Wydział Elektrotechniki Automatyki
Informatyki i Inżynierii Biomedycznej
Informatyka

Rok studiów: IV
Rok akademicki: 2020/21

Aplikacja do tworzenia zbiorów danych do rozpoznawania emocji

Studio projektowe 2

Przemysław Bielecki
Izabela Pachel
Ewa Tabor
Maciej Wilk

Spis treści

Spis rysunków

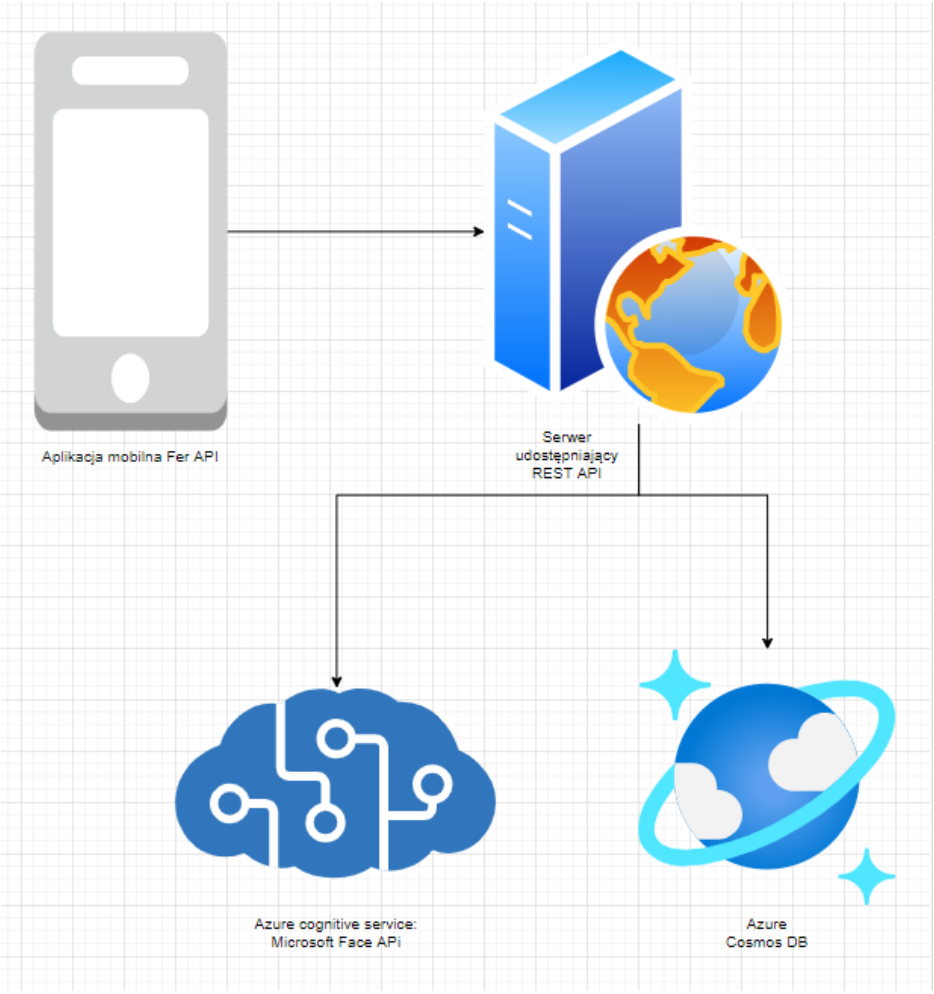
1 Opis systemu

Aplikacja „FERApp” służy do tworzenia zbiorów danych wykorzystywanych przy rozpoznawaniu emocji na podstawie mimiki twarzy (*facial emotion recognition*). W pierwszym kroku, użytkownikowi aplikacji mobilnej wyświetlana jest nazwa losowej emocji wraz z jej ikoną. Dostępne emocje to: **radość, smutek, strach, odraza, złość, zaskoczenie, neutralność**. Użytkownik robi zdjęcie twarzy wyrażającej podaną emocję, powtarza zdjęcie lub zmienia proponowaną emocję i wysyła zdjęcie na serwer, gdzie...

2 Architektura systemu

System składa się z dwóch kluczowych elementów (rys. 1):

- Aplikacja mobilna
- Serwer udostępniający REST API, który komunikuje się z:
  - Microsoft Face Api
  - Azure Cosmos DB



Rys. 1: Architektura systemu

3 Aplikacja mobilna

Aplikacja mobilna została napisana w języku Java na mobilny system operacyjny Android, minimalna wersja systemu to 5.0 (Lollipop). Te technologie zostały wybrane ze względu na posiadane przez autorów urządzenia z tym systemem, co zdecydowanie ułatwiło tworzenie oraz testowanie aplikacji. System Android posiada również szeroką bazę dokumentacji i tutoriali oraz ogromną społeczność developerów, co niewątpliwie usprawnia pracę, a oprogramowanie potrzebne do tworzenia aplikacji, Android Studio, jest darmowe.

3.1 GUI

Interfejs graficzny aplikacji mobilnej składa się z jednego ekranu, który może znajdować się w dwóch stanach:

- przed wykonaniem zdjęcia,
- po wykonaniu zdjęcia.

W skład responsywnego layoutu aplikacji mobilnej wchodzą kontrolki:

- etykieta emocji, która ma zostać pokazana na zdjęciu,
- ikona emoji emocji, która ma zostać pokazana na zdjęciu (źródło: `https://www.pngegg.com`),
- przycisk z ikoną aparatu, który otwiera wbudowaną aplikację kamery,
- widok zdjęcia, które zostało wykonane i zaakceptowane po uruchomieniu aplikacji kamery,
- przycisk „TRY AGAIN”, który umożliwia ponowne wykonanie zdjęcia aktualnej emocji,
- przycisk „CHANGE EMOTION”, który zmienia emocję, która ma zostać pokazana oraz stan aplikacji,
- przycisk „SEND RESULT”, który wysyła dane na serwer.

3.2 Backend

Do zakodowania obrazu z użyciem kodowania Base64 (które służy do kodowania ciągu bajtów za pomocą ciągu znaków) użyto biblioteki Apache Commons Codec.

Do składania danych w format JSON użyto biblioteki JSON-Java (org.json).

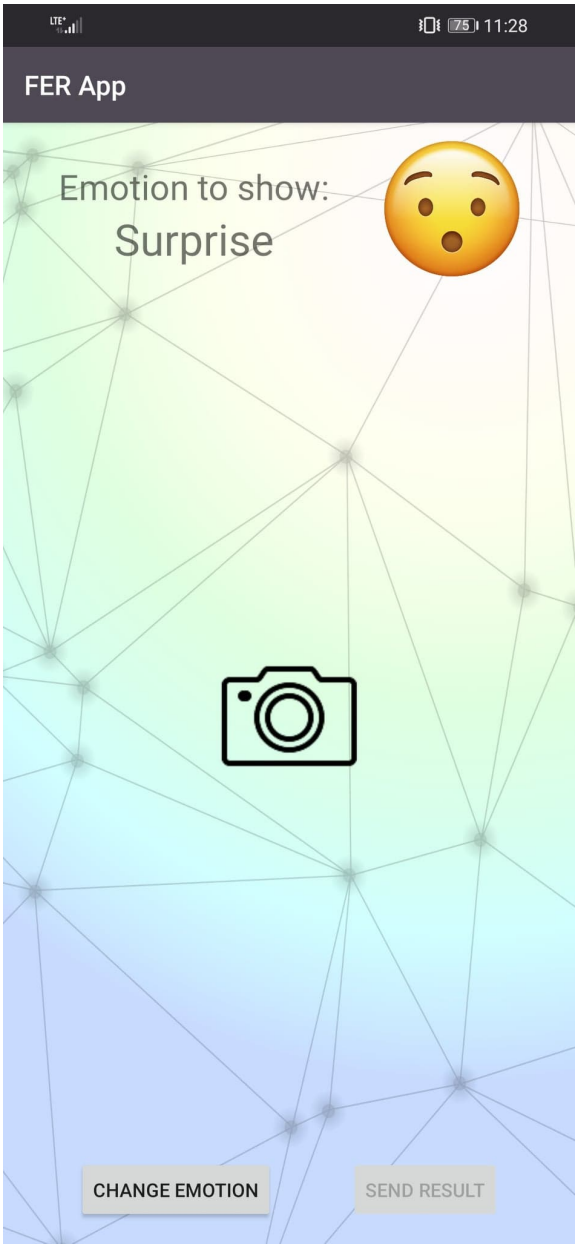
Do wykonywania i budowania żądań HTTP użyto biblioteki Volley, która specjalizuje się w obsłudze sieci. Żądania są dodawane do kolejki i wykonywane asynchronicznie, dzięki czemu narzut czasowy, który wynika z łączenia z siecią, nie wpływa negatywnie na doświadczenia użytkownika i nie blokuje aplikacji.

Aplikacja wymaga uprawnień do uruchamiania kamery, dostępu do galerii oraz dostępu do połączenia internetowego.

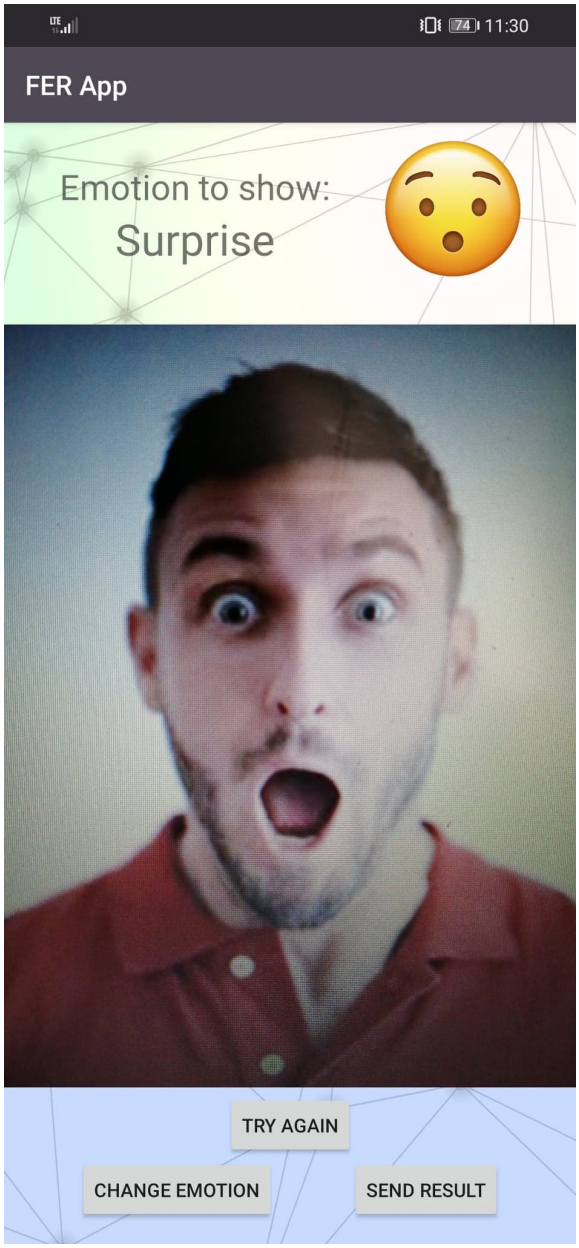
4 Serwer

Serwer został napisany w technologii .net core. Dobór technologii został uzasadniony wybranym środowiskiem hostującym rozwiązanie w chmurze: Microsoft Azure. Dodatkowo używana jest baza danych zapewniana przez Microsoft: CosmosDB.

Integracja z powyższymi rozwiązaniami jest oczywiście możliwa w innych technologiach, natomiast ze względu na intuicyjność rozwiązań dotnetowych, jak i dostępność materiałów edukacyjnych przygotowanych przez Microsoft skierowanych głównie na .net, to wybór tej technologii wydawał się najsensowniejszy.

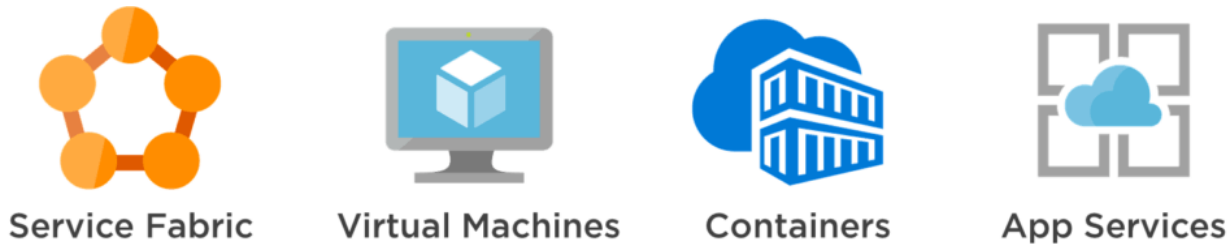


(a) Przed wykonaniem zdjęcia



(b) Po wykonaniu zdjęcia

Rys. 2: Ekran aplikacji mobilnej



Rys. 3: Usługi hostingowe
`https://stackify.com/azure-deployment-models`

4.1 Hosting

Azure portal zapewnia wiele możliwości hostowania aplikacji:

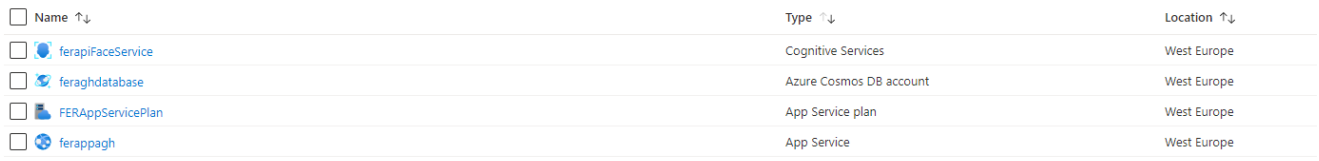
Każda z nich ma swoje wady i zalety, które należy uwzględnić wyborze. Jako że serwis udostępniany dla aplikacji mobilnej to pojedynczy http POST endpoint, to wybór Service Fabrica oraz stawiania maszyny wirtualnej był opcją zbyt kosztowną na potrzeby jakie miałaby zapewniać.

Pomiędzy integracją z kontenerami, a app services zdecydowała wygoda oraz pricing rozwiązania drugiego: przy zakładanym ruchy koszty miesięczne użycia nie powinny wykraczać poza kilka/kilkanaście centów.

4.2 Użyte zasoby na potrzeby obsługi serwera

Użyte zasoby Azure portal to:

- Resource Group
- App Service
- App Service plan
- CosmosDB
- Cognitive Services



Rys. 4: Zasoby wewnątrz FERGroup: opracowanie własne

Z powyższego listingu wynika więc, że wszystkie zasoby użyte na potrzeby serwera zostały zawarte wewnątrz jednej grupy zasobów Azure portal, ułatwiając ich zarządzaniem.

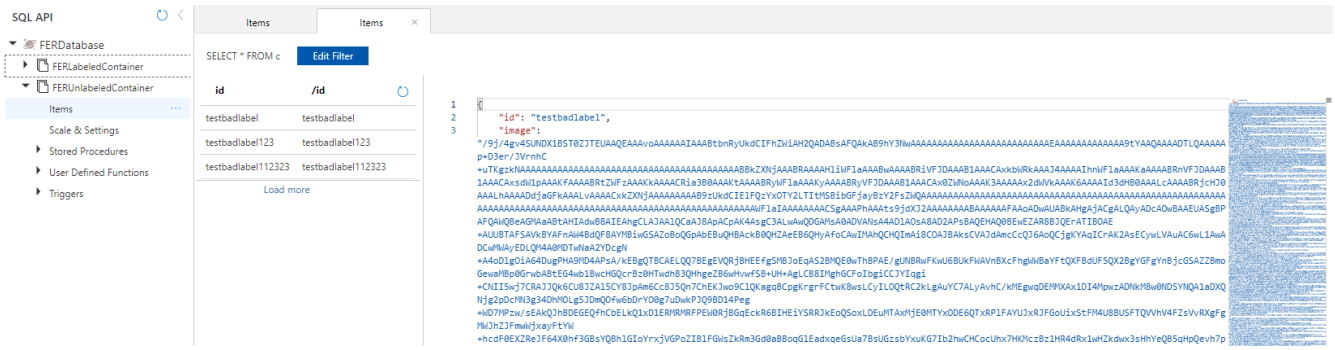
4.2.1 Baza Danych CosmosDB

Dane przechowywane są w bazie CosmosDB, która podzielona została na dwa kontenery.

Do kontenera labeled trafiają te zdjęcia, dla których Microsoft Face API przyporządkowało dominującą emocje z wartością większą niż 0.7 (powyższe api zwraca zestaw wszystkich emocji ze stanem od 0 do 1 suma stanu ze wszystkich emocji równa 1).

Pozostałe trafiają do kontenera unlabeled, w przypadku gdy na zdjęciu została rozpozna twarz, jeśli tego nie było zdjęcie zostaje odrzucone.

Zdjęcia są przechowywane w formacji Base64.



Rys. 5: Baza danych CosmosDB: opracowanie własne

5 Interfejsy

5.1 Aplikacja mobilna – serwer

Serwer wystawia REST API, a aplikacja mobilna wysyła dane w formacie JSON na odpowiedni endpoint.

- `https://ferappagh.azurewebsites.net/api/FER`
  Opis: Wysyłanie zdjęcia na serwer
  Metoda HTTP: POST
  Content-Type: application/json
  Request:

```
{
    "id"        : "IMG_20201110_192239",
    "image"     : "VBORw0KGgoAA...RBVHhe7J0FuFSF",
    "emotionID" :  5
}
```

  Response:

  - 200 Success
  - 400 Bad Request

Nazwa pola	Typ danych	Opis
id	string	Unikalny identyfikator zdjęcia
image	string	Zdjęcie zakodowane w Base64
emotionID	number	Identyfikator emocji (1-7)

Tab. 1: Pola wysyłanego JSONa

5.2 Serwer – Face API

Serwer komunikuje się z Microsoft Face API poprzez zdefiniowany przez nie endpoint Face Detect, który poprzez sparametryzowane zapytania w odpowiedzi zwrotnej przesyła nazwę i rozpoznanie każdej z wylistowanych emocji.

Rys. 6: Przykładowe zapytanie Face API
`https://azure.microsoft.com/en-us/services/cognitive-services/face/#demo`

Niestety Microsoft nie udostępnia informacji jakie modele użyte zostały podczas procesowania zdjęć i rozpoznania na nich emocji, ale bazując na własnej wiedzy zakładamy, że rozpoznanie emocji bazuje na zastosowaniu konwolucyjnych sieci neuronowych.

Dokumentację do Face API można znaleźć pod tym adresem: `https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395236`

6 Kod aplikacji

Całość kodu można znaleźć w repozytorium:

`https://github.com/notnamenot/FERApp`