

در [قسمت دوم](#) با نحوه‌ی بارگذاری تصاویر در OpenCVSharp آشنا شدیم. در این قسمت قصد داریم با نحوه‌ی ایجاد یک clone و نمونه‌ای مشابه از تصویر اصلی بارگذاری شده آشنا شویم. برای مثال هرچند متد LoadImage، دارای پارامتر بارگذاری تصویر، به صورت سیاه و سفید است، اما توصیه نمی‌شود که در بدو امر، تصویر را سیاه و سفید بارگذاری کنید. چون هرگونه تغییری در تصویر اصلی، امکان استفاده‌ی از آن را در سایر متدها و الگوریتم‌ها با مشکل مواجه می‌کند و استفاده‌ی از حالت LoadMode.GrayScale جهت بالا بردن سرعت عملیات، در کارهای پردازش تصویر بسیار معمول است.

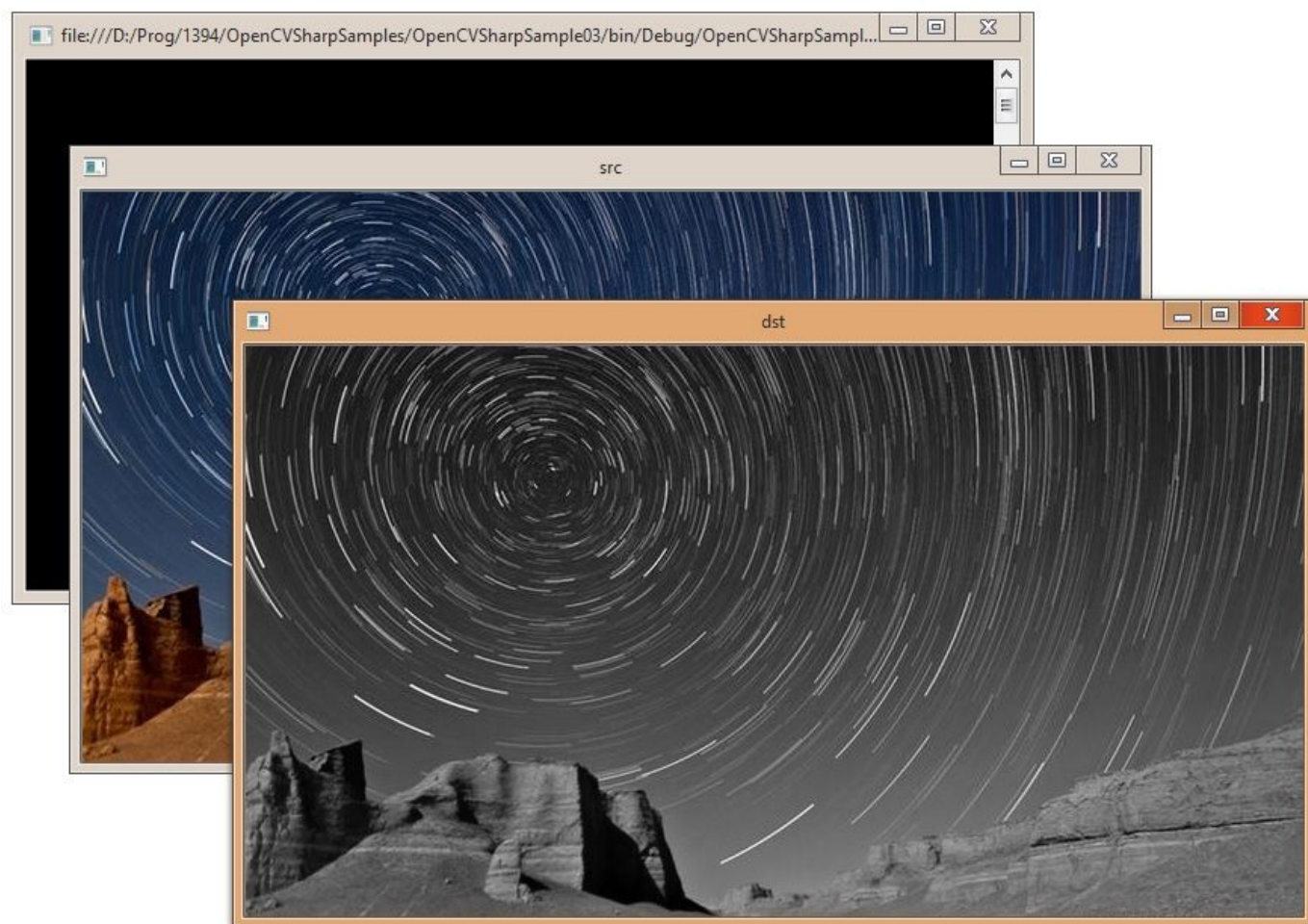
تهیه‌ی یک نمونه‌ی سیاه و سفید از تصویر اصلی در OpenCVSharp

برای تهیه‌ی یک نمونه‌ی مشابه تصویر اصلی، از متد CreateImage استفاده می‌شود:

```
using (var src = Cv.LoadImage(@"..\..\images\ocv02.jpg", LoadMode.Color))
using (var dst = Cv.CreateImage(new CvSize(src.Width, src.Height), BitDepth.U8, 1))
{
    Cv.CvtColor(src, dst, ColorConversion.BgrToGray);

    using (new CvWindow("src", image: src))
    using (new CvWindow("dst", image: dst))
    {
        Cv.WaitKey();
    }
}
```

با این خروجی



معرفی متد CreateImage

پارامتر اول متد [CreateImage](#)، اندازه‌ی تصویر تولیدی را مشخص می‌کند. پارامتر دوم آن تعداد بیت تصویر را تعیین خواهد کرد. این تعداد بیت عموماً بر اساس نیاز متدهای مختلف پردازش تصویر، متغیر خواهند بود و برای تعیین آن نیاز است مستندات هر متد را مطالعه کرد. BitDepth.U8 به معنای 8bit unsigned است.

پارامتر سوم این متد، تعیین کننده‌ی تعداد کانال تصویر است. تصاویر رنگی دارای سه کانال سبز، قرمز و آبی، هستند. چون در اینجا قصد داریم تصویر را سیاه و سفید کنیم، تعداد کانال را به عدد یک تنظیم کرده‌ایم.

متد CreateImage جهت سازگاری با اینترفیس C مربوط به OpenCV در اینجا وجود دارد. معادل

```
using (var dst = Cv.CreateImage(new CvSize(src.Width, src.Height), BitDepth.U8, 1))
```

را می‌توان به نحو ذیل نیز نوشت:

```
var dst = new IplImage(new CvSize(src.Width, src.Height), BitDepth.U8, 1)
```

و یا حتی پارامتر تعیین اندازه‌ی تصویر را نیز می‌توان ساده‌تر کرد:

```
using (var dst = new IplImage(src.Size, BitDepth.U8, 1))
```

تبدیل تصویر به حالت سیاه و سفید

متد [CvtColor](#) جهت تغییر color space بکار می‌رود که در اینجا BGR (Blue/Green/Red) را به Gray تبدیل کرده‌است:

```
Cv.CvtColor(src, dst, ColorConversion.BgrToGray);
```

این متد را در OpenCVSharp به نحو ذیل نیز می‌توان بازنویسی کرد:

```
src.CvtColor(dst, ColorConversion.BgrToGray);
```

بنابراین به صورت خلاصه می‌توان کدهای ابتدای بحث را به صورت زیر نیز نوشت که با کلاس‌های OpenCVSharp بیشتر سازگاری دارد:

```
using (var src = new IplImage(@"..\..\images\ocv02.jpg", LoadMode.Color))
//using (var dst = new IplImage(new CvSize(src.Width, src.Height), BitDepth.U8, 1))
using (var dst = new IplImage(src.Size, BitDepth.U8, 1))
{
    src.CvtColor(dst, ColorConversion.BgrToGray);

    using (new CvWindow("src", image: src))
    using (new CvWindow("dst", image: dst))
    {
        Cv.WaitKey();
    }
}
```

کدهای کامل این مثال را [از اینجا](#) می‌توانید دریافت کنید.