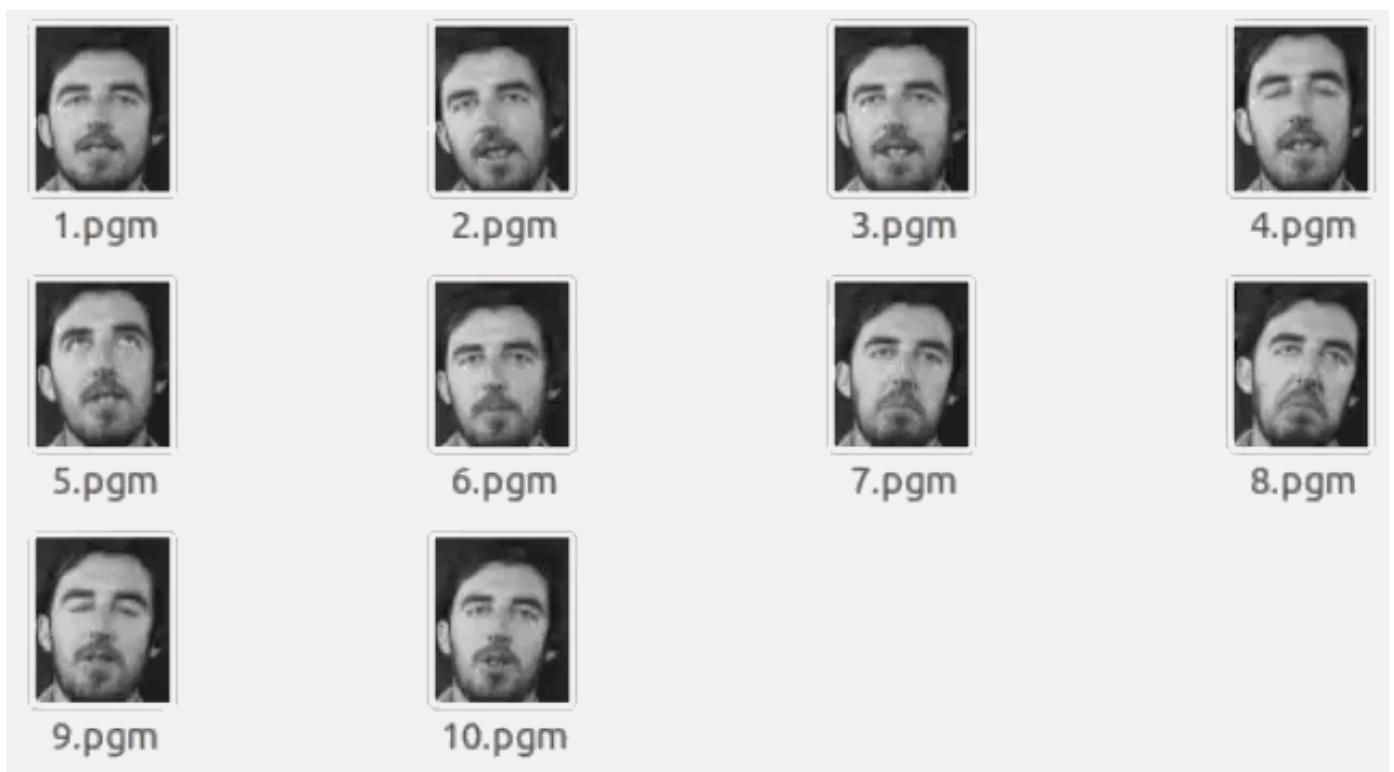


تشخیص اشخاص به کمک OpenCV

فرض کنید قصد دارید یک سیستم حضور غیاب مبتنی بر تشخیص چهره را طراحی کنید. قسمت استخراج چهره، از تصویر کلی رسیده [را بررسی کردیم](#) . اما در ادامه چگونه تشخیص دهیم که این چهره متعلق به چه شخصی است؟ با توجه به اینکه تصویر چهره‌ی یک شخص می‌تواند از زوایای مختلفی تهیه شود و یا حتی حالات روحی منعکس شده‌ی در صورت نیز در تغییر بیت و بایت‌های تصویر چهره مؤثر هستند.

بانک اطلاعاتی تصاویر چهره‌های اشخاص

در اینجا از تصاویر « [The Database of Faces](#) » استفاده خواهیم کرد. این مجموعه شامل تصاویر 40 شخص، در 10 حالت مختلف است.



برای بارگذاری این تصاویر و استفاده‌ی از آن‌ها در الگوریتم FisherFaceRecognizer نیاز به ساختار ذیل است:

```
public class ImageInfo
{
    public Mat Image { set; get; }
    public int ImageGroupId { set; get; }
    public int ImageId { set; get; }
}
```

در اینجا Image، محتوای تصویر انتخابی است. مقدار ImageGroupId مساوی مقدار عددی نام پوشه‌ی تصاویر منهای یک، تنظیم

می‌شود. برای مثال پوشه‌ی s1 به گروه صفر تنظیم می‌شود. ImageId نیز به یک مقدار خود افزایش یابنده معادل شمارهی جاری تصویر، تنظیم می‌گردد؛ به این صورت:

```
var images = new List<ImageInfo>();
var imageId = 0;
foreach (var dir in new DirectoryInfo(@"..\..\Images").GetDirectories())
{
    var groupId = int.Parse(dir.Name.Replace("s", string.Empty)) - 1;
    foreach (var imageFile in dir.GetFiles("*.pgm"))
    {
        images.Add(new ImageInfo
        {
            Image = new Mat(imageFile.FullName, LoadMode.GrayScale),
            ImageId = imageId++,
            ImageGroupId = groupId
        });
    }
}
```

ابتدا پوشه‌های دیتابیس تصاویر یافت شده و سپس از نام هر پوشه یک شمارهی گروه (یا شمارهی شخص) استخراج می‌شود. سپس تصاویر این پوشه به لیست تصاویر اصلی اضافه خواهند شد.

تشخیص یک چهره‌ی اتفاقی

پس از تشکیل لیست تصاویر، اکنون کار با الگوریتم FisherFaceRecognizer به نحو ذیل خواهد بود:

```
var model = FaceRecognizer.CreateFisherFaceRecognizer();
model.Train(images.Select(x => x.Image), images.Select(x => x.ImageGroupId));

var rnd = new Random();
var randomImageId = rnd.Next(0, images.Count - 1);
var testSample = images[randomImageId];

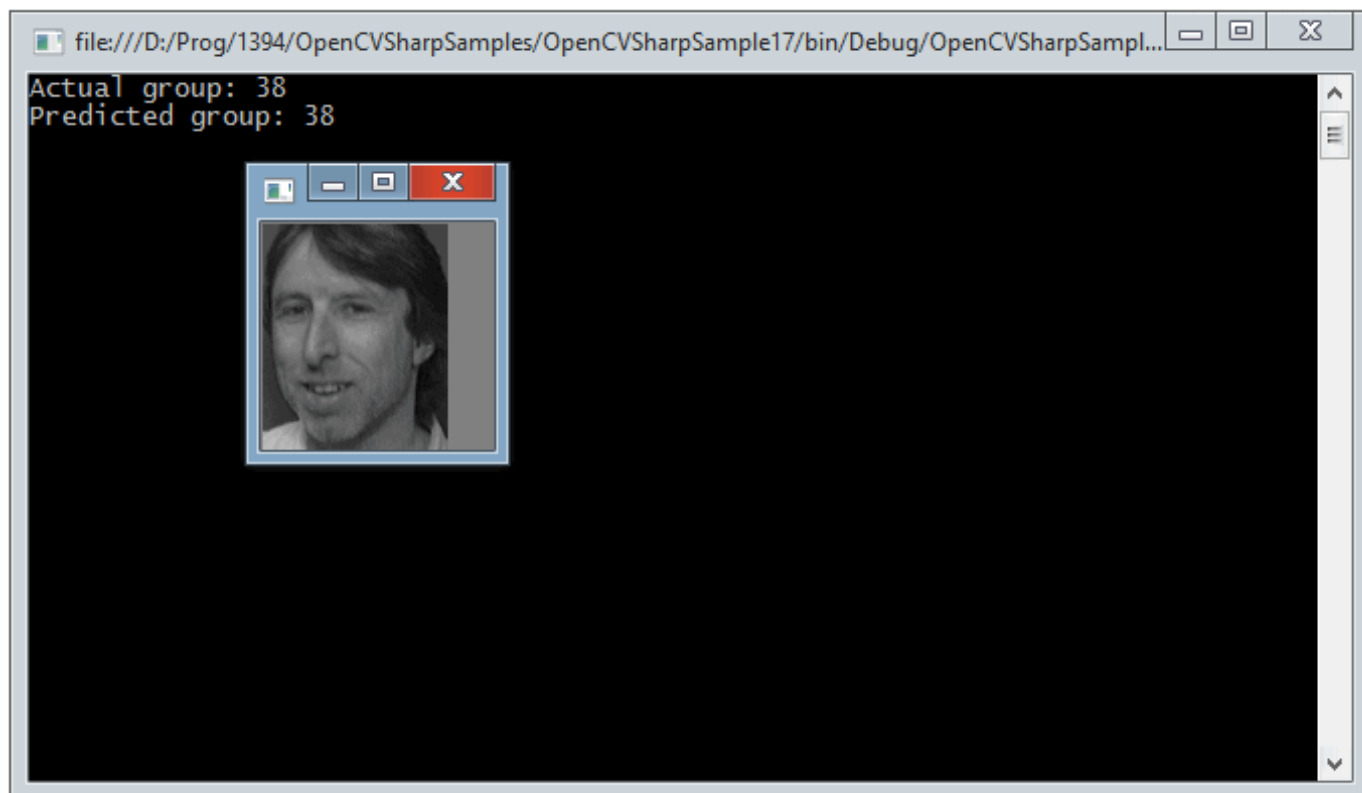
Console.WriteLine("Actual group: {0}", testSample.ImageGroupId);
Cv2.ImShow("actual", testSample.Image);

var predictedGroupId = model.Predict(testSample.Image);
Console.WriteLine("Predicted group: {0}", predictedGroupId);
```

پارامتر اول متد Train، لیست تصاویر است و پارامتر دوم، لیست شماره گروه‌های متناظر با هر تصویر است که در اینجا به عنوان برچسب نیز نامگذاری شده‌است.

سپس با استفاده از کلاس Random، یک تصویر اتفاقی انتخاب می‌شود.

اکنون این تصویر اتفاقی به متد Predict ارسال شده و نتیجه‌ی آن، شماره گروه چهره‌ی تشخیص داده شده‌است. به این ترتیب می‌توان تشخیص داد که یک تصویر مفروض ورودی، متعلق به چه شخصی (یا در اینجا گروه یا برچسب) است.



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

نظرات خوانندگان

نویسنده: م منفرد
تاریخ: ۱۸:۴۴ ۱۳۹۴/۰۴/۰۱

با این اوصاف، اگر عکسی از حالت مثلاً خنده فرد در بانک ثبت شده است و ما حالت عصبانیت ایشان را داشته باشیم، قابل تشخیص است؟

نویسنده: صادق ایمانی
تاریخ: ۲۰:۱۴ ۱۳۹۴/۰۴/۰۱

خیلی ممنون؛ من خودم چند ماه قبل داشتم به سیستم تشخیص و شناسایی چهره با کمک این [سایت](#) مینوشتم... که خوب متأسفانه نتیجه خوبی نگرفتم به این خاطر که بعضی چهره‌ها رو به اشتباه تشخیص می‌داد... البته تعداد گروه‌های عکس‌هایی که من Train کردم زیاد بود شاید به این خاطر اشتباه تشخیص می‌داد... فک کنم در حدود 50 گروه بود که در هر گروه 10 عکس برای Train وجود داشت. به نظر شما بهترین Library برای تشخیص و شناسایی چهره چیه؟ و اینکه این درست است که اگر تعداد گروه‌های Train زیاد باشه احتمال تشخیص صحیح کم میشه یا اینکه من به جای کار اشتباه کردم؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۲۲ ۱۳۹۴/۰۴/۰۱

یک عکس کافی نیست. این‌ها الگوریتم‌های machine learning هستند و متد Train ایی که اینجا بکار گرفته شده، دقیقاً به همین منظور است. هر چقدر تعداد نمونه‌های بیشتری از یک شخص داشته باشید (مثل تصویر ابتدای بحث یا [مطلب تهیه trained data](#)), دقت کار بیشتر می‌شود.

نویسنده: وحید نصیری
تاریخ: ۲۰:۲۵ ۱۳۹۴/۰۴/۰۱

در [مطلب تهیه trained data](#) اشاره‌ای به این موضوع شده‌است. اگر تعداد stage معرفی شده بیشتر شود، دقت بیشتر خواهد شد و به همین ترتیب، تهیه‌ی داده‌ی آموزش داده شده کندتر می‌شود. همچنین در اینجا الگوریتم‌های زیادی هم برای آموزش دادن داده‌ها وجود دارند؛ نمونه‌ای که بکار گرفته شد LBP یا Local binary Pattern بود که بسیار سریع هست. روش‌های دیگر دقت بالاتری دارند اما کند هستند. در کل باید روی الگوریتم‌ها و تعداد stages بیشتر بررسی کنید. به علاوه الگوریتم‌های FaceRecognizer دیگری هم وجود دارند که نیاز به بررسی بیشتری دارند. در مطلب جاری فقط الگوریتم فیشر این سری بررسی شد.