package com.prac.prac;

import java.awt.\*;

import java.awt.image.BufferedImage;

import java.io.\*;

import java.util.\*;

import javax.imageio.ImageIO;

public class Exatraction {

private static BufferedImage orignal,answer;

public static void main(String[] args) throws IOException{

File orignal\_f=new File("download.jpg");

orignal = ImageIO.read(orignal\_f);

answer=imageHistogram(orignal);

writeImage("featureExtraction");

}

private static void writeImage(String output) throws IOException {

File file = new File(output+".jpg");

ImageIO.write(answer, "jpg", file);

}

private static int colorToRGB(int alpha, int red, int green, int blue) {

int newPixel = 0;

newPixel += alpha; newPixel = newPixel << 8;

newPixel += red; newPixel = newPixel << 8;

newPixel += green; newPixel = newPixel << 8;

newPixel += blue;

return newPixel;

}

public static BufferedImage imageHistogram(BufferedImage input) {

BufferedImage redGraph = new BufferedImage(input.getWidth(),input.getHeight(),input.getType());

for(int i=0; i<input.getWidth(); i++)

{

for(int j=0; j<input.getHeight(); j++)

{

int alpha =new Color(input.getRGB (i, j)).getAlpha();

int red = new Color(input.getRGB (i, j)).getRed();

int green = new Color(input.getRGB (i, j)).getGreen();

int blue = new Color(input.getRGB (i, j)).getBlue();

redGraph.setRGB(i, j, colorToRGB(alpha, 0,0,blue));

}

}

return redGraph;

}

}