Facial Identification with Kinect and the Face API

Gerald Spenlingwimmer, MSE - SEM 2 - MUS

Idea

A person shall be identified from a picture

The result shall be sent with Whatsapp as message

If the person has been recognized successfully the name is returned

Else a standard message like “unknown person” shall be sent

The result is also spoken by the computer

What the Kinect can do

Video and therefore Images

Depth channel, Color channel, …

With help of the SDK:

Face tracking, Body tracking, Gesture Basics, Face Basics, …

Face Data but no comparison

Face API

https://azure.microsoft.com/en-us/services/cognitive-services/face/

30.000 Requests per Month are Free

Get Face Data

Compare Faces on two seperate images

Identify People from Images

Lifecycle:

1) Train a network for facial identification
2) Deploy a network for facial identification
3) Identify people from images from 3-End of time

What the Face API provides

Gender

Age

Face Points/Coordinates

Hair Colour

Wears Glasses

Wears Makeup

Image Quality

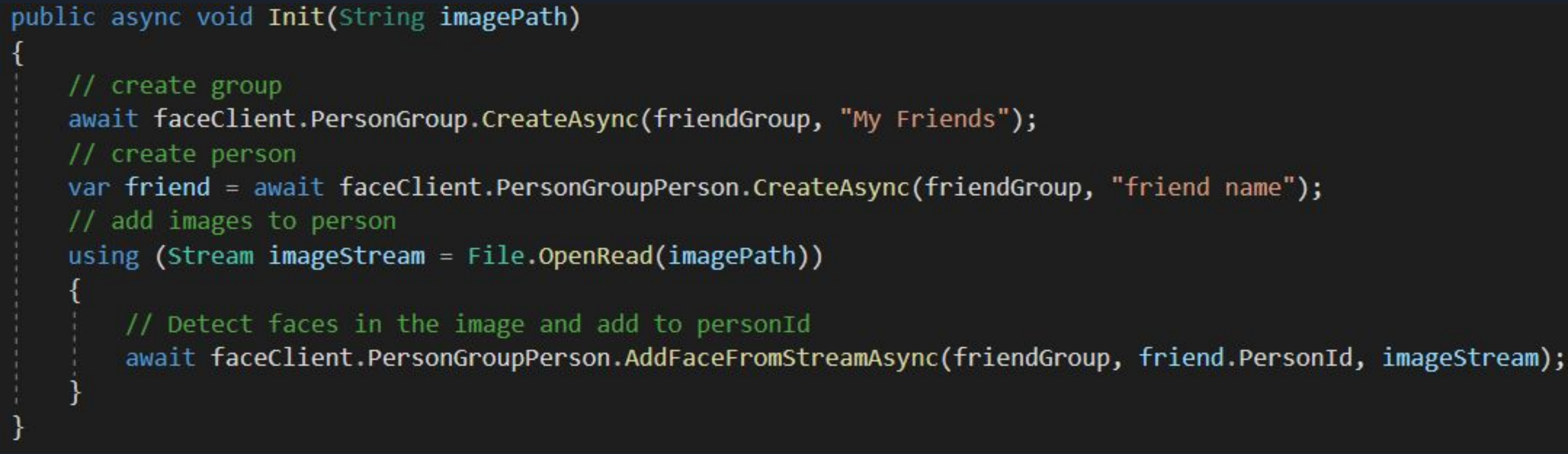
Emotion

Model Properties

PersonGroup … As the name implies a group of persons e.g. Family, Friends, Colleagues

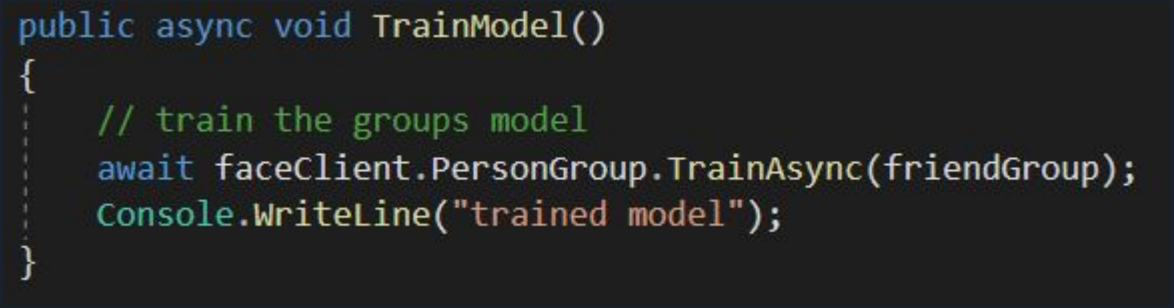
Person … A person in a PersonGroup where image data is stored to

Create the Model



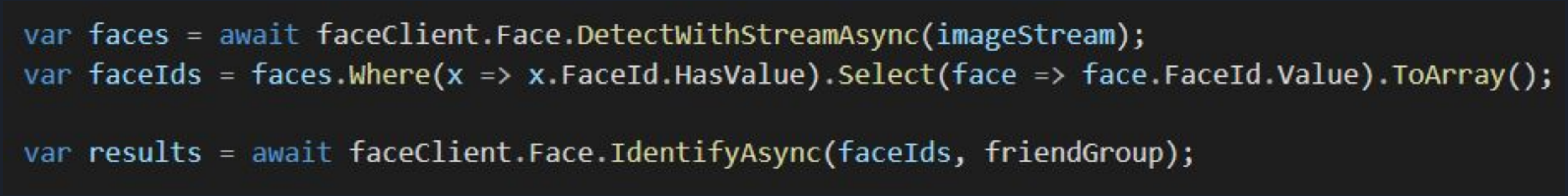
```
public async void Init(String imagePath)
{
    // create group
    await faceClient.PersonGroup.CreateAsync(friendGroup, "My Friends");
    // create person
    var friend = await faceClient.PersonGroupPerson.CreateAsync(friendGroup, "friend name");
    // add images to person
    using (Stream imageStream = File.OpenRead(imagePath))
    {
        // Detect faces in the image and add to personId
        await faceClient.PersonGroupPerson.AddFaceFromStreamAsync(friendGroup, friend.PersonId, imageStream);
    }
}
```

Train the Model



```
public async void TrainModel()
{
    // train the groups model
    await faceClient.PersonGroup.TrainAsync(friendGroup);
    Console.WriteLine("trained model");
}
```

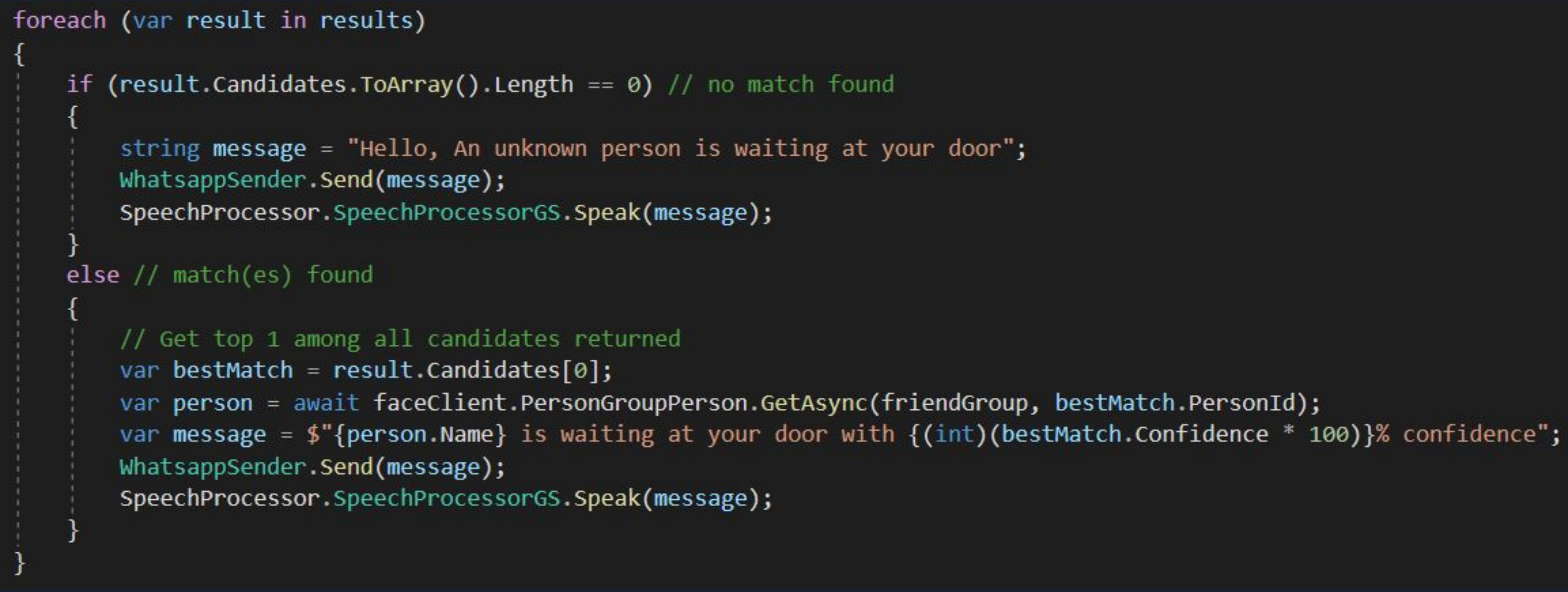
Identify a Person



```
var faces = await faceClient.Face.DetectWithStreamAsync(imageStream);
var faceIds = faces.Where(x => x.FaceId.HasValue).Select(face => face.FaceId.Value).ToArray();

var results = await faceClient.Face.IdentifyAsync(faceIds, friendGroup);
```

Identify a Person



```
foreach (var result in results)
{
    if (result.Candidates.ToArray().Length == 0) // no match found
    {
        string message = "Hello, An unknown person is waiting at your door";
        WhatsappSender.Send(message);
        SpeechProcessor.SpeechProcessorGS.Speak(message);
    }
    else // match(es) found
    {
        // Get top 1 among all candidates returned
        var bestMatch = result.Candidates[0];
        var person = await faceClient.PersonGroupPerson.GetAsync(friendGroup, bestMatch.PersonId);
        var message = $"{person.Name} is waiting at your door with {(int)(bestMatch.Confidence * 100)}% confidence";
        WhatsappSender.Send(message);
        SpeechProcessor.SpeechProcessorGS.Speak(message);
    }
}
```

Whatsapp Messages with Twilio

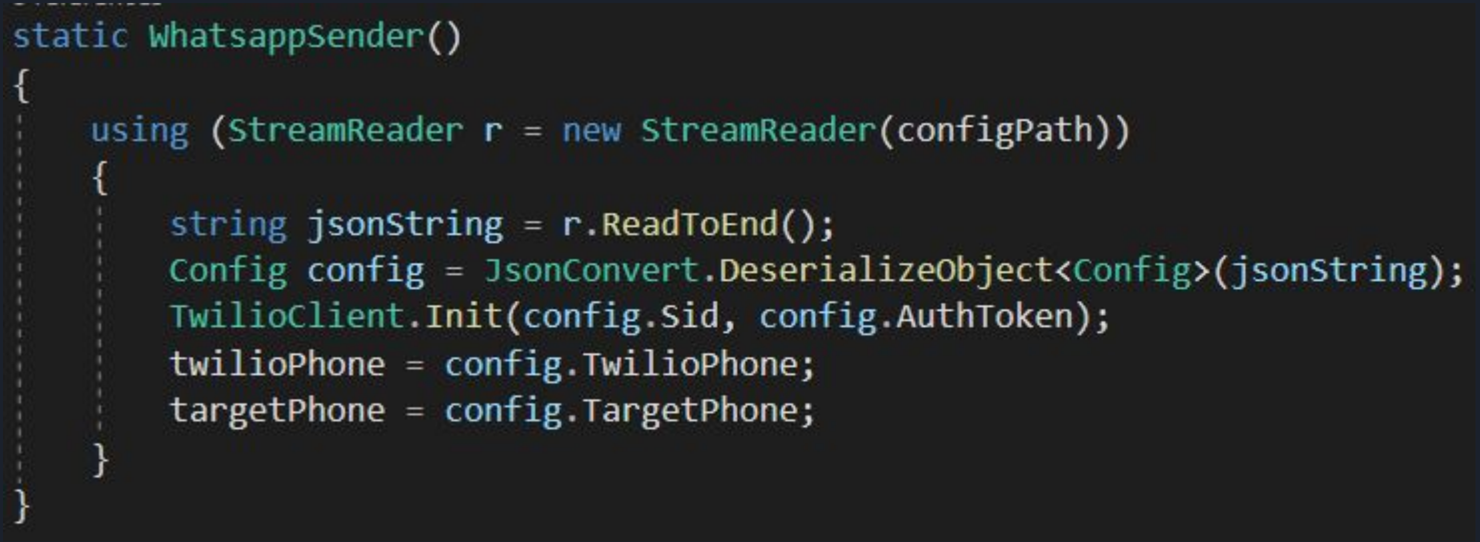
https://www.twilio.com/whatsapp

Lifecycle:

1) Register Client Phone to “Channel” via a Message
2) Send a Predefined Message Template to Client
3) Answer by Sending a Message from the Client
4) Now Custom Text messages can be exchange for 24 hours

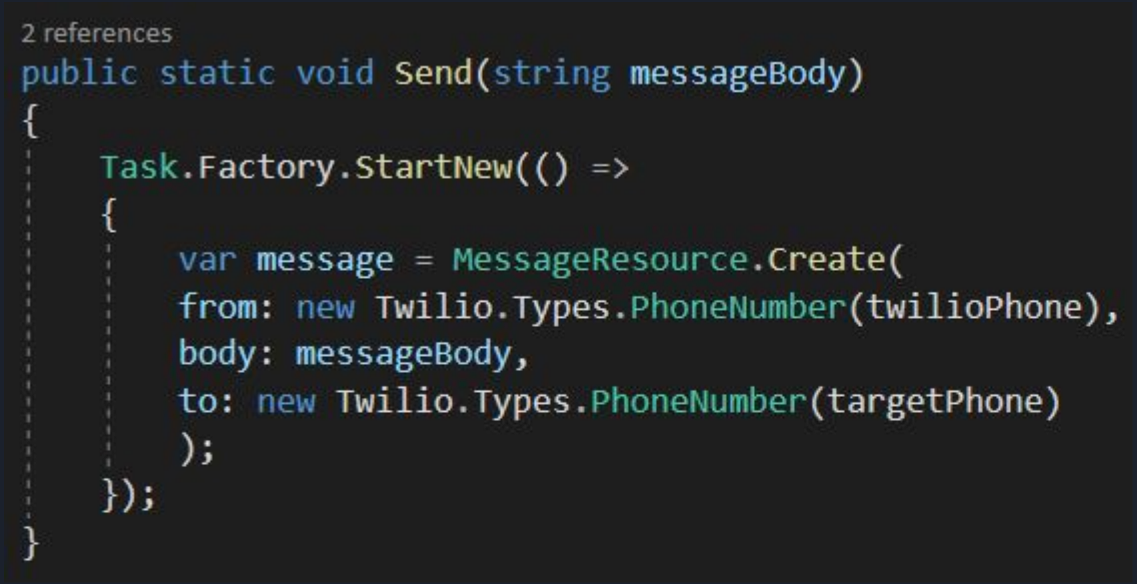
Prevents spamming

Twilio API - Init



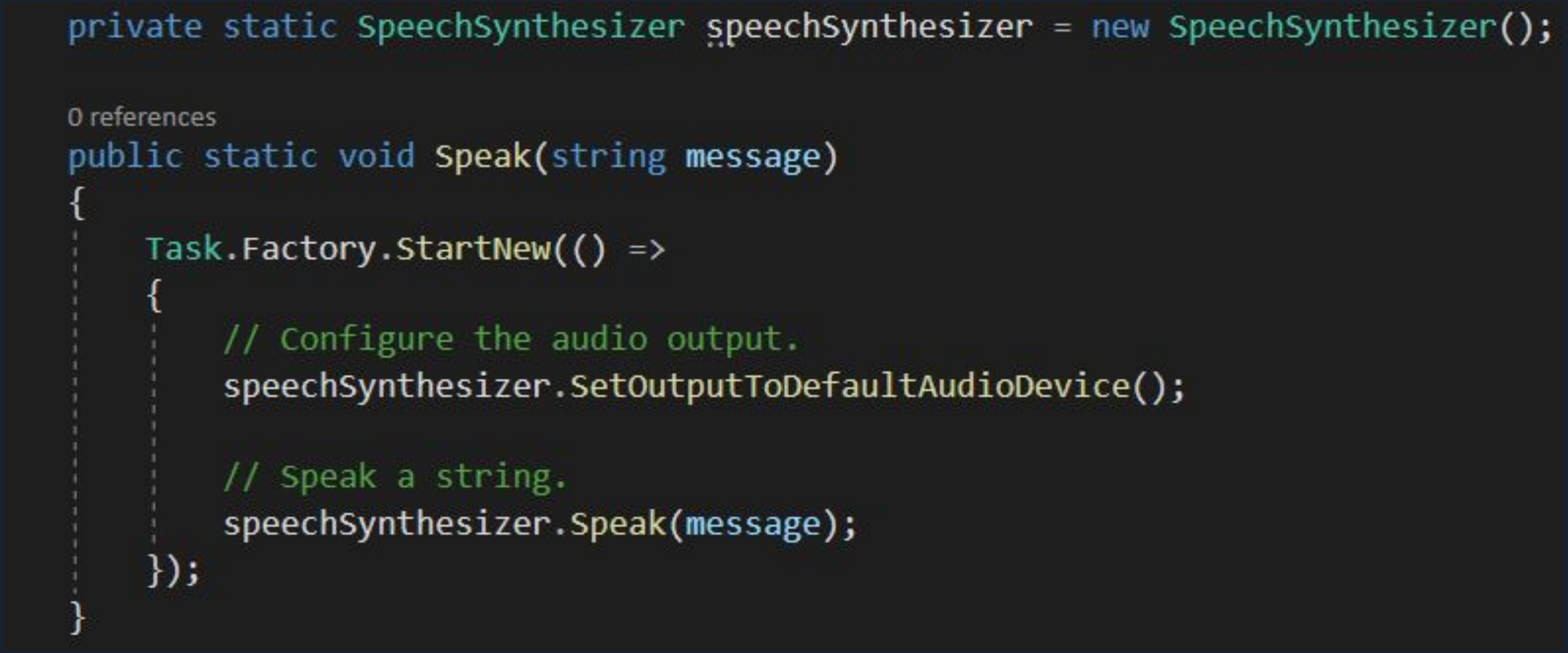
```
static WhatsappSender()
{
    using (StreamReader r = new StreamReader(configPath))
    {
        string jsonString = r.ReadToEnd();
        Config config = JsonConvert.DeserializeObject<Config>(jsonString);
        TwilioClient.Init(config.Sid, config.AuthToken);
        twilioPhone = config.TwilioPhone;
        targetPhone = config.TargetPhone;
    }
}
```

Twilio API



```
2 references
public static void Send(string messageBody)
{
    Task.Factory.StartNew(() =>
    {
        var message = MessageResource.Create(
        from: new Twilio.Types.PhoneNumber(twilioPhone),
        body: messageBody,
        to: new Twilio.Types.PhoneNumber(targetPhone)
        );
    });
}
```

Speech Output



```
private static SpeechSynthesizer speechSynthesizer = new SpeechSynthesizer();

0 references
public static void Speak(string message)
{
    Task.Factory.StartNew(() =>
    {
        // Configure the audio output.
        speechSynthesizer.SetOutputToDefaultAudioDevice();

        // Speak a string.
        speechSynthesizer.Speak(message);
    });
}
```

Findings

Easy to Use

Fast Development

Quite Long Duration from Start to End (15 - 40 sec) until the Whatsapp Message reaches the Phone

Result

Kinect Integrated

Face API Model Trainend

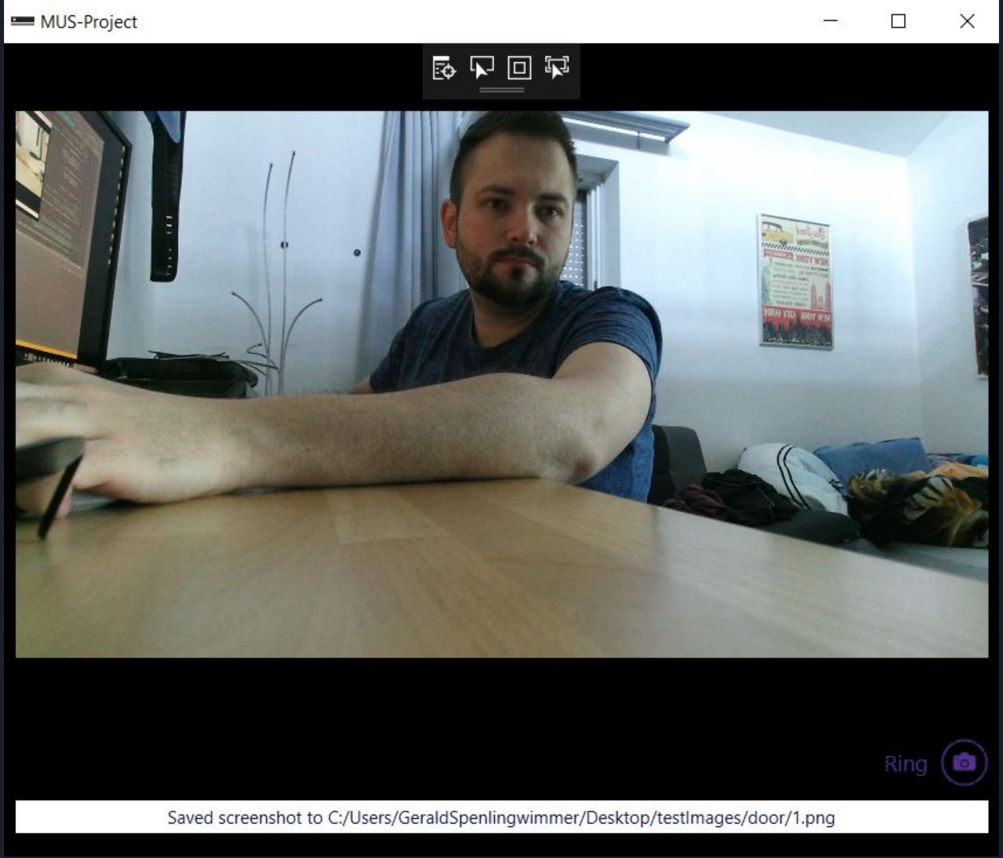
Face API Identification Works

WPF App Live Camera with Virtual Doorbell

Speech Output

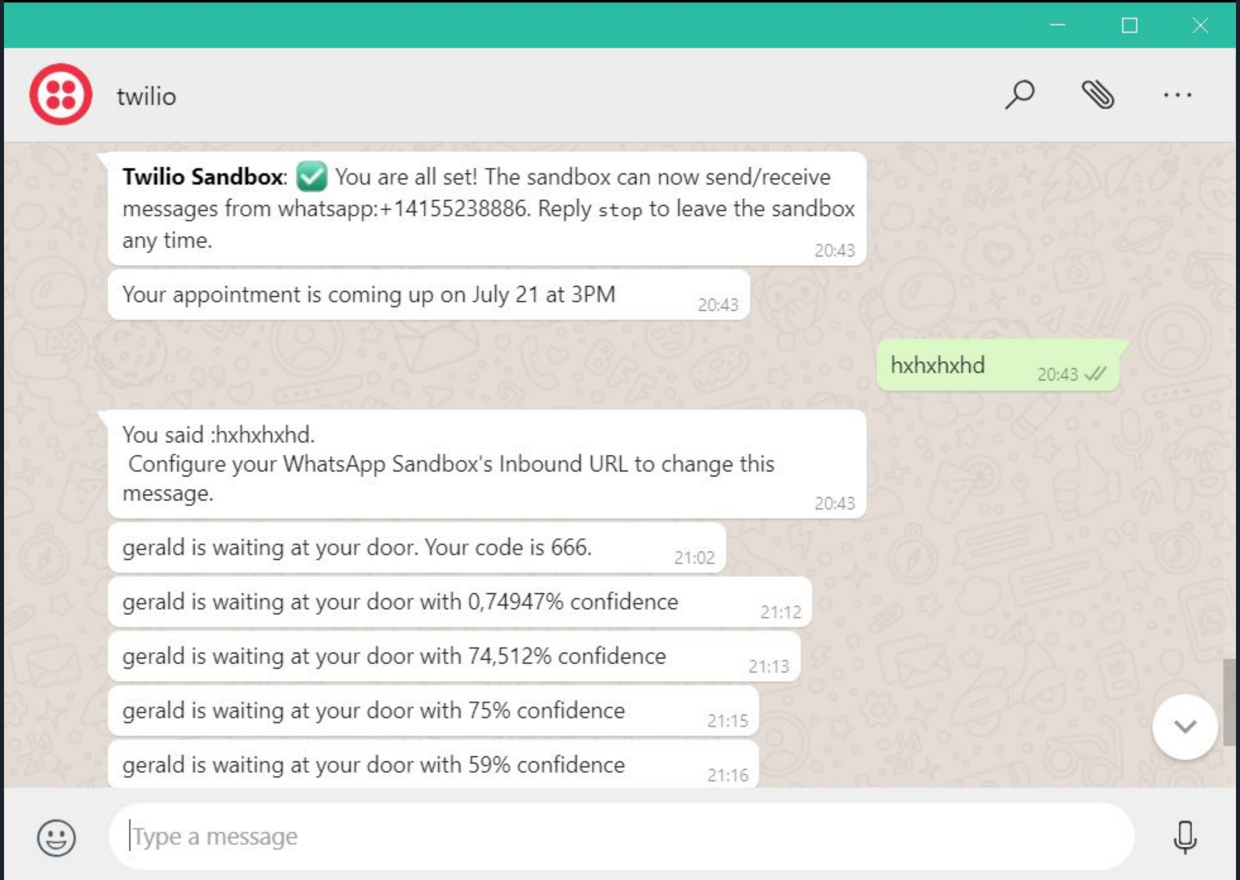
Whatsapp Messages

The “Doorbell”



Saved screenshot to C:/Users/GeraldSpenlingwimmer/Desktop/testImages/door/1.png

Whatsapp Messages



Twilio Sandbox: ✅ You are all set! The sandbox can now send/receive messages from whatsapp:+14155238886. Reply `stop` to leave the sandbox any time. 20:43

Your appointment is coming up on July 21 at 3PM 20:43

hxhxhxhd 20:43 ✓✓

You said :hxhxhxhd.
 Configure your WhatsApp Sandbox's Inbound URL to change this message. 20:43

gerald is waiting at your door. Your code is 666. 21:02

gerald is waiting at your door with 0,74947% confidence 21:12

gerald is waiting at your door with 74,512% confidence 21:13

gerald is waiting at your door with 75% confidence 21:15

gerald is waiting at your door with 59% confidence 21:16

The End?