

بررسی morphology (ریخت شناسی) تصاویر

به تصویر زیر دقت کنید:



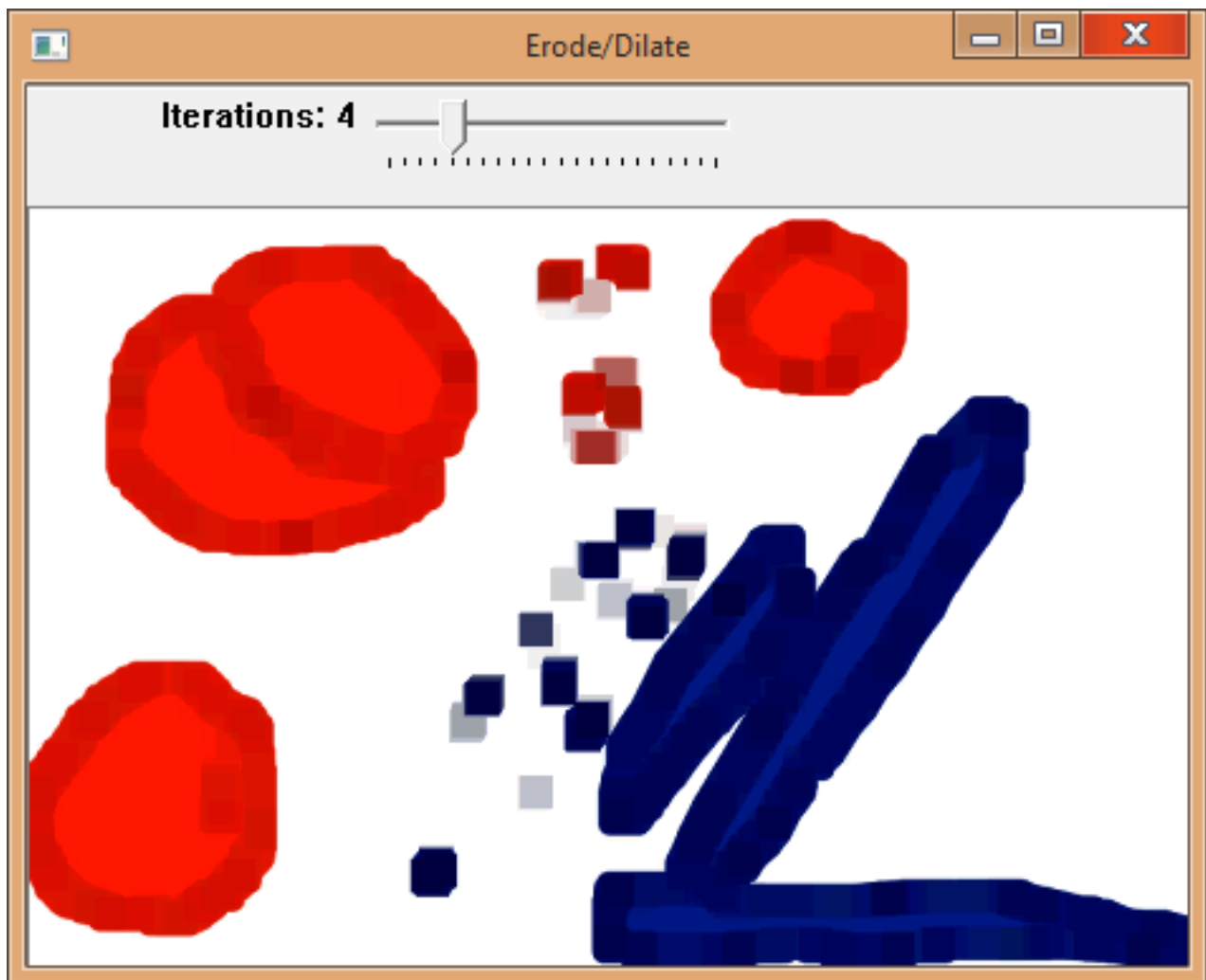
فرض کنید در اینجا قصد دارید تعداد توپ‌های قرمز را شمارش کنید. از دیدگاه یک انسان، شاید سه توپ قرمز قابل مشاهده باشد. اما از دیدگاه یک برنامه، توپ وسطی به دو توپ تفسیر خواهد شد و همچنین نویزهای قرمزی که بین توپ‌ها در صفحه وجود دارند نیز شمارش می‌شوند. بنابراین بهتر است پیش از پردازش این تصویر، ریخت شناسی آن را بهبود بخشید. برای مثال توپ وسطی را یکی کرد، حفره‌های توپ‌های دیگر را پوشاند و یا نویزهای قرمز را حذف نمود. به علاوه خطوط آبی رنگی را که با یکدیگر تماس یافته‌اند نیز می‌خواهیم اندکی از هم جدا کنیم.

متدهایی که مورفولوژی تصاویر را تغییر می‌دهند

در OpenCV سه متد یا [فیلتر](#) مهم، کار تغییر مورفولوژی تصاویر را انجام می‌دهند:

(1) [Cv2.Erode](#)

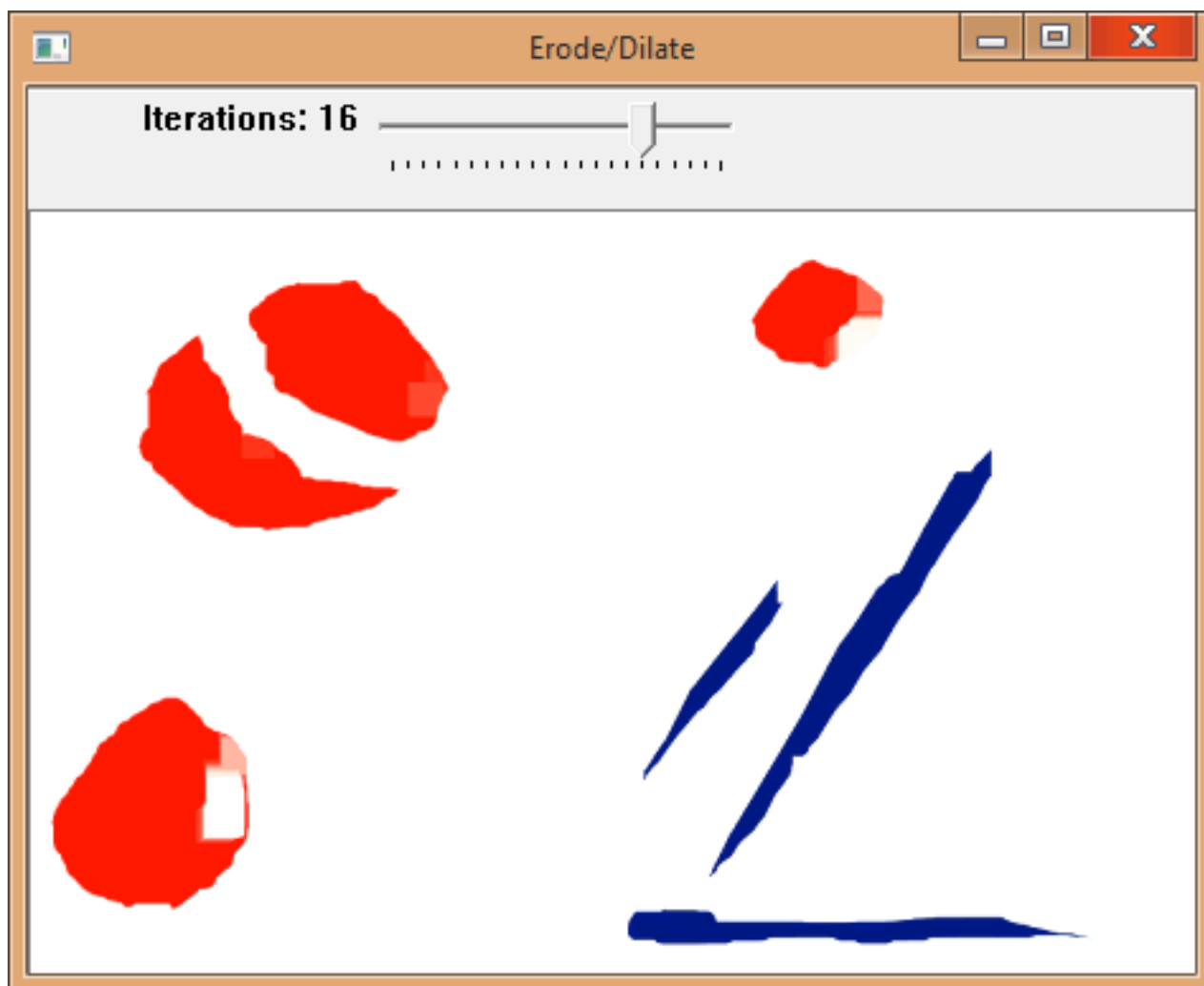
تحلیل/فرسایش یا erosion سبب می‌شود تا نواحی تیره‌ی تصویر «رشد» کنند.



در اینجا فیلتر Erode کار یکی کردن اجزای جدای توپ‌های رنگی را انجام داده‌است.

[Cv2.Dilate](#) (2)

اتساع یا dilation سبب خواهد شد تا نواحی روشن تصویر «رشد» کنند.

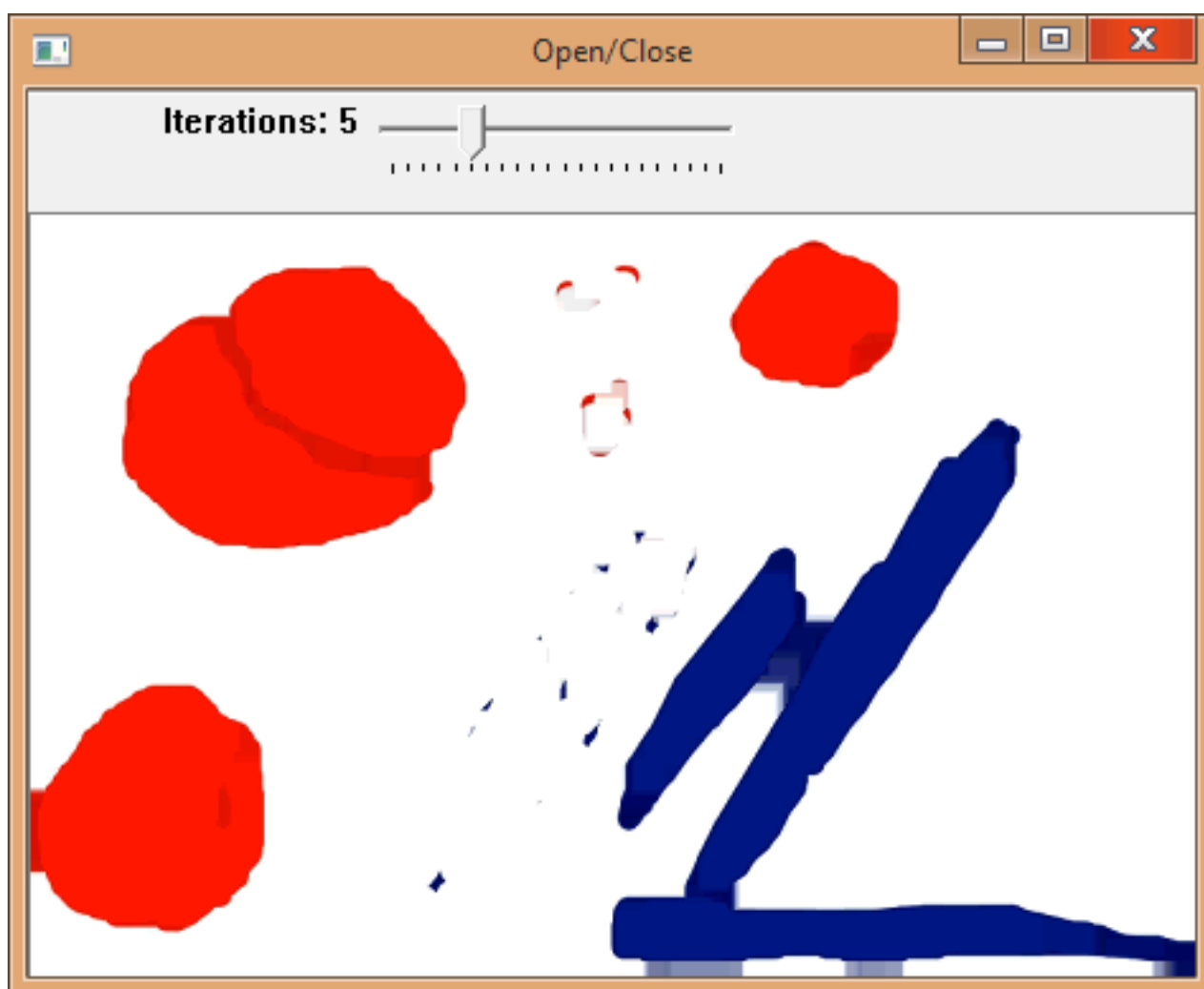


بکارگیری فیلتر Dilate سبب شده است تا نویزهای تصویر محو شوند و اشیاء به هم پیوسته از هم جدا گردند.

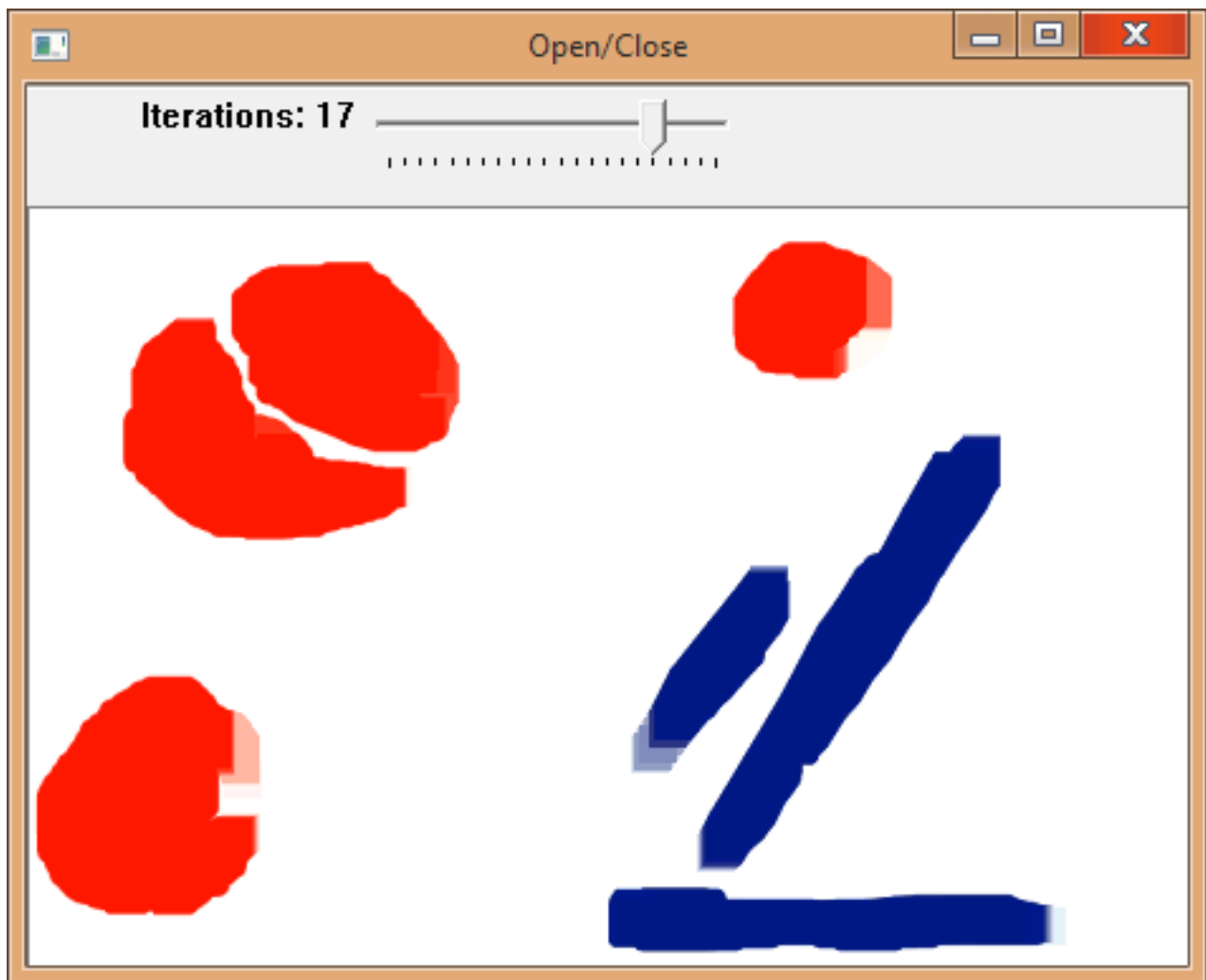
[Cv2.MorphologyEx](#) (3)

کار این متد انجام اعمال پیشرفته‌ی مورفولوژی بر روی تصاویر است و در اینجا ترکیبی از erosion و dilation، با هم انجام می‌شوند.

اگر پارامتر سوم آن به MorphologyOperation.Open تنظیم شود، ابتدا erosion و سپس dilation انجام خواهد شد:



و اگر این پارامتر به `MorphologyOperation.Close` مقدار دهی شود، ابتدا `dilation` و سپس `erosion` انجام می‌شود:



در تمام این حالات، پارامتر آخر که Structuring Element نام دارد، یکی از مقادیر اشیاء مستطیل، به علاوه و بیضی را می‌تواند داشته باشد. این اشیاء و اندازه‌ی آن‌ها، مشخص‌کننده‌ی میزان تحلیل و یا اتساع نهایی هستند.

Structuring element shapes

Rect



Cross



Ellipse



استفاده از متدهای مورفولوژی در عمل

در اینجا مثالی را از نحوه‌ی بکارگیری متدهای اتساع و فرسایش، ملاحظه می‌کنید:

```
using (var src = new Mat(@"..\..\Images\cvmorph.Png", LoadMode.AnyDepth | LoadMode.AnyColor))
using (var dst = new Mat())
{
    src.CopyTo(dst);

    var elementShape = StructuringElementShape.Rect;
    var maxIterations = 10;

    var erodeDilateWindow = new Window("Erode/Dilate", image: dst);
    var erodeDilateTrackbar = erodeDilateWindow.CreateTrackbar(
        name: "Iterations", value: 0, max: maxIterations * 2 + 1,
        callback: pos =>
        {
            var n = pos - maxIterations;
            var an = n > 0 ? n : -n;
            var element = Cv2.GetStructuringElement(
                elementShape,
                new Size(an * 2 + 1, an * 2 + 1),
                new Point(an, an));
            if (n < 0)
            {
                Cv2.Erode(src, dst, element);
            }
            else
            {
                Cv2.Dilate(src, dst, element);
            }

            Cv2.PutText(dst, (n < 0) ?
                string.Format("Erode[{0}]", elementShape) :
                string.Format("Dilate[{0}]", elementShape),
```

```

        new Point(10, 15), FontFace.HersheyPlain, 1, Scalar.Black);
        erodeDilateWindow.Image = dst;
    });

    Cv2.WaitKey();
    erodeDilateWindow.Dispose();
}

```

اینترفیس به کار گرفته شده، [همان API C++](#) است و در اینجا ابتدا یک تصویر و کپی آن تهیه می‌شوند. سپس پنجره‌ی سازگار با C++ API ایجاد شده و به این پنجره یک شیء tracker اضافه می‌شود. این tracker یا slider، چندسکوپی است. بدیهی است اگر قرار بود چنین کاری را صرفاً با یک برنامه‌ی دات نتی انجام داد، می‌شد قسمت ایجاد پنجره و tracker آن‌را حذف کرد و بجای آن‌ها از یک [picture box](#) و یک slider به همراه مدیریت روال رخدادگردان تغییر مقادیر slider کمک گرفت. اما در اینجا تنها جهت آشنایی با این امکانات توکار OpenCV، از همان متدهای native آن استفاده شده‌است. در این مثال، روال رخدادگردان تغییر مقادیر tracker یا slider، یک function pointer است که معادل آن در دات نت یک delegate می‌باشد که در پارامتر callback متد ایجاد tracker قابل مشاهده است. هر بار که مقدار tracker تغییر می‌کند، مقدار pos را به callback خود ارسال خواهد کرد. از این مقدار جهت ایجاد شیء ساختاری و همچنین انتخاب بین حالات اتساع و فرسایش، کمک گرفته شده‌است. کار متد PutText، نوشتن یک متن ساده بر روی پنجره‌ی native مربوط به OpenCV است.

همچنین در ادامه کدهای بکارگیری متد MorphologyEx را که کار ترکیب اتساع و فرسایش را با هم انجام می‌دهد، ذکر شده‌است و نکات بکارگیری آن همانند مثال اول بحث است:

```

using (var src = new Mat(@"..\..\Images\cvmorph.Png", LoadMode.AnyDepth | LoadMode.AnyColor))
    using (var dst = new Mat())
    {
        src.CopyTo(dst);

        var elementShape = StructuringElementShape.Rect;
        var maxIterations = 10;

        var openCloseWindow = new Window("Open/Close", image: dst);
        var openCloseTrackbar = openCloseWindow.CreateTrackbar(
            name: "Iterations", value: 0, max: maxIterations * 2 + 1,
            callback: pos =>
            {
                var n = pos - maxIterations;
                var an = n > 0 ? n : -n;
                var element = Cv2.GetStructuringElement(
                    elementShape,
                    new Size(an * 2 + 1, an * 2 + 1),
                    new Point(an, an));

                if (n < 0)
                {
                    Cv2.MorphologyEx(src, dst, MorphologyOperation.Open, element);
                }
                else
                {
                    Cv2.MorphologyEx(src, dst, MorphologyOperation.Close, element);
                }

                Cv2.PutText(dst, (n < 0) ?
                    string.Format("Open/Erosion followed by Dilation[{0}]", elementShape)
                    : string.Format("Close/Dilation followed by Erosion[{0}]", elementShape),
                    new Point(10, 15), FontFace.HersheyPlain, 1, Scalar.Black);
                openCloseWindow.Image = dst;
            });

        Cv2.WaitKey();
        openCloseWindow.Dispose();
    }

```

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