

ci2.3.429Filtracja krawędzi o małym gradiencie jasnościsubsection.2.3.4

Politechnika Śląska  
Wydział Automatyki, Elektroniki i Informatyki  
kierunek: informatyka

Mateusz Trzeciak

# **Określenie wieku twarzy na podstawie tekstury**

praca dyplomowa magisterska

promotor: dr hab. inż. Karolina Nurzyńska

Gliwice, październik 2019



załącznik nr 2 do zarz. nr 97/08/09

## Oświadczenie

Wyrażam zgodę / Nie wyrażam zgody\* na udostępnienie mojej pracy  
dyplomowej / rozprawy doktorskiej\*.

Gliwice, dnia 19 października 2019

.....  
(podpis)

.....  
(poświadczenie wiarygodności  
podpisu przez Dziekanat)

\* podkreślić właściwe



## Oświadczenie promotora

Oświadczam, że praca „Określenie wieku twarzy na podstawie tekstury” spełnia wymagania formalne pracy dyplomowej magisterskiej.

Gliwice, dnia 19 października 2019

.....

(podpis promotora)



# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Metoda bazowa - wrinkle feature</b>	<b>3</b>
2.1	Metoda wykrywania twarzy . . . . .	5
2.1.1	Konwersja do skali szarości oraz przestrzenie barw . . .	10
2.1.2	Algorytm Haar Cascade . . . . .	13
2.2	Wyznaczanie stref . . . . .	20
2.3	Wykrywanie zmarszczek - detektor Canny . . . . .	26
2.3.1	Redukcja szumów na obrazie filtrem Gaussa . . . . .	27
2.3.2	Szukanie gradientów jasności . . . . .	28
2.3.3	Zastosowanie techniki pocieniania krawędzi . . . . .	29
2.3.4	Filtracja krawędzi o małym gradiencie jasności . . . . .	29
2.3.5	Filtracja poprzez histerezę . . . . .	30
2.4	Wyliczanie wrinkle feature . . . . .	30
2.5	Algorytm trenowania . . . . .	31
2.6	Grupowanie danych - FCM . . . . .	31
2.6.1	Wstęp do grupowania danych . . . . .	31
2.6.2	Metoda FCM . . . . .	31
2.7	Wyznaczanie wieku . . . . .	31
<b>3</b>	<b>Modyfikacje metody bazowej</b>	<b>33</b>
3.1	Odjęcie wybranej strefy . . . . .	33
3.1.1	Zmiana algorytmu względem metody bazowej . . . . .	33
3.2	Zastosowanie metody HOG . . . . .	33
3.2.1	Opis algorytmu HOG . . . . .	33

3.2.2	Zastosowanie w projekcie . . . . .	33
3.3	Metoda HOG oraz grupowanie KNN . . . . .	33
3.3.1	Grupowanie KNN . . . . .	33
3.3.2	Zastosowanie w projekcie . . . . .	33
<b>4</b>	<b>Badania</b>	<b>35</b>
<b>5</b>	<b>Podsumowanie</b>	<b>37</b>



# Rozdział 1

## Wstęp

Wiek jest cechą, którą niełatwo człowiekowi odczytać z czyjejś twarzy. Dla komputera rozpoznawanie wieku jest trudniejsze niż dla człowieka. Dlatego do wyznaczania wieku z pomocą programu komputerowego należy podchodzić z dystansem. Mimo trudności programiści i naukowcy udoskonalają algorytmy, tak aby ocena wieku danej osoby była coraz dokładniejsza.

Istnieje wiele sposobów wyznaczania wieku. Większość metod skupia się na analizie tekstury twarzy. Idąc dalej - z obrazu danej osoby lub jego części, np. tułowia, musi zostać wykryta twarz. Wykrycie twarzy na teksturze jest możliwe dzięki algorytmom rozpoznawaniu obrazu. Rozpoznawanie obrazu jest stosowane w wizji komputerowej i polega na wyodrębnieniu z obrazu jakichś szczegółów. Mogą to być osoby, pojazdy, przedmioty itp. (Rys. 1.1)

Można znaleźć wiele witryn internetowych, które udostępniają interfejsy programistyczne umożliwiające zaimplementowanie rozpoznawania wieku z obrazu. Istnieją algorytmy przetwarzania obrazu, które oprócz wieku wyznaczają z pewnym prawdopodobieństwem płeć danej osoby. Oprócz płci mogą one także wyznaczyć mine oraz czy dana osoba nosi okulary.

Z weryfikacją wieku danej osoby można się spotkać przed wejściem do niektórych miejsc, tj. klub nocny. Większość osób musi okazać ważny dowód osobisty, co generuje duże kolejki do wejścia. Aplikacje analizujące wiek na podstawie obrazu twarzy z kamery przed wejściem do takich miejsc znacząco usprawniłyby weryfikację wieku. Rozpoznawanie wieku może być wykorzy-



Rysunek 1.1: Przykład rozpoznawania obiektów na zdjęciu ulicy. [16]

stywane przy analizie średniego wieku ludzi w jakimś miejscu np. podczas demonstracji.

Wiele gier posiada treści nieodpowiednie dla młodszych użytkowników. Możliwe jest stosowanie technologii wykrywania wieku użytkownika przed udostępnieniem mu treści, która wymaga odpowiedniego wieku.

Można znaleźć o wiele więcej potencjalnych zastosowań przetwarzania obrazu oraz rozpoznawania wieku na podstawie tekstury (obrazu) twarzy. Z biegiem lat z pewnością będzie można zauważyć dalszy rozwój tej dziedziny, która opiera się w głównej mierze na sztucznej inteligencji [22].

## Rozdział 2

# Metoda bazowa - wrinkle feature

Istnieje wiele metod wyznaczania wieku z obrazu twarzy. W literaturze spotkano rozwiązania, w których wyznaczany jest konkretny wiek osoby przez algorytm lub przedział wiekowy. Jedną z pierwszych metod szacowania wieku opierała się na wyznaczaniu proporcji twarzy, a następnie na detekcji i interpretacji zmarszczek. Była ona w stanie ze stu procentową poprawnością wyznaczyć czy dana osoba jest osobą dorosłą lub dzieckiem [25].

W kolejnych latach algorytmy i techniki szacowania wieku były udoskonalane. Badano wpływ starzenia się osób na wygląd skóry. Oprócz naturalnych zmian skóry pod wpływem starzenia się skóry należało uwzględnić także inne czynniki. Takimi czynnikami są min. płeć, poziom stresu, ekspozycja na działanie środowiska zewnętrznego. Powyższe metody zastosowano w pracy „Toward automatic simulation of aging effects on face images” autorstwa A. Lanitis, Ch. J. Taylor oraz T. F. Cootes [1]. Należy dodać, że w powyższej pracy stosowano trenowanie zbioru zdjęć. Trenowanie polega na wykryciu relacji pewnych cech twarzy do wieku osób.

W kolejnych latach pojawiło się podejście porównywania cech twarzy tej samej osoby w różnym wieku. Różnice w powyższych cechach posłużyły do zbudowania statystyki zmian cech twarzy wraz ze starzeniem się. Powyższe podejście zostało zaprezentowane w pracy „Face verification across age

progression” autorstwa N. Ramanathan oraz R. Chellappa [15].

Rozwinięciem tego pomysłu była praca „Automatic age estimation based on facial aging patterns” autorstwa X. Geng, Z. Zhou i K. Smith-Miles [24]. W tej pracy porównywano sekwencje wielu zdjęć twarzy jednej osoby. Zdjęcia przedstawiały twarz w różnym wieku. Powyższe badania pozwoliły na zbudowanie wzorca starzenia się twarzy.

Praca „A new algorithm for age recognition from facial images” autorstwa M.M. Dehshibi oraz A. Bastanfard [7] przy szacowaniu wieku analizuje proporcje twarzy oraz ilość zmarszczek.

Praca „Age Estimation from Face Images: Challenging Problem for Audience Measurement Systems” autorstwa Vladimira Khryashcheva, Alexandra Ganina, Olgi Stepanovej oraz Antona Lebedeva podsumowała techniki szacowania wieku [11]. Z podsumowania wynikało, że najczęściej stosuje się do wyodrębniania cech z twarzy BIF, czyli biologically inspired features. Powyższa metoda została zaprezentowana w książce „Human Age Estimation Using Bio-inspired Features” autorstwa Guodong Guo i in. Mniej popularne metody analizujące cechy twarzy to filtry Gabora oraz LBP- local binary patterns.

Metoda bazowa została opisana w artykule „Age Estimation from Face Image using Wrinkle Features” [19]. Wykrywanie wieku dzieli się na kilka faz. Na początku należy wykryć twarz. Zastosowany algorytm wykrywania został opisany w sekcji 2.1 Następnie należy wyznaczyć strefy zmarszczkowe na twarzy. W artykule [19] udowodniono, że istnieje kilka konkretnych stref, w których następuje znacząca zmiana ilości zmarszczek wraz z wiekiem. Powyższe strefy zostały wymienione w sekcji 2.2. Sekcja ?? przedstawia technikę wykrywania zmarszczek znajdujących się w strefach. Wykryte zmarszczki pozwalają na obliczenie wrinkle feature dla danej twarzy, zgodnie z opisem w sekcji 2.4. W tym miejscu kończy się faza wyznaczania wrinkle feature dla danej osoby (Rysunek 2.1). Kolejna faza jest potrzebna do znalezienia relacji pomiędzy wrinkle feature a wiekiem. Do tego celu należy zastosować algorytm trenujący, który został opisany w sekcji 2.5. Wynikiem algorytmu trenującego jest zbiór danych, który należy pogrupować, tak jak to opisano w sekcji 2.6. Ostatnią fazą algorytmu jest wykrywanie wieku na podstawie

wyników działania FCM - sekcja 2.7 (Rysunek 2.2).



Rysunek 2.1: Faza 1 algorytmu



Rysunek 2.2: Faza 2 algorytmu

## 2.1 Metoda wykrywania twarzy

W literaturze można odnaleźć wiele metod wykrywania twarzy. Istnieje kilka podejść aby skutecznie wykrywać twarz na danym obrazie [20]:

- metoda oparta na nauce
- metoda niezmienności cech
- metoda dopasowania szablonu twarzy
- metoda bazująca na wyglądzie

Metoda oparta na nauce kieruje się wiedzą na temat wyglądu twarzy, a bardziej precyzyjnie chodzi o charakterystyczne cechy, dzięki którym na zdjęciu można wyodrębnić obszar twarzy. Mowa tutaj o cechach takich jak kształt twarzy, kolor, miejsca o różnej jasności czy specyficzne krawędzie tworzone np przez usta.

W metodzie niezmienności cech wyszukuje się takie strukturalne cechy twarzy, które są widoczne w każdych warunkach oświetleniowych. Ponadto te cechy są widoczne bez względu na punkt widzenia, nawet jeśli twarz jest widoczna z profilu czy przechylona pod kątem.

Z kolei metoda dopasowania szablonu twarzy wykorzystuje kilka standardowych wzorów opisujących twarz. Na wejściu algorytmu obraz jest porównywany z tymi wzorami, a na wyjściu dostajemy informację, w jakim stopniu obraz jest dopasowany do szablonu twarzy.

Ideą metody bazującej na wyglądzie jest trenowanie dużego zbioru obrazów twarzy, tak aby wychwycić zmienność cech twarzy. Tak wytrenowany model jest później wykorzystywany do wykrywania twarzy.

Ponadto w procesie ekstrakcji twarzy z obrazu istnieje wiele problemów [20].

Jednym z problemów jest nieodpowiednia poza. Wiąże się to z różnymi ustawieniami twarzy wobec aparatu fotograficznego lub kamery. Twarz może być nachylona, przechylona lub odchylona. Inaczej mówiąc może mieć różne położenie w trzech wymiarach, a niektóre części twarzy lub jej cechy mogą zostać przysłonięte. Im mniej cech widocznych na twarzy, tym mniej danych, które algorytm może z niej wyodrębnić, a im mniej danych, którymi algorytm operuje, tym mniejsze prawdopodobieństwo prawidłowego wykrycia twarzy.

Niektóre twarze mogą zawierać pewne wyróżniające je cechy takie jak broda, blizny czy okulary. Różnorodność tych cech także wpływa na efektywność wykrywania twarzy.

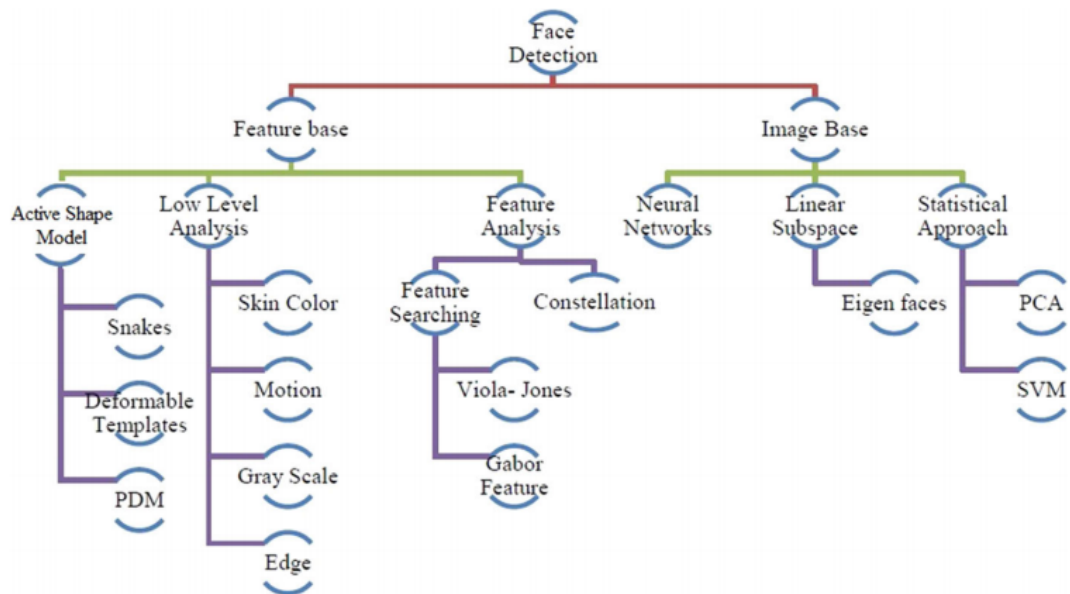


Rysunek 2.3: Przykład twarzy oświetlonej twardym (twarz po lewej) oraz miękkim światłem. [8]

Ilość zmarszczeń na twarzy jest zmienna, w zależności od wyrazu mimicznego. Przy różnych minach zmienia się kształt ust, a czasem pojawiają się ostre krawędzie wynikające z pracy mięśni twarzowych. Widoczne mogą być różne pofałdowania skóry.

Zdarza się, że twarz zostaje częściowo przysłonięta przez jakiś inny obiekt. Na przykład na zdjęciu, które obejmuje grupę wielu osób, część danej twarzy może być przysłonięta przez inną twarz. Takie przysłonięcie przez inny obiekt wiąże się z utratą informacji o części twarzy, co zmniejsza prawdopodobieństwo prawidłowego jej wykrycia.

Kolejnym istotnym elementem jest oświetlenie twarzy. Gdy twarz oświetlona jest tzw. twardym światłem, występują na niej cienie i światła określane jako ostre (Sekcja 2.3). W tym przypadku ryzyko utraty szczegółów oświetlanej twarzy jest większe. Kiedy twarz jest skierowana na wprost słońca, z dużym prawdopodobieństwem można stwierdzić, że zostanie oświetlona twardym światłem. Z kolei miękkie światło jest generowane na przykład przez zachmurzone niebo. Istotne jest także źródło światła, które może być punktowe lub rozproszone. Przy punktowym źródle światła cała twarz jest okryta jednolitym cieniem, którego intensywność zależy od "twardości" światła. Natomiast przy świetle rozproszonym intensywność cieni jest mniejsza.



Rysunek 2.4: Różne techniki wykrywania twarzy. [3]

Na Rys 2.4 przedstawione są różne techniki wykrywania twarzy.

Jak widać metod wykrywania twarzy jest sporo. Omówienie każdej z nich zajęłoby dużo czasu. Poniżej zostaną przytoczone dwie metody wykrywania twarzy. Dodatkowo zostanie omówiona metoda, która posłużyła do wykrywania twarzy w niniejszej pracy.

W pracy „An efficient algorithm for human face detection and facial feature extraction under different conditions” [12] przedstawiono opisaną w skrócie poniżej technikę wykrywania twarzy. W pierwszym etapie procesu obszary, gdzie może znajdować się ludzkie oko, są wykrywane przez przeprowadzenie testów na zacienionych rejonach obrazu. Pary takich obszarów wyodrębnia się na podstawie algorytmu genetycznego, aby następnie wyznaczyć możliwy obszar twarzy. Dla każdego obszaru mierzy się wartość dopasowania na podstawie jego projekcji na wektory własne, tzw. eigenfaces. Aby wiarygodność wykrywania była wyższa, każdy możliwy obszar twarzy normalizuje się pod kątem oświetlenia. Proces ten powtarza się pewną ilość razy, a następnie do dalszej weryfikacji są wybierane możliwe obszary twarzy o wysokiej wartości dopasowania. Na tym etapie mierzy się symetrię twarzy oraz



sprawdza się, czy na każdym wybranym obszarze istnieją rysy twarzy. Rysy określa się przez ewaluację rzeźby topograficznej - wystających i wklęsłych elementów różnych regionów obszaru twarzy, poddanego uprzednio normalizacji. Algorytm jest w stanie wykryć także obszar twarzy, gdy głowa jest przechylona

W roku 1997 w pracy pt. „Vision for man-machine interaction” opisano metodę wykrywania twarzy bazującą na wykrywaniu cechy jaką jest kolor skóry [6]. Kolor skóry jest najbardziej widoczną cechą twarzy zarówno dla człowieka jak i dla maszyny. Ponadto kolor jest przetwarzany znacznie szybciej od innych cech. Przy dobrych warunkach oświetleniowych ustawienie twarzy nie ma wpływu na skuteczność wykrywalności twarzy opisywaną metodą. Każda metoda wykrywania twarzy posiada wady. Jedną z tych wad jest problem wykrywalności twarzy przy nierównomiernym oświetleniu. Problem pojawia się także, gdy na obrazie widoczny jest obszar skóry z poza twarzy np. z rąk. Warto zaznaczyć, że kolor twarzy na obrazie jest zależny od względnego kierunku oświetlenia. Obszar twarzy w omawianym algorytmie jest wykrywany poprzez normalizację histogramu kolorów. Normalizacja jest potrzebna do redukcji wpływu luminancji na kolor.

Algorytm Haar Cascade jest najpopularniejszym algorytmem do wykrywania twarzy w bibliotece OpenCV. Właśnie ta biblioteka była jednym z niezbędnymi elementami programu przy wykrywaniu wieku z tekstury twarzy. W związku z powyższym w wykrywaniu twarzy zastosowano algorytm Haar Cascade. Omawiany algorytm został zaprezentowany w książce „Rapid Object Detection using a Boosted Cascade of Simple Features” w 2001 i składa się z trzech faz [18]. W pierwszej obraz wejściowy przekształcany jest na obraz scałkowany. Następnie wykorzystywany jest algorytm do boostingu, który zmniejsza ilość klasyfikatorów tylko do tych najbardziej istotnych. W ostatniej fazie klasyfikatory łączone są w kaskady w celu przyspieszenia procesu wykrywania twarzy. Znaczna większość metod wykrywania obiektów na obrazie (w tym twarzy) wymaga wstępnego przekształcenia obrazu do skali szarości.

### 2.1.1 Konwersja do skali szarości oraz przestrzenie barw

Każdy piksel w trybie kolorowym ma określoną reprezentację barwy z określonego modelu. Najczęściej są to 3 lub 4 wartości [10]. Pierwszą przestrzenią barw była CIEXYZ. Została ona stworzona w 1931 przez Międzynarodową Komisję ds. Oświetlenia (International Commission on Illumination). Przestrzeń barw CIEXYZ została specjalnie stworzona, by odtworzyć sposób postrzegania barw przez ludzkie oko. Barwa jest opisywana w trzech współrzędnych trójkromatycznych X,Y,Z. Powyższe współrzędne są zależne od składowych - sprawności wizualnych czopków. Czopki to światłoczułe receptory siatkówki ludzkiego oka [10]. Współrzędne X,Y,Z wyliczane są na podstawie trzech podstawowych barw R (czerwonej), G (zielonej) i B (niebieskiej). Współrzędne XYZ są często reprezentowane przez luminancję Y oraz współrzędne x, y chromatyczności. Wyliczanie współrzędnych x, y przedstawiono na wzorach 2.1 oraz 2.2.

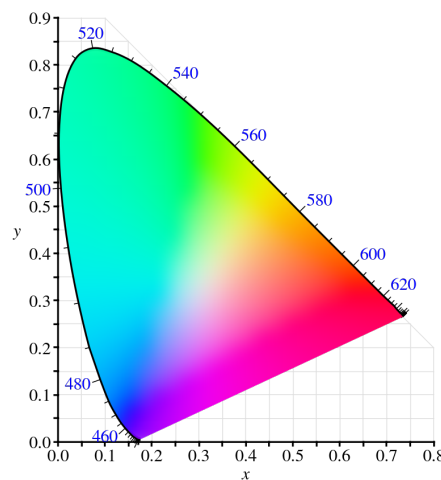
$$x = \frac{X}{X + Y + Z} \quad (2.1)$$

$$y = \frac{Y}{X + Y + Z} \quad (2.2)$$

Na Rysunku 2.5 przedstawiono diagram chromatyczności reprezentujący przestrzeń barw CIEXYZ.

Kolory mogą być także odwzorowane przez przestrzeń barw CMYK. Skrót CMYK oznacza odpowiednio:

- Cyan - odcień niebieskiego
- Magenta - kolor karmazynowy
- Yellow - kolor żółty
- K - key colour - kolor czarny



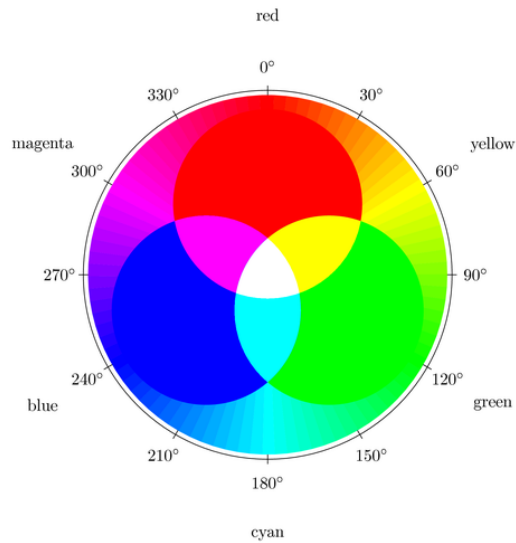
Rysunek 2.5: Przykład rozpoznawania obiektów na zdjęciu ulicy. [5]

Barwa wynikowa powstaje poprzez mieszanie trzech kolorów - niebieskiego, karmazynowego oraz żółtego. Mieszanie zachodzi według zasady syntezy subtraktywnej [10]. Synteza subtraktywna polega na mieszaniu kolorów przez odejmowanie promieniowań widzialnych różnych długości. Przykładem syntezy subtraktywnej jest np. mieszanie farb o różnych kolorach.

Najczęściej barwy są reprezentowane przez przestrzeń barw RGB [10]. Przestrzeń kolorów RGB składa się z trzech kanałów [9]:

- R - czerwonego (z angielskiego Red)
- G - zielonego (z angielskiego Green)
- B - niebieskiego (z angielskiego Blue)

Barwy mieszane są poprzez syntezę addytywną. W przeciwieństwie do syntezy subtraktywnej barwa wynikowa powstaje poprzez sumowanie wiązek światła widzialnego o różnych długościach [10]. Każdy piksel opisany za pomocą przestrzeni barw RGB ma trzy 8-bitowe wartości reprezentujące każdy kanał. Spotykane są 12- lub 16-bitowe reprezentacje kanałów, jednak 8-bitowa jest najpopularniejsza. Dla 8-bitowych kanałów wartość „0” danego kanału oznacza brak jasności, natomiast „255” maksymalną jasność. Poprzez



Rysunek 2.6: Mieszanie kanałów RGB [14].



Rysunek 2.7: Kolor R=153 G=217 B=234.

mieszanie jasności tych trzech kanałów można uzyskać szerokie spektrum barw (Rysunek 2.6).

Przykładowo kolor o reprezentacji R=153 G=217 B=234 przedstawiono na Rysunku 2.7

Kolor (Rysunek 2.7) może być też reprezentowany w kodzie szesnastkowym #99D9EA. Każda wartość heksadecymalna odpowiada kolejno kanałowi R, G, B.

Obraz może też być przedstawiony stosując odcienie jednej barwy. Taka obraz nazywa się obrazem monochromatycznym. Najczęściej stosowaną barwą w takich obrazach jest szarość [10].

Istnieją 3 metody konwersji obrazu z przestrzeni RGB na monochromatyczny [14].

- największej jasności

- średnia
- luminancji

Metoda największej jasności konwertuje na skalę szarości wg wzoru 2.3.

$$\frac{(max(R, G, B) + min(R, G, B))}{2} \quad (2.3)$$

Metoda średnia bazuje na wzorze 2.4, natomiast metodę luminancji ilustruje wzór 2.5.

$$\frac{(R + G + B)}{3} \quad (2.4)$$

$$0.21R + 0.72G + 0.07B \quad (2.5)$$

W niniejszej pracy zastosowano konwersję za pomocą metody średniej.

### 2.1.2 Algorytm Haar Cascade

Haar Cascade jest algorytmem służącym do wykrywania obiektów na obrazach. Został stworzony przez Paula Viola oraz Michaela Jonesa w 2001 roku [18].

Opiera się na zbudowaniu kaskadowej funkcji za pomocą trenowania wielu zdjęć. Zdjęcia są dzielone na dwie kategorie - pozytywne oraz negatywne. Na zdjęciach klasyfikowanych jako pozytywne istnieje obiekt, który ma zostać wykryty, natomiast na zdjęciach negatywnych nie ma tego obiektu.

Ekstrakcja cech w algorytmie Violi i Jonesa jest realizowana przez filtry Haara. Są to prostokątne okienka nakładane na obraz, które analizują jasność pikseli (Rysunek 2.8).

Przed zastosowaniem filtru Haara obraz musi zostać przekształcony do skali szarości. W niniejszej pracy należało przekształcić każdy obraz z trybu kolorowego na monochromatyczny co opisano w sekcji 2.1.1.

Każde okienko zawiera białe oraz czarne prostokąty. Wyznaczana jest suma jasności pikseli w obu rodzajach prostokątów, a następnie dla każdego



Rysunek 2.8: Filtr Haara a) krawędziowy b) liniowy c) szachownica [2]

okna obliczana jest różnica pomiędzy białymi a czarnymi. Opisywany algorytm ma zastosowanie w wykrywaniu krawędzi. Na granicy krawędzi istnieje różnica w jasności pikseli (Rysunek 2.9).

W celu poprawy efektywności sumowania pikseli stosowane są rozwiązania zwane w języku angielskim Summed-area table [17]. Summed-area table jest również nazywana w literaturze Integral Image, czyli obrazem scałkowanym. Ideą obrazu scałkowanego jest, aby każdy obraz został przekształcony w tabelę, w której każdy element  $x, y$  tej tabeli odpowiada sumie jasności wszystkich pikseli według wzoru 2.6.

$$I(x, y) = \sum_{x' \leq x \cap y' \leq y} i(x', y') \quad (2.6)$$

gdzie  $I(x, y)$  jest wartością na pozycji  $x, y$  w tabeli (tabela obrazu scałkowanego),  $i(x, y)$  - jasność piksela o współrzędnych  $x, y$  na obrazie.

Na Rysunku 2.10 przedstawiona jest tabela prezentująca jasność pikseli przed zastosowaniem całkowania obrazu.

Po całkowaniu otrzymujemy tabelę podobną do przedstawionej na Rysunku 2.11.

Sumowanie przykładowego okna (Rysunek 2.12) wymaga 4 operacji (Wzór 2.7).



Rysunek 2.9: Filtr Haara nałożony na krawędź twarzy [2]

Image

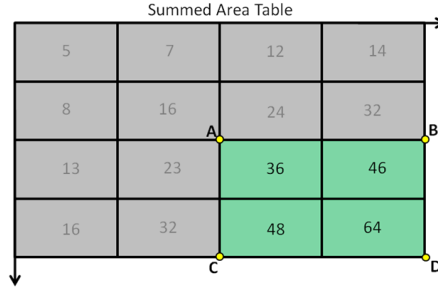
5	2	5	2
3	6	3	6
5	2	5	2
3	6	3	6

Rysunek 2.10: Tabela jasności poszczególnych pikseli przed zastosowaniem całkowania [4]

Summed Area Table

5	7	12	14
8	16	24	32
13	23	36	46
16	32	48	64

Rysunek 2.11: Filtr Haara nałożony na krawędź twarzy [4]



Rysunek 2.12: Sumowanie okna [4]

$$\sum_{x_0 \leq x \leq x_1 \cap y \leq y_1} i(x, y) = I(D) + I(A) - I(B) - I(C) \quad (2.7)$$

gdzie lewa część równania oznacza sumę jasności pikseli zaznaczonego okna tj. na Rysunku 2.12,  $I(A)$  - wartość scałkowanego obrazu przy punkcie A (analogicznie  $I(B)$ ,  $I(C)$ ,  $I(D)$ ) - (Rysunek 2.12).

W związku z powyższym obliczenie wartości dla krawędziowego filtru Haara wymaga obliczenia różnicy dwóch sum co wymaga ośmiu operacji. Reprezentacja obrazu za pomocą obrazu scałkowanego znacznie zwiększa efektywność obliczania wartości w filtrze Haara.

Liczba cech wykrywanych w zdjęciu za pomocą filtru Haara jest znacznie większa od liczby pikseli na obrazie [18]. Dla obrazu o rozmiarze 384x288 liczba cech wynosi ponad 180000. Autorzy algorytmu stwierdzili, że dla zwiększenia szybkości algorytmu należy wybrać małą grupę cech, które razem mogą stworzyć jeden efektywny klasyfikator obiektu. W celu wyodrębnienia tych istotnych cech zastosowano algorytm Adaboost, który został opisany poniżej.

Zbiór  $n$  obrazów do trenowania można oznaczyć tak jak we wzorze 2.8:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (2.8)$$

oznacza, że obraz jest odpowiednio negatywny lub pozytywny. Następnym krokiem jest inicjalizacja wag (Wzór 2.9).



$$w_{1,i} = \frac{1}{2m}, \frac{1}{2l} \quad (2.9)$$

odpowiednio dla  $y_i = 0, 1$ , gdzie  $m, l$  oznacza odpowiednio liczbę negatywnych oraz pozytywnych zdjęć. Następnie Dla  $t = 1, \dots, T$ :

1. Normalizowane są wagi (Wzór 2.10):

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}} \quad (2.10)$$

$w_t$  jest rozkładem prawdopodobieństwa

2. Dla każdej cechy  $j$ , trenowany jest klasyfikator  $h_j$ , który używa tylko jedną cechę wyliczoną z filtru Haara. Błąd jest wyliczony następująco (Wzór 2.11):

$$w_t, \epsilon_j = \sum_i w_i h_j(x_i) - y_i \quad (2.11)$$

3. Wybierany jest klasyfikator  $h_t$  z najmniejszym błędem  $\epsilon_t$ .
4. Następuje aktualizacja wag (Wzór 2.12):

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \quad (2.12)$$

, gdzie  $e_i = 0$ . Jeśli  $x_i$  jest sklasyfikowane prawidłowo, wtedy  $e_i = 1$ , w innym wypadku  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ .

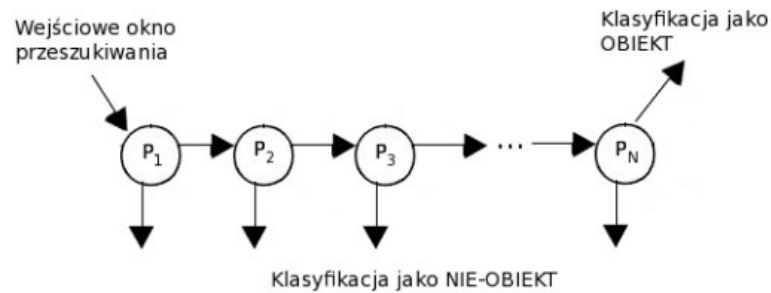
Silny klasyfikator  $h(x)$  jest opisany równaniem:

$$h(x) = \begin{cases} 1 & \text{gdy } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{w przeciwnym wypadku} \end{cases} \quad (2.13)$$

, gdzie  $\alpha_t = \lg \frac{1}{\beta_t}$

Algorytm Adaboost zmniejsza ilość cech Haara z ponad stu tysięcy do kilkuset - do tych najistotniejszych cech.

Ostatnim etapem jest wytworzenie kaskady klasyfikatorów. Zwiększa ona



Rysunek 2.13: Kaskada klasyfikatorów. [23]

znacznie szybkość wykrywania pożądanego obiektu na obrazie. Ideą kaskady jest zgrupowanie klasyfikatorów, które powstały w poprzednim procesie - procesie boostingu. Klasyfikatory są grupowane w okna. Okna są połączone ze sobą tak jak na rysunku 2.13. Okna są oznaczone jako  $P_1, P_2, \dots, P_n$ . Gdy dane okno wykryje obiekt przechodzi do kolejnego okna w kaskadzie. W przeciwnym wypadku algorytm przerywa działanie i na danym obrazie nie zostaje zidentyfikowany obiekt. Okna są poustawiane tak aby każde z nich klasyfikowało obiekt z różnym prawdopodobieństwem wykrycia oraz prawdopodobieństwem błędu. Składnikami wyżej wymienionego prawdopodobieństwa jest macierz pomyłek (Rysunek 2.14).

Z macierzy 2.14 można odczytać czy klasyfikator poprawnie sklasyfikował dane testowe. W macierzy użyto 4 pojęcia:

- TP (true positive) - poprawna klasyfikacja, jako obiekt
- TN (true negative) - poprawna klasyfikacja, jako nie - obiekt
- FP (false positive) - błędna klasyfikacja jako obiekt
- FN (false negative) - błędna klasyfikacja jako nie - obiekt

Prawdopodobieństwo błędu wyliczane jest ze współczynnika FPR (false positive rate) (Wzór 2.14).

$$FPR = \frac{FP}{FP + TN} \quad (2.14)$$

		Odpowiedź klasyfikatora	
		TAK	NIE
Istnieje obiekt?	TAK	True Positive	False Negative
	NIE	False Positive	True Negative

Rysunek 2.14: Macierz pomyłek.

Natomiast prawdopodobieństwo wykrycia obiektu wyliczane jest ze współczynnika TPR (true positive ratio) (Wzór 2.15)

$$TPR = \frac{TP}{TP + FN} \quad (2.15)$$

Pierwsze okna posiadają klasyfikatory o słabszym TPR oraz FPR niż kolejne okna. Oznacza to, że prawdopodobieństwo TPR w oknie  $P_{x-1}$  jest mniejsze od tego w  $P_x$ . Natomiast prawdopodobieństwo FPR w oknie  $P_{x-1}$  jest większe od tego w  $P_x$ . Ostatnie okna mają największy współczynnik TPR oraz najmniejszy FRP ze wszystkich. Takie ustawienie okien ma na celu wstępne przepuszczenie przez okna obrazy, które z dużym prawdopodobieństwem zawierają szukany obiekt. Natomiast ostatnie okna w kaskadzie analizują niewielką część obrazu wejściowego.

Biblioteka OpenCV zawiera wytrenowane klasyfikatory, które zostały użyte w tej pracy magisterskiej. Wykorzystano je do wykrycia twarzy, ust oraz oczu. Klasyfikatory mają postać plików xml, które można znaleźć na oryginalnym repozytorium projektu OpenCv.



Rysunek 2.15: Wykryty nos oraz oczy.

## 2.2 Wyznaczanie stref

Przed wyznaczeniem stref zmarszczkowych należy zidentyfikować na twarzy oczy oraz nos (Rysunek 2.15).

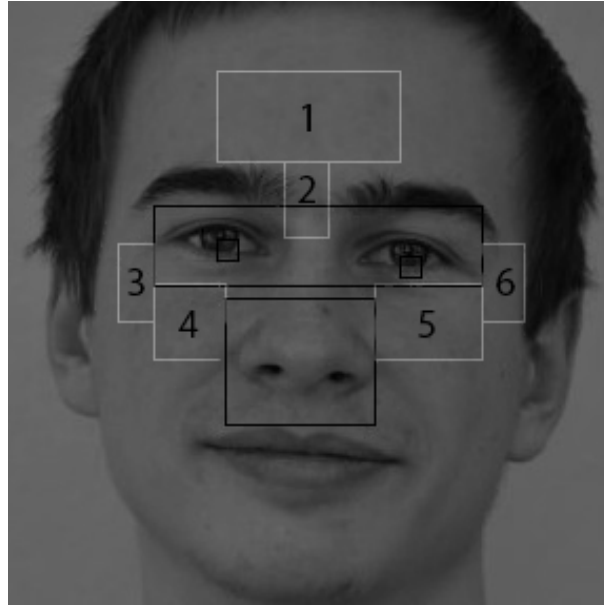
Należy podkreślić, że algorytm przerywa działanie jeśli nie zostanie wykryta twarz, oczy lub nos.

Gdy zostanie wykryty obszar twarzy oczu oraz nosa wyznaczone zostaje sześć stref zmarszczkowych (Rysunek 2.16).

Strefy zmarszczkowe są na czole (Strefa „1”), w górnej części nosa (Strefa „2”), górnej części policzków (Strefa „4” i „5”) oraz w okolicach powiek (Strefa „3” i „6”). To właśnie te miejsca zostały uznane przez autorów książki „Age Estimation from Face Image using Wrinkle Features” [19] za najbardziej znaczące w detekcji wieku.

Po detekcji oczu należy zmierzyć odległość pomiędzy środkiem lewego oka  $(x_l, y_l)$ , a prawego  $(x_p, y_p)$  (Wzór 2.16).

$$d = \sqrt{(x_r - x_l)^2 + (y_r - y_l)^2} \quad (2.16)$$



Rysunek 2.16: Strefy zmarszczkowe widoczne w białych prostokątach.

Odległość  $d$  służy do wyznaczanie strefy znajdującej się na czole (Rysunek 2.17).

Autorzy [19] algorytmu założyli, że odległość od linii oczu do linii brwi wynosi  $0,4 * d$ . Natomiast wymiary „strefy czoła” wynoszą  $d \times 0,5d$ .

Na Rysunku 2.18 przedstawione są współrzędne, które są pomocne w wyznaczaniu stref zmarszczkowych.

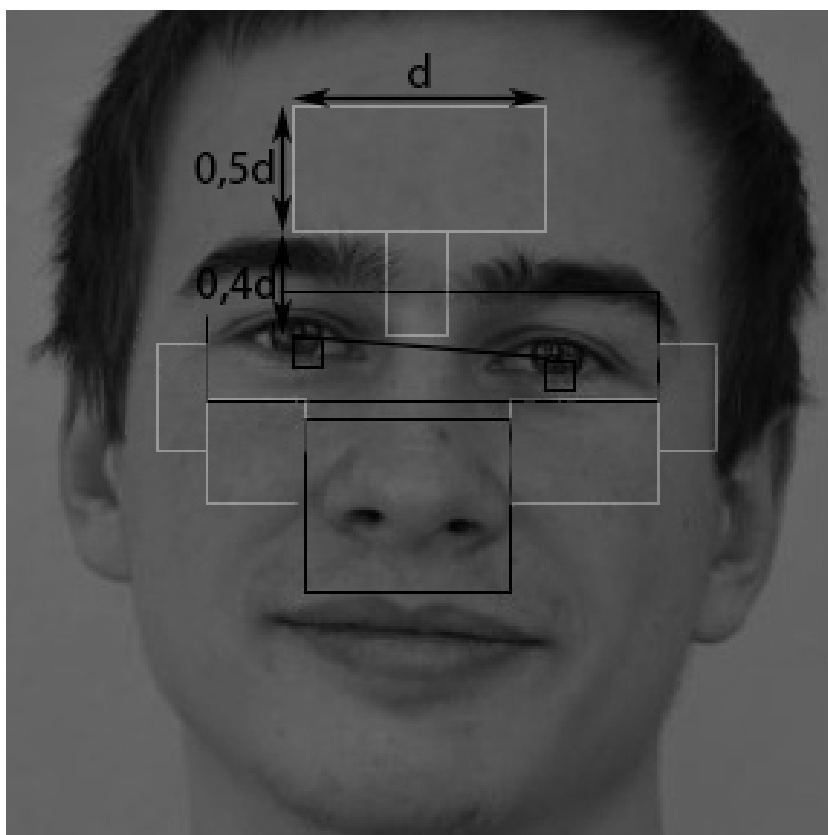
Strefa 2 została wyznaczona znając położenie prawego oka, dystansu między oczami oraz strefy „1”. Każda strefa może być wyznaczona przez jeden punkt (Rysunek 2.19) oraz jej wymiary.

Współrzędna  $x$  punktu A została wyznaczona ze wzoru 2.17

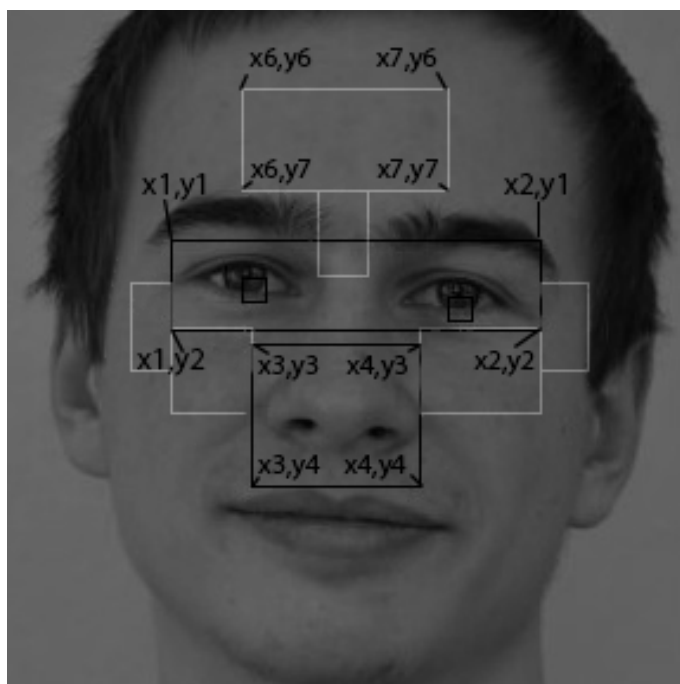
$$x_A = x_6 + 0,375 \times d \quad (2.17)$$

, gdzie  $x_6$  to współrzędna z Rysunku 2.18. Natomiast  $d$  jest odległością pomiędzy oczami (Wzór 2.16).

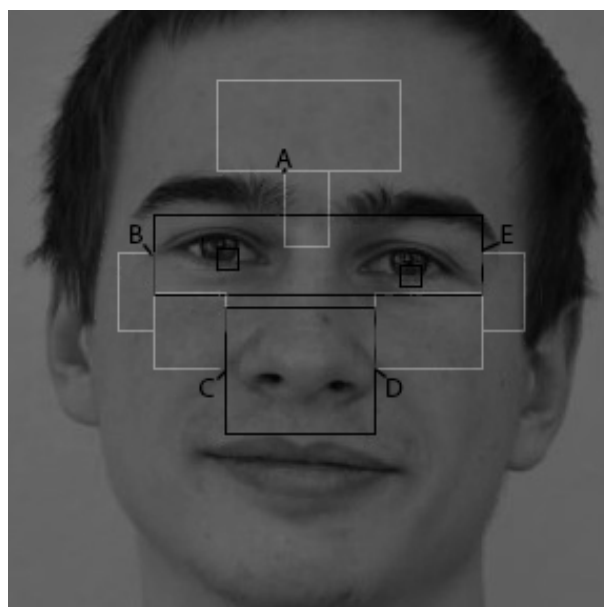
Współrzędna  $y$  punktu A - Wzór 2.18:



Rysunek 2.17: Wyznaczenie strefy znajdującej się na czole.



Rysunek 2.18: Pomocnicze współrzędne do wyliczania stref zmarszczkowych.



Rysunek 2.19: Punkty wyznaczające położenie stref.

$$y_A = y_7 \quad (2.18)$$

Współrzędna  $y_7$  analogicznie jak  $x_6$  znajduje się na Rysunku 2.18.

Zakładając, że prawe oko ma współrzędne  $x_o$  oraz  $y_o$  wyznaczana jest wysokość strefy 2 (Wzór 2.19).

$$h_A = y_o - y_7 \quad (2.19)$$

Szerokość strefy 2 (Wzór 2.20):

$$w_A = 0,25 * d \quad (2.20)$$

Wartość 0,25 ze wzoru 2.20 została dobrana empirycznie.

Współrzędne punktu C (strefa 4) zostają wyznaczone ze Wzoru 2.21 oraz 2.22.

$$x_C = x_3 \quad (2.21)$$

$$y_C = \frac{y_4 - y_3}{2} \quad (2.22)$$

Wysokość oraz szerokość strefy 4 (Wzór 2.23 oraz 2.24)

$$h_C = y_C - y_3 \quad (2.23)$$

$$w_C = x_3 - x_1 \quad (2.24)$$

Strefa 5 zawiera punkt D - Wzór 2.25 oraz 2.26.

$$x_D = x_4 \quad (2.25)$$



$$y_D = y_C \quad (2.26)$$

Dla strefy 5 wyznaczamy wysokość (Wzór 2.27) oraz szerokość (Wzór 2.28).

$$h_D = h_C \quad (2.27)$$

$$w_D = x_2 - x_4 \quad (2.28)$$

Strefa 3 zawiera punkt B wyznaczany ze Wzorów 2.29 i 2.30

$$x_B = x_1 \quad (2.29)$$

$$y_B = \frac{y_2 - y_1}{2} \quad (2.30)$$

Wysokość (Wzór 2.31) oraz szerokość (Wzór 2.32) strefy 3.

$$h_B = y_C - y_B \quad (2.31)$$

$$w_B = w_C * 0,4 \quad (2.32)$$

Na samym końcu zostaje wyznaczona strefa 6. Wyznaczona zostaje współrzędna E - Wzór 2.33 i 2.34.

$$x_E = x_2 \quad (2.33)$$

$$y_E = y_B \quad (2.34)$$

Szerokość oraz wysokość strefy 6.

$$h_E = h_B \quad (2.35)$$

$$w_E = w_D * 0,4 \quad (2.36)$$

Należy zauważyć, że szerokość strefy 3 i 6 jest pomnożona przez 0,4 razy szerokość odpowiednio strefy 4 i 5. Wartość „0,4” została dobrana empirycznie.

## 2.3 Wykrywanie zmarszczek - detektor Canny

Następnym krokiem algorytmu po wyznaczeniu stref zmarszczkowych jest wyodrębnienie zmarszczek. W celu identyfikacji zmarszczek zastosowano detektor Canny. Detekcja krawędzi pozwala na wyodrębnienie wielu użytecznych cech z obrazu. Ponadto znacznie zmniejsza ilość informacji do przetworzenia przez algorytm, który wyodrębnia cechy z obrazu. Istnieje wiele detektorów krawędzi, jednak detektor Canny jest jednym z najbardziej dokładnych i niezawodnych. Metoda detekcji została opracowana przez Johna F. Canny w 1986 roku. Oprócz samej implementacji algorytmu Canny zaprezentował teorię obliczeniową, która wyjaśnia działanie tej metody. Canny ponadto zauważył, że wymagania dotyczące implementacji detekcji krawędzi są podobne do siebie w wielu systemach wizyjnych. W związku z powyższym jego algorytm może zostać zastosowany w wielu systemach wizyjnych. Najważniejsze zasady, które są niezbędne do dobrej detekcji krawędzi w opisywanym algorytmie są przedstawione poniżej:

- Detekcja krawędzi z niskim prawdopodobieństwem błędu: Oznacza to, że algorytm powinien wykryć jak najwięcej krawędzi, które rzeczywiście istnieją. Natomiast ilość krawędzi wykrytych błędnie powinna być jak najmniejsza.
- Precyzja detekcji: Algorytm powinien precyzyjnie zlokalizować krawędź

- Brak redundantnych detekcji: Krawędź powinna być zlokalizowana jednokrotnie. Szum na obrazie nie powinien generować dodatkowych krawędzi.

W algorytmie detekcji Canny używa rachunku wariacyjnego. Rachunek wariacyjny to dziedzina analizy matematycznej. Dziedzina ta analizuje przestrzenie funkcyjne i znajduje w nich ekstrema funkcjonałów. Funkcjonały natomiast odwzorowują przestrzeń wektorową na liczby rzeczywiste. Reasumując, rachunek wariacyjny pomaga znaleźć charakterystyczną funkcję. Funkcjonał dla powyższej funkcji przyjmuje wartość ekstremalną. Algorytm detektora jest podzielony na kilka kroków [13]:

- Redukcja szumów na obrazie filtrem Gaussa.
- Szukanie gradientów jasności.
- Zastosowanie techniki pocieniania krawędzi.
- Likwidacja krawędzi o małym gradiencie jasności.
- Filtracja poprzez histerezę.

### 2.3.1 Redukcja szumów na obrazie filtrem Gaussa

Szum na obrazie to piksele o losowym kolorze, jasności oraz umiejscowieniu (współrzędnych na obrazie). Powyższe piksele są nadmiarowymi informacjami o obrazie oraz są efektem ubocznym przetwarzania obrazu przez matrycę cyfrową (Rysunek 2.20).

Efekt szumu jest spotykany również w fotografii analogowej. W fotografii cyfrowej szum wzrasta na skutek zwiększania czułości matrycy lub przez wzrost jej temperatury. Redukcję szumów można uzyskać stosując filtr Gaussa. Splot filtru Gaussa z obrazem daje w wyniku wygładzony obraz ze zmniejszoną ilością szumu. Filtr Gaussa może mieć różne wymiary. Najczęściej stosowanym jest 5x5. Rozmiar filtra ma wpływ na wydajność wygładzania obrazu, a co za tym idzie na czułość wykrywania szumu [13].



Rysunek 2.20: a) Obraz z nieznaczną ilością szumów b) Obraz ze znaczną ilością szumu. [21]

### 2.3.2 Szukanie gradientów jasności

Istotnym parametrem w wykrywaniu krawędzi jest gradient jasności. Określa on jak bardzo zmienia się jasność danego piksela względem sąsiadujących pikseli. Gradient jasności otrzymany zostaje ze wzoru 2.37:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.37)$$

gdzie  $G_x$  jest zmianą jasności danego piksela w kierunku poziomym, a  $G_y$  w kierunku pionowym. Kierunek gradientu opisuje wzór 2.38

$$\Theta = \text{atan2}(G_y, G_x) \quad (2.38)$$

Krawędzie na obrazie charakteryzują się pewną zmianą jasności na danym obszarze obrazu. Krawędź na obrazie może być położona pod różnym kątem. Opisywany detektor używa 4 filtry w celu wykrycia pionowych, poziomych, ukośnych krawędzi.

Kierunek jest zaokrąglony do jednego z filtrów kierunku: poziomego ( $\Theta = 0^\circ$ ), pionowego ( $\Theta = 90^\circ$ ) lub ukośnego ( $\Theta = 45^\circ$  lub  $\Theta = 135^\circ$  stopni).

Każdy piksel obrazu otrzymuje wartość gradientu (Wzór 2.37) oraz kierunek - zgodny z powyższym filtrem kierunku. W wyniku powyższego działania zostaje otrzymany obraz gradientowy [13].

### 2.3.3 Zastosowanie techniki pocieniania krawędzi

Pocienianie krawędzi służy do filtracji krawędzi powstałych po poprzednim kroku. Krawędzie wykryte za pomocą gradientu są rozmyte (nieostre). Technika pocieniania krawędzi szuka na małym obszarze gradientów o największej wartości. Wszystkie mniejsze są ignorowane. Dzięki temu zostawiane są gradienty o największej wartości, które identyfikują najostrzejsze krawędzie. Algorytm operuje na obrazie gradientowym powstałym przez algorytm opisany w sekcji 2.3.2 Działa on następująco.

Odczytana zostaje wartość gradientu oraz jego kierunek dla danego piksela. Następnie wartość gradientu zostaje porównana z wartościami gradientu dla dwóch sąsiednich pikseli. Pierwszy sąsiedni piksel jest wyznaczony przed aktualny kierunek gradientu, a drugi przez przeciwny kierunek. Jeśli wartość danego piksela jest największa spośród pikseli wzdłuż wyżej wymienionych linii, to dany piksel należy do najostrzejszej krawędzi [13].

### 2.3.4 Filtracja krawędzi o małym gradiencie jasności

Zastosowanie techniki pocieniania krawędzi zostawia wiele krawędzi, które są wygenerowane przez szum i zmiany kolorów na obrazie. Kolejnym krokiem jest filtracja powyższych krawędzi, gdyż są one nadmiarowe. W tym celu ustawiany jest pewien próg - próg mały. Oprócz progu małego ustawiany jest próg duży w celu wyodrębnienia krawędzi o wysokim gradiencie jasności. Jeśli wartość gradientu dla danej krawędzi jest mniejsza od progu małego, to zostaje ona usunięta. W przypadku, gdy wartość jest większa od progu małego, ale mniejsza od progu dużego to dana krawędź zostaje oznaczona jako słaba. Krawędź zostaje oznaczona jako mocna, gdy wartość gradientu dla niej jest większa od wartości progu dużego [13].

### 2.3.5 Filtracja poprzez histerezę

W wyniku działania algorytmu do tej pory zostały uzyskane słabe oraz mocne krawędzie. Mocne krawędzie zostaną oznaczone jako prawdziwe krawędzie znajdujące się na obrazie. Słabe krawędzie mogą być częścią mocnych krawędzi. Ponadto mogą zostać wygenerowane przez szum lub zmiany kolorów. Powyższe krawędzie nie są prawdziwymi krawędziami w obrazie, więc w ostatnim kroku algorytmu powinny zostać przefiltrowane. Słabe krawędzie, które są powiązane z mocnymi znajdują się w najbliższym sąsiedztwie z mocnymi krawędziami. W celu odnalezienia tych powiązań pomiędzy krawędziami zostaje zastosowana analiza spójności krawędzi. Jeśli zostaje zidentyfikowane powiązanie pomiędzy mocną, a słabą krawędzią to słaba krawędź zostaje w obrazie. W przypadku, gdy słaba krawędź nie jest powiązana z żadną mocną krawędzią - zostaje usunięta [13].

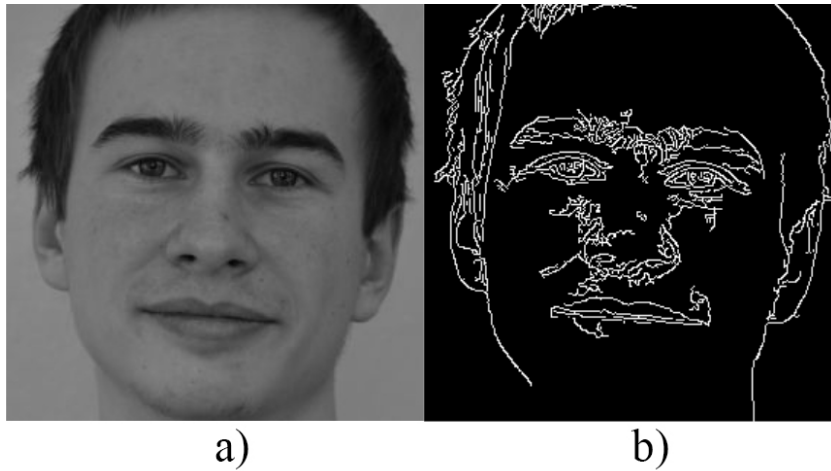
## 2.4 Wyliczanie wrinkle feature

W wyniku działania detektora Cannego, który został opisany w sekcji 2.3 wygenerowany zostaje obraz binarny z wyodrębnionymi krawędziami. Obraz binarny zawiera piksele o dwóch wartościach jasności pikseli. Piksel może być albo biały albo czarny. Krawędzie można rozpoznać jako białe piksele, natomiast czarne piksele oznaczają brak krawędzi (Rysunek 2.21).

Na Rysunku 2.21 widoczne są krawędzie, które wyróżniają tło od twarzy. Ponadto widoczne są poszczególne części twarzy tj. nos, oczy, brwi. Można także zauważyć dodatkowe krawędzie w strefach zmarszczkowych (Seksja 2.2), które identyfikują zmarszczki. Ilość białych pikseli w strefach zmarszczkowych jest wprost proporcjonalna do ilości zmarszczek na twarzy danej osoby.

Z każdej strefy zmarszczkowej obliczany jest stosunek ilości białych pikseli do wszystkich pikseli (Wzór 2.39).

$$W_{s1} = \frac{PB_{s1}}{PW_{s1}} \quad (2.39)$$



Rysunek 2.21: a) Oryginalny obraz b) Obraz z wykrytymi krawędziami.

, gdzie  $W_{s1}$  jest stosunkiem białych pikseli do wszystkich w strefie 1,  $PB_{s1}$  - suma białych pikseli w strefie 1,  $PW_{s1}$  - suma wszystkich pikseli w strefie 1. Analogicznie obliczane są stosunki pikseli dla pozostałych stref. Ostatnim etapem jest sumowanie wszystkich stosunków pikseli (Wzór 2.40).

$$WF = W_{s1} + W_{s2} + W_{s3} + W_{s4} + W_{s5} + W_{s6} \quad (2.40)$$

, gdzie WF to wrinkle feature - parametr określający ilość zmarszczek dla danej osoby.

## 2.5 Algorytm trenowania

## 2.6 Grupowanie danych - FCM

### 2.6.1 Wstęp do grupowania danych

### 2.6.2 Metoda FCM

## 2.7 Wyznaczanie wieku





## Rozdział 3

# Modyfikacje metody bazowej

### 3.1 Odjęcie wybranej strefy

#### 3.1.1 Zmiana algorytmu względem metody bazowej

### 3.2 Zastosowanie metody HOG

#### 3.2.1 Opis algorytmu HOG

#### 3.2.2 Zastosowanie w projekcie

### 3.3 Metoda HOG oraz grupowanie KNN

#### 3.3.1 Grupowanie KNN

#### 3.3.2 Zastosowanie w projekcie



## Rozdział 4

### Badania



## Rozdział 5

## Podsumowanie



# Bibliografia

- [1] T. F. Cootes A. Lanitis, Ch. J. Taylor. *Toward automatic simulation of aging effects on face images*. IEEE Transactions of Pattern Analysis and Machine Intelligence, 2002.
- [2] Abid K. Alexander Mordvintsev. Face Detection using Haar Cascades . [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_objdetect/py\\_face\\_detection/py\\_face\\_detection.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html), 2019. [Online; Dostęp: 28.09.19].
- [3] Munish Kumar Ashu Kumar, Amandeep Kaur. *Face detection techniques: a review*. Springer, 2018.
- [4] BADGERATI. <https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/>, 2010. [Online; Dostęp: 28.09.19].
- [5] Brent Bergherm. <https://brentbergherm.com/cie-chromaticity-diagram/>, 201. [Online; Dostęp: 10.10.19].
- [6] Coutaz J Crowley JL. *Vision for man-machine interaction*. Robotics and Autonomous Systems, 1997.
- [7] M.M. Dehshibi, A. Bastanfard. *A new algorithm for age recognition from facial images*. Signal Processing, 2010.
- [8] Erik Fritts. <https://www.videomaker.com/article/c03/18848-tell-your-story-more-effectively-with-the-correct-application-of-hard-and-soft-light>, 2017. [Online; Dostęp: 07.10.19].

- [9] Robert Hirsch. *Exploring Colour Photography: A Complete Guide*. Laurence King Publishing, 2004.
- [10] Noor A. Ibraheem, Mokhtar M. Hasan, Rafiqul Z. Khan, Pramod K. Mishr. *Understanding Color Models: A Review*. ARPN Journal of Science and Technology, 2012.
- [11] Vladimir Khryashchev, Alexander Ganin, Olga Stepanova, Anton Lebedev. *Age Estimation from Face Images: Challenging Problem for Audience Measurement Systems*. Yaroslavl State University, 2014.
- [12] Wan-Chi Siu Kwok-Wai Wong, Kin-Man Lam. *Age Estimation from Face Images: Challenging Problem for Audience Measurement Systems*. Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, 2000.
- [13] Xiaojun Ma, Bo Li, Ying Zhang, Ming Yan. *The Canny Edge Detection and Its Improvement*. Springer, 2012.
- [14] Henrik Skov Midtiby. Example: Rgb color mixing, 2019. [Online; Dostęp: 28.09.19].
- [15] R. Chellappa N. Ramanathan. *Face verification across age progression*. IEEE Transactions on Image Processing, 2006.
- [16] Moses Olafenwa. Przykład rozpoznawania obrazów. <https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606/>, 2018. [Online; Dostęp: 30.09.18].
- [17] M. Jones P. Viola. *Robust Real-Time Object Detection*. International Journal of Computer Vision, 2002.
- [18] Michael Jones Paul Viola. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Accepted conference on computer vision and pattern recognition, 2001.



- [19] Rituparna Sahaa Ranjan Janaa, Debaleena Dattaa. *Age Estimation from Face Image using Wrinkle Features*. RCC Institute of Information Technology, 2014.
- [20] Qaim Mehdi Rizvi. *A Review on Face Detection Methods*. Qassim University, 2011.
- [21] Przemyslaw Spurek. <http://analizaobrazu.x25.pl/articles/11>, 2018. [Online; Dostęp: 19.10.19].
- [22] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.
- [23] Piotr Wilkowski. *Wykorzystanie algorytmu detekcji i lokalizacji w zadaniu chwytania*. Politechnika Warszawska, Wydział Elektroniki I Technik Informacyjnych, 2009.
- [24] K. Smith-Miles X. Geng, Z. Zhou. *Automatic age estimation based on facial aging patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.
- [25] N. da V. Lobo Y. H. Kwon. *Age classification from facial images*. *Computer Vision and Image Understanding*. 1999.



# Dodatki



# Dokumentacja techniczna

$X$

---

# Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model-view-controller*)

$N$  liczebność zbioru danych

$\mu$  stopień przynależności do zbioru

$\mathbb{E}$  zbiór krawędzi grafu

$\mathcal{L}$  transformata Laplace’a





# Zawartość dołączonej płyty

Do pracy dołączona jest płyta CD z następującą zawartością:

- praca w formacie pdf,
- źródła programu,
- zbiory danych użyte w eksperymentach.



# Spis rysunków

1.1	Przykład rozpoznawania obiektów na zdjęciu ulicy. [16]	2
2.1	Faza 1 algorytmu	5
2.2	Faza 2 algorytmu	5
2.3	Przykład twarzy oświetlonej twardym (twarz po lewej) oraz miękkim światłem. [8]	7
2.4	Różne techniki wykrywania twarzy. [3]	8
2.5	Przykład rozpoznawania obiektów na zdjęciu ulicy. [5]	11
2.6	Mieszanie kanałów RGB [14].	12
2.7	Kolor R=153 G=217 B=234.	12
2.8	Filtr Haara a) krawędziowy b) liniowy c) szachownica [2]	14
2.9	Filtr Haara nałożony na krawędź twarzy [2]	15
2.10	Tabela jasności poszczególnych pikseli przed zastosowaniem całkowania [4]	15
2.11	Filtr Haara nałożony na krawędź twarzy [4]	15
2.12	Sumowanie okna [4]	16
2.13	Kaskada klasyfikatorów. [23]	18
2.14	Macierz pomyłek.	19
2.15	Wykryty nos oraz oczy.	20
2.16	Strefy zmarszczkowe widoczne w białych prostokątach.	21
2.17	Wyznaczenie strefy znajdującej się na czole.	22
2.18	Pomocnicze współrzędne do wyliczania stref zmarszczkowych.	23
2.19	Punkty wyznaczające położenie stref.	23
2.20	a) Obraz z nieznaczną ilością szumów b) Obraz ze znaczną ilością szumu. [21]	28

2.21 a) Oryginalny obraz b) Obraz z wykrytymi krawędziami. . . .	31
--	----

# Spis tablic