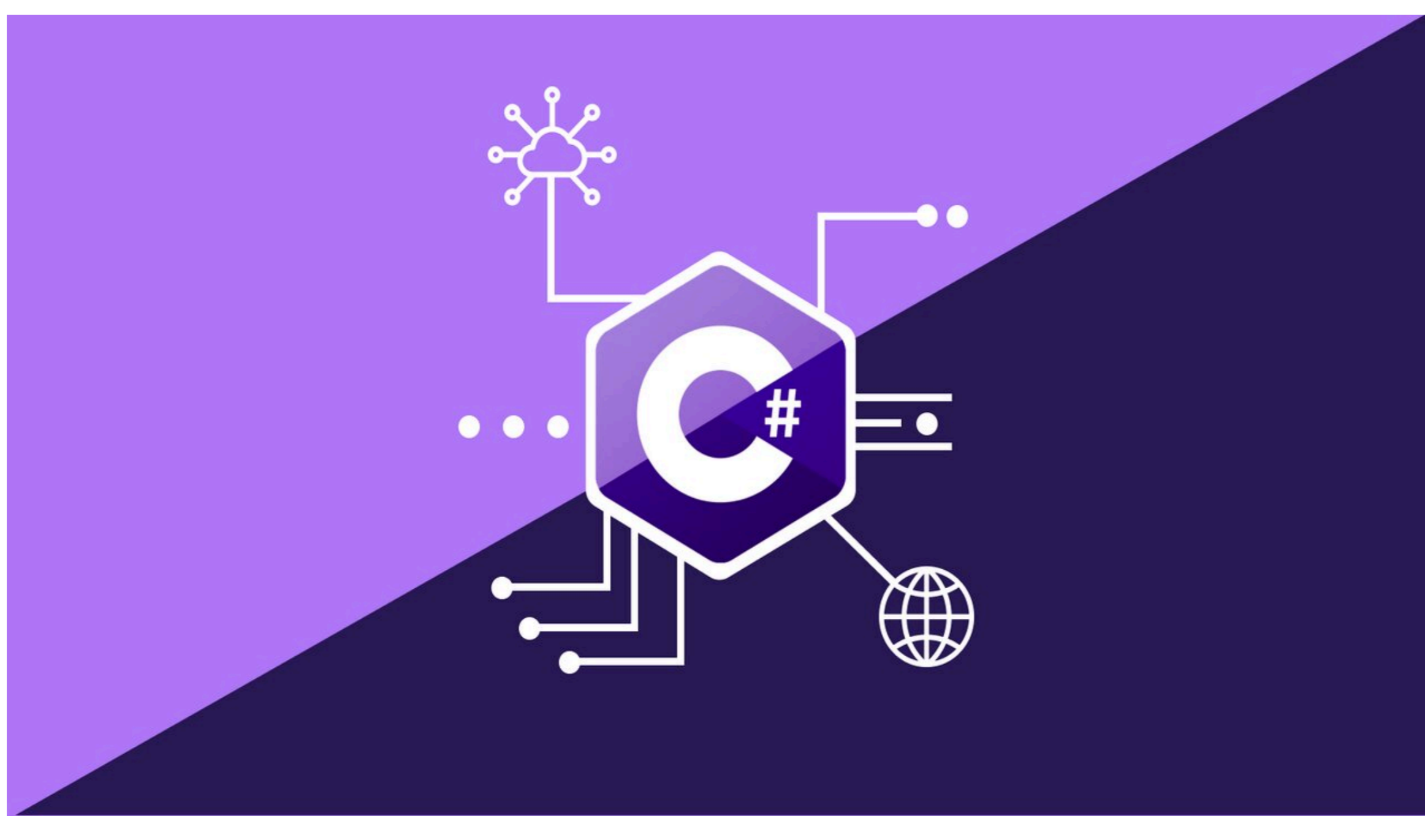


C#中轻松实现二维码和条形码识别：OpenCvSharp和ZXing详细教程

作者：微信公众号：【架构师老卢】

12-4 8:24

59



概述：本教程使用OpenCvSharp和ZXing库，详细介绍了在C#中识别二维码和条形码的步骤。通过导入必要的命名空间、加载图像，并使用ZXing库进行二维码和条形码的识别，提供了清晰的示例代码。这方便了开发人员将集成二维码和条形码识别功能。

要使用OpenCvSharp来分别识别二维码和条形码，首先需要确保已经安装了OpenCvSharp库。你可以通过以下步骤在C#中使用OpenCvSharp来实现这两种识别。

步骤 1：安装 OpenCvSharp

在 Visual Studio 中，通过 NuGet 包管理器安装 OpenCvSharp。右击项目 -> NuGet 包管理器 -> 管理 NuGet 包，搜索 OpenCvSharp 并安装。

步骤 2：导入必要的命名空间

在代码文件的开头，导入必要的命名空间。

```
1 using OpenCvSharp;
2 using OpenCvSharp.Extensions;
3 using ZXing;
```

步骤 3：加载图像

加载包含二维码或条形码的图像。

```
1 Mat image = new Mat("path_to_your_image.jpg");
```

步骤 4：二维码识别

使用 ZXing 库来识别二维码。

```
1     /// <summary>
2     /// 只识别二维码
3     /// </summary>
4     static void DecodeQRcodes()
5     {
6         var img = Cv2.ImRead("2.jpg");
7         string[] qrCodeTexts = null;
8         // 二维码识别
9         using (QRCodeDetector qrCodeDetector = new QRCodeDetector())
10        {
11            Point2f[] points;
12            var hasQRCode = qrCodeDetector.DetectMulti(img, out points);
13            if (hasQRCode)
14            {
15                qrCodeDetector.DecodeMulti(img, points, out qrCodeTexts);
16            }
17        }
18
19        if (qrCodeTexts != null)
20        {
21            Console.WriteLine($"检测到{qrCodeTexts.Length}个二维码:");
22            for (int i = 0; i < qrCodeTexts.Length; i++)
23            {
24                Console.WriteLine($"第{i + 1}个的内容为: {qrCodeTexts[i]}");
25            }
26        }
27    }
```

步骤 5：条形码识别

使用 ZXing 库来识别条形码。

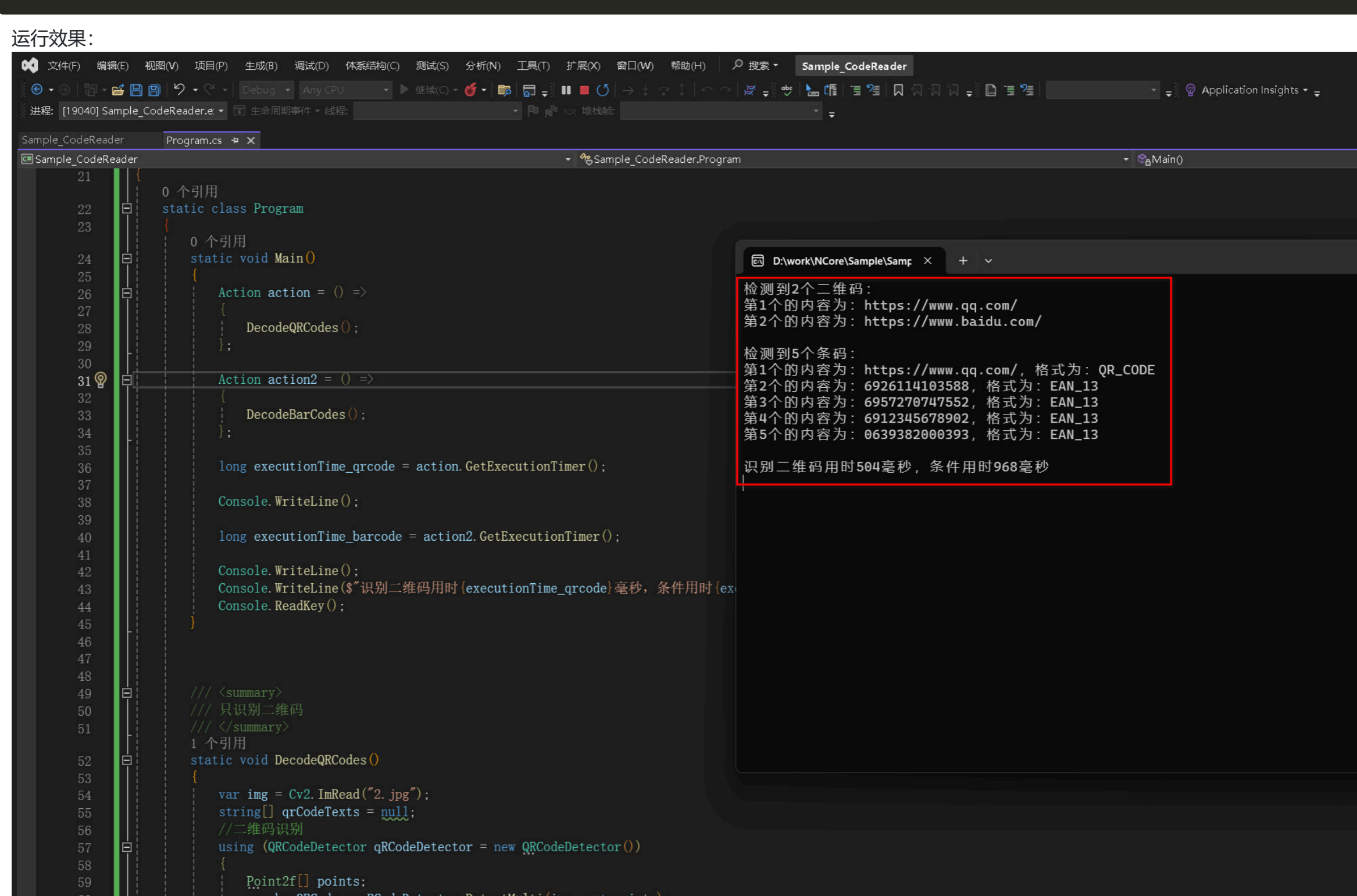
```
1     /// <summary>
2     /// 识别条码（也可以识别二维码）
3     /// </summary>
4     static void DecodeBarCodes()
5     {
6         // 读取图像
7         var img = Cv2.ImRead("1.jpg");
8
9         BarcodeReader barcodeReader = new BarcodeReader
10        {
11            Options = new DecodingOptions
12            {
13                TryHarder = true,
14                PossibleFormats = new[]
15                {
16                    BarcodeFormat.CODE_128,
17                    BarcodeFormat.QR_CODE, // 这个是二维码
18                    BarcodeFormat.EAN_13,
19                    BarcodeFormat.EAN_8,
20                    BarcodeFormat.CODE_39,
21                    BarcodeFormat.CODE_93
22                }
23            },
24            AutoRotate = true,
25        };
26
27        // 进行条形码解码
28        var barcodes = barcodeReader.DecodeMultiple(img);
29
30        if (barcodes != null)
31        {
32            Console.WriteLine($"检测到{barcodes.Length}个条码:");
33            for (int i = 0; i < barcodes.Length; i++)
34            {
35                Console.WriteLine($"第{i + 1}个的内容为: {barcodes[i].Text}, 格式为: {barcodes[i].BarcodeFormat}");
36            }
37        }
38    }
```

示例代码

下面是完整的示例代码，其中包括了加载图像、二维码识别和条形码识别的步骤：

```
1 using System;
2 using OpenCvSharp;
3 using OpenCvSharp.Extensions;
4 using ZXing;
5
6 class Program
7 {
8     static void Main()
9     {
10        Action action = () =>
11        {
12            DecodeQRcodes();
13        };
14
15        Action action2 = () =>
16        {
17            DecodeBarCodes();
18        };
19
20        long executionTime_qrcode = action.GetExecutionTimer();
21
22        Console.WriteLine();
23
24        long executionTime_barcode = action2.GetExecutionTimer();
25
26        Console.WriteLine();
27        Console.WriteLine($"识别二维码用时{executionTime_qrcode}毫秒, 条件用时{executionTime_barcode}毫秒");
28        Console.ReadKey();
29    }
30 }
```

运行效果：







请注意，示例中的路径 "path_to_your_image.jpg" 应替换为你实际图像文件的路径。此外，确保图像中包含有效的二维码或条形码。

相关代码下载地址



重要提示！：取消关注公众号后将无法再启用回复功能，不支持解封！

第一步：微信扫码关注公众号“架构师老卢”

第二步：在公众号聊天框发送 **code: 27064**，如：  **code: 27064**  获取下载地址

第三步：恭喜你，快去下载你想要的资源吧