

وزارت علوم تحقیقات و فناوری

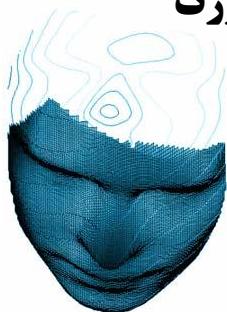


موسسه آموزش عالی علم و فن – ازومیه
(غیر دولتی – غیر انتفاعی)

گزارش کار پروژه کارشناسی ناپیوسته / گرایش نرم افزار

موضوع:

پردازش تصویر - تشخیص صورت



از طریق وب کم

(با استفاده از الگوریتم لبه ای Edge)

استاد راهنمای:

آقای مهندس قاسم زاده

دانشجو:

یونس احمدپور

شماره دانشجویی :

۸۸۱۱۱۲۵۰۰۸



تقدیر و تشکر

سپاس خدای را که سخنوران، در ستودن او بمانند و شمارندگان، شمردن نعمت های او ندانند و کوشندگان، حق او را گزاردن نتوانند.

شکر شایان نثار ایزد منان که توفیق را رفیق راهم ساخت تا این پایان نامه را به پایان برسانم . از استاد فاضل و اندیشمند جناب آقای مهندس قاسم زاده به عنوان استاد راهنما که همواره نگارنده را مورد لطف و محبت خود قرار داده اند، کمال تشکر را دارم و امیدوارم انسالله به لطف خدا و شایستگی هایتان جنابعالی را در بهترین دانشگاههای ایران و جهان مشاهده کنیم. از خداوند متعال توفیقات روز افزون و سلامتی برای خودتان و خانواده محترم خواستارم.

همیشه از اینکه شاگرد شما بوده ام افتخار میکنم.

این فضل تواریخ من یاد کنم
 بی پند عمر خویش برباد کنم
 یک قطره زبر عشق مولانا را
 تقدیم به آستان استاد کنم

مقدمه

شناسایی چهره

شناسایی چهره یکی از سیستم های گروه شناسایی بیولوژیک (بیومتیریک) می باشد. در این سیستم هدف شناسایی افراد در شرائط مختلف پس از دریافت یکسری تصویر چهره از افراد می باشد. تصاویری که از ابتدا برای شناسایی افراد به سیستم آموزش داده می شود را پایگاه چهره آموزش سیستم (Training Set) نامیده و تصاویری که برای تست قدرت سیستم در مقابل تصاویر جدید بوده تشکیل گروه Probe را می دهدند. شناسایی چهره در بخش اول یکی از مباحث Object detection و Object recognition در پردازش تصویر می باشد. ولی به علت پیچیدگی های ذاتی چهره و تغییرات فراوان ویژگی های آن الگوریتم های این دو بخش را بصورت خاصی برای خود بازنویسی کرده است. الگوریتم های متفاوتی برای شناسایی چهره در ۲۰ سال اخیر معرفی گشته که در زیر برخی از پرکاربردترین الگوریتم ها به اختصار توضیح داده شده است.

الگوریتم های تشخیص چهره

۱. Eigenface

از تبدیل Karhunen-Loeve's جهت تولید یکسری بودار ویژه ولی با بعد بسیار کمتر از بعد ماتریس کوواریانس اصلی استفاده می شود. هدف اصلی تعریف تصاویر اولیه با یک ترکیب خطی از ماتریس های Eigenface می باشد که از Projection تصاویر اصلی بر روی بردارهای ویژه انتخاب شده بدست می آیند.

قابلیت مهم سیستم سرعت اجراء و عدم پیچیدگی پیاده سازی می باشد ولی در مورد تغییرات شدت نور و زاویه چهره بسیار ضعیف می باشد.

۲. LDA

در این مدل با استفاده از ساخت ماتریس پراکندگی درون کلاسی و بیرون کلاسی و کاهش فاصله درون کلاسی و افزایش فاصله بین کلاسی اقدام به یکپارچه سازی توزیع داده در زیر فضای بدهست آمده نسبت به حالت Eigenface انجام می شود. از مزیت های این سیستم به توانایی یادگیری دامنه تغییرات درون کلاسی با مهیا سازی تصاویر با تغییرات ویژگی متناظر نام برد ولی برای پیاده سازی یک سیستم مناسب بر این اساس احتیاج به حجم وسیعی از تصاویر بوده که همواره نمی توان تمامی تغییرات را ثبت کرد.

:ICA .^۳

در این مدل اقدام به یافتن بردار پایه ای نموده که داده ها (تصاویر) پس از Projection آنها از لحاظ آماری بی اشتراک بوده که این امر با ماسیمم سازی وابستگی درجه دوم و درجه های بالاتر داده بدست می آید.

دو مدل پیاده سازی معرفی گشته که نقاط ضعف و قوت گروه الگوریتم های محاسبه در زیر فضای دلیل عدم وابستگی به هیچ کدام از ویژگی های داخلی چهره به ارت می برد.

:EBGM .^۴

چهره با یک گراف نمایش داده شده که نقاط آن با استفاده از ضرائب wavelet گابور آن نقطه تعریف می گردند.

:AAM .^۵

یک مدل دستی (در بعضی از سیستم ها اتوماتیک) از نقاط مهم و ویژگی های مهم تصویر چهره ساخته شده و سیستم برای شناسایی اقدام به Fit کردن مدل ساخته شده بر روی تصاویر ورودی می کند. مدل ساخته شده با استفاده از تصاویر درون گروه آموزش Deform شده و ویژگی نهایی برای کلاس بندی فاصله و دقت در مرحله Fit و سوار کردن مدل بر روی چهره می باشد. از مدل معروف Lucas canade برای این مرحله استفاده می شود.

:Edge .^۶

روشی آسان و جذاب میباشد که شرح آن در ادامه آمده است.

مقدمه‌ای بر رنگها

رنگها سرعت پردازش را بالا می‌برند و در تشخیص تفاوت‌های هندسی نمونه چهره‌ها بسیار قوی‌اند. همچنین تجربه نشان میدهد که پوست انسان یک رنگی خاص دارد که به سادگی توسط انسانها تشخیص داده می‌شود. بنابراین تلاش برای به کارگیری مدلسازی رنگ پوست در آشکارسازی چهره، ایده‌ای بود که هم توسط خصوصیات این کار و هم توسط حس عام پیشنهاد شد.

در هنگام ایجاد سیستمی که از رنگ پوست به عنوان ویژگی برای آشکارسازی چهره استفاده می‌کند، محققان عموماً با ۳ مشکل مواجه می‌شوند:

ابتدا اینکه چه فضای رنگی باید انتخاب کنند!

دوم اینکه، دقیقاً رنگ پوست چگونه باید مدلسازی شود!

سوم اینکه، روش پردازش قطعه بندی (تقسیم بندی) رنگ برای آشکارسازی پوست چه باشد!

فضاهای رنگ

به نمایش فیزیکی یک گروه رنگ فضای رنگ گفته می‌شود سه مدل فضای رنگ معروف عبارتند از:

RGB: مورد استفاده برای گرافیک کامپیوتری

YCRCB: مورد استفاده برای سیستم‌های ویدیویی

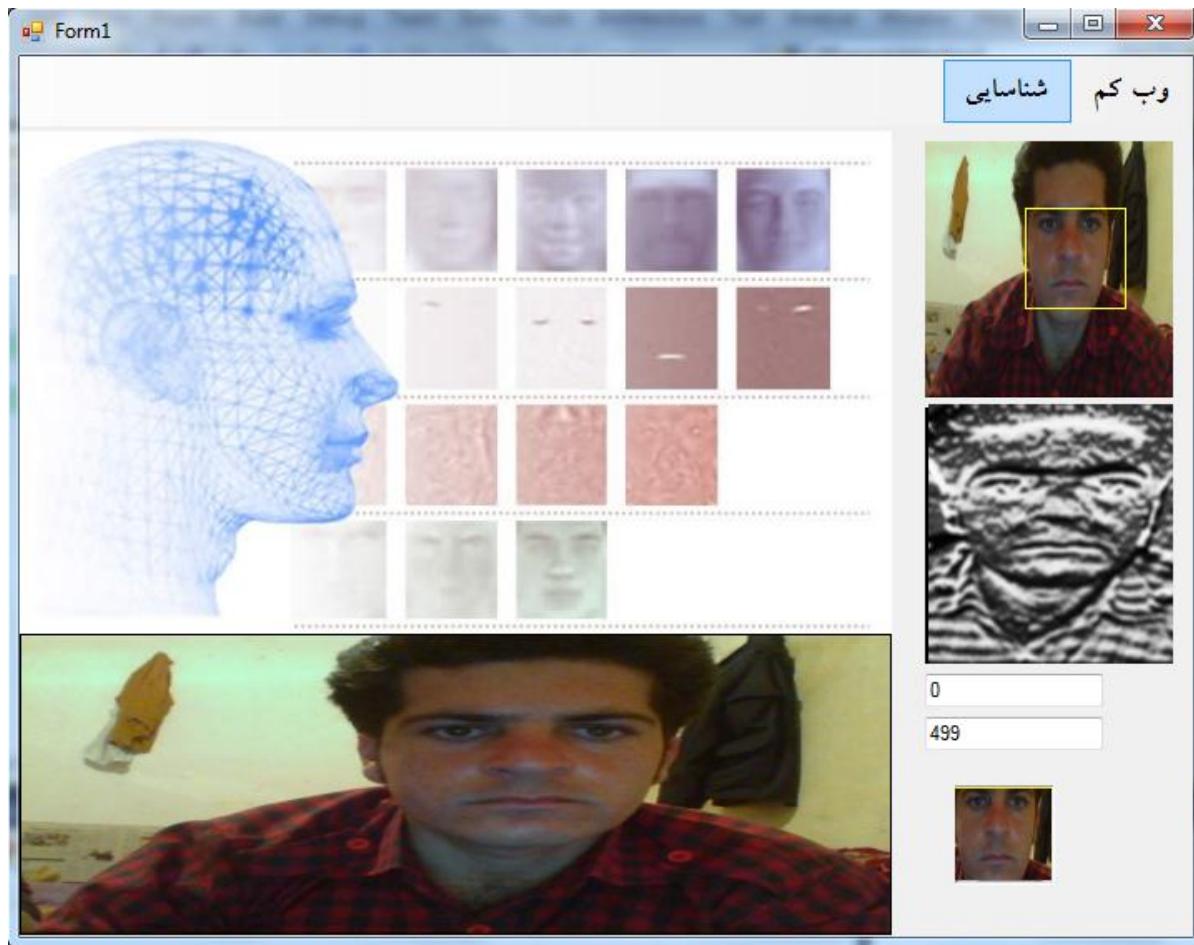
CMYK: مورد استفاده برای چاپ کردن

در **فاز اول پروژه** روش‌های تشخیص پیکسلی پوست را با روش لبه ای بررسی می‌کنیم، که هر پیکسل را به عنوان پوست یا غیر پوست بدون توجه به اطرافش طبقه بندی می‌کند.

در **فاز دوم پروژه** با استفاده از کتابخانه **Aforg** و **EMGU** روش‌های دیگر را بررسی می‌کنیم.

فاز اول ✓

نمای صفحه اصلی



شرح کد:

معمولًا برای تشخیص صورت و پوست از فرمات استفاده نمیشود(RGB)! چون این فرمات وابسته به شرایط نور است(تصویری که از ترکیب ۳ رنگ آبی، سبز، قرمز، که ترکیب RGB را بوجود میآورند برای تشخیص رنگ پوست ضعیف میباشند لذا با تبدیلاتی باید رنگهای اضافی مانند نور را حذف نماییم). نور میتواند تشخیص پوست را با شکست مواجه کند یعنی با تغییر شرایط از جمله نور، محیط خروجی نیز متفاوت میشود. از میان فضاهای رنگ در این پروژه از فضای رنگ YCBCR استفاده شده است. برای اینکه بتوانیم اطلاعات مربوط به نور یا درخشندگی را از تصویر حذف نماییم از تبدیلات زیر میتوان با توجه به کاربرد استفاده کرد:

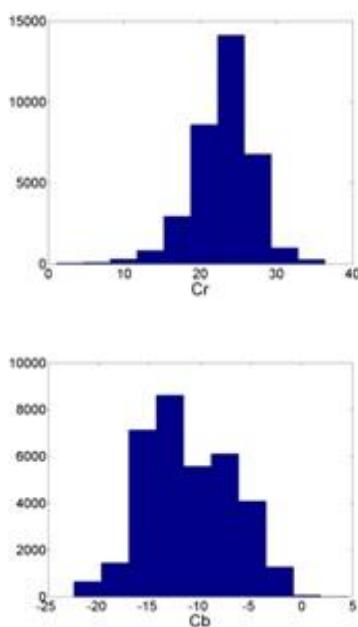
$$Y = 0.299R + 0.587G + 0.114B \rightarrow \text{از بین بردن نور در تصویر}$$

$$Cb = -0.169R - 0.332G + 0.500B$$

$$Cr = 0.500R - 0.419G - 0.081B$$

در فرآیند تشخیص رنگ پوست، هر پیکسل را به عنوان پوست یا غیر پوست بر اساس اجزای رنگ آن طبقه بندی میکنیم. پنجه ره تشخیص رنگ پوست بر اساس میانگین و انحراف استاندارد از کامپوننت CR, CB تعیین شده است.

برای محاسبه ما کامپوننت ۷ را از تصویر ورودی حذف میکنیم و تصویر را به صورت CR یا CB نمایش میدهیم کامپوننتی که در این الگوریتم استفاده شده است کامپوننت CR است به این علت که کامپوننت CR دارای مقادیر بین ۰ تا ۳۸ است. هیستوگرام آنها در شکل زیر نشان داده شده است:



CB به این دلیل استفاده نشده است چون دارای مقادیر بین -۲۵ تا ۰ است و اگر تصویر را با CB تبدیل نماییم مقادیر زیادی که در رنج منفی هستند از بین میروند و تا حدودی تصویر را بعد از محاسبات از دست میدهیم!

جداسازی رنگ‌ها

عملیات جداسازی رنگ بر روی تصویر ورودی اعمال میشود و بعد از جداسازی روشنایی تصویر فوق به صورت زیر در می آید:



برخی از اشیاء که رنگ پوست را به خود گرفته‌اند در تصویر مشاهده می‌شود باید تغییرات زیر را اعمال کنیم تا اشیایی که دارای فضای رنگ پوست هستند از بین روند:



```
public int[] FaceDetection(Form1 Main, PictureBox PicFirst,
    PictureBox Picresult, TextBox TWskin, PictureBox face, TextBox TBskin)
{
    Testdetection = 0;
    Graphics hh = PicFirst.CreateGraphics();
    Bitmap PreResult = new Bitmap(PicFirst.Image);
    Bitmap Result = new Bitmap(PicFirst.Width, Picresult.Height);
    Bitmap HisResult = new Bitmap(PicFirst.Width, Picresult.Height);
    Bitmap Template50 = new Bitmap("sample50.bmp");
    Bitmap faceee = new Bitmap(face.Width, face.Height);
    for (int i = 0; i < PicFirst.Width; i++)
    {
        for (int j = 0; j < PicFirst.Height; j++)
        {
            // گرفتن لایه قرمز از تصویر
            R = PreResult.GetPixel(i, j).R;
            // گرفتن لایه سبز از تصویر
            G = PreResult.GetPixel(i, j).G;
            // گرفتن لایه آبی از تصویر
            B = PreResult.GetPixel(i, j).B;
            // YCBCR به RGB تبدیل
            //Y = 0.299 * R + 0.587 * G + 0.114 * B;
            //Cb = -0.169*R - 0.332*G + 0.500*B;
            Cr = 0.500 * R - 0.419 * G - 0.081 * B;
            // جک کردن برای مقادیر منفی
            if (Cr < 0)
                // تبدیل مقادیر ممکن به مثبت
                Cr = 0;
            HisResult.SetPixel(i, j, Color.FromArgb((int)Cr, (int)Cr, (int)Cr));
        }
    }
}
```



دایره های کوچکی را که در تصویر مشاهده میکنید از دایره مانند صورت کوچکتر هستند که باید آنها را پر نمود:

```
if (Cr > 10)
{
    Cr = 255;
    Result.SetPixel(i, j, Color.FromArgb((int)Cr, (int)Cr, (int)Cr));
}
```

در این مرحله برای جدا کردن مناطق یکپارچه از الگوریتم تشخیص لبه Roberts Cross استفاده میکنیم:

```
// عمل فیلتر روی تصویر
FiltersSequence processingFilter = new FiltersSequence();
// نمایاندن لبه ها
processingFilter.Add(new Edges());
WPresentdetection=0;
BPresentdetection1= 0;
Picresult.Image = processingFilter.Apply( Result);
```

در این مرحله برای تشخیص صورت، تصویر بدست آمده را با Template ای که در برنامه قرار داده ایم مقایسه میکنیم (این Template نتیجه ۲۵ صورت تست شده هست)



```
for (int d = 0; d < 150-50; d+=20)
{
    Testdetection = 0;
    WPresentdetection = 0;
    BPresentdetection1 = 0;
    for (int c = 0; c < 150-50; c+=20)
    {
        Testdetection = 0;
        WPresentdetection = 0;
        BPresentdetection1 = 0;
        for (int w = 0; w < Template50.Width; w++)
        {
            for (int h = 0; h < Template50.Height; h++)
            {
                if (Template50.GetPixel(w, h).B == 255)
                {
                    if (Result.GetPixel(d + w, c + h).B == Template50.GetPixel(w, h).B)
                    {
                        WPresentdetection++;
                    }
                }
            }
        }
    }
    TWskin.Text = WPresentdetection.ToString();
    TBskin.Text = BPresentdetection1.ToString();
}
```

```
if (WPresentdetection > 377) //&&BPresentdetection1>1579
{
    hh.DrawRectangle(new Pen(Color.Red), d, c, 60, 60);
    for (int i = 0; i < 60; i++)
    {
        for (int j = 0; j < 60; j++)
        {
            r = PreResult.GetPixel(i + d, j + c).R;
            facee.SetPixel(i, j, Color.FromArgb((int)r, (int)r, (int)r));
            face.Image = facee;
        }
    }
    TWskin.Text = WPresentdetection.ToString();
    TBskin.Text = BPresentdetection1.ToString();
    WPresentdetection = 0;
    BPresentdetection1 = 0;
    Testdetection = 1;
    break;
}
if (Testdetection == 1 || Testdetection == 2)
{
    WPresentdetection = 0;
    BPresentdetection1 = 0;
    Testdetection = 0;
    break;
}
}
```

فاز دوم ✓

در وحله اول برنامه با پنجره **User Control** برای تست صحت رمز عبوری اجرا میشود:



صحت کلمه عبور و پسورد را با تابع زیر در برنامه بررسی نموده‌ام:

```

if (textBox1.Text == "1")
    if (textBox2.Text == "2")
    {
        MessageBox.Show("آمدید خوش چهره تشخیص سیستم به.", "WellCome");
        Start StartMenu = new Start();
        StartMenu.Show();
        StartMenu.Show();
        this.Hide();
    }

if (textBox1.Text != "1")
{
    MessageBox.Show("کنید دقت", "است اشتباه عبور کلمه!");
    textBox1.Text = "";
    textBox1.Focus();
    pictureBox2.Visible = false;
    picb1.Visible = true;

}

if (textBox2.Text != "2")
{
    MessageBox.Show("کنید دقت", "است اشتباه پسورد!");
    textBox2.Text = "";
    textBox2.Focus();
    pictureBox3.Visible = false;
    picb2.Visible = true;

}

if (textBox1.Text != "1" & textBox2.Text != "2")
{
    textBox1.Text = "";
    textBox1.Focus();
    picb1.Visible = true;
    picb2.Visible = true;
}

```

صفحه اصلی پروژه - فاز ۲



همانطور که مشاهده میشود پردازش روی تصاویر و وب کم در دو منوی جدا قرار گرفته‌اند. در کار با تصاویر از دو کتابخانه معروف و جامع در زمینه پردازش تصویر به نامهای Emgu و Aforg استفاده شده است بسته به تنوع کار میتوانید یکی از آنها را برای کار بر روی تصاویر انتخاب نمایید:

برای جلوگیری از حجم گزارش کار یکی را در ادامه بررسی مینماییم:



کتابخانه **Emgu** متدهای زیادی را در اختیار کاربر قرار میدهد تا بتواند به جزئیات ریز یک تصویر دسترسی داشته باشد "البته بین این دو کتابخانه معروف نمیتوان یکی را به عنوان برترین نسبت به دیگری انتخاب نمود چون بسته به کاربرد، هر کدام کارایی بیشتری نسبت به یکدیگر دارند!"
به عنوان مثال با **Emgu** مکان ماوس را با چند خط کد ساده میتوان بدست آورد و در برنامه هایی که نیاز به مکان ماوس هست از آن مختصات استفاده شود (مانند بازی هایی که بیشتر با ماوس سروکار دارند)

```
private void pictureBox1_MouseMove_1(object sender, MouseEventArgs e)
{
    if (pictureBox1.Image != null)
    {
        lbl1.Text = "X: " + e.X.ToString();
        lbl2.Text = "Y: " + e.Y.ToString();
    }
}
```

عکس بارشده را پس از تغییر میتوان هم در **حافظه** و هم در **پایگاه داده** برای پردازش ذخیره نمود.

- کدی که عکس لود شده را در **حافظه** ذخیره میکند در زیر آمده است:

```
string strFileName; → افراد عکس را ذخیره میکنند.
if (pictureBox1.Image != null) → نمایش پنجره ذخیره
{
    SaveFileDialog save = new SaveFileDialog(); → عکس به صورت پنهان فوشی پاپسوند JPEG ذخیره میشود
    save.DefaultExt = "jpg"; → اگر بخواهیم پاپسوند دیگر تصویر ذخیره شود پایه کد زیر را اعمال نماییم که با وارد نمودن آن کد جمهه پاکس پسند به پنجره ذخیره اضافه میشود:
    // save.FileName = strFileName;

    =====
    save.Filter = "Image files (*.jpg,*.png,*.tif,*.bmp,*.gif)|*.jpg;*.png;*.tif;*.bmp;*.gif|JPEG
    fil" + "es (*.jpg)|*.jpg|PNG files (*.png)|*.png|TIF files"
    " (*.tif)|*.tif|BMP files (*.bmp)|*.bmp|GIF files (*.gif)|*.gif";
    =====

    Save as type: Image files (*.jpg,*.png,*.tif,*.bmp,*.gif) → در کد زیر تصویر را به پایگاهی در حافظه توسط پافرها تبدیل مینماییم:
    =====
    save.FilterIndex = 1;
    save.OverwritePrompt = true;
    save.Title = "Demo Save File Dialog";
    =====
    در کد زیر تصویر را به پایگاهی در حافظه توسط پافرها تبدیل مینماییم:
    MemoryStream ms = new MemoryStream();
    bim = pic.ToBitmap();
    bim.Save(ms, ImageFormat.Jpeg);
    byte[] bmpbyte = ms.GetBuffer();
    // bim.Dispose();
    ms.Close();
    =====

    if (save.ShowDialog() == DialogResult.OK) → اگر ok را کلیک کنیم:
    {
        // نام عکس
        strFileName = save.FileName;
        نوشته جریان ورودی / خروجی پایگاهی موجود در حافظه رم را در چاکی که ما مشخص نمودهایم مینویسد:
        System.IO.File.WriteAllBytes(strFileName, bmpbyte);
    }
    else
    {
        MessageBox.Show(" لطفا عکس مورد نظر را انتخاب نماید بعد اقدام به ذخیره آن نمایید " +
        this.Refresh();
    }
}
```

- کدی که عکس لود شده را در پایگاه داده ذخیره میکند در زیر آمده است:

```

{
if (pictureBox1.Image != null)
{
    try
    {
        byte[] imageData = ReadFile(u2); نحواندن پایگاهی عکس و ذخیره در اولیه
        =====
اتصال به پایگاه داده
SqlConnection CN = new SqlConnection("Data Source=(local);Initial Catalog=FaceDetection;Integrated Security=true;");

گذخیره در جدول در پایگاه داده
(عکس_محتویات, عکس_آدرس)
values ("عکس_محتویات@, عکس_آدرس@");

SqlCommand SqlCom = new SqlCommand(qry, CN);
SqlCom.Parameters.Add(new SqlParameter("@عکس_آدرس", (object)u2));
SqlCom.Parameters.Add(new SqlParameter("@عکس_محتویات", (object)imageData));
//Open connection and execute insert query.
CN.Open();
SqlCom.ExecuteNonQuery();
CN.Close();

MessageBox.Show ("شد ذخیره موفقیت با نظر مورد تصویر.");
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
}

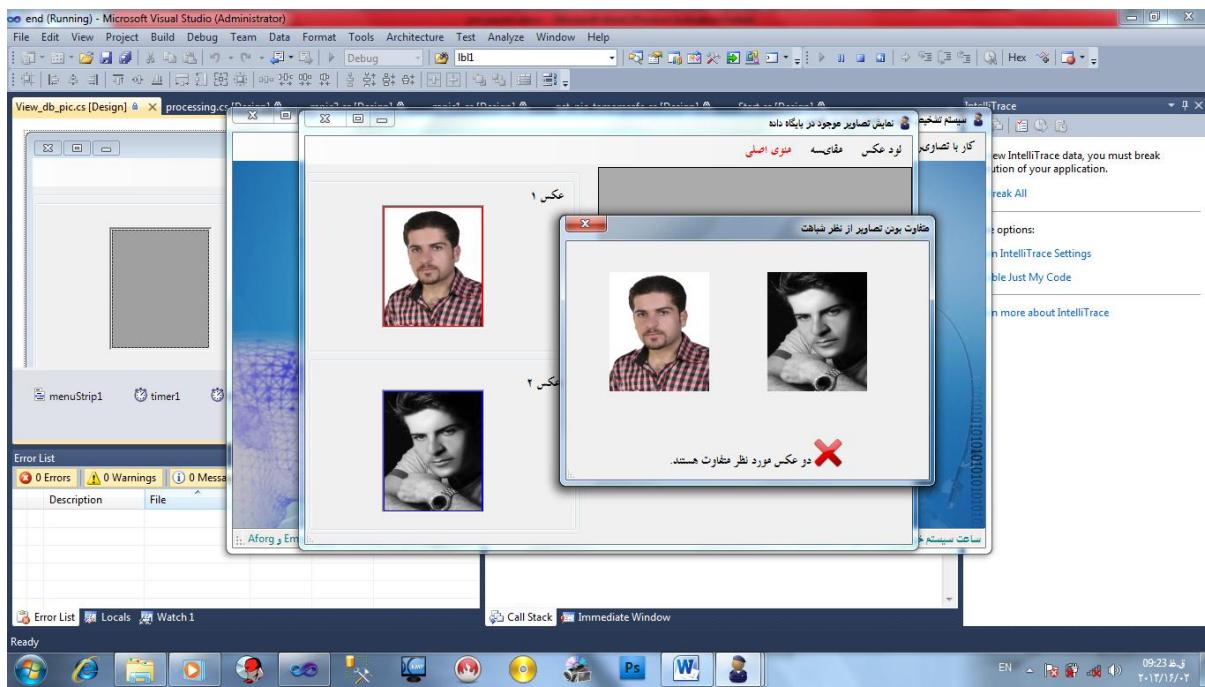
else{
    ("لطفا عکس مورد نظر را انتخاب نمایید بعد اقدام به ذخیره آن نمایید")
    MessageBox.Show("this.Refresh();");
}
}

```

نمایش اطلاعات پایگاه داده

مقایسه دو عکس از حافظه یا پایگاه داده از نظر تشابه

- برای جلوگیری از حجم گزارش کار برای دیدن کد این فرم به پروژه مراجعه شود.



نمایش تصویر از طریق وب کم

در این منو دو گزینه با عنوان **حالت معمولی** و **حالت تشخیص چهره** قرار داده شده است که بنا به کاربرد یکی از آنها را میتوان استفاده نمود:

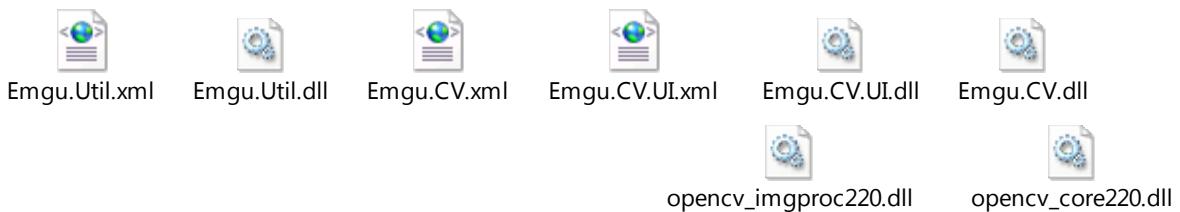


حالت تشخیص چهره را انتخاب میکنیم و مختصوی در مورد کد آن توضیح میدهم:

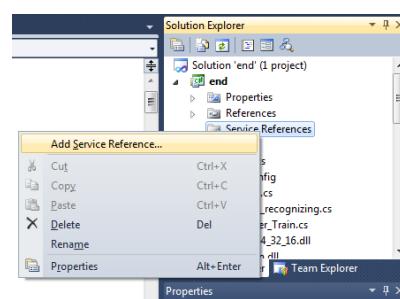


روش بار کردن وبکم

برای کار با **Emge** باید فایل‌های زیر به برنامه افزوده شود:



این فایلها را به صورت زیر به برنامه اعمال کنید:



و در قسمت **using** باید فضاهای نام زیر افزوده شود:

```
using Emgu.CV.UI;
using Emgu.CV;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;
```

فراخوانی تصویر ورودی

```
void FrameGrabber_Parrellel(object sender, EventArgs e)
{
    currentFrame = grabber.QueryFrame().Resize(320, 240,
    Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
```

گرفتن چریان از وب گم

```
if (currentFrame != null)
{
    gray_frame = currentFrame.Convert<Gray, Byte>();
```

تشخیص صورت

```
MCvAvgComp[][] facesDetected = gray_frame.DetectHaarCascade(Face, 1.2,
10, Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(50, 50));
```

//Action for each element detected

```
Parallel.ForEach(facesDetected[0], face_found =>
{
    try
    {
```

```
        result = currentFrame.Copy(face_found.rect).Convert<Gray,
byte>().Resize(100, 100, Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
        gray_frame = result.Convert<Gray, Byte>();
        result._EqualizeHist();
        pictureBox1.Image = new Bitmap(result.ToBitmap());
        uni = result.ToBitmap();
        uni1 = currentFrame.ToBitmap();
        Point n0 = new Point(20, 20);
```

کشیدن مستطیل دور صورت

```
        currentFrame.Draw(face_found.rect, new Bgr(Color.Green), 2);
```

```

    }
    catch
    {
    }
});
```

نمایش صورت در چهاره عکس گوچگ

```
image_PICBX.Image = currentFrame.ToBitmap();
```

```
}
```

لازم به ذکر است که توابعهای زیادی برای تشخیص چهره در برنامه استفاده شده است این تکه کد نمونه‌ای از کارآیی Emgu برای آشنایی با دستورات آن بود و برای فهمیدن چگونگی کار از مطالب میخواهیم به کدهای بکار رفته در پروژه مراجعه کند.

منابع

بسیاری از منابع را از استاد محترم تحويل نموده‌ام و بعضی از سایتهای مراجعه شده و کتابها در زیر لیست شده‌اند.

کتابها

1. Probabilistic recognition of human faces from video
2. On Face Recognition using Gabor Filters
Al-Amin Bhuiyan, and Chang Hong Liu
3. Bayesian Face Recognition
Baback Moghaddam
Tony Jebara
Alex Pentland
4. An Introduction to Object Recognition

سایتها

1. www.farsibooks.ir
2. www.Msjavan.tk
3. www.codeproject.com

بسیاری دیگر از سایتها و کتابها استفاده شده که لیست آنها در DVD موجود میباشد.