int width = im_rgb.cols;//stores the width of the image

   //saving the grey scale image in the system
   imwrite(“grayscale.pgm”,im_gray);
   //forming the three dimentional matrix
   int ***logic = (int***)malloc(height*sizeof(int**));
                  for (int i=0; i<height; i++)
               {
```

```
            logic[i] = (int**)malloc(width*sizeof(int*));
                  for (int j=0; j<width; j++)
                     logic[i][j] = (int*)malloc(4*sizeof(int));
            }

// initializing the first three elements as the original
//pixel
for(int i=0; i<im_gray.rows;i++)
    {
                   uchar* rowi = im_gray.ptr/*<uchar>*/(i);
                  for(int j=0; j<im_gray.cols; j++)
                  {
                        logic[i][j][0] = logic[i][j][1] =
 logic[i][j][2] = (int) rowi[j];
                  }
    }

//iterating to find the mean with different filter sizes

for ( int I =3;i<N;i++)
{
 //calling mean filter from the library using blur
    blur(im_gray, im_smooth, Size(2*i+1,2*i+1));
    compute_logic( im_smooth, logic);




}

int Cv=8;// setting threshold as 8

int Rg = Cv/2;// setting R as half of the threshold
int Vp;

//im_tr image object to store the text region
Mat im_tr = Mat(height, width, CV_32F);
//to store the near text region
Mat im_ntr =Mat(height, width, CV_32F);
//to store the background region
Mat im_bgr = Mat(height, width, CV_32F);




for(int i=0; i<im_gray.rows;i++)
    {


                  for(int j=0; j<im_gray.cols; j++)
            {

  //setting background for text by default
  //as 255(white)
                        im_tr.at<float>(I, j) = (float) 255;
  //by default as white
                        im_ntr.at<float>(I, j) = (float) 255;
  //by default as black
                        im_bgr.at<float>(I, j) = (float) 0;

//computing vp
                        Vp = logic[i][j][2] – logic[i][j][1];


                        if ( Vp <= Cv)
                        {    //its background set as white
                              im_bgr.at<float>(I, j) = (float) 255;

                              logic[i][j][3] = 3;
                        }
                        else
                        {

                        if ( logic[i][j][0] – logic[i][j][1] < Rg)
                              {    // its near text set as black
                              im_tr.at<float>(I, j) = (float) 0;
                                    logic[i][j][3] = 1;

                              }


                        if ( logic[i][j][2] – logic[i][j][0] <= Rg)
                              {
     // its text set as black
                              im_ntr.at<float>(I, j) = (float) 0;
                                    logic[i][j][3] = 2;

                              }







                        }
                        }
            }


    // save text image in directory
    imwrite(“textregion.jpg”,im_tr;
   // save near text region image in directory
  imwrite(“neartextregion.jpg”,im_ntr);
    // save background region in directory
    imwrite(“backgroundregion.jpg”,im_bgr);
}


//function that stores the minimum mean un 1st logic[i][j][1]
//and stores maximum in 2nd logic[i][j][2]
void compute_logic ( Mat im_smooth, int ***logic)
{

int val ;
            for(int i=0; i<im_smooth.rows;i++)
            {
                         uchar* rowi = im_smooth.ptr/*<uchar>*/(i);
                        for(int j=0; j<im_smooth.cols; j++)
                        {
                                    val = (int)rowi[j];

                                    if ( val < logic[i][j][1] )
                                    logic[i][j][1] = val;


                                    if ( val > logic[i][j][2] )
                                    logic[i][j][2] = val;






                  }

            }




}
```

PHP files for the server

```
//creating form
<form action="upload.php" method="POST" enctype="multipart/form-data">
<pre>		<input type="file" name="file" /><br><br></pre>
<pre>		<input type="submit" value="Send File" /></pre>
</form>
<?php
//saving the command to execute the processing of the image
echo $name=$_FILES['file']['name'];
$cmd="cd uploads/ && ./project ";
$cmd .=$name;
$handle = fopen("name.txt","w");
//saving command for execution in name.txt
fwrite($handle,$cmd);
fclose($handle);
 $tmp_name=$_FILES['file']['tmp_name'];
move_uploaded_file($tmp_name,'uploads/'.$name);
echo "\n";
echo 'uploaded';
?>
//waiting for server to download and then process image using exec function
<?php
sleep(5);
$cmd = implode(" ",file('name.txt'));
exec($cmd);
?>
```

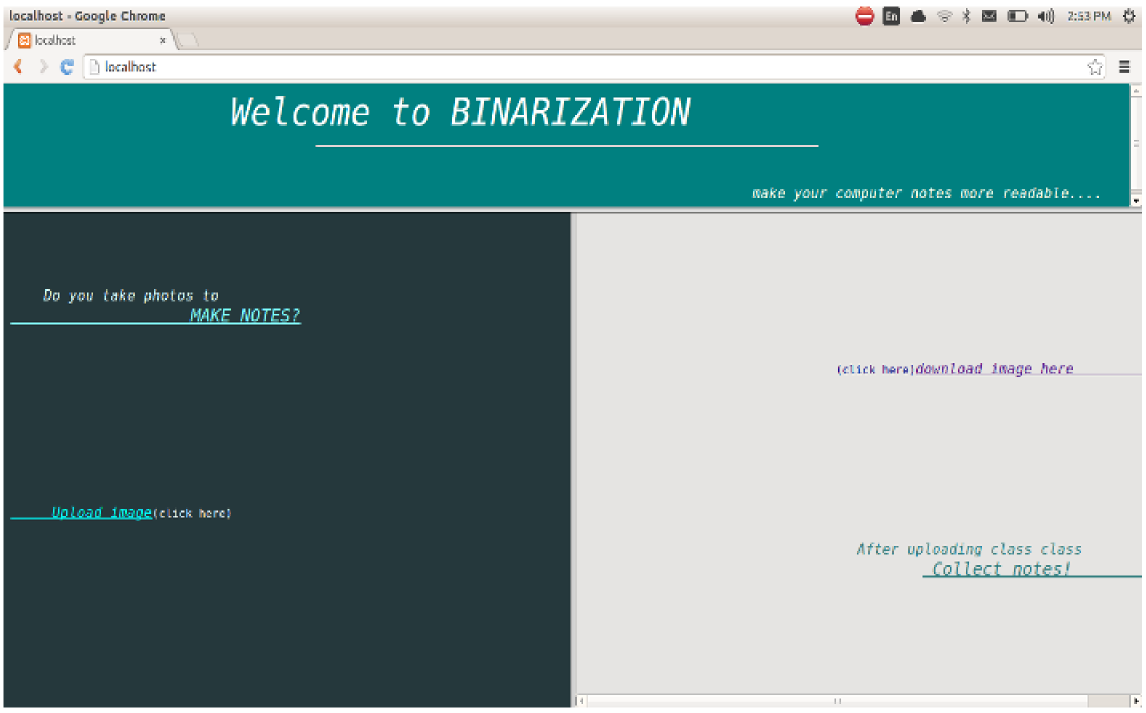
APPENDEX 2

Browser Display

Here the step by step view of the browser for uploading,processing and downloading image.

1.Outlook

The index page which appears when the client types the IP address of the server system. The index page is divided into two frames for uploading in the left and downloading in the right.



Welcome to BINARIZATION

make your computer notes more readable....

Do you take photos to
MAKE NOTES?

Upload image(click here)

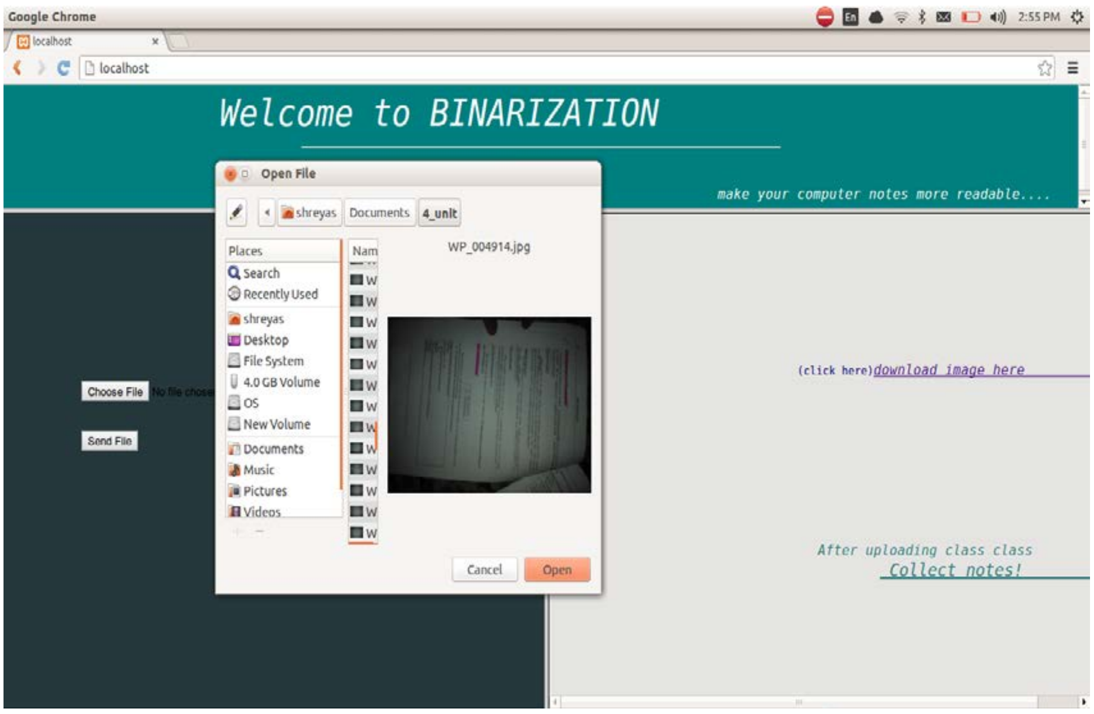
(click here)download image here

After uploading class class
Collect notes!

browser display

The client clicks on upload image (click here) to get to the form to upload image.

2.Uploading Image

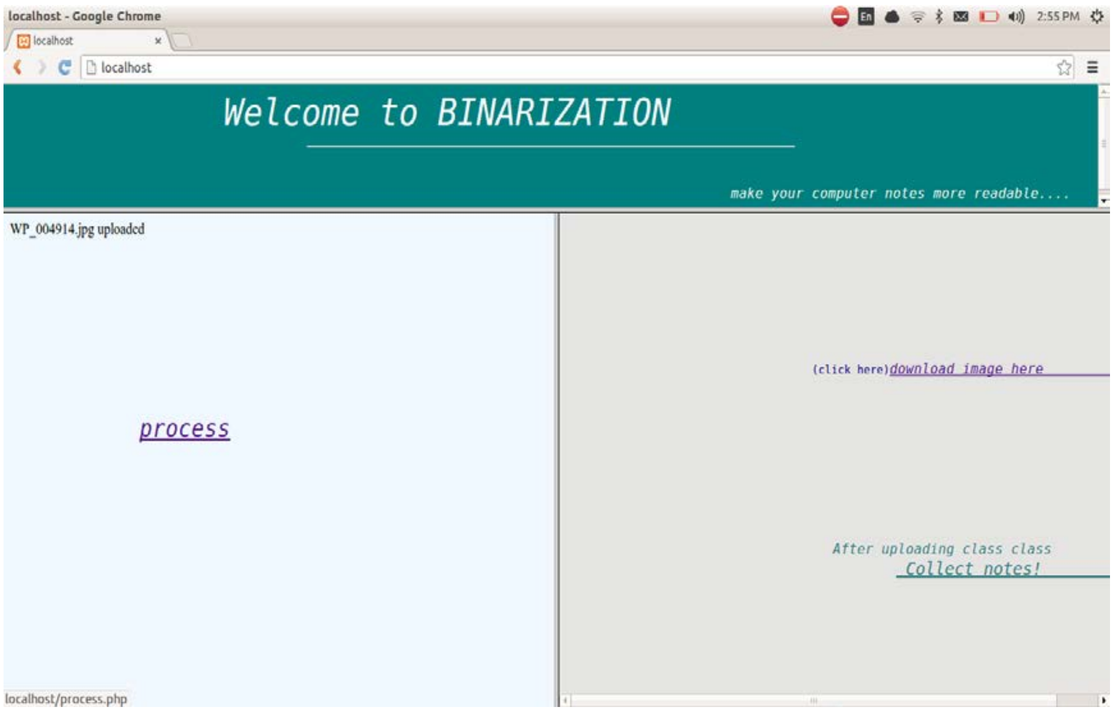


click on upload

The client selects the photo and clicks on submit.The client is shown that the picture is uploaded and is ready to be processed. The client clicks on process to execute the code with the image.

3.Processing Image

The client next clicks on "process" to process the uploaded image.



Welcome to BINARIZATION

make your computer notes more readable....

WP_004914.jpg uploaded

process

(click here)download image here

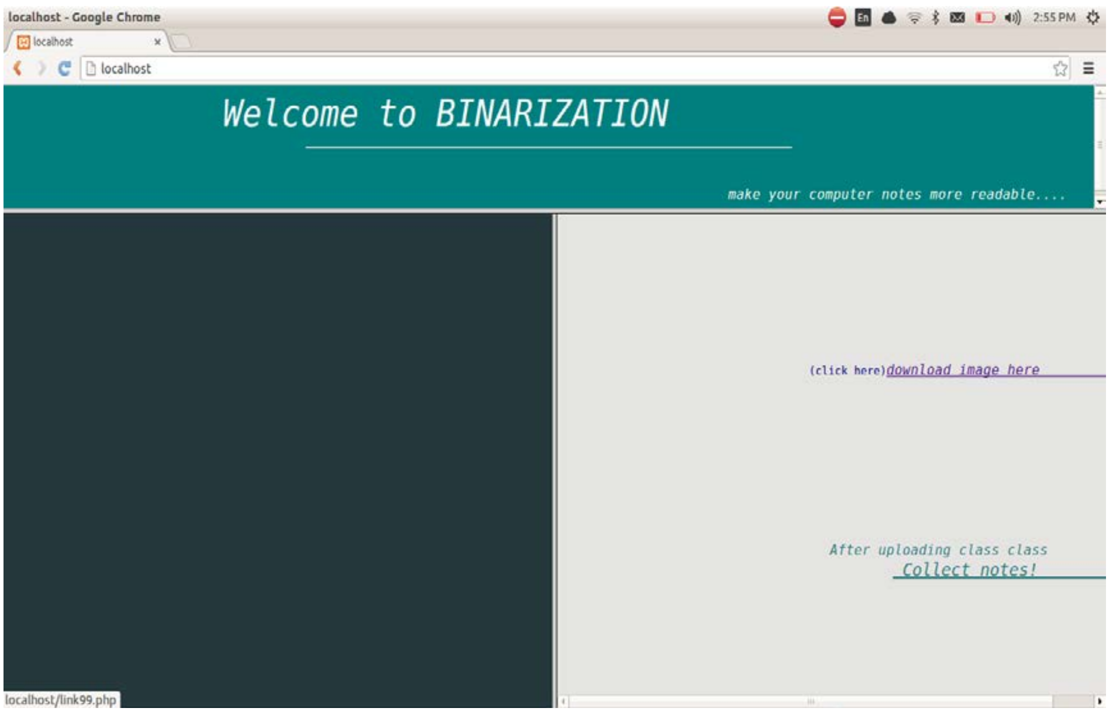
After uploading class class
Collect notes!

localhost/process.php

click on process

4.Processing Done

Now the image is processed and uploaded again by the server. The client clicks on download image here to save the image.

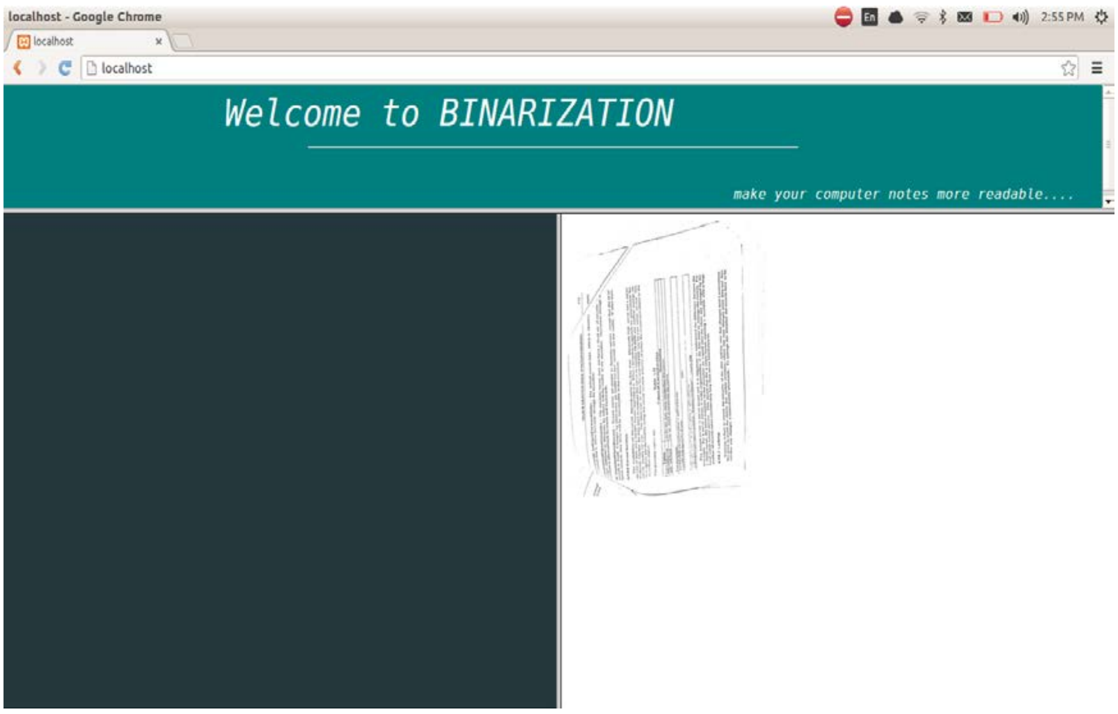

localhost - Google Chrome
localhost
localhost
Welcome to BINARIZATION
make your computer notes more readable....
(click here)download image here
After uploading class class
Collect notes!
localhost/link99.php

click on download

A thumbnail appears which the client has to right click and save the image to his chosen directory.

5.Saving Image

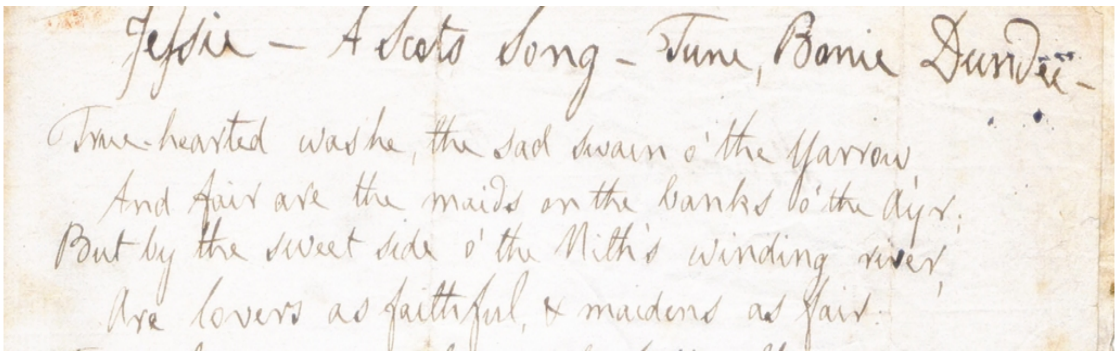
After the processing is done the client has to right click on the thumbnail and save it in a directory of his choice.


localhost - Google Chrome
localhost
localhost
2:55 PM
Welcome to BINARIZATION
make your computer notes more readable....

save image

Few Examples

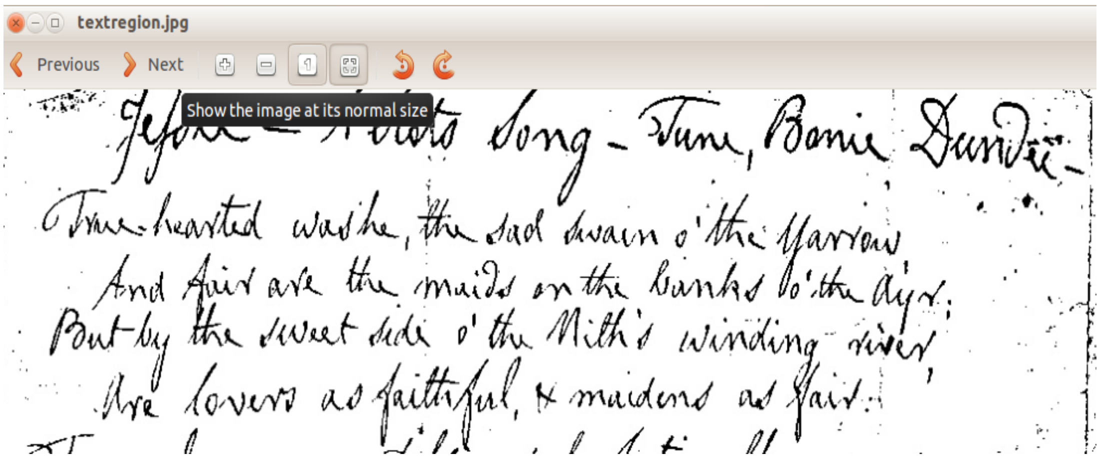
Example 1



Jeſsie – A Scots Song – Tune, Bonie Dundee.
True-hearted was he, the sad swain o' the Yarrow
And fair are the maids on the banks o' the Ayr;
But by the sweet side o' the Nith's winding river,
Are lovers as faithful, & maidens as fair.

Here the body of the poem is very light as compared to the heading which is also having different grey scale values. We want to have a uniform color text.

Output:

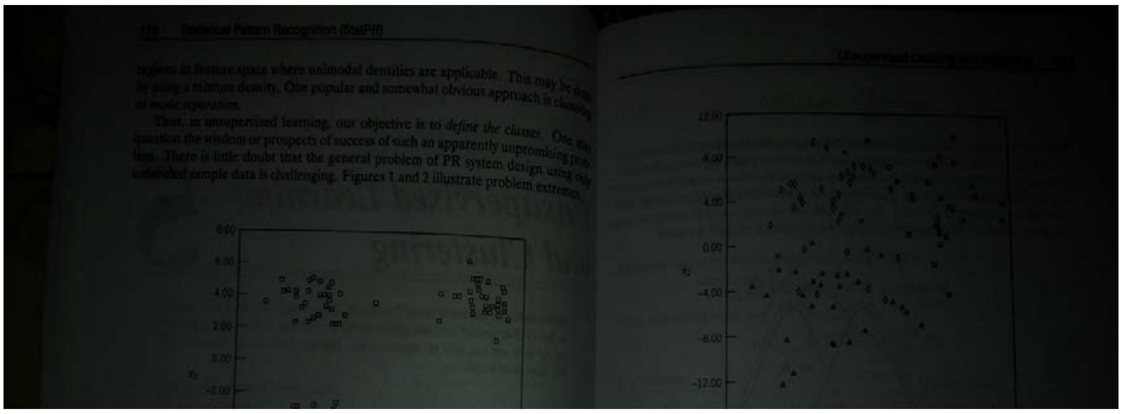


Jefore [illegible] Song - Tune, Bonie Dundee

True-hearted was he, the sad swain o' the Yarrow
And fair are the maids on the banks o' the Ayr;
But by the sweet side o' the Nith's winding river,
Are lovers as faithful, & maidens as fair:

Example2

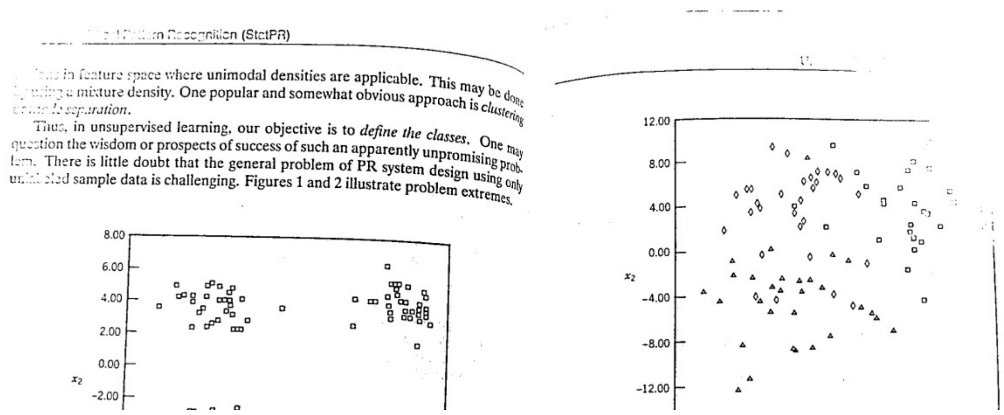
Here the corners of the image is very dark.


118 Statistical Pattern Recognition (StatPR)

regions in feature space where unimodal densities are applicable. This may be [illegible] by using a mixture density. One popular and somewhat obvious approach is clustering or mode separation.

Thus, in unsupervised learning, our objective is to *define the classes*. One may question the wisdom or prospects of success of such an apparently unpromising problem. There is little doubt that the general problem of PR system design using only unlabeled sample data is challenging. Figures 1 and 2 illustrate problem extremes.

And the text is almost not readable



[illegible] Pattern Recognition (StatPR)

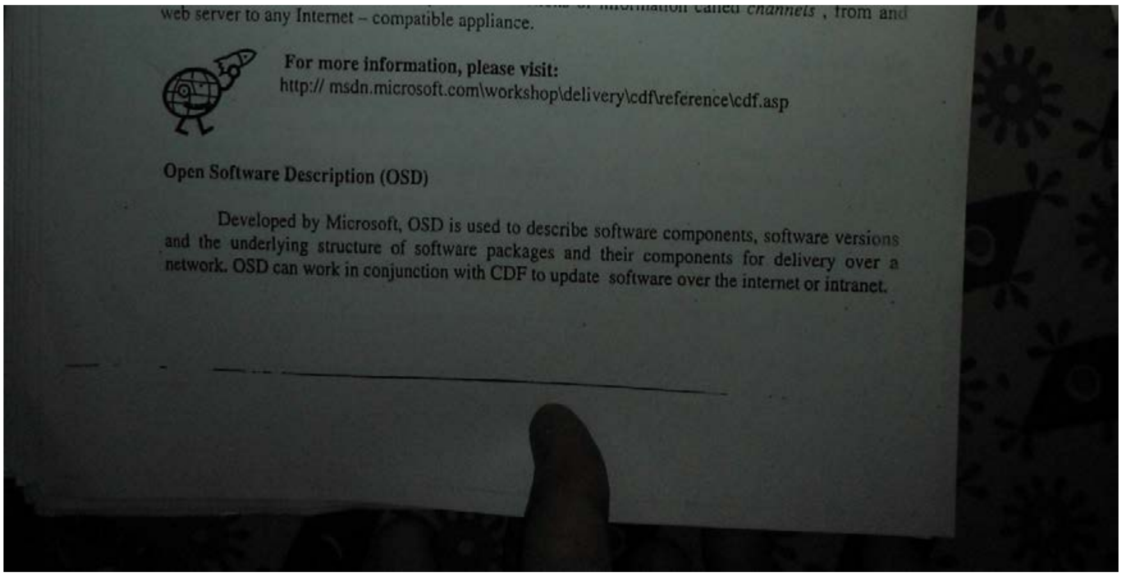
[illegible] in feature space where unimodal densities are applicable. This may be done [illegible] using a mixture density. One popular and somewhat obvious approach is clustering [illegible] mode separation.

Thus, in unsupervised learning, our objective is to *define the classes*. One may question the wisdom or prospects of success of such an apparently unpromising problem. There is little doubt that the general problem of PR system design using only unlabeled sample data is challenging. Figures 1 and 2 illustrate problem extremes.

The output is almost very remarkable.

Example 3

Here the image contains a tumb and a background which is black in color.hence to show its not a normal thresholding function.



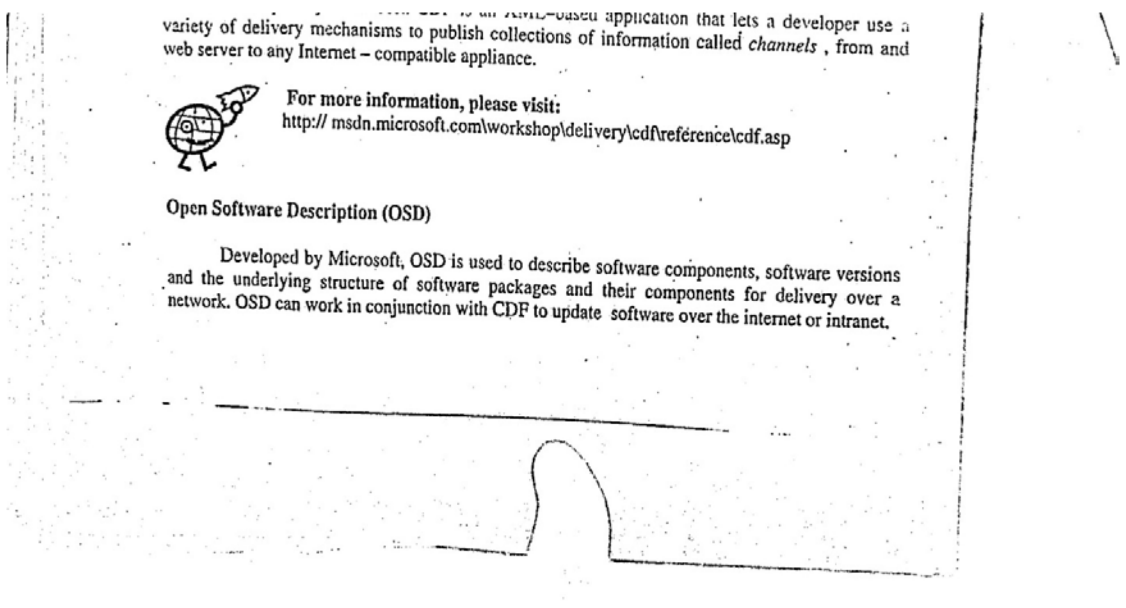
[illegible] information called *channels* , from and web server to any Internet – compatible appliance.

For more information, please visit:
http:// msdn.microsoft.com\workshop\delivery\cdf\reference\cdf.asp

Open Software Description (OSD)

Developed by Microsoft, OSD is used to describe software components, software versions and the underlying structure of software packages and their components for delivery over a network. OSD can work in conjunction with CDF to update software over the internet or intranet.

We can see that the thumb is eliminated along with the background which was a bedsheet.



CDF is an XML-based application that lets a developer use a variety of delivery mechanisms to publish collections of information called *channels* , from and web server to any Internet – compatible appliance.

For more information, please visit:
http:// msdn.microsoft.com\workshop\delivery\cdf\reference\cdf.asp

Open Software Description (OSD)

Developed by Microsoft, OSD is used to describe software components, software versions and the underlying structure of software packages and their components for delivery over a network. OSD can work in conjunction with CDF to update software over the internet or intranet.

REFERENCES

[1] Opencv tutorial by opencv.org

[2] B.Su, S.Lu , and C. L. Tan"Binarization of historical handwritten Document images using local maximum and minimum filter," in Proc Int. Workshop Document Anal. Syst. Jun.2010, pp.159-166.

[3] G. Leedham, C. Yan, K. Takru,J.Hadi, N. Tan ,and Limian,"comparison of some thresholding algorithms for text/background segmentation in difficult docummment images,"in Proc Int. Conf. Document Anal. Recongnit., vol. 13.2003,pp. 859-864

[4] Keshou Wu, Wuhan, Lijuan "The Design and Implementation of Image Compression Comparator in a Component-Based Simulation System for Digital Image processing."

[5] N. Otsu," A threshold selection method from grey scale histogram."IEEE Trans.Syst.,Man,Cybern., vol. 19,no. 1,pp. 62-66,Jan. 1979.

[6] W.Niblack An Introduction to Digital Image Processing. 1986.

[7] Php tutorials using youtube and w3school.com

[8] Robust Document Image Binarization Technique for Degraded Document Images, Bolan Su, Shijian Lu, and Chew Lim , Tan, Senior Member, IEEE

[9] Stack overflow.com

[10] Siva Rama Sastry Gumma OCR Preprocessing.

[11] Wang Jianxin, Chen Songqiao, Jia Weijia, Pei Huiming, the Design and Implementation of Virtual Laboratory Platform in Internet, Platform in Internet, Proceedings of ICWL 2002 , Hong Kong,2002.8,160-168.

[13] Wang Jianxin, Peng Bei, Jia Weijia, Design and Implementation of Virtual Computer Network Lab Based on NS2 in the Internet, Proceeding of ICWL 2004, Lecture Notes in Computer Science 3143, 2004, 346-353.