PWO 3D-VISION

Age and gender example in C#

Bart Ribbens

KdG Expert
Karel de Grote Hogeschool

1.1 Age and gender example in C#

1.1.1 Register for Microsoft Azure face API

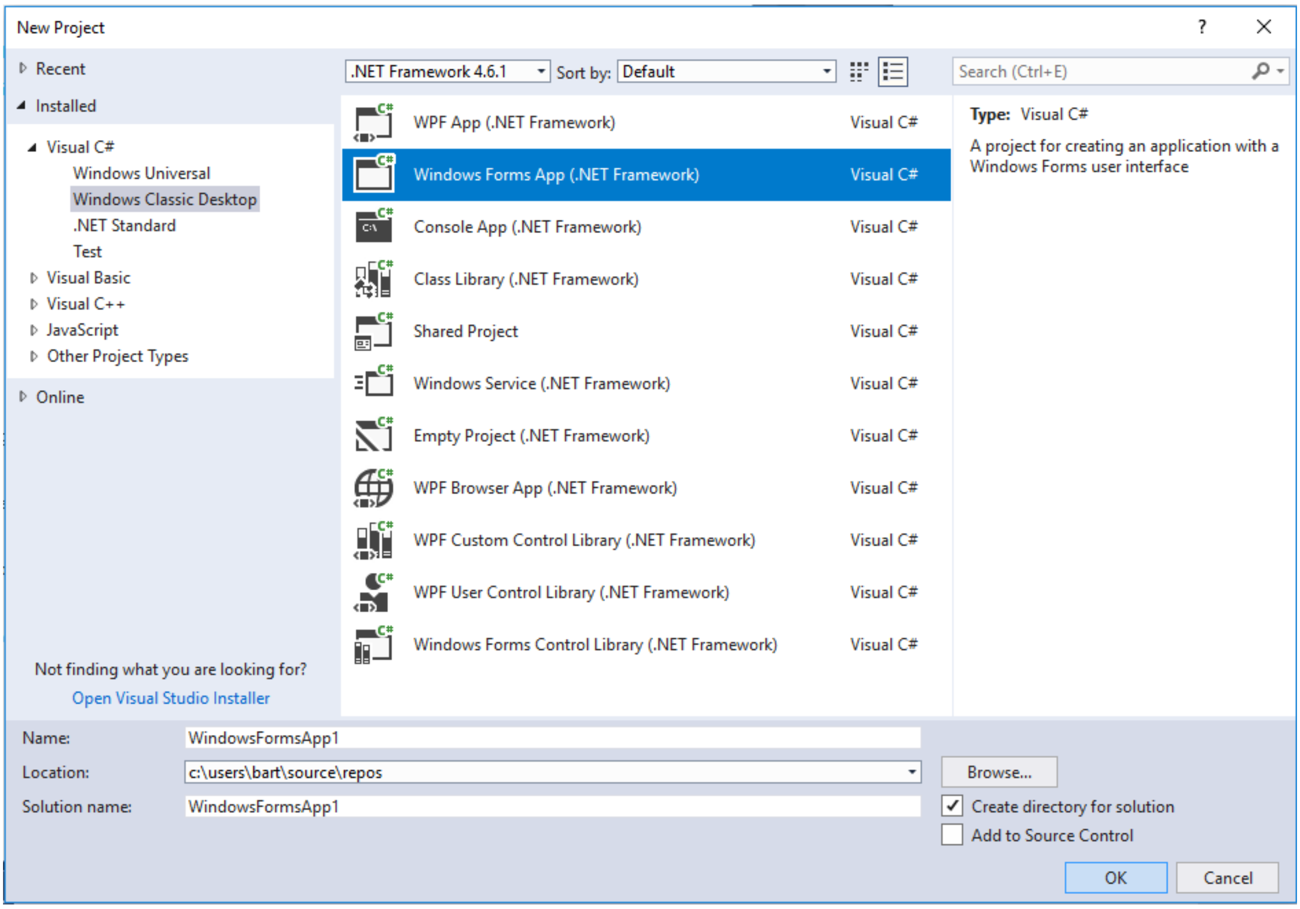
Information can be found here:

- https://azure.microsoft.com/en-us/try/cognitive-services/?api=face-api

You need an API key to use the software. There is a 30-day trial and afterwards you can register for a free version but verification with credit card is obligated.

You need the key in 1.1.4.2 ("PutYourKeyHere")

1.1.2 Create standard C# application



New Project

WPF App (.NET Framework)	Visual C#
Windows Forms App (.NET Framework)	Visual C#
Console App (.NET Framework)	Visual C#
Class Library (.NET Framework)	Visual C#
Shared Project	Visual C#
Windows Service (.NET Framework)	Visual C#
Empty Project (.NET Framework)	Visual C#
WPF Browser App (.NET Framework)	Visual C#
WPF Custom Control Library (.NET Framework)	Visual C#
WPF User Control Library (.NET Framework)	Visual C#
Windows Forms Control Library (.NET Framework)	Visual C#

Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: WindowsFormsApp1

Location: c:\users\bart\source\repos

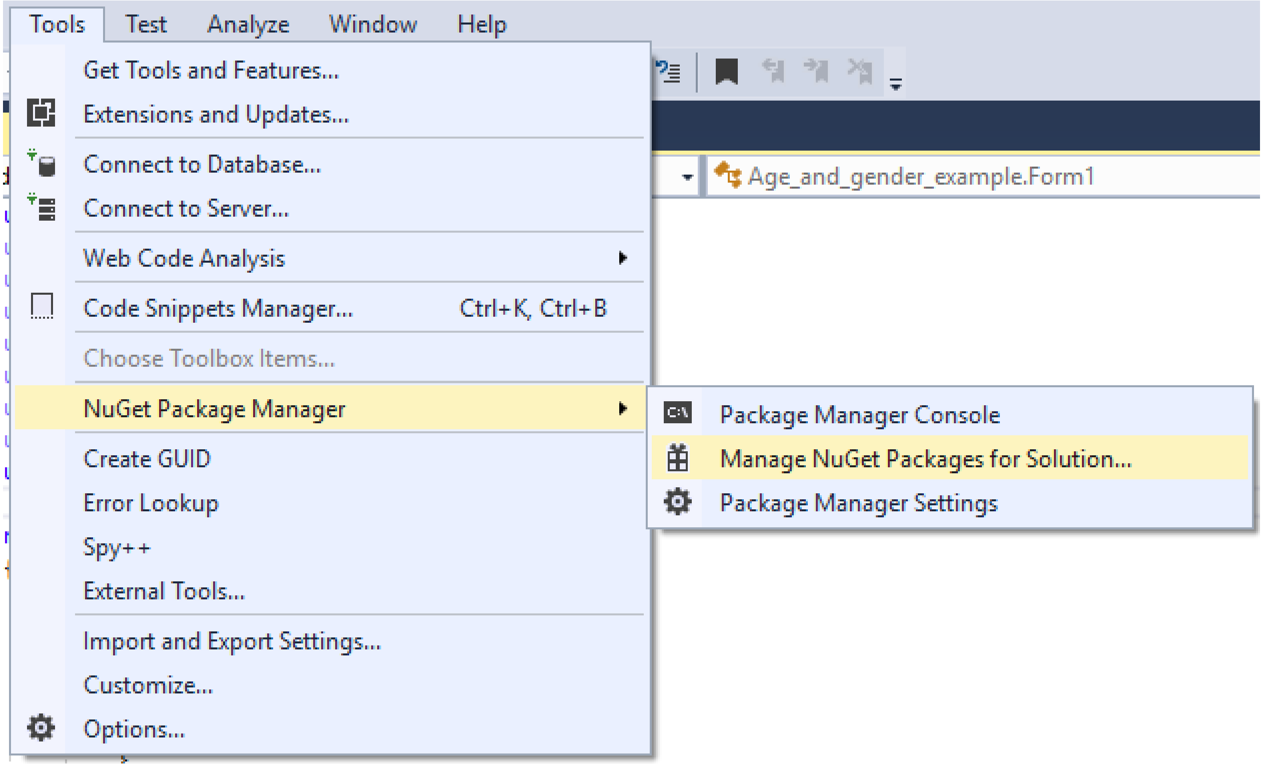
Solution name: WindowsFormsApp1

☑ Create directory for solution

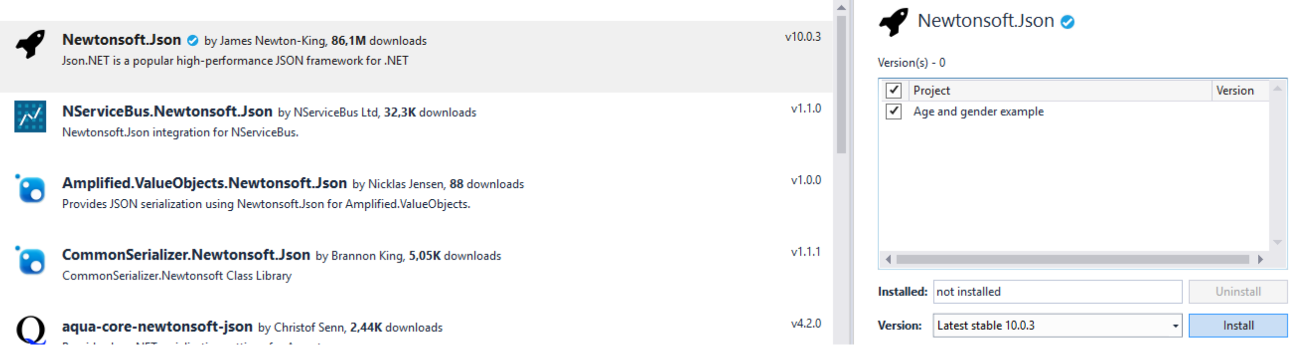
□ Add to Source Control

1.1.3 Manage NuGet Packages

1.1.3.1 Go to Nuget Package Manager



1.1.3.2 Install Newtonsoft_Json



Newtonsoft.Json by James Newton-King, **86,1M** downloads v10.0.3
Json.NET is a popular high-performance JSON framework for .NET

NServiceBus.Newtonsoft.Json by NServiceBus Ltd, **32,3K** downloads v1.1.0
Newtonsoft.Json integration for NServiceBus.

Amplified.ValueObjects.Newtonsoft.Json by Nicklas Jensen, **88** downloads v1.0.0
Provides JSON serialization using Newtonsoft.Json for Amplified.ValueObjects.

CommonSerializer.Newtonsoft.Json by Brannon King, **5,05K** downloads v1.1.1
CommonSerializer.Newtonsoft Class Library

aqua-core-newtonsoft-json by Christof Senn, **2,44K** downloads v4.2.0

Newtonsoft.Json

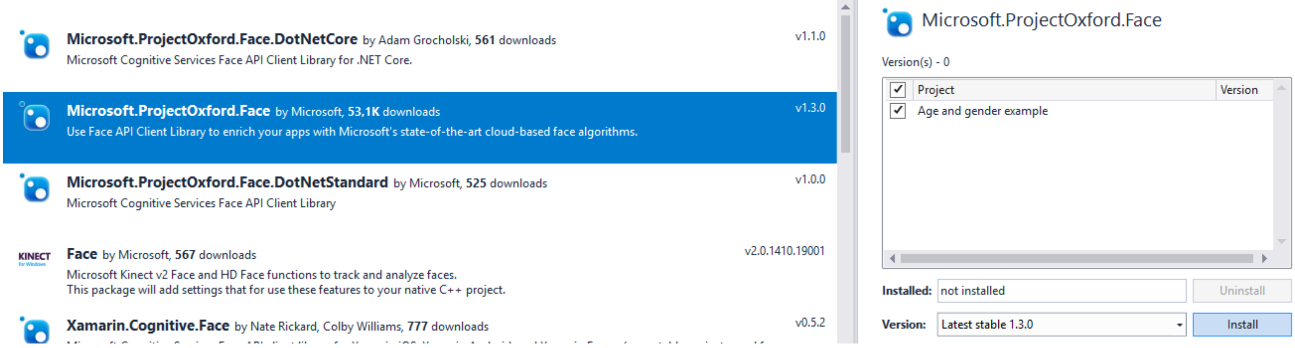
Version(s) - 0

☑ Project	Version
☑ Age and gender example	

Installed: not installed

Version: Latest stable 10.0.3

1.1.3.3 Install ProjectOxford Face



Microsoft.ProjectOxford.Face.DotNetCore by Adam Grocholski, **561** downloads v1.1.0
Microsoft Cognitive Services Face API Client Library for .NET Core.

Microsoft.ProjectOxford.Face by Microsoft, **53,1K** downloads v1.3.0
Use Face API Client Library to enrich your apps with Microsoft's state-of-the-art cloud-based face algorithms.

Microsoft.ProjectOxford.Face.DotNetStandard by Microsoft, **525** downloads v1.0.0
Microsoft Cognitive Services Face API Client Library

Face by Microsoft, **567** downloads v2.0.1410.19001
Microsoft Kinect v2 Face and HD Face functions to track and analyze faces.
This package will add settings that for use these features to your native C++ project.

Xamarin.Cognitive.Face by Nate Rickard, Colby Williams, **777** downloads v0.5.2

Microsoft.ProjectOxford.Face

Version(s) - 0

☑ Project	Version
☑ Age and gender example	

Installed: not installed

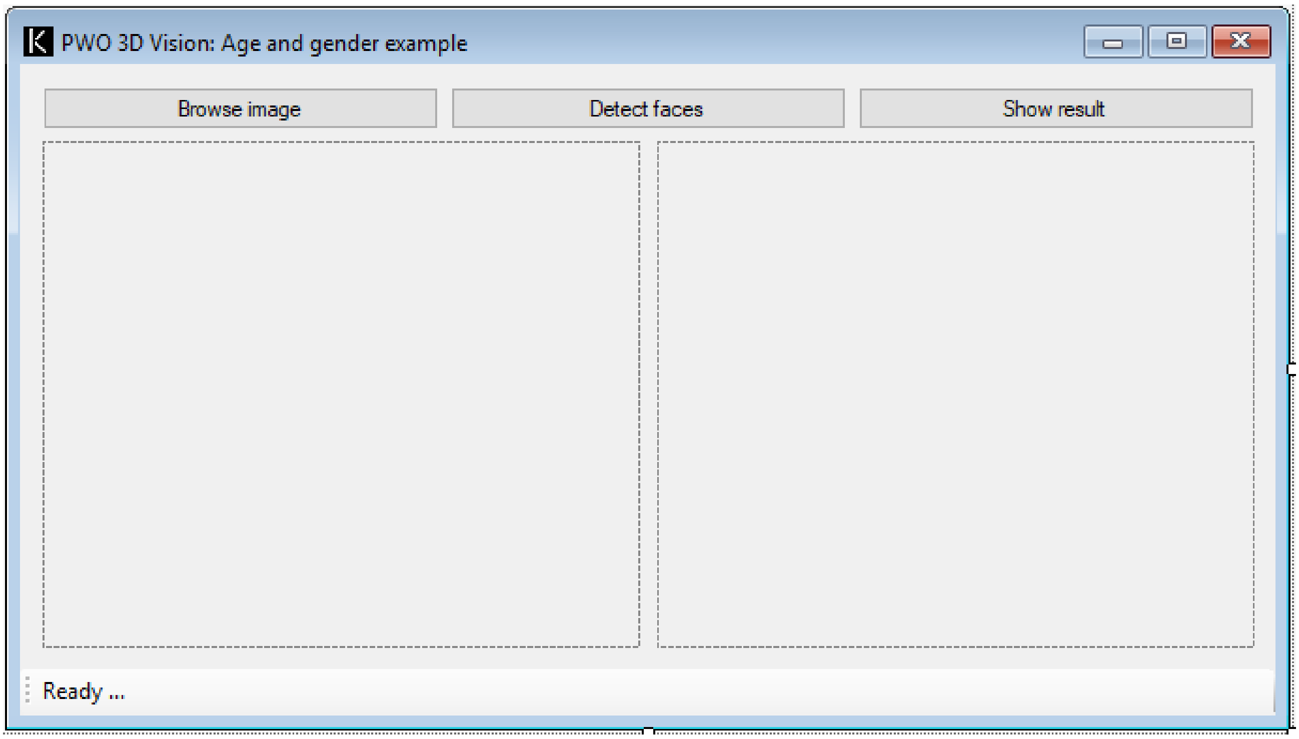
Version: Latest stable 1.3.0

1.1.4 Add references

- PresentationCore
- PresentationFramework
- WindowsBase

1.1.5 Design the program form

- Add a button to browse for an image
- Add a picturebox to show the original image
- Add a picturebox to show the resulting image
- Add a button to detect the faces
- Add a button to show the result
- Add Toolstrip
- Add a Toolstriplabel



1.1.6 Code the example

1.1.6.1 Include libraries

```
using System.IO;

using Microsoft.ProjectOxford.Common.Contract;

using Microsoft.ProjectOxford.Face;

using Microsoft.ProjectOxford.Face.Contract;
```

1.1.6.2 *Insert Azure Face API key*

```
private readonly IFaceServiceClient faceServiceClient = new
FaceServiceClient("PutYourKeyHere",
"https://westeurope.api.cognitive.microsoft.com/face/v1.0");
```

1.1.6.3 *Create the global variables*

```
Bitmap srcImage;                // The original image
Face[] faces;                   // The list of detected faces.
String[] faceDescriptions;      // The list of descriptions for the detected faces.
```

1.1.6.4 *Program btnBrowseImage_Click*

```
private void btnBrowseImage_Click(object sender, EventArgs e)
{
    OpenFileDialog openDlg = new OpenFileDialog
    {
        Filter = "JPEG Image(*.jpg)|*.jpg"
    };

    if (openDlg.ShowDialog() != DialogResult.OK)
    {
        return;
    }
    else
    {
        imageFilePath = openDlg.FileName;
        tslStatus.Text = "Image loaded: " + imageFilePath;
        srcImage = (Bitmap)Bitmap.FromFile(imageFilePath);
        pbOriginal.Image = srcImage;
        faces = null;
        pbResult.Image = null;
    }
}
```

1.1.6.5 *Program btnDetectFaces_Click*

```
private async void btnDetectFaces_Click(object sender, EventArgs e)
{
    // Detect any faces in the image.
    tslStatus.Text = "Detecting faces ...";
    faces = await UploadAndDetectFaces();
    tslStatus.Text = String.Format("Detection Finished. {0} face(s) detected",
faces.Length);
}
```

1.1.6.6 *Program UploadAndDetectFaces*

```
private async Task<Face[]> UploadAndDetectFaces()
{
    // The list of Face attributes to return.
    IEnumerable<FaceAttributeType> faceAttributes = new FaceAttributeType[] {
        FaceAttributeType.Gender,
        FaceAttributeType.Age,
        FaceAttributeType.Smile,
        FaceAttributeType.Emotion,
        FaceAttributeType.Glasses,
        FaceAttributeType.Hair
    };

    // Call the Face API.
    try
    {
        using (Stream imageFileStream = File.OpenRead(imageFilePath))
        {
            Face[] faces = await faceServiceClient.DetectAsync(imageFileStream,
returnFaceId: true, returnFaceLandmarks: false, returnFaceAttributes: faceAttributes);
            return faces;
        }
    }
    // Catch and display Face API errors.
    catch (FaceAPIException f)
    {
        MessageBox.Show(f.ErrorMessage, f.ErrorCode);
        return new Face[0];
    }
    // Catch and display all other errors.
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Error");
        return new Face[0];
    }
}
```

1.1.6.7 Program FaceDescription

```
private string FaceDescription(Face face)
{
    StringBuilder sb = new StringBuilder();

    sb.Append("Face: ");

    // Add the gender, age, and smile.
    sb.Append(face.FaceAttributes.Gender);
    sb.Append(", ");
    sb.Append(face.FaceAttributes.Age);
    sb.Append(", ");
    sb.Append(String.Format("smile {0:F1}%, ", face.FaceAttributes.Smile * 100));

    // Add the emotions. Display all emotions over 10%.
    sb.Append("Emotion: ");
    EmotionScores emotionScores = face.FaceAttributes.Emotion;
    if (emotionScores.Anger >= 0.1f) sb.Append(String.Format("anger {0:F1}%, ",
emotionScores.Anger * 100));
    if (emotionScores.Contempt >= 0.1f) sb.Append(String.Format("contempt {0:F1}%, ",
emotionScores.Contempt * 100));
    if (emotionScores.Disgust >= 0.1f) sb.Append(String.Format("disgust {0:F1}%, ",
emotionScores.Disgust * 100));
    if (emotionScores.Fear >= 0.1f) sb.Append(String.Format("fear {0:F1}%, ",
emotionScores.Fear * 100));
    if (emotionScores.Happiness >= 0.1f) sb.Append(String.Format("happiness {0:F1}%, ",
emotionScores.Happiness * 100));
    if (emotionScores.Neutral >= 0.1f) sb.Append(String.Format("neutral {0:F1}%, ",
emotionScores.Neutral * 100));
    if (emotionScores.Sadness >= 0.1f) sb.Append(String.Format("sadness {0:F1}%, ",
emotionScores.Sadness * 100));
    if (emotionScores.Surprise >= 0.1f) sb.Append(String.Format("surprise {0:F1}%, ",
emotionScores.Surprise * 100));

    // Add glasses.
    sb.Append(face.FaceAttributes.Glasses);
    sb.Append(", ");

    // Add hair.
    sb.Append("Hair: ");
    // Display baldness confidence if over 1%.
    if (face.FaceAttributes.Hair.Bald >= 0.01f) sb.Append(String.Format("bald {0:F1}% ",
face.FaceAttributes.Hair.Bald * 100));

    // Display all hair color attributes over 10%.
    HairColor[] hairColors = face.FaceAttributes.Hair.HairColor;
    foreach (HairColor hairColor in hairColors)
    {
        if (hairColor.Confidence >= 0.1f)
        {
            sb.Append(hairColor.Color.ToString());
            sb.Append(String.Format(" {0:F1}% ", hairColor.Confidence * 100));
        }
    }

    // Return the built string.
    return sb.ToString();
}
```

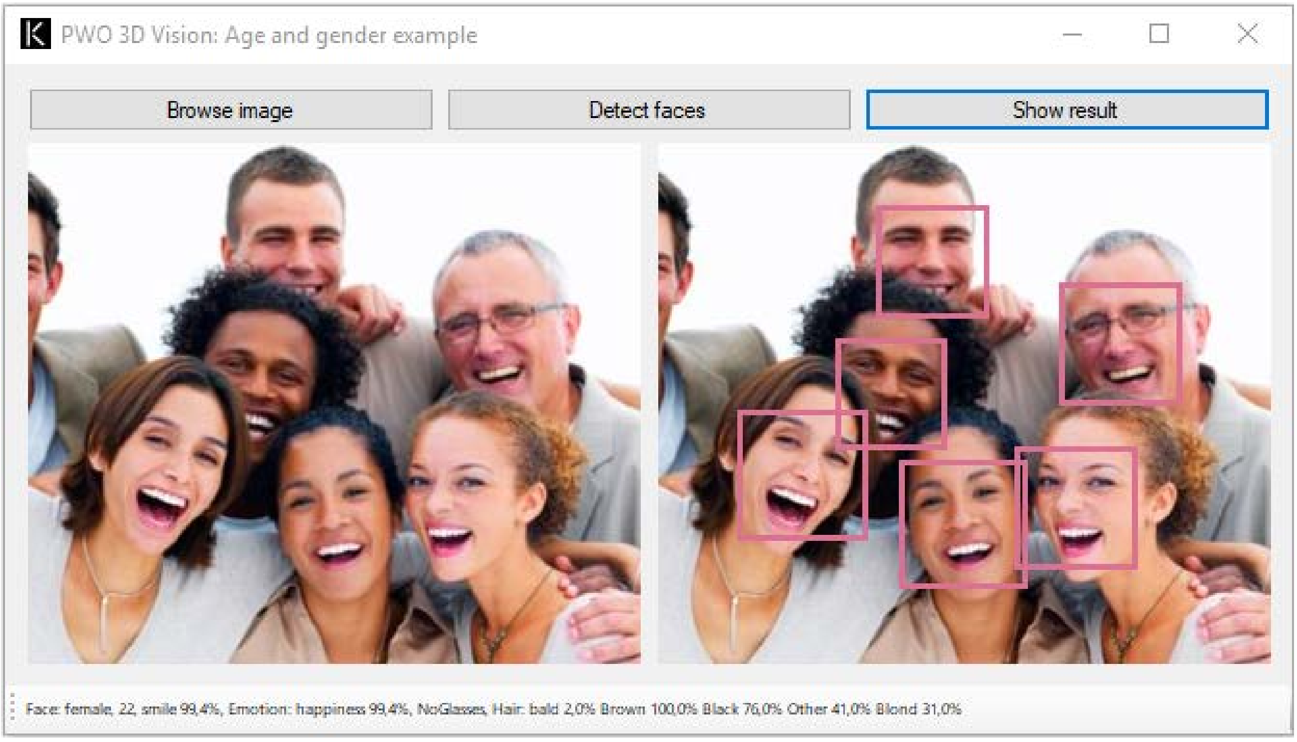
1.1.6.8 *Program btnShowResult_Click*

```
private void btnShowResult_Click(object sender, EventArgs e)
{
    if (faces.Length > 0)
    {
        faceDescriptions = new String[faces.Length];

        pbResult.Image = pbOriginal.Image;
        using (Graphics g = Graphics.FromImage(pbResult.Image))
        {
            for (int i = 0; i < faces.Length; ++i)
            {
                Pen myPen = new Pen(Color.PaleVioletRed, 3);

                g.DrawRectangle(myPen, faces[i].FaceRectangle.Left,
faces[i].FaceRectangle.Top, faces[i].FaceRectangle.Width, faces[i].FaceRectangle.Height);

                // Store the face description.
                faceDescriptions[i] = FaceDescription(faces[i]);
            }
        }
        pbResult.Invalidate();

        // Set the status bar text.
        tslStatus.Text = "Place the mouse pointer over a face to see the face description.";
    }
}
```

1.1.6.9 Program pbResult_MouseMove

```
private void pbResult_MouseMove(object sender, MouseEventArgs e)
{
    // If the REST call has not completed, return from this method.
    if (faces == null)
        return;

    // Find the mouse position relative to the image.
    Point mouseXY = new Point(e.X, e.Y);

    //// Check if this mouse position is over a face rectangle.
    bool mouseOverFace = false;

    for (int i = 0; i < faces.Length; ++i)
    {
        FaceRectangle fr = faces[i].FaceRectangle;
        double left = fr.Left;
        double top = fr.Top;
        double width = fr.Width;
        double height = fr.Height;

        // Display the face description for this face if the mouse is over this face
rectangle.
        if (mouseXY.X >= left && mouseXY.X <= left + width && mouseXY.Y >= top && mouseXY.Y
<= top + height)
        {
            tslStatus.Font = new Font("Segoe UI", 6);
            tslStatus.Text = faceDescriptions[i];
            mouseOverFace = true;
            break;
        }
    }
    // If the mouse is not over a face rectangle.
    if (!mouseOverFace)
    {
        tslStatus.Font = new Font("Segoe UI", 9);
        tslStatus.Text = "Place the mouse pointer over a face to see the face description.";
    }
}
```

1.1.7 Final result



Face: female, 22, smile 99,4%, Emotion: happiness 99,4%, NoGlasses, Hair: bald 2,0% Brown 100,0% Black 76,0% Other 41,0% Blond 31,0%