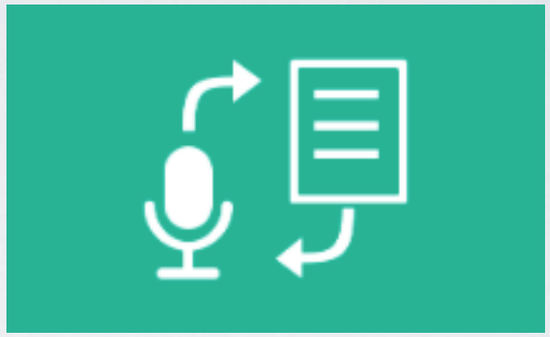
PROJECT OXFORD

Face & Speech API

Steven Dostert, Natalie Kurz, Benjamin Eder, Johannes Schuster

FACE & SPEECH API



Ziel: Entwicklung einer App, welche nach erfolgreicher Gesichtserkennung Emails per Sprachsteuerung an entsprechende Personen schickt.

FACE API

- API erkennt Gesichter und markiert diese
- Durch „Trainieren“ können Personen abgespeichert und wiedererkannt werden

FACE API

```
await faceServiceClient.CreatePersonGroupAsync(personGroupId, personGroupId);
await AutoDetectFaces(@"..\..\..\example_pictures\Anna", "s_d@uni.de");
await AutoDetectFaces(@"..\..\..\example_pictures\Bill", "natalie-kurz1@gmx.de");
await StartTraining();
BrowseButton.IsEnabled = true;
```

Personengruppe wird erzeugt

```
var friendFaceIds = friendFaces.Select(face => face.FaceId).ToArray();
var friend = await faceServiceClient.CreatePersonAsync(personGroupId, friendFaceIds, email);
```

Person wird erzeugt und der Gruppe hinzugefügt

```
await faceServiceClient.TrainPersonGroupAsync(personGroupId);
```

Personengruppe wird „trainiert“

FACE API

```
private async Task<Face[]> UploadAndDetectFaces(string imageFilePath) {
    try {
        using (Stream imageFileStream = File.OpenRead(imageFilePath)) {
            var faces = await faceServiceClient.DetectAsync(imageFileStream);
            return faces;
        }
    }
    catch (Exception) {
        Console.WriteLine("Error: Could not detect faces.");
        return new Face[0];
    }
}
```

Bildupload und Gesichterkennung ausgeführt

liefert Gesichtsmerkmale als Objekt zurück

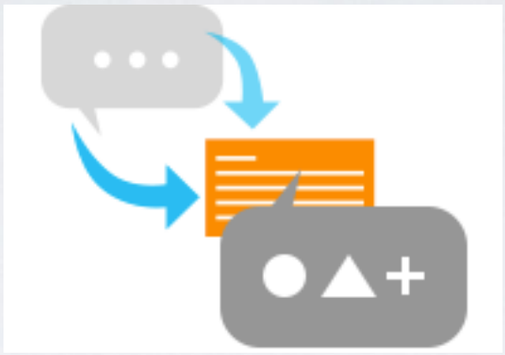
FACE API

```
private async void SearchSimilarFaces(Face[] faces) {
    var faceIds = faces.Select(face => face.FaceId).ToArray();
    var results = await faceServiceClient.IdentifyAsync(personGroupId, faceIds, 4);

    foreach (var identifyResult in results)
    {
        if (identifyResult.Candidates.Length != 0)
        {
            var candidateId = identifyResult.Candidates[0].PersonId;
            var person = await faceServiceClient.GetPersonAsync(personGroupId, candidateId);
            mailList.Add(person.Name);
            this.speechButton.IsEnabled = true;
        }
    }
    this.WriteLine("Found {0} similar faces. Speech API enabled!", mailList.Count);
}
```

Identifizierung zuvor erkannter Personen in Personengruppe

SPEECH API



- Ermöglicht Sprachsteuerung

- Durch Reguläre Ausdrücke können Audiosignale in Aktionen übersetzt werden

SPEECH API

```
m_micClient = SpeechRecognitionServiceFactory.CreateMicrophoneClient(m_recoMode, "de-de", m_primaryOrSecondaryKey);

// Event handlers for speech recognition results
m_micClient.OnResponseReceived += this.OnResponseReceivedHandler;
m_micClient.OnPartialResponseReceived += this.OnPartialResponseReceivedHandler;
m_micClient.OnConversationError += this.OnConversationErrorHandler;
m_micClient.OnMicrophoneStatus += this.OnMicrophoneStatus;
```

Initialisierung von MicrophoneClient und EventHandler

SPEECH API

```
string speechText = e.PhraseResponse.Results[0].DisplayText;
```

Spracheingabe als String

```
string regex1 = "(?i)(versende|sende).*?(email|e-mail).*?mein.*?gruppe";
```

Regular Expressions zur Befehlserkennung