

Identyfikacja twarzy z wykorzystaniem Sztucznych Sieci Neuronowych oraz PCA

Michał Pieróg
pierogmichal@gmail.com

Jakub Jaśkowiec
qbajas@gmail.com

Abstrakt— Identyfikacja twarzy jest zadaniem polegającym na weryfikacji tożsamości danej osoby z wykorzystaniem zdjęć jej twarzy. Proces identyfikacji może być użyty w różnych biometrycznych systemach bezpieczeństwa. W niniejszym artykule przedstawiono proces identyfikacji twarzy wykorzystujący sztuczną sieć neuronową typu Feed Forward oraz statystyczną metodę PCA (Principal Component Analysis). Zdjęcia wejściowe zostały poddane wstępnej filtracji i prostej segmentacji, następnie zredukowano rozmiar danych przy użyciu PCA. Kolejnym krokiem była klasyfikacja przy użyciu sieci neuronowej. Przeprowadzono testy z różnymi parametrami PCA oraz sieci, uzyskując stosunkowo dużą skuteczność identyfikacji dla zbioru testowego, składającego się z 2200 zdjęć w skali szarości.

Słowa kluczowe- *identyfikacja twarzy; Sztuczne Sieci Neuronowe; PCA; Back Propagation; Resilient Propagation*

I. WSTĘP

Burzliwy rozwój techniki umożliwił znaczący postęp w dziedzinie bezpieczeństwa. Dlatego coraz większą uwagę naukowców przyciągają biometryczne systemy identyfikacji, szczególnie te oparte na automatycznym rozpoznawaniu twarzy. Identyfikacja osób na podstawie twarzy jest jedną z ważniejszych gałęzi w przetwarzaniu obrazów. Prawdziwym wyzwaniem jest stworzenie automatycznego systemu, który mógłby dorównać ludzkiej zdolności rozpoznawania twarzy.

Nie poznano dokładnie mechanizmu pozwalającego ludziom w szybki sposób rozpoznawać twarze. Chociaż ludzie radzą sobie bardzo dobrze z rozpoznawaniem znanych twarzy, nie są oni w stanie poradzić sobie z dużymi ilościami twarzy nieznanymi. Komputery z ogromną pamięcią i mocą obliczeniową mogłyby przewyższyć te ograniczenia.

Głównym problemem związanym z identyfikacją twarzy jest sposób jej reprezentacji. Istnieją 2 główne metody identyfikacji twarzy. Algorytmy feature-based opierają się na ekstrakcji wektorów cech charakterystycznych części twarzy takich jak oczy, nos i usta wykorzystując zaawansowane modele matematyczne i elastyczne modele kształtu. Najczęściej wykorzystywanymi metodami feature-based są Elastic Bunch Graph Matching (EBGM) [1] oraz Active Appearance Model (AAM) [2].

Metody holistyczne oparte są z kolei na statystyce. Cała twarz rzutowana jest na nową przestrzeń, w której może być opisana wektorem o znacznie mniejszej długości niż w oryginalnej przestrzeni obrazu. Najczęściej wykorzystywanymi

metodami holistycznymi w systemach identyfikacji twarzy są Principal Component Analysis (PCA) [3] oraz Linear Discriminant Analysis (LDA) [4].

II. PROPONOWANE ROZWIĄZANIE

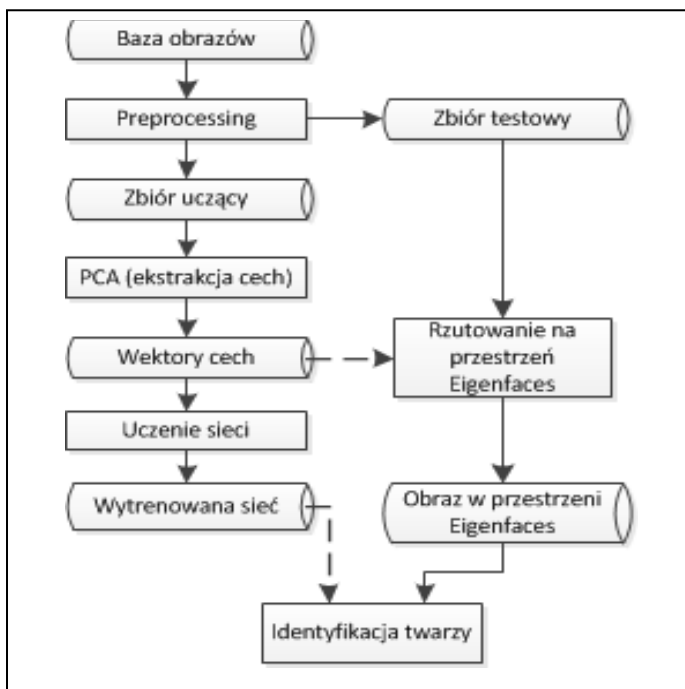
W proponowanym systemie skupiono się wyłącznie na etapie identyfikacji twarzy. Założono, że zarówno zdjęcia znajdujące się w bazie danych jak i zdjęcia do identyfikacji zostały poddane obróbce i posiadają odpowiednie parametry. Podany proces należy rozpatrywać, jako komponent pewnego, bardziej złożonego systemu rozpoznawania twarzy.

Pierwszy etap identyfikacji polega na wstępnym przetworzeniu obrazów. Jego celem jest przeskalowanie i przycięcie zdjęć do odpowiednich rozmiarów oraz częściowe usunięcie szumów. Preprocessing bazy danych wykonywany jest w trybie offline, zdjęcia przygotowywane są z wyprzedzeniem, zanim zostaną wykorzystane. Zdjęcia, które będą poddane identyfikacji również muszą przejść wcześniej etap preprocessingu, nie jest on wykonywany w trakcie identyfikacji, gdyż wydłuża czas działania systemu i wprowadza dodatkowe utrudnienia.

Kolejnym krokiem jest przygotowanie danych dla sztucznej sieci neuronowej. Zdjęcia w bazie danych poddawane są analizie PCA, która jest popularną statystyczną metodą wyszukiwania wzorców w wielowymiarowych danych. Redukcja wymiarów przy użyciu PCA jest przeprowadzana w celu zmniejszenia rozmiaru danych wejściowych, usunięcia danych nadmiarowych oraz ekstrakcji cech najlepiej opisujących twarze. Rozmiar wektora wyjściowego może być regulowany, co pozwala na ścisłą kontrolę redukcji danych. Analiza PCA jest zadaniem o dużej złożoności obliczeniowej, pamięciowej i czasowej. Wystarczające jest przeprowadzanie analizy tylko w przypadku zmiany bazy obrazów, dlatego raz obliczone współczynniki PCA są zapisywane w bazie i mogą być wykorzystane w późniejszych etapach identyfikacji, bez konieczności przeprowadzania czasochłonnych obliczeń.

Po wstępnym przetworzeniu danych następuje klasyfikacja twarzy. Wektory własne twarzy, tzw. Eigenfaces stanowią wejście dla sztucznej sieci neuronowej. Sieć jest trenowana z wykorzystaniem algorytmu Back Propagation lub Resilient Propagation. Po fazie treningu można przystąpić do identyfikacji twarzy. Każdy neuron wyjściowy z sieci odpowiada jednej osobie w bazie, wygrywa neuron o najwyższej wartości na wyjściu. Wprowadzono jednak dodatkowy parametr będący progamiem. Jeżeli wartość żadnego

neuronu nie przekroczy wartości progowej uznaje się, że wektor wejściowy nie opisuje twarzy zadanej osoby znajdującej się w bazie. Przebieg procesu identyfikacji przedstawiono na rys. 1.



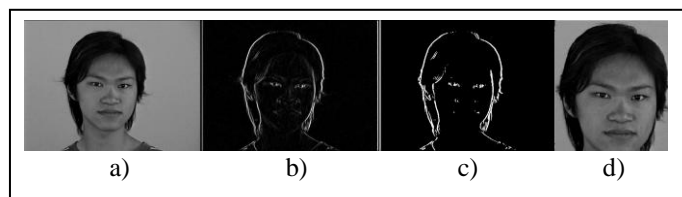
Rys. 1 Proces identyfikacji. Linie ciągłe- wejście/wyjście procesu. Linie przerywane- wykorzystanie komponentu w procesie.

A. Przetwarzanie wstępne obrazów

Zdjęcia w dostępnej bazie nie mogły być wykorzystane w systemie bez wstępnej obróbki. Były one zaszumione, a twarz nie była wyśrodkowana. Ponadto zdjęcia zawierały duży obszar tła. Czynniki te mogłyby wpłynąć negatywnie na skuteczność systemu, dlatego zdjęcia zostały poddane prostemu przetwarzaniu wstępnemu. Etap preprocessingu został zaimplementowany z wykorzystaniem bibliotek Java Advanced Imaging (JAI) [5]. W celu zredukowania szumów użyto kilkukrotnej filtracji medianowej, z małą maską. Filtr medianowy bardzo dobrze usuwa zakłócenia typu sól-pieprz oraz w bardzo małym stopniu wpływa negatywnie na krawędzie obrazu powodując tylko minimalne rozmycie obrazu [6].

Po usunięciu zakłóceń następuje przycięcie obrazu. Zdjęcia przycinane są z góry i z dołu przy użyciu stałego marginesu, dlatego nie jest to rozwiązanie uniwersalne i sprawdza się tylko dla wykorzystanej bazy. Przycięcie boków zdjęcia z wykorzystaniem stałych, ustalonych współrzędnych krawędzi było niemożliwe, gdyż twarze miały różne położenia na zdjęciach. Aby wyciąć poprawnie twarz należy ustalić jej lokalizację. Przy użyciu filtru Prewitt'a [7] wykryto krawędzie na obrazie, następnie obraz poddano binaryzacji ze stałym progiem oraz operacji morfologicznego zamknięcia [8], dzięki czemu uzyskano stosunkowo dobry kontur twarzy. Kolejnym krokiem było już wyznaczenie maksymalnej i minimalnej

współrzędnej X otrzymanego konturu, która posłużyła do przycięcia obrazu. Kolejne kroki preprocessingu przedstawiono na rys. 2.



Rys 2. Obraz w kolejnych etapach preprocessingu, a- obraz wejściowy, b- obraz przetworzony operatorem Prewitt'a, c- obraz zbinaryzowany, d- przycięty obraz wyjściowy

B. Principal Component Analysis

Principal Component Analysis (ang. Analiza Głównych Składowych) opiera się na transformacji Karhunen-Loévego [9]. Dla danego s -wymiarowego wektora wejściowego, reprezentującego twarz w zbiorze testowym, PCA poszukuje t -wymiarowej podprzestrzeni, której wektory bazowe odpowiadają kierunkom maksymalnych wariacji w przestrzeni obrazu wejściowego. Nowa podprzestrzeń cechuje się zazwyczaj znacznie mniejszymi wymiarami ($t \ll s$), co pozwala na dobrą redukcję danych wejściowych.

Analizę PCA zaimplementowano z wykorzystaniem biblioteki Efficient Java Matrix Library (EJML) [10]. Pierwszym krokiem jest obliczenie uśrednionego obrazu twarzy. Średnia jest obliczana dla każdego piksela obrazu osobno, tj.

$$P_j = \frac{\sum_{i=1}^n x_{ij}}{n}, j = 1..m \quad (1)$$

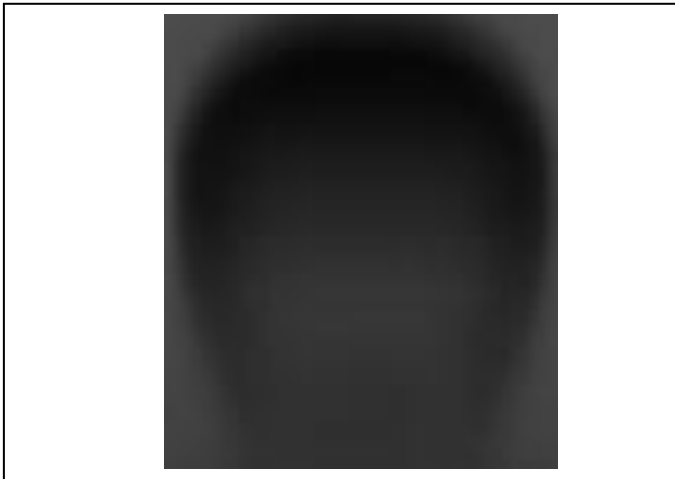
Gdzie:

- P_j – wartość piksela o numerze j
- n – liczba obrazów wejściowych
- m – rozmiar obrazu
- j – numer piksela
- i – numer obrazu
- x_{ij} – wartość piksela o numerze j w obrazie wejściowym i

Uśredniony obraz twarzy ze zbioru testowego został przedstawiony na rys. 3.

Kolejnym krokiem algorytmu jest obliczenie wektorów różnic pomiędzy obrazami wejściowymi a obrazem średnim. Z wektorów różnic tworzona jest macierz A , która podlega dekompozycji SVD [11], w wyniku, której możemy otrzymać macierz komponentów wiodących. Jest to macierz prostokątna o wymiarze $n \times m$, gdzie n jest liczbą obrazów wejściowych, a m ilością pikseli w obrazie. Możemy dokonać redukcji danych poprzez odrzucenie składowych o małym znaczeniu, odrzucając wiersze o wysokich numerach. W praktyce nie ma ograniczenia na stopień redukcji danych, jednak im więcej komponentów wiodących zostanie odrzuconych, tym mniejsza skuteczność działania systemu. Dekompozycja SVD jest najbardziej złożoną pamięciowo i obliczeniowo operacją w

systemie, co powodowało wiele problemów związanych z niewystarczającymi zasobami sprzętowymi.



Rys. 3 Twarz uśredniona

C. Sieć neuronowa

Zastosowana została sieć typu feed-forward. W tego typu sieci neurony zgrupowane są w formie warstw. Warstwy otrzymują wejście od warstwy poprzedniej i przekazują swoje wyjście do następnej warstwy. Połączenie pomiędzy neuronami tej samej lub poprzedniej warstwy nie są możliwe [12].

Dla sieci zostały ustalone następujące parametry:

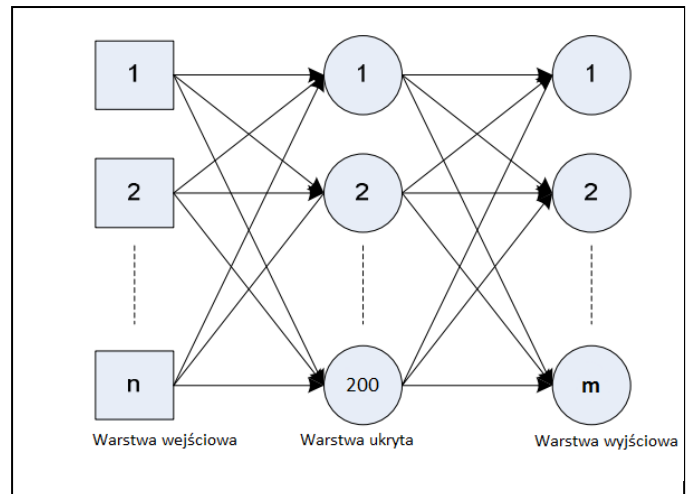
- jedna warstwa ukryta
- 200 neuronów w warstwie ukrytej
- liczba neuronów w warstwie wejściowej jest zależna od rozmiaru PCA
- liczba neuronów w warstwie wyjściowej jest równa ilości osób, które sieć na zadanie rozpoznać (dla testów przyjęto 30)

Wyjścia warstwy ukrytej i wyjściowej zostały poddane Sigmoidalnej Funkcji Aktywacji.

D. Trening sieci neuronowej

Trening sieci odbywa się przy wykorzystaniu zdjęć z bazy danych.

Sieć można trenować za pomocą dwóch metod: Resilient Propagation oraz Back Propagation. Obie metody są przykładem metod trenowania za pomocą propagacji. Taki trening polega na porównaniu wejścia systemu z pożądanym wyjściem, system jest dopasowywany dopóki różnica pomiędzy dwoma nie jest zminimalizowana. W takim treningu błędy propagowane są od warstwy wyjściowej do wyjściowej sieci. Propagacja wymaga, aby funkcja aktywacji była różniczkowalna.



Rys. 5 Architektura sieci

E. Resilient Propagation

Jest jedną z najlepszych uniwersalnych metod treningu sieci. Sprawdza się dobrze w większości przypadków. [13].

Resilient Propagation zazwyczaj daje lepsze rezultaty niż Back Propagation. Dodatkowo, metoda ta nie wymaga ustalania parametrów, które są konieczne przy Back Propagation. Wysoka skuteczność sieci uzyskiwana jest już przy niewielkiej liczbie iteracji, co obrazuje rys. 5.

Opis działania:

Działanie tej metody składa się z dwóch kroków. W pierwszym, przeliczamy wagi przy użyciu następującego wzoru:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & , \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & , \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & , \text{ w przeciwnym przypadku} \end{cases}$$

Znajdujemy zmianę wagi (delta w) w zależności od zmiany wartości (delty).

Po policzeniu wag, ustalamy nową deltę. Uzyskujemy ze wzoru:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \cdot \Delta_{ij}^{(t-1)} & , \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- \cdot \Delta_{ij}^{(t-1)} & , \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} \cdot \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)} & , \text{ w przeciwnym przypadku} \end{cases}$$

gdzie:

Symbol	Znaczenie
$\Delta_{ij}^{(t)}$	Zmiana wartości dla obecnej iteracji t.
$\Delta_{ij}^{(t-1)}$	Zmiana wartości dla iteracji t-1.
$\frac{\partial E}{\partial w_{ij}}^{(t)}$	Gradient wagi z i do j dla iteracji t.
$\frac{\partial E}{\partial w_{ij}}^{(t-1)}$	Gradient wagi z i do j dla iteracji t-1.
E	Błąd treningu
$E(t-1)$	Błąd treningu dla iteracji t-1
$\Delta w_{ij}^{(t)}$	Zmiana wagi z i do j w obecnej iteracji (t).

a. Znaczenie użytych symboli

G. Implementacja

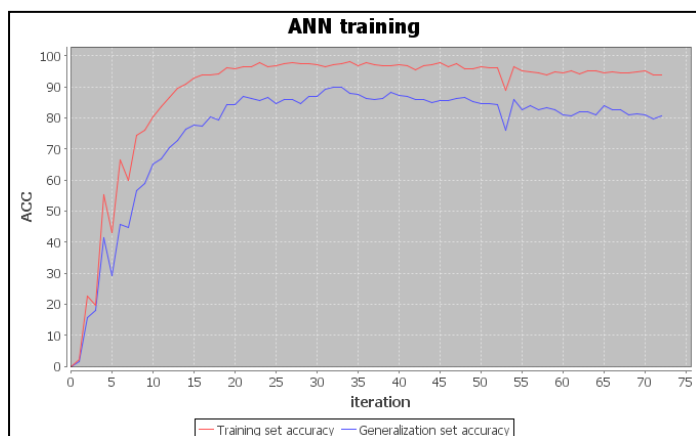
Sieć została zaprogramowana w języku programowania Java, przy użyciu biblioteki Encog [14]. Encog jest zaawansowanym frameworkiem obsługującym sieci neuronowe oraz inne algorytmy wykorzystywane w uczeniu maszynowym.

Kod odpowiedzialny za stworzenie sieci wygląda następująco:

```
network = new BasicNetwork();
network.addLayer(new BasicLayer(null, true,
inputs));
network.addLayer(new BasicLayer(new
ActivationSigmoid(), true, hidden));
network.addLayer(new BasicLayer(new
ActivationSigmoid(), false, outputs));
network.getStructure().finalizeStructure();
network.reset();
```

H. Testy sieci

Testowanie sieci odbyło się przy użyciu bazy danych ze zdjęciami twarzy dostępnej pod linkiem: [15]. Baza składała się ze zdjęć 30 osób; zdjęcia twarzy zostały zarejestrowane pod kątem od 0 do 90 stopni (z krokiem 5 stopni) z obydwu stron. Zbiór testowy był rozłączny ze zbiorem treningowym.

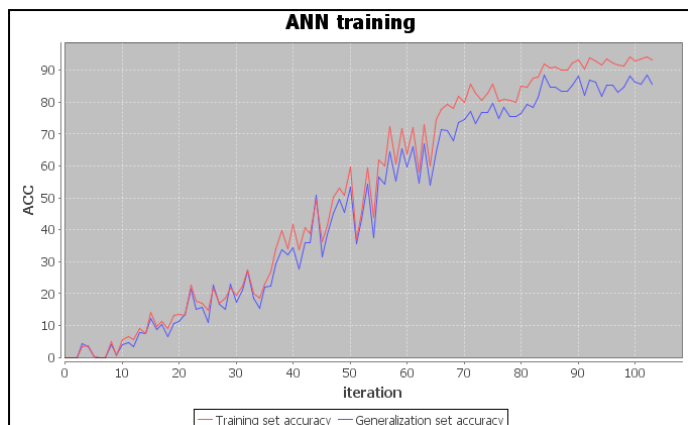


Rys. 6 Statystyki treningu dla metody Resilient propagation

F. Back Propagation

Wymaga ustalenia szybkości uczenia (learning rate) oraz pędu (momentum). Znalazienie optymalnych wartości tych parametrów może być trudne.

Przy użyciu tej metody skuteczność sieci wzrasta systematycznie, jednak wolniej (rys. 7)



Rys. 7 Statystyki treningu dla metody Back propagation



Rys. 8 Obrazy testowe

Wraz ze zwiększeniem rozmiaru PCA, czas treningu oraz skuteczność sieci wzrasta. Podobne zachowanie obserwujemy zwiększając liczbę neuronów w warstwie ukrytej.

Dodanie dodatkowej warstwy ukrytej nie zwiększyło skuteczności sieci.

Prezentowana tabela pokazuje zależność między rozmiarem PCA oraz ilością neuronów w warstwie ukrytej, a skutecznością sieci dla zbioru testowego. Sieć została trenowana metodą Resilient Propagation, wykonanych zostało 150 iteracji.

Ilość neuronów w warstwie ukrytej	Rozmiar PCA	Skuteczność sieci
50	20	56%
50	50	68%
50	100	71%
50	200	73%
50	300	78%
200	20	74%
200	50	83%
200	100	85%
200	200	88%
200	300	89%
300	20	84%
300	100	92%
300	300	89%
400	100	88%

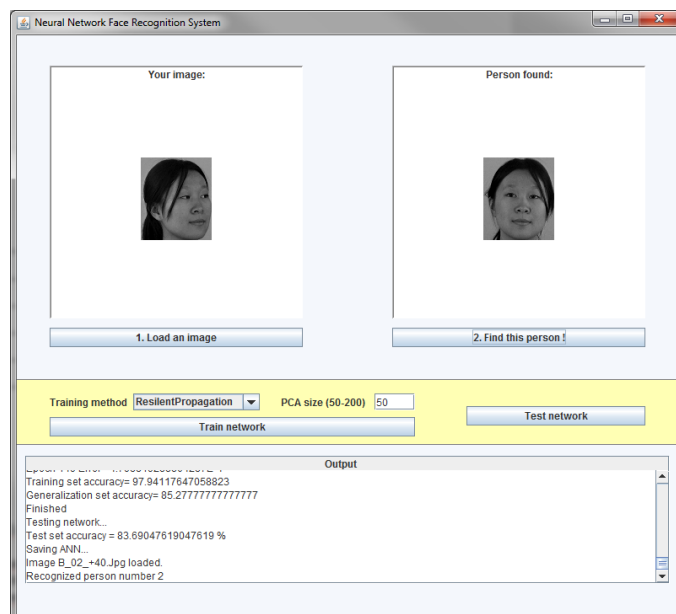
b. Wyniki testów sieci

W dalszej części procesu testowania, obrazy testowe zostały poddane modyfikacjom: została zmieniona jasność zdjęć lub wprowadzono zakłócenia. Skuteczność sieci dla takich zdjęć była zdecydowanie niższa.

1. Działanie programu

Po treningu sieci, program jest w stanie rozpoznawać twarze osób zarejestrowane pod różnymi kątami.

W pierwszym kroku ładowane jest zdjęcie osoby. Następnie po kliknięciu 'Find this person' prezentowane jest zdjęcie znalezionej osoby zarejestrowane pod kątem 0 stopni oraz na konsoli wypisywany jest odpowiedni komunikat.



Rys. 9 Rozpoznanie osoby

III. WNIOSKI

Testy wykazały wysoką skuteczność klasyfikacji sieci dla bazy testowej. Należy jednak zauważyć, że zarówno obrazy w bazie jak i obrazy wykorzystywane do testów były odpowiednio przygotowane. Gdy do obrazu zostaną dodane zakłócenia, jak np. szumy, nierównomierne oświetlenie itp. skuteczność klasyfikacji drastycznie spada. Również w przypadku przesunięcia lub obrotu twarzy na obrazie skuteczność identyfikacji była bardzo niska i wynosiła tylko kilkanaście procent. Problemy te mogą zostać rozwiązane z wykorzystaniem bardziej zaawansowanych metod preprocessingu i segmentacji. W przypadku usuwania zakłóceń typu pieprz-sól najlepszym rozwiązaniem byłoby użycie adaptacyjnego filtra medianowego, jednak szumy tego rodzaju występują rzadko na obrazach. Znacznie częściej występującym rodzajem zakłóceń są szumy o rozkładzie Gaussa. Można je w znacznym stopniu usunąć stosując filtry dolno-przepustowe, jednak filtry te powodują zniekształcenia krawędzi obrazu. Dobre efekty może dać połączenie filtracji medianowej z filtracją dolno-przepustową lub wykorzystanie filtra Weinera [16]. Ewentualne problemy spowodowane nierównomiernym oświetleniem w trakcie wykonywania zdjęcia można usunąć z użyciem filtra Top-hat [17].

W trakcie testów dużym problemem było przeprowadzenie analizy PCA. Obliczenie PCA wymaga bardzo dużych zasobów sprzętowych, system operacyjny nie mógł przydzielić odpowiedniej ilości pamięci dla maszyny wirtualnej Javy. Z tego powodu musieliśmy ograniczyć zbiór testowy do 30 osób, z 90 osób dostępnych w bazie i przeprowadzić skalowanie obrazów do rozmiarów 84x98 piksela. Pomimo tego na innych maszynach może nie być możliwe wykonanie PCA. Dodatkowo obliczenia dla 2220 zdjęć zajmowały ok. 10 min. Problem z dużą złożonością pamięciową można rozwiązać

wykorzystując metodę iteracyjną obliczania PCA [18], jednak wydłuża ona czas obliczeń. Drugą możliwością jest wykorzystanie Linear Discriminant Analysis (LDA) do redukcji danych wejściowych. Wykazuje ona lepszą skuteczność w stosunku do PCA [19], jednak również jest metodą o dużej złożoności pamięciowej i obliczeniowej.

Etap uczenia sieci neuronowej może być długotrwały w zależności od liczby neuronów. Aby zwiększyć szybkość obliczeń można wykorzystać wsparcie sprzętowe. Biblioteka Encog dostarcza odpowiednich algorytmów wspieranych sprzętowo.

Skuteczność klasyfikacyjną sztucznej sieci neuronowej oceniamy jako bardzo dobrą, kluczowe znaczenie w ogólnej efektywności systemu ma etap przetwarzania wstępnego oraz redukcji danych.

BIBLIOGRAFIA

- [1] L. Wiskott, J.M. Fellous, N. Kruger, C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," Intelligent Biometric Techniques in Fingerprint and Face Recognition, eds. L.C. Jain et al., publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396, (1999).
- [2] T.F. Cootes, K. Walker, C.J. Taylor, "View-Based Active Appearance Models," IEEE International Conference on Automatic Face and Gesture Recognition, 26-30 March 2000, pp. 227-232.
- [3] M.A. Turk, A.P. Pentland, "Face Recognition Using Eigenfaces," IEEE Conference on Computer Vision and Pattern Recognition, 3-6 June 1991, pp. 586-591
- [4] K. Etemad, R. Chellappa, "Discriminant Analysis for Recognition of Human Face Images," Journal of the Optical Society of America A, Vol. 14, No. 8, August 1997, pp. 1724-1733
- [5] R. Gonzalez and R. Woods, "Digital image processing," 2nd edition, Prentice Hall, 2002. Chap 4 Sec 4.3, 4.4; Chap 5 Sec 5.1 – 5.3
- [6] <http://java.sun.com/javase/technologies/desktop/media/jai/>
- [7] R. Maini and H. Aggarwa, "Study and Comparison of Various Image Edge Detection Techniques," International Journal of Image Processing (IJIP), vol. 3
- [8] R. Haralick and L. Shapiro, "Computer and Robot Vision," Vol. 1, Addison-Wesley Publishing Company, 1992, pp 174 - 185.
- [9] A. Levy and M. Lindenbaum, "Sequential Karhunen–Loeve Basis Extraction and its Application to Images," IEEE Transactions on image processing, Vol.9, No. 8
- [10] <http://code.google.com/p/efficient-java-matrix-library/>
- [11] J. Demmel and W. Kahan (1990). "Accurate singular values of bidiagonal matrices," Journal on Scientific and Statistical Computing
- [12] A. Eleyan and H. Demirel, "PCA and LDA Based Face Recognition Using Feedforward Neural Network Classifier," Springer-Verlag Berlin Heidelberg 2006
- [13] http://www.heatonresearch.com/wiki/Resilient_Propagation
- [14] <http://www.heatonresearch.com/encog>
- [15] http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm#Our_Database_
- [16] N. Shimkin, Estimation and Identification in Dynamical Systems Lecture Notes," Technion, Israel Institute of Technology, Department of Electrical Engineering
- [17] <http://www.mathworks.com/help/toolbox/images/ref/imshow.html>
- [18] S. Roweis "EM Algorithms for PCA and SPCA," Advances in Neural Information Processing Systems, MIT Press, 1998.
- [19] A. M. Martinez, A. C. Kak, "PCA versus LDA,"