

# Computer Vision - Homework 6

## 開發環境

- OS: Windows 10 Pro
- Program Language: C# (with .Net Core 3.1)
- IDE: Visual Studio 2019
- Project: Console Application

## 程式說明

程式碼主要寫在 Program.cs，各題目程式皆已實作個別方法，由 Main entry 進行呼叫，答案結果儲存於 answers 資料夾。

各題目程式碼片段、參數及相關演算法說明如下：

(A). Yokoi connectivity number: 先將圖片轉為 binary Image 後進行 down sampling ( 64\*64 )，再針對每一個 pixel 進行 Yokoi 的演算法 ( 逆時針算出 q, r, s 並計算 r 及 q 的數量進行判斷 ) 計算出 number。

```
/// <summary>
/// 取得 Yokoi Connectivity Number Matrix
/// </summary>
private static int[,] GetYokoiConnectivityNumberMatrix(Bitmap srcImg)
{
    var result = new int[64, 64];

    for (var x = 0; x < srcImg.Width; x++)
    {
        for (var y = 0; y < srcImg.Height; y++)
        {
            var center = srcImg.GetPixel(x, y).R;
            if (center == 0)
            {
                result[x, y] = 0;
                continue;
            }

            // a1
            var b = center;
            var c = x < srcImg.Width - 1 ? srcImg.GetPixel(x + 1, y).R : 0;
            var d = x < srcImg.Width - 1 && y >= 1 ? srcImg.GetPixel(x + 1, y -
1).R : 0;
            var e = y >= 1 ? srcImg.GetPixel(x, y - 1).R : 0;

            var a1 = H(b, c, d, e);

            // a2
            c = e;
            d = x >= 1 && y >= 1 ? srcImg.GetPixel(x - 1, y - 1).R : 0;
            e = x >= 1 ? srcImg.GetPixel(x - 1, y).R : 0;

            var a2 = H(b, c, d, e);
```

```

        // a3
        c = e;
        d = x >= 1 && y < srcImg.Height - 1 ? srcImg.GetPixel(x - 1, y +
1).R : 0;
        e = y < srcImg.Height - 1 ? srcImg.GetPixel(x, y + 1).R : 0;

        var a3 = H(b, c, d, e);

        // a4
        c = e;
        d = x < srcImg.Width - 1 && y < srcImg.Height - 1 ?
srcImg.GetPixel(x + 1, y + 1).R : 0;
        e = x < srcImg.Width - 1 ? srcImg.GetPixel(x + 1, y).R : 0;

        var a4 = H(b, c, d, e);
        result[x, y] = F(a1, a2, a3, a4);
    }
}

return result;
}

/// <summary>
/// 根據 h(b, c, d, e) 的算式求得 q, r, s
/// </summary>
/// <param name="b"></param>
/// <param name="c"></param>
/// <param name="d"></param>
/// <param name="e"></param>
/// <returns></returns>
private static char H(int b, int c, int d, int e)
{
    // q if b == c and (d != b || e != b)
    if (b == c && (d != b || e != b))
        return 'q';
    // r if b == c and (d == b && e == b)
    if (b == c && d == b && e == b)
        return 'r';
    // b != c
    return 's';
}

/// <summary>
/// 根據 f(a1, a2, a3, a4) 的算式求得 number
/// </summary>
/// <param name="a1"></param>
/// <param name="a2"></param>
/// <param name="a3"></param>
/// <param name="a4"></param>
/// <returns></returns>
private static int F(char a1, char a2, char a3, char a4)
{
    var count = 0;

    // 5 if a1 == a2 == a3 == a4 == r
    if (a1 == 'r' && a2 == 'r' && a3 == 'r' && a4 == 'r')
        return 5;

```

```

// n where number of q
if (a1 == 'q') count += 1;
if (a2 == 'q') count += 1;
if (a3 == 'q') count += 1;
if (a4 == 'q') count += 1;

return count;
}

```

## 結果圖片

