

PixelAnalyzer

PixelAnalyzer kan opsøge døde pixels i CCD'er, ved at man loader et sort billede taget med tildækket linse og lav ISO, og justere RGB-værdierne indtil pixels lige bliver synlige, hvorefter antallet samt disses intensitet beregnes.

Programmet kan også fungere som 'online' EXIF-monitor og billedviser for billeder hvis sti og filnavn kopieres til clipboard, hvilket f.eks. kan gøres ved at højreklikke billedet i **ACDSee Manager** eller **Lister PhotoViewer**.

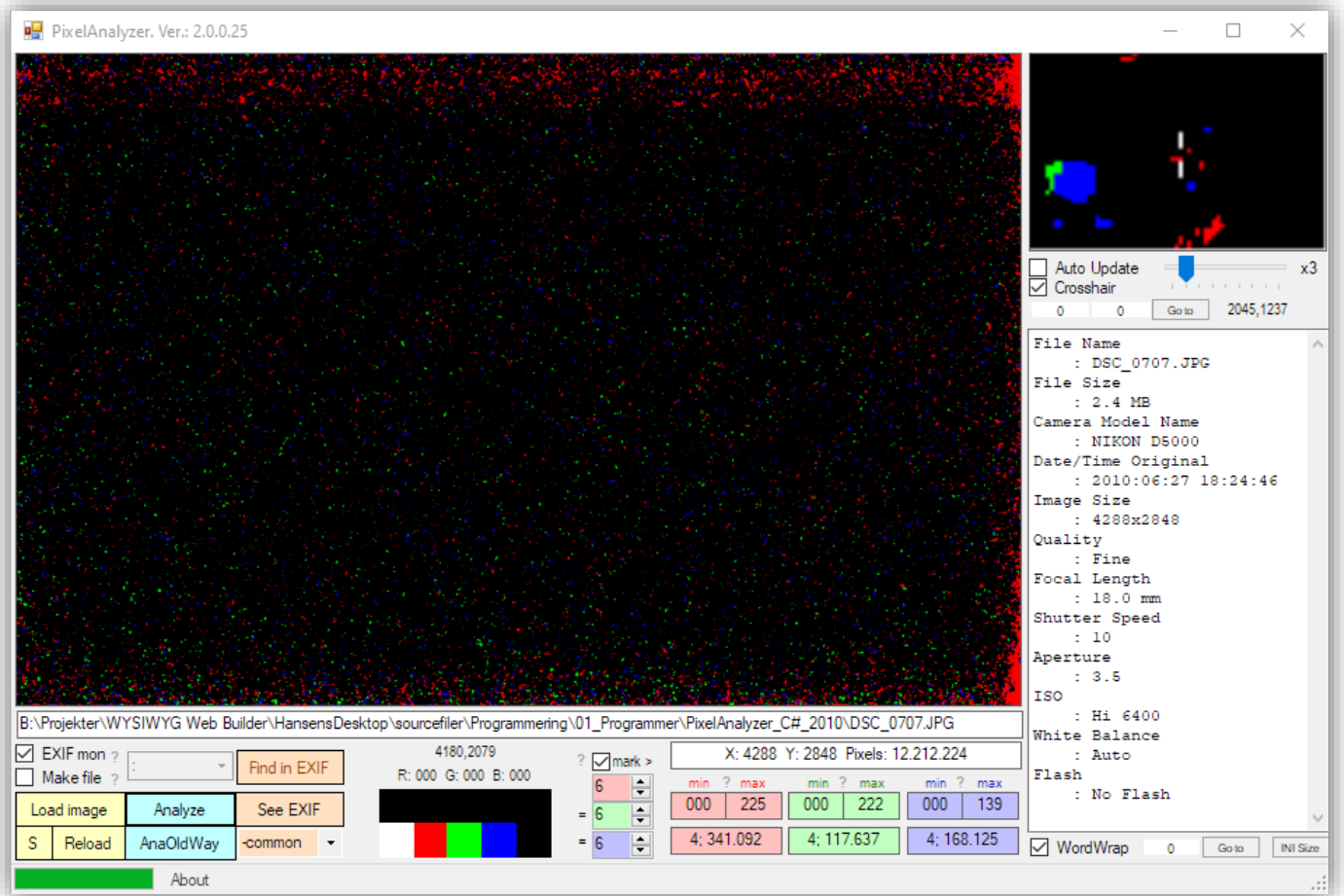


Fig.1 Her er der kørt 'Analyze' på et billede med mange 'døde pixels', hvilket dog udelukkende skyldes ISO-støj, da billedet er taget ved ISO6400 for illustrationens skyld. Øverst til højre vises området under musens markør zoomet 3x.

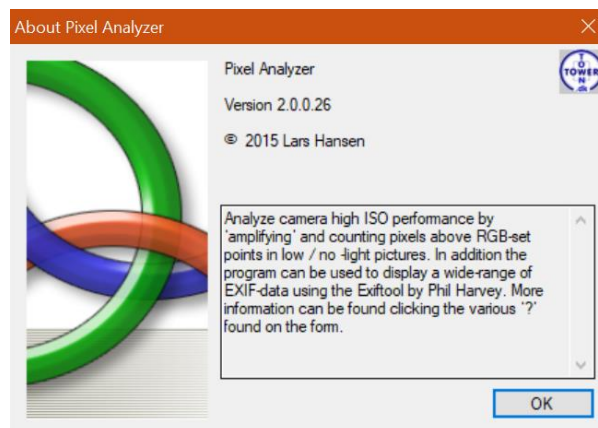


Fig.2

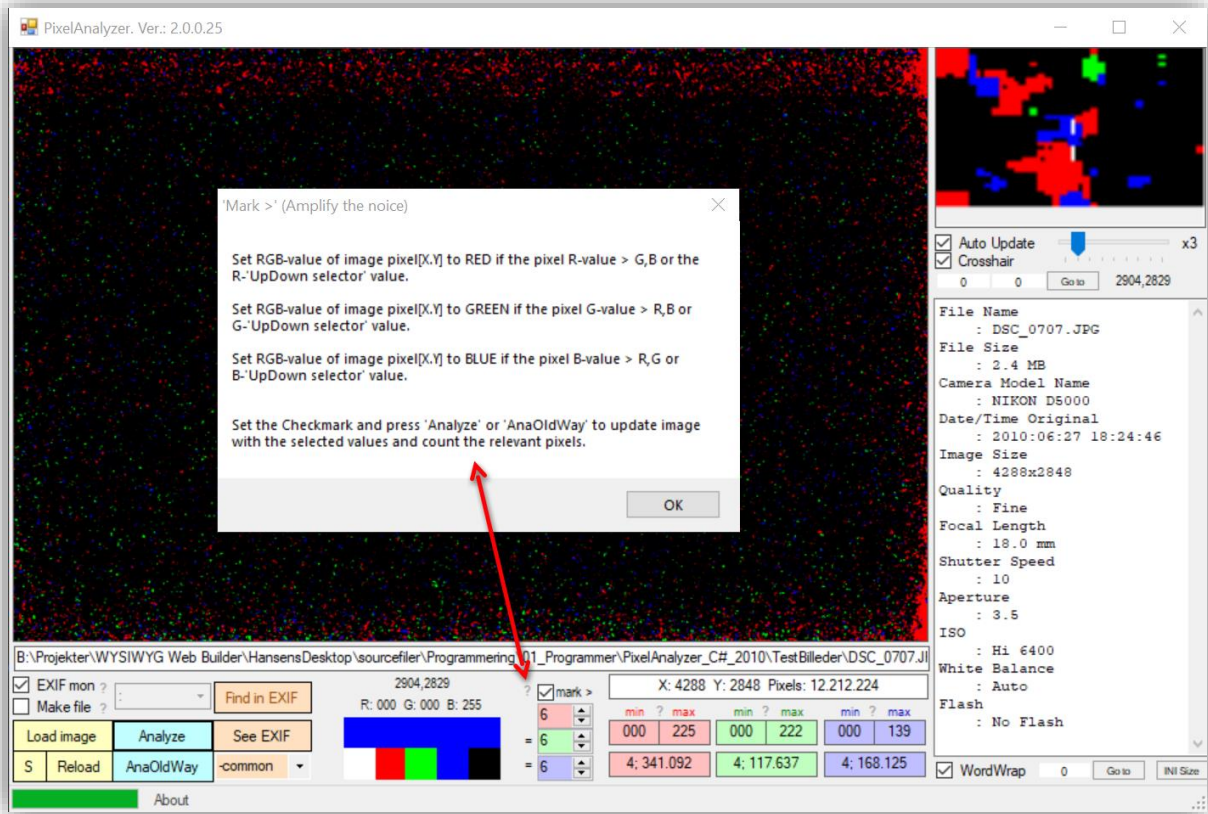


Fig.3 Et pixel i punkt X,Y farves rød, hvis dets R-værdi > værdierne for G,B i punktet, eller valgte R-værdi osv.

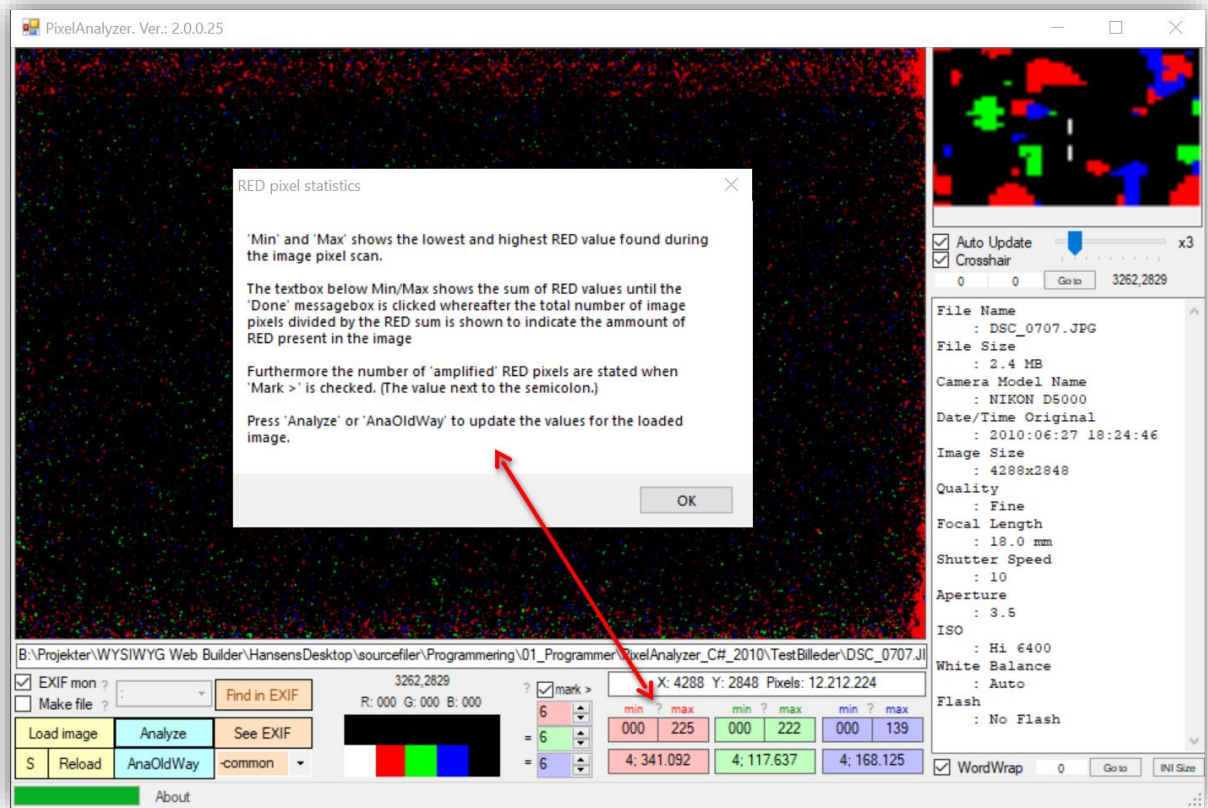


Fig.4

```

unsafe
{
    //generer en tom bitmap i samme størrelse som TestBitmap
    Bitmap newBitmap = new Bitmap(TestBitmap.Width, TestBitmap.Height);

    //lås TestBitmap i hukommelsen
    BitmapData TestBitmapData = TestBitmap.LockBits(
        new Rectangle(0, 0, TestBitmap.Width, TestBitmap.Height),
        ImageLockMode.ReadOnly, PixelFormat.Format24bppRgb);

    //lås den nye bitmap i hukommelsen
    BitmapData newData = newBitmap.LockBits(
        new Rectangle(0, 0, TestBitmap.Width, TestBitmap.Height),
        ImageLockMode.WriteOnly, PixelFormat.Format24bppRgb);

    //sæt antal bytes pr pixel
    int pixelSize = 3;

    for (int y = 0; y < TestBitmap.Height; y++)
    {
        if (GemTilFil)
        {
            ListIndhold = y.ToString("Y: 000" + "-----");
            TW.WriteLine(ListIndhold);
        }

        //hent data fra TestBitmap
        byte* oRow = (byte*)TestBitmapData.Scan0 + (y * TestBitmapData.Stride);

        //hent data fra det nye billede
        byte* nRow = (byte*)newData.Scan0 + (y * newData.Stride);

        for (int x = 0; x < TestBitmap.Width; x++)
        {
            PixelIalt++;

            //generer gråskala version
            byte grayScale =
                (byte)((oRow[x * pixelSize] * .11) + //B
                    (oRow[x * pixelSize + 1] * .59) + //G
                    (oRow[x * pixelSize + 2] * .3)); //R

            blue = Convert.ToInt16(oRow[x * pixelSize]);
            green = Convert.ToInt16(oRow[x * pixelSize + 1]);
            red = Convert.ToInt16(oRow[x * pixelSize + 2]);

            if (red < rMin) { rMin = red; }
            if (red > rMax) { rMax = red; }

            if (green < gMin) { gMin = green; }
            if (green > gMax) { gMax = green; }
        }
    }
}

```

Fig.5

Eksempel på programmets C#-kode.