

多媒體概論與數位科技概論期末作業 —— J P E G 壓縮

指導老師——蔡崇煒

班級資工二——第八組

組長——胡卜升(B0343029)

組員——周冠伶(B0343001)

——陳柏睿(B0343023)

——劉威伸(B0343035)

——何昭儀(B0343042)

程式語言：Java 1.6.0_45

編譯程式：Eclipse、CMD、NotePad++

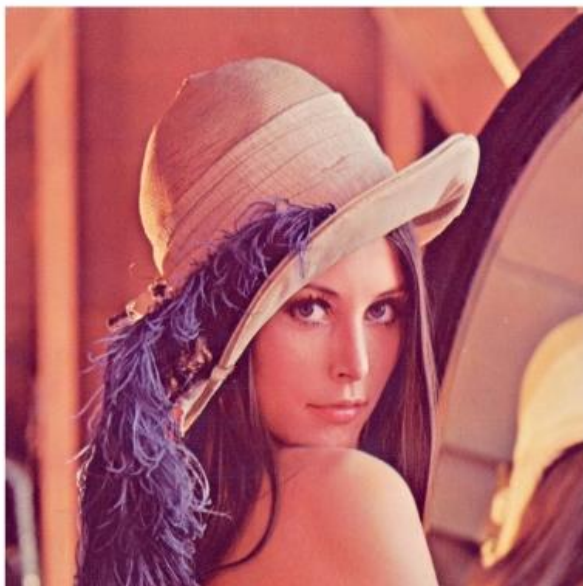
作業系統：Windows 8.1（64 位元）、Windows XP SP3（32 位元）

開發日期：2015 年 12 月 2 日至 2016 年 1 月 10 日

主編：周冠伶(B0343001)

共同討論：陳柏睿(B0343023)、胡卜升(B0343029)、郭家瑋(B0343025)

執行結果



執行前



執行後

1	[261	5	4	0	0	0	-8	10	246	18	2	6	-8	12	0	-10	225	8	-3	-9	0	6	0	0	213	2
2	[8	-2	2	-6	4	0	0	0	11	-2	2	7	-4	0	10	0	10	-2	0	-6	0	9	10	-1	4	0
3	[-6	0	-4	4	0	0	0	0	-3	2	-3	-4	0	-1	0	0	-3	0	0	4	0	0	-1	0	-4	-1
4	[2	0	4	0	0	0	0	0	-2	0	0	0	0	0	-1	-1	-1	0	0	0	0	0	0	2	0	
5	[0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	
6	[0	0	-1	0	0	0	0	1	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
7	[0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	[0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
9	[228	6	-6	0	-1	0	0	0	232	6	2	0	8	6	0	0	225	4	-2	0	-4	-1	0	0	210	-6
10	[-4	-2	-4	0	4	0	0	0	4	4	2	3	0	0	0	9	-2	-1	-2	0	0	0	0	-4	5	
11	[5	-4	3	0	-6	0	0	0	1	2	-3	4	0	1	0	0	-4	2	-3	-1	0	1	1	0	2	-1
12	[4	-6	4	0	0	0	0	0	2	0	3	0	8	0	0	0	2	3	0	0	0	0	0	0	2	4
13	[0	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0	3	0	1	-10	0	0	0	0	0	0
14	[1	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	-1	6	1	0	0	0	0	0	0	6
15	[0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	
16	[0	0	0	-1	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	[225	4	-2	0	0	1	0	0	234	-3	2	0	4	0	1	0	231	5	2	-3	4	0	-8	0	233	-2
18	[2	2	-6	4	4	0	0	0	-4	3	2	0	-4	0	0	0	-3	-5	-3	-2	4	0	0	0	-6	0
19	[-2	3	4	0	0	0	0	0	-2	2	-3	0	0	0	1	-9	-2	-4	-3	4	0	0	0	0	-2	-2
20	[2	0	1	0	0	0	0	0	-2	0	-4	1	8	0	0	0	-2	0	-4	0	0	0	0	0	-2	-3
21	[0	-4	-1	0	0	0	0	0	3	-1	0	-1	0	0	0	0	3	-4	0	0	0	0	0	0	0	0
22	[0	0	0	0	1	1	0	0	0	-6	0	0	0	0	0	0	4	6	0	0	0	0	1	0	-4	0
23	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	1	0	0	9	0
24	[0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
25	[231	-5	2	0	-4	0	0	0	246	2	-6	0	4	0	-8	0	248	-1	3	0	4	0	-8	0	288	-2
26	[-12	-4	6	0	-4	0	1	0	-4	2	2	-3	0	0	0	0	-10	6	0	0	4	1	0	1	-20	-6

DCT後Y値

1	[-178	4	-1	1	0	0	1	-1	-195	5	1	1	0	0	-1	-1	-216	8	-2	-4	2	0	3		
2	[13	1	0	-1	0	0	0	12	0	0	0	0	0	-1	0	4	3	0	1	0	0	1	8	-5	
3	[-4	0	0	0	0	0	0	-5	0	0	0	0	0	0	0	-4	0	0	0	0	0	1	-4	0	
4	[0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	[0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	[-1	-1	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	[0	1	-1	0	0	0	0	1	0	0	0	0	-1	0	0	1	0	0	0	0	0	0	0	1	
8	[0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	
9	[-218	-3	1	0	0	0	0	0	-222	5	1	0	0	0	0	-1	-223	2	3	-2	0	0	-2	0	
10	[8	-3	1	0	0	0	0	6	4	1	0	0	0	0	-1	4	0	0	0	0	0	0	3	0	
11	[0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	-2	1	0	0	0	0	7	-2	
12	[-1	0	-1	0	0	0	0	1	-1	0	0	0	-1	0	0	7	0	0	-1	0	0	0	7	1	
13	[1	0	-1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	-1	
14	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-1	0	
15	[0	0	0	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	
16	[1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	
17	[-218	3	0	1	0	0	0	-1	-228	1	1	1	0	-1	0	0	-219	-5	0	-1	-1	0	0	0	
18	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-6	-2	1	0	0	1	0	0	-1	
19	[5	0	0	-1	0	1	0	3	1	1	0	0	0	0	0	-1	8	-1	0	0	0	0	0	-9	
20	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	
21	[-1	0	1	-1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	-1
22	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	1	-1	0	
23	[0	0	-2	0	0	0	1	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	
24	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	-1	-1	0
25	[-217	-1	0	0	0	0	0	0	-227	5	2	0	1	-1	0	-1	-218	-6	-1	-1	-1	0	-1	0	
26	[4	-5	0	0	0	0	1	2	4	0	0	0	0	0	0	-1	3	-5	0	0	0	0	8	3	

DCT後U値

1	[366	-5	1	0	0	0	0	1	377	2	1	0	0	0	0	375	-6	1	0	0	0	0	1	381	3
2	[-6	-2	0	0	0	0	0	0	-12	5	1	1	0	0	0	-1	-12	-1	1	0	0	0	1	-6	-1
3	[5	0	0	0	0	0	0	0	4	0	0	0	0	0	0	4	0	-1	-1	0	0	0	4	-1	0
4	[0	0	0	-1	0	0	0	0	-1	0	-1	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0
5	[-1	0	-1	0	0	0	0	0	1	0	1	0	0	0	0	-1	0	0	1	0	0	0	0	0	0
6	[0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	-1	0	0	0	0	-1	0
7	[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
8	[0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
9	[387	-1	1	0	0	0	0	0	394	-2	-1	0	0	0	0	394	-1	0	1	0	0	0	0	393	0
10	[-4	0	0	0	0	0	0	0	-2	0	-1	0	0	0	0	-2	0	0	0	0	0	0	0	-3	1
11	[0	1	0	0	0	0	0	0	1	-1	-1	0	0	0	0	4	2	1	0	0	0	0	0	-4	1
12	[1	0	0	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	[0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	1	0	0	0	0	0	-1
14	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	-1
15	[0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0
16	[0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
17	[393	0	0	0	0	0	0	0	396	-3	0	1	0	-1	0	392	-2	0	0	0	0	0	0	381	6
18	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	2	0	-1	0	0	0	0	0	1
19	[0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	1	0	0	0	4	1	0
20	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	-1
21	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0
22	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	0	0	0	0	0	0	0	1
23	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
24	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
25	[392	-1	-1	0	0	0	1	0	395	0	-1	0	0	0	0	384	0	1	0	0	0	0	0	369	7
26	[0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	5	-1

```

import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import java.io.*;
import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;

public class DCTandVQ extends Frame implements WindowListener
{
    //function
    static int DCT_2D( int u, int v, int YUV )
    {
        double sum = 0;
        double Cv, Cu;
        double [][ ] intensity;

        //Declaration &Distinguish intensity from YUV
        intensity = new double[N][N];
        for( int a=0; a<N; a++ )
        {
            for( int b=0; b<N; b++ )
            {
                if( YUV==1 )
                {
                    intensity[a][b] = intensity_Y[u-u%N+a][v-v%N+b];
                }
                else if( YUV==2 )
                {
                    intensity[a][b] = intensity_U[u-u%N+a][v-v%N+b];
                }
                else if( YUV==3 )
                {
                    intensity[a][b] = intensity_V[u-u%N+a][v-v%N+b];
                }
            }
        }

        //Set C(u) and C(v)
        if( v%N==0 )
        {
            Cv = ( Math.sqrt(2) / 2 );
        }
        else
        {
            Cv = 1;
        }
        if( u%N==0 )
        {
            Cu = ( Math.sqrt(2) / 2 );
        }
    }
}

```

```

else
    {
        Cu = 1;
    }

for( int i=0; i<N; i++ )
{
    for( int j=0; j<N; j++ )
    {
        sum +=
        ( ( Math.cos(((i+0.5)*(u%N)*(Math.PI)/N))*Math.cos(((j+0.5)*(v%N)*(Math.PI)/N)) ) *intensity[i][j] );
    }
}

return ( (int)(Math.round( (2*Cu*Cv/N) *sum ) ) );
}

static int VQ( int value, int u, int v, int YUV )
{
    int Output = 0;

    //Declaration &Distinguish VQ_Table from YUV
    int [][] VQ_Table_Y =
    {
        {16, 11, 10, 16, 24, 40, 51, 61},
        {12, 12, 14, 19, 26, 58, 60, 55},
        {14, 13, 16, 24, 40, 57, 69, 56},
        {14, 17, 22, 29, 51, 87, 80, 62},
        {18, 22, 37, 56, 68, 109, 103, 77},
        {24, 35, 55, 64, 81, 104, 113, 92},
        {49, 64, 78, 87, 103, 121, 120, 101},
        {72, 92, 95, 98, 112, 100, 103, 99}
    };

    int [][] VQ_Table_UV =
    {
        {17, 18, 24, 47, 99, 99, 99, 99},
        {18, 21, 26, 66, 99, 99, 99, 99},
        {24, 26, 56, 99, 99, 99, 99, 99},
        {47, 66, 99, 99, 99, 99, 99, 99},
        {99, 99, 99, 99, 99, 99, 99, 99},
    }

```

```
{99, 99, 99, 99, 99, 99, 99, 99},  
{99, 99, 99, 99, 99, 99, 99, 99},  
{99, 99, 99, 99, 99, 99, 99, 99}
```

```
};
```

```
if( YUV==1 )  
    {    Output = (int)( Math.round((((double)(value))/VQ_Table_Y[u%N][v%N])) );    }  
else if( YUV==2 || YUV==3 )  
    {    Output = (int)( Math.round((((double)(value))/VQ_Table_UV[u%N][v%N])) );    }
```

```
return Output;
```

```
}
```

```
static int InverseDCT_2D( int i, int j, int YUV )
```

```
{
```

```
    double sum = 0;
```

```
    double Cv, Cu;
```

```
    int [][] F;
```

```
    //Declaration &Distinguish F from YUV
```

```
    F = new int[N][N];
```

```
    for( int a=0; a<N; a++ )
```

```
    {
```

```
        for( int b=0; b<N; b++ )
```

```
        {
```

```
            if( YUV==1 )
```

```
                {    F[a][b] = InverseVQ_Value_Y[i-i%N+a][j-j%N+b];    }
```

```
            else if( YUV==2 )
```

```
                {    F[a][b] = InverseVQ_Value_U[i-i%N+a][j-j%N+b];    }
```

```
            else if( YUV==3 )
```

```
                {    F[a][b] = InverseVQ_Value_V[i-i%N+a][j-j%N+b];    }
```

```
        }
```

```
    }
```

```
    for( int u=0; u<N; u++ )
```

```
    {
```

```

for( int v=0; v<N; v++ )
{
    //Set C(u) and C(v)
    if( v==0 )
        {    Cv = ( Math.sqrt(2) /2 );}
    else
        {    Cv = 1;    }
    if( u==0 )
        {    Cu = ( Math.sqrt(2) /2 );    }
    else
        {    Cu = 1;    }

    sum += ( Cv *Cu *( Math.cos(( i%N)+0.5 )*u*(Math.PI)/N) )
*( Math.cos((( j%N)+0.5 )*v*(Math.PI))/N) ) *F[u][v] );
}
}

return ( (int)(Math.round( 2*sum/N )) );
}

static int InverseVQ( int value, int u, int v, int YUV )
{
    int Output = 0;

    //Declaration &Distinguish VQ_Table from YUV
    int [][] VQ_Table_Y =
        {
            {16, 11, 10, 16, 24, 40, 51, 61},
            {12, 12, 14, 19, 26, 58, 60, 55},
            {14, 13, 16, 24, 40, 57, 69, 56},
            {14, 17, 22, 29, 51, 87, 80, 62},
            {18, 22, 37, 56, 68, 109, 103, 77},
            {24, 35, 55, 64, 81, 104, 113, 92},
            {49, 64, 78, 87, 103, 121, 120, 101},
            {72, 92, 95, 98, 112, 100, 103, 99}
        };

    int [][] VQ_Table_UV =

```

```

        {
            {17, 18, 24, 47, 99, 99, 99, 99},
            {18, 21, 26, 66, 99, 99, 99, 99},
            {24, 26, 56, 99, 99, 99, 99, 99},
            {47, 66, 99, 99, 99, 99, 99, 99},
            {99, 99, 99, 99, 99, 99, 99, 99},
            {99, 99, 99, 99, 99, 99, 99, 99},
            {99, 99, 99, 99, 99, 99, 99, 99},
            {99, 99, 99, 99, 99, 99, 99, 99}
        }

    };

    if( YUV==1 )
        {    Output = value*VQ_Table_Y[u%N][v%N]; }
    else if( YUV==2 || YUV==3 )
        {    Output = value*VQ_Table_UV[u%N][v%N];    }

    return Output;
}

```

```

//Declaration
static DCTandVQ DctAndVq;
static JFileChooser chooser;
static BufferedImage img;
static String Title;
static String Path;
static int N;
static int imgH, imgW;
static int[][] InverseVQ_Value_Y;
static int[][] InverseVQ_Value_U;
static int[][] InverseVQ_Value_V;
static int[][] DCT_Value_Y;
static int[][] DCT_Value_U;
static int[][] DCT_Value_V;
static int[][] InverseDCT_Value_Y;
static int[][] InverseDCT_Value_U;
static int[][] InverseDCT_Value_V;
static int[][] pixel, red, green, blue;

```



```

static int[][] New_red, New_green, New_blue;
static double[][] intensity_Y, intensity_U, intensity_V;
static FileWriter writer_DCT_Y, writer_DCT_U, writer_DCT_V;
static Color cr, New_cr;

//Constructor
DCTandVQ()
{
    //Set Window
    this.setSize( imgW-1+100+imgW+50, imgH-1+100 );
    this.setLocation( 200, 100 );
    this.setLayout( null );
    this.setVisible( true );
    this.setTitle( Title );

    this.addWindowListener( this );
}

public static void main( String[] arg )
{
    try
    {
        do
        {
            System.out.println("Please choose a file(.jpg): \nNotice: image pixels must be
multiple of 8");

            //Declaration
            chooser = new JFileChooser();

            //Limit File Name Extension by .jpg
            chooser.setAcceptAllFileFilterUsed( false );
            chooser.addChoosableFileFilter( new FileNameExtensionFilter( "JPG(*.jpg)",
"jpg" ) );

            //Set Title of Constructor and Path of Image
            if( chooser.showOpenDialog( null ) == JFileChooser.APPROVE_OPTION )
            {

```

```

        Path = chooser.getSelectedFile().getAbsolutePath();
        Title = chooser.getSelectedFile().getName();
    }
    else
    {
        System.out.println( "Something error." ); }

//Set Image Path, Height, and Width
img = ImageIO.read( new File ( Path ) );
imgH = img.getHeight();
imgW = img.getWidth();

}while( ( (imgH%8)!=0 ) || ( (imgW%8)!=0 ) );

//Create Aarray and Color
pixel = new int[imgH][imgW];
red    = new int[imgH][imgW];
green  = new int[imgH][imgW];
blue   = new int[imgH][imgW];
New_red    = new int[imgH][imgW];
New_green  = new int[imgH][imgW];
New_blue   = new int[imgH][imgW];
intensity_Y = new double[imgH][imgW];
intensity_U = new double[imgH][imgW];
intensity_V = new double[imgH][imgW];
DCT_Value_Y = new int[imgH][imgW];
DCT_Value_U = new int[imgH][imgW];
DCT_Value_V = new int[imgH][imgW];
InverseVQ_Value_Y = new int[imgH][imgW];
InverseVQ_Value_U = new int[imgH][imgW];
InverseVQ_Value_V = new int[imgH][imgW];
InverseDCT_Value_Y = new int[imgH][imgW];
InverseDCT_Value_U = new int[imgH][imgW];
InverseDCT_Value_V = new int[imgH][imgW];

//Set Pixel, RGB, and Intensities Value
for( int x=0; x<imgH; x++ )
{
    for( int y=0; y<imgW; y++ )

```

```

{
    //Set Pixel Value; RGB Binary to Hexadecimal Convert
    pixel[x][y]=img.getRGB( y, x );
    red[x][y]   = ( pixel[x][y] >> 16 ) &0xFF;
    green[x][y] = ( pixel[x][y] >> 8  ) &0xFF;
    blue[x][y]  = ( pixel[x][y] >> 0  ) &0xFF;

    //Set Intensities Value; elapse ( -128 ~ 127 )
    intensity_Y[x][y]=
    ( ( (double)(red[x][y]) ) *0.2990 )+( ( (double)(green[x][y]) ) *0.5870 )+( ( (double)(blue[x][y]) ) *0.1140 )
    -128;

    intensity_U[x][y]= ( ( (double)(red[x][y]) ) * -0.169 ) -
    ( ( (double)(green[x][y]) ) *0.3310 )+( ( (double)(blue[x][y]) ) *0.5000 );
    intensity_V[x][y]= ( ( (double)(red[x][y]) ) *0.5000 ) -
    ( ( (double)(green[x][y]) ) *0.4190 )-( ( (double)(blue[x][y]) ) *0.0810 );
}
}

N = 8;
try
{
    //create Files.txt
    writer_DCT_Y = new FileWriter( new File( "DCT_Value_Y.txt" ) );
    writer_DCT_U = new FileWriter( new File( "DCT_Value_U.txt" ) );
    writer_DCT_V = new FileWriter( new File( "DCT_Value_V.txt" ) );

    //Input the content
    for( int u=0; u<imgH; u++ )
    {
        for( int v=0; v<imgW; v++ )
        {
            DCT_Value_Y[u][v] = DCT_2D( u, v, 1 );
            DCT_Value_U[u][v] = DCT_2D( u, v, 2 );
            DCT_Value_V[u][v] = DCT_2D( u, v, 3 );

            InverseVQ_Value_Y[u][v] = InverseVQ( VQ( DCT_Value_Y[u][v], u,
v, 1 ), u, v, 1 );

```

```

InverseVQ_Value_U[u][v] = InverseVQ( VQ( DCT_Value_U[u][v], u,
v, 2 ), u, v, 2 );

InverseVQ_Value_V[u][v] = InverseVQ( VQ( DCT_Value_V[u][v], u,
v, 3 ), u, v, 3 );

```

```

//elapse Y to 0~255; U,V -128~127

```

```

InverseDCT_Value_Y[u][v] = InverseDCT_2D( u, v, 1 ) +128;

```

```

InverseDCT_Value_U[u][v] = InverseDCT_2D( u, v, 2 );

```

```

InverseDCT_Value_V[u][v] = InverseDCT_2D( u, v, 3 );

```

```

if( v==0 )

```

```

{
    writer_DCT_Y.write( "[\t" );
    writer_DCT_U.write( "[\t" );
    writer_DCT_V.write( "[\t" );
}

```

```

writer_DCT_Y.write( DCT_Value_Y[u][v]+"\\t" );

```

```

writer_DCT_U.write( DCT_Value_U[u][v]+"\\t" );

```

```

writer_DCT_V.write( DCT_Value_V[u][v]+"\\t" );

```

```

//\\n .txt no change; .doc line wrap

```

```

//\\r .txt get space; .doc line wrap

```

```

//\\r\\n .txt and .doc line wrap

```

```

//\\n\\r .txt get space; .doc two line wrap

```

```

if( v==imgW-1 )

```

```

{
    writer_DCT_Y.write( "]\r\n" );
    writer_DCT_U.write( "]\r\n" );
    writer_DCT_V.write( "]\r\n" );
}

```

```

}

```

```

}

```

```

//Close FileOutputStream

```

```

writer_DCT_Y.close();

```

```

writer_DCT_U.close();

```

```

writer_DCT_V.close();

```

```

    }

    //print out the error reason and location
    catch( Exception e )
    {
        e.printStackTrace();
    }

    //Create Constructor
    DctAndVq = new DCTandVQ();
}

catch( IOException e )
{
    System.out.println( "Something error." );
}

//Override
public void paint( Graphics g )
{
    for( int x=0; x<imgH; x++ )
    {
        for( int y=0; y<imgW; y++ )
        {
            //Showing the selected Image means computing ended.
            cr = new Color( red[x][y], green[x][y], blue[x][y] );
            g.setColor( cr );
            g.drawLine( y+50, x+50, y+50, x+50 );

            New_red[x][y] =
(int)(InverseDCT_Value_Y[x][y]+InverseDCT_Value_V[x][y]*1.1400);
            if( New_red[x][y] > 255 )
            {
                New_red[x][y] = 255;
            }
            else if( New_red[x][y]<0 )
            {
                New_red[x][y] = 0;
            }
            New_green[x][y] = (int)(InverseDCT_Value_Y[x][y]-
InverseDCT_Value_U[x][y]*0.3940-InverseDCT_Value_V[x][y]*0.5810);
            if( New_green[x][y] > 255 )
            {
                New_green[x][y] = 255;
            }
            else if( New_green[x][y] < 0 )
            {
                New_green[x][y] = 0;
            }

```

```

        New_blue[x][y] =
(int)(InverseDCT_Value_Y[x][y]+InverseDCT_Value_U[x][y]*2.0320);
        if( New_blue[x][y] > 255 )
            {      New_blue[x][y] = 255;  }
        else if( New_blue[x][y] < 0 )
            {      New_blue[x][y] = 0;    }
        New_cr = new Color( New_red[x][y], New_green[x][y], New_blue[x][y] );
        g.setColor( New_cr );
        g.drawLine( y+50+imgW+50, x+50, y+50+imgW+50, x+50 );

    }
}

```

```

//WindowsListener
public void windowActivated( WindowEvent e )
{    }

public void windowClosed( WindowEvent e )
{    }

public void windowClosing( WindowEvent e )
{    dispose(); }

public void windowDeactivated( WindowEvent e )
{    }

public void windowDeiconified( WindowEvent e )
{    }

public void windowIconified( WindowEvent e )
{    }

public void windowOpened( WindowEvent e )
{    }
}

```

//Date: 2015/12/02-2011/1/10

//Writer: Chou-Kuan-Lin(B0343001)

//Co-discusser: Guo-Jia-Wei(B0343025), Chen-Bo-Rui(B0343023), Hu-Bu-Sin(B0343029), Chen-Wei-Ming(Teacher)

//All rights reserved

//Deliberately Outward flow must investigate.