# CITS 5503 Cloud Computing

# Australia Voter Identification App Project

## Cover Letter

Huiying Hu

21742778

Semester 2, 2016

# Australian Voter Identification App

It is a secure online voting solution with facial recognition that verifies the voter's identity.

# Codes

The codes are submitted in github:

# Set Up

* This is a WPF application, written in C#. The required environment is Visual Studio 2015.
* When running this project in VS, please open the file VoterIdentificationApp\VoterIdentificationApp\VoterIdentificationApp.csproj
* One thing worth mentioning is that there are two references needed to be installed through NuGet Package Management if reference errors occur when building this project.

# Functional Requirements

1. In section 'Detect Faces', inputting an image and detecting faces from test image.
2. In section 'Define Person Group', creating a new person group.
3. In section 'Register Person', a person with a folder of registered face images is registered to one selected person group.
4. In section 'Train', a defined person group with a bunch of registered person will be trained, before stepping into next section 'Identify Voters'.
5. In section 'Identify Voters', inputting a test image and identifying voters.
6. Ideally voters could enrol into this system for voter identification.

# Non-Functional Requirements

1. This system performances fast enough as little downtime required.
2. This system reaches a high accuracy rate.
3. This system is easy to deploy in a cost effective manner.
4. This system has a significant capability on extensibility. It could be extended to become a high-quality voter election system based on API provided by Microsoft Project Oxford.

# The references of original tutorials

1. https://www.microsoft.com/cognitive-services/en-us/face-api/documentation/Get-Started-with-Face-API/GettingStartedwithFaceAPIinCSharp
2. https://www.microsoft.com/cognitive-services/en-us/face-api/documentation/face-api-how-to-topics/howtoidentifyfacesinimage
3. All APIs provided by Microsoft ProjectOxford are downloaded in 'ClientLibrary'. Reading through all APIs helps me build this project.

# Preparation

1. My subscription key is '58bcb00181d84c059c248d788f66fa2e'.
2. Open ‘VoterIdentificationApp.csproj’ in VS2015.
3. A personGroupId 'hhy' is created for preparation.

{

"personGroupId": "hhy",

"name": "group1",

"userData": "user-provided data attached to the person group"

}

1. In the personGroup with id 'hhy', a person with id 'cora' and three face images in 'VoterIdentificationApp\Data\PersonGroup\Family1-Daughter' is created beforehand.

{

"personId": "c1d8d468-5947-4de1-94bb-33fc3806204e",

"persistedFaceIds": [

"0386f563-b671-418f-a15b-f1e1bfee7294",

"69a4efb1-b11d-4b70-aa9c-0e47f62602eb",

"f314c93b-e98f-4e4e-b8b0-28fb466f5745"

],

"name": "cora",

"userData": null

}

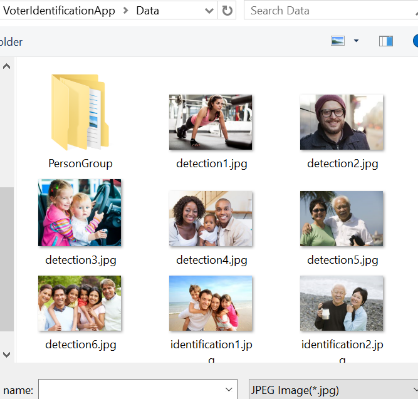
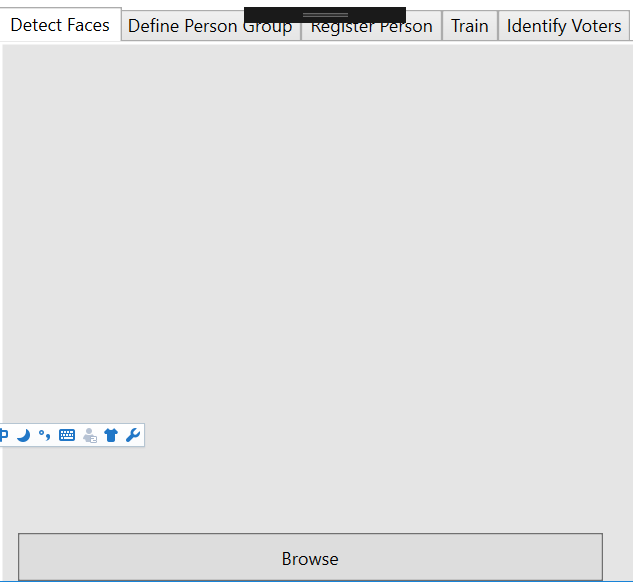
1. In section 'Identify Voters', select PersonGroupID as 'hhy', and select an image 'identification1.jpg' in 'Data' folder.
2. The registered voter 'cora' is identified.

### Notice

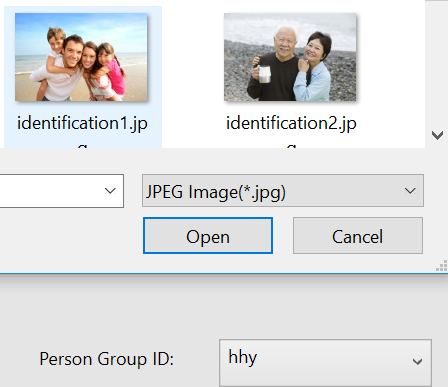
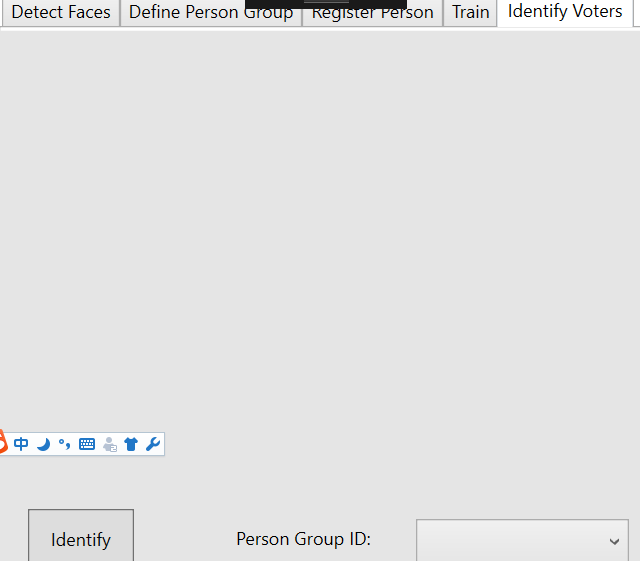
1. When inputting groupID, group name and person name in section 'Define Person Group' and 'Register Person', the input should be **lower case** and **no space**.

### Example I – Register a new voter ‘clare’ to a defined person group ‘hhy’

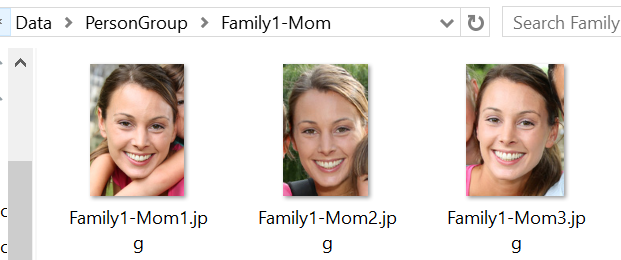
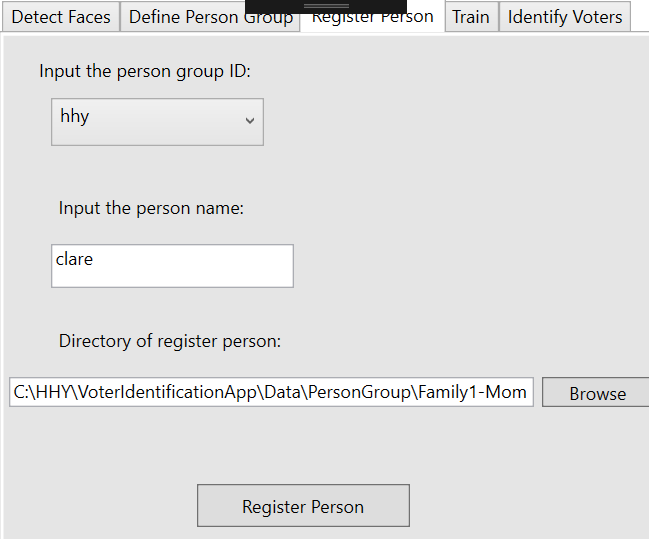
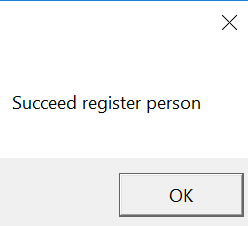
1. In section 'Detect Faces', inputting an image and detecting faces from test image.



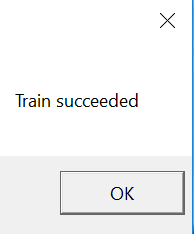
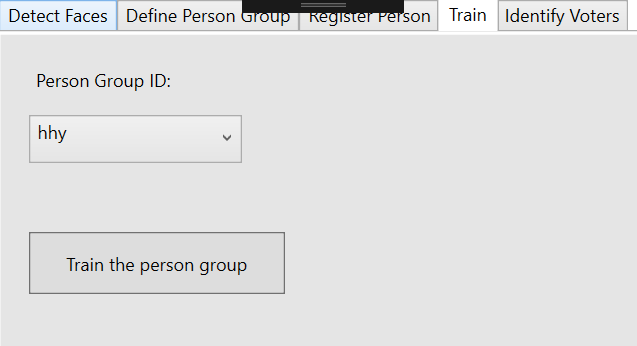
1. In section 'Identify Voters', inputting a test image and identifying voters. Only ‘cora’ is identified as a voter in the ‘hhy’ person group.



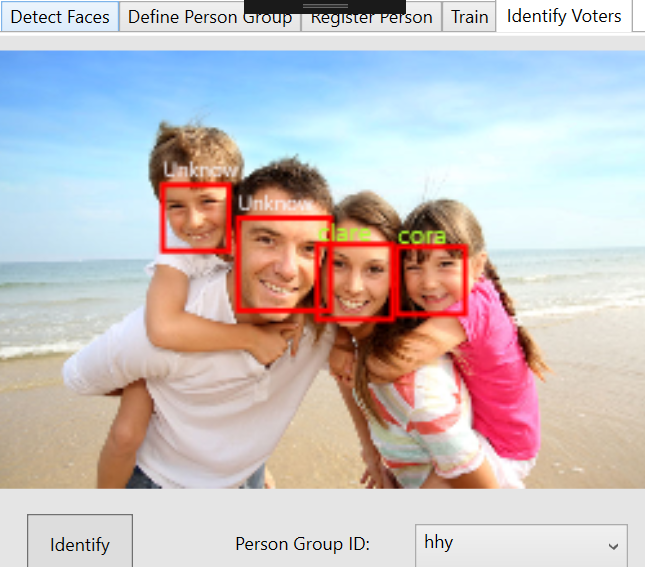
1. In section 'Register Person', a new voter ‘clare’ with a folder ‘Family1-Mom’ of registered face images is registered to one selected person group ‘hhy’.

1. In section 'Train', a defined person group ‘hhy’ with a bunch of registered person ‘cora’ and ‘clare’ will be trained, before stepping into next section 'Identify Voters'.

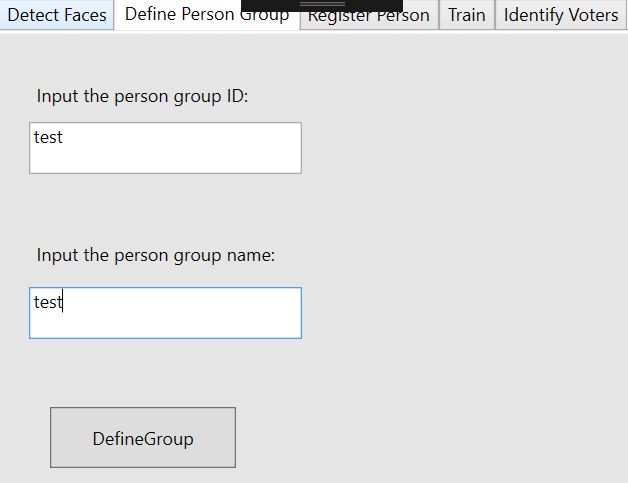
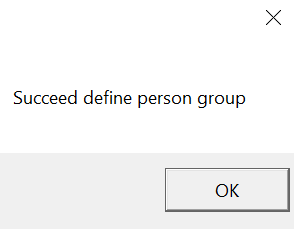


1. In section 'Identify Voters', inputting a test image and identifying voters. Now both ‘cora’ and ‘clare’ are identified as voters.



### Example I I– Define a new person group ‘test’ and register one voter ‘anna’ for this person group

1. In section 'Define Person Group', creating a new person group ‘test’.

1. Check the new person group has been created or not. Useful Face API provide:

<https://dev.projectoxford.ai/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395248/console>

It is ‘Person Group - List Person Groups’, which returns the list of person in this person group. Input ‘start’ with ‘0’. And input ‘Ocp-Apim-Subscription-Key’ with ‘your subscription key’. Output demonstrates that the new person group ‘test’ has been succeed created.

{

"personGroupId": "hhy",

"name": "group1",

"userData": "user-provided data attached to the person group"

},

{

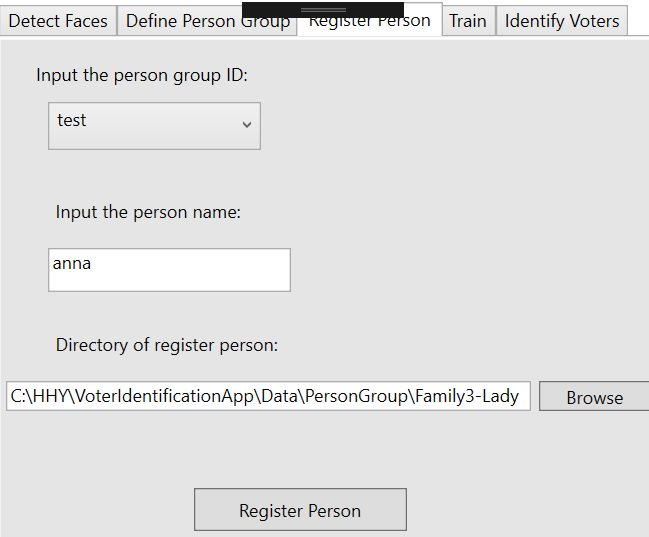
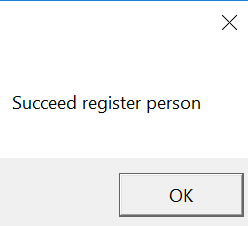
"personGroupId": "test",

"name": "test",

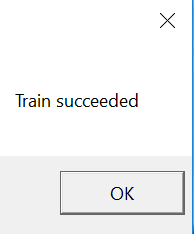
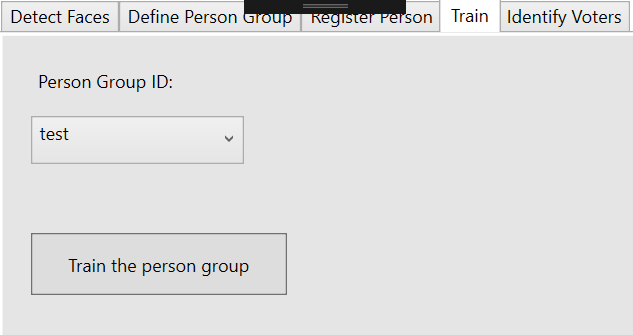
"userData": null

}

1. In section 'Register Person', a new voter ‘anna’ with a folder ‘Family3-Lady’ of registered face images is registered to one selected person group ‘test’.

1. In section 'Train', a defined person group ‘test’ with a bunch of registered person ‘anna’ will be trained, before stepping into next section 'Identify Voters'.



1. In section 'Identify Voters', inputting a test image and identifying voters. Now both ‘anna’ is identified as voters.

