

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, wrzesień 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 29 września 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 29 września 2019

.....

(podpis promotora)

Spis treści

1	Wstęp	1
2	Metoda bazowa - wrinkle feature	3
2.1	Metoda wykrywania twarzy	4
2.1.1	Wstęp do algorytmu Haar Cascade	5
2.1.2	Konwersja do skali szarości	6
2.1.3	Kontynuacja opisu algorytmu Haar Cascade	8
2.2	Wyznaczanie stref	11
2.3	Wykrywanie zmarszczek - detektor Canny	11
2.4	Wyliczanie wrinkle feature	11
2.5	Algorytm trenowania	11
2.6	Grupowanie danych - FCM	12
2.6.1	Wstęp do grupowania danych	12
2.6.2	Metoda FCM	12
2.7	Wyznaczanie wieku	12
3	Modyfikacje metody bazowej	13
3.1	Odjęcie wybranej strefy	13
3.1.1	Zmiana algorytmu względem metody bazowej	13
3.2	Zastosowanie metody HOG	13
3.2.1	Opis algorytmu HOG	13
3.2.2	Zastosowanie w projekcie	13
3.3	Metoda HOG oraz grupowanie KNN	13
3.3.1	Grupowanie KNN	13
3.3.2	Zastosowanie w projekcie	13

4	Badania	15
5	Podsumowanie	17

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. [6]

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 [10].

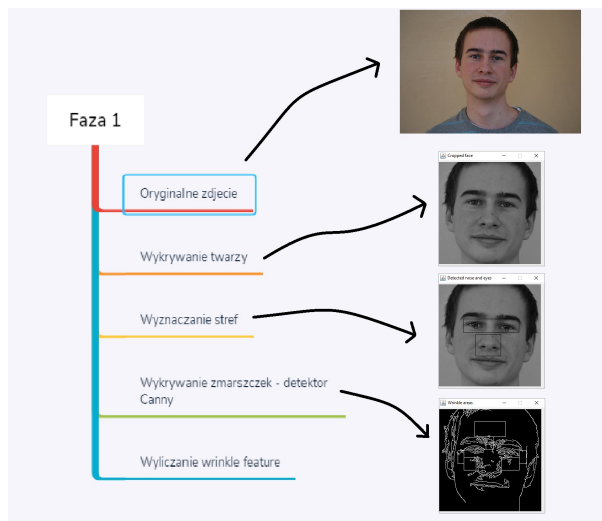
Rozdział 2

Metoda bazowa - wrinkle feature

Istnieje wiele metod wyznaczania wieku z obrazu twarzy. Każda z nich opiera się na wykrywaniu zmarszczek, zmian koloru skóry lub proporcji twarzy. W omawianej pracy zastosowano metodę korzystającą z wykrywania zmarszczek. [3]

Metoda bazowa została opisana w artykule „Age Estimation from Face Image using Wrinkle Features” [9]. 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 [9] 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 2.3 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 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. Niektóre z nich do skutecznej kwalifikacji stosują ekstrakcję cech z obrazu. Ekstrakcja

polega na przekształceniu obrazu do zbioru zmiennych, które zostaną później użyte w wykrywaniu obiektu lub obiektów na obrazie. Warto wymienić najczęściej stosowane metody służące do ekstrakcji cech [10]:

- Filtry Gabora
- Filtry Haara
- Funkcje jądra
- Analiza składowych górnych

Algorytm Haar Cascade jest najpopularniejszym algorytmem do wykrywania twarzy w bibliotece OpenCV. Biblioteka OpenCV została tutaj wymieniona, ponieważ wykorzystano ją przy wykrywaniu wieku z tekstury twarzy. W związku z powyższym do wykrywania twarzy zastosowano algorytm Haar Cascade.

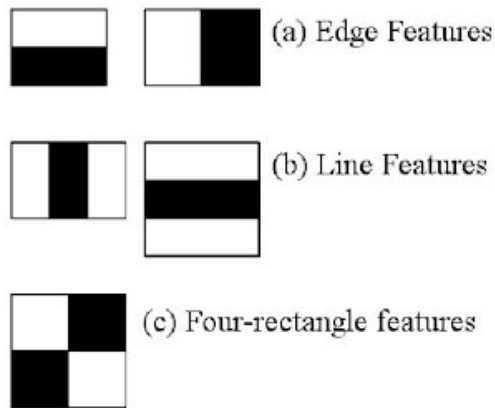
2.1.1 Wstęp do algorytmu Haar Cascade

Haar Cascade jest algorytmem z dziedziny uczenia maszynowego służącym do wykrywania obiektów w wizji komputerowej. Został stworzony przez Paula Viola oraz Michaela Jonsa [8]. Algorytm opiera się na zbudowaniu kaskadowej funkcji za pomocą trenowania wielu zdjęć. Zdjęcie są dzielone na dwie kategorie - pozytywne oraz negatywne. Na pozytywnych zdjęciach istnieje obiekt, który ma zostać wykryty. Analogicznie na negatywnych zdjęciach nie ma tego obiektu.

Ekstrakcja cech w algorytmie Violi i Jonsa jest realizowana przez filtry Haara. Filtry Haara to prostokątne okienka nakładane na obraz, które analizują jasność pikseli 2.3.

Przed zastosowaniem filtru Haara obraz musi zostać przekształcony do skali szarości. Skala szarości jest rodzajem odwzorowania koloru. Wyróżnia się 3 podstawowe tryby koloru:

- kolorowy
- monochromatyczny (skala szarości)



Rysunek 2.3: Filtr Haara a) krawędziowy b) liniowy c) szachownica [1]

- czarno-biały

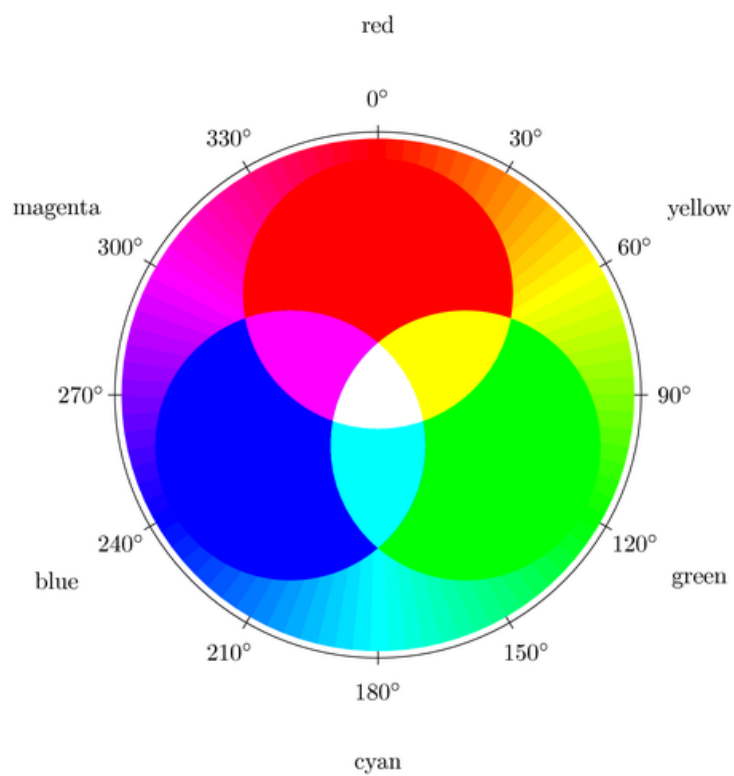
W niniejszej pracy należało przekształcić każdy obraz z trybu kolorowego na monochromatyczny.

2.1.2 Konwersja do skali szarości

Każdy piksel w trybie kolorowym ma określoną reprezentację barwy z określonego modelu. Najczęściej spotykana reprezentacja to RGB. Przestrzeń kolorów RGB to złożenie 3 kanałów [4]:

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

Każdy piksel opisany za pomocą przestrzenie barw RGB ma 3 8-bitowe wartości reprezentujący każdy kanał. Spotykane są reprezentacje 12 lub 16 bitowe kanałów, jednak 8 bitowa jest najpopularniejsza. Dla 8-bitowych kanałów wartość "0" danego kanału oznacza brak jasności, natomiast "255" maksymalną jasność. Poprzez mieszanie jasności 3 kanałów można uzyskać szerokie spektrum barw 2.4.



Rysunek 2.4: Mieszanie kanałów RGB. [5]



Rysunek 2.5: Kolor R=153 G=217 B=234.

Przykładowo kolor o reprezentacji $R=153$ $G=217$ $B=234$ przedstawiono na 2.5

Kolor 2.5 może być też reprezentowany w kodzie szesnastkowym #99D9EA. Każda wartość heksadecymalna odpowiada kolejno kanałowi R, G, B.

Istnieją 3 metody konwersji obrazu z przestrzeni RGB na monochromatyczny [5].

- największej jasności
- średnia
- luminancji

Metoda największej jasności konwertuje na skalę szarości wg wzoru 2.1.

$$\frac{(\max(R, G, B) + \min(R, G, B))}{2} \quad (2.1)$$

Metoda średnia bazuje na wzorze 2.2. Natomiast luminancji na wzorze 2.3.

$$\frac{(R + G + B)}{3} \quad (2.2)$$

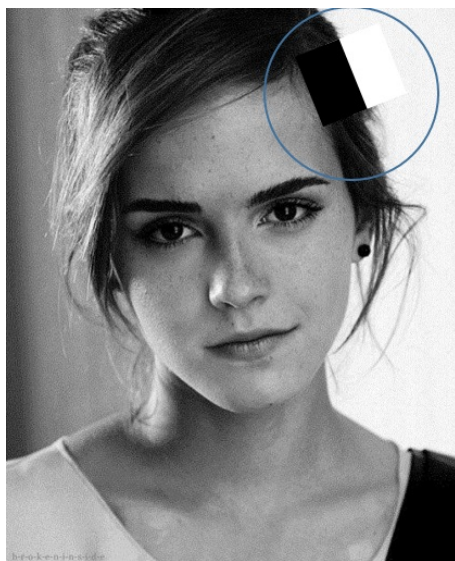
$$0.21R + 0.72G + 0.07B \quad (2.3)$$

W pracy zastosowano konwersję za pomocą metody średniej.

2.1.3 Kontynuacja opisu algorytmu Haar Cascade

Każde okienko posiada białe oraz czarne prostokąty. Wyznaczana jest suma jasności pikseli w białych prostokątach oraz czarnych. Następnie dla każdego okna obliczana jest różnica pomiędzy białymi a czarnymi kwadratami.

Opisywany filtr ma zastosowanie w wykrywaniu krawędzi. Na granicy krawędzi istnieje różnica w jasności pikseli 2.6. Im większa wartość dla danego okna tym większe prawdopodobieństwo, że została wykryta krawędź.



Rysunek 2.6: Filtr Haara nałożony na krawędź twarzy [1]

Z każdego obrazu podczas treningu klasyfikatora otrzymujemy dużą ilość danych. W celu poprawy efektywności sumowania pikseli w filtrze Haara stosowane są rozwiązania zwane po angielsku Summed-area table [7]. 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 2.4.

$$I(x, y) = \sum_{x' \leq x \cap y' \leq y} i(x', y') \quad (2.4)$$

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 2.7 przedstawiona jest tabela przedstawiająca jasność pikseli przed zastosowaniem całkowania obrazu.

Po całkowaniu otrzymujemy tabele tj. na Rysunku 2.8.

Sumowanie przykładowego okna (Rysunek 2.9) wymaga 4 operacji (Wzór 2.5).

Image

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

Rysunek 2.7: Tabela jasności poszczególnych pikseli przed zastosowaniem całkowania [2]

Summed Area Table

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

Rysunek 2.8: Filtr Haara nałożony na krawędź twarzy [2]

Summed Area Table

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

A B
C D

Rysunek 2.9: Sumowanie okna [2]

$$\sum_{x_0 \leq x \leq x_1 \cap y \leq y \leq y_1} i(x, y) = I(D) + I(A) - I(B) - I(C) \quad (2.5)$$

gdzie lewa część równania oznacza sumę jasności pikseli zaznaczonego okna tj. na Rysunku 2.9, $I(A)$ - wartość scałkowanego obrazu przy punkcie A (analogicznie $I(B)$, $I(C)$, $I(D)$) - Rysunek (2.9).

Zakładając większy rozmiar okna np. 9×9 sumowanie okna zwykłym sposobem wymagałoby $9 \times 9 = 81$ operacji. Natomiast używając obrazu scałkowanego liczba operacji jest stała i wynosi dokładnie 4.

-----//pierdolenie na koniec rozdziału

Biblioteka OpenCV posiada wytrenowane klasyfikatory, które zostały użyte w pracy magisterskiej. Użyto klasyfikatorów do wykrycia twarzy, ust oraz oczu.

2.2 Wyznaczanie stref

sdfsd

2.3 Wykrywanie zmarszczek - detektor Canny

sdfsd

2.4 Wyliczanie wrinkle feature

sdfsd

2.5 Algorytm trenowania

sfsd

2.6 Grupowanie danych - FCM

sfsd

2.6.1 Wstęp do grupowania danych

2.6.2 Metoda FCM

sdfsd

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] 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, 2019. [Online; Dostęp: 28.09.19].
- [2] BADGERATI. <https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/>, 2010. [Online; Dostęp: 28.09.19].
- [3] M.H. Yang D. Kriegman, N. Ahuja. *Detecting faces in images: a survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.
- [4] Robert Hirsch. *Exploring Colour Photography: A Complete Guide*. Laurence King Publishing, 2004.
- [5] Henrik Skov Midtiby. Example: Rgb color mixing, 2019. [Online; Dostęp: 28.09.19].
- [6] Moses Olafenwa. Przykład rozpoznawania obrazów. <https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606/>, 2018. [Online; Dostęp: 30.09.18].
- [7] M. Jone P. Viola. *Robust Real-Time Object Detection*. International Journal of Computer Vision, 2002.
- [8] Michael Jones Paul Viola. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Accepted conference on computer vision and pattern recognition, 2001.

- [9] Rituparna Sahaa Ranjan Janaa, Debaleena Dattaa. *Age Estimation from Face Image using Wrinkle Features*. RCC Institute of Information Technology, 2014.
- [10] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.

Dodatki

Dokumentacja techniczna

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

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

N liczebność zbioru danych

μ stopień przynależności do zbioru

\mathbb{E} zbiór krawędzi grafu

\mathcal{L} transformata Laplace’a

X

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. [6]	2
2.1	Faza 1 algorytmu	4
2.2	Faza 2 algorytmu	4
2.3	Filtr Haara a) krawędziowy b) liniowy c) szachownica [1] . . .	6
2.4	Mieszanie kanałów RGB. [5]	7
2.5	Kolor R=153 G=217 B=234.	7
2.6	Filtr Haara nałożony na krawędź twarzy [1]	9
2.7	Tabela jasności poszczególnych pikseli przed zastosowaniem całkowania [2]	10
2.8	Filtr Haara nałożony na krawędź twarzy [2]	10
2.9	Sumowanie okna [2]	10

Spis tablic