

Karol Czub  
Adam Dębaczak

Informatyka Semestr VI  
Podstawy Teleinformatyki

Rozpoznawanie twarzy i śledzenie ruchu

# Spis treści

1. Opis ogólny aplikacji.....	
1.1. Menu główne aplikacji.....	
1.2. Identyfikacja osób.....	
1.3. Dodawanie nowej osoby.....	
2. Uzasadnienie wyboru tematu.....	
3. Podział prac	
4. Funkcjonalności oferowane przez aplikację.	
4.1. Identyfikacja osób.	
4.2. Dodawanie nowej osoby.	
4.3. Wyświetlanie listy rozpoznawanych osób.	
4.4. Usuwanie rozpoznawanej osoby z bazy.	
4.5. Ustanowienie połączenia z kamerą przy użyciu IP.	
4.6. Wyświetlanie instrukcji dla użytkownika.	
4.7. Wyświetlanie komunikatów o błędach.	
5. Użyte technologie.	
5.1) OpenCV.	
5.2) Numpy.	
5.3) Tensorflow.	
5.4) JSON.	
5.5) Tkinter.	
6. Architektura rozwiązania.	
7. Problemy podczas implementacji aplikacji oraz testy.	
8. Instrukcja użytkownika.	
9. Bibliografia.	

## 1. Opis ogólny aplikacji.

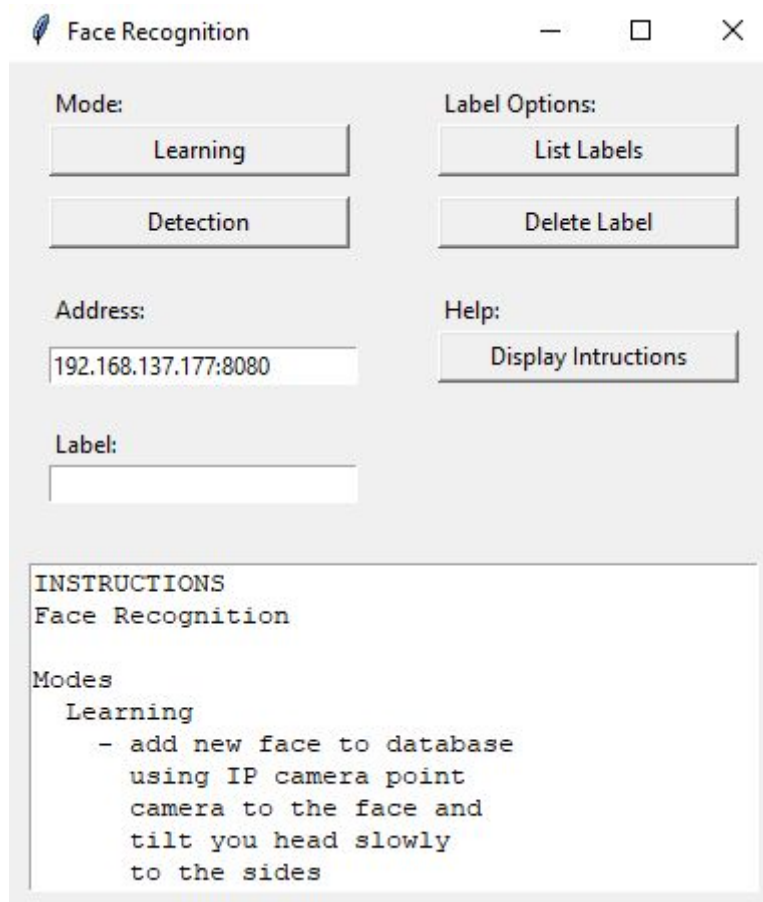
Aplikacja, napisana na systemy operacyjne Windows, do której prawidłowego działania potrzebna jest możliwość podłączenia do komputera kamery IP. Aplikacja ta służy do identyfikacji osób, które znajdują się w polu widzenia kamery. Zarówno interfejs jak i komunikaty wyświetlane użytkownikowi są w języku angielskim, ze względu na największą popularność i międzynarodowość języka. Sama aplikacja udostępnia trzy główne funkcjonalności:

- **Dodawania nowej osoby**, która służy do uczenia się twarzy pojedynczej osoby, znajdującej się przed kamerą. Wprowadza się imię i nazwisko takiej osoby, a następnie po włączeniu się podglądu wykrywanej twarzy, osoba ta powinna obracać się pod różnym kątem przed kamerą, zmieniając mimikę twarzy, tak by zapewnić jak najlepsze wykrywanie tej osoby w przyszłości, niezależnie od jej wyrazu twarzy. W dowolnym momencie użytkownik może zakończyć naukę. Wtedy twarz, wraz z przypisanym do niej imieniem i nazwiskiem, zostanie dopisana do bazy wyuczonych twarzy, którą aplikacja tworzy w swojej lokalizacji, w postaci pliku tekstowego facerec\_128D.
- **Identyfikacji osób**, w której otwierany jest podgląd widoku z kamery. Twarze wszystkich widocznych na nim osób zakreślone są kwadratem, a dodatkowo te, które zostaną rozpoznane jako wcześniej nauczone, zostaną opisane nazwą, podaną przy nauce danej twarzy jako imię i nazwisko.
- **Usuwanie danych konkretnej osoby** z bazy danych tak, by nie było możliwe dalsze rozpoznawanie tej osoby w trybie identyfikacji osób opisanym powyżej.

W celu poprawnego działania, kamera powinna znajdować się w tej samej sieci co komputer obsługujący aplikację. Następnie połączenie następuje poprzez wpisanie adresu IP kamery w interfejsie aplikacji.

## Ekran aplikacji:

### 1.1. Menu główne aplikacji:



The screenshot shows the main menu of the 'Face Recognition' application. The window has a title bar with the text 'Face Recognition' and standard window controls. The interface is divided into several sections:

- Mode:** Two buttons, 'Learning' and 'Detection', are stacked vertically.
- Label Options:** Two buttons, 'List Labels' and 'Delete Label', are stacked vertically.
- Address:** A text input field containing the IP address '192.168.137.177:8080'.
- Help:** A button labeled 'Display Instructions'.
- Label:** An empty text input field.
- INSTRUCTIONS:** A large text area at the bottom containing the following text:

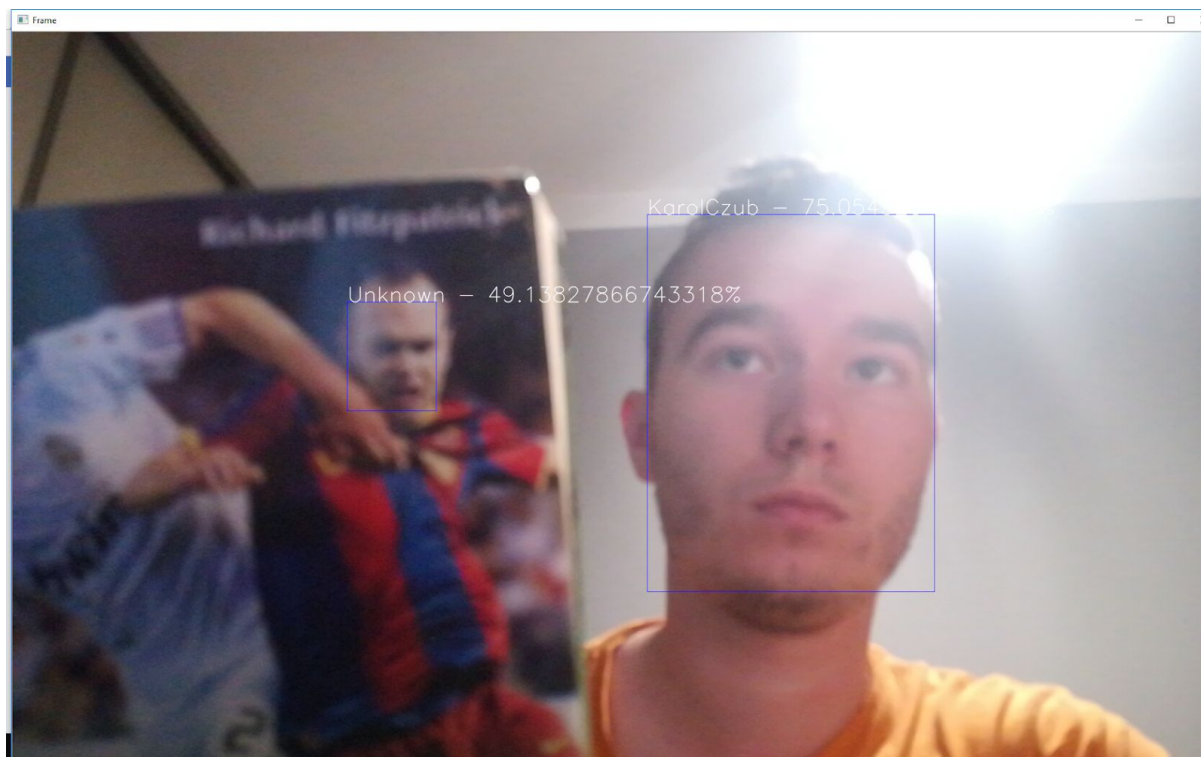
```
INSTRUCTIONS
Face Recognition

Modes
  Learning
    - add new face to database
      using IP camera point
      camera to the face and
      tilt you head slowly
      to the sides
```

Jest to Menu główne aplikacji, które umożliwia:

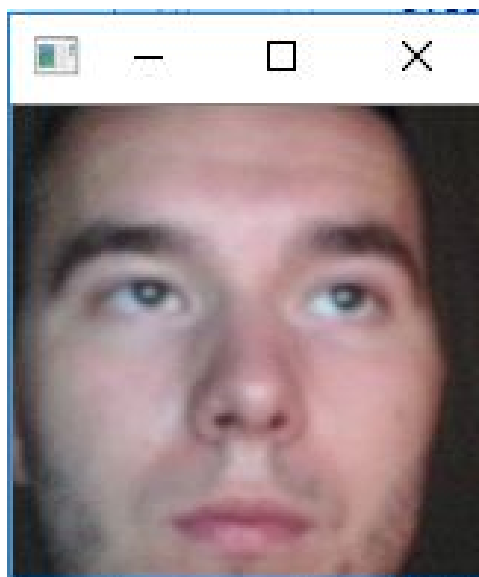
- Wybranie odpowiedniego trybu pracy: Learning, czyli tryb dodawania nowej osoby lub Detection, czyli tryb identyfikacji osób
- Wpisanie adresu IP kamery, której aplikacja będzie używać (pole Address).
- Sprawdzenie listy osób, które aplikacja rozpoznaje (przycisk List Labels)
- Usunięcie danych konkretnej osoby z bazy osób, które aplikacja rozpoznaje (przycisk Delete Label)
- Wybranie nazwy nowej osoby, która zostanie dodana lub wskazanie nazwy osoby, której dane mają zostać usunięte (pole Label)
- Wyświetlenie instrukcji odnośnie użytkowania aplikacji (przycisk Display Instructions), która jest automatycznie wyświetlana na samym początku lecz może przestać się wyświetlać w przypadku na przykład wystąpienia komunikatu o błędzie
- Komunikację z użytkownikiem za pomocą pola tekstowego na dole ekranu aplikacji. Służy do wyświetlania instrukcji oraz komunikatów o błędach

## 1.2. Identyfikacja osób:



Ekran podglądu z kamery IP, wraz z zaznaczonymi w niebieskie prostokąty twarzami osób, które aplikacja wykrywa. Jeżeli dana osoba została rozpoznana, to nad górną krawędzią kwadratu widnieć będzie jej nazwa, która została podana przy dodawaniu jej do bazy. W przeciwnym wypadku widnieć będzie napis "Unknown". Oprócz tego obok etykiety wyświetla się procentowa wartość dopasowania, do najbardziej zbliżonego ze wzorców. Wartość powyżej 70% oznacza uznanie za tą samą osobę.

### 1.3. Dodawanie nowej osoby:



Ekran podglądu wycinku z kamery, zawierającego wykrywaną twarz, dzięki czemu użytkownik ma pewność, że jest prawidłowo wykrywany, a pokazywane do kamery profile twarzy lub zmieniana mimika zdążyła zostać zarejestrowana. Od tego jak dokładne i różnorodne oblicza swojej twarzy użytkownik pozwoli zapamiętać aplikacji, zależy skuteczność identyfikacji osób w przyszłości.

## 2. Uzasadnienie wyboru tematu.

Temat ten został przez nas wybrany z powodu zainteresowania uczeniem maszynowym. Działanie sztucznej inteligencji i jej możliwości w postaci między innymi identyfikowania osób i obiektów są nie tylko ciekawym zagadnieniem, ale też czymś, na czego rozumienie i umiejętność implementacji jest coraz większe zapotrzebowanie. Dodatkowo kierunek ten się cały czas rozwija, ponieważ możliwości sztucznej inteligencji są praktycznie nieograniczone, więc programistom zajmującym się uczeniem maszynowym raczej nie grożą problemy ze znalezieniem ciekawej pracy. Uznaliśmy, że wiedza którą trzeba przyswoić do implementacji tej aplikacji - technologie, metody i algorytmy używane do rozpoznawania obiektów przez sztuczną inteligencję, jak i sama implementacja będą ważnym elementem naszego rozwoju w tej dziedzinie. Dodatkowo uznaliśmy, że branie udziału w takim projekcie pozwoli nam sprawdzić czy kierunek ten jest rzeczywiście tym, w czym chcielibyśmy pracować w przyszłości.

Projekt okazał się całościowym przeglądem technik używanych w dziedzinach takich jak: Face Detection, Face Recognition, Motion Tracking oraz ich wad, zalet i ogólnych możliwości. Przydatną wartością jest znajomość ograniczeń poszczególnych metod jak na przykład odporność na zmiany oświetlenia czy

efektywność wykrywania względem odległości od urządzenia rejestrującego obraz. Mając te kwestie na uwadze można odpowiednio dobrać metodę detekcji dla danego zadania, jest to szczególnie istotne, gdyż żaden algorytm nie cechuje się absolutną elastycznością i odpornością na wszelkie zakłócenia. Uczenie maszynowe jest pomocne ze względu na swoją uniwersalność (pod warunkiem odpowiedniego wyszkolenia), jednak wymaga stworzenia, bądź znalezienia odpowiedniej wielkości zbioru uczącego, a samo szkolenie modelu jest czasochłonne.

### 3. Podział prac między członków zespołu.

Zadanie	Wykonawcy
Organizacja sprzętowa	Karol Czub
Konfiguracja bezprzewodowego przechwytywania obrazu z kamery IP	Karol Czub
Implementacja wykrywania twarzy	Adam Dębaczak
Implementacja rozpoznawania twarzy za pomocą metod udostępnianych przez bibliotekę OpenCV	Karol Czub
Implementacja rozpoznawania twarzy za pomocą TensorFlow	Adam Dębaczak, Karol Czub
Implementacja dodawania nowej osoby do zbioru twarzy nauczonych	Adam Dębaczak, Karol Czub
Implementacja usuwania istniejącej osoby ze zbioru twarzy nauczonych	Adam Dębaczak, Karol Czub
Interfejs aplikacji	Adam Dębaczak

### 4. Funkcjonalności oferowane przez aplikację.

W aplikacji zostało zaimplementowano siedem głównych funkcjonalności, w których skład wchodzi: Identyfikacja osób, Dodawanie nowej osoby, Wyświetlanie listy rozpoznawanych osób, Usuwanie rozpoznawanej osoby z bazy, Ustanowienie połączenia z kamerą przy użyciu IP, Wyświetlanie instrukcji dla użytkownika oraz Wyświetlanie komunikatów o błędach.

#### 4.1. Identyfikacja osób.

Aplikacja otwiera okno, w którym wyświetla podgląd ekranu z kamery. Następnie za pomocą algorytmu MTCNN wykorzystującym sieci neuronowe, znajduje wszystkie twarze jakie występują na tymże podglądzie, zarówno od frontu jak i z profilu, które zaznacza na ekranie podglądu niebieskim prostokątem. Kolejnym krokiem jest porównanie każdej z wykrytych twarzy z twarzami zapisanymi w bazie danych. Aby to zrobić, pobierane są dane wszystkich osób z pliku `facerec_128D.txt`, który znajduje się w głównym folderze aplikacji. Dane te przechowywane są w formacie JSON i składają się z etykiet użytkowników, podawanych przy wprowadzaniu kolejnych osób do bazy danych oraz ze stu dwudziestoośmio wymiarowego wektora dla frontu i każdego profilu twarzy, w którym przetrzymywane są odległości pomiędzy cechami twarzy. Wszystkie wykryte twarze podczas identyfikacji osób zostają przeskalowane w jednakowy sposób, co przy wprowadzaniu nowej osoby, dzięki czemu niwelowana jest różnica odległości osoby od kamery. Następnie, każda z nich również zostaje przedstawiona w postaci opisanego wyżej wektora, a następnie następuje porównanie wektorów z tymi, które są zapisane w bazie. Działa ono w taki sposób, że sprawdzane są różnice odległości cech twarzy wykrytej i tych z bazy danych. Wybierana jest wtedy z pliku twarz najlepiej dopasowana i jeżeli zapamiętany model pokrywa się z wykrytym w minimum siedemdziesięciu procentach, to aplikacja uznaje, że jest to ta sama osoba i podpisuje dany kwadrat etykietą tej osoby, nad górną krawędzią kwadratu. W przeciwnym wypadku zamiast etykiety wpisane zostaje "Unknown". Do tego oprócz etykiety wyświetlana jest procentowo wartość, z jaką dana twarz pokrywa się z najlepiej dopasowanym modelem z bazy.



## **4.2. Dodawanie nowej osoby.**

Aplikacja łączy się z kamerą i rozpoczyna wykrywanie twarzy. Po znalezieniu jakiegś na obrazie z kamery, uruchamiane jest okno, na którym wyświetla się podgląd nagrywanej twarzy. Użytkownik powinien pokazać się zarówno od frontu jak i z obu profili, gdyż aplikacja zapisuje osobny wektor odległości między cechami dla każdego profilu i frontu twarzy. Jest to niezwykle ważne, aby uzyskać w przyszłości dobre wyniki przy identyfikacji osób. Równie istotne jest, aby podczas dodawania nowej osoby do bazy, w polu widzenia kamery znajdowała się tylko jedna osoba. W przypadku kiedy więcej niż jedna twarz będzie wykrywana podczas tego trybu, aplikacja będzie przeskakiwać pomiędzy każdą z nich i zapisane dane będą mieszaną cech każdej z twarzy, co w efekcie doprowadzi w większości przypadków do nierozpoznawania żadnej z nich. Kiedy użytkownik zakończy naukę nowej osoby przez aplikację, zestaw wektorów opisujących jego twarz zostaje przypisany do etykiety podanej przy rozpoczynaniu dodawania, następnie całość zostanie spakowana do formatu JSON i zapisana do pliku `facerec_128D.txt`, znajdującego się w folderze głównym aplikacji.

## **4.3. Wyświetlanie listy rozpoznawanych osób.**

Użytkownik ma możliwość wyświetlenia listy rozpoznawanych przez aplikację osób bezpośrednio z menu głównego. Działa to poprzez wczytanie pliku będącego bazą wszystkich zapamiętanych twarzy o nazwie `facerec_128D.txt`, znajdującego się w folderze głównym aplikacji, a następnie wyodrębnienie z niego samych etykiet, które zostały wpisane przez użytkownika przy dodawaniu konkretnych osób. Funkcjonalność ta przydaje się zarówno do upewnienia się, czy dana osoba jest w bazie danych aplikacji jak i do usuwania, gdyż wymaga ono wpisania etykiety osoby do usunięcia z bazy.

## **4.4. Usuwanie rozpoznawanej osoby z bazy.**

Aplikacja umożliwia użytkownikowi usunięcie danych dotyczących konkretnej osoby z bazy. Opcja ta jest dostępna z menu głównego aplikacji i wystarczy do niej wpisać etykietę osoby, którą chce się usunąć, a następnie kliknąć Delete Label. Po naciśnięciu przycisku, aplikacja pobiera wszystkie obiekty JSON z pliku facerec\_128D.txt, znajdującego się w folderze głównym, a następnie rozpakowuje je, sprawdzając etykietę osoby, której dane są przetrzymywane wewnątrz każdego z obiektów. Po znalezieniu etykiety, odpowiadającej wpisanej przez użytkownika nazwie, obiekt taki jest usuwany, a plik ten jest nadpisywany już bez niego.

#### **4.5. Ustanowienie połączenia z kamerą przy użyciu IP.**

Po podaniu przez użytkownika adresu IP kamery w formacie x.x.x.x:y, gdzie każdy x odpowiada liczbie z przedziału od 0 do 255, a y jest maksymalnie pięciocyfrowym numerem portu oraz wybraniu trybu pracy (Detection lub Learning), aplikacja łączy się z podanym adresem i zaczyna przechwytywanie obrazu. Funkcjonalność ta jest udostępniona przez biblioteka OpenCV poleceniem VideoCapture. W przypadku gdy obraz do przechwytywania nie został wykryty na podanym adresie, wtedy aplikacja wyświetla użytkownikowi komunikat o błędzie związanym z niepoprawnym adresem IP.

#### **4.6. Wyświetlanie instrukcji dla użytkownika.**

Aplikacja informuje użytkownika o możliwych do wybrania funkcjonalnościach i sposobie ich użycia. Dzieje się to poprzez pole tekstowe w menu głównym aplikacji. Po uruchomieniu aplikacji wyświetla się w nim komunikat, który opisuje ogólnie działanie każdego z trybów pracy oraz znaczenie pól, które użytkownik musi wypełnić (jak na przykład adres IP kamery). Oprócz tego po wybraniu danego trybu pracy instrukcja zostaje zmieniona na taką, która opisuje dokładnie tryb wybrany przez użytkownika i mówi w jaki sposób z niego korzystać. Po powrocie do menu głównego istnieje również przycisk Display Instructions, który służy do przywrócenia instrukcji głównej, która wyświetlała się na samym początku po uruchomieniu aplikacji.

#### **4.7. Wyświetlanie komunikatów o błędach.**

W przypadku wystąpienia jakiegoś błędu, aplikacja informuje o nim użytkownika za pomocą wyskakującego popup'u, wraz z opisem problemu. Przykładem jest wpisanie niepoprawnego IP kamery, z którego aplikacja nie jest w stanie pobrać obrazu. Wyświetla się wówczas popup z komunikatem "Wrong address, Camera connection error".

## **5. Użyte technologie.**

### **5.1. OpenCV.**

### **5.2. Numpy.**

### **5.3. Tensorflow.**

### **5.4. JSON.**

### **5.5. Tkinter.**

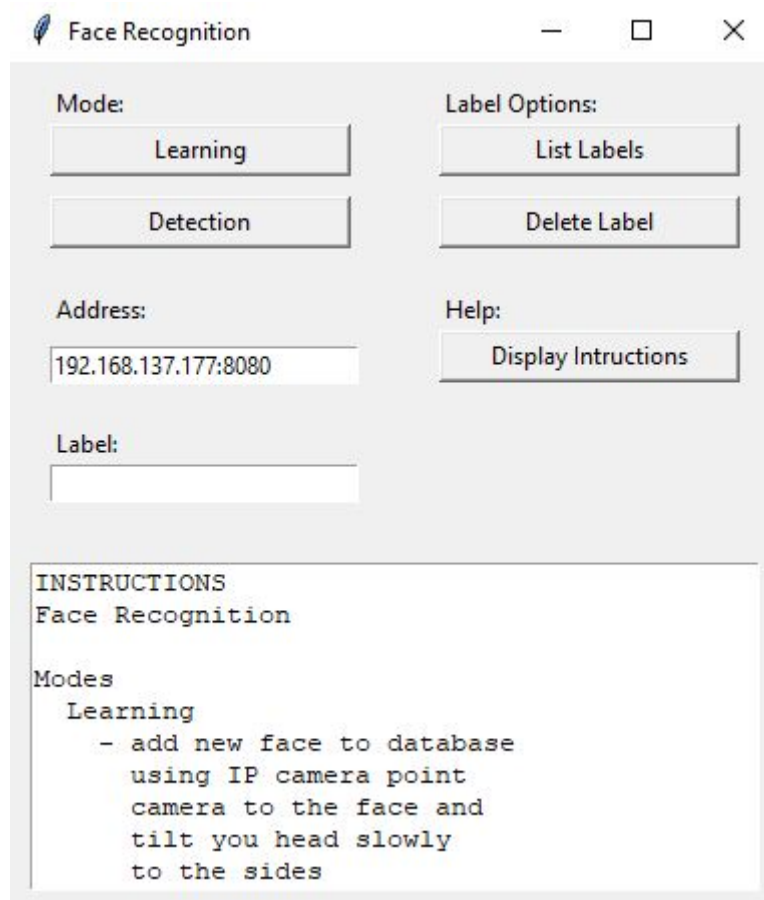
## **6. Architektura rozwiązania.**

## **7. Problemy podczas implementacji aplikacji oraz testy.**

### **Testy:**

## 8. Instrukcja użytkowania.

Aby uruchomić naszą aplikację, wymagany jest Python, wraz z zainstalowanymi bibliotekami: OpenCV, Numpy, Tensorflow, Tkinter. Kiedy mamy już tak skonfigurowany interpreter, należy pobrać aplikację z githuba: <https://github.com/Nagerrack/PT-Face-Detection-Tracking/tree/master/Face%20Recognition> oraz modele, niezbędne do prawidłowego rozpoznawania twarzy z google drive'a: <https://drive.google.com/file/d/0Bx4sNrhaBr3TDRMMUN3aGtHZzg/view> . Modele po pobraniu, należy umieścić w folderze models, który to znajduje się w głównym katalogu projektu. Następnie wystarczy jedynie uruchomić plik GUI.py, aby aplikacja uruchomiła się i wyświetliła interfejs użytkownika:



Pola interfejsu, w jakich użytkownik może coś wpisać to pola: Address oraz Label.

Pole Address jest miejscem, w którym użytkownik musi wprowadzić IP wraz z portem, na którym jest połączony z kamerą IP. Jest to absolutnie niezbędne i bez tego aplikacja nie pozwoli na działanie w żadnym trybie pracy.

Pole Label jest niezbędne jedynie w dwóch przypadkach: w przypadku wybrania trybu Learning lub kliknięcia przycisku Delete Label, których opis można znaleźć poniżej. Aplikacja wymaga, aby osoba, która będzie dodana do bazy była opisana przy pomocy niepustej etykiety lub w przypadku usuwania rekordu z bazy, by określić etykietę przeznaczoną do usunięcia ze zbioru.

Kiedy już wiemy, które pole musimy wypełnić w jakim przypadku, możemy wybrać tryb pracy naszej aplikacji:

Przycisk Learning służy do uruchomienia trybu dodawania nowej osoby do bazy danych. Do jego działania należy wpisać adres IP w polu Address oraz etykietę dodawanej osoby, która będzie się przy niej wyświetlać w polu Label.

## 9. Bibliografia.

- OpenCV kaskada Haara  
[https://docs.opencv.org/3.3.0/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html)

- Kaskada Haara vs Histogram LBP  
<https://www.superdatascience.com/opencv-face-detection/>
- Konwolucyjne sieci neuronowe w wykrywaniu i rozpoznawaniu twarzy  
<http://cs231n.stanford.edu/reports/2017/pdfs/222.pdf>
- Klasyfikacja emocji  
[https://github.com/oarriaga/face\\_classification](https://github.com/oarriaga/face_classification)
- Kaskada konwolucyjnych sieci neuronowych  
[https://kpzhang93.github.io/MTCNN\\_face\\_detection\\_alignment/](https://kpzhang93.github.io/MTCNN_face_detection_alignment/)
- Przegląd metod segmentacji obrazów  
<https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4>
- Tkinter dokumentacja  
<https://docs.python.org/2/library/tkinter.html>
- OpenCV dokumentacja  
<https://docs.opencv.org/3.4.1/>