Computer Vision Hw1 Report

- Discription
    - 1. Use B_PIX to write a program to generate
        - (a) upside-down lena.im
        - (b) right-side-left lena.im
        - (c) diagonally mirrored lena.im
    - 2. Use Photoshop to
        - (a) rotate lena.im 45 degrees clockwise
        - (b) shrink lena.im in half
        - (c) binarize lena.im at 128 to get a binary image
    - java code , eclipse project
    - input : put lena.bmp in the folder "input"
    - output : result folder "output"
    - photoshop 6.1
- Algorithm
    - Up Side Down
        - Step1.取得圖片的 RGB Matrix
        - Step2.對每一個 Colum,由外向內 Swap 首尾兩個 Pixel 的 RGB 值
    - Right Side Left
        - 同 Up Side Down 但 Row , Col 處理順序互換
    - Diagonally Mirror

- Up Side Down + Right Side Left
- Parameters (if any)
  - no
- Principal Code Fragment

main()   (src/hw1/BasicImageManipulation.java)

```java
public class BasicImageManipulation {
    public static final String inputFolder="input/";
    public static final String inputFileName="lena.bmp";
    public static final String outputFolder="output/";

    public static void main(String[] args) {

        //read image
        System.out.println("read img ...");
        BufferedImage lena = FileUtil.readImg(inputFolder+inputFileName);

        //image up side down
        System.out.println("img up side down ...");
        BufferedImage lena_upSideDown=ImgUtil.imgUpSideDown(lena);
        FileUtil.writeImg(lena_upSideDown,outputFolder+"lena_upSideDown.bmp");

        //image right side left
        System.out.println("img right side left ...");
        BufferedImage lena_rightSideLeft=ImgUtil.imgRightSideLeft(lena);
        FileUtil.writeImg(lena_rightSideLeft, outputFolder+"lena_rightSideLeft.bmp");

        //image diagonally mirror
        System.out.println("img diagonally mirror ...");
        BufferedImage lena_diag=ImgUtil.imgDiagonallyMirror(lena);
        FileUtil.writeImg(lena_diag, outputFolder+"lena_DiagMirror.bmp");

        System.out.println("done");

    }
}
```

2

Function Up Side Down (src/cv1.util.cv/ImgUtil.java)

```java
public static BufferedImage imgUpSideDown(BufferedImage bi){
    /*
     * image up side down
     * left right & col by col
     */
    BufferedImage result = new BufferedImage(bi.getWidth(), bi.getHeight(), bi.getType());
    for (int i = 0; i < bi.getWidth(); i++) {
        for (int j = 0; j < (bi.getHeight()+1)/2 ; j++) {
            int RGB_head=bi.getRGB(i, j);
            int RGB_tail=bi.getRGB(i, bi.getWidth()-j-1);
            result.setRGB(i, bi.getWidth()-j-1  , RGB_head);
            result.setRGB(i, j                  , RGB_tail);
        }
    }
    return result;
}
```

Function Right Side Left (src/cv1.util.cv/ImgUtil.java)

```java
public static BufferedImage imgRightSideLeft(BufferedImage bi){
    /*
     * image Right Side Left
     * top down & row by row
     */
    BufferedImage result = new BufferedImage(bi.getWidth(), bi.getHeight(), bi.getType());
    for (int i = 0; i < bi.getHeight(); i++) {
        for (int j = 0; j < (bi.getWidth()+1)/2 ; j++) {
            int RGB_head=bi.getRGB(j              , i);
            int RGB_tail=bi.getRGB(bi.getWidth()-j-1, i);
            result.setRGB(bi.getWidth()-j-1 , i , RGB_head);
            result.setRGB(j                 , i , RGB_tail);
        }
    }
    return result;
}
```

Function Diagonally Mirrored (src/cv1.util.cv/ImgUtil.java)

```java
public static BufferedImage imgDiagonallyMirror(BufferedImage bi){
    /*
     * up side down
     * right side left
     */
    BufferedImage result = bi;
    result=imgUpSideDown(result);
    result=imgRightSideLeft(result);
    return result;
}
```

- Resulting Images
  - Program Result
    - Up Side Down

    - Right Side Left

- Diagonally Mirrored

  o Photoshop Result (photoshop 6.1)

    - Rotate (step:影像 -> 影像旋轉 -> 任意 -> 45 度)

■ Shrink Half (step:影像 -> 影像尺寸 -> 寬度&高度 256)

■ Binarize (step:影像 -> 調整 -> 臨界值 -> 128)