

Projekt: Uwierzytelnianie użytkownika poprzez rozpoznanie twarzy

Techniki Multimedialne - Adam Kasperowicz 279046

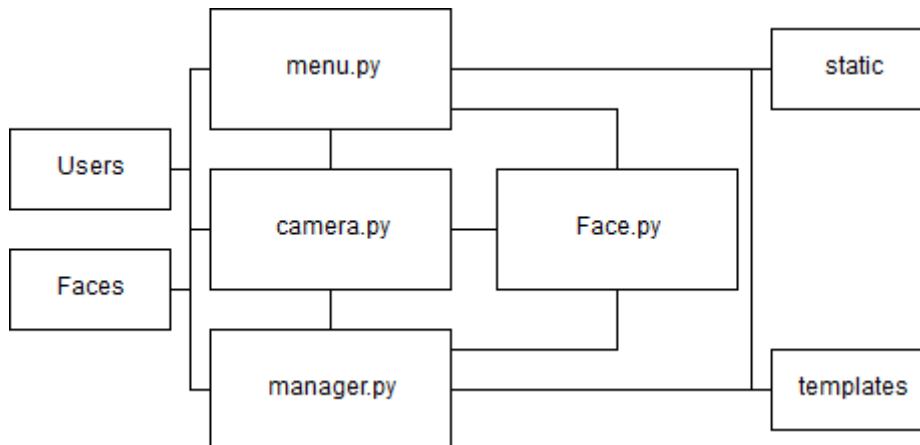
Opis

Program pozwala na utworzenie, zapis i późniejszy odczyt notatek danemu zalogowanemu użytkownikowi. Logowanie nie wykorzystuje hasła. Osoba logująca się podaje swój login. Następnie, kamera internetowa dokonuje zdjęcia twarzy. Twarz zostaje poddana procesowi porównania z twarzą zapisaną w bazie danych podczas rejestracji, jeśli podobieństwo jest znaczące, użytkownik zostaje zalogowany.

Wykorzystane technologie

- GUI – htmlPy
- Obsługa kamery – OpenCV
- Algorytmy do analizy zdjęcia – dlib (Python API)
- Pozostałe biblioteki – numPy

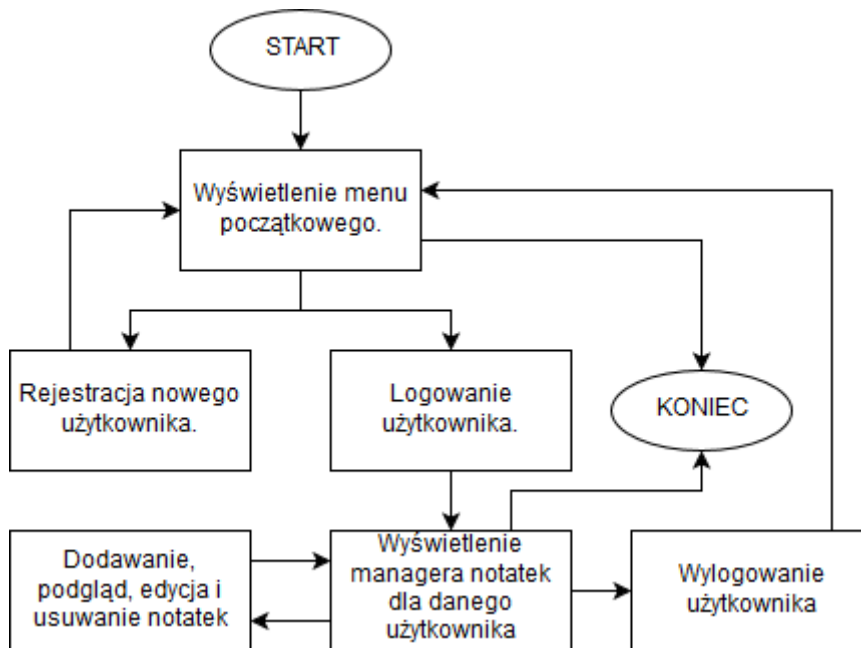
Struktura programu



- **Face.py** – odpowiada za włączenie aplikacji i załączenie odpowiednich bibliotek.
- **camera.py** – obsługuje kamerę internetową oraz dokonuje analizy twarzy użytkownika.
- **manager.py**, **menu.py** – odpowiednio służą do obsługi wydarzeń wywołanych przez użytkownika dla widoków managera notatek i menu logowania.
- **users** – folder zawierający podfoldery dla każdego użytkownika, dany podfolder zawiera notatki utworzone przez użytkownika

- **faces** – folder zawierający zakodowane twarze zarejestrowanych użytkowników. Każdy kod twarzy przechowywany jest w oddzielnym pliku, którego nazwą jest login użytkownika.
- **static, templates** – foldery zawierające pliki html, css i js wykorzystywane przez bibliotekę htmlPy do wyświetlenia GUI.

Działanie programu



Algorytm analizy zdjęcia

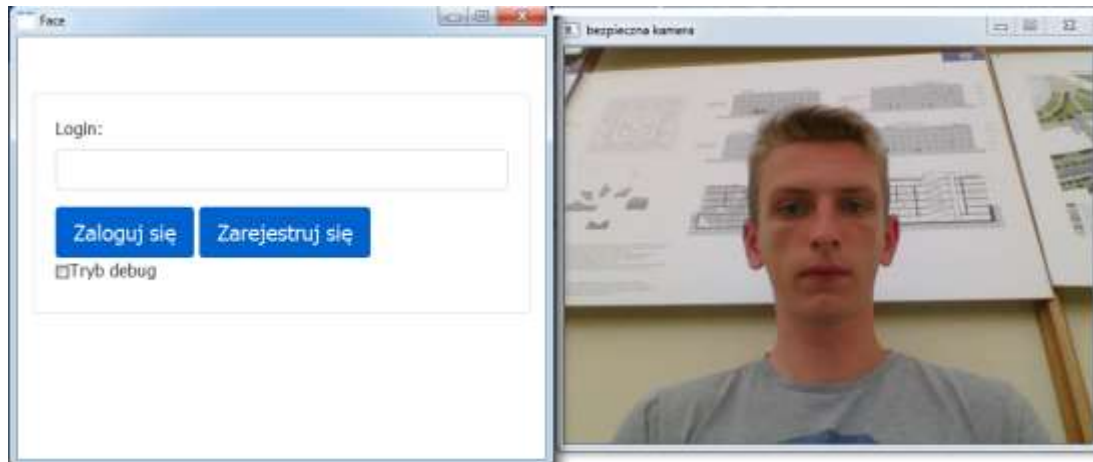
Wykorzystane zostają przetrenowane sieci neuronowe dostarczane przez bibliotekę dlib. Pierwsza sieć wykrywa twarz w obrazie. Druga sieć wykrywa punkty charakterystyczne twarzy. Trzecia sieć wykorzystuje obszar zdjęcia, w którym znajduje się twarz oraz punkty charakterystyczne twarzy do wyliczenia wektora o długości 128 elementów. Wektor ten jest zdefiniowany w przestrzeni, która pozwala mu jednoznacznie określić daną twarz. Do wyliczenia podobieństwa między dwoma twarzami wyliczana jest odległość euklidesowa dwóch wektorów.

Tryb debugowania

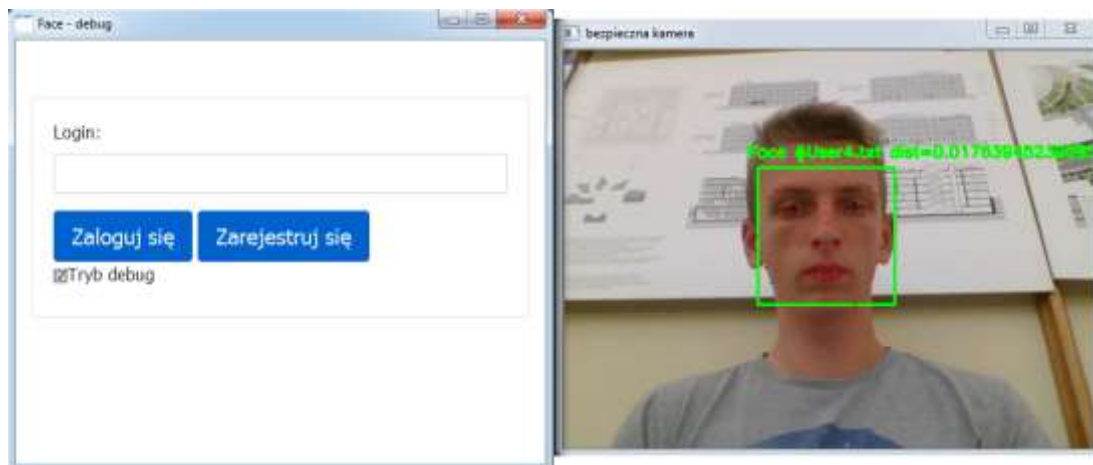
W menu logowania możliwe jest odznaczenie kontrolki „Tryb debug”. Włączenie tego trybu powoduje pokazanie na obrazie sczytanej z kamery informacji o tym jakie twarze komputer widzi oraz do jakich zarejestrowanych użytkowników te twarze są najbardziej podobne.

Interfejs użytkownika

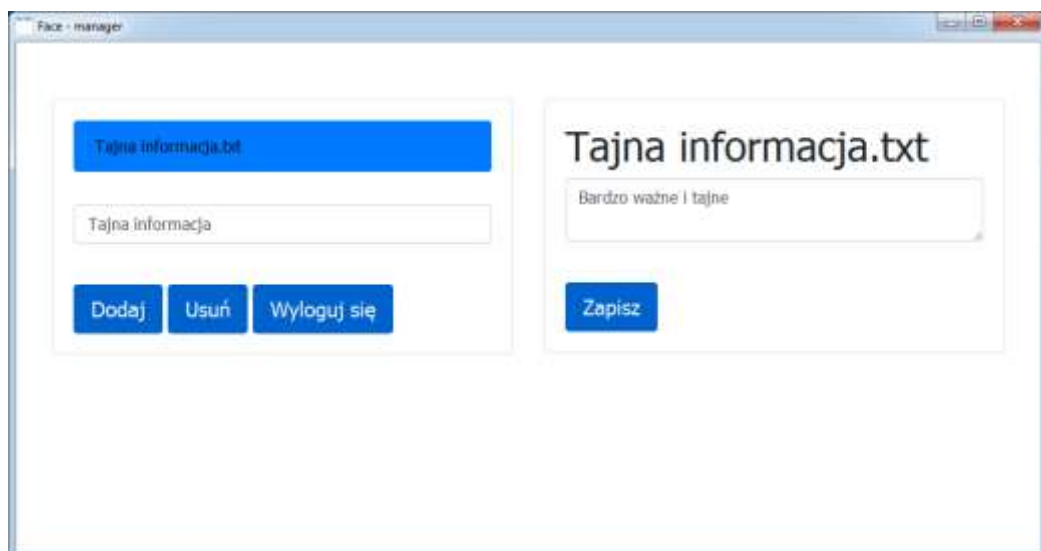
Menu logowanie



Menu logowanie z trybem debug



Manager notatek



Wnioski

W trakcie projektu zauważyć można było następujące zależności:

1. Wykrywanie twarzy, punktów charakterystycznych twarzy oraz analiza tych obiektów jest trywialna dzięki gotowym bibliotekom. Możliwe jest szybkie stworzenie dobrze działającego programu wykorzystującego najnowsze osiągnięcia w danej dziedzinie nauki przy pomocy małej ilości kodu.
2. Dużym kosztem obliczeniowym obciążone są działania wykrywania twarzy oraz analizy tej twarzy. Prosta implementacja wykorzystana w tym projekcie mogłaby nie spełniać wymagań normalnego użytkownika ze względu na jej powolność. Możliwe jest jednak łatwe zaradzenie temu poprzez np.: zmniejszenie rozdzielczości wykorzystywanego zdjęcia, użycie skali szarości zamiast palety RGB, zrównoleglenie obliczeń za pomocą takich narzędzi jak CUDA. Działania te pozwoliłyby uzyskać komfortową szybkość działania przy bardzo niskiej lub żadnej utracie jakości działania.
3. Sieci neuronowe dostarczane przez bibliotekę dlib nie są bezbłędne. Zauważyć można, że czasami nawet skrajnie niepodobne twarzy uznane zostają przez komputer za te same, zaś inny kąt padania światła może zmienić osobowość tej samej osoby. Łatwym sposobem rozwiązania tego problemu może być wielokrotne dokonanie pomiaru twarzy a następnie wybranie najczęściej występującego wyniku. Rozwiązanie te niestety zwiększa długość obliczeń.