Name: Ahmed Mohamed Ali Mohamed Badawy

Sec: 5C-T3

Id: 4200284

1. Clarify the following terms:

Dynamic range: It describes the ratio between the maximum and minimum intensities in the Image.

Image transformation: map points in one space to points in another.

Affine transformation: Used to correct geometric distortions that occur with non-ideal camera angles

Image translation: Translation is the shifting of an object's location.

Histogram: Histogram is a graph, which shows intensity distribution of an image.

Logarithmic transformation: Used to map a narrow range of dark input values into a wider range of output values

2. State the enhancement technique and its transformation function for each of the following cases:
   a. Obtain negative of an image with gray levels in the range [0, 255]
      Linear transformation(image negative)
      The transformation function used in image negative is S=intensity max − $\boldsymbol{r}$.

```
Mat src = imread("neg.png", 0)
for (int i = 0; I < src.rows; i++)
        for (int j = 0; j < src.cols; j ++)
                src.at<uchar>(I, j) = 255 – src.at<uchar>(I, j);
namedWindow("NEGATIVE", 0)
imshow("NEGATIVE", src)
```

   b. Highlight gray range of interest to a viewer.

      Gray level slicing
      - Transformation highlights range [A, B] of gray level and reduces all others to a constant level
        S = 255 if A > r <B, C $otherwise$
      - Transformation highlights range [A, B] but preserves all other levels
        S = 255 if A > r <B, input $otherwise$

```
Mat src = imread("neg.png", 0)
for (int i = 0; I < src.rows; i++)
        for (int j = 0; j < src.cols; j ++){
                if(src.at<uchar>(I, j) > 130 && src.at<uchar>(I, j) < 200)
                        src.at<uchar>(I, j) = 255;
        }
namedWindow("NEGATIVE", 0)
imshow("NEGATIVE", src)
```

c. Decrease image brightness.

Power transformation(gamma correction)
The transformation function used is $s=c*r\gamma$

```
Mat src = imread("img.png", 0)
Mat dst(src.rows, src.cols, CV_32FC1);
for (int i = 0; I < src.rows; i++)
        for (int j = 0; j < src.cols; j ++)
                dst.at<uchar>(I, j) = powf(src.at<uchar>(I, j), 1);
normalize(dst, dst, 0, 255, NORM_MINMAX);
convertScaleAbs(dst, dst);
namedWindow("OUTPUT", 0)
imshow("OUTPUT", dst)
```

d. Enhance low contrast dark or washed out images.

Histogram Equlization
Steps:
1. find intensity count(Histogram)
2. calculate PMF(probability mass function)
3. calculate CDF(cumulative distributive function)
4. new intensity=CDF*L-1

```
Mat src = imread("hist.jpg", 0);
Mat dst;
namedWindow("Src Image", 0);
imshow("Src Image ", src);
equalizeHist(src, dst);
namedWindow("histogram equlization", 0);
imshow("histogram equlization", dst);
```

3. A 4-bit 75x40 image has the following intensities; Find out the equalized histogram gray levels for the image and Comment on the result.

A 4-bit --> L = 2^4 = 16
M*N = 75*40 = 3000

Gray level	**count**	**PMF=count/(M*N)**	**CDF =** $\Sigma pr(rj)$	**EH = CDF * L -1**
8	520	0.173	0.173	2
9	420	0.140	0.313	4
10	1180	0.393	0.706	10
11	732	0.244	0.950	14
12	148	0.049	1.000	15

The histogram equalization stretch out the intensity range from 8:12 to become 2:15 which improve the image contrast

```
Mat src3 = imread("hist.jpg", 0);
Mat dst3;
namedWindow("Src Image", 0);
imshow("Src Image ", src3);
equalizeHist(src3, dst3);
namedWindow("histogram equlization", 0);
 imshow("histogram equlization", dst3);
```

4. For the following 3x3 image intensities find out its bit planes, which bit plane contain most of the significant visual information?
Since the max gray level is 7, then its 3-bit image

001	010	011
100	101	000
111	010	110

0	1	1
1	0	0
1	1	1

0	1	1
0	0	0
1	1	1

1	0	1
0	1	0
1	0	0

```
Mat src = imread("bit.png", 0)
for (int i = 0; I < src.rows; i++){
    for (int j = 0; j < src.cols; j ++){
            if(src.at<uchar>(I, j) & 64)   src.at<uchar>(I, j) = 255;
            else  src.at<uchar>(I, j) = 0;
    }
}
namedWindow("BIT_PLANE", 0)
imshow("BIT_PLANE", src)
```

5. Write a program to enhance the image shown, by one of the image enhancement techniques. Explain why? Show the results of running the program and Comment on the result.

```
Mat src = imread("img.jpg", 0);
Mat dst;
```

a)

<u>Using Rotate Transform By 90deg</u>

```
Mat R = getRotateMatrix2D(point2f(src.cols/2, src.rows/2), 90, 0)
wrapAffine(src, dst, R, src.size())
```

b)

sampling, aliasing, Moiré patterns, and image zooming and shrinking. The new material and the manner in which these two chapters were reorganized address directly the first two findings in the market survey mentioned above.

Chapters 3 though 6 in the current edition cover the same concepts as Chapters 3 through 5 in the previous edition, but the scope is expanded and the presentation is totally different. In the previous edition, Chapter 3 was devoted exclusively to image transforms. One of the major changes in the book is that image transforms are now introduced when they are needed. This allowed us to begin discussion of image processing techniques much earlier than before, further addressing the second finding of the market survey. Chapters 3 and 4 in the current edition deal with image enhancement, as opposed to a single chapter (Chapter 4) in the previous edition. The new organization of this material does not imply that image enhancement is more important than other areas. Rather, we used it as an avenue to introduce spatial methods for image processing (Chapter 3), as well as the Fourier transform, the frequency domain, and image filtering (Chapter 4). Our purpose for introducing these concepts in the context

<u>Using Skewing Transform To Straightening the image</u>

```
Point2f src_p[3];
Src_p[0] = Point2f(0,0);
Src_p[1] = Point2f(src.cols-1, 0);
Src_p[2] = Point2f(100, src.rows-1);
Point2f dst_p[3];
dst_p[0] = Point2f(0,0);
dst_p[1] = Point2f(src.cols-1, 0);
dst_p[2] = Point2f(0, src.rows-1);
```

```
Mat S = getAffineTransform(src_p, dst_p)
wrapAffine(src, dst, S, src.size())
```

c)

Using Translate Transform By 50 At X-axis, 100-axis At Y

```
Int tx = 50, ty = 100;
Mat T = (Mat_<float>(2,3)<< 1, 0, tx, 0, 1, ty)
wrapAffine(src, dst, T, src.size())
```

d)

Using Flip Transform around Y-axis

```
Flip(src, dst, 1)
```

Convert From GrayScale Into RGB

```
cv2.cvtColor(src, cv2.COLOR_GRAY2RGB)
```

e)

Using Histogram Equlization

```
equalizeHist(src, dst);
namedWindow("histogram equlization", 0);
imshow("histogram equlization", dst);
```