Face Recognition WPF Application

WPF Application - Face Recognition

This is a sample WPF (Windows Presentation Foundation) Application for Face Recognition. This application uses the webcam to capture picture and then detects who is in the picture. It uses the Microsoft Azure Face API to detect the face and WebCam_Capture package to access and capture the image using the built-in webcam of your device. Download or clone the file to your system. Open the Face folder and open face.sln file on Visual Studio 2017 edition and .NET framework version 4.7.03056.

Face Recognition using Microsoft Azure and .NET Framework

The project first trains the system using the FACE API of Microsoft Azure using subscription key (Make a Microsoft Azure account or you can get free seven day subscription key from here.) After training, it uses the webcam to capture image and process it to identify the person in the image, alternatively you can also browse and upload the image for the same. It then returns whether the person in the image is identified or not.

Requirements

- Visual Studio 2017
- .NET Framework 4.7.03056
- Microsoft Azure Account
- FACE API subscription key
- Built-in Webcam

How to run

Download or clone the project to your system. Open the folder and open face.sln file. This will open the solution including all the projects in it. Open nu-get console, add nu-get packages:

1. Install-Package Microsoft.Azure.CognitiveServices.Vision.Face -Version 2.0.0-preview
2. Install-Package Microsoft.ProjectOxford.Face -Version 1.4.0

The first package is used to access the Face API and FaceServiceClient , while the second package is used to train large dataset i.e., Million Scale support via LargePersonGroup/LargeFaceList.

This Solution consists of three projects:

1. Face : To train the dataset
2. Testing : To run the application

Face Project:

Open the MainWindow.xaml.cs file, this file makes an object of FaceServiceClient to access the FACE API, so authorize the API call and then edit the subscription key to your key and also change your access location to the place you are accessing the Azure cloud space.

Now Create the person ID to add person to your person group to train the system. Add images of each friend to that particular ID. Sample list of friends along with their images is already present in this project, you can add more person to this list to train your system. For training the data you need to create a new personGroupId every time. Change the path given to the images and files according to your system. Now run the project, and thus your system is trained.

Testing Project:

Open the MainWindow.xaml.cs file of testing project. Add reference WebCam_Capture.dll to the Testing Project under references. Declare a FaceServiceClient object to authorize the API call same way as we did in the Face project. MainWindow.xaml is the file which is used to design the UI of the application.

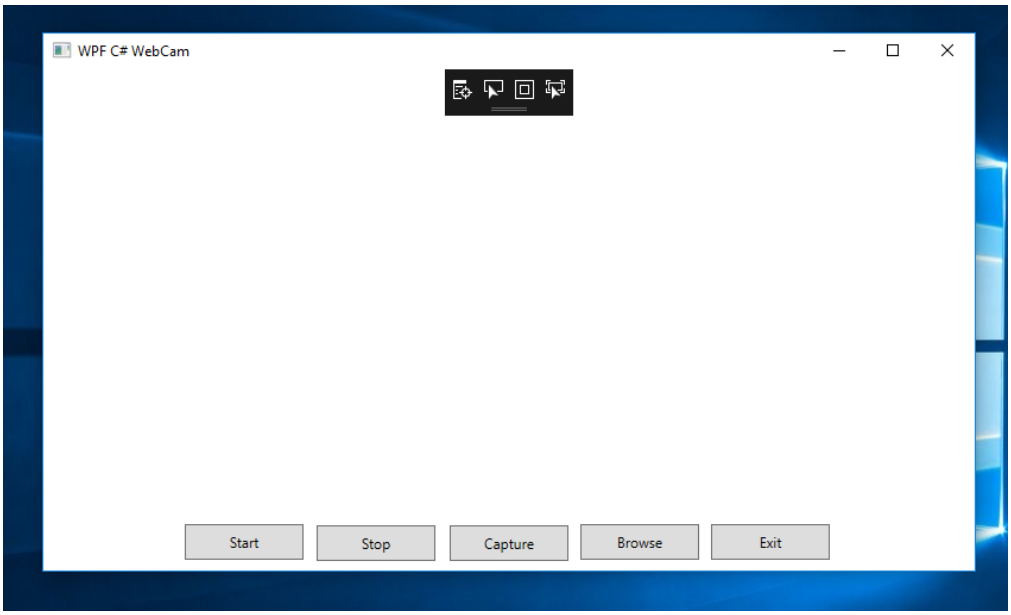
In this we have defined 5 buttons whose functionalities are:

1. Start: Is used to start the webcam.
2. Stop: To stop the webcam.
3. Capture: Saves the image with default name and at default location. After saving, it runs and tells whether the person in the picture is identified or not.
4. Browse: You can also browse and upload an image and then it tells whether the person in the picture is identified or not.
5. Exit: Used to close the app.

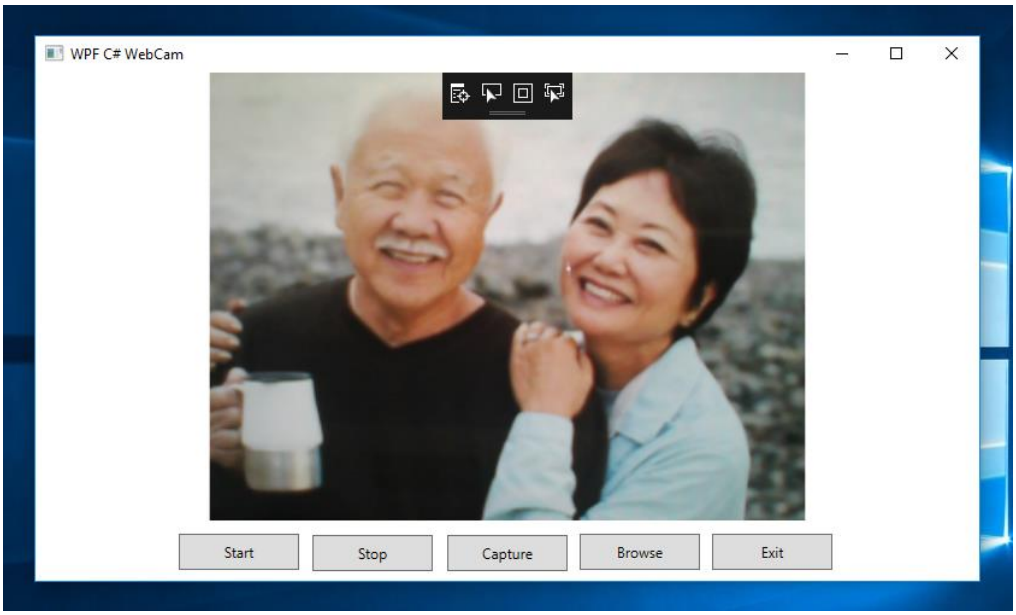
Before running the app, make sure you have updated the personGroupId in testing project same as the one used in Face project. Also make sure you change the filepath in the code to path of the file where you saved the captured image in your system. It detects all the faces present in the picture and tries to identify them.

Sample Output:

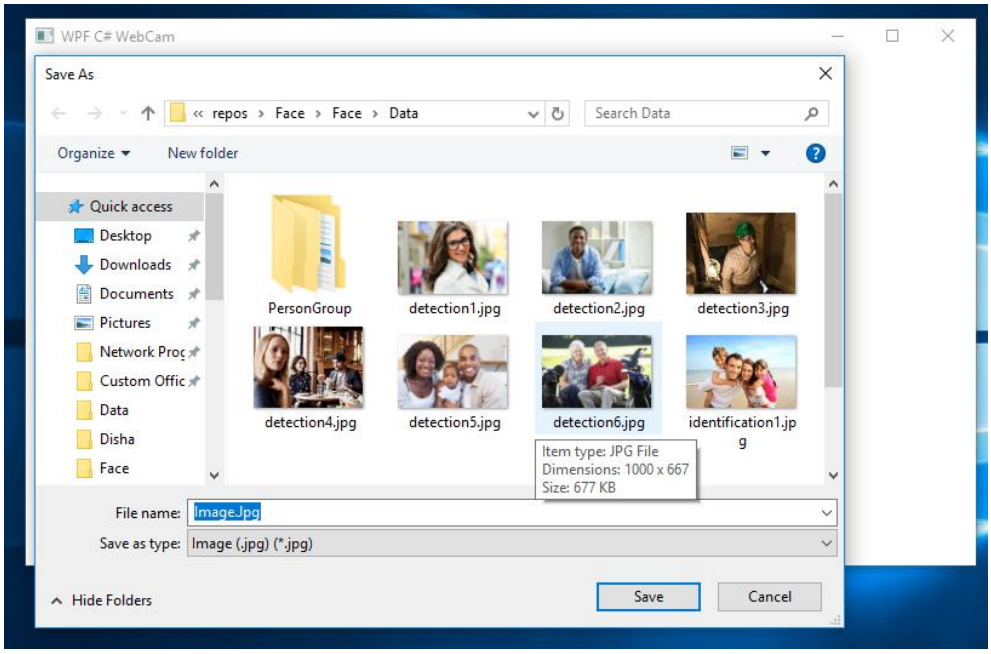
1. Run the application MainWindow.xaml.cs of Face project only once to train the data.
2. Start the application MainWindow.xaml.cs of Testing project.

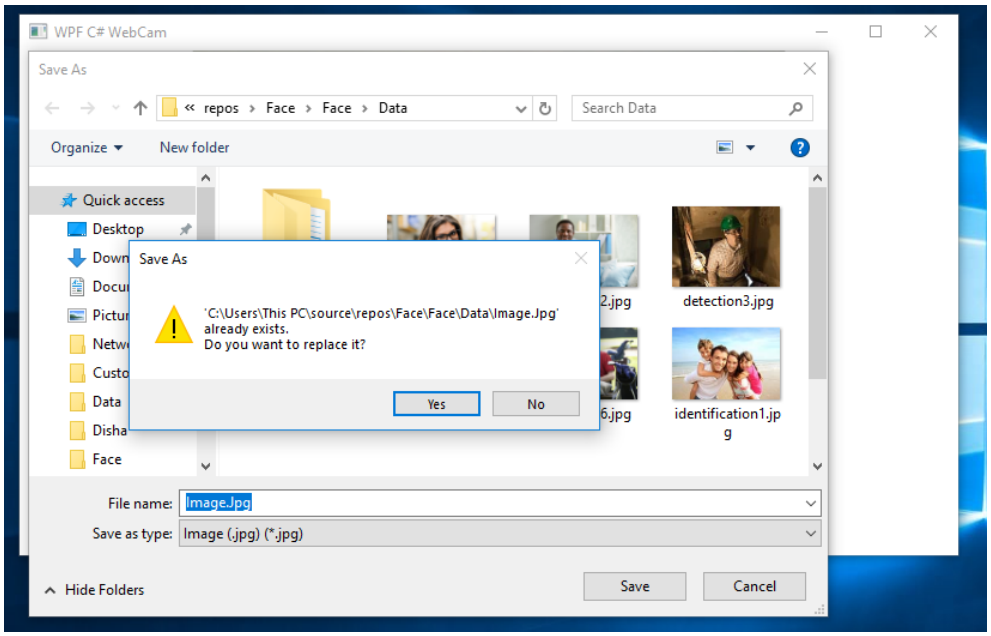


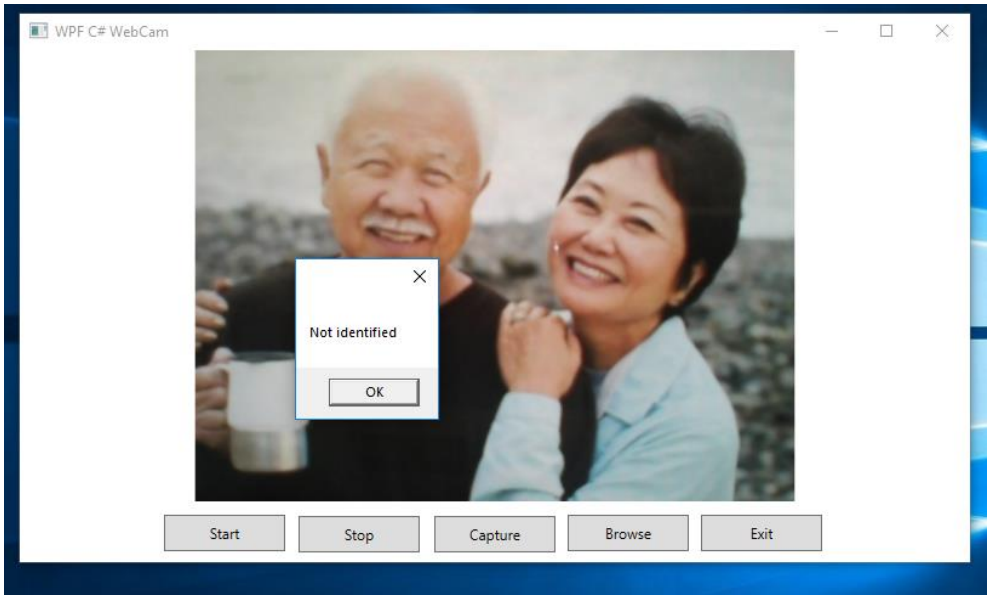
3. Start the webcam, adjust the camera and then stop.



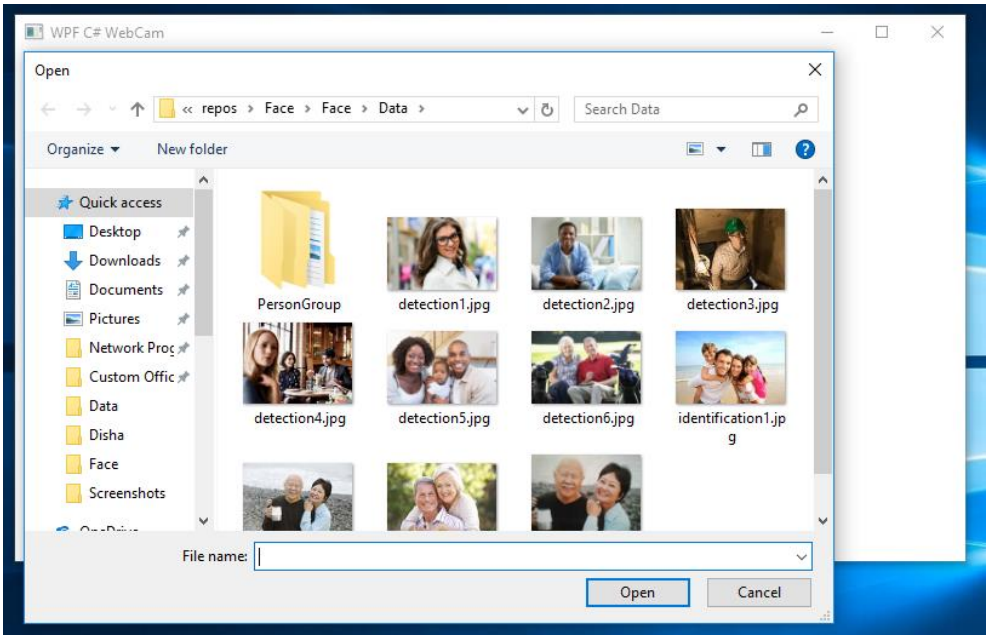
4. Capture the image and save with default name at default location. Overwrite the image if you capture again with same name.

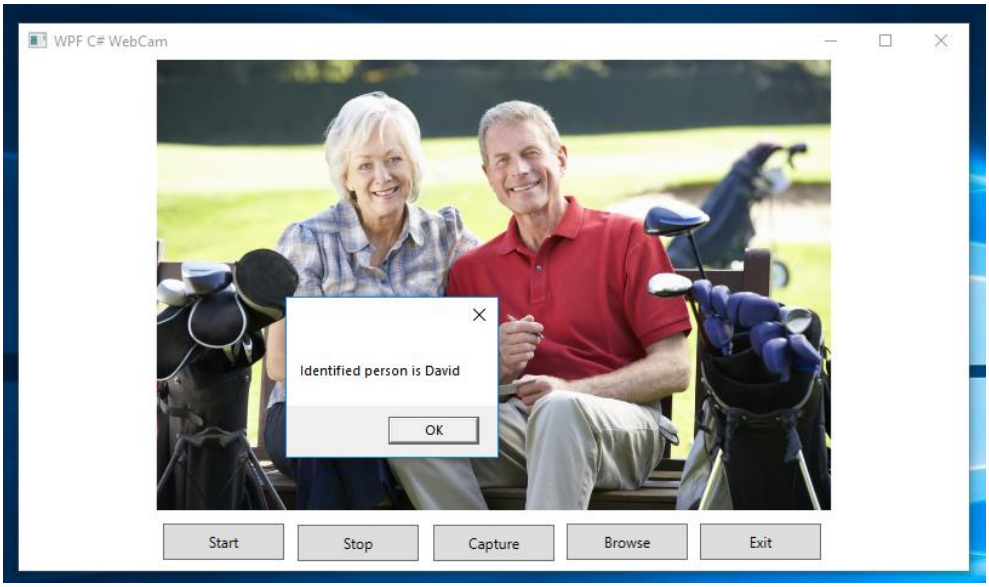


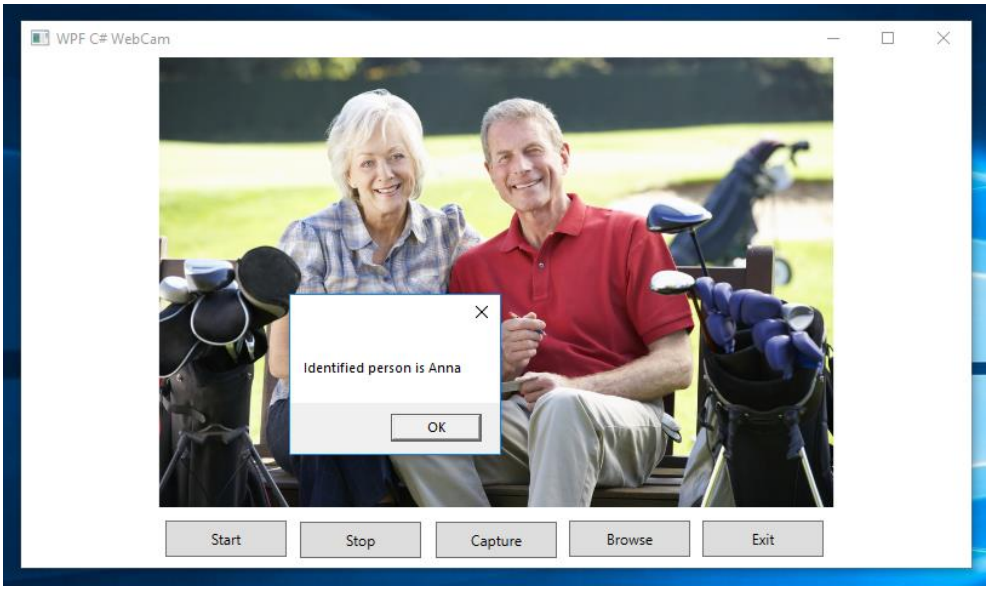




5. You can also browse and upload the image for identification.







6. Exit the application.