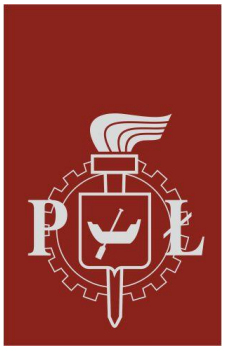
07.10.2018, Łódź



System rozpoznawania twarzy

Dokumentacja

1. Wprowadzenie

Ludzka twarz odgrywa bardzo istotną rolę w interakcjach społecznych, przekazując informacje dotyczące tożsamości czy emocji. Twarz jest najbardziej naturalnym identyfikatorem tożsamości w codziennym życiu a jej rozpoznawanie, mimo że łatwe dla ludzi, stanowi nie lada wyzwanie dla maszyn starających się imitować tę umiejętność. Wymaga ono opracowania wielu skomplikowanych algorytmów, które należy poprawnie wdrożyć mając na uwadze ogromne ilości danych do przetworzenia. Algorytmy te służą do:

- Detekcji twarzy
- Ekstrakcji cech twarzy
- Budowy bazy danych twarzy
- Identyfikacji twarzy

Są to podstawowe składniki systemu identyfikacji osób, które w zależności od wybranych metod będą charakteryzować się różną skutecznością. Sama identyfikacja osób jest o tyle problematyczna, ponieważ twarz charakteryzuje się naturalną zmiennością zależną od mimiki twarzy, emocji a także zmiennością wynikającą z procesu starzenia się. Czynników utrudniających identyfikację osób jest znacznie więcej i stanowią one wyzwanie w skutecznym rozpoznawaniu twarzy. Problem rozpoznawania twarzy (w kategoriach wizji komputerowej) jest formułowany jako:

Mając zdjęcie bądź klatkę wideo sprawdź czy jedna bądź więcej osób należy do osób przechowywanych w bazie zidentyfikowanych twarzy – i jeśli tak, ustal do której.

Należy tutaj rozróżnić również weryfikację rozpoznawania twarzy od jej identyfikacji. Weryfikacja ma na celu potwierdzenia tożsamości danej osoby, czyli porównanie zadanej twarzy z wzorcem w bazie danych i określenie czy dana twarz należy do osoby ze wzorca. Identyfikacja natomiast zajmuje się ustaleniem tożsamości zadanej twarzy, czyli ocena podobieństwa twarzy ze wszystkimi wzorcami w bazie danych. Te operacje znacząco różnią się stopniem trudności i złożonością obliczeniową, dlatego też istotnym jest ich rozróżnianie.

Główny sposób klasyfikacji metod rozpoznawania twarzy to podział na metody analityczne i metody bazujące na podejściu globalnym.

- **Metody analityczne** - polegają na użyciu strukturalnych cech twarzy, gdzie lokalizuje się i mierzy charakterystyczne cechy twarzy. Przykładowe cechy brane pod uwagę to kolor, geometria, kontury twarzy, położenie ust, oczu, nosa, kształt tych elementów, etc. Są to cechy niezależne od oświetlenia.
- **Metody globalne** (holistyczne) – bazują na podejściu globalnym, a więc rozpoznanie twarzy bazuje na jej całościowym obrazie, bez wyodrębnienia cech strukturalnych. Bardzo częstym jest utworzenie podprzestrzeni o mniejszej liczbie wymiarów w celu optymalizacji danych przy jednoczesnym zachowaniu własności statystycznych obrazów twarzy.

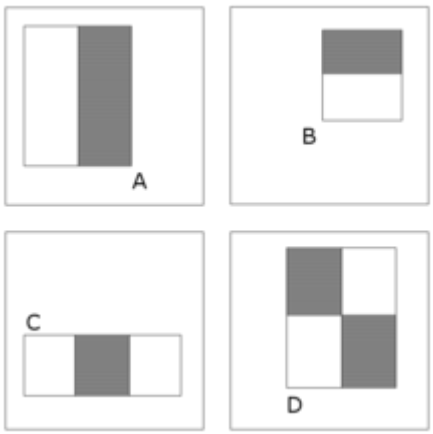
Metody zaimplementowane w aplikacji należą do grupy metod globalnych rozpoznawania twarzy

1.1. Detekcja twarzy

W celu identyfikacji twarzy w pierwszej kolejności twarz na zdjęciu bądź w klatce wideo musi zostać wykryta. Detekcja twarzy może zostać określana jako konkretny przypadek wykrywania obiektów, którego celem jest znalezienie wszystkich obiektów o wszystkich rozmiarach na zdjęciu należących do danej klasy. Zważywszy na fakt „trójwymiarowości" twarzy najczęściej spotykane detektory twarzy skupiają się na frontalnych twarzach. Najbardziej rozpowszechnioną metodą obecna w większości smartphonów i kamer jest metoda Viola-Jones skupiająca się na tzw. falkach Haar'a.

Detekcja twarzy metodą Viola-Jones jest pierwszą strukturą wykrywania twarzy zapewniającą wysoki współczynnik wykrywalności twarzy w czasie rzeczywistym. Struktura ta nie jest jedynie skuteczna w detekcji twarzy, jako że może być wytrenowana do poszukiwania innych obiektów, ale jej stworzenie było motywowane wykrywaniem twarzy.

Algorytm Viola-Jones jest często nazywany klasyfikatorem Haar'a, ponieważ wykorzystuje on tzw. cechy Haar'a. Poszukiwane cechy są związane z sumą pikseli obrazu w obrębie prostokątnych obszarów. W tej formie mają pewne podobieństwo do podstawowych funkcji Haar'a, ale z racji operowania na prostokątnych powierzchniach są bardziej skomplikowane. Przykładowe poszukiwane cechy Haar'a są widoczne poniżej:



Rys. 1. Przykładowe funkcje prostokątne, tzw. cecha Haar'a.

Cechy te zawierają w sobie informacje o zmianie wartości kontrastu pomiędzy prostokątnymi grupami umożliwiając określanie relatywnie jasnych i ciemnych obszarów. Eksploatują one fakt, że wszystkie twarze mają pewne podobieństwa, które mogą zostać wykryte z użyciem cech Haar'a. Proporcje twarzy są podobne między ludźmi, region oczy jest ciemniejszy niż górna część polików, nos jest jaśniejszy niż obszar obok nosa itd. Wartość danej cechy jest obliczana jako suma pikseli w regionie jasnych prostokątach minus suma

pikseli w regionie ciemnych. Obecność cechy Haar'a jest zdeterminowana przekroczeniem wartości szumu pikseli przez wartość cechy. Z racji łatwości skalowania cech można je stosować do jednego zdjęcia w wielu skalach, umożliwiając wykrycie twarzy o różnych wielkościach.

Rys. 2. Cecha Haar'a podobna do regionu oczu.

Metoda Viola-Jones'a wykorzystuje dużą liczbę cech Haar'a, które służą jako słabe klasyfikatory – które posiadają prawidłową odpowiedź (dotyczącą wykrycia twarzy) nieco częściej niż przy przypadkowym zgadywaniu. Mając znaczącą liczbę klasyfikatorów wskazujących nieznacznie w stronę jednej z odpowiedzi można się spodziewać konkretnego rozwiązania. Podsumowując, z wielu słabych klasyfikatorów tworzy się jeden silny używając metody AdaBoost. AdaBoost wybiera zestaw słabych klasyfikatorów przypisując im wagi. Mając łańcuch filtrów, gdzie każdy determinuje występowanie danej cechy Haar'a detekcja twarzy jest pozytywna, gdy dany analizowany region przejdzie przez filtry w całym łańcuchu, w przeciwnym razie twarz nie zostanie wykryta. Kolejność filtrów w łańcuchu określona jest na podstawie wcześniej wspomnianych wag, gdzie filtry z największymi wagami są używane jako pierwsze, eliminując regiony bez twarzy, co skutkuje usprawnieniem szybkości działania algorytmu.

Wizualizacja metody detekcji twarzy Viola-Jones jest dostępna pod poniższym adresem: https://www.youtube.com/watch?v=hPCTwxF0qf4.

1.2. Twarze własne

Jedna z pierwszych metod identyfikacji osób na podstawie twarzy opracowana przez M. Truk i A. Pentland w 1991 roku, która została zapoczątkowana przez próbę reprezentacji zdjęć twarzy w mniejszej wymiarowości – z użyciem mniejszej liczby danych.

1.2.1. Wstępne przetwarzanie obrazu

Należy przygotować zbiór danych zawierających pewną liczbę osób i zdjęcia dla każdej z osób z odpowiednimi etykietami.

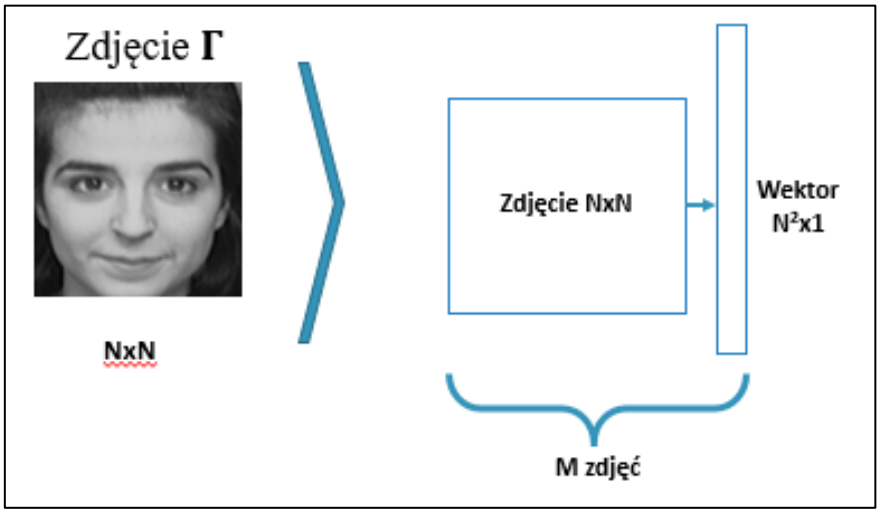
Region twarzy w odcieniach szarości powinien zostać wycięty biorąc pod uwagę takie same wymiary dla każdego kolejnego zdjęcia. W najlepszym wypadku oczy i usta powinny znajdować się na tej samej wysokości w każdym ze zdjęć a także zdjęcia powinny być wykonane w stałym oświetleniu.

Kolejnym krokiem jest normalizacja wartości w zdjęciu, wycięty obszar musi być poddany operacji wyrównania histogramów.

1.2.2. Macierz zdjęć

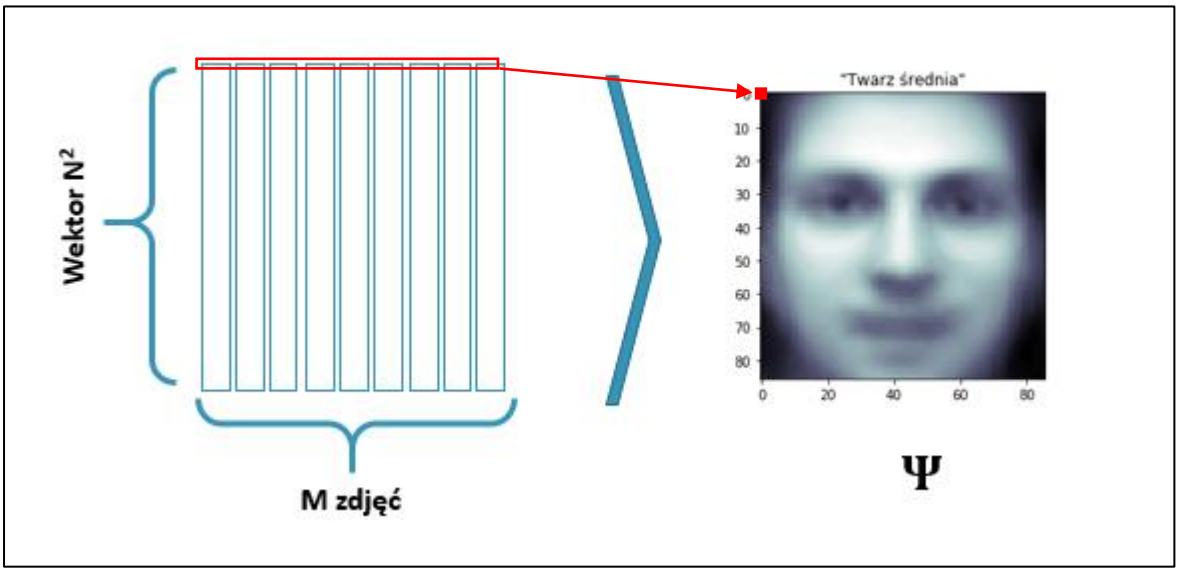
Dla uproszczenia zostanie przyjęte, iż zdjęcia w bazie danych użyte do wyznaczenia twarzy własnych są o wymiarach NxN, mimo że nie jest niezbędnym, aby zdjęcia były kwadratowe.

Pierwszym krokiem jest „spłaszczenie” wcześniej przetworzonych obrazów i przekształcenie ich do formy wektora.



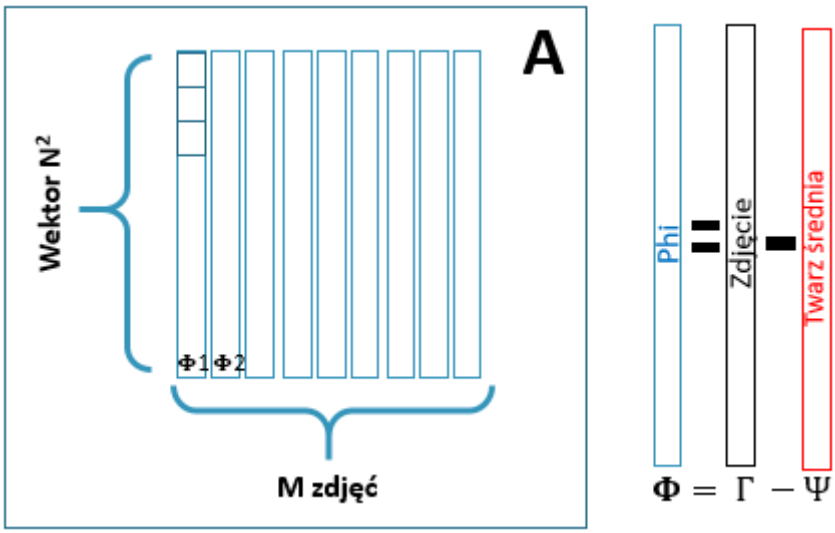
Rys. 3. Spłaszczenie zdjęcia do formy wektora.

Kolejnym krokiem jest stworzenie macierzy zawierającej wszystkie obrazy w formie wektorów (gdzie każdy obraz to jeden wektor), a następnie policzenie „twarzy średniej" z bazy danych. Taki „uśredniony" obraz jest otrzymywany poprzez wyliczenie średniej arytmetycznej dla wszystkich pikseli znajdującej się na danej pozycji, tj. średnia pierwszych wartości we wszystkich wektorach, potem drugich itd.



Rys. 4. Stworzenie macierzy ze zdjęciami i obliczenie z jej pomocą „twarzy średniej".

Po obliczeniu twarzy średniej należy odjąć twarz średnią od wszystkich zebranych w macierzy zdjęć co jest zaprezentowane na rysunku 5.



Rys. 5. Macierz A stworzona przez odjęcie od każdego oryginalnego zdjęcia twarzy średniej.

1.2.3. PCA

Jedną z głównych idei za twarzami własnymi jest użycie PCA w celu znalezienia w obrazach twarzy cech znaczących w formie nowego wektora cech. Samo PCA polega na znalezieniu takiego obrotu układu współrzędnych, który maksymalizuje wariancję kolejnych współrzędnych konstruując tym samym nową przestrzeń obserwacji. Wynikiem przekształcenia danych do nowej przestrzeni jest reprezentacja tych danych w mniejszej wymiarowości, czyli z użyciem mniejszej liczby informacji. Mając macierz A o wymiarach M na N^2 należy wyznaczyć macierz kowariancji do dalszych obliczeń metodą PCA.

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \qquad \text{Macierz } N^2xN^2$$

$$\text{gdzie} \quad A = [\Phi_1\ \Phi_2 \dots \Phi_M] \qquad \text{Macierz } N^2xM$$

Obliczona tym sposobem macierz kowariancji będzie mieć wymiary N^2xN^2 co powodowałoby obliczeniowe trudności wynikające z ogromnej liczby elementów dla którego znalezienie wektorów i wartości własnych byłoby nieefektywne. Problem ten został rozwiązany w następujący sposób: z racji niewielkiej liczności zbioru zdjęć twarzy (M) w porównaniu do ilości punktów obrazu (N^2) poszukiwanie wartości i wektorów własnych dla macierzy A^TA jest znacznie łatwiejsze niż dla macierzy AA^T. Z pozoru te zadania nie są ze sobą związane, jednakże biorąc pod uwagę macierz kowariancji:

$$(\mathbf{A}^T\mathbf{A})\mathbf{v}_i = l_i\mathbf{v}_i$$

(gdzie v_i to wektory własne zaś l_i to wartości własne) poprzez lewostronne przemnożenie obu stron równania przez macierz A otrzymujemy następującą zależność:

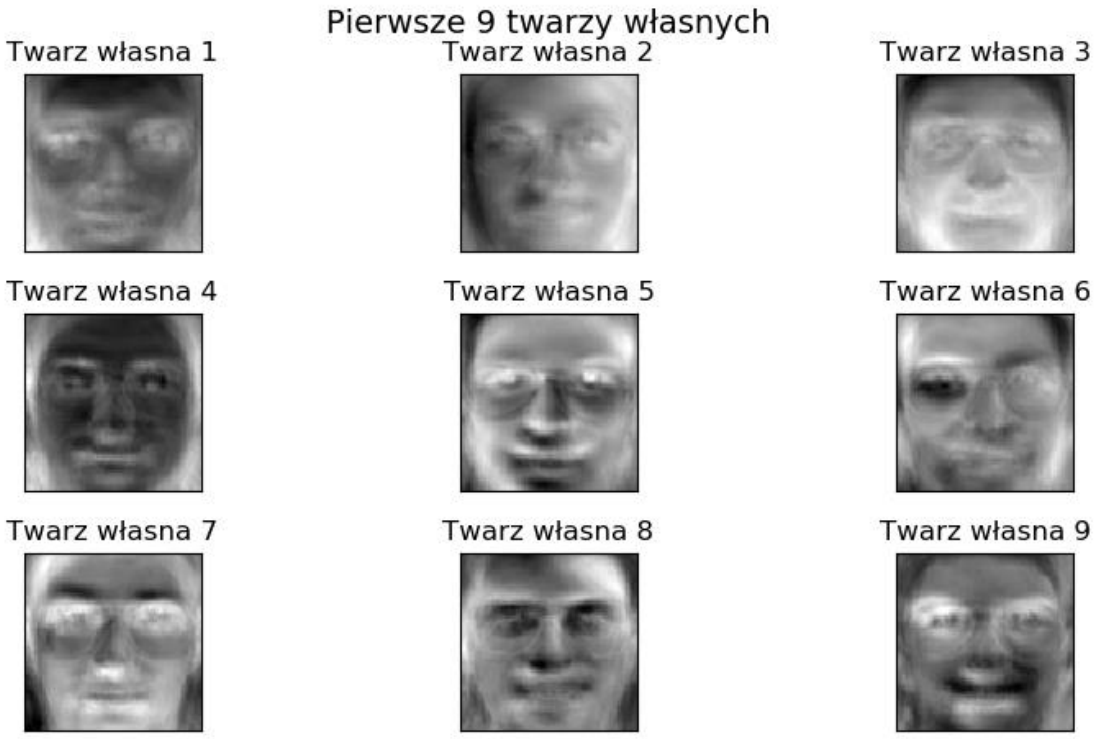
$$\mathbf{A}(\mathbf{A}^T\mathbf{A})\mathbf{v}_i = \mathbf{A}l_i\mathbf{v}_i \Rightarrow (\mathbf{A}\mathbf{A}^T)\mathbf{A}\mathbf{v}_i = l_i\mathbf{A}\mathbf{v}_i \Rightarrow (\mathbf{A}\mathbf{A}^T)\mathbf{w}_i = \lambda_i\mathbf{w}_i$$

gdzie wcześniej poszukiwane wektory własne zbioru próbek twarzy (w_i) są liniowymi kombinacjami możliwych do policzenia wektorów własnych macierzy $\mathbf{A}^T\mathbf{A}$ ($\mathbf{w}_i=\mathbf{A}\mathbf{v}_i$). Dlatego też analiza PCA dla zbioru obrazów twarzy polega na określeniu wszystkich rozwiązania $(\mathbf{A}^T\mathbf{A})\mathbf{v}_i = l_i\mathbf{v}_i$, a następnie na wyznaczeniu wektorów własnych w_i. Ostatnim krokiem jest utworzenie nowej przestrzeni cech zawierającej p-wektorów własnych o największych wartościach własnych.

1.2.4. Obliczenie reprezentacji twarzy

Aby uzyskać reprezentację obrazów należy go najpierw „rzutować” na p-kierunków nowej przestrzeni (dzięki obliczonym wcześniej wektorom własnym), co sprowadza się do obliczenia p-iloczynów skalarnych z wektorem obrazu jako argumentem wraz z wektorami własnymi. Obliczony zostanie w ten sposób p-elementowy wektor, który jest reprezentacją twarzy w nowej przestrzeni o p-wymiarach, umożliwiający opisanie twarzy używając mniejszej liczby informacji.

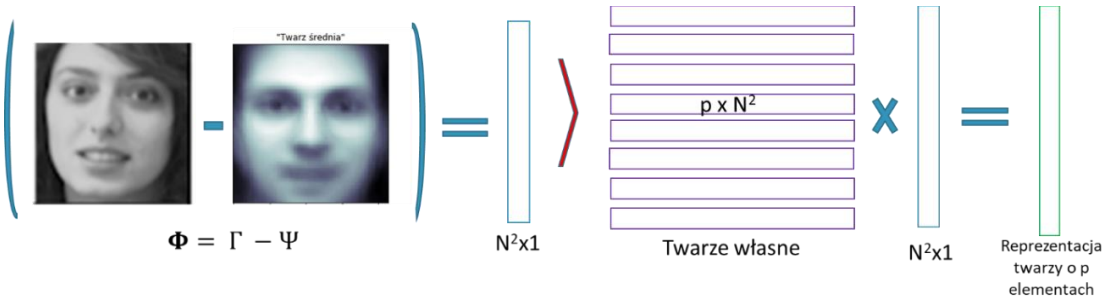
Nazwa metody „twarze własne” wynika z faktu, iż znalezione wektory własne mają identyczną strukturę do wektorów odpowiadających obrazom twarzy. Z tego też względu wektory własne można przedstawić w formie obrazu i przedstawiają one zniekształcone twarze – stąd też nazwa – twarze własne. Najjaśniejszy piksel w każdej z twarzy własnych oznacza, że w danym miejscu piksel najbardziej różni się między zdjęciami dla danej składowej głównej.



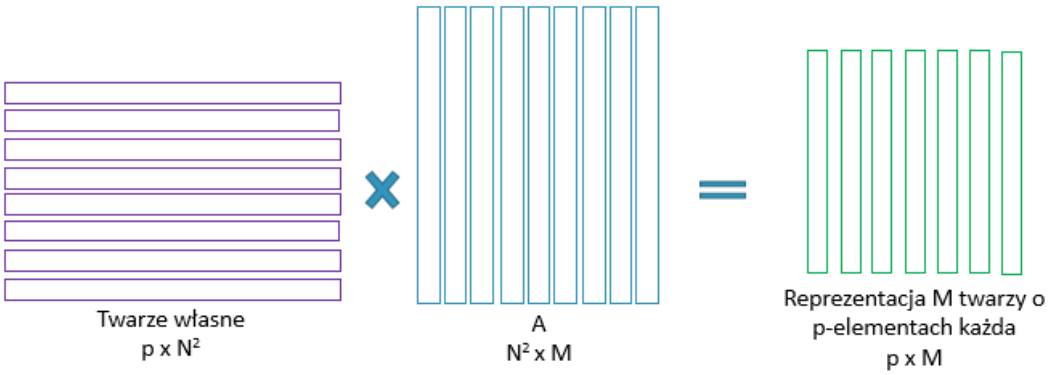
Rys. 6. Dziewięć pierwszych twarzy własnych.

Proces „rzutowania” twarzy z bazy danych na nową przestrzeń o mniejszej wymiarowości jest niezbędny, jako że twarze własne służą jedynie do otrzymania reprezentacji twarzy w nowej przestrzeni. Same w sobie nie umożliwiają identyfikacji twarzy. W pierwszej kolejności dla każdej z twarzy w bazie musi zostać obliczony wektor o p-elementach, aby móc zaprezentować twarz jako punkt w przestrzeni wielowymiarowej bądź jako wektor o mniejszej

wymiarowości niż wektor zdjęcia twarzy. Idea ta została zaprezentowana na rysunku 7 dla obliczenia reprezentacji dla jednej osoby (uwzględniając wstępne przetwarzanie obrazu i odjęcie twarzy średniej). Na rysunku 8 pokazane jest „rzutowanie" z użyciem macierzy A, zawierającej wszystkie wstępnie przetworzone zdjęcia wraz z odjęciem „uśrednionej twarzy.



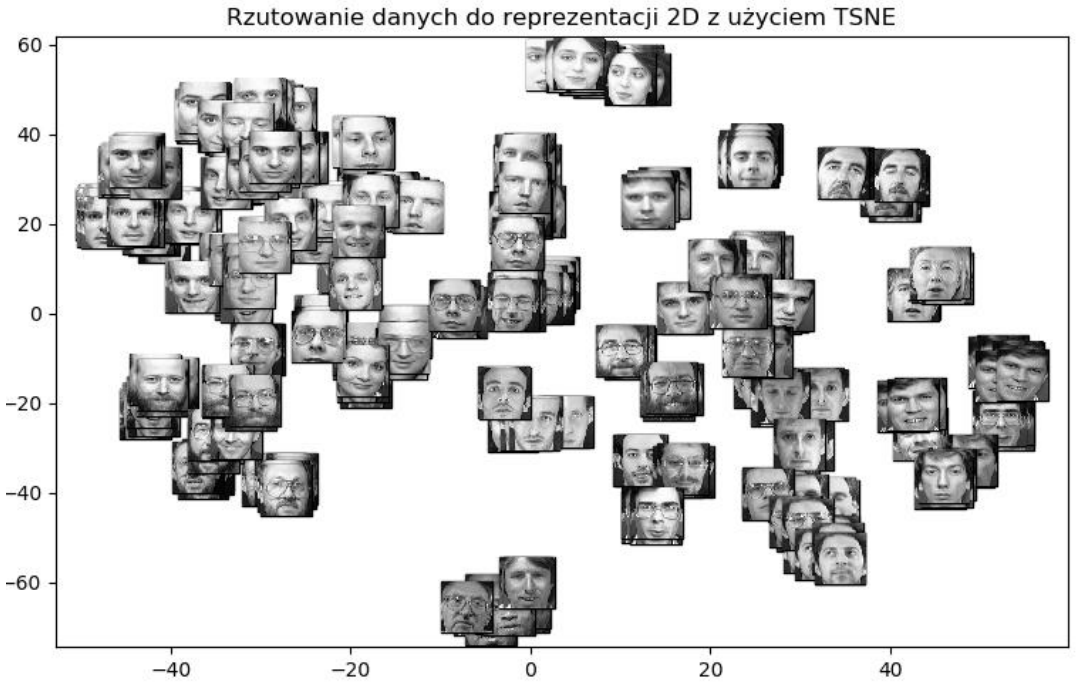
Rys. 7. Obliczanie reprezentacji pojedynczego zdjęcia w nowej przestrzeni.



Rys. 8. Obliczanie reprezentacji zbioru zdjęć w nowej przestrzeni.

1.2.5. Wizualizacja wielowymiarowych danych – t-SNE

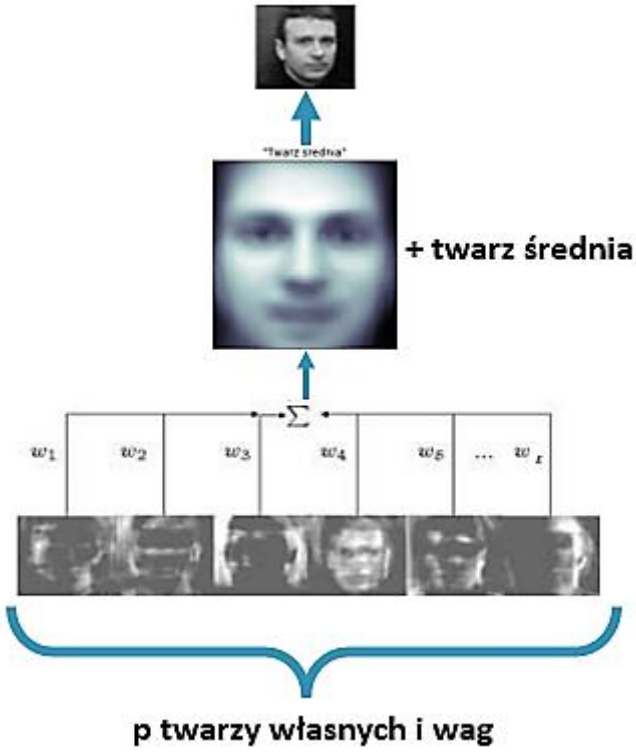
„Rzutując" wszystkie twarze na p-kierunków nowej przestrzeni cech jest wymagane, aby identyfikacja osób na podstawie twarzy była możliwa. Mając reprezentację wszystkich twarzy istnieje możliwość wizualizacji danych p-wymiarowych z użyciem metody t-distributed stochastic neighbor embedding (w skrócie t-SNE). Metoda ta umożliwia wizualizację wysokowymiarowych danych poprzez redukcję wymiarowości z zachowaniem odległości między poszczególnymi punktami (twarzami, w przestrzeni o p-wymiarach). Dane z bazy danych dla twarzy własnych zostały w ten sposób zwizualizowane na rysunku 9. Widać na niej wyraźnie, że niektóre z twarzy są praktycznie w jednym punkcie, jednakże zależnie od oświetlenia, mimiki czy orientacji twarzy w poszczególnych zdjęciach mogą wystąpić pewne różnice w pozycji.



Rys. 9. Wizualizacja punktów w przestrzeni p-wymiarowej w zredukowanej wymiarowości z użyciem metody t-SNE dla 40 osób.

1.2.6. Rekonstrukcja Twarzy

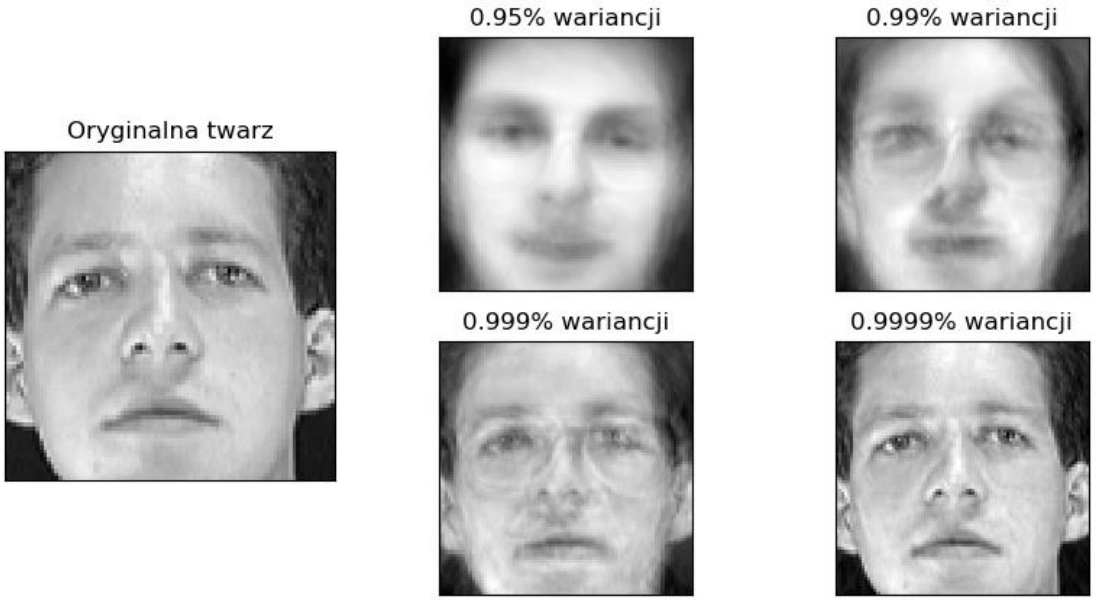
Mając już reprezentację każdej twarzy jako wektor p-elementowy poza klasyfikacją nowo dodanych twarzy możliwa jest rekonstrukcja twarzy na podstawie p-elementowego wektora wag. Przy wyznaczaniu reprezentacji twarzy najpierw odejmuje się od zdjęcia „średnią" twarz i później „rzutuje" się zdjęcie na p-kierunków nowej przestrzeni (obliczenie p-iloczynów skalarnych). W przypadku rekonstrukcji twarzy używamy twarzy średniej jako „bazę", do której dodajemy kolejne twarze własne przemnożone przez odpowiadające im wagi. Procedura ta została zaprezentowana na rysunku 10.



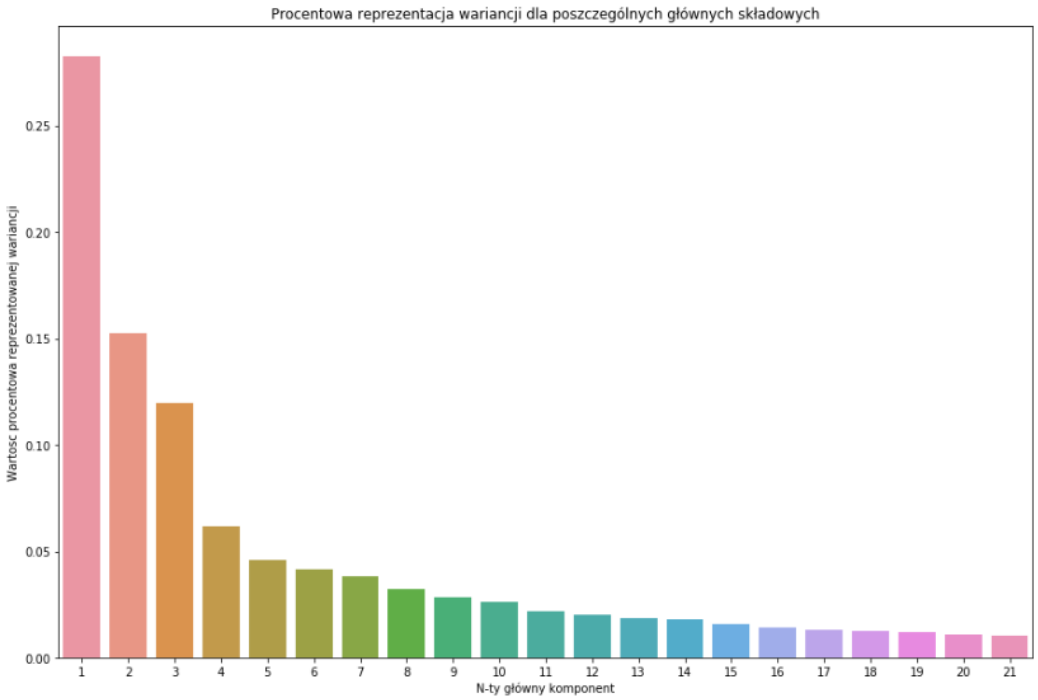
Rys. 10. Schemat rekonstrukcji twarzy

1.2.7. Wybór wartości procentowej wariancji w obliczaniu PCA

O tym jak bardzo podobno będzie twarz z rekonstrukcji zależy od procentowej wariancji „zawartej" w składowych głównych. Tym więcej wariancji zostanie zawarte używając większej ilości składowych głównych – tym dokładniejsze będzie zdjęcie po rekonstrukcji. Jako przykład na rysunku 11 została zaprezentowana rekonstrukcja twarzy z różną wartością procentową wariancji użytą do obliczeń PCA.



Rys. 11. Porównanie rekonstrukcji twarzy zależne od procentowej wariancji przy obliczeniu PCA.



Wyk. 12. Procentowa reprezentacja wariancji dla poszczególnych składowych głównych – prezentacja procentowego „udziału” poszczególnych składowych głównych.

Dodatkowo, na podstawie wykresu 12 widać, że pierwsza składowa główna odpowiada za około 28 % wariancji pomiędzy zdjęciami. Oznacza to, że w największym stopniu rozróżnia zdjęcia twarzy między sobą i każda kolejna składowa główna odpowiada za mniejszy procent

wariancji zbioru. W implementacji algorytmów PCA bardzo często nie określa się liczby składowych głównych do obliczenia, ale określa się procent wariancji jaki ma być „zawarty” w tych składowych i na tej podstawie uzyskuje się różną liczbę składowych, zależną od zbioru danych.

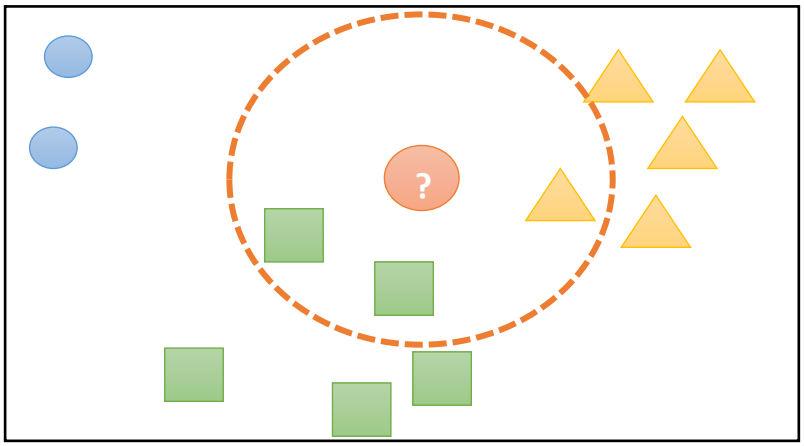
Na rysunku 13 jest przedstawiony wykres zdjęć po rzutowaniu ich jedynie na dwa pierwsze komponenty główne. Widoczne jest, że pierwszy komponent główny w większym stopniu „rozróżnia” między poszczególnymi zdjęciami i jest prawdopodobnie związany z oświetleniem twarzy. Nie należy się jednak kierować intuicją przy interpretacji składowych głównych, jako że mogą one nie mieć odniesienia do rzeczywistości, szczególnie dla kolejnych składowych głównych.

Rys. 13. Wizualizacja twarzy na wykresie, gdzie lokacje twarzy odpowiadają ich rzutowaniu na dwie pierwsze składowe główne.

1.2.8. Klasyfikacja k-najbliższych sąsiadów

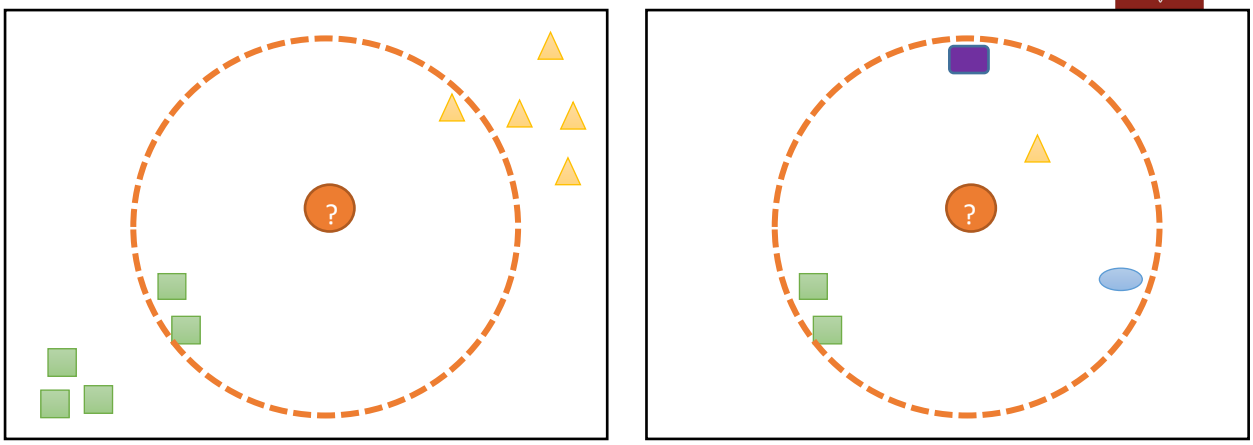
Po obliczeniu reprezentacji, wektorów wag, dla każdej twarzy z baz danych każdą twarz można interpretować jako punkt w przestrzeni wielowymiarowej. Istnieje wiele metod klasyfikacji, które mogłyby spełnić swoją rolę w identyfikacji osób a metodą wybraną na potrzeby aplikacji jest k-najbliższych sąsiadów (w skrócie knn, z ang. k-nearest neihbours).

Mając zbiór uczący z danymi (wektory wag) z przypisanymi im etykietami (identyfikujące osobę) istnieje potrzeba przewidzenia czy nowy wektor wag bez etykiety będzie należeć do którejś klasy w bazie danych, gdzie każdej klasa (osoba) zawiera pewną liczbę zdjęć danej osoby. Od nowej twarzy jako punkt w przestrzeni p-wymiarowej, można policzyć odległość do poszczególnych punktów, jednakże najbliższy punkt może nie być najlepszym kryterium do sprawdzenia przynależności. Z tego też względu w algorytmie k-najbliższych sąsiadów określa się stałą, całkowitą wartość, którą jest liczba sąsiadów. Liczba sąsiadów to liczba najbliższych obserwacji branych pod uwagę przy klasyfikacji. Schemat takiej klasyfikacji pokazany jest na rysunku 14.



Rys. 14. Metoda k-najbliższych sąsiadów, z 3 sąsiadami branymi pod uwagę. Algorytm zaklasyfikuje pomarańczową, nieznaną obserwację do klasy zielonych kwadratów.

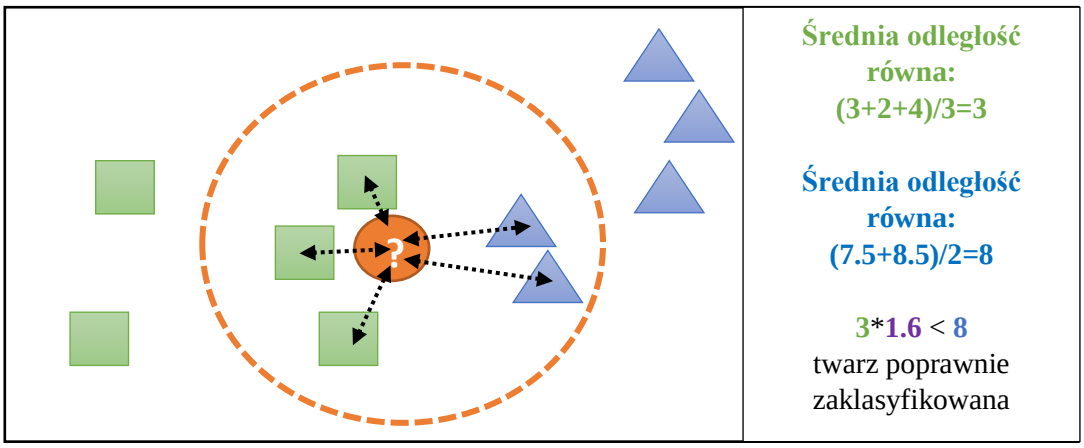
Na potrzeby klasyfikacji z użyciem twarzy własnych algorytm musiał zostać w pewnym stopniu zmodyfikowany z powodu pewnych problemów, które zostały zaprezentowane na rysunku 15. W jego rdzennej postaci nie byłoby możliwości określenia czy dana twarz znajduje się w bazie danych, ponieważ zawsze przyporządkuje on obserwację do jakiejś klasy. Dodatkowo przy liczbie rozważanych sąsiadów równej 5, gdyby pojawiło się 4 różnych kandydatów określenie przynależności do danej klasy również nie byłoby jednoznaczne.



Rys. 15. Problemy przy klasyfikacji twarzy z użyciem metody knn dla: nowej obserwacji przy rozważanych 3 sąsiadach (po lewo), dla dużej liczby kandydatów i wynikającej z niej niepewności co do proponowanej klasyfikacji przy rozważanych 5 sąsiadach (po prawo).

Zaimplementowany algorytm knn bierze pod uwagę 5 najbliższych sąsiadów. Z racji powyższych zastrzeżeń problemy zostały zaadresowane przez dodanie do algorytmu odpowiednich kryteriów:

- Jeśli przy klasyfikacji nowej obserwacji (twarzy) jest więcej niż 3 kandydatów twarz uznaje się za nową/nie można stwierdzić jej przynależności.
- Jeśli twarz została zaklasyfikowana jako należąca do klasy A należy jeszcze sprawdzić średnie arytmetyczne odległości do nowej obserwacji dla poszczególnych klas branych pod uwagę w knn. Następnie średnia wartość odległości proponowanej klasy, w tym wypadku A, zostanie przemnożona przez stałą wartość 1.6. Jeśli wartość po przemnożeniu będzie niższa od pozostałych klas – klasa zostaje uznana za poprawnie wyznaczoną. W przeciwnym razie uznaje się twarz za nową/nie można stwierdzić jej przynależności. Zobrazowanie tego kryterium widoczne jest na rysunku 15.



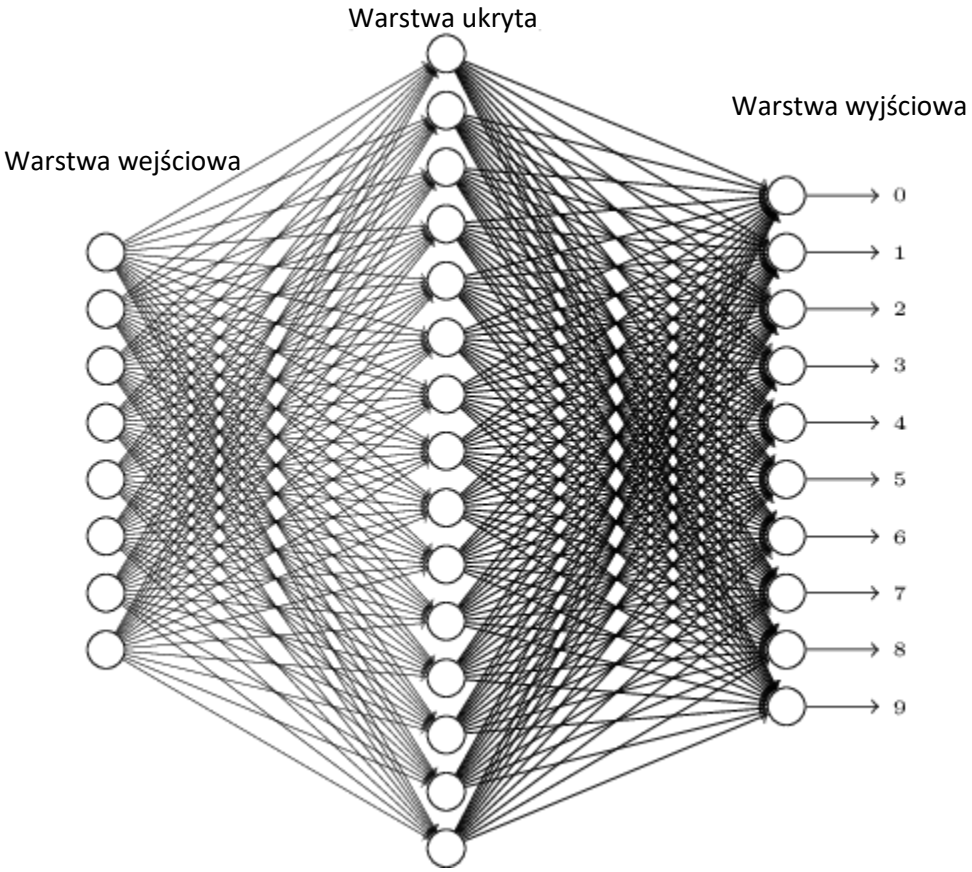
Rys. 16. Sposób sprawdzania poprawności klasyfikacji w zmodyfikowanej metodzie knn.

1.3. Konwolucyjne sieci neuronowe

1.3.1. Sieci neuronowe

Sieci neuronowe to nazwy struktur realizujących obliczenia lub przetwarzanie sygnałów poprzez rzędy (warstwy) elementów, które są nazywane sztucznymi neuronami i które wykonują podstawowe operacje na otrzymanych danych wejściowych. Mimo że wzorowane na układzie neuronowym, sieci neuronowe są w rzeczywistości wyłącznie strukturą matematyczną dostosowaną do konkretnego problemu. Generalnie warstwy sieci neuronowych da się podzielić na trzy typy:

- Warstwa wejściowa – tutaj przekazywane są dane początkowe, które mają zostać przeanalizowane. Każdy neuron z warstwy wejściowej przesyła dane do każdego neuronu w warstwie ukrytej.
- Warstwa ukryta – w warstwie tej zachodzi „uczenie" się sieci, czyli dostosowywanie jej parametrów, aby otrzymać wyniki zgodne z przekazanymi wcześniej, oczekiwanymi rezultatami.
- Warstwa wyjściowa – na wyjściu sieci znajduje się odpowiedź sieci, która wynika z danych wejściowych i przetworzenia ich przez warstwę ukrytą.

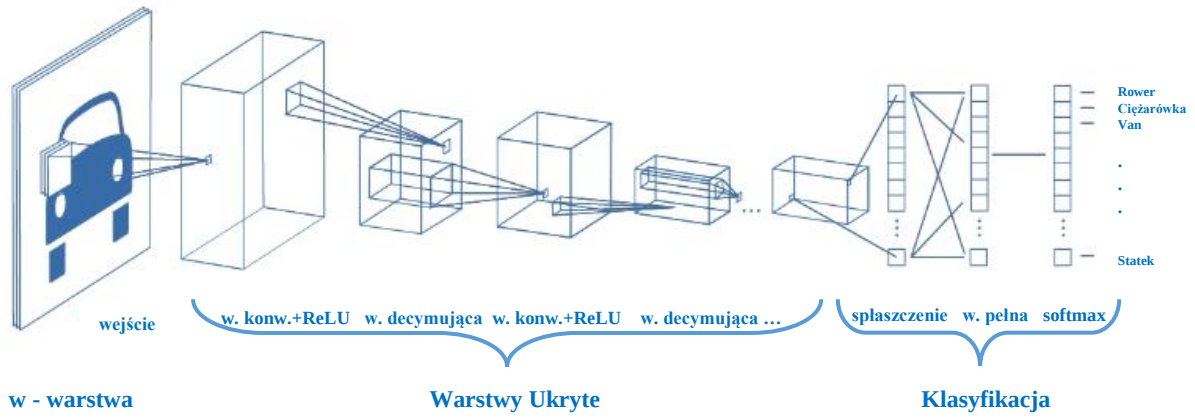


Rys. 17. Przykładowa sieć neuronowa z jedną warstwa ukrytą.

Opis ten znacząco upraszcza to czym są i co dzieje się w trakcie uczenia sieci neuronowych. Mogą one mieć ogromne ilości warstw i parametrów a same ich struktury mogą być znacznie bardziej skomplikowane niż ta widoczna na rysunku 17. Przykładowo, dla każdego piksela ze zdjęcia wymagany jest jeden neuron w warstwie wejściowej, czyli dla zdjęcia 860x640 wymagane by było 550400 neuronów, nie wspominając o warstwie ukrytej. Z racji wielkości tych liczb największą przeszkodą w używaniu sieci jest ogrom danych i mocy obliczeniowej wymagany do wytrenowania sieci. Nie są rzadkością sieci, które uczą się godzinę, dzień lub kilka dni mimo zapewnienia wystarczającej mocy obliczeniowej. Dodatkowo wymagają one wielkich zbiorów danych wejściowych, aby móc poprawnie dostosować wszystkie wagi w sieci. Mimo to sieci neuronowe cieszą się sporym zainteresowaniem wynikającym z ich skuteczności w m.in. klasyfikacji obrazów.

1.3.2. Konwolucyjne sieci neuronowe

Konwolucyjne sieci neuronowe są klasami najczęściej używanymi do analizy obrazów wizualnych, ponieważ nie wymagają one znacznego wstępnego przetwarzania danych na warstwie wejściowej sieci. Zawdzięczają to dużej liczbie tzw. warstw konwolucyjnych.

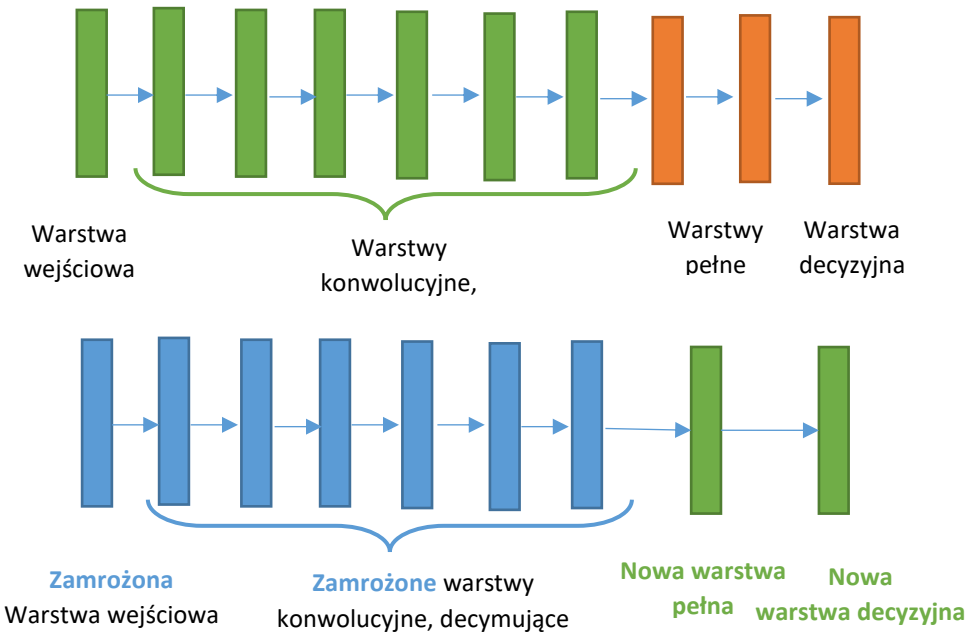


Rys. 18. Przykład struktury konwolucyjnej sieci neuronowej.

Warstwa konwolucyjna ma na celu wykrycie pewnych niskopoziomowych cech w obrazie za pomocą filtrów, które są dostosowywane w procesie uczenia sieci. Neurony z tej warstwy są zazwyczaj połączone z neuronami z funkcją aktywacji ReLU (warstwa ReLU) a następnie z warstwą decymującej, w której wydobywane są najważniejsze cechy z wcześniejszych warstw (zazwyczaj poprzez policzenie średniej lub maksymalnej wartości z neuronów z wcześniejszych warstw). Tak przetworzone informacje przekazywane są do kolejnych warstw w sieci. Może istnieć wiele bloków składających się z warstwy konwolucyjnej, ReLU i decymującej umożliwiając tym samym wykrywanie złożonych wzorców w obrazie. Na końcu każdej sieci konwolucyjnej znajdują się warstwy pełne (ang. fully-connected layer), w której wszystkie neurony z warstwy poprzedniej połączone są ze wszystkimi neuronami w warstwie pełnej. W zależności od problemu i jego złożoności może wystąpić więcej niż jedna warstwa tego typu. Ważnym jest, żeby ostatnia warstwa miała liczbę neuronów równą liczbie klas, kategorii, do których klasyfikowane będą dane. Jest to tzw. warstwa decyzyjna softmax, która oblicza dla każdego neuronu z ostatniej warstwy prawdopodobieństwo przynależności do określonej kategorii.

1.3.3. Transfer wiedzy

Zważywszy na czasochłonność uczenia konwolucyjnych sieci neuronowych, a także biorąc pod uwagę ilość wymaganych danych, często praktykowany jest transfer wiedzy, dostrajanie (ang. knowledge transfer, fine-tuning). Koncept ten wykorzystuje fakt, iż w konwolucyjnych sieciach warstwy konwolucyjne uczą się jedynie wydobywać istotne szczegóły ze zdjęć, które przydadzą się dla późniejszej klasyfikacji. Ostatnie zaś warstwy uczą się wykorzystywać wyróżnione wcześniej informacje i na tej podstawie klasyfikować dane. Dlatego też istnieje możliwość użycia przetrenowanej wcześniej sieci wykorzystując jej warstwy konwolucyjne i zastąpić wcześniejsze warstwy pełne i warstwę decyzyjną warstwami dostosowanymi do własnych potrzeb. W tym momencie warstwy konwolucyjne powinny zostać zamrożone – nie biorą udziału w procesie aktualizacji wag, uczenia, a jedynie wykrywają cechy w obrazie i przekazują je do nowo dodanych warstw, które uczą się wykorzystywać te informacje do klasyfikacji.



Rys. 19. Idea transferu wiedzy – zamiana ostatnich warstw pełnych i warstwy decyzyjnej.

Przykładowo, mając sieć klasyfikującą zwierzęta – 1000 gatunków - wytrenowaną na ogromnych ilościach danych, niemożliwych do zgromadzenia w „domowych" warunkach kopiowany jest jej model z wyłączeniem ostatnich warstw. Następnie dodaje się warstwy pełne i decyzyjną zależnie od potrzeb – przykładowo do klasyfikacji 150 gatunków zwierząt – i uczy się ostatnie warstwy aż do osiągnięcia satysfakcjonujących wyników. Wymaga to znacznie mniejszej ilości danych i czasu, a także pamięci, jako że jedyne dane wymagane do transferu wiedzy to struktura sieci neuronowej i parametry (wagi) neuronów w poszczególnych warstwach.

1.3.4. Zastosowana sieć

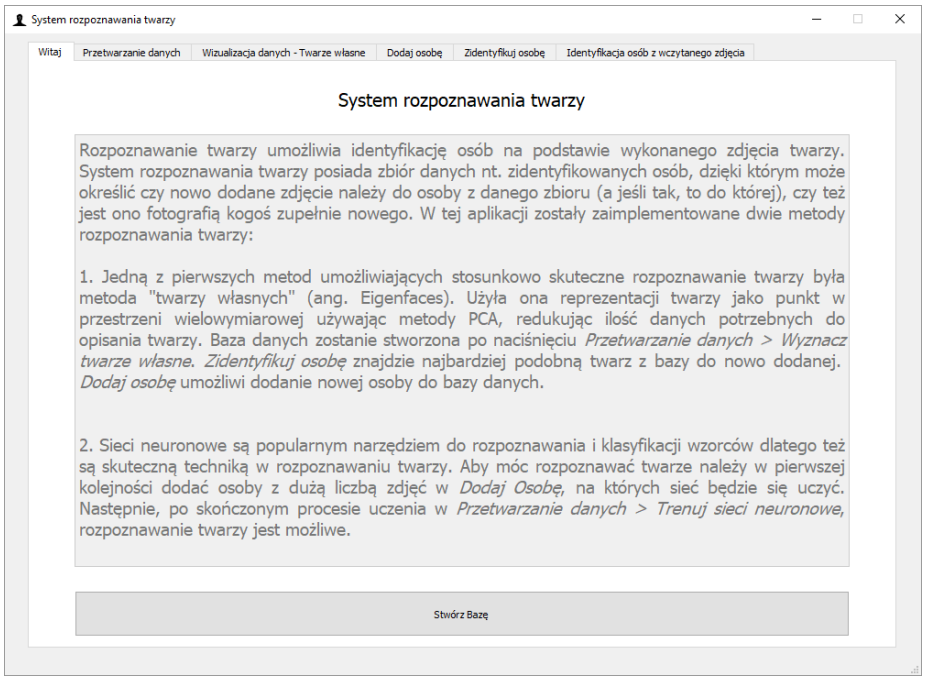
Użyta na potrzeby systemu rozpoznawania sieć to konwolucyjna sieć VGG-face, która została wytrenowana na 2.6 miliona zdjęć dla około 2600 osób i była w stanie osiągnąć precyzję w okolicach 97-98 % (więcej informacji pod: www.robots.ox.ac.uk/~vgg/software/vgg_face). Model przedstawiony jest na rysunku 20, gdzie *conv* to warstwa konwolucyjna, *maxpool* to warstwa decymująca, a *fc* to warstwa pełna.

Rys. 20. Model VGG-face i zastosowany zmodyfikowany model poniżej.

Z racji znacznie mniejszej ilości osób do rozpoznania i biorąc pod uwagę warunki wykonania zdjęć decyzja o zredukowaniu struktury i ilości neuronów w poszczególnych warstwach została podjęta. Z tego też względu trzy ostatnie warstwy zostały zamienione na warstwę pełną o 64 neuronach i warstwę pełną o 32 neuronach. Ostatnia warstwa to dalej softmax z ilością neuronów równą ilości klas (osób) do rozpoznania.

2. Aplikacja

Funkcjonalność aplikacji składa się z dwóch rozdzielnych części, którymi są zaimplementowane różne metody rozpoznawania twarzy. Każdą z metod można używać oddzielnie bądź razem bez żadnych komplikacji. Po uruchomieniu aplikacji pokazuje się okno startowe w skrócie opisujące jak posługiwać się aplikacją (Rys. 21).



Rys. 21. Okno powitalne aplikacji.

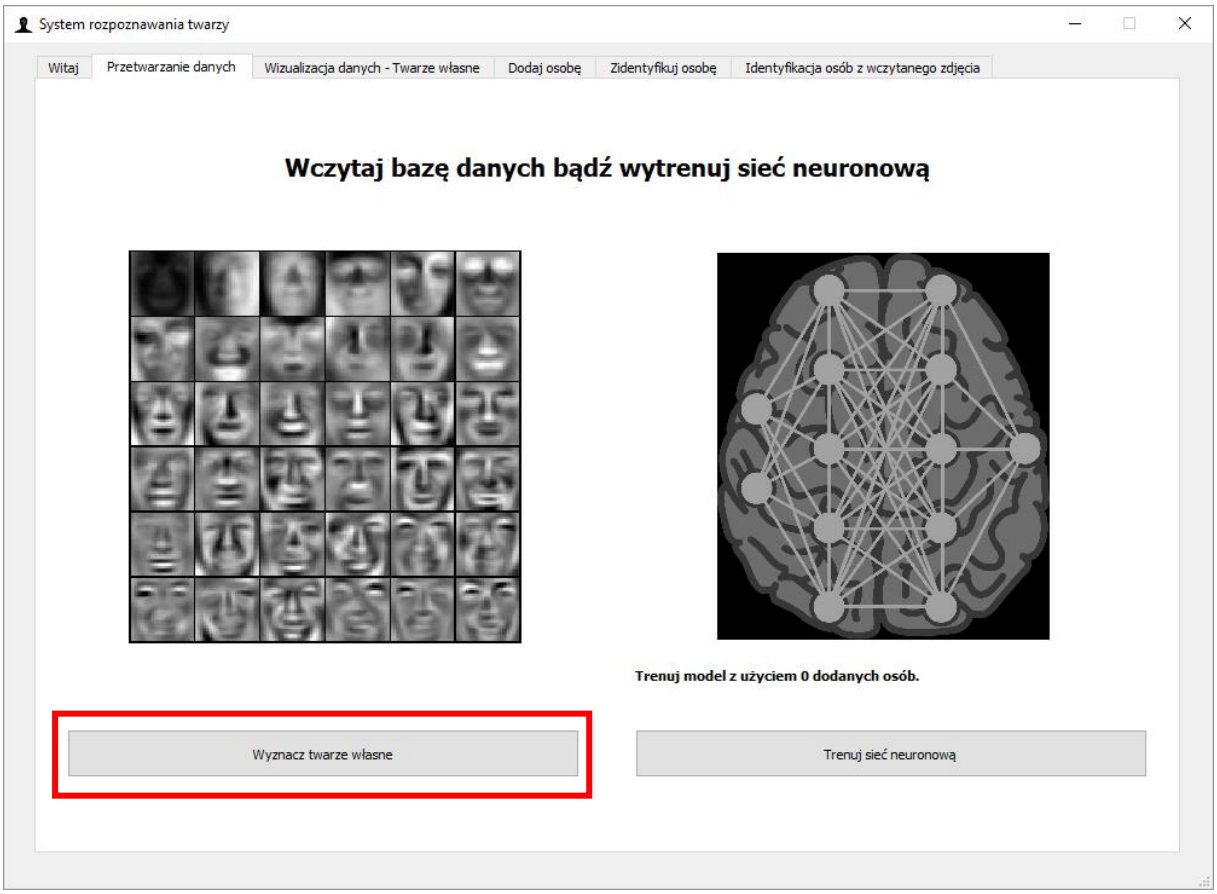
Wizualna część aplikacji jest zbudowana w formie zakładek o różnej funkcjonalności w celu ułatwienia nawigacji w programie. Do pełnego użytkowania aplikacji wystarczy jedynie kontroler jakim jest myszka – klawiatura nie jest niezbędna. Dodatkowo w aplikacji zostały przewidziane błędy w kolejności wykonywania czynności wraz z stosownym komunikatem wyświetlanym w wyskakujących oknach. W trakcie używania aplikacji wskazanym jest aby:

- wyłącznie jedna osoba była w obiektywie kamery aby nie wprowadzić błędnych danych (zdjęć) innej osoby. Jeśli nie jest to możliwe – ważne aby osoba, którą chcemy dodać do bazy danych, była najbliżej kamery. Ma to znaczenie, ponieważ aplikacja oznacza wszystkie wykryte twarze, ale dodaje tylko największą, najbliższą z nich. Każda z osób, klas, powinna być dodawana tylko i wyłącznie raz.
- W przypadku problemów z powolnym działaniem aplikacji pomóc może używanie tylko jednej z metod (twarze własne bądź sieci neuronowe). Gdy dana metoda zostanie w pełni przetestowana należałoby zamknąć aplikację i uruchomić ją ponownie, tym razem używając innej z metod.

1.1. Twarze własne

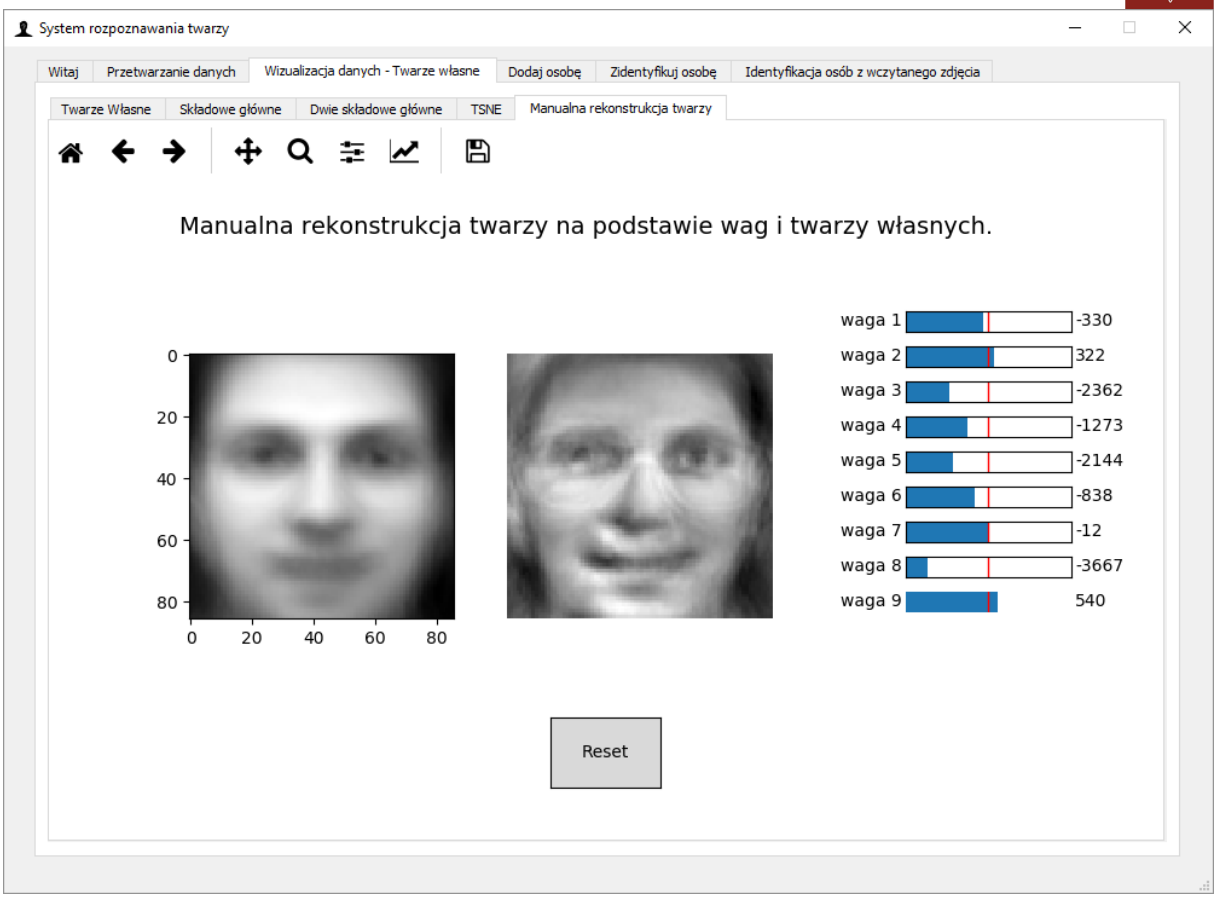
1.1.1. Wyznaczanie twarzy własnych

W celu wyznaczenia twarzy własnych należy przejść do zakładki *Przetwarzanie danych* i nacisnąć *Wyznacz twarze własne* (Rys. 22). Spowoduje to wczytanie wcześniej przygotowanej bazy danych osób i wyznaczenie twarzy własnych jak opisano w punktach 1.2.1-1.2.5.



Rys. 21. Wyznaczanie twarzy własnych.

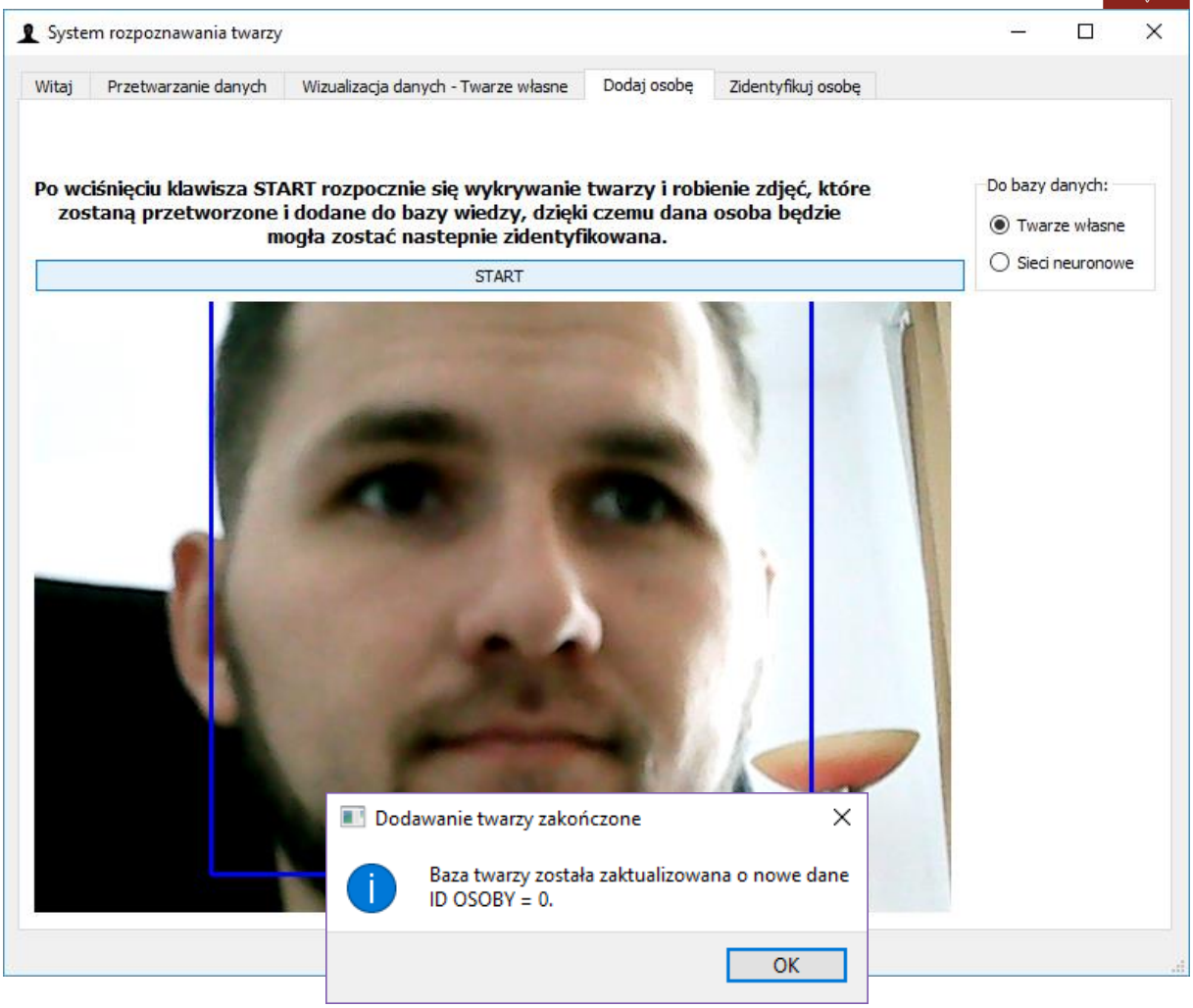
Po wyznaczeniu twarzy własnych w zakładce *Wizualizacja danych – Twarze własne* można zaobserwować wcześniej prezentowane w dokumentacji wykresy i wizualizacje takie jak pierwsze dziewięć twarzy własnych, składowe główne i procent wariancji, rzutowanie danych na dwie pierwsze składowe główne i wizualizacja danych, t-SNE a także *Manualną rekonstrukcję twarzy*. Umożliwia ona rekonstrukcję twarzy z użyciem pierwszych 9 wag i pierwszych 9 twarzy własnych. Jako obraz bazowy zaprezentowana jest twarz średnia a suwakami dopasowywane są wartości wag. W razie potrzeb przycisk *Reset* resetuje całe okno. Okno to przedstawione jest na rys. 22.



Rys. 22. Manualna rekonstrukcja twarzy.

1.1.2. Dodawanie osób do bazy danych

W celu dodaniu nowej osoby do bazy danych do obliczenia twarzy własnych należy udać się do zakładki *Dodaj osobę* a następnie, upewniając się, że *Twarze własne* są zaznaczone, należy nacisnąć *START*. Spowoduje to wykonanie serii zdjęć dla wykrytych twarzy, które to zdjęcia zostaną od razu przetworzone i dodane do bazy jako nowa osoba. Obraz z kamery jest wyświetlany w czasie rzeczywistym i zapewnia krótkie odstępy pomiędzy dodanymi zdjęciami, aby uniemożliwić dodanie zdjęć, które są zbyt podobne do siebie (rys. 23). Należy wspomnieć, że nie wskazane jest dodawanie tej samej osoby więcej niż raz z racji tego, że każda nowo dodana osoba dodawana jest z inną etykietą. W sytuacji ponownego dodania jednej osoby oczywiście aplikacja znajdzie najbardziej podobną osobę, ale jako że będą dwie klasy osób ze zdjęciami jednej twarzy – nie określi jej jako będącą w bazie/niemożność identyfikacji.



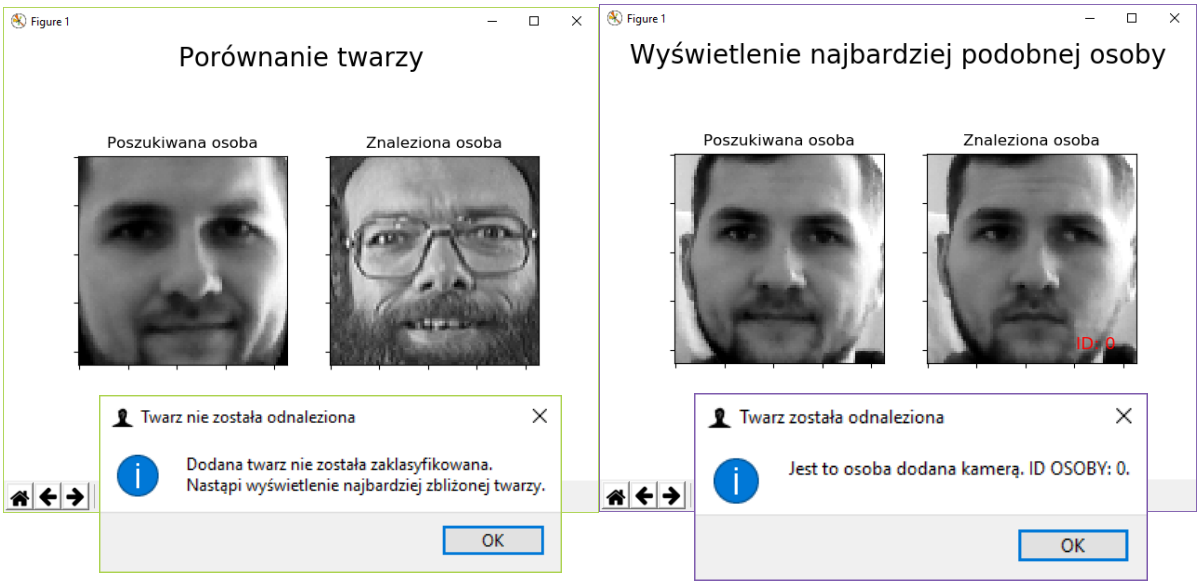
Rys. 23. Dodawanie twarzy z użyciem aplikacji.

Po każdorazowym dodaniu nowej osoby wyświetli się przypisane do niej unikatowe ID, które nastepnie będzie wyświetlane w przypadku poprawnej identyfikacji osoby. Należy pamiętać, że ID przypisywane do osób/klas w różnych metodach (twarze własne/sieci) są różne i niezależne!

1.1.3. Identyfikacja osób

Identyfikacja osób jest dostępna od czasu wyznaczenia twarzy własnych z przygotowanej wcześniej bazy danych po wykonaniu akcji *Wyznacz twarze własne* w *Przetwarzanie danych*. Oczywiście bez dodania własnego zdjęcia zostanie się jedynie porównanym do zdjęć obecnych w bazie danych. Identyfikacja osób jest analogiczna do dodawania osób z tą różnicą, że wykonywane jest jedno zdjęcie (po 1-2 sekundach na przygotowanie) i zostanie wyświetlony wynik identyfikacji. Tak jak zostało to opisane w 1.2.8 twarz zostanie uznana za zidentyfikowaną bądź nową/niemożliwą do pewnej identyfikacji. Nie mniej jednak, w każdym przypadku zostanie wyświetlona najbardziej podobna osoba wg.

kryteriów algorytmu jak zaprezentowano na rys. 24. Zdjęcie wyświetlone jako ‘znaleziona osoba’ to zdjęcie z klasy uznanej jako dana osoba, które było ‘najbliższe’ w przestrzeni wielowymiarowej (odległość euklidesowa) do twarzy w ‘poszukiwana osoba’.



Rys. 23. Identyfikacja nowej twarzy (po lewo) i twarzy obecnej w bazie (po prawo).

Istnieją trzy możliwe wyniku identyfikacji:

- Znaleziono osobę w bazie danych i jest to osoba dodana kamerą – wyświetlone zostanie zdjęcie wraz z ID.
- Znaleziono osobę w bazie danych i uznano, że jest to osoba z początkowej, wczytywanej bazy danych – wyświetlenie samego zdjęcia, gdyż ID jest przypisywane jedynie do osób dodanych kamerą.
- Twarz nie została odnaleziona – zostanie wyświetlona najbardziej podobna twarz.

1.1.4. Warto wiedzieć

Precyzja metody twarzy własnych jest bardzo podatna na zdjęcia twarzy zrobione w różnej orientacji i w różnym oświetleniu. Zdjęcie twarzy zrobione z lewego profilu i zdjęcie twarzy zrobione z prawego profilu będą mieć stosunkowo różne reprezentacje twarzy. Wynika to m.in. z tego, iż pierwsze składowe główne dla zdjęć o różnych orientacjach mogą być związane z oświetleniem z lewej bądź prawej strony, a także elementy twarzy mogą być widoczne w inny sposób. Przykład takich zdjęć widoczny jest na rysunku 24.

Rys. 24. Zdjęcia tej samej klasy wykonane z różnych perspektyw.

1.2. Konwolucyjne sieci neuronowe

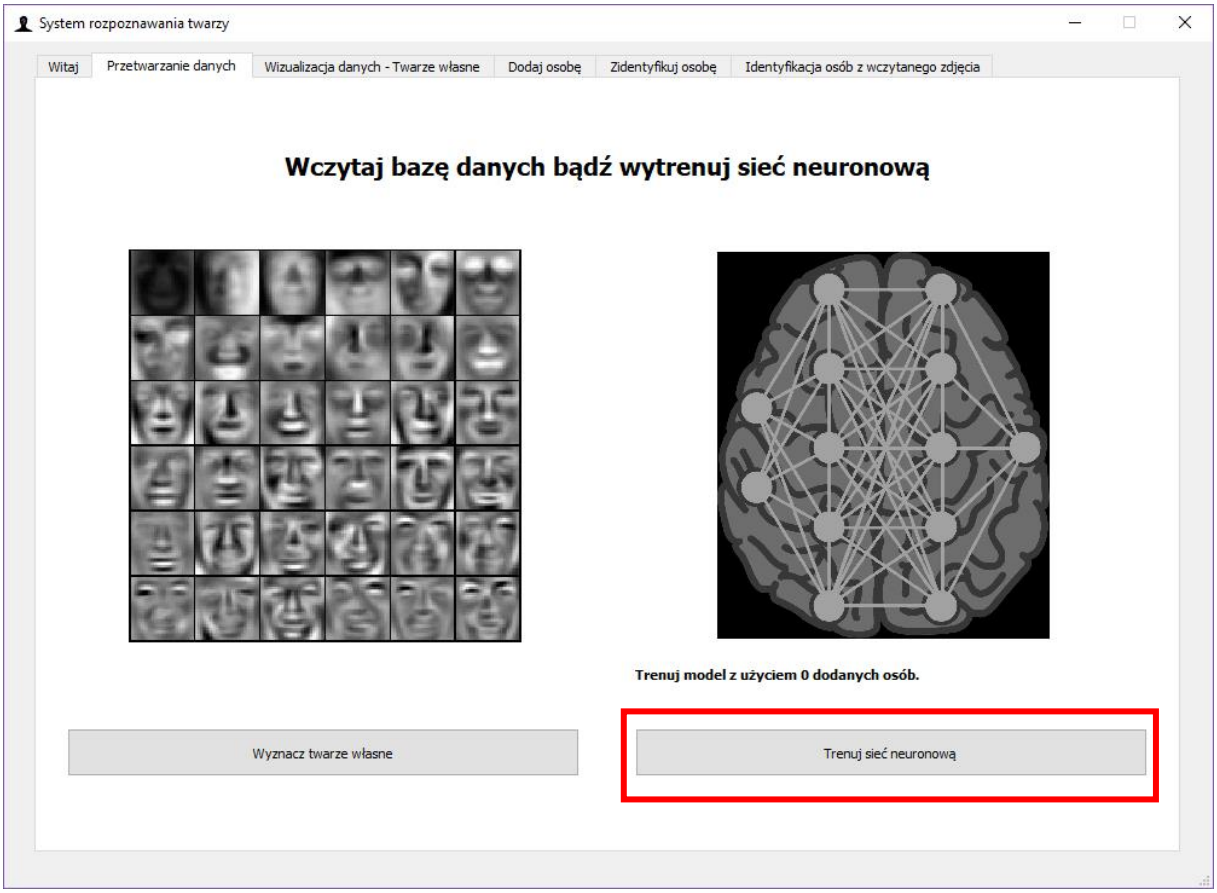
1.2.1. Trening sieci

Proces dodawania czy identyfikacji osób nie różni się w żadnym stopniu w użytkowaniu z punktu widzenia interfejsu graficznego jednakże jak wcześniej opisano w punkcie 1.3 metody użyte są całkowicie inne. Dodatkowo, obliczenia na procesorze są zbyt czasochłonne (niemniej jednak możliwe) i dlatego też stanowisko z kartką graficzną z CUDA powinno być przygotowane do treningu sieci. W przypadku sieci neuronowych należy:

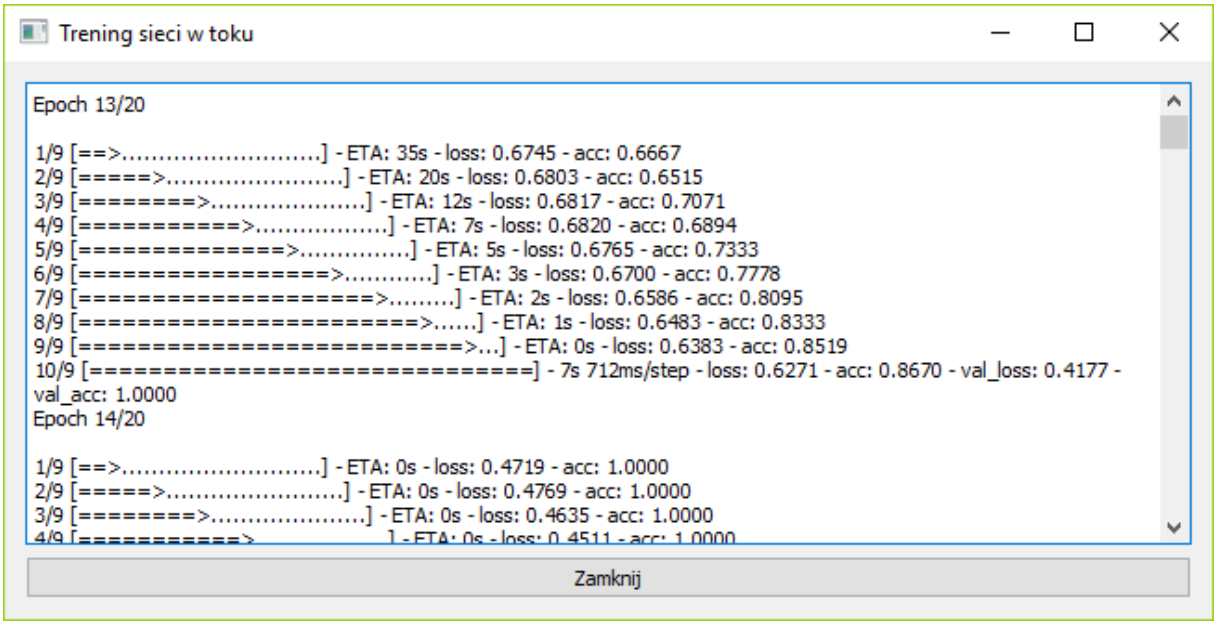
- Dodać osoby (co najmniej dwie) w zakładce *Dodaj osobę* zaznaczając uprzednio *Sieci neuronowe.*
- Wytrenować sieć co jest możliwe w zakładce *Przetwarzanie danych*, która kontroluje na bieżąco ilość osób przygotowanych do treningu sieci (rys 24). Sieć można trenować wielokrotnie, nawet po dodaniu nowych osób po treningu sieci, jednakże nie jest to dotrenowanie sieci – nie aktualizuje się wcześniej wytrenowanych parametrów, a trenuje się cały model (wyłączając warstwy konwolucyjne z transferu wiedzy) na nowych zdjęciach od nowa.

  W trakcie treningu wyświetlone zostanie okno z informacjami dotyczącymi pozostałej liczby cykli do wytrenowania sieci, na której można także śledzić precyzję algorytmu w trakcie uczenia (rys. 25).

  Trening może zostać ukończony przed maksymalną ilością epok gdy precyzja sieci nie wzroście o więcej niż 1 % po 5 epokach.
- Po treningu sieci identyfikacja osób z użyciem sieci neuronowej jest możliwa. Jest ona analogiczna do identyfikacji z użyciem twarzy własnych z tym wyjątkiem, że wyświetli ona najlepszym zdaniem sieci osobę bez informacji dot. Tego czy twarz jest nowa czy istniejąca w bazie, gdyż nie ma ona takiej funkcjonalności.



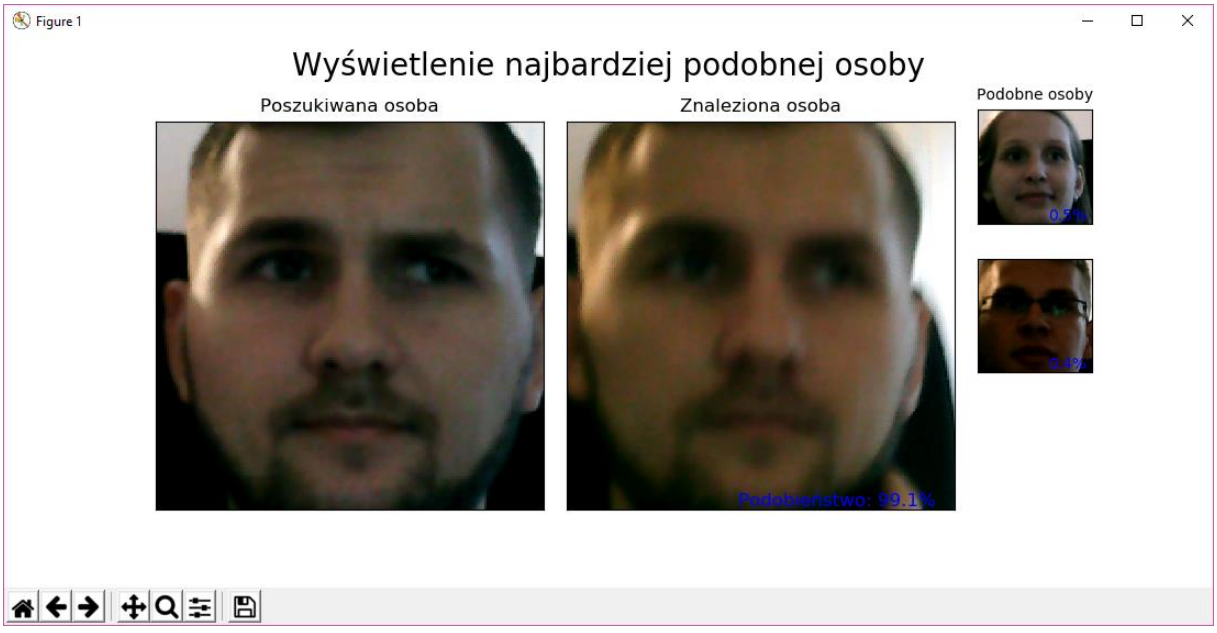
Rys. 25. Trening sieci neuronowej.



Rys. 26. Okno prezentujące stan uczenia się sieci w trakcie treningu.

1.2.2. Identyfikacja osób

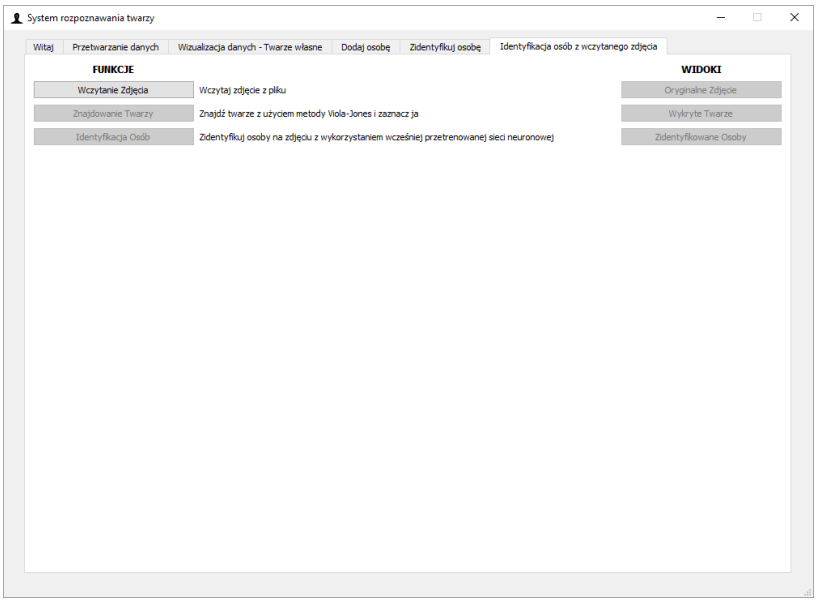
Proces identyfikacji osób jest identyczny jak w przypadku twarzy własnych, jednakże różnicą jest prezentacja i znaczenie wyników. W przypadku twarzy własnych zostały stworzone kryteria według których można było stwierdzić przynależność danego zdjęcia do jakiejś klasy bądź występowanie klasy w bazie danych. W przypadku sieci sytuacja jest inna, gdyż ostatnią warstwą sieci jest wcześniej wspomniana warstwa Softmax, która zwraca prawdopodobieństwo przynależności do określonej klasy. Z tego też względu o ile istniałaby możliwość stworzenia kryterium gdzie przykładowo 95 % prawdopodobieństwa przynależności oznaczałoby iż klasyfikacja jest poprawna, jednakże nie jest to niezbędne w przypadku stworzonego systemu. Przykład okna identyfikacji widoczny jest na rysunku 27. Zależnie od ilości dodanych osób w oknie może być widocznych do 4 kandydatów z wartościami procentowymi prawdopodobieństw. W przypadku twarzy własnych jako zdjęcie reprezentujące daną klasę przy identyfikacji było wyświetlane „najbliższe" zdjęcie – jednakże w przypadku sieci nie ma takiej możliwości. Z tego powodu jako reprezentacja danej klasy zawsze wybierane jest 4 zdjęcie wykonane podczas *Dodaj osobę*.



Rys. 27. Przykład identyfikacji twarzy z użyciem sieci neuronowej.

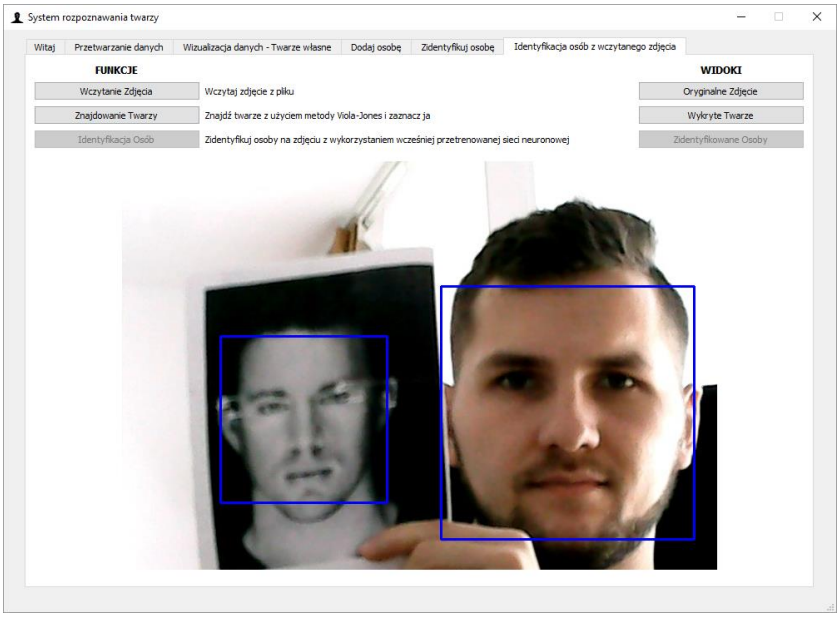
1.2.3. Identyfikacja osób z wczytanego zdjęcia

Możliwa jest również detekcja twarzy oraz identyfikacja osób z użyciem wcześniej przetrenowanej sieci w zakładce *Identyfikacja osób z wczytanego zdjęcia* widocznej na rysunku 28.



Rys. 28. Zakładka w aplikacji służąca do identyfikacji osób z wczytanego zdjęcia.

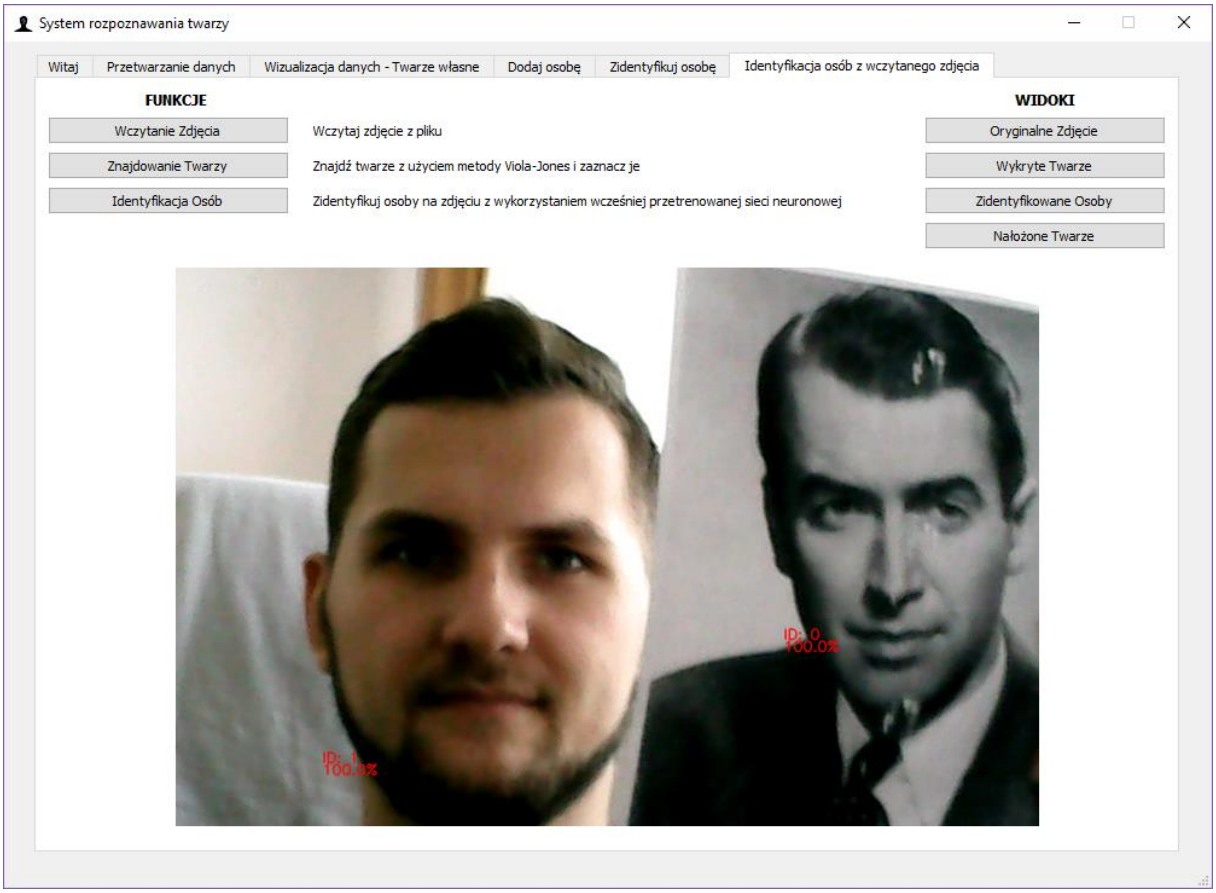
Po użyciu *Wczytanie Zdjęcia* uruchomi się panel do wyboru pliku zdjęciowego. Następnie możliwa jest detekcja twarzy po naciśnięciu *Znajdowanie Twarzy* – twarze wykryte zostaną oznaczone niebieskim prostokątem jak widać na rysunku 29.



Rys. 29. Zakładka w aplikacji służąca do identyfikacji osób z wczytanego zdjęcia.

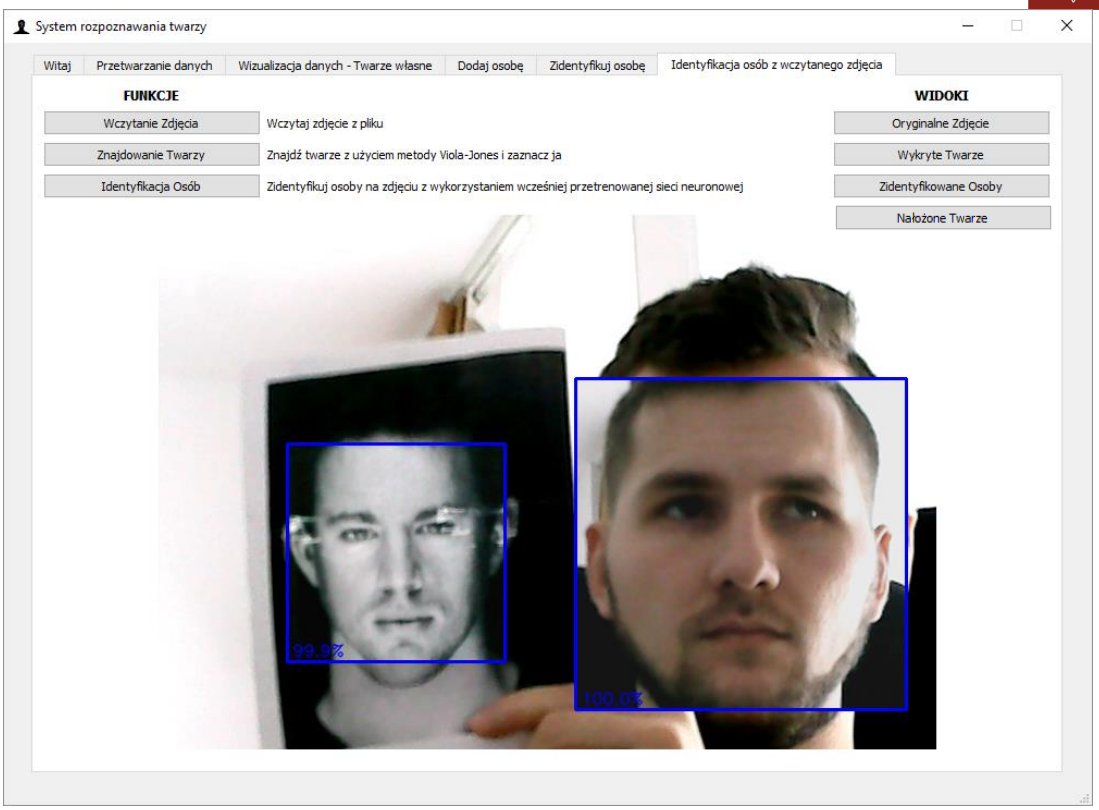
Dodatkowo, w przypadku potrzeby zmiany widoku wszystkie wcześniejsze widoki są dostępne po prawej stronie okna. Jeśli sieć została wcześniej przetrenowana możliwe będzie naciśnięcie *Identyfikacja Osób*, która umożliwi identyfikację na podstawie wcześniej dodanych osób. Nowa osoba na zdjęciu nie będąca w bazie naturalnie nie będzie mogła zostać wykryta poprawnie. Po naciśnięciu przycisku identyfikacji sieć wyznaczy najbardziej podobnych kandydatów.

W widoku *Zidentyfikowane Osoby* przy wykrytych twarzach pojawi się wartość procentowa prawdopodobieństwa wraz z ID osoby uznanej za osobę na zdjęciu. Widok ten jest zobrazowany na rysunku 30.



Rys. 30. Zakładka w aplikacji służąca do identyfikacji osób z wczytanego zdjęcia – widok *Zidentyfikowane Osoby*.

W przypadku widoku *Nałożone Twarze* twarze wyznaczonych kandydatów nadpiszą wcześniejsze twarze znajdujące się na zdjęciu. (Twarz wyświetlona jest zawsze piątą twarzą dodaną przy *Dodaj osobę* przy używaniu sieci neuronowych.) Wynik identyfikacji wraz z wartościami procentowymi obrazuje rysunek 31.



Rys. 31. Zakładka w aplikacji służąca do identyfikacji osób z wczytanego zdjęcia – widok *Nałożone Twarze*.