EXPT. NO:2
DATE: 06.02.2024

IMPLEMENTATION OF INTENSITY TRANSFORMATION TECHNIQUES USING MATLAB

AIM:

To implement the following intensity transformation techniques using MATLAB:

1. Image Negation
2. Power Law Transformations
3. Logarithmic Transformation

SOFTWARE USED:

MATLAB version 2014a.

THEORY:

Image transformation techniques are a set of methods used to modify the appearance or characteristics of digital images. These techniques are widely employed in various fields including computer vision, image processing, medical imaging, and digital art.

Image Negation: Image negation is a simple transformation that is applied to invert the pixelvalues of an image. In a typical grayscale image, each pixel value ranges from 0 to 255, where 0 represents black and 255 represents white. Negating the image involves subtracting each pixel value from the maximum value (255 in this case). This results in an image where darker areas become lighter and vice versa, effectively inverting the colors.

Power Law Transformation: Power law transformation, also known as gamma correction, is a nonlinear transformation applied to adjust the contrast of an image. It's particularly useful for enhancing images with low contrast or adjusting the brightness level.

$$Y = Cx^{\Gamma}$$

Logarithmic Transformation: Logarithmic transformation is another nonlinear transformation used for enhancing the contrast of an image, particularly for images with a wide range of intensity levels. The transformation function is given by:

$$Y = C \log(1+x)$$

The logarithmic function compresses the dynamic range of pixel values, enhancing the visibility of details in dark regions while suppressing high-intensity values.

OUTPUT:

1. IMAGE NEGATION

[Image type: RGB Image Format: .jpg Image Size: 315x420x3 uint8]

FUNCTION TABLE:

1	0	1
1	0	0
1	0	1

254	255	254
254	255	255
254	255	254

FUNCTION:

- uint8(x): Converts a double-precision variable to an 8-bit unsigned integer.

2. POWER LAW TRANSFORMATION

[Image type: RGB Image Format: .jpg Image Size: 315x420x3 uint8]

Gamma Value < 1

FUNCTION TABLE:

0	0	1
0	0	1
0	1	1

Gamma Value > 1

FUNCTION TABLE:

0	1	1
0	0	1
0	1	1

FUNCTION:

- uint8(x): Converts a double-precision variable to an 8-bit unsigned integer.

3. LOGARITHMIC TRANSFORMATION

[Image type: RGB Image Format: .jpg Image Size: 315x420x3 uint8]

FUNCTION TABLE

108	98	98
99	89	88
98	89	88

FUNCTION:

- uint8(x): Converts a double-precision variable to an 8-bit unsigned integer.

PROGRAM:

1. IMAGE NEGATION

```
clc;
clear all;
close all;
p= imread('Sample_img.jpg');
[x,y,z] =size(p);
if(z==1)
else
p= rgb2gray(p);
end
maxgray=max(max(p));
maxgray=im2uint8(maxgray);
p= im2double(p);
for i=1:x
for j= 1:y
pnegative(i,j)=maxgray-p(i,j);
end
end
intArray = uint8(p);
figure,imshowpair(p,pnegative,'montage');
title('Image Negation')
```

2. POWER LAW TRANSFORMATION

```
clc;
clear all;
close all;
i=imread('rgbimg.jpeg');
idouble=im2double(i);
r,c]=size(idouble)
c= input('Enter the value of c=');
e=input('Enter the value of Gamma=');
for i=1:r
for j=1:c
imout(i,j)= c* power(idouble(i,j),e);
end
end
mat = uint8(idouble);
figure,imshowpair(i,imout,'montage');
title(‘Power Law Transformation’);
```

3. LOGARITHMIC TRANSFORMATION

```
clc;
clear all;
close all;
p= imread('rgbimg.jpeg');
[x,y,z] = size(p);
if(z==1);
else
p=rgb2gray(p);
end
c=input('Enter the value of c= ');
[m,n]=size(p);
for x=1:m
for y=1:n
m=double(p(x,y));
z(x,y)=c.*log10(1+m);
end
end
array = uint8(p); figure,imshowpair(p,z,'montage');
title('Logarithmic Transformation');
```

RESULT:

Image transformation techniques image negation, power law transformation and logarithmic transformation using MATLAB was successfully executed.