Computer Engineering Case Study – Image Features Extraction using Gray Level Cooccurrence Matrix (GLCM)

Yousef Al-Jazzazi

ENGR-AD 1000, Fall 2022

STEP 1: *Problem Identification and Statement*

The problem is extracting the features, such as contrast, energy, and homogeneity, from a file containing image data/pixel information.

STEP 2: *Gathering of Information and Input and Output Description*

Gray Level Co-occurrence Matrix (GLCM) represents the spatial relationships of pixels in the image. GLCM examines the spatial relationship among pixels and defines how frequently a combination of pixels is present in an image in a given direction **θ** and distance **d**. Using GLCM, you can derive several statistics like 'contrast,' 'energy,' and 'homogeneity.' These statistics provide information about the texture of an image.

<u>Creating GLCM from image pixel information:</u>
Before creating GLCM, we should find the maximum gray level in the image- as the dimensions of the GLCM depend on the maximum gray level in an image. The dimensions of GLCM should be set as the maximum value plus one.
For the values inside GLCM, the figure below illustrates the process used to create the GLCM from an image with 8 gray levels.

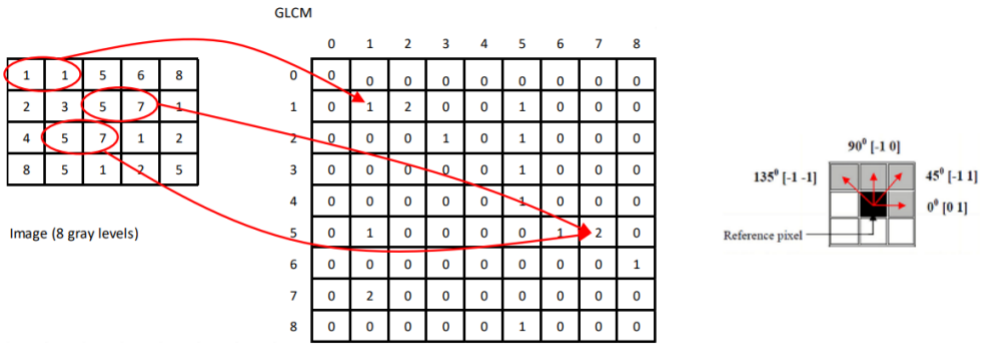


Figure 1 Input Image and the generated GLCM with θ = 0, and d = 1

In the output GLCM, element (1,1) contains the value 1 because there is only one instance in the input image where two horizontally adjacent pixels have the values 1 and 1, respectively. Element (1,2) contains the value 2 because there are two instances where two horizontally adjacent pixels have values 1 and 2. Element (1,3) in the GLCM has the value 0 because there are no instances of two horizontally adjacent pixels with the values 1 and 3.

Creating normalized GLCM from the GLCM matrix:

A normalized GLCM matrix has the same dimension as the GLCM matrix. The values inside the GLCM matrix at a given point (i,j) is equal to the value of the point in GLCM that has the same location divided by the sum of values in GLCM. Look equation below:

$$p(i,j) = \frac{M(i,j)}{\sum_{i,j} M(i,j)}$$

Finding the values of the attributes of an image (contrast, energy, homogeneity) using normalized GLCM matrix: a

The attributes can be calculated using these equations where p(i,j) is the value in the Normalized GLCM matrix at location (i,j):

- Contrast

$$contrast = \sum_{i,j} |i-j|^2 \times p(i,j)$$

- Energy

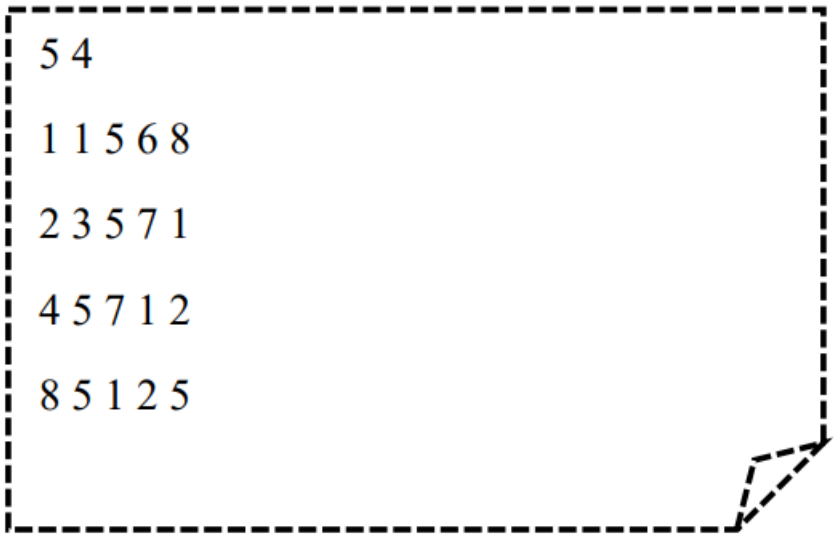
$$energy = \sum_{i,j} p(i,j)^2$$

- Homogeneity (Similarity)

$$homogeneity = \sum_{i,j} \frac{p(i,j)}{1+|i-j|^2}$$

Dealing with Files:

The image data information is going to be passed by file. The data in the file is assumed to have the dimensions of the image stored in the first line as (columns, rows), followed by the pixel information. Look example below:



Furthermore, it is assumed that the file contains at least information of size 2X2 and that all the values inside are positive integers.

Note: The file will be given by the user. Our algorithm is concerned with running the calculation from the information in the file not producing it.

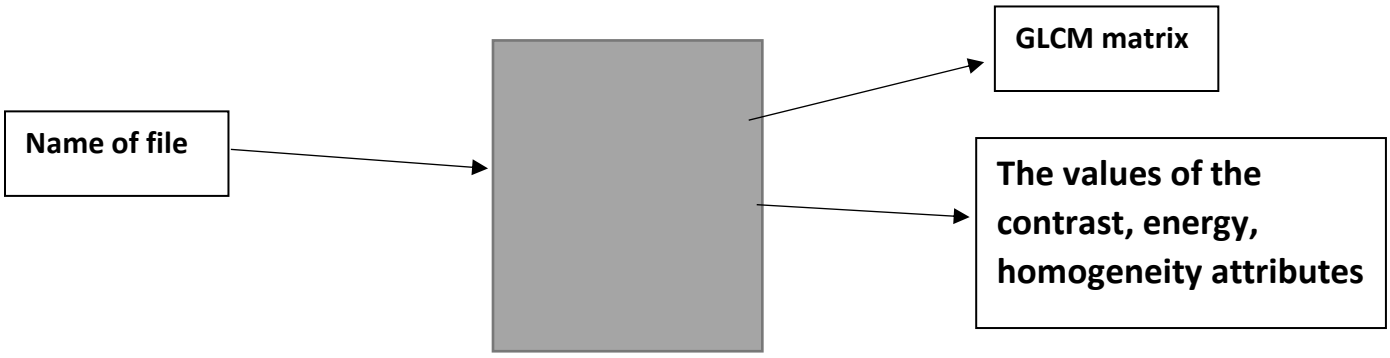
This will be organized through a menu where it is going to ask the user what function of the program they want to work with, namely (1) Create and normalize the GLCM matrix (2) Compute the statistical parameters of the texture – we assume for this function that the user wants to see the results as well, so they are printed when the user calls this function(3) Print the GLCM matrix- the normalized matrix will not be printed(4) quit the program. It is important to mention that the program will ask the user for a file name even before the user gets to the menu, and unless the user gives a valid file name, they will not be able to access the menu. They can choose to exit the program by typing 0.

In case the user selects the first option from the menu (Create and normalize the GLCM matrix), the program is going to create the GLCM matrix and create

another identical matrix that will be normalized. The program then returns to the main menu. If the user selects the second option and they did not create the GLCM matrix, the program will print a message stating that they should create the matrix first. If they choose the second option and the matrix has been created, the program is going to calculate the attributes (energy, homogeneity, and contrast) of the image using the normalized GLCM and print these attributes. The program then returns to the main menu. If the user selects the third option and they did not create the GLCM matrix, the program will print a message stating that they should create the matrix first. If they choose the third option and the matrix has been created, the program will print the GLCM matrix. If the user selects the exit option, the program prints a terminating message and quits the program.

I/O description:

The program will take three inputs and produce one output:



STEP3: Test Cases and Algorithm Design:

A) Test cases

The values of the attributes were calculated using website mathworks.

Test case #1:

File: “image.txt”

5 4

1 1 5 6 8

2 3 5 7 1

4 5 7 1 2

8 5 1 2 5

File Name Input [String]	Menu input [String]	Output	Menu input [String]	Output	Menu Input [String]
image.txt	1	Done	2	Contrast: 8.93 Energy: 0.086 Homogeneity: 0.3762	3

Continue

Output	Go back to menu	Output
0 0 0 0 0 0 0 0 0 0 1 2 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 1 0 2 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	4	Thanks for using our program

Test case #2:

File: “image.txt”

10 10
1 5 1 5 3 1 3 3 3 2
1 4 0 3 3 1 1 1 3 0
3 4 3 0 1 1 0 2 5 0
2 0 2 1 3 0 1 5 1 1
1 2 4 3 4 4 0 5 2 2
3 2 4 2 3 5 1 1 2 5
1 4 0 5 3 4 4 0 2 1
4 0 0 0 5 4 5 2 3 4
1 3 0 3 2 3 3 0 4 5
1 4 3 2 5 4 5 2 5 1

File Name Input [String]	**Output**	**File Name Input [String]**	**Menu Input [String]**	**Output**	**Menu input [String]**	**Output**
gg.txt	Error. The file doesn't exist. Please try again	image.txt	2	Please create GLCM before calculating attributes	1	Done

Continue

Menu Input[string]	**Output**	**Menu Input**	**Output**
3	2 2 3 2 1 3 1 5 2 4 3 3 1 2 1 3 2 4 5 2 4 4 4 1 5 0 1 3 2 3 1 4 3 2 2 0	2	Contrast: 6.12 Energy: 0.0358 Homogeneity: 0.4428

Test case #3:

File: “image.txt”

3 3

1 2 3

9 5 4

6 6 7

File Name Input [String]	**Output**	**File Name Input [String]**	**Menu Input [String]**
Yousef.txt	Error. The file doesn't exist. Please try again	0	Thanks for using our porgram

Test case #4:

File: “image.txt”

3 3

1 2 3

9 5 4

6 6 7

File Name Input [String]	**Menu Input [String]**	**Output**	**Menu input [String]**	**Output**	**Menu input [String]**	**Output**
Image.txt	1	done	3	GLCM matrix: 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 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 1 0 0 0 0 0 0 0 0 0 0 0 1 1 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 1 0 0 0 0	4	Thanks for using our program

B) Algorithm (Pseudocode)

```
Include the iostream, cmath,fstream, libraries


Initialize function CreateMatrix void with a parameter containing two double
pointer type int, and two integers

Initialize function CreateNormalized void with a parameter containing one
double pointer type int, one double pointer type double, and one integer.

Initialize functions printMatritype void with a parameter containing one
double pointer type int, and one integer.

Initialize functions contrast, homogeneity, and Energy void with a parameter
containing one double pointer type double with one integer.

Initialize function importImage that returns a double pointer type int with
parameter containing 3 references to integer variables

Main function
      Declare two bool variables repeat and GLCMcreated. Initialize repeat to
true and GLCMcreated to false.
      Declare two double pointers type int GLCM and image
      Declare double pointer type double
      Declare three int variables max, rows, and columns and initialize them
to 0
      Declare string variable choice and initialize to an empty string.
      Print a message explaining the program.
      Give the variable image the value returned by calling the function
importImage with parameter row, columns, and max


      Initialize GLCM double pointer to point to an array of pointers type
int of size equal to one plus max
      Initialize normalizedGLCM double pointer to point to an array of
pointers type double of size equal to one plus max

Start for loop that iterates max plus 1 times
            initialize pointer at position i of GLCM to point to an array of
integer size max plus 1, have all the values in integer array equal to 0
      initialize pointer at position i of NormalizedGLCM to point to an array
of doubles size max plus 1, have all the values in the array of doubles to
equal 0

      end loop

      Start loop while choice string variable isn't equal to 4
            Display a message that explains the options of the menu and ask
user for input choice

            input value from user and store it in variable choice

            if choice is equal to 1
```

```
                  call createMatrix-pass GLCM, image, rows, and columns - and
createNormalized- pass GLCM, Normalized, and max- functions

                  set GLCMcreated to true

            else if choice is equal to 2

                  if GLCMcreated is true

                        call contrast, Energy, and Homogeneity functions-
pass normalizedGLCM, and max for each

                  if GLCM is not true

                        print a message stating that the user should create
the GLCM first

            else if choice is equal to 3

                  if GLCMcreated is true

                        call printMatrix- pass GLCM and max

                  if GLCM is not true

                        print a message stating that the user should create
the GLCM first

            else if choice is 4

                  print a message thanking the user

            else choice is equal to anything else

                  display message that asks the user to input one of the
options


      End loop


      Free double pointer variable image GLCM and normalized

End main

Start createMatrix function type void with parameters two double pointer type
int: GLCM and image, two integer: rows and columns.
      Start for loop with variable i that iterate rows number of times
            Start for loop with variable j that interates one less than
columns number of times
                  Increment the in GLCM that has dimensions where row: is the
value in image array that has position (i,j) and column: the value in image
array that has position (i,j+1) by one

End function
```

Start createNormalized function type void with parameters one double pointer type int GLCM, one double pointer type double NormalizedGLCM , and an integer max.

Declare variable type double called sum and equal it to 0

Start nested for loop with variables i and j where each loop iterates max plus one number of times

Increment sum by the value in GLCM that has position (i,j)

End Nested loop

Start nested for loop with variables i and j where each loop iterates max plus one number of times

Make value in normalizedGLCM with positions (i,j) equal to the value in GLCM position (i,j) devidied by the sum

End nested loop

End function

Start void printMatrix function with parameters double pointer type int GLCM and integer variable max

Start for loop with variables i that iterates max plus one number of times

Start for loop with variables j that iterates max plus one number of times

Print value of GLCM that with location (i,j)

End loop

Print new line

End loop

End function

Start void function contrast with double pointer type double NormalziedGLCM and integer max

Declare variable type double contrast and intalize it to 0

Start nested for loop with variables i and j where each loop iterates max plus one number of times

Increment contrast by the square absolute difference between i and j times the value of Normalized GLCM at (i,j)

End nested loop

Print contrast value

End function

Start void function Energy with double pointer type double NormalziedGLCM and integer max

Declare variable type double Energy and intalize it to 0

Start nested for loop with variables i and j where each loop iterates max plus one number of times

Increment energy by the square of the value of Normalized GLCM at (i,j)

End nested loop

Print energy value

End function

Start void function Homogenity with double pointer type double NormalziedGLCM and integer max

Declare variable type double homogeneity and intalize it to 0

Start nested for loop with variables i and j where each loop iterates max plus one number of times

Increment homogeneity by value of Normalized GLCM at (i,j)devided by 1 plus the square absolute difference between i and j

End nested loop

Print energy value

End function

Start importImage function with parameter reference location to integers row, columns and max

Declare double pointer type int

Declare ifstream variable file

Declare string variable nameFile

Display a message asking the user for file name input (or 0 if they want to exit the program)

Input value from the user and store it in nameFile

Open file using the nameFile string variable

Start while loop that keeps iterating as long as opening file failed

If nameFile is equal to 0

Exit program

Display error message that no file with that name exist and asking the user to input another name

Input value from the user and store it in nameFile

Open file using the nameFile string variable

```
      End Loop

      Input values from file into columns and rows

      Initialize image pointer to point to an array of pointers type int with
size rows

      Start for loop that iterate rows number of time with variable i

            Set pointer in location i of image array of pointer to point to
an array of integers size columns

            Start loop that iterate columns number of time with variable j

                  Input from file into the value of image with location i,j

                  If value of image with location i,j greater than max

                        Then set max to equal this value

            End loop

      End loop

      Close file

      return double pointer variable image

End function
```

Step 4: Implementation

```
/*
Name: Yousef Al-Jazzazi
netID: ya2225
Asssignment 2 for CPE class with prof. Mohammad Eid
Date: November 13th 2022
Project name:  Computer Engineering Case Study – Image Features Extraction using Gray Level Cooccurrence
Matrix (GLCM)
Description: The program takes a file that contains the pixel information of an image and use changes it to Gray
level coocurrence Matrix
to find some of the features of the image

*/

#include <iostream>
#include <cmath>
#include <fstream>

using namespace std;


//Initilizing functions:
```

```
void createMatrix(int** GLCM, int** image, int rows, int columns);
void createNormalized(int** GLCM, double** normalized, int max);

void printMatrix(int** GLCM, int max);

void contrast(double** NormalizedGLCM, int max);
void Homogenity(double** NormalizedGLCM, int max);
void Energy(double** NormalizedGLCM, int max);
// A helper function to import the image, take the values into a 2D array, and returns the pointer for the 2D
array
int** importImage(int& rows, int& columns, int& max);

int main() {
	//Initalize variables
	bool repeat = true, GLCMCreated = false;
	int **image = NULL, **GLCM = NULL;
	double** normalizedGLCM = NULL;

	int max(0), rows(0), columns(0);
	string choice = "";

	//Explain the program for the user and ask for the file- inside importImage function
	cout << "Hello User!!!\nThis program is designed to take input file that has the values in pixels of an
image"
		<< " and creat a GLCM matrix and then use the GLCM matrix to find some of the attributes of
the image" << endl;
	image = importImage(rows, columns, max);


	//Initialize GLCM and normalizedGLCM with dimension max +1, and initalize all the values to be 0
	GLCM = new int* [max + 1];
	normalizedGLCM = new double* [max + 1];
	for (int i = 0; i < max + 1; i++) {
		GLCM[i] = new int[max + 1] {0};
		normalizedGLCM[i] = new double[max + 1] {0};
	}

	//The menu  will be called continuesly until repeat is false which is when the user choice is 4
	while (choice != "4") {
		cout << "There are 3 functions of the program."
			<< "\nPress 1 to create and normalize GLCM matrix"
			<< "\nPress 2 Compute the statistical parameters of the texture"
			<< "\nPress 3 Print the GLCM matrix"
			<< "\nPress 4 to exit the code" <<endl;

		//I choose the combination of string and if statement rather than an switch statement because
		//in the case a user input a string in an int, it will result in a failed comand with an infinite loop

		cin >> choice;
		if (choice == "1") {
```

```
                createMatrix(GLCM, image, rows, columns);
                createNormalized(GLCM, normalizedGLCM, max);
                GLCMCreated = true;
                cout << "Done\n\n" << endl;
        }
        else if (choice == "2")
                //Only call function if GLCM was created
                if (GLCMCreated) {
                        contrast(normalizedGLCM, max);
                        Energy(normalizedGLCM, max);
                        Homogenity(normalizedGLCM, max);
                }
                else
                        cout << "Please create GLCM before calculating attributes" << endl;
        else if (choice == "3") {
                //Only Print GLCM if GLCM was created
                if (GLCMCreated)
                        printMatrix(GLCM, max);
                else
                        cout << "Please create GLCM before printing" << endl;
        }
        else if (choice == "4") {
                cout << "Thanks for using our porgram" << endl;
        }
        else
                cout << "Please input one of the options" << endl;
    }


        //Freeing the image 2D array
        for (int i = 0; i < rows; i++)
                delete[] image[i];
        delete[] image;
        //Freeing the GLCM and normalizedGLCM arrays
        for (int i = 0; i < max + 1; i++)
                delete[] GLCM[i], normalizedGLCM[i];
        delete[] GLCM, normalizedGLCM;


    return(0);
}


void createMatrix(int** GLCM, int** image,int rows, int columns) {

    //Using the maximum found in the image 2D array to initalize a GLCM 2D array where all values are
zero
    //Traverse through the image 2D array and imputting the nessessary values into GLCM 2D array
    for (int i = 0; i < rows ; i++) {
        for (int j = 0; j < columns - 1; j++) {
            ++GLCM[image[i][j]][image[i][j + 1]];
```

```
		}
	}

}

void createNormalized(int** GLCM, double** NormalizedGLCM, int max) {
	//Find the sum by traversing through GLCM
	double sum(0);
	for (int i = 0; i < max + 1; i++) {
		for (int j = 0; j < max + 1; j++) {
			sum += GLCM[i][j];

		}
	}

	//Creating NormalizedGLCM
	for (int i = 0; i < max + 1; i++) {
		for (int j = 0; j < max + 1; j++) {
			//If sum is zero then it will print this message
			if (sum == 0) {
				cout << "Normalized was not created successfully" << endl;
				return;
			}
			 NormalizedGLCM[i][j]= GLCM[i][j] / sum;
		}
	}

}

void printMatrix(int** GLCM, int max) {
	cout << "GLCM matrix:\n\n";
	//Printing GLCM matrix
	for (int i = 0; i < max + 1; i++) {
		for (int j = 0; j < max + 1; j++) {
			cout << GLCM[i][j] << " ";

		}
		cout << "\n";
	}

}

//A function that Finds the contrast and prints it
void contrast(double** NormalizedGLCM, int max) {

	double contrast(0);
	for (int i = 0; i < max + 1; i++) {
		for (int j = 0; j < max + 1; j++) {
			contrast += pow(abs(i - j), 2) * NormalizedGLCM[i][j];

		}
```

```
		}
		cout << "\nContrast: " << contrast << endl;

}

//A function that Finds the Energy and prints it
void Energy(double** NormalizedGLCM, int max) {

		double energy(0);
		for (int i = 0; i < max + 1; i++) {
			for (int j = 0; j < max + 1; j++) {
				energy += NormalizedGLCM[i][j] * NormalizedGLCM[i][j];

			}

		}
		cout << "Energy: " << energy <<  endl;
}

//A function that Finds the homogenity and prints it
void Homogenity(double** NormalizedGLCM, int max) {
		double homogenity(0);
		for (int i = 0; i < max + 1; i++) {
			for (int j = 0; j < max + 1; j++) {

				homogenity += (NormalizedGLCM[i][j] / (1 + pow(abs(i - j), 2)));
			}

		}
		cout << "Homogeneity: " << homogenity << endl;
}

//the function takes rows, columns, and max variables as refrences to implement changing on them
//We passed them as refrence rather than creating as pointers as pointers gives felxibility(with complexity) that
is not needed in this case
int** importImage(int& rows, int& columns, int& max) {
		int** image;
		ifstream file;
		string nameFile;

		cout << "Before starting, please enter the name of the first file: (Press 0 to exit program)" << endl;
		cin >> nameFile;
		file.open(nameFile, ios::in);
		//if the name inputed by the user doesnt work, keep asking the user for a valid name.
		//The user can opt out of the porgram by inputting a 0
		while (file.fail()) {
			if (nameFile == "0") {
				cout << "Thanks for using our porgram" << endl;
				exit(1);
			}
```

```
                cerr << "Error. The file doesn't exist. Please try again" << endl;
                cin >> nameFile;
                file.open(nameFile, ios::in);

        }

        file >> columns >> rows;

        image = new int*[rows];
        //Inputting the values into a 2D image array and finding the maximum value
        for (int i = 0; i < rows; i++) {
                image[i] = new int[columns];
                for (int j = 0; j < columns; j++) {
                        file >> image[i][j];
                        if (image[i][j] > max)
                                max = image[i][j];
                }
        }
File.close();
        // return a pointer for the 2D array
        return image;

}
```

Step 5: Test Cases and verification

Errors: I noticed that the value of homogeneity calculated through website mathworks gave a different value. After further inspection, it appeared that the website used a different equation for homogeneity. The values of homogeneity produced by my program was compared to other peers, and it appears that all of us got the same value- verifying that my program runs correctly

Test #1:

File: “image.txt”

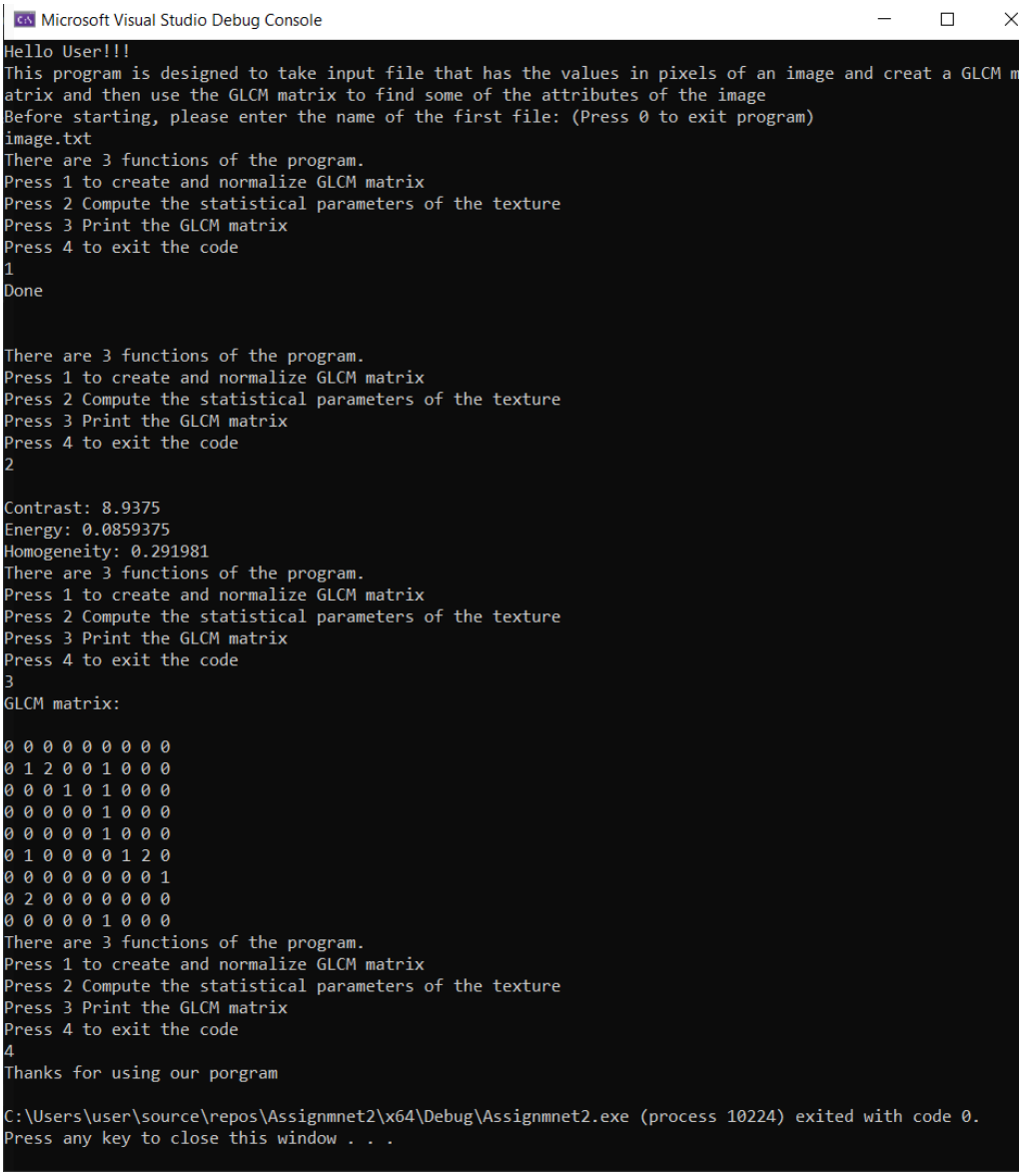
5 4

1 1 5 6 8

2 3 5 7 1

4 5 7 1 2

8 5 1 2 5



Test #2:

File: “image.txt”

10 10
1 5 1 5 3 1 3 3 3 2
1 4 0 3 3 1 1 1 3 0
3 4 3 0 1 1 0 2 5 0
2 0 2 1 3 0 1 5 1 1
1 2 4 3 4 4 0 5 2 2
3 2 4 2 3 5 1 1 2 5
1 4 0 5 3 4 4 0 2 1
4 0 0 0 5 4 5 2 3 4
1 3 0 3 2 3 3 0 4 5
1 4 3 2 5 4 5 2 5 1

```
Hello User!!!
This program is designed to take input file that has the values in pixels of an image and creat a GLCM m
atrix and then use the GLCM matrix to find some of the attributes of the image
Before starting, please enter the name of the first file: (Press 0 to exit program)
gg.txt
Error. The file doesn't exist. Please try again
image.txt
There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
2
Please create GLCM before calculating attributes
There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
1
Done


There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
3
GLCM matrix:

2 2 3 2 1 3
1 5 2 4 3 3
1 2 1 3 2 4
5 2 4 4 4 1
5 0 1 3 2 3
1 4 3 2 2 0
There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
2

Contrast: 6.12222
Energy: 0.0358025
Homogeneity: 0.364651
```

Test # 3:

File: “image.txt”

3 3

1 2 3

9 5 4

6 6 7

```
Hello User!!!
This program is designed to take input file that has the values in pixels of an image and creat a GLCM m
atrix and then use the GLCM matrix to find some of the attributes of the image
Before starting, please enter the name of the first file: (Press 0 to exit program)
Yousef.txt
Error. The file doesn't exist. Please try again
0
Thanks for using our porgram

C:\Users\user\source\repos\Assignmnet2\x64\Debug\Assignmnet2.exe (process 22868) exited with code 1.
Press any key to close this window . . .
```

Test #4 :

File: “image.txt”

3 3

1 2 3

9 5 4

6 6 7

Microsoft Visual Studio Debug Console

```
Hello User!!!
This program is designed to take input file that has the values in pixels of an image and creat a GLCM m
atrix and then use the GLCM matrix to find some of the attributes of the image
Before starting, please enter the name of the first file: (Press 0 to exit program)
image.txt
There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
1
Done


There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
3
GLCM matrix:

0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 1 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 1 0 0 0 0 0
0 0 0 0 0 0 1 1 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 1 0 0 0 0
There are 3 functions of the program.
Press 1 to create and normalize GLCM matrix
Press 2 Compute the statistical parameters of the texture
Press 3 Print the GLCM matrix
Press 4 to exit the code
4
Thanks for using our porgram
```