

**POLITECHNIKA WROCŁAWSKA**  
**WYDZIAŁ ELEKTRONIKI**

---

KIERUNEK: Informatyka (INF)  
SPECJALNOŚĆ: Systemy informatyki w medycynie (IMT)

**PRACA DYPLOMOWA  
INŻYNIERSKA**

Rozpoznawanie emocji na podstawie ekspresji  
twarzys

Emotion recognition from facial expressions

AUTOR:  
Łukasz Korycki

PROWADZĄCY PRACĘ:  
Prof. dr hab. inż. Marek Kurzyński,  
Katedra Systemów i Sieci Komputerowych

OCENA PRACY:

# Spis treści

<b>1 Wprowadzenie</b>	<b>3</b>
1.1 Interakcja człowiek-komputer . . . . .	3
1.2 Rozpoznawanie emocji . . . . .	5
1.3 Schemat przetwarzania ekspresji twarzy . . . . .	7
1.4 Przykłady istniejących rozwiązań . . . . .	9
1.5 Cel i zakres pracy . . . . .	13
<b>2 Rejestracja obrazu twarzy</b>	<b>15</b>
2.1 Obraz wejściowy . . . . .	15
2.2 Detekcja twarzy . . . . .	16
2.2.1 Kaskadowy klasyfikator cech Haara . . . . .	16
2.2.2 Wykorzystanie detektora . . . . .	18
2.2.3 Wpływ rozdzielczości na szybkość przetwarzania . . . . .	20
<b>3 Ekstrakcja punktów charakterystycznych twarzy</b>	<b>21</b>
3.1 Ogólny schemat przetwarzania . . . . .	21
3.2 Wyznaczanie zgrubnych obszarów . . . . .	22
3.2.1 Kaskada detektorów . . . . .	22
3.2.2 Statyczne obszary . . . . .	23
3.3 Ekstrakcja punktów . . . . .	26
3.3.1 Oczy . . . . .	26
3.3.2 Brwi . . . . .	29
3.3.3 Usta . . . . .	31
3.3.4 Zęby . . . . .	33
3.3.5 Nos . . . . .	34
3.4 Opis geometryczny ekspresji twarzy . . . . .	35
<b>4 Klasyfikacja emocji</b>	<b>37</b>

4.1	Zadanie klasyfikacji . . . . .	37
4.2	Wykorzystane klasyfikatory . . . . .	38
4.2.1	Sztuczne sieci neuronowe . . . . .	39
4.2.2	SVM . . . . .	42
4.2.3	Algorytm k-NN . . . . .	45
4.2.4	Zespół klasyfikatorów . . . . .	45
<b>5</b>	<b>Badanie klasyfikatorów</b>	<b>47</b>
5.1	Zbiory danych . . . . .	47
5.2	Plan eksperymentu . . . . .	48
5.3	Wyniki . . . . .	49
5.3.1	Podejścia <i>user-dependent</i> i <i>user-independent</i> . . . . .	50
5.3.2	Testy dla sekwencji obrazów . . . . .	52
5.4	Ocena skuteczności . . . . .	57
5.5	Przykłady . . . . .	59
<b>6</b>	<b>Podsumowanie</b>	<b>64</b>
6.1	Aktualny stan . . . . .	64
6.2	Perspektywy rozwoju . . . . .	65
	<b>Bibliografia</b>	<b>65</b>

# Rozdział 1

## Wprowadzenie

Pierwszy rozdział niniejszej pracy inżynierskiej stanowi wprowadzenie do zagadnienia rozpoznawania emocji. Przedstawia on kontekst realizowanego projektu, wyjaśniając kluczowe dla tematu pojęcia – od ogólnej *interakcji człowiek-komputer* po subtelny w znaczeniu *afekt*. Ów opis ukierunkowany jest na rozpoznawanie emocji w oparciu o ekspresje mimiczne – wynika to oczywiście z przyjętej przez autora konkretnej metody. Poza kontekstem pojęciowym autor prezentuje również już istniejące, opisane rozwiązania poruszanego problemu, przechodząc następnie do własnej koncepcji.

### 1.1 Interakcja człowiek-komputer

**Interakcja człowiek-komputer** (ang. *human-computer interaction*, HCI) to szeroko rozwinięte pojęcie, traktowane często jako interdyscyplinarna nauka zajmująca się badaniem sposobów komunikacji pomiędzy ludzkim użytkownikiem a maszyną. Obejmuje wiele zagadnień związanych z projektowaniem interfejsów, użytecznością systemów, czy tzw. *user experience* – tworzeniem produktów spełniających wymagania użytkownika. Od początków istnienia tego terminu, sięgających lat 80. XX wieku [1], usystematyzowanych zostało wiele zasad i pojęć jego dotyczących. Fundamentalnym wydaje się stwierdzenie, że systemy komputerowe komunikujące się z człowiekiem powinny cechować się **funkcjonalnością i użytecznością** [2]. Pierwsze daje nam możliwości, ale to drugie zapewnia efektywne ich wykorzystanie. Oba te czynniki są dostosowywane do potencjalnej aktywności użytkownika, mogącej odbywać się na trzech poziomach: fizycznym (mechanika interakcji), kognitywnym (zrozumienie systemu) oraz afektywnym (odczuć) [3].

Interakcja pomiędzy człowiekiem i maszyną dynamicznie ewoluje wraz z rozwojem możliwości aplikacji komputerowych. Kiedyś przełomowym było zastosowanie myszy komputerowej i okienkowego systemu operacyjnego. Dziś nowy kontroler czy przezroczysty interfejs graficzny może już nie wystarczyć. Dlaczego? Ponieważ z ludzkiego punktu widzenia horyzontem rozwoju interakcji pomiędzy człowiekiem i maszyną jest relacja możliwie naturalna i intuicyjna, nieograniczona sztucznymi warunkami technologicznymi, spontaniczna. Wraz z postępem oczekujemy od systemów możliwości komunikowania się z nimi przy pomocy mediów bardziej dla nas bezpośrednich, takich jakich używamy do porozumiewania się z innymi ludźmi. Aby się wzajemnie rozpoznawać, nie potrzebujemy loginu tylko obrazu twarzy drugiej osoby. Aby

poprosić o wykonanie czegoś, nie używamy słów pisanych na klawiaturze, zamiast tego wydajemy komendę głosową. Prędzej wskażemy coś palcem czy ruchem dłoni, niż użyjemy do tego kurSORA myszy komputerowej. Mnogość rozwijanych technologii sprawia, że coraz częściej mówi się o interfejsach adaptacyjnych czy inteligencji środowiskowej. Niemal oczywistym wydaje się, że rozwój interfejsów człowiek-komputer ukierunkowany jest w stronę rozpoznawania przez maszynę naturalnych dla nas sposobów zachowywania się i wyrażania.



Rysunek 1.1: Przykłady zaawansowanej interakcji człowiek-komputer: Kinect 2 z funkcją śledzenia ruchów ciała i rozpoznawania gestów oraz komendy głosowe we wnętrzu samochodu. Źródła: [123kinect.com](http://123kinect.com) i [arstechnica.com](http://arstechnica.com)

Współczesne systemy HCI mogą posiadać różną architekturę. Różnią się one między innymi pod względem używanych kanałów transmisji informacji. W pierwszej kolejności mówimy o rozwiązaniach **unimodalnych**, czyli takich, które korzystają wyłącznie z jednego medium przekazu. Dzielimy je na trzy podstawowe kategorie [3]:

- wizyjne – najbardziej rozbudowana grupa systemów, opierających swoje działanie na informacji obrazowej, wykorzystywana w przypadku wielu różnych problemów, np. do śledzenia ruchów ciała, rozpoznawania gestów, czy wspomnianego rozpoznawania twarzy,
- dźwiękowe (audio) – systemy przetwarzające komunikaty w postaci sygnałów dźwiękowych, znalazły zastosowanie przy zadaniach z zakresu rozpoznawania mowy lub mówcy oraz w interakcji muzycznej,
- sensoryczne – wspomagane fizycznymi sensorami, takimi jak np. sensory ruchu, ciśnienia, zapachu, a także kontrolery komputerowe (mysz, klawiatura, pad), współcześnie częściowo wypierane na rzecz systemów audio-wizyjnych.

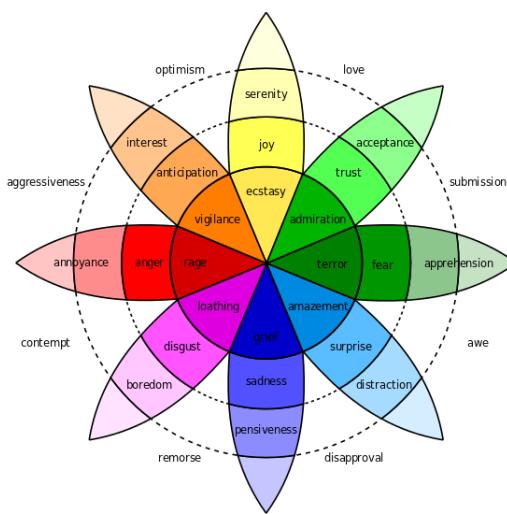
Drugim rodzajem omawianej architektury są rozwiązania **multimodalne**, które nie ograniczają się wyłącznie do jednego kanału, a wykorzystują informacje pochodzące z co najmniej dwóch różnych. Przykładem może być system rozpoznający mowę na podstawie sygnału głosowego oraz śledzonego kształtu ust [4]. Ponieważ ludzka komunikacja multimodalna działa na zasadzie komplementarności źródeł, nie wydaje się być dobrym pomysłem traktowanie informacji z różnych kanałów jako niezależnych. Bardziej kompleksowym i ciekawym podejściem jest rozpatrywanie danych we wspólnej, połączonej przestrzeni cech. Oczywiście generuje to szereg problemów związanych z jej rozmiarami, formatem danych czy synchronizacją czasową [5].

## 1.2 Rozpoznawanie emocji

Kiedy mówimy o konieczności bardziej naturalnej komunikacji z człowiekiem, nie da się pominąć kwestii rozumienia jego emocji. Ich rozpoznawanie już dawno przestało być zagadnieniem ograniczonym wyłącznie do sterylnych warunków laboratoryjnych i znalazło pionierskie zastosowanie w wielu komercyjnych aplikacjach użytkowych [6, 7]. Rozpoznawanie emocji jest wykorzystywane od medycyny (terapie psychiatryczne, leczenie autyzmu, reakcje na rehabilitację), przez robotykę (roboty społeczne), po rozrywkę (gry, dopasowywanie muzyki) i marketing (odczucia potencjalnych klientów). Mimo znaczącego postępu w tej dziedzinie wciąż pojawia się wiele istotnych niejednoznaczności w kwestiach koncepcyjnych i nawet więcej ograniczeń po stronie technicznej. Okazuje się, że problemem może być już odpowiedzenie na podstawowe pytanie: co tak naprawdę dokładnie chcemy, aby maszyny były w stanie rozpoznawać? Jakie emocje mają one rozróżnić?

Psychologowie, antropolodzy, socjolodzy a nawet filozofowie zajmują się tym od lat. Arystoteles twierdził, że należy wyróżnić 14 emocji [8]. Wskazywał on jak wpływają one na ludzką moralność i wybory, dostrzegał połączenie między ciałem, umysłem i emocjami.

Amerykański psycholog Robert Plutchik, na podwalinach prac Karola Darwina [9], określił emocje jako prymitywy biologiczne, wytworzone na drodze ewolucyjnej adaptacji zwierząt (w tym człowieka), zwiększające szanse przetrwania. W 1980 roku opublikował swoją teorię, w której przedstawił koło – znane później jako koło Plutchika – złożone z 8 elementarnych, związanych ze sobą emocji o różnych intensywnościach [10].

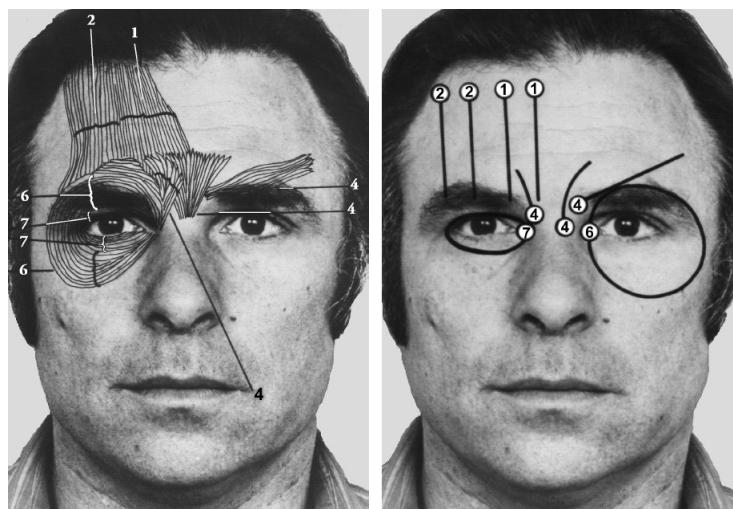


Rysunek 1.2: Koło Plutchika. Źródło: [study.com](http://study.com)

W modelu tym mają one charakter bipolarny, tzn. każda posiada swoje przeciwwieństwo, jak np. smutek-radość, zaufanie-odraza. Ponadto emocje podstawowe mogą łączyć się ze sobą, tworząc emocje pochodne, odczuwane częściej lub rzadziej w zależności od intensywności. I tak np. kombinacja zaskoczenia i smutku daje w rezultacie zawód, rozczarowanie. Sąsiedztwo emocji i ich barwy nie są przypadkowe – odzwier-

ciedlają one stopień ich wzajemnego powiązania.

Bardziej anatomiczne podejście zaproponował inny amerykański psycholog – Paul Ekman. Jego pionierskie badania w zakresie ruchowo-mimicznej ekspresji emocji doprowadziły do stworzenia ich definicji w oparciu wyłącznie o zmiany położenia mięśni twarzy. Odpowiednie reguły zostały zgromadzone w dokumencie opisującym autorski *System Kodowania Akcji Mimicznych* (ang. *Facial Action Coding System*, FACS) [11]. Opierają się one o niepodzielne jednostki akcji (ang. *action units*, AU), odnoszące się do spięcia lub rozluźnienia poszczególnych mięśni oraz ich grup. Dodatkowo poszczególne ruchy mogą mieć różną intensywność w skali od najsłabszego A do najwyraźniejszego E. Przykładowo, aby stwierdzić zajście ekspresji zaskoczenia, potrzebujemy kombinacji: AU 1 (podniesienie wewnętrznej części brwi), AU 2 (podniesienie zewnętrznej części brwi), AU 5B (co najmniej niewielkie podniesienie górnej powieki oka) oraz AU 26 (obniżenie szczelek).

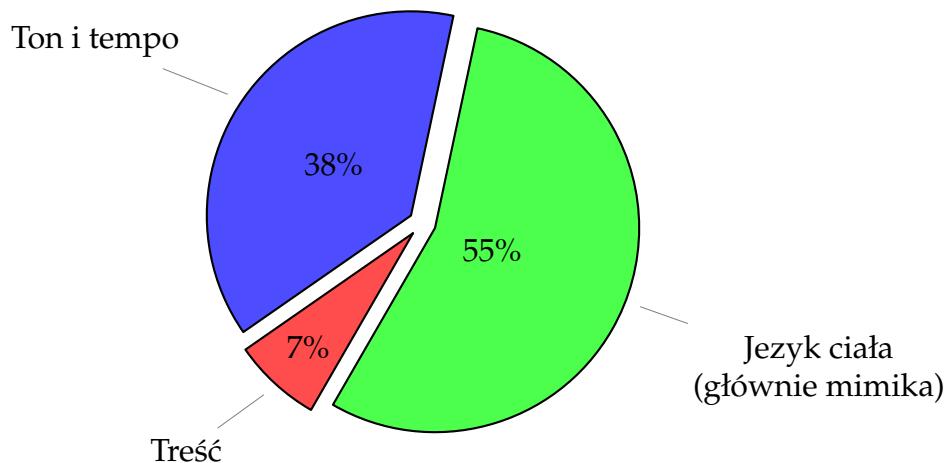


Rysunek 1.3: Oznaczenia niektórych mięśni twarzy (po lewej) i akcje mimiczne (po prawej). Źródło: [paulekman.com](http://paulekman.com)

Tym sposobem, badając reakcje ludzi na całym świecie, Ekman wyróżnił 6 podstawowych emocji, uniwersalnych dla rodzaju ludzkiego, niezależnie od rasy i pochodzenia. Są to: **szczęście, smutek, zaskoczenie, strach, złość i odraza (zniesmaczenie)**. W późniejszym okresie zbiór ten został uzupełniony o jeszcze jeden stan – pogardę. Jednak nawet i to, zdawałoby się precyzyjne podejście, nie jest do końca pewne. Istnieją teorie silnie korelujące strach i zaskoczenie oraz złość i zniesmaczenie, jako emocje parami aktywujące bardzo podobne mięśnie twarzy. Nie pozostaje to bez znaczenia przy próbach automatycznego ich rozróżniania.

Zdolność do rozpoznawania emocji – jakiekolwiek by one nie były – czyli umiejętność ich rozróżniania, etykietowania i wykorzystywania zdobytej informacji emocjonalnej, to tzw. **inteligencja emocjonalna** [12]. Na bazie czego możemy ją wykorzystywać? Co stanowi medium ekspresji emocji u człowieka? Skąd czerpać informacje o nich, aby móc podjąć próbę rozpoznania? Temat niniejszej pracy dyplomowej czy nadmieniona wcześniej praca Ekman'a wskazują na jedno z najbardziej oczywistych źródeł – mimikę. Nie jest to jednak jedyna odpowiedź. Ważnym kanałem przekazu jest mowa człowieka. Może być ona analizowana pod względem parametrycznym (ton głosu, szybkość mówienia) jak i semantycznym (znaczenie słów, kontekst zdań). Nie bez znaczenia jest także mowa pozostałych części ciała oraz całej jego postawy. Emocje

mogą być również wykrywane w bardziej bezpośredni i inwazyjny sposób poprzez analizę różnego rodzaju biosygnałów. Przykładem niech będą próby rozróżniania stanów emocjonalnych człowieka na podstawie zapisu EEG [13]. Ze wszystkich możliwości najlepszym z praktycznego punktu widzenia źródłem okazuje się być wspomniana jako pierwsza mimika. Wynika to nie tylko z empirycznych badań implementacji komputerowych, ale również z silnego podłożu teoretycznego i statystycznego. Należy w tym miejscu wspomnieć postać Alberta Mehrabiana – amerykańskiego psychologa zajmującego się komunikacją niewerbalną, w szczególności tą emocjonalną. Z jego badań wynika, że dominujący w przekazie emocji jest język ciała, w którym **główną rolę odgrywa ekspresja twarzy** [14]. Znaczący jest również ton i tempo wypowiadanych słów oraz samo ich znaczenie. Podkreślał on ponadto, że w przypadkach sprzeczności pomiędzy przekazem werbalnym i niewerbalnym istotniejsze są komunikaty drugiego rodzaju.



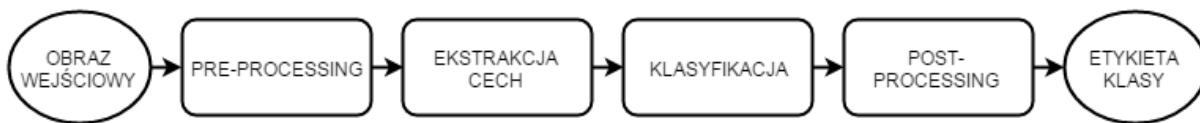
Rysunek 1.4: Reguła 55-38-7 Mehrabiana dotycząca procentowego udziału poszczególnych źródeł informacji w komunikacji emocjonalnej

Przytoczone fundamenty teoretyczne zagadnienia rozpoznawania emocji pokazują mnogość założeń koncepcyjnych, które można przyjąć przy projektowaniu systemów zdolnych do wykonywania tego zadania. Podstawowym etapem jest wybór emocji, które chcemy rozpoznawać oraz określenie sposobu ich definiowania na podstawie rozważanych źródeł. Dominującym podejściem, dającym najlepsze wyniki w najbardziej ogólnej grupie przypadków, jest wykorzystywanie mimiki. Warto w tym miejscu wspomnieć, że tym, co w istocie rozpoznajemy, nie jest emocja sama w sobie, ale jej ekspresja zwana **afektem** [15]. Dalsza część pracy skupia się już wyłącznie na systemach analizujących ów afekt.

### 1.3 Schemat przetwarzania ekspresji twarzy

Zdecydowana większość systemów analizujących mimikę działa na podobnych zasadach [16]. Współdzielą one wiele architektur i technik przetwarzania z systemami dokonującymi rozpoznawania twarzy – duża część cech wykrywanych w jednych z nich może posłużyć za istotną informację w drugich. Systemy budowane są zazwyczaj jako

sekwencyjnie połączone procesy, które zostały zaprezentowane na poniższym diagramie.



Rysunek 1.5: Podstawowy schemat przetwarzania ekspresji twarzy

Warto zauważyć, że schemat ten jest charakterystyczny nie tyle dla metod rozpoznawania twarzy i emocji, co dla ogólniejszej techniki **rozpoznawania wzorców** (ang. *pattern recognition*) – zmienne jest praktycznie tylko wejście systemu. W omawianym modelu wykonywanych jest pięć podstawowych operacji.

1. **Akwizycja obrazu** – pobierany jest obraz mający docelowo zawierać twarz użytkownika. Może on być pojedynczy lub stanowić część sekwencji wideo oraz posiadać różne formaty. Najczęściej używanymi są obrazy monochromatyczne (wydajność przetwarzania, utrata informacji barwnej), RGB (mniejsza wydajność, obecność informacji barwnej), a także obrazy w bliskiej podczerwieni (zwiększena odporność na warunki oświetleniowe, utrata dużej ilości informacji o barwach i szczegółach obiektów).
2. **Pre-processing** – czyli przetwarzanie wstępne obrazu. Proces ten jest odpowiedzialny za *oczyszczenie* danych wejściowych, dostosowywanie ich do takiej postaci, która będzie mogła być wydajnie analizowana przez system. Wiąże się to zazwyczaj z usuwaniem szumów z obrazu (filtracje dolno- i górnoprzepustowe), jego transformacjami (skalowanie, obracanie, przesuwanie) czy zawężaniem obszarów analizy (ang. *region of interest*, ROI). Ponadto dokonywana jest segmentacja obrazu, tj. tworzenie jednorodnych obszarów reprezentujących obiekty o poszukiwanej charakterystyce (np. oczy, usta).
3. **Ekstrakcja cech** – transformacja obrazu do postaci opisanej na wyższym poziomie abstrakcji. Zazwyczaj przyjmuje on formę wektora określającego geometryczne cechy analizowanej ekspresji twarzy lub zawierającego informacje o zajęciu poszczególnych akcji mimicznych (AU). Może on również dotyczyć innych aspektów, jak np. kinetyki zmian (ang. *optical flow*) czy różnego rodzaju przekształceń, np. do lokalnych wzorców binarnych (ang. *local binary patterns*, LBP). Etap ten związany jest też często z redukcją cech, czyli zmniejszeniem wymiarowości wektora opisującego twarz. W przypadku przetwarzania obrazu ma to fundamentalne znaczenie dla szybkości działania systemu.
4. **Klasyfikacja** – podejmowanie decyzji o przynależności analizowanej ekspresji, opisanej uzyskanym w poprzednim kroku wektorem, do jednej z wyszczególnionych emocji (klasy). Proces ten związany jest z obliczaniem funkcji wsparć dla poszczególnych kategorii przez odpowiednio skonstruowany algorytm klasyfikujący.
5. **Post-processing** – operacje mające na celu zwiększenie wydajności rozpoznawania, np. poprzez zastosowanie kodów korekcyjnych Hamminga dla problemów z wyjściem w kodzie binarnym.

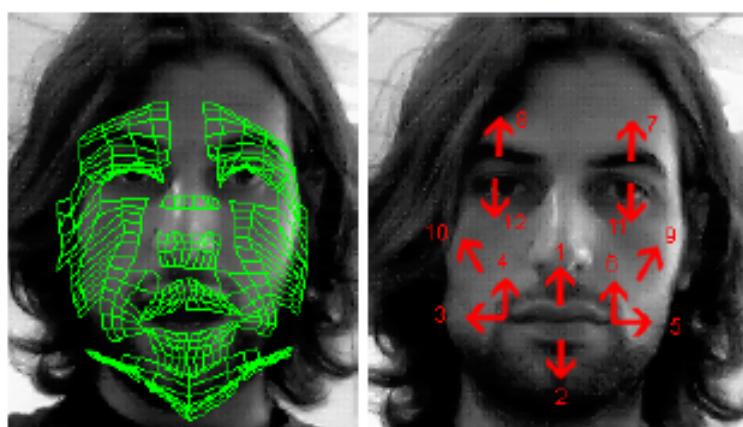
Implementacja poszczególnych procesów jest ukierunkowana na uzyskiwanie możliwie wydajnego rozpoznawania zarówno pod względem jakościowym jak i czasowym. Obecne rozwiązania są jednak silnie uzależnione od warunków, w których system musi pracować. Ze względu na zmienne warunki oświetleniowe, różne pozycje twarzy i dużą ich różnorodność problematyczne okazuje się stworzenie uniwersalnych rozwiązań [17].

## 1.4 Przykłady istniejących rozwiązań

Systemy rozpoznawania emocji na podstawie ekspresji twarzy różnią się zasadniczo we dwóch najważniejszych procesach: ekstrakcji cech i klasyfikacji. Jednak także i w nich znaleźć można pewne popularne i intensywnie rozwijane podejścia. Zdaniem autora warto przedstawić nie tylko kompletne systemy wykonujące postawione w temacie zadanie, ale również i biblioteki odpowiedzialne za pozyskiwanie wektora cech. Jest to bowiem najbardziej charakterystyczny aspekt tego rodzaju systemów, poruszany w licznych pracach badawczych [18, 19, 20].

### InSight SDK

System ten wraz ze środowiskiem deweloperskim powstał na bazie pracy [17]. Detekcja punktów charakterystycznych twarzy została oparta o metodę śledzenia częstokwadrnych deformacji wolumenów powierzchni Bezier'a (ang. *Piecewise Bezier Volume Deformation*, PBVD) opisaną w [21]. Na początku użytkownik ręcznie dopasowuje trójwymiarowy model neutralnej twarzy złożony z 16 takich powierzchni. Od tego momentu system jest w stanie automatycznie dostosować położenie modelu, dzięki czemu zachodzi możliwość śledzenia ruchów twarzy. Mierzone zmiany położenia punktów siatek w przestrzeni 2D służą do estymacji rzeczywistych ruchów 3D. Ruchy poszczególnych części twarzy są opisywane przy pomocy wektora jednostek ruchowych (ang. *Motion Units*, MU), przypominających jednostki akcji (AU) Ekman'a.



Rysunek 1.6: Śledzenie ruchów twarzy przy pomocy metody PBVD. Źródło: [17]

System jest w stanie rozpoznawać 6 rodzajów emocji wyróżnionych przez nadmienionego psychologa oraz stan neutralny. Klasyfikacja dokonywana jest przy pomocy

naiwnego klasyfikatora bayesowskiego, konstruowanego na bazie rozkładów prawdopodobieństwa każdej z cech:

$$\hat{y} = \arg \max_y p(y|\mathbf{x}) = \arg \max_y (p(y) \prod_{i=1}^n p(x^{(i)}|y)), \quad (1.1)$$

gdzie  $y$  to numer klasy,  $\mathbf{x}$  to wektor  $n$  cech  $x^{(i)}$ , a  $p$  oznacza prawdopodobieństwo. Obiekt jest etykietowany klasą o największym prawdopodobieństwie *a posteriori*. Mimo że klasyfikator ten zakłada niezależność cech twarzy (co wydaje się sprzeczne z intuicją, gdy rozważamy ruchy sąsiednich punktów twarzy), okazuje się być zaskakująco skuteczny dla stawianego przed nim zadania. Średnia skuteczność rozpoznawania wynosi: 93,2% dla jednego użytkownika, ale już tylko 63,4% dla większej liczby różnych twarzy<sup>1</sup>. Obrazuje to wspomniany wcześniej problem z generalizacją tworzących modeli systemów. W prezentowanych zestawieniach wyników widoczne jest, że najwięcej błędów generują pary złość-odraza oraz zaskoczenie-strach, co również było już poruszane.

System Insight SDK został mocno rozbudowany w ramach działalności firmy SightCorp [22]. Aktualnie jest on w stanie automatycznie wykrywać twarze użytkowników i dopasowywać do nich modele. Zaimplementowana została również możliwość równoległego analizowania grupy ludzi (funkcja *CrowdSight*). System jest ponadto zdolny m.in. do estymacji wieku, określania płci, pozycji głowy, czy przynależności etnicznej.

## IntraFace

To aplikacja, której rdzeń stanowią techniki ekstrakcji i klasyfikacji opisane w pracach [23, 24]. Pierwsza z nich polega na dopasowywaniu dwuwymiarowego modelu twarzy. Jest to zadanie z zakresu optymalizacji, w którym minimalizowana jest nielinowa funkcja najmniejszych kwadratów (ang. *Non-linear Least Squares*, NLS). Teoretycznie najlepszymi metodami do tego zadania są metody różniczkowe drugiego rzędu (np. metoda Newtona). Jednak w przypadku problemu komputerowej wizji pojawiają się dwie zasadnicze wady: funkcja może nie być różniczkowalna analitycznie oraz obliczane wartości hesjanu mogą okazać się nieskończone. Z tego powodu autorzy projektu zaproponowali alternatywne rozwiązanie przy pomocy własnej metody nazwanej nadzorowanym spadkiem (ang. *Supervised Descent Method*, SDM) [23]. Jej zaletą jest to, że po odpowiednim treningu, w trakcie którego uczy się ona dobierania kierunków minimalizacji funkcji NLS, eliminowana jest konieczność przeprowadzania wspomnianych obliczeń. Metoda okazała się być wysoce skuteczna i odporna.

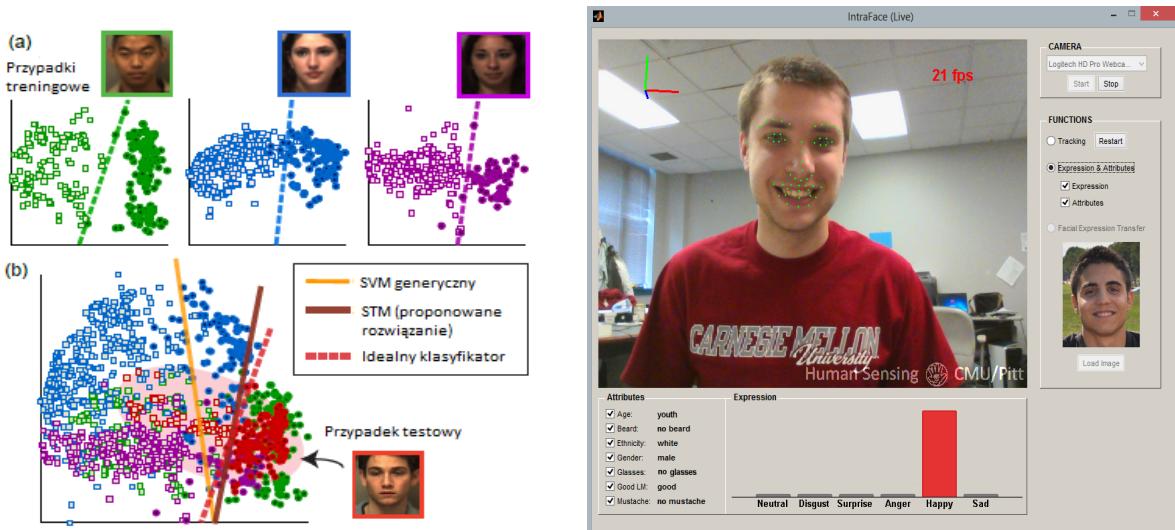
Sama klasyfikacja emocji odbywa się na podstawie wykrywania zajścia poszczególnych akcji mimicznych. Dla każdej z nich kreowana jest osobna maszyna wektorów nośnych (ang. *Support Vector Machine*, SVM). Jest to popularne podejście ze względu na

<sup>1</sup>Warto zaznaczyć, że próby tworzenia sieci bayesowskiej z połączonymi węzłami cech nie przyniosły lepszych rezultatów, odpowiednio: 53,8% oraz 62,1%.

binarny charakter działania owego klasyfikatora (akcja zaszła lub nie). Autorzy również i przy tym zadaniu zdecydowali się na modyfikację tradycyjnego podejścia w celu zwiększenia stopnia generyczności systemu. Aby umożliwić efektywniejsze rozpoznanie nie tylko emocji osób z danych treningowych, ale również i tych wcześniejsiej nie-przetwarzanych, zaproponowano metodę selektywnego transferu (ang. *Selective Transfer Machine*, STM) [24]. Jej idea polega na iteracyjnym, nienadzorowanym dopasowywaniu granicy decyzyjnej każdego generycznego klasyfikatora (utworzonego na bazie treningowej) do danych aktualnie analizowanego, nowego użytkownika. Odbywa się to głównie poprzez odpowiednie zwiększenie wag próbek treningowych znajdujących się bliżej próbek testowych. Założymy, że mamy zbiór  $X^{tr}$  składający się z  $n$  par treningowych ( $x_i, y_i$ ). Zadaniem jest znalezienie takiego zestawu wektorów nośnych  $\mathbf{v}$  oraz wektora ich wag  $\mathbf{s}$ , że:

$$(\mathbf{v}, \mathbf{s}) = \arg \min_{\mathbf{v}, \mathbf{s}} (R_v(X^{tr}, \mathbf{s}) + \lambda \Omega_s(X^{tr}, X_{te})), \quad (1.2)$$

gdzie  $R_v(X^{tr}, \mathbf{s})$  to funkcja określająca wartość ryzyka zdefiniowanego na zbiorze treningowym  $X^{tr}$ , gdzie każda instancja  $x_i$  jest przemnożona przez odpowiadającą jej wagę  $s_i$ , natomiast  $\Omega_s(X^{tr}, X^{te})$  oznacza stopień rozbieżności pomiędzy rozkładem zbioru treningowego i testowego  $X^{te}$  w zależności od dobieranego wektora  $\mathbf{s}$ . Współczynnik  $\lambda > 0$  służy do balansowania wpływu ryzyka i rozbieżności rozkładów.



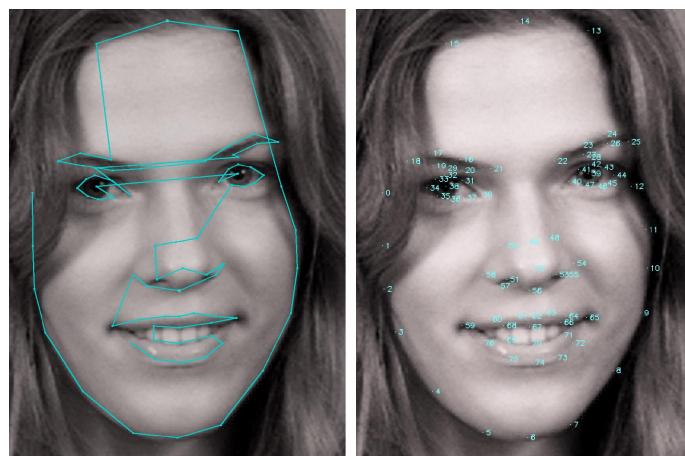
Rysunek 1.7: Idea działania proponowanej metody STM personalizującej generyczny klasyfikator SVM (po lewej) i działanie aplikacji (po prawej). Źródła: [24, 25]

Cała implementacja umożliwia śledzenie 66 punktów twarzy [25]. Skuteczność wykrywania AU wynosi około 91%, co pośrednio przekłada się na skuteczność w rozpoznawaniu emocji, których jest 6 (obecny stan neutralny, brak strachu). Dużą zaletą tego rozwiązania jest satysfakcjonująca niezależność od warunków oświetleniowych, orientacji twarzy (w pewnym stopniu) oraz użytkownika. Autorzy projektu udostępniają kod źródłowy implementacji na różnych licencjach w tym akademickiej.

## STASM

Kolejnym przykładem jest biblioteka służąca do wyznaczania i śledzenia punktów charakterystycznych twarzy poprzez wykorzystanie aktywnego modelu kształtu (ang. *Active Shape Model*, ASM) [26]. Modele te to aktualnie jedno z najbardziej obiecujących podejść w tej dziedzinie. Cechuje je elastyczność i uniwersalność, skutecznie dopasowują się one do zmian kształtów śledzonych obiektów po właściwie przeprowadzonym treningu. Kształtowanie modelu w przypadku twarzy polega na ręcznym nanoszeniu punktów na odpowiednie miejsca przy różnych ekspresjach i orientacjach tejże twarzy. Algorytm jest dzięki temu w stanie dopasowywać model względem pokazanych wcześniej wzorców odniesienia. STASM [27] to jedna z wielu dostępnych implementacji tego procesu, związana z pracą [28]. Ocena stopnia dopasowania modelu jest budowana na dwóch kryteriach :

- kryterium modelu profilowego – związane z oceną otoczenia dla każdego punktu z osobna – każdy punkt posiada pewne charakterystyczne dla siebie sąsiedztwo, inną charakterystykę będą miały okolice krawędzi ust, a inną krawędzi oczu,
- kryterium modelu kształtu – uwzględniające wzajemne położenie punktów twarzy – wszystkie punkty razem powinny tworzyć w przybliżeniu jeden z wyuczonych wcześniej kształtów twarzy, co ogranicza w znacznym stopniu dowolność położenia punktów.



Rysunek 1.8: Przykładowe dopasowanie modelu ASM przy pomocy biblioteki STASM.  
Źródło: [stackoverflow.com](http://stackoverflow.com)

Głównym problemem związanym z tego typu rozwiązaniami jest wydajność obliczeniowa. Biblioteka ta umożliwia śledzenie do 77 punktów, z których każdy jest osobno analizowany. Ponadto konieczne są również obliczenia korelujące. STASM został napisany w języku C/C++ i jest dostępny na licencji *open source*.

## Inne

Poza zaprezentowanymi przykładami istnieje bardzo wiele innych aplikacji dokonujących rozpoznawania emocji w oparciu o ekspresję twarzy. Mogą one przybierać

formę dużych, rozbudowanych systemów, jak np. Emotient [29], FaceReader [30] czy nVisio [31], które poza podstawowymi funkcjonalnościami oferują webowe i mobilne aplikacje, umożliwiają integrację z danymi zbieranymi przy pomocy biosensorów czy dostarczają wielu złożonych analiz. Wyraźnie widoczne jest ukierunkowanie tego typu systemów na wspomaganie działalności marketingowej. Rozwiązań nie muszą być jednak zawsze tak zaawansowane ani pod względem funkcjonalnym, ani w zakresie stosowanych metod przetwarzania. Przykładem niech będzie system do ekstrakcji cech na potrzeby rozpoznawania twarzy [32] czy praca [33]. Metody wykorzystane w tych projektach opierają się głównie na analizie stopnia jasności pikseli, używają prostych projekcji wertykalnych i horyzontalnych histogramów oraz metod gradientowych do określania konturów części twarzy. Pokazują one, że nawet przy pomocy tak prostych technik możliwe jest uzyskanie cech opisujących twarz w pewnym ograniczonym stopniu. Oczywiście prostota rozwiązań, choć zmniejsza złożoność obliczeń i implementacji, skutkuje znaczącym spadkiem odporności na zmienność przypadków. Rozwiązań te mogą znaleźć praktyczne zastosowanie jedynie w ścisłe określonych warunkach. W rzeczywistości projektowanie tego i innego rodzaju systemów to poszukiwanie rozwiązań, które spełnią wymagania użytkownika przy uwzględnieniu występujących ograniczeń.

## 1.5 Cel i zakres pracy

Celem niniejszej pracy inżynierskiej jest **stworzenie systemu umożliwiającego automatyczne rozpoznawanie emocji na podstawie rejestrowanej ekspresji twarzy użytkownika**. Jak pokazano, takie określenie celu mówi wszystko i zarazem nic. Konieczne jest więc doprecyzowanie. Ogólne założenia to:

- analiza w czasie rzeczywistym,
- frontalny obraz twarzy użytkownika,
- stabilne i równomierne oświetlenie,
- rozpoznawanie 6 emocji (FACS) oraz stanu neutralnego.

System powinien realizować trzy zasadnicze bloki przetwarzania w następującym zakresie.

### 1. Pobieranie i przetwarzanie obrazu:

- (a) akwizycja sekwencji obrazów wideo z kamery internetowej, o rozdzielczości co najmniej 640x480, w formacie RGB,
- (b) implementacja specjalizowanych metod przetwarzania obrazu na bazie do której biblioteki w celu umożliwienia efektywnej ekstrakcji cech.

### 2. Detekcja twarzy i ekstrakcja cech:

- (a) detekcja twarzy przy pomocy wybranego detektora (implementacja nie jest tematem projektu),

- (b) implementacja ekstrakcji punktów charakterystycznych twarzy w oparciu o rozdzielną analizę barwną głównych obszarów (oczy, brwi, nos, usta),
- (c) generowanie opisu geometrycznego twarzy.

### 3. Rozpoznawanie emocji:

- (a) klasyfikacja przy pomocy gotowych klasyfikatorów, konfigurowanych z wykorzystaniem własnych danych uczących,
- (b) rozpoznawanie w oparciu o opis geometryczny twarzy.

Niniejsza praca prezentuje przede wszystkim metody rozwiązania problemu, zaimplementowane w ramach systemu.

# Rozdział 2

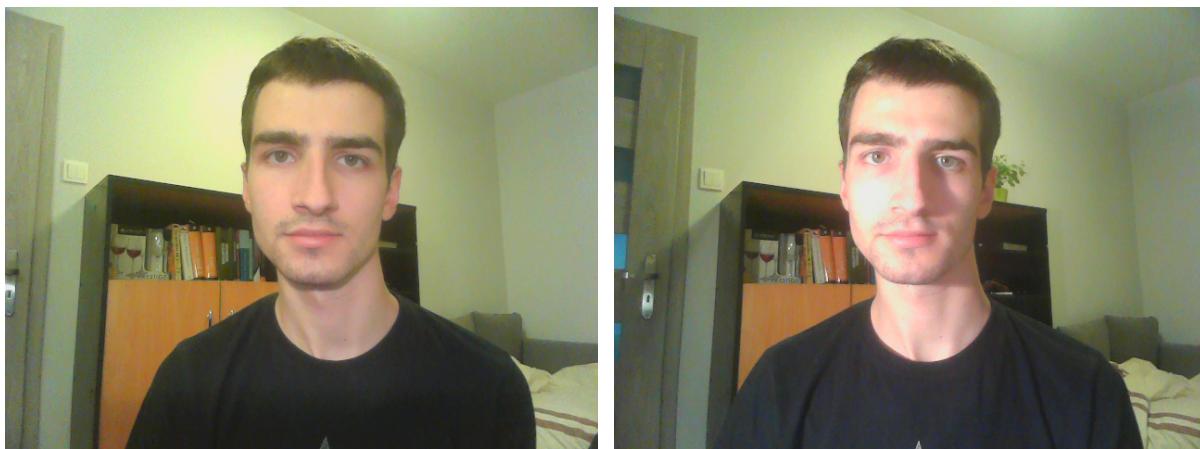
## Rejestracja obrazu twarzy

Rozdział ten opisuje proces akwizycji obrazu wideo zawierającego twarz użytkownika oraz sposób jej detekcji na potrzeby ekstrakcji cech w kolejnym etapie.

### 2.1 Obraz wejściowy

Całość procesów i ich testowania została zrealizowana na laptopie Dell Vostro 2520. Jest on wyposażony m.in. w procesor Intel Core i3 2 x 2.5GHz, 4 GB pamięci RAM oraz zintegrowaną kartę graficzną Intel HD 3000. Obraz wejściowy potrzebny do przetwarzania jest pozyskiwany przy pomocy wbudowanej kamery internetowej o rozdzielcości 1-megapiksla. Dostępny laptop to standardowe urządzenie do użytku domowego, co narzuca ograniczenia w kwestii możliwości i szybkości operacji. Jednocześnie spełnia on podstawowe wymogi konieczne do zrealizowania założonych funkcjonalności w podstawowym zakresie. Wszystkie operacje związane z pozyskiwaniem i dalszym przetwarzaniem obrazów zostały obsłużone przy pomocy biblioteki OpenCV 3.0 [34].

Obraz wejściowy z kamery jest przekazywany w formacie RGB. Dostęp do niego umożliwia klasa `VideoCapture`, przy pomocy której formowana jest macierz obrazu `Mat` kompatybilna z wykorzystywaną biblioteką. Wymiary obrazu wejściowego wynoszą 640x480 i jak się okazuje są one wystarczające do wydajnego przetwarzania. Zgodnie z założeniami projektu obraz ma prezentować osobę, której emocje będą rozpoznawane. Powinna ona siedzieć na przeciwnie do kamery tak, aby dostępny był frontalny obraz jej twarzy. Ponadto powinna być ona oświetlona w sposób możliwie równomierny, bez przepaleń i cieni. Ograniczenia te nie mają dużego znaczenia w przypadku samej detekcji twarzy, której metoda jest wystarczająco odporna, są natomiast kluczowe dla procesu ekstrakcji cech. Poniżej zaprezentowano przykłady wspomnianych obrazów.



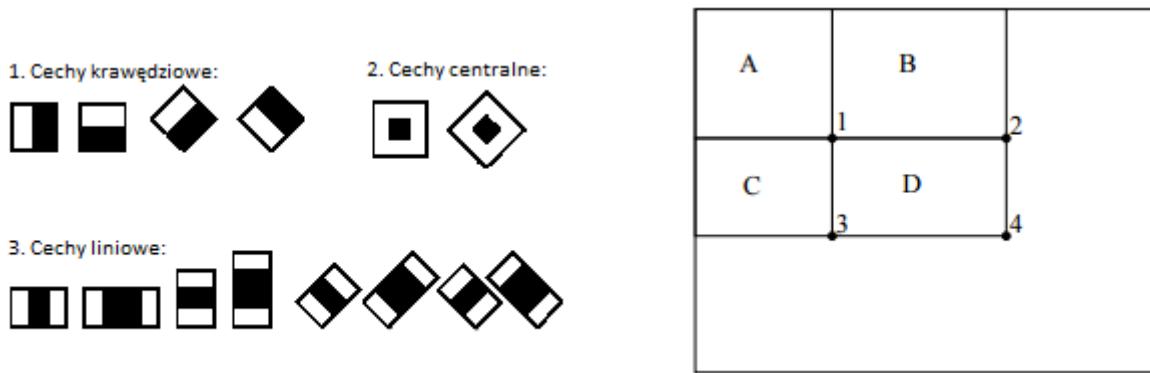
Rysunek 2.1: Przykład poprawnego obrazu (po lewej) – użytkownik siedzi blisko kamery, twarz jest dobrze wyeksponowana i oświetlona. Niepoprawny obraz (po prawej) – twarz jest oświetlona ostrym światłem bocznym, powstały przepalenia powodujące utratę szczegółów oraz cienie mogące wpływać na segmentację części twarzy

## 2.2 Detekcja twarzy

Detekcja twarzy użytkownika została oparta o klasyfikator wykorzystujący tzw. **cechy Haara**. Jego implementacja zawarta jest w bibliotece OpenCV, a jej teoretyczny fundament stanowi praca [35].

### 2.2.1 Kaskadowy klasyfikator cech Haara

Algorytm opiera swoje działanie na metodach uczenia maszyn, a efektywność jego działania zależy od przygotowanych obrazów treningowych – pozytywnych, zawierających szukany obiekt (w tym przypadku twarz) oraz negatywnych. Obrazy te nie są opisywane bezpośrednio poprzez piksele. Wykorzystywane są do tego odpowiednio wyekstrahowane cechy. Określone są one przy pomocy prostokątnych, przesuwanych po obrazie macierzy binarnych, stanowiących podstawę dla tzw. funkcji bazowych. Piksele, które znajdują się pod białym polem są odejmowane od sumowanych pikseli znajdujących się pod polem czarnym. Tym sposobem określone są wartości wektora cech. Zastosowanie wspomnianych funkcji bazowych zostało zaczerpnięte z ogólniej traktującej owo zagadnienie pracy [36]. Takie podejście nie gwarantuje jednak satysfakcyjną wydajność, ponieważ pokrycie obrazu cechami jest wymagające obliczeniowo. Dzieje się tak z dwóch powodów.



Rysunek 2.2: Przykładowe macierze bazowe (po lewej) oraz wizualizacja idei obrazów integralnych (po prawej)

Pierwszym problemem jest każdorazowe obliczanie sum pikseli należących do obszarów prostokątnych. Aby przyspieszyć ten proces, wprowadzone zostały tzw. integralne obrazy (ang. *integral images*), będące szeroko rozpowszechnioną w przetwarzaniu obrazów tablicą sumowanego obszaru (ang. *summed area table*) [37]. Zaproponowany algorytm dla każdego piksela oblicza w pierwszym przebiegu sumę wartości pikseli powyżej i poniżej niego. Tym sposobem uzyskiwanie sumy pikseli w prostokątnych obszarach macierzy bazowych sprowadza się do jedynie trzech operacji dodawania przy dowolnym rozmiarze. Przykładowo suma pikseli w obszarze D (rysunek 2.2) to wartość integralnego obrazu w P4 pomniejszona o P3 i P2 oraz powiększona o P1 (ponieważ dwa razy odejmujemy tę wartość). Rozwiążanie to nie zmniejsza jednak rozmiarów otrzymywanej wektora cech, co stanowi drugi z problemów. Zauważmy, że dla subokna o wymiarach 24x24, ze wszystkich kombinacji prostokątów możliwe jest uzyskanie ponad 160.000 cech (!) [38].

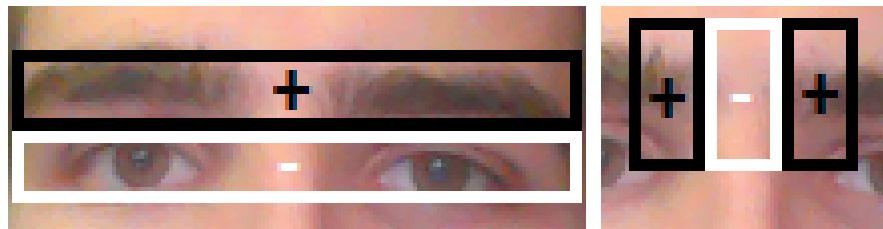
Redukcja wymiaru cech dokonywana jest przy pomocy algorytmu AdaBoost. Jego działanie polega na konstruowaniu jednego silnego klasyfikatora  $\Psi(x)$  z wielu słabysznych  $\psi_j(x)$ , które są wyspecjalizowane do rozpoznawania zbiorów obrazów  $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  pod kątem wyłącznie jednej cechy  $j$ . Każdy z obrazów  $\mathbf{x}_i$  posiada przypisaną do niego wagę  $w_i$ . Algorytm w kolejnych obiegach wybiera klasyfikator  $\psi_t(x)$  (a więc i cechę) z najmniejszym błędem  $e_t$  uzyskanym w trakcie rozpoznawania zadanego zbioru obrazów:

$$e_j = \sum_{i=1}^N w_i |\psi(\mathbf{x}_i) - y_i|, \quad (2.1)$$

gdzie  $y_i$  do docelowa etykieta klasy. Następnie zwiększane są wagi błędnie zaklasyfikowanych obrazów. Cykl ten powtarza się, aż do odpowiedniego zmniejszenia błędu lub uzyskania wybranej liczby cech. Końcowy klasyfikator korzysta z  $T$  klasyfikatorów, który wagi decyzyjne wynoszą:

$$\alpha_t = \ln \frac{e_t}{1 - e_t}, \quad (2.2)$$

a więc, jak łatwo zauważyc, im mniejszy błąd, tym silniejszy jest wpływ danego klasyfikatora. Tym sposobem AdaBoost dokonuje jednocześnie selekcji cech oraz konstrukcji silnego klasyfikatora. Testy autorów pokazały, że już 6000 cech jest wystarczające do uzyskania skuteczności rozpoznawania powyżej 90% [35].



Rysunek 2.3: Dwie pierwsze cechy wybrane przez algorytm AdaBoost w trakcie treningu. Pierwsza podkreśla znaczenie linii brwi, druga jaśniejszego środka nosa wraz z przerwą między brwiami. Operacje wykonywane są na obrazie w skali szarości

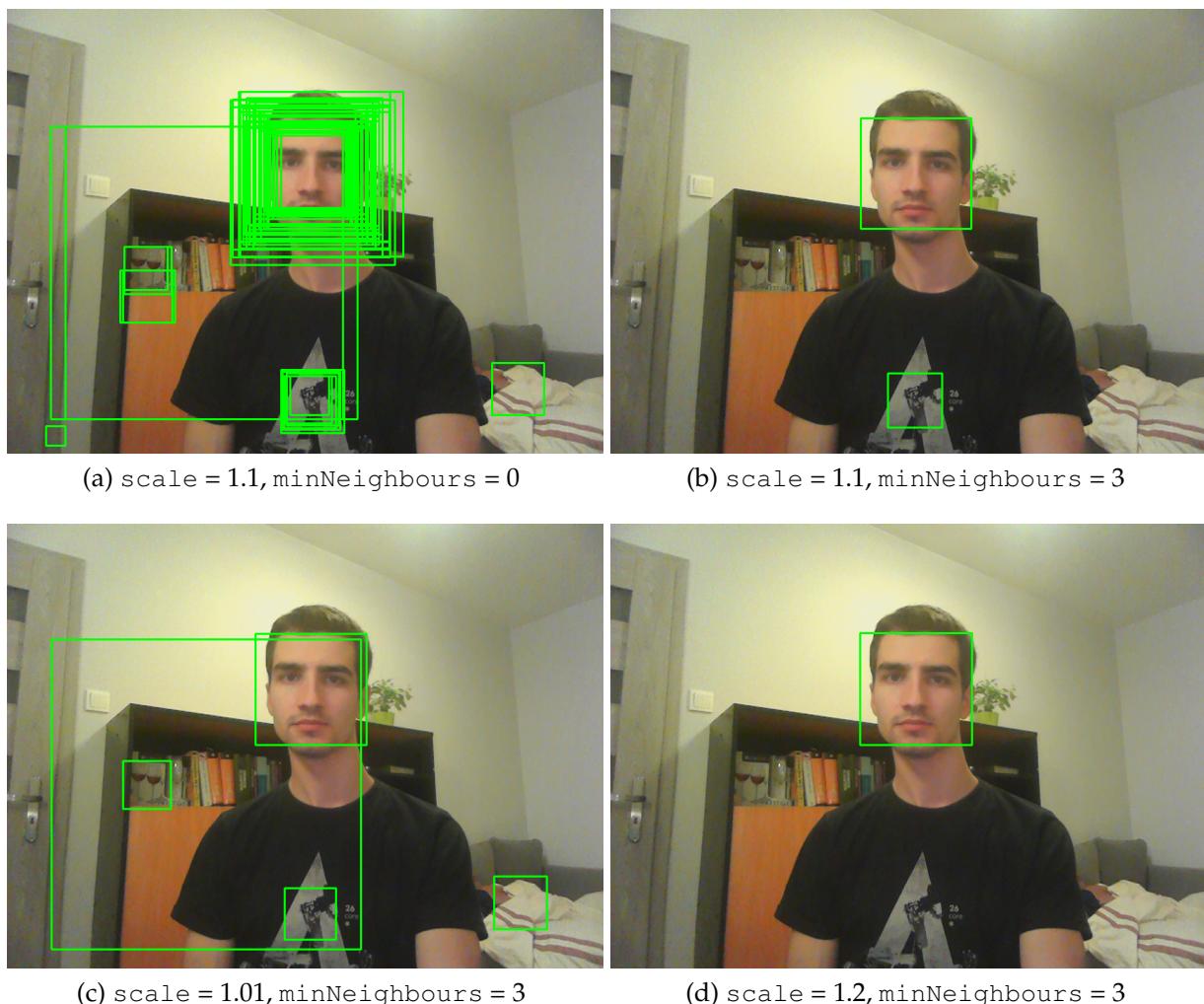
Jeśli się dalej zastanowić, to uzyskany rozmiar wektora – choć niebagatelnie pomniejszony – wciąż jest duży. Ostatnim krokiem optymalizującym przetwarzanie jest zastosowanie kaskady klasyfikatorów. Idea rozwiązania opiera się na spostrzeżeniu, że już mała ilość cech wystarczy do odrzucenia zdecydowanej większości okien niezawierających twarzy. Z tego powodu zamiast analizować wszystkie 6000 cech dla każdego obszaru obrazu wejściowego, klasyfikacja dzielona jest na etapy, w których stopniowo branych jest pod uwagę coraz więcej cech. Etap  $i$  nie następuje, jeśli na etapie  $i - 1$  nie uzyskano pozytywnej odpowiedzi. Autorzy pracy wykazali, że średnio tylko 10 cech jest używanych, co wynika z bardzo dużej liczby szybko odrzucanych okien [35].

Złączenie wszystkich trzech metod sprawia, że opisany detektor jest stosunkowo szybki i efektywny. Ponadto łatwo dostosowuje się do różnych rozmiarów twarzy – wymaga to jedynie skalowania obrazu wejściowego.

### 2.2.2 Wykorzystanie detektora

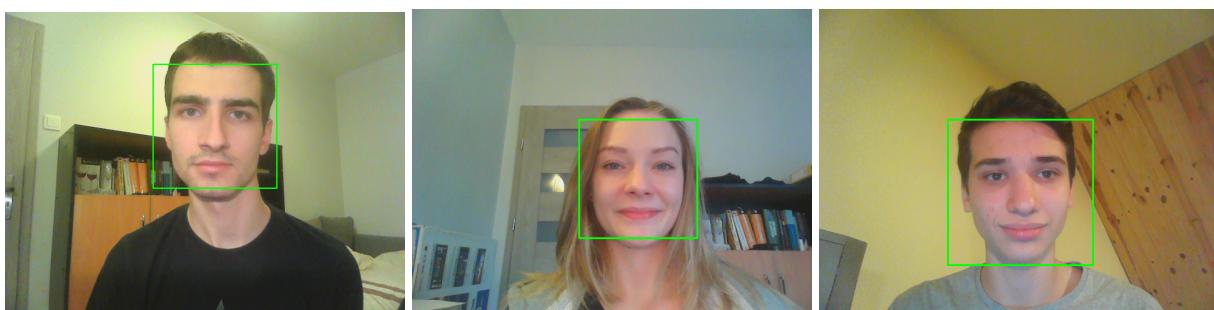
Implementacja opisanego detektora posłużyła do realizacji wykrywania twarzy użytkownika. Jest ona dostępna w bibliotece OpenCV jako klasa `CascadeClassifier`. Aby klasyfikator był w stanie poprawnie rozpoznawać twarze, konieczne jest jego odpowiednie skonstruowanie. W projekcie wykorzystano jedną z wytrenowanych już kaskad, dostępnych w ramach biblioteki. Są one opisane przy pomocy pliku `.xml` i wczytywane przez moduł. Wejściem dla klasyfikatora jest obraz zapisany w skali szarości. Metoda `detectMultiScale` umożliwia wykrywanie twarzy o różnych rozmiarach. Parametr skalowania ustala jak powiększany jest rozmiar obrazu wejściowego, np. wartość 1.15 oznacza, że kolejne analizowane obrazy są zwiększone o 15%. Dodatkowo możliwe jest ustalenie minimalnej liczby sąsiednich detekcji do uznania danego obszaru za szukany obiekt<sup>1</sup>. Na rysunku 2.4 prezentowane są przykładowe wyniki działania detektora dla różnych parametrów.

<sup>1</sup>W dokumentacji biblioteki nie została podana definicja określająca sąsiedztwo.



Rysunek 2.4: Przykłady działania detektora dla różnych parametrów `scale` i `minNeighbours`

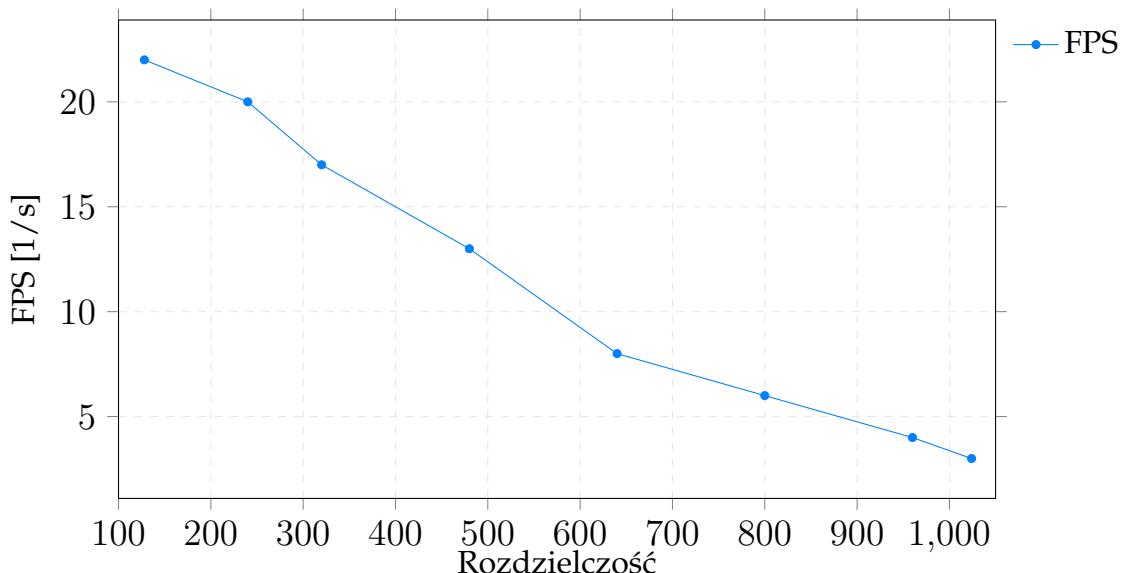
Widoczne jest, że detektor przedżej popełnia błędy drugiego rodzaju niż pierwszego, tzn. częściej wykrywa obiekty nie będące twarzami, choć ona sama również zawiera się w wyznaczonym zbiorze. Można więc stwierdzić, że cechuje go pewna nadwrażliwość. Zwiększenie liczby wymaganych sąsiadów redukuje duże skupiska do jednego reprezentanta, zaś zwiększenie parametru skali ogranicza również i ich liczbę. Dla ostatniej konfiguracji (rysunek 2.4) udało się osiągnąć pożądany efekt. Należy mieć jednak na uwadze, że zbytnie wyostrzenie wymagań może z kolei doprowadzić do ignorowania faktycznych obszarów twarzy. Z tego powodu proponowane jest zachowanie stosunkowo wysokich parametrów, ale z jednoczesnym uwzględnieniem zapasu dla trudniejszych przypadków. Na podstawie przeprowadzonych obserwacji przyjęte zostały więc wartości `scale = 1.1, minNeighbours = 3`. Aby wyeliminować nadmiarowe obiekty, **wybierany jest jedynie ten największy**, domyślnie mający być szukaną twarzą. Jest to rozsądne podejście, zważywszy na fakt, że użytkownik powinien siedzieć dostatecznie blisko kamery. Ryzyko wykrycia większego obiektu jest w takim przypadku zredukowane do minimum.



Rysunek 2.5: Przykłady działania detektora dla ustalonych parametrów i z wyborem największego proponowanego obiektu twarzy

### 2.2.3 Wpływ rozdzielczości na szybkość przetwarzania

Poza analizą skuteczności detektora sprawdzony został wpływ rozdzielczości obrazu wejściowego na szybkość przetwarzania, mierzoną w liczbie klatek na sekundę (ang. *frames per second*, FPS).



Rysunek 2.6: Liczba przetwarzanych klatek na sekundę (FPS) w zależności od rozdzielczości obrazu (podana jest tylko szerokość, dla trzech pierwszych wymiarów zachowany jest stosunek 1:1, dla pozostałych 4:3)

Choć większa rozdzielczość może dostarczać więcej użytecznej informacji o obrazie, to kluczowa pozostaje zdolność urządzenia do sprawnego przetwarzania. Algorytm detekcji powoduje na tyle istotne opóźnienia, że wprowadzanie obrazu powyżej wymiarów 640x480 staje się zbyt wymagające dla możliwości wykorzystywanego komputera. Z drugiej strony należy zachować odpowiednią liczbę szczegółów dla ekstraktora cech.

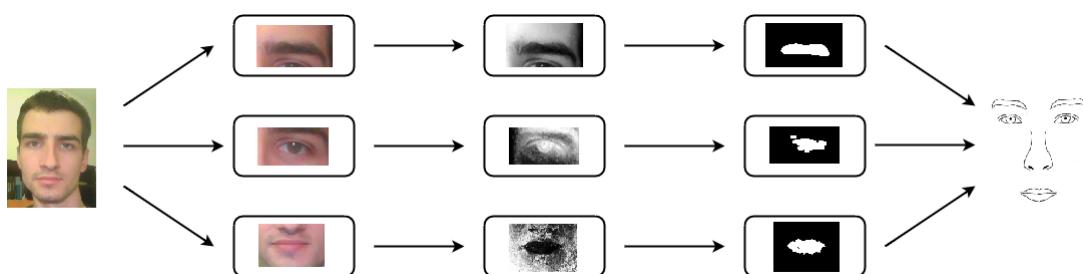
# Rozdział 3

## Ekstrakcja punktów charakterystycznych twarzy

Prezentowany rozdział skupia się na fundamencie proponowanego systemu – ekstrakcji cech twarzy z obrazu wejściowego. Zawarty w nim został szczegółowy opis poszczególnych etapów uzyskiwania geometrycznej reprezentacji kolejnych części twarzy: oczu, brwi, ust oraz zębów i nosa. Zgromadzony wektor ma posłużyć do konstrukcji i wykorzystania klasyfikatorów w kolejnym etapie.

### 3.1 Ogólny schemat przetwarzania

Obraz wejściowy stanowi obszar twarzy wykryty przez detektor opisany w poprzednim rozdziale. Aby ograniczyć liczbę koniecznych obliczeń oraz ryzyko wykrycia nieprawidłowych punktów twarzy, analiza każdej z rozważanych części odbywa się w ramach niezależnego okna ROI. Każde z nich poddawane jest specyficznym dla przetwarzanego obszaru operacjom.



Rysunek 3.1: Ogólny schemat ekstrakcji cech z obrazu twarzy

W każdym przypadku wykonywana jest segmentacja ograniczonego obrazu wejściowego, mająca na celu utworzenie jednolitych, binarnych obszarów reprezentujących potencjalne części twarzy. W ramach segmentacji wykonywana jest operacja **mapowania** pikseli obrazów wejściowych  $I$  na piksele przestrzeni obrazu wyjściowego  $I_{map}$ . Sposób mapowania jest określony na podstawie ustalonej funkcji transformującej  $f$ , co opisujemy prostą formułą:

$$f : I \rightarrow I_{map}. \quad (3.1)$$

Uzyskany obraz jest następnie **binaryzowany** przy pomocy możliwie efektywnej metody. Obraz wyjściowy w większości przypadków zawiera w sobie nadmiarową liczbę wysegmentowanych obiektów  $\mathbf{x} \in I_{map}$  nie będących w istocie tym szukanym (np. cienie na twarzy). Wyszukiwanie właściwego obiektu opiera się na prostych heurystykach  $h(\mathbf{x})$  nadających każdemu z nich określoną wartość oceny (zazwyczaj jest to pole powierzchni). W obiekcie tym można następnie wyznaczyć odpowiednie punkty, które posłużą do opisu geometrycznego danego elementu twarzy, np. szerokości ust. Wszystkie punkty razem umożliwiają pełny opis ekspresji.

Proponowane podejście posiada bogatą bazę prac badawczych i różnego rodzaju eksperymentów. Zwiążane jest to z faktem, że praktycznie każdy element twarzy może stanowić podstawę do odrębnego badania w zakresie jego detekcji. Przykładem niech będzie praca doktorska [39], której poruszonym problemem jest wyłącznie segmentacja obszaru ust na potrzeby rozpoznawania ich charakterystycznych stanów. Autor projektu prześledził działanie wielu z opisanych metod. Część z nich, dającą najlepsze perspektywy, posłużyła jako fundament lub stała się inspiracją do wykrywania szukanych części twarzy.

Prezentowane metody przetwarzania (w tym opisana wcześniej detekcja twarzy) zostały zaimplementowane i zorganizowane w ramach klasy `FacialFeatures` wspomaganej klasami `ImageProcessor` (przetwarzanie obrazu) oraz `ImageAnalyzer` (analiza obiektów) autorskiej biblioteki.

## 3.2 Wyznaczanie zgrubnych obszarów

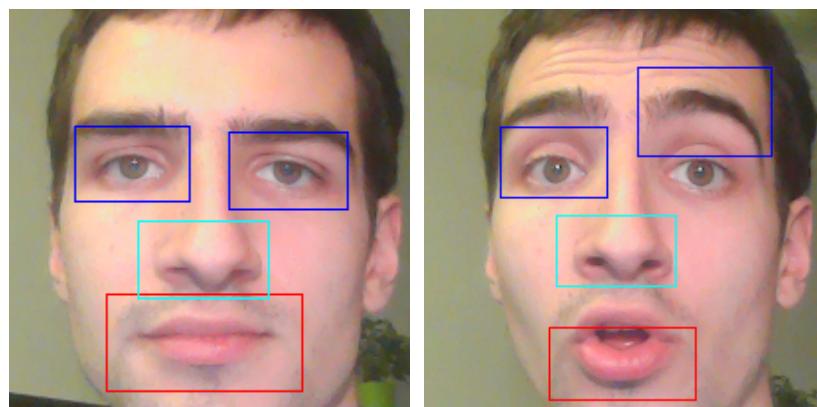
W pierwszym kroku wykryty obraz twarzy jest dzielony na mniejsze, mające zawierać poszczególne jej części, których punkty chcemy określić. Są to obszary oczu, brwi, ust oraz nosa. Jak wspomniano w poprzednim rozdziale, ma to na celu znaczne obniżenie liczby obliczeń w miejscach, w których jest to z dużym prawdopodobieństwem całkowicie zbędne. Zwiększa się tym samym szansa na znalezienie prawidłowych obiektów.

### 3.2.1 Kaskada detektorów

Jednym z podejść do wyznaczania zgrubnych obszarów części twarzy jest zastosowanie opisanego wcześniej detektora cech Haara. Jego uniwersalność umożliwia bowiem wykrywanie dowolnych obiektów w zależności od skonstruowanych kaskad. Móglby on więc dostarczyć wymaganych obszarów do bardziej szczegółowej analizy. Występuje jednak w jego przypadku szereg istotnych problemów:

- dostępne kaskady przeznaczone do wykrywania poszczególnych części twarzy są dostosowane głównie do ich kształtów przy neutralnej ekspresji, mogą wystąpić problemy z utrzymaniem poprawnej detekcji dla np. szeroko otwartych ust czy oczu,

- wykrywanie jedynie wstępnych obszarów przy pomocy tak złożonej formuły obliczeniowej jest niewydajne,
- nadwrażliwość detektorów sprawia, że wykrywają one wiele nieprawidłowych obiektów na rozbudowanej pod względem kształtów i odcieni twarzy, co powoduje, że proste kryterium rozmiaru może być niewystarczające (jego rozbudowywanie o zależności od lokalizacji obiektów sprowadza się praktycznie do metody opisanej w następnej punkcie).

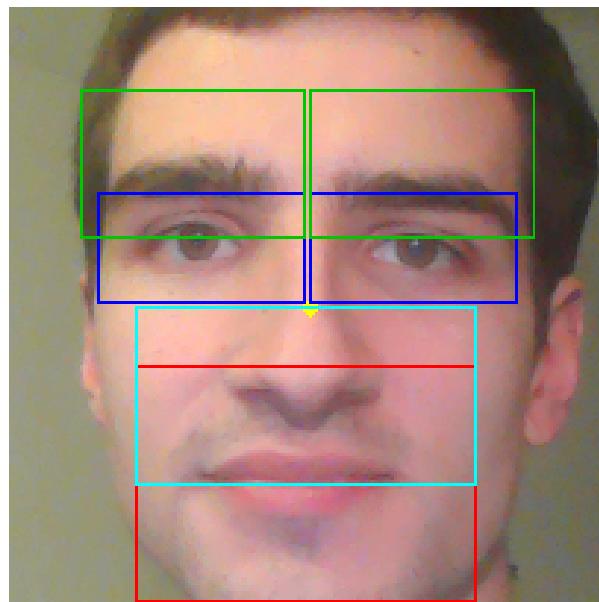


Rysunek 3.2: Przykładowe obszary wyznaczone przy pomocy detektorów. Po lewej – prawidłowa detekcja (stosunkowo rzadka bez dodatkowych warunków), po prawej – ekspresja zaskoczenia i niepoprawne oznaczenie obszaru prawego oka oraz niepełne objęcie otwartych ust

Oczywiście wpływ pierwszego z wymienionych problemów oraz w pewnym stopniu trzeciego, można by zmniejszyć, trenując własne, dostosowane do problemu klasyfikatory. Ciężko jednak byłoby uzyskać lepsze rozwiązania niż te uzyskane na bazie dużych i zróżnicowanych zbiorów. Mając to na uwadze oraz pamiętając o kwestii wydajności, zdecydowano się na inne podejście – znaczco szybsze choć mniej uniwersalne i odporne na zróżnicowanie twarzy użytkowników.

### 3.2.2 Statyczne obszary

Projekt zakłada, że użytkownik korzystający z systemu powinien siedzieć naprzeciw komputera, a kamera ma rejestrować frontalny obraz jego twarzy. Twarze użytkownika mogą mieć różne rozmiary, jednak pomiędzy obszarami zawierającymi jej główne części zachowywane są pewne ograniczone proporcje. Jest to zagadnienie szczególnie dobrze znane rysownikom. Przykładowo linia oczu wyznacza mniej więcej połowę wysokości twarzy, a górną granicę obszaru z ustami wyznacza jedna trzecia tejże wysokości. Bardziej szczegółowe informacje można znaleźć w literaturze naukowej traktującej o antropometrycznych proporcjach twarzy [40]. W projekcie oparto się na rozmiarach zaproponowanych w pracy [41], w której określone były obszary oczu i brwi. Zostały one ustalone empirycznie tak, by pokryły poprawnie 144 przypadki testowe. Na ich podstawie ustalonono pozostałe wymiary, mając na uwadze przytoczone informacje i praktyczną analizę.



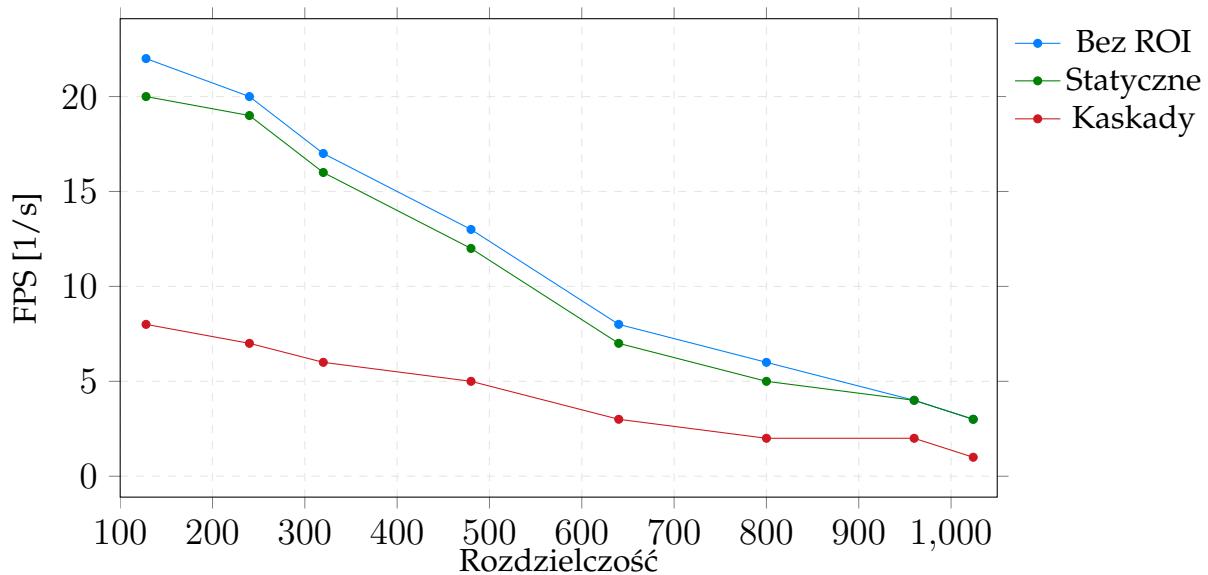
Rysunek 3.3: Ustalone obszary ROI na wykrytej twarzy

W prezentowanym podziale przyjmuje się następujące oznaczenia:  $C = (c_y, c_x)$  – punkt środkowy obrazu twarzy,  $f_w$  – szerokość obrazu,  $f_h$  – wysokość obrazu. Obszary są tworzone na podstawie lewego górnego rogu prostokąta  $R_i = (r_x, r_y)$ , szerokości  $r_w$  oraz wysokości  $r_h$ . Wartości parametrów są ustalane zgodnie z poniższą tabelą. Układ współrzędnych implementacji jest **odwrócony**, tj. punkt zerowy znajduje się w lewym górnym rogu.

Tabela. 3.1: Lewe górne narożniki obszarów ROI oraz ich wysokości i szerokości

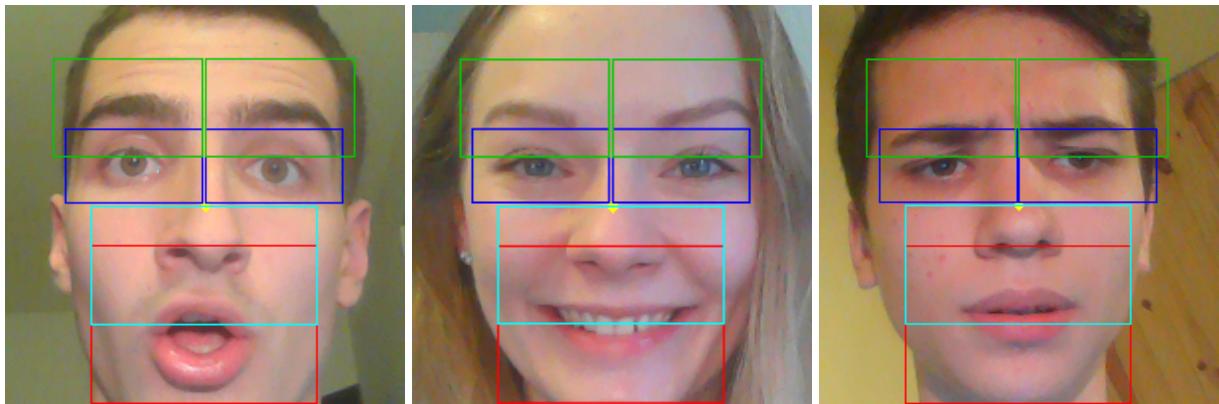
ROI	$R_i$	$r_x$	$r_y$	$r_w$	$r_h$
Lewa brew	$R_1$	$c_x - 0.38f_w$	$c_y - 0.33f_h$	$0.38f_w$	$0.22f_h$
Prawa brew	$R_2$	$c_x$	$c_y - 0.33f_h$	$0.38f_w$	$0.22f_h$
Lewe oko	$R_3$	$c_x - 0.35f_w$	$c_y - 0.19f_h$	$0.35f_w$	$0.19f_h$
Prawe oko	$R_4$	$c_x$	$c_y - 0.19f_h$	$0.35f_w$	$0.19f_h$
Nos	$R_5$	$c_x - 0.29f_w$	$c_y$	$0.58f_w$	$0.3f_h$
Usta	$R_6$	$c_x - 0.29f_w$	$c_y + 0.1f_h$	$0.58f_w$	$0.4f_h$

W porównaniu do pierwotnych wartości zawężono nieznacznie obszary oczu oraz podniesiono ich wysokość, tak aby kolejno wyłączyć z nich boczne linie brwi i pozostawić wolną przestrzeń dla szerzej otwieranych oczu. Podobnie postąpiono w przypadku brwi – odcięcie włosów w bocznych obszarach i uwzględnienie wysokości uniesionych brwi. Wysokość punktu referencyjnego obu brwi stanowi wysokość punktu centralnego zwiększoną o 60% wysokości obszarów oczu. Początek obszaru ust został tak ustalony, by w przybliżeniu znajdował się na jednej trzeciej wysokości twarzy, z pewnym zapasem dla otwarcia ust. Szerokość obejmuje trzy czwarte obszaru brwi po obu stronach. Początek obszaru nosa pokrywa się ze środkiem twarzy, zaś szerokość jest taka sama jak tak ust.



Rysunek 3.4: Liczba przetwarzanych klatek na sekundę (FPS) w zależności od rozdzielcości obrazu i zastosowanej metody ustalania ROI (podana szerokość)

Przyjęte podejście jest nieporównywalnie szybsze oraz stabilniejsze i mimo, że łatwo sobie wyobrazić przypadki, które przekreślałyby jego skuteczność (jak choćby większe rotacje głowy), to przy zaprezentowanych założeniach jest ono wystarczające. Warto jednocześnie zauważyć, że również i metoda oparta o detektor nie byłaby dośćatecznie odporna bez wytworzenia odpowiednich kaskad. Wybrane proporcje wymiarów dobrze sprawdzają się dla różnych użytkowników przy różnych ekspresjach.



Rysunek 3.5: Przykłady nałożenia statycznych obszarów dla różnych osób i przypadków

### 3.3 Ekstrakcja punktów

Ekstrakcja punktów charakterystycznych twarzy to kluczowa część całego systemu. Celem tego procesu jest detekcja takich punktów, które umożliwiają rozróżnianie stanów poszczególnych części twarzy, np. otwarte oko, rozszerzone usta. Efektywność dobranych metod w fundamentalny sposób wpływa na skuteczność systemu w rozpoznawaniu emocji. Prawidłowo wykryte punkty dostarczają użytecznej informacji dla klasyfikatorów. Niepoprawna detekcja nie tylko takich danych nie niesie, ale wprowadza ponadto dodatkowe zaszumienie w procesie uczenia. Zastosowane metody nie są ani szczególnie ukierunkowane na precyzyjne wyznaczanie szukanych punktów, ani też nie można nazwać ich metodami zgrubnymi. Stworzone rozwiązania udostępniają na tyle szczegółową informację, by rozróżnianie ekspresji było możliwe oraz jednocześnie na tyle zgeneralizowaną, by uniknąć analizy zbyt skomplikowanych obrazów i tym samym ograniczyć liczbę koniecznych obliczeń. Przy konstruowaniu opisywanych algorytmów odnoszono się do wielu innych opublikowanych rozwiązań, starając się wybrać podejście optymalne.

#### 3.3.1 Oczy

Oczy stanowią zdecydowanie najbardziej zróżnicowany obszar twarzy. Występują na nimi różne kolory pochodzące od białej twardówki, barwnej tęczówki i czarnej żrenicy. Dodatkowo ich otoczenie stanowi nie tylko skóra, ale również i ciemne rzęsy. Oczy znajdują się też stosunkowo blisko wyrazistych brwi, łatwo o powstawanie cieni w oczodołach. Te wszystkie czynniki sprawiają, że dokładne wykrycie konturu oczu nie jest zadaniem trywialnym. Celem modułu jest detekcja dwóch par punktów horyzontalnych i wertykalnych oraz punktu centralnego dla każdego oka, jednoznacznie opisujących ich elipsy. Badanie położenia owych punktów pozwoli na opisywanie stopnia otwarcia i zamknięcia oczu oraz ich położenia względem brwi.

Do rozwiązania problemu można podejść na wiele sposobów. Istnieją metody skupiające się na segmentacji wybranych części oka. Przykładem może być wykrywanie twardówki oka [42]. Biała twardówka w istotny sposób odróżnia się od obszarów sąsiednich. Autorzy cytowanego rozwiązania tworzą statystyczny model jej barwy na bazie dostarczanych przykładów treningowych. Detekcja twardówki może być jednak niewystarczająca ze względu na fakt jej przysłonięcia powiekami (wertykalnie) oraz występowanie bardzo ograniczonego obszaru wewnętrznej części oka. Określenie pionowej pary można by stosunkowo łatwo wykonać, segmentując intensywnie czarną żrenicę oka. Utracilibyśmy jednak wtedy informację o punktach horyzontalnych. Podejście hybrydowe, wymagałoby zaś wykonywania dwóch niezależnych algorytmów, co z kolei ma swoje wady wydajnościowe.

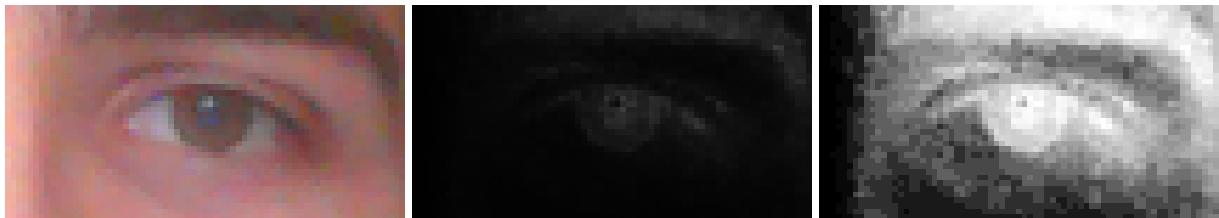
Zamiast próbować segmentować mocno zróżnicowane oko, można podjąć się od-filtrowania wszystkiego (z najbliższego otoczenia) co nim nie jest – głównie skóry. Ta posiada zdecydowanie bardziej jednolity charakter i skłania ku próbom tworzenia reguł barwnych ją wydzielających. Ponieważ operujemy na obrazie RGB, mamy dostęp do trzech podstawowych kanałów: czerwonego, zielonego i niebieskiego. Na ich podstawie można również uzyskać składowe innych przestrzeni, np. HSV (ang. *hue, saturation, value*) czy YCbCr (ang. *luminance, chroma – blue difference, chroma – red difference*). Istnieje wiele metod modelujących kolor skóry w różnych przestrzeniach [43]. Po

licznych eksperymentach autor pracy zdecydował się na bezpośrednie wykorzystanie przestrzeni RGB. Najbardziej obiecujące okazały się bowiem podejście podkreślające dominację kanału czerwonego w rozkładzie koloru skóry. Ciemna żrenica, kontury oka i rzęsy to obszary zawierające mało rozważanej składowej. Wszystkie razem mogą wystarczyć do utworzenia elipsy oka, nawet z pominięciem zawierającej dużo czerwonego twardówki. Kluczowe okazało się wykorzystanie odpowiedniej funkcji mapującej, eksponującej potrzebne fragmenty badanego obszaru. Wykorzystano do tego formułę zaproponowaną w pracy [41] mapującą piksele  $p_r$  kanału czerwonego obrazu wejściowego  $I$  na piksele  $p_{map}$  mapy oczu  $I_{map}$  zapisanej w skali szarości.

$$f_{map}(p_r) = \exp((255 - p_r) \frac{\ln(255)}{255}). \quad (3.2)$$

Formuła ta pozwala na nadanie obszarom nienależącym do skóry (i twardówki) większych wartości przy jednoczesnym wzmacnieniu różnicy z wykorzystaniem operatora eksponentjalnego. Przemnażanie wyniku przez drugi człon pozwala na znormalizowanie zakresu wartości do przedziału [0,255].

Metoda ta okazuje się być najlepszą z praktycznego punktu widzenia. Dobrze podkresla pożądaną obszary i jednocześnie nie generuje dużo zbędnego zaszumienia w częściach sąsiadujących (co miało miejsce np. przy rozważaniu jasności pikseli od razu w skali szarości).



Rysunek 3.6: Od lewej: obraz wejściowy, obraz po wykonaniu operacji mapowania, obraz mapy ze zwiększonym kontrastem dla lepszej wizualizacji

Kolejnym krokiem jest binaryzacja otrzymanej mapy. Ponieważ poprzedni etap daje stosunkowo dobre rozróżnienie, ten jest już ułatwiony. Określanie wartości wyjściowych odbywa się zgodnie z następującą formułą [41]:

$$f_{bin}(p_{map}) = \begin{cases} 1 & \text{dla } p_{map} > (\bar{I}_{map} + 0.9\sigma) \\ 0 & \text{w przeciwnym wypadku,} \end{cases} \quad (3.3)$$

gdzie  $\bar{I}_{map}$  oznacza średnią wartość pikseli w mapie, a  $\sigma$  to ich odchylenie standarde. Formuła ta wybiera najjaśniejsze piksele należące do krańcowego obszaru rozkładu jasności pikseli obrazu. Współczynnik stojący przy wariancji pozwala na kalibrację liczby wybieranych pikseli. Jego wartość została dobrana empirycznie. Następnym krokiem koniecznym do przeprowadzenia jest usunięcie obszarów przylegających do krawędzi okna, mogących reprezentować np. brwi. Wykorzystywany jest do tego algorytm *flood fill* [44], który w tym przypadku wypełnia kolorem czarnym ciągły, biały obszar przylegający do obramowania. Ponieważ na tym etapie metoda nie zwraża jednolitych segmentów, konieczne jest ich połączenie przy pomocy operacji dylacji

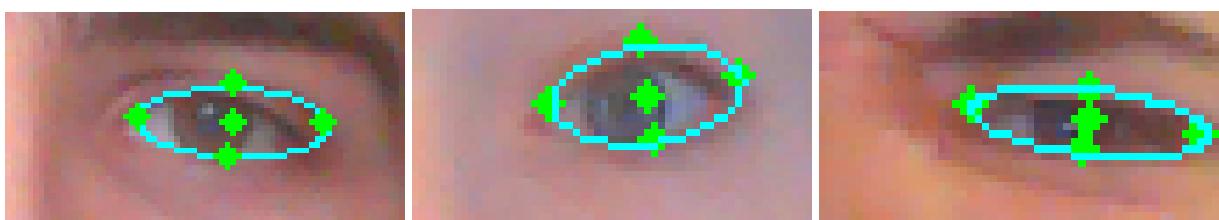
[45]. Jest to metoda morfologiczna polegająca na rozbudowywaniu obszaru w oparciu o wybrany element strukturyzujący. W rozważanym przypadku elementem tym jest macierz prostokątna o rozmiarach  $5 \times 3$ . Jest ona przesuwana po analizowanym obszarze. Dla każdego położenia okna piksel centralny jest zastępowany wartością maksymalną w danym oknie. W przypadku obrazu binarnego filtr ten działa więc jak filtr maksymalizujący. Metoda ta pozwala na wypełnienie dziur pomiędzy szukanymi fragmentami.



Rysunek 3.7: Od lewej: obraz po binaryzacji, obraz po wykonaniu operacji czyszczenia krawędzi, obraz mapy po operacji dylacji

Warto zaznaczyć, że segment oka może wydawać się niepotrzebnie nadbudowany. Zastosowanie erozji poprzedzającej dylację (operacja odwrotna) doprowadziłoby jednak do zbytniej degeneracji obszaru, a w konsekwencji do braku połączenia między fragmentami, np. przy szeroko otwartych oczach. Podobnie z praktycznego punktu widzenia lepsze okazało się zastosowanie prostokątnego, a nie eliptycznego elementu strukturyzującego.

Na tym etapie możliwe jest już efektywne wykrywanie poszukiwanego segmentu oka. Z obrazu binarnego wybierany jest obszar największy. Aby uodpornić algorytm na ewentualne przedostanie się obszaru brwi, wybrany segment powinien mieć szerokość mniejszą niż 65% analizowanego okna. Poszukiwane punkty nie są ustalane bezpośrednio na bazie kształtu (konturu) wybranego obszaru, ponieważ wprowadzałoby to duże wahania w ich położeniu (już jeden piksel robi różnicę). Z tego powodu określenie owych punktów poprzedzone jest dopasowaniem elipsy w oparciu o metodę najmniejszych kwadratów [46]. Dopiero na jej podstawie wybierane są pary skrajnych punktów horyzontalnych i wertykalnych oraz punkt centralny. Takie podejście generalizuje kształt oka w satysfakcyjującym stopniu. Jego wysokość i szerokość są dobrze zachowane, dokładne położenie kącików oczu nie istotne.



Rysunek 3.8: Przykłady detekcji punktów oka dla różnych stanów i osób

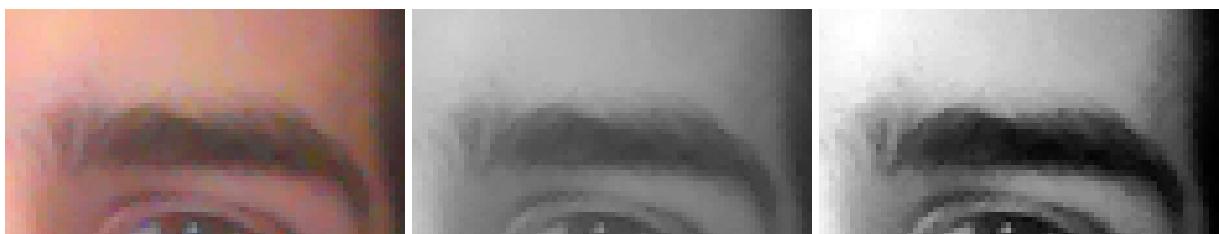
### 3.3.2 Brwi

Kolejnym analizowanym obszarem jest okno brwi. Ich segmentacja stanowi o wiele prostsze zadanie niż miało to miejsce w przypadku oczu. Najbardziej rozpowszechnione są metody bazujące na jasności pikseli. Część z nich wykorzystuje histogramy horyzontalne do określenia skrajnych punktów brwi, stanowiących głębokie minima [47]. Obszar brwi jest zazwyczaj zdecydowanie ciemniejszy niż jego otoczenie w postaci skóry. Jego oddzielenie wymaga najczęściej doboru odpowiedniego poziomu progowania. Znane są metody korzystające z większej liczby wartości granicznych, np. dla pikseli poniżej i powyżej potencjalnego obszaru brwi [48].

Autor pracy zdecydował się na prostą metodę polegającą na wyborze 15% najciemniejszych pikseli z analizowanego okna. Sama binaryzacja jest jednak poprzedzona trzema uzupełniającymi operacjami. Pierwszą jest wyrównywanie histogramu [49], w której do pikselom  $p_{in}$  obrazu wejściowego  $I$  w skali szarości są przypisywane nowe wartości  $p_{map}$  zgodnie z ich dystrybuantą:

$$f_{map}(p_{in}) = H(p_{in}) = \sum_{i=0}^{p_{in}} h(i), \quad (3.4)$$

gdzie  $H(i)$  jest dystrybuanta, a  $h(i)$  funkcją histogramu. Wartości wyjściowe są normalizowane do zakresu  $[0,255]$ . Operacja ta pozwala na zwiększenie kontrastu między ciemnymi brwiami i jaśniejszą skórą.



Rysunek 3.9: Kolejne etapy wstępnej przetwarzania brwi, od lewej: obraz wejściowy, obraz w skali szarości, obraz po wzmacnieniu kontrastu

Problemem ponownie mogą okazać się fragmenty innych obiektów znajdujących się na obszarze sąsiadującym z krawędziami okna, np. włosy, części oczu. Zastosowanie metody takiej jak we wcześniejszym przypadku nie jest jednak wskazane. Jest tak ze względu na fakt, że brwi mogą do tych krawędzi przylegać, np. gdy są wysoko uniesione lub gdy są po prostu szerokie. Obszar krawędziowy nie jest więc całkowicie czyszczony. Zamiast tego nakładana jest na niego funkcja kary  $\nu$ , rozjaśniająca piksele proporcjonalnie do ich względnej – mianownikiem jest szerokość lub wysokość analizowanego okna – odległości  $d$  od obramowania okna:

$$\nu(d) = -\frac{m}{w}d + m, \quad (3.5)$$

gdzie  $m$  oznacza maksymalne procentowe wzmacnianie w przedziale  $[0-1]$ , natomiast  $w$  to względna granica rozjaśniania, również w takim przedziale. Nowe wartości dane są wzorem:

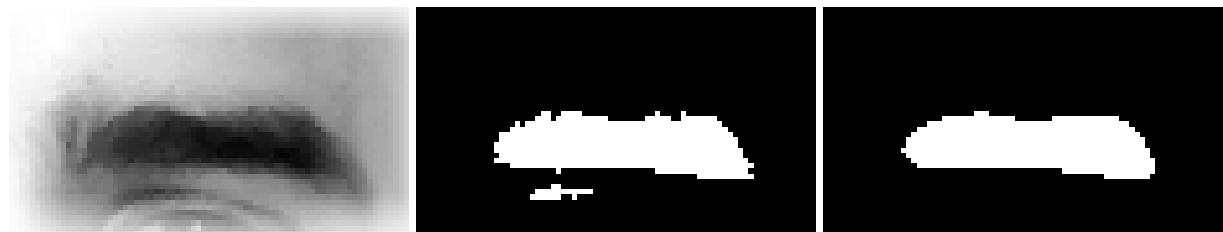
$$p'_{map} = p_{map} + \nu(d)(255 - p_{map}). \quad (3.6)$$

Zastosowanie takiej metody pozwala na zniwelowanie wpływu włosów i innych zbędnych fragmentów na segmentację, przy jednoczesnym zachowaniu w pewnym stopniu obszarów brwi znajdujących się blisko krawędzi. Funkcję tą można wykorzystać również dla dowolnej granicy. Wartości parametrów  $m$  oraz  $w$  powinny być odpowiednio kalibrowane. Na drodze eksperymentów dla autora projektu ustalone zostały następujące wartości.

Tabela 3.2: Wartości parametrów  $m$  i  $w$  dla poszczególnych krawędzi okna brwi.

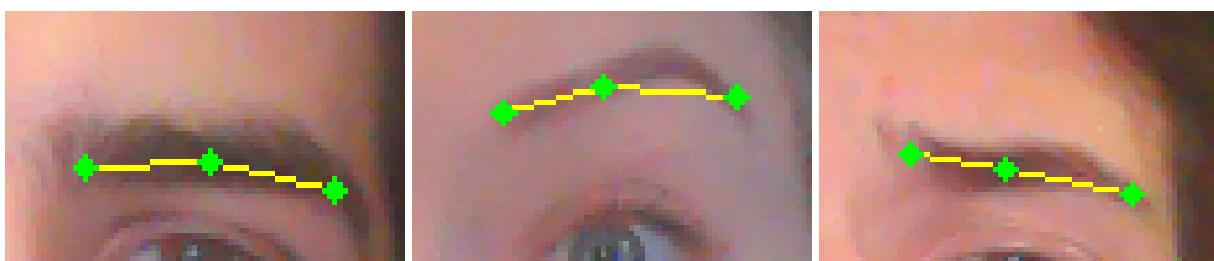
Krawędź	$m$	$w$
Zewnętrzna boczna	0.8	0.25
Wewnętrzna boczna	0.8	0.3
Górna	0.75	0.15
Dolna	0.9	0.25

Dodatkowo wykorzystywana jest informacja o położeniu wykrytego oka – wszystkie fragmenty poniżej jego górnego punktu są usuwane. Dla takiego obrazu możliwe jest już wykonanie wspomnianej binaryzacji (15% najciemniejszych pikseli) oraz oczyszczenie obrazu binarnego z szumu przy pomocy filtracji medianowej. Filtr ten za piksel centralny podstawa medianę dla analizowanego okna [50].



Rysunek 3.10: Od lewej: przykład działania funkcji liniowo czyszczącej krawędzie okna brwi, obraz po binaryzacji, obraz oczyszczony przy pomocy filtru medianowego

Jako obiekt reprezentujący brew wybierany jest największy segment. Określone są trzy punkty: dwa skrajne horyzontalne oraz punkt centralny konturu. Tak skonstruowana metoda jest szybka i prosta w implementacji. Dobrze sprawdza się dla osób posiadających na tyle ciemne brwi, by skutecznie odróżniać je od występujących cieni. W przypadku o jaśniejszych brwiach detekcja punktów nie jest precyzyjna.



Rysunek 3.11: Przykłady detekcji punktów brwi dla różnych stanów i osób

### 3.3.3 Usta

Analiza obszaru ust to kolejny etap ekstrakcji punktów twarzy. Ich segmentacja również – jak w przypadku oczu – charakteryzuje się pewną zwiększoną złożonością. Bynajmniej nie jest problematyczne wydzielenie ust w naturalnej, zamkniętej pozycji. Szczególnego podejścia wymaga uwzględnienie zróżnicowania możliwych ich stanów, jak np. mocno otwarте (widoczne zęby, ciemne wnętrze, język, duża rozpiętość wertykalna), z wyszczerzonymi zębami (dominacja białych zębów, duża rozpiętość horizontałna). Sytuacje te sprawiają, że nie każda metoda skupiająca się na podstawowej ciesze ust – jaką jest ich mocno czerwona barwa – są wystarczająco skuteczne. Podobnie jak w przypadku oczu poszukiwane są dwie pary punktów.

Próby segmentacji ust na bazie ich charakterystyki barwnej są jednak jak najbardziej uzasadnione. Podstawowy problem stanowi odróżnienie ich od skóry. Jest to sytuacja podobna w pewnym stopniu do tej, która miała miejsce w przypadku oczu. Analiza barw obszarów może odbywać się w różnych modelach, np. przy wykorzystaniu składowej odcienia koloru w modelu HSV [51], który jest odporny na luminację, ale mało wiarygodny przy niskiej saturacji (nasyceniu). Efektywniejsze z punktu widzenia autora okazało się podejście bazujące właśnie na saturacji otrzymywanej ze składowych RGB, przy założeniu, że obszar ust zawiera najwięcej składowej czerwonej spośród wszystkich trzech składowych. Podejście to zostało zaprezentowane w pracy [52]. Wspomniany warunek oraz obserwacje rozkładu kolorów warg, języka i skóry prowadzą do redukcji formuły na wartość saturacji  $S$ :

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \Rightarrow \frac{R - G}{R}, \quad (3.7)$$

ponieważ obszar ust i skóry zawierają najwięcej składowej czerwonej, a najlepsze rozróżnienie samych ust możliwe jest przy porównywaniu składowej czerwonej i zielonej [52]. Ponadto przeprowadzana jest następująca normalizacja do zakresu [0-1]:

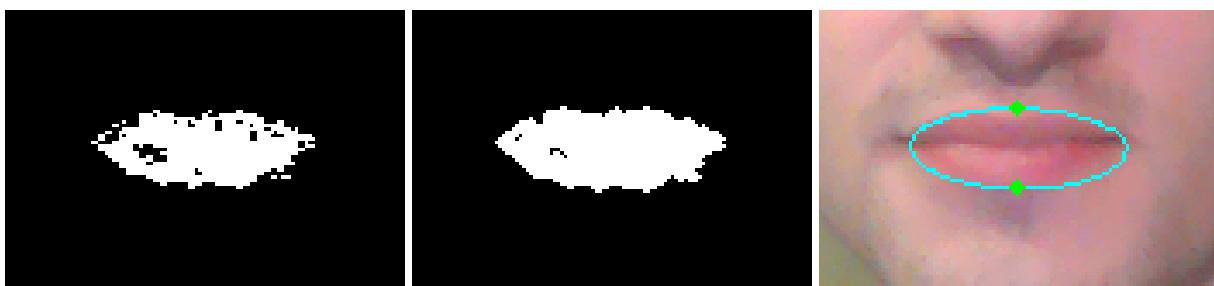
$$S = 2 \arctan\left(\frac{R - G}{R}\right)/\pi, \quad (3.8)$$

Aby uzyskać wyraźniejszy efekt, zwiększany jest kontrast metodą wyrównywania histogramu. Dodatkowo czyszczona jest górną i dolną krawędź przy pomocy funkcji opisanej w poprzednim punkcie dokumentu. Zmniejsza to wpływ np. powstających zmarszczek na podbródku. Wartości parametrów wynoszą  $m = 1.0$  dla obu krawędzi oraz  $w = 0.35$  dla dolnej części okna i  $w = 0.1$  dla górnej.



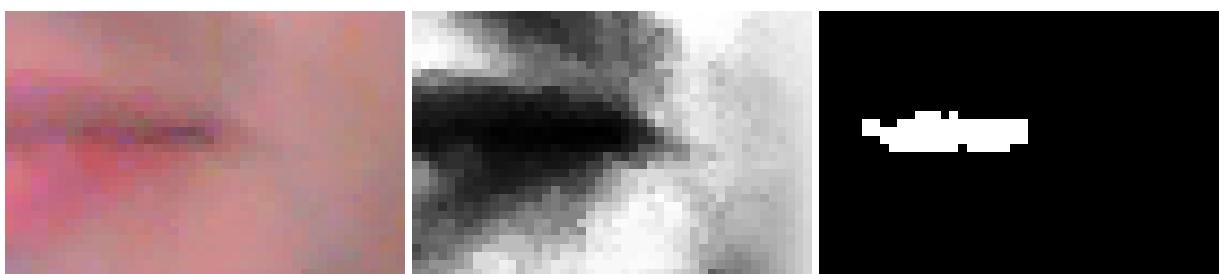
Rysunek 3.12: Od lewej: obraz wejściowy, obraz po mapowaniu omówioną transformacją, obraz po wzmocnieniu kontrastu i oczyszczeniu krawędzi okna

Binaryzacja odbywa się już standardowo. Wybierane jest 10% najciemniejszych pikseli utworzonej mapy. Łączenie segmentów ust odbywa się na podstawie metody dylacji przy pomocy elipsy (okręgu) o rozmiarach  $3 \times 3$ . Uzyskany obiekt jest spójny przy różnych stanach ust, dla większości przypadków. Do fragmentu dopasowywana jest elipsa, co zapewnia odpowiedni poziom generalizacji obiektu i większą stabilność.



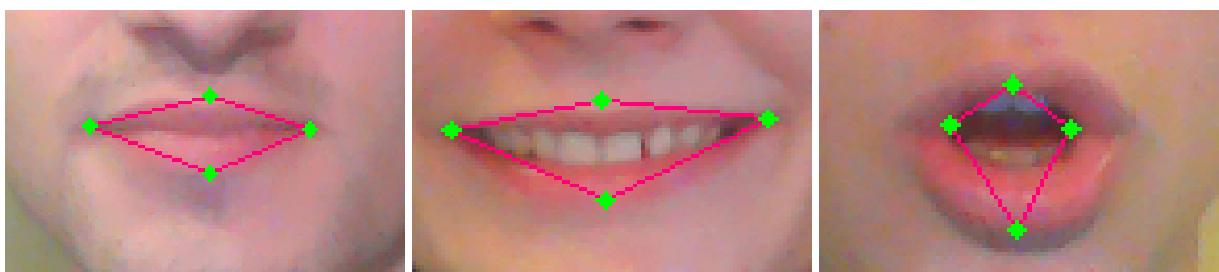
Rysunek 3.13: Od lewej: obraz po binaryzacji, obraz po operacji dylacji, dopasowana do segmentu elipsa i para punktów wertykalnych ust

Przyjęta metoda daje dobre rezultaty dla pary wertykalnej, jednak nie jest dostatecznie stabilna i wiarygodna dla punktów poziomych. Z tego powodu są one wyznaczane niezależną metodą, wykorzystującą fakt, że kątiki ust są ciemniejsze od sąsiadujących obszarów. Aby uzyskać szukane punkty, wyznaczane są dwa nowe okna, mające zawierać rejon wspomnianych kątek. Ich środki stanowią skrajne horyzontalne punkty dopasowanej elipsy, a rozmiary wynoszą 40% szerokości i wysokości okna ust. Na obu oknach wykonywane jest czyszczenie zewnętrznych krawędzi (również wspomnianą funkcją) z parametrami  $m = 0.8$  i  $w = 0.3$ . Następnie przeprowadzane jest wyrównywanie histogramu i binaryzacja z wyborem jedynie 5% najciemniejszych pikseli. W uzyskanych największych segmentach binarnych poszukiwane są najmocniej wysunięte na zewnątrz punkty – są to szukane kątiki, niosące informację o szerokości ust.



Rysunek 3.14: Detekcja prawego kącika ust. Od lewej: obraz wejściowy, obraz po wzmocnieniu kontrastu i oczyszczeniu prawej krawędzi, obraz po binaryzacji

Zastosowane podejście sprawia, że moduł jest w stanie skutecznie radzić sobie z szeroko otwartymi ustami, przy jednoczesnym zachowaniu precyzji detekcji ich kącików, szczególnie przy szerokim uśmiechu.

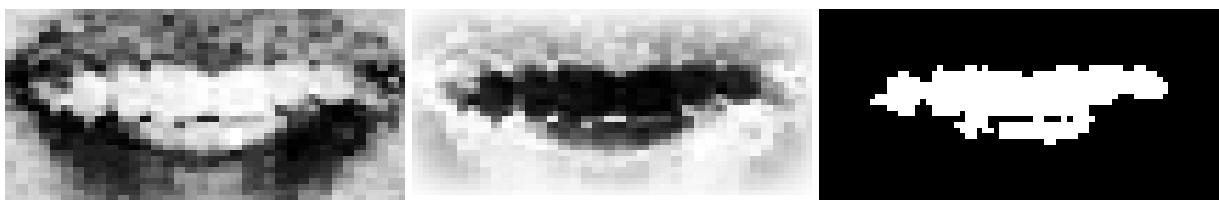


Rysunek 3.15: Przykłady detekcji punktów ust dla różnych stanów i osób

### 3.3.4 Zęby

Wykrycie obecności zębów może w istotny sposób poprawić rozpoznawalność takich intensywnych emocji jak radość czy gniew. Informacja o nich stanowi pewne uzupełnienie dla danych pochodzących z ust, bowiem nie każde ich otwarcie w podobny – pod względem położenia punktów – sposób znaczy to samo. Metoda ich detekcji jest prosta. Polega na odpowiedniej binaryzacji mocno białych zębów i policzeniu stosunku ich liczby do pozostałych pikseli okna.

Podobnie jak w przypadku kącików ust, analizowany obszar jest ustalany na podstawie wykrytych punktów ust. Przy ich wykorzystaniu tworzony jest odpowiedni obszar prostokątny. Do utworzenia mapy zębów używana jest metoda mapowania ust. Dzięki niej jasne punkty wskazują teraz na zęby. Dodatkowo wykonywane jest liniowe czyszczenie krawędzi, tak aby zmniejszyć wpływ widocznej skóry i odblasków na mokrej powierzchni warg. Piksele uzyskanego obrazu są binaryzowane na podstawie dobranego empirycznie górnego progu wynoszącego  $t = 75$  (dla zanegowanego obrazu).



Rysunek 3.16: Od lewej: obraz po mapowaniu ust, zanegowany obraz po czyszczeniu krawędzi, obraz po binaryzacji

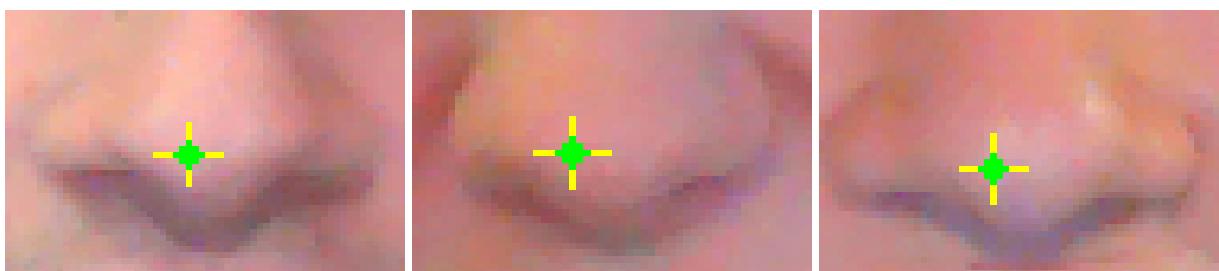
Na koniec obliczany jest wspomniany stosunek pikseli białych do czarnych  $\omega$  w uzy-skany obrazie. Jeśli wynosi on więcej niż 0.1, stwierdzana jest obecność zębów.



Rysunek 3.17: Przykłady detekcji zębów dla różnych stanów i osób. Wartości parametru  $\omega$  wynoszą kolejno 0.216, 0.204, 0.112

### 3.3.5 Nos

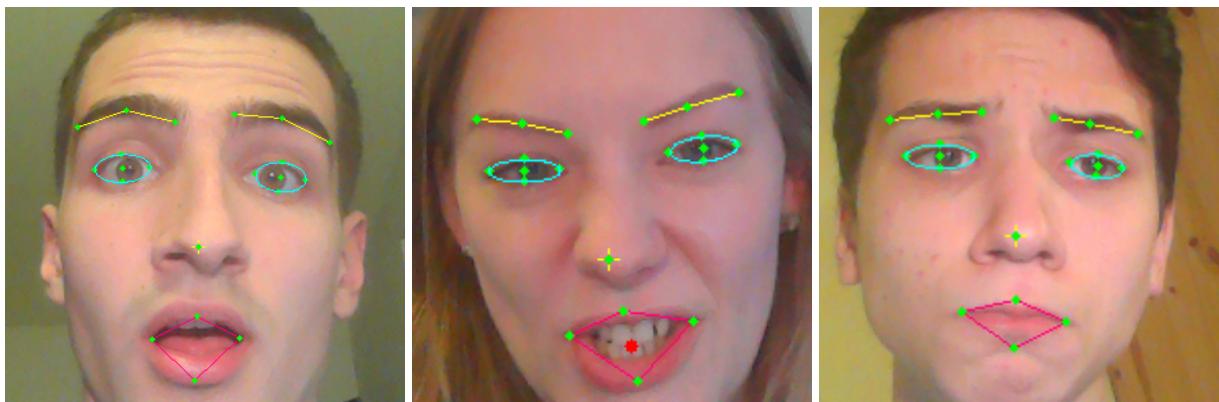
Ostatnim rozważanym obszarem jest region nosa. Autor pracy nie podjął się ekstrakcji bardziej szczegółowych punktów nosa, jak np. nozdrzy. Wykonywana jest jedynie aproksymacja położenia jego czubka przy pomocy jednego punktu. Ze względu na względną stabilność jego współrzędna na osi rzędnych jest dobrym punktem odniesienia dla wychyleń punktów ust, np. przy mierzeniu obniżenia dolnej lub górnej wargi. Skoro wymagane jest jedynie zgrubne oszacowanie, wykorzystywany jest tutaj detektor cech Haara, operujący na znacząco ograniczonym obszarze zaprezentowanym w rozdziale 3.2.2. Kaskada potrzebna do detekcji nosa została pobrana z dostępnego źródła. Jest ona w stanie wykrywać samą jego końcówkę, tak więc szukany punkt czubka to centrum prostokąta zwróconego przez detektor.



Rysunek 3.18: Przykłady detekcji nosa dla różnych stanów i osób

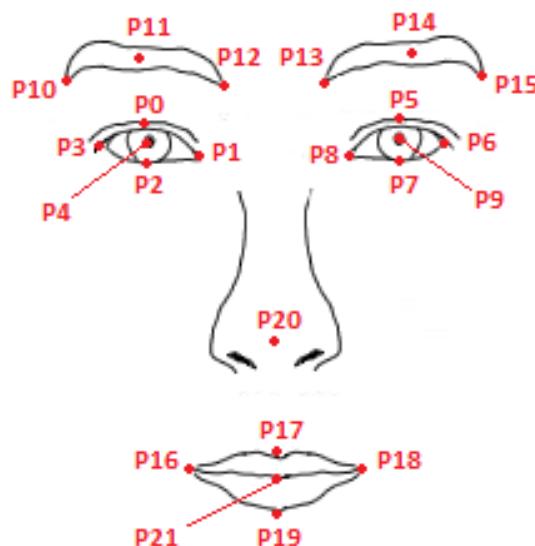
### 3.4 Opis geometryczny ekspresji twarzy

Uzyskane punkty charakterystyczne są zbierane i stanowią podstawę do stworzenia wektora opisującego ekspresję mimiczną, czyli do przeprowadzenia właściwej ekstrakcji cech. Na rysunku 3.19 prezentowane są przykładowe, pełne wyniki procesu z poprzednich punktów pracy.



Rysunek 3.19: Przykłady pełnej detekcji punktów twarzy dla różnych stanów i osób

Również na tym etapie tworzenia opisu ekspresji możliwe są różne podejścia. Często stosowaną metodą jest obliczanie różnic między położeniem punktów twarzy w stanie naturalnym a umiejscowieniem punktów stanu aktualnego [17, 25]. Niewątpliwą zaletą takiego rozwiązania jest zwiększyony stopień uniwersalności – mniejsze zróżnicowanie występuje dla samych ruchów mimicznych różnych użytkowników niż dla kształtów poszczególnych części ich twarzy. Możliwa jest również detekcja pojedynczych jednostek akcji mimicznych (AU) i wykorzystanie systemu regułowego na bazie FACS. Wadą jest konieczność wcześniejszego dostarczenia dobrze dobranego, referencyjnego obrazu twarzy o wyrazie naturalnym. Innym podejściem jest opisanie kształtu poszczególnych części twarzy przy pomocy uzyskanych punktów.



Rysunek 3.20: Wykryte punkty na etapie ekstrakcji

Podstawowe wielkości potrzebne do rozróżniania ekspresji twarzy zostały wybrane przez autora pracy. Są one formalnie reprezentowane jako cechy  $v^{(i)}$ , z których budowany jest wektor  $\mathbf{v} = (v^{(1)}, v^{(2)}, \dots, v^{(n)})$ , gdzie  $n$  jest równe 16. Poniżej zamieszczony został sposób obliczania poszczególnych wartości. Kolejne punkty są parą dwóch współrzędnych  $P_i = (x_i, y_i)$ .

- szerokość lewego oka:  $v^{(1)} = x_1 - x_3$ ,
- wysokość lewego oka:  $v^{(2)} = y_2 - y_0$ ,
- szerokość prawego oka:  $v^{(3)} = x_6 - x_8$ ,
- wysokość prawego oka:  $v^{(4)} = y_7 - y_5$ ,
- odległość między środkami lewego oka i brwi:  $v^{(5)} = y_4 - y_{11}$ ,
- odległość między zewnętrznymi punktami lewego oka i brwi:  $v^{(6)} = y_3 - y_{10}$ ,
- odległość między wewnętrznymi punktami lewego oka i brwi:  $v^{(7)} = y_1 - y_{12}$ ,
- odległość między środkami prawego oka i brwi:  $v^{(8)} = y_9 - y_{14}$ ,
- odległość między zewnętrznymi punktami prawego oka i brwi:  $v^{(9)} = y_6 - y_{15}$ ,
- odległość między wewnętrznymi punktami prawego oka i brwi:  $v^{(10)} = y_8 - y_{13}$ ,
- szerokość ust:  $v^{(11)} = x_{18} - x_{16}$ ,
- wysokość ust:  $v^{(12)} = y_{19} - y_{17}$ ,
- odległość dolnej wargi od nosa:  $v^{(13)} = y_{19} - y_{20}$
- odległość lewego kącika ust od nosa:  $v^{(14)} = y_{16} - y_{20}$
- odległość prawego kącika ust od nosa:  $v^{(15)} = y_{18} - y_{20}$
- punkt zębów:  $v^{(16)} = \omega$  (patrz: 3.3.4)

Z perspektywy następnego etapu działania systemu, tj. klasyfikacji, wskazane jest, aby powyższe wartości cech twarzy znajdowały się w pewnym znormalizowanym przedziale, najlepiej [0-1]. Konieczne jest wybranie odpowiedniej wartości odniesienia (mianownika normalizacji). Powinna być ona większa od wszystkich powyższych wartości i w pewnym stopniu dynamiczna dla różnych twarzy użytkowników. Może się nią być np. szerokość lub wysokość wykrytej twarzy. Aby w bardziej równomierny sposób rozłożyć wartości na przedziale [0-1] wybrana została odległość horyzontalna między zewnętrznymi krańcami lewego i prawego oka. Wielkość ta jest większa od wszystkich pozostałych, ale jednocześnie nie na tyle duża, by przenieść ciężar zbioru w kierunku zera.

# Rozdział 4

## Klasyfikacja emocji

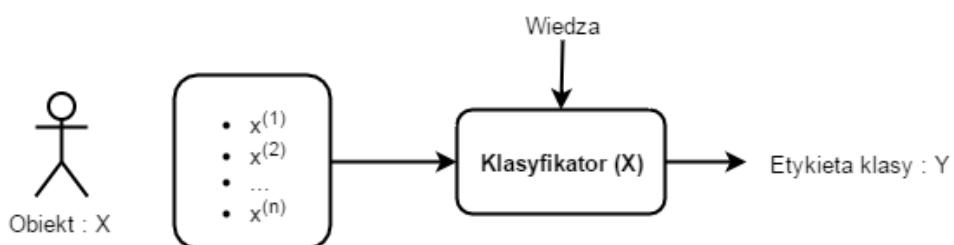
Ostatni etap pracy systemu to podjęcie decyzji o przynależności analizowanej ekspresji mimicznej do jednej z wyróżnionych klas emocji. Rozpoznawanie odbywa się na podstawie wektora cech twarzy, który został wyekstrahowany w poprzedzającym klasyfikację kroku. W niniejszym rozdziale zaprezentowane zostały metody i algorytmy wybrane do rozpoznawania klas emocji w owym wektorze. Ich skuteczność zostanie oceniona na podstawie odpowiednio przygotowanego zbioru danych.

### 4.1 Zadanie klasyfikacji

Klasyfikacja lub inaczej rozpoznawanie to zadanie z pogranicza statystyki i algorytmiki. Może być wykonywana przy pomocy różnych metod i algorytmów, jednak ogólna formuła jest wspólna dla nich wszystkich. Zdefiniujmy cechy  $x \in X$ , gdzie  $X$  to przestrzeń cech oraz decyzje  $y \in Y$ , gdzie  $Y$  to zbiór decyzji, które możemy podjąć. Decyzje mogą przyjmować wartości ze zbioru liczb całkowitych. Dla liczb rzeczywistych mówimy natomiast o zadaniu regresji. Cechy opisują obiekt, który będziemy utożsamiać z jego wektorem cech, tj.  $\mathbf{x}$ . Klasyfikacja jest *de facto* przekształceniem wektora z przestrzeni cech na odpowiedź w przestrzeni decyzji [53]. Dokonuje tego algorytm klasyfikacji:

$$\psi : X \rightarrow Y, \quad (4.1)$$

który zwraca etykietę klasy na podstawie wprowadzonego wektora cech opisującego obiekt, tj.  $\psi(\mathbf{x}) = y$ .



Rysunek 4.1: Ogólny schemat zadania klasyfikacji

W modelu kanonicznym klasyfikatora dla każdej z klas  $j$  wyznacza on wsparcie  $g_j(\mathbf{x})$ . Wartości te są zazwyczaj znormalizowane i mają charakter prawdopodobieństwa. Decyzja podejmowana jest na zasadzie reguły maksymalizującej, czyli:

$$\psi(\mathbf{x}) = j \Leftrightarrow g_j(\mathbf{x}) = \max_i g_i(\mathbf{x}). \quad (4.2)$$

Aby owy algorytm był w stanie poprawnie rozpoznawać obiekty, musi zostać odpowiednio do tego przygotowany. Mówimy o tzw. uczeniu klasyfikatora. Może mieć ono dwie formy: nadzorowaną i nienadzorowaną. W przypadku niniejszej pracy brana jest pod uwagę jedynie pierwsza z metod, druga jest wykorzystywana m.in. do zadań klasteryzacji (grupowania danych). Proces ten to tak naprawdę pokazywanie klasyfikatorowi przykładów cech i poprawnych dla nich odpowiedzi. Dane te są zebrane w postaci zbioru treningowego  $S = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n))$ , czyli par cech i pożąданej odpowiedzi (klasy). Trening powinien zostać wykonany tak, aby algorytm poznał wystarczająco naturę problemu, stworzył jego model i był w stanie rozwiązywać nie tylko dostarczone przykłady, ale również i nowe, z którymi nie miał wcześniej kontaktu. Zjawisko to nazywamy **generalizacją wiedzy**. Nauczenie klasyfikatora nie gwarantuje, że zawsze będzie działał on poprawnie. Niekoniecznie musi on wytworzyć poprawny model. Wpływa na to wiele czynników, jak np. niewystarczający zbiór uczący, źle dobrane parametry algorytmu, warunki niepewności.

Ze względu na swój uniwersalny charakter klasyfikacja może być stosowana w przypadku praktycznie każdego rzeczywistego problemu (choć inną kwestią jest jej efektywność). Należy jedynie przedstawić dowolny obiekt, który chcemy klasyfikować przy pomocy odpowiednio – w zależności od potrzeb – opisującego go zestawu cech. Sposób adaptacji algorytmów sprawia, że nie musimy dokładnie rozumieć natury problemu i konstruować rozwiązań bezpośrednio w sposób analityczny. Klasyfikatory są w stanie same wytworzyć potrzebne wzorce postępowania, by udzielać poprawnych odpowiedzi – wystarczy pokazać im przykłady. Pozwala to na traktowanie ich jako czarne skrzynki (ang. *black box*) i dbanie jedynie o wejście oraz wyjście systemu, które mogą być łatwiej interpretowane nawet przez laików. Oczywiście lepsze wykorzystanie algorytmów klasyfikacji jest możliwe jedynie przy znajomości wpływu ich parametrów na sprawność działania, a to wymaga już bardziej wnikliwej wiedzy.

## 4.2 Wykorzystane klasyfikatory

Uniwersalny charakter algorytmów klasyfikacji sprawia, że praktycznie każdy z nich może zostać z lepszym lub gorszym skutkiem wykorzystany do zadania rozpoznawania emocji na podstawie ekspresji twarzy. Nie istnieją jedyne słuszne algorytmy. Fundamentalne znaczenie ma efektywność wyznaczania modelu twarzy, wybrane punkty oraz parametryzacja jej opisu.

Część możliwych podejść została zaprezentowana w przytoczonych już rozwiązańach innych autorów, np. w pracy [17] wykorzystano naiwne klasyfikatory Bayesa operujące na zmianach położenia trójwymiarowej siatki. Projekt [54] to prezentacja możliwości maszyny wektorów nośnych. Autorzy pracy [55] wykorzystali blokową klasyfikację emocji z użyciem algorytmu k-najbliższych sąsiadów, aplikowanego do kilku rozłącznych obszarów twarzy. Znane są też podejścia z wykorzystaniem binar-

nych drzew decyzyjnych [56]. Proponowanych rozwiązań jest wiele – od bardziej zaawansowanych po prostsze koncepcyjne. Mogą one bazować na standardowych algorytmach lub dokonywać ich modyfikacji. Widoczne jest duże zainteresowanie stosowaniem maszyn wektorów nośnych do detekcji akcji mimicznych, których wyniki mogą być wykorzystywane do indukowania emocji na bazie wybranego systemu regułowego, np. FACS. Często pojawiają się również rozwiązania kombinujące działanie różnych klasyfikatorów.

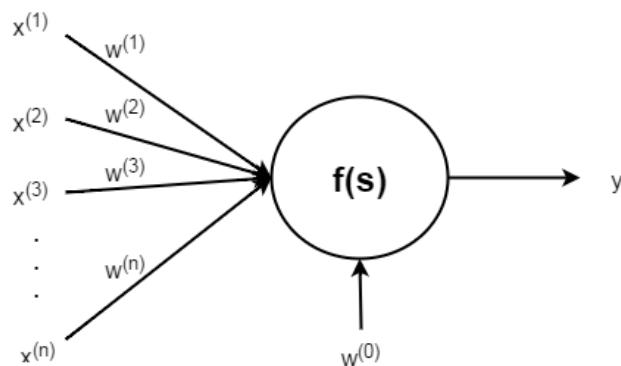
Autor niniejszej pracy wybrał do eksperymentów trzy popularne algorytmy: sztuczne sieci neuronowe, maszynę wektorów nośnych i algorytm k-najbliższych sąsiadów. Różnią się one przede wszystkim pod względem zaawansowania wewnętrznego mechanizmu działania. Poza zastosowaniem indywidualnych algorytmów, zdecydowano się również na zbudowanie prostego ich zespołu. Wszystkie podejścia opierają swoje działanie na bezpośrednim opisie geometrycznym twarzy w postaci jednego wektora.

#### 4.2.1 Sztuczne sieci neuronowe

Sztuczne sieci neuronowe to model obliczeniowy ukierunkowany na podejmowanie decyzji, jego struktura została zainspirowana budową mózgu i jego uproszczoną zasadą działania. Podstawową jednostką przetwarzającą sieci neuronowej jest neuron [53]. Posiada on wejścia  $x^{(i)}$ , przez które doprowadzane są wartości z zewnętrznego środowiska (cechy obiektu) lub z innych neuronów. Każde z nich posiada swoją wagę  $w^{(i)}$ . Wyjście neuronu  $y$  stanowi wartość obliczana na podstawie wewnętrznej funkcji aktywacji  $f(s)$ , operującej na sumie  $s$  pochodzącej od  $n$  sygnałów wejściowych:

$$s = \sum_{i=1}^n w^{(i)}x^{(i)} - w^{(0)}, \quad (4.3)$$

gdzie  $w_0$  to tzw. próg aktywacji, powyżej którego neuron jest aktywowany.



Rysunek 4.2: Model pojedynczego neuronu

Model ten to proste odwzorowanie sposobu działania rzeczywistego neuronu biologicznego. Otrzymuje on sygnały elektryczne i jeśli zostanie dostatecznie pobudzony, generuje własny sygnał, który przekazuje dalej. Wspomniana funkcja aktywacji może

mieć różną postać. Najbardziej rozpowszechniona jest funkcja sigmoidalna, która dla przypadku unipolarnego dana jest wzorem:

$$y = f(s) = \frac{1}{1 + \exp(-\beta s)}, \quad (4.4)$$

gdzie parametr  $\beta$  odpowiada za kształtowanie krzywej pomiędzy funkcją liniową a funkcją progową. Możliwy jest również przypadek bipolarny. Powyższa formuła modeluje więc tak naprawdę pewną hiperpowierzchnię decyzyjną w n-wymiarowej przestrzeni. Klasyfikacja wyjścia odbywa się na podstawie różnicy pomiędzy nim a możliwymi wartościami wyjściowymi. Jeśli mamy dwie klasy oznaczone jako 0 i 1, to przyjęta zostanie ta wartość, która jest bliżej tej zwróconej z funkcji aktywacji neuronu, tj.  $f(s)$ . Owe progowanie nie jest jednak konieczne. Zachowując wartość rzeczywistą z przedziału [0-1] (a taką zwraca funkcja sigmoidalna), można powiedzieć nie tylko na jaką klasę (bądź składową jej kodu) wskazuje neuron, ale również na jakim poziomie pewności jest otrzymana odpowiedź.

Jak już wspomniano, wejście do neuronu mogą stanowić wyjścia z innych. Oznacza to, że tworzą one połączoną strukturę na wzór rzeczywistej sieci neuronowej. Zazwyczaj wyróżniamy w niej trzy rodzaje warstw: wejściową, ukrytą i wyjściową. W takim przypadku mówimy o wielowarstwowej sieci neuronowej. Warstwa ukryta może składać się z wielu warstw, jednak z praktycznego punktu widzenia ich liczba nie przekracza zazwyczaj dwóch lub trzech. Siła takiego modelu wynika z połączonych możliwości pojedynczych neuronów. Wspólnie są one w stanie rozwiązywać nie tylko stosunkowo łatwo separowalne problemy, ale mogą również modelować teoretycznie dowolne zależności, wymagające skomplikowanych, wielowymiarowych przestrzeni. Praktycznym ograniczeniem jest czas i dostępne dane treningowe, co niestety ma kluczowe znaczenie i dotyczy wszystkich metod uczenia maszynowego.

W jaki sposób przebiega proces uczenia takiego algorytmu? Najbardziej popularną i efektywną metodą jest algorytm **propagacji wstecznej**. Celem samego procesu jest utworzenie modelu problemu, generującego możliwe mały błąd na podstawie zbioru treningowego. Jego kształtowanie odbywa się poprzez poprawianie wag wejść neuronów. Dopasowanie  $Q(\mathbf{w})$  pojedynczego neuronu, który posiada wektor wag wejściowych  $\mathbf{w}$ , jest oceniane na podstawie kwadratu odchylenia wartości zwróconej  $y = f(s)$  od wartości docelowej  $d$  przy zadanym wektorze wejściowym  $\mathbf{x}$ . Dla uproszczenia notacji możemy przyjąć, że istnieje  $x^{(0)}$  zawsze równe -1:

$$Q(\mathbf{w}) = \frac{(f(s) - d)^2}{2} = \frac{(f(\mathbf{w}^T \mathbf{x}) - d)^2}{2}. \quad (4.5)$$

Optymalizacja jest przeprowadzana metodą gradientową. Gradient błędu w funkcji wag neuronu wyraża się wzorem [53]:

$$\frac{\partial Q(\mathbf{w})}{\partial \mathbf{w}} = (f(s) - d) \frac{\partial f(s)}{\partial s} \mathbf{x} = (y - d) \beta f'(\mathbf{w}^T \mathbf{x}) [\mathbf{1} - \mathbf{w}^T \mathbf{x}] \mathbf{x}. \quad (4.6)$$

Uzyskany wektor gradientu wskazuje kierunek, w którym następuje największy wzrost błędu. Zmiany, których dokonujemy, powinny minimalizować błąd, tak więc wektor

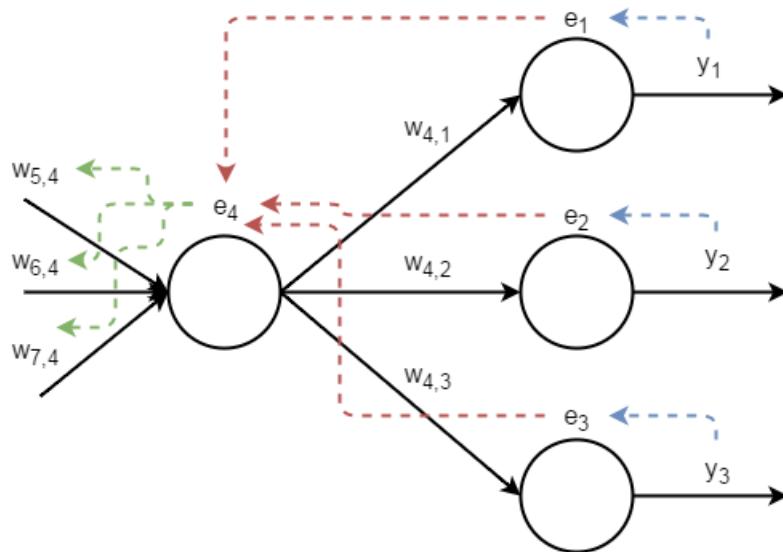
wag połączeń wejściowych jest przesuwany w kierunku przeciwnym do uzyskanego wektora gradientu. Owe aktualizacje mogą być wykonywane w wielokrotnych cyklach uczenia, tzw. epokach. Zmiana wektora wag pojedynczego neuronu w  $k$ -tym kroku wyraża się wzorem:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta \frac{\partial Q(\mathbf{w}_k)}{\partial \mathbf{w}_k}, \quad (4.7)$$

gdzie  $\eta$  oznacza szybkość uczenia lub inaczej intensywność wpływu błędu [57, s. 249-252]. Tak opisana formuła uczenia jest poprawna jedynie dla neuronów warstwy wyjściowej, dla której znamy wartość wyjściową  $y = f(s)$ . Dla neuronów wcześniejszych warstw błąd jest propagowany z warstw po nich następujących. Korekcja ich wag odbywa się zgodnie z formułą:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta \sum_{i=1}^m \frac{\partial Q(\mathbf{w}_{k,i})}{\partial \mathbf{w}_{k,i}}, \quad (4.8)$$

czyli uwzględnia ona błędy pochodzące ze wszystkich  $m$  neuronów w warstwie następnej, z którymi dany neuron jest połączony. Wynika z tego, że podejmowana jest próba dopasowania wag neuronu do wszystkich neuronów, na które on wpływa. Cały proces wykonywany jest dla różnych danych wejściowych, zawartych w danych treningowych.



Rysunek 4.3: Wizualizacja sposobu działania algorytmu propagacji wstecznej dla fragmentu sieci. Błędy  $e_i$  są propagowane na neurony warstw wcześniejszych

Poza omówioną jednokierunkową siecią z propagacją wsteczną i unipolarną funkcją aktywacji, istnieje wiele innych jej odmian, jak np. sieci rekurencyjne czy wielowarstwowe perceptrony. Nie zostały one jednak wykorzystane w opisywanym systemie.

Sieci neuronowe ze względu na swój względnie zaawansowany mechanizm działania cechują się zdolnością do rozwiązywania złożonych problemów, z którymi mogą nie radzić sobie prostsze systemy i algorytmy. Ta zaleta jest jednocześnie pew-

nym utrudnieniem przy korzystaniu z nich. Związane jest to ze skłonnością sieci neuronowych do przeuczania oraz długim czasem treningu. Ze względu na możliwość przeprowadzania niezależnych, równoległych obliczeń obciążenie obliczeniowe może być rozkładane pomiędzy wiele jednostek przetwarzających. W architekturze NIMD (ang. *Neural Instruction Multiple Data*) każdy procesor jest pojedynczym neuronem, a jednostkowe wyniki mogą być dystrybuowane po całej strukturze fizycznie zaimplementowanej sieci [57, s. 236].

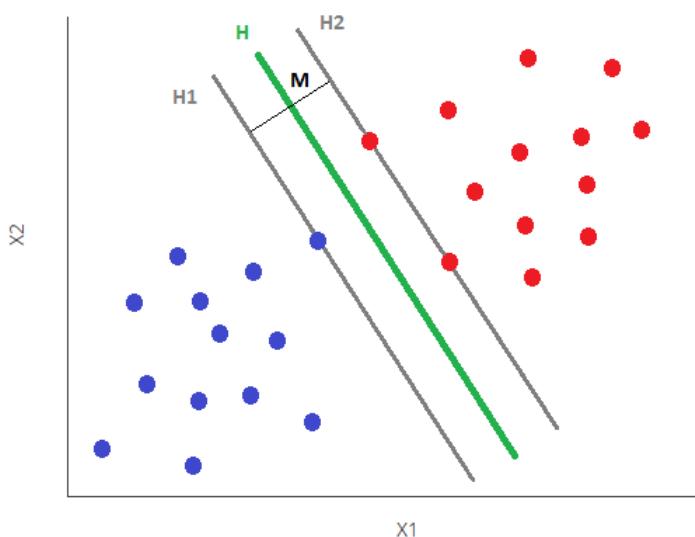
W systemie wykorzystano implementację sieci z biblioteki FANN [59].

### 4.2.2 SVM

Maszyny wektorów nośnych (ang. *support vector machines*, SVM) to algorytmy należące do klasy metod jądrowych (ang. *kernel methods*). Wykorzystują one w swoich obliczeniach tzw. **funkcje jądrowe**, które pozwalają na operowanie w wysokich wymiarach przestrzeni, bez konieczności transformowania samych wektorów cech. W algorytmie SVM dąży się do znalezienia optymalnej hiperpłaszczyzny  $H$  rozdzielającej dwie klasy w przestrzeni cech. Przy standardowym podejściu SVM jest klasyfikatorem binarnym. Założymy, że mamy wektory cech  $\mathbf{x}$  oraz numery klas  $y \in \{-1, 1\}$ . Celem treningu algorytmu jest ukształtowanie takiej funkcji  $f(\mathbf{x})$ , że:

$$f(\mathbf{x}) = \begin{cases} \mathbf{w}^T \mathbf{x} + w^{(0)} \geqslant 1 & \text{dla } y = 1 \\ \mathbf{w}^T \mathbf{x} + w^{(0)} \leqslant -1 & \text{dla } y = -1 \end{cases} \Rightarrow y(\mathbf{w}^T \mathbf{x} + w^{(0)}) \geqslant 1, \quad (4.9)$$

gdzie  $\mathbf{w}$  oraz  $w^{(0)}$  to współczynniki szukanej hiperpłaszczyzny [57, s. 311]. Jej położenie w przestrzeni powinno być takie, by maksymalizowało margines  $M$ , czyli odległość pierwszych punktów obu klas (może być ich więcej niż po jednym), znajdujących się po odpowiednich, przeciwnie skierowanych stronach.



Rysunek 4.4: Prosty przykład wyznaczonej hiperpłaszczyzny  $H$ , separującej dwie klasy w przestrzeni 2-wymiarowej

Wyznaczane są więc również dwie równoległe hiperpłaszczyzny  $H_1$  oraz  $H_2$ , ulokowane w równej odległości od głównej hiperpłaszczyzny  $H$ , takie że:

$$|\mathbf{w}^T \mathbf{x}_h + w^{(0)}| = 1, \quad (4.10)$$

gdzie  $\mathbf{x}_h$  to wspomniane punkty,  $h$  jest indeksem klasy [57, s. 313]. Warunek ten pozwala na uzyskanie kanonicznej postaci  $H$  oraz ujednoznacznienie poszukiwanego rozwiązania. Punkty należące do takich hiperpłaszczyzn definiują je i są nazywane **wektorami nośnymi**. Wektory te stanowią więc podzbiór danych treningowych. Margines jest równy dwóm odległościom dowolnej hiperpłaszczyzny wspomagającej od głównej:

$$M = 2 \frac{\mathbf{w}^T \mathbf{x}_h + w^{(0)}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}. \quad (4.11)$$

Mając określone warunki, należy wyznaczyć wartości szukanego wektora  $\mathbf{w}$ . Zadanie polega na maksymalizacji marginesu, czyli minimalizacji następującego wyrażenia:

$$(\mathbf{w}, w^{(0)}) = \arg \min_{\mathbf{w}, w^{(0)}} \frac{\|\mathbf{w}\|^2}{2}. \quad (4.12)$$

przy ograniczeniu zadanym równaniem (4.9). Zadanie to sprowadza się do optymalizacji kwadratowej z liniowymi ograniczeniami i jest rozwiązywane w sposób analityczny przy pomocy mnożników Lagrange'a. Jej wynikiem jest hiperpłaszczyzna decyzyjna dana wzorem:

$$H : \sum_{i=1}^n (\alpha_i y_i \mathbf{x}_i) \mathbf{x} + w^{(0)} = 0, \quad (4.13)$$

gdzie  $n$  to liczba danych użytych w czasie treningu [58]. Wektorami nośnymi są te punkty cech  $\mathbf{x}_i$ , dla których współczynniki  $\alpha_i > 0$ . Na jej podstawie jest już możliwa poprawna klasyfikacja, która polega na określaniu po której stronie hiperpłaszczyzny  $H$  znajduje się rozważany wektor  $\mathbf{x}$  opisujący nowy obiekt.

Takie podejście zakłada, że problemy są liniowo separowalne. Co jeśli tak nie jest i nie możliwości wyznaczenia bezbłędnej granicy decyzyjnej? Po pierwsze wprowadzany jest współczynnik relaksacyjny  $\zeta \geq 0$ , który zmniejsza margines separacji i jest określany dla każdego przykładu uczącego. Może on doprowadzić do wprowadzenia punktu na obszar marginesu lub nawet spowodować błąd klasyfikacji ( $\zeta > 1$ ). Wartości te powinny być minimalizowane ( $\zeta \rightarrow 0$ ). Warunek ograniczający przyjmuje teraz postać:

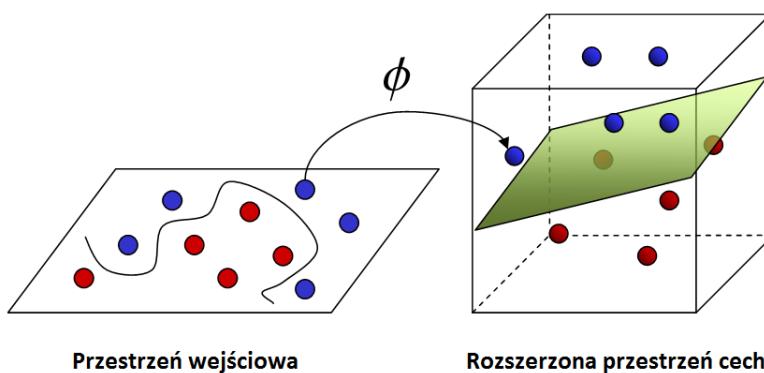
$$y_i(\mathbf{w}^T \mathbf{x} + w^{(0)}) \geq 1 - \zeta. \quad (4.14)$$

Dodatkowo stosowany jest parametr kary  $C$ , określający poziom straty dla każdego punktu. Rozwiążanie optymalne może być otrzymane dla  $C = \infty$ . Jeśli uwzględnimy człon karny, zadaniem, które należy rozwiązać, jest teraz [57, s. 315]:

$$(\mathbf{w}, w^{(0)}, \zeta_i) = \arg \min_{\mathbf{w}, w^{(0)}, \zeta_i} \left( \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n \zeta_i \right). \quad (4.15)$$

Nowa hiperpłaszczyzna różni się od poprzedniej jedynie tym, że współczynniki  $\alpha_i$  wybranych wektorów nośnych są teraz ograniczone:  $0 \leq \alpha_i \leq C$ . Praktyczna interpretacja parametru kary jest taka, że reguluje on stosunek szerokości marginesu do wartości błędu na zbiorze treningowym. Duże wartości  $C$  dają lepsze dopasowanie do zbioru treningowego, ale mogą skutkować osłabioną zdolnością algorytmu do generalizacji problemu. Mniejsze wartości mają tendencję do odwracania tego stosunku.

Algorytm SVM radzi sobie również z problemami nieliniowymi, przeprowadzając transformacje punktów do wyższych wymiarów, w których możliwa jest liniowa separacja.



Rysunek 4.5: Idea rozwiązywania problemów nieliniowych w algorytmie SVM. Źródło: [www.reddit.com](http://www.reddit.com)

Przekształcenia te nie są jednak wykonywane wprost. Wykorzystywana jest wspomniana funkcja jądrowa  $K(\mathbf{x}_i, \mathbf{x}_j)$ , która zwraca wartość skalarną dla dowolnego wymiaru. Możliwość zastosowania takiego podejścia wynika z faktu, że w rozwiązywanym Lagrangianie brane są pod uwagę jedynie iloczyny skalarne  $\mathbf{x}_i^T \mathbf{x}_j$  [57, s. 319-321]. Potencjalnych transformacji  $\phi(\mathbf{x})$  do przestrzeni o wyższym wymiarze możemy uniknąć właśnie dzięki prostszej, odpowiednio dobranej funkcji jądrowej. Zwróci nam ona wymaganą wartość w oryginalnej przestrzeni, bez czasochłonnych transformacji i iloczynów skalarnych w przestrzeni rozszerzonej:

$$\phi(\mathbf{x}_i)\phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j). \quad (4.16)$$

Okazuje się, że istnieje wiele takich przekształceń i związanych z nimi funkcji jądrowych, jak np. liniowe, wielomianowe rzędu  $p$  czy radialne (ang. *radial-basis functions*, RBF).

Mimo, że pojedynczy algorytm SVM jest w stanie rozróżnić jedynie dwie klasy, istnieje możliwość jego stosowania do problemów wieloklasowych. Najbardziej popularnym rozwiązaniem jest podejście *one-vs-all*, w którym dla każdej klasy tworzony jest osobny klasyfikator, a ostateczna odpowiedź to najpewniejsza, pozytywna decyzja jednego z nich.

Duża popularność algorytmów SVM nie jest przypadkowa. Wpływa na to m.in. analityczne znajdowanie optymalnego rozwiązania, szybkie obliczenia z funkcjami jądrowymi, radzenie sobie ze złożonymi problemami nieliniowymi oraz ograniczona do minimum potrzeba heurystycznego dopasowywania parametrów uczenia.

W systemie użyto implementacji algorytmu z biblioteki OpenCV 3.0 [34].

### 4.2.3 Algorytm k-NN

Algorytm k-najbliższych sąsiadów (ang. *k-nearest neighbours*, k-NN) to podejście o wiele prostsze niż dwa wcześniej opisane. Jego zrozumienie jest intuicyjne: badany obiekt  $\mathbf{x}$  należy do klasy  $y$ , jeśli znajduje się bliżej innych obiektów należących do niej niż do pozostałych. Analizowane jest otoczenie złożone z  $k$  sąsiadów, decyzja podejmowana jest zazwyczaj przez głosowanie większościowe. Obiekt  $\mathbf{x}_i$  jest bliżej obiektu  $\mathbf{x}_j$ , jeśli jest do niego bardziej podobny w rozważanej przestrzeni cech. Aby ocenić odległość dwóch wektorów od siebie, konieczny jest wybór jednej z metryk. Najczęściej stosowaną jest metryka euklidesowa dana wzorem:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i - \mathbf{x}_j \| = \sqrt{\sum_{m=1}^n (x_i^{(m)} - x_j^{(m)})^2}, \quad (4.17)$$

gdzie  $n$  oznacza wymiar przestrzeni, w której znajdują się porównywane punkty. innymi popularnymi metrykami są manhattańska i Mahalanobisa. Uczenie takiego algorytmu to jedynie wprowadzenie punktów wektora treningowego z docelowymi etykietami klas. Przyjęta liczba sąsiadów  $k$  i metryka  $d(\mathbf{x}_i, \mathbf{x}_j)$  jednoznacznie definiują podział przestrzeni decyzyjnej.

Choć metoda k-NN jest bardzo prymitywna pod względem swojego aparatu matematycznego, często jest ona w stanie nie tylko dorównywać bardziej wysublimowanym podejściom, ale nawet działać lepiej od nich. Wynikać to może z pewnej swoistej naturalności tego podejścia do rozwiązywania problemów. Niewątpliwą zaletą algorytmu jest bardzo duża szybkość działania. Wadą jest ścisła zależność od zbioru treningowego. Jeśli dane nie są odpowiednie, nie jest praktycznie możliwe, by nadrobić to konfiguracją algorytmu.

Podobnie jak w przypadku algorytmu SVM skorzystano z implementacji zawartej w bibliotece OpenCV 3.0 [34].

### 4.2.4 Zespół klasyfikatorów

Zaprezentowane dotąd algorytmy klasyfikacji bazują jedynie na własnych, indywidualnych możliwościach. Istnieje jednak inne podejście, w którym dąży się do wykorzystania potencjału współdziałania wielu takich algorytmów. Mówimy o tzw. zespołach (ang. *ensemble*) klasyfikatorów. Zdefiniujmy zbiór treningowy  $S$  oraz pojedyncze klasyfikatory  $\psi_i(\mathbf{x})$ . Każdy z nich uczony jest na podstawie zadanego zbioru lub jego części. Decyzje poszczególnych  $n$  algorytmów są kombinowane ze sobą na określony sposób w module **metaklasyfikatora**  $\Phi$ , który zwraca ostateczną odpowiedź  $y$ :

$$y = \Phi(\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \dots, \psi_n(\mathbf{x})). \quad (4.18)$$

Cz³on kombinacyjny mo¿e by realizowany na róne sposoby. Wyróniane s trzy zaadnicze podejcia [53]:

- selekcja – wybór oparty jest na miarach kompetencji klasyfikatorów, badany jest kady z nich, ale wybierany ten, który zwraca najpewniejsze/najlepsze wyniki,
- fuzja – czyli scalanie odpowiedzi klasyfikatorów, moe by realizowane np. w formie głosowania większociowego lub waonego, moliwe s róne modyfikacje jak np. boosting,
- hybryda – podejcie selektywno-fuzyjne, czyli aczenie decyzji najlepszych klasyfikatorów.

Podejcia te s zaliczane do grupy kombinacji wieloekspertowych [57, s. 423], posiadajcych równeleg charakter dziaania. Wyróni mona równeleg kombinacje wieloetapowe, w których klasyfikatory dziaaj膮 sekwencyjnie. S one sortowane pod wzgledem zloonoci. Bardziej zaawansowane algorytmy beda podejmowa decyzje dopiero, gdy prostsze zwróa pewn odpowiedz, np. na mniej zloonym wycinku danych. Przykadem s, opisane wczeej, kaskadowe klasyfikatory cech Haara.

Podstawow cecha, jak musza charakteryzowa si zespoły klasyfikatorów, jest ich **dywersyfikacja** (zrónicowanie) [53]. Ma to znaczenie, poniewa kady z klasyfikatorów powinien analizowa problem z innej perspektywy, dostarcza innego rodzaju informacji. Tylko w takim przypadku metody kombinujce rozwizania maj sens (np. przy głosowaniu). Dywersyfikacja moe by realizowana na róne sposoby. W przypadku wykorzystywania odmiennych klasyfikatorów lub tego samego typu, ale o innych parametrach mówimy o systemach heterogenicznych. Jeli za sosujemy takie same algorytmy, ale zmieniamy form danych lub ich zbiór, mamy do czynienia z zespołem homogenicznym.

Tworzenie zespołów klasyfikatorów to zloony problem, na który wpływ ma wiele czynników. Oczywistą zaletą tego typu rozwizań jest moliwoc faktycznego uzyskiwania wyszej jakości wyników. Utrudnieniem jest zwikszona zloonoc obliczeniowa i pamiciowa oraz trudnoc w jednoznacznej interpretacji wyników. W przeprowadzonych eksperymetach stworzono prosty heterogeniczny zespł zloony z pojedynczych instancji opisanych wczeej algorytmów. Decyzja o przynaleoci obiektu do klasy podejmowana jest na zasadzie głosowania większociowego.

# Rozdział 5

## Badanie klasyfikatorów

Efektywność zastosowania każdego z wymienionych klasyfikatorów została empirycznie zbadana. Rozdział ten obejmuje opis wykorzystanych zbiorów danych, sposobu mierzenia sprawności algorytmów oraz prezentację i dokładną analizę uzyskanych wyników.

### 5.1 Zbiory danych

Aby możliwe było wykorzystanie opisanych algorytmów oraz zbadanie skuteczności ich działania, konieczne było przygotowanie odpowiedniego zbioru uczącego. Autor pracy zdecydował się na utworzenie własnej bazy zdjęć twarzy. Zbierane obrazy twarzy powinny spełniać wymagania postawione w podrozdziale 1.5:

- obraz w formacie RGB,
- obraz frontalny twarzy,
- równomierne oświetlenie, dostateczna ostrość i rozdzielczość,
- twarz wyraża jeden z 7 wybranych przez autora stanów (neutralny, radość, zdziwienie, gniew, smutek, zniesmaczenie, strach).

Podczas kolekcjonowania bazy zdjęć, należało również zadbać o optymalną odległość siedzenia użytkownika. Zbyt duża odległość skutkowała bowiem niską rozdzielczością samej twarzy. Nie stwierdzono zaś problemów z siedzeniem zbyt blisko, system dobrze radzi sobie także z wyższym poziomem szczegółowości twarzy. Wymagane było również **wyraźne akcentowanie** okazywanych emocji – czasami były one w pewnym stopniu przerysowane. Wynikało to z potrzeby kompensacji naturalnej trudności wywoływanego silnych, zauważalnych emocji na zwołanie.

Obrazy zbierano przy pomocy prostego interfejsu, umożliwiającego użytkownikowi przechwytywanie z kamery wideo pojedynczej klatki. Użytkownik siedział przed komputerem, wyrażał określone emocje ekspresją twarzy, obserwował efekt działania ekstraktora cech i naciskał przycisk w momencie mniej więcej poprawnej detekcji punktów. Zapamiętywany był obraz twarzy i przypisany do niego wektor cech (plik

tekstowy). W większości przypadków wymagało to pewnego **dopasowania się użytkownika** do możliwości ekstraktora (np. odpowiednie położenie głowy, odgarnięte włosy). Obrazy z niedopuszczalnie błędna (np. połączenie segmentu oka z brwią) detekcją punktów były ignorowane.

W celu zwiększenia stopnia ogólności wyników testów, w zbieraniu obrazów brały udział trzy osoby – jedna kobieta (U2) i dwóch mężczyzn (U1, U3). Cechowały się one innym charakterem budowy poszczególnych części twarzy, co jest wyraźnie widoczne na obrazach zamieszczonych w poprzednich rozdziałach. Poniżej prezentowane są przykładowe klatki zebrane przez poszczególnych użytkowników. Widoczne są podobne, aczkolwiek nie jednakowe warunki oświetleniowe.



Rysunek 5.1: Przykładowe obrazy treningowe zabrane przez użytkowników, kolejno: U1, U2, U3

W efekcie opisanego procesu zebrano ponad 1000 obrazów twarzy, odpowiadających im obrazów z detekcją punktów i plików z wektorami wyekstrahowanych cech (1GB danych). Średnio na jednego użytkownika przypadało po 50 zdjęć dla każdej emocji. Do treningu klasyfikatorów wybrano około połowę najlepszych z nich, prezentujących odpowiedni poziom zróżnicowania przypadków. Z wybranych obrazów 10-15 przeznaczono bezpośrednio do treningu pozostałe zaś do testowania.

Istotne z punktu widzenia klasyfikatorów były oczywiście same pliki opisujące ekspresję twarzy. Wykonano odpowiednie scalanie pojedynczych obserwacji w ich zbiorze przy pomocy skryptów w powłoce Bash i w interpretowanym języku Perl. W każdym takim pliku znaleźć można wiersze poszczególnych przypadków, na które składają się kolumny cech oraz etykieta wyrażanej klasy emocji. Utworzono dwa typy takich plików: składające się z obserwacji jedynie dla użytkownika, od którego pochodzą oraz zawierające przykłady od wszystkich z nich. Miało to na celu umożliwienie testowania klasyfikatorów pod kątem podejścia *user-dependent* oraz *user-independent*.

## 5.2 Plan eksperymentu

W ramach zrealizowanych eksperymentów zdecydowano się na przeprowadzenie pomiarów skuteczności klasyfikacji poszczególnych algorytmów dla każdej z emocji. Podstawowe testy podzielono na dwa podejścia. Pierwsze to *user-dependent*, w którym kształtowane są indywidualne, dedykowane klasyfikatory, uczone i testowane na danych pochodzących tylko od jednego użytkownika (dla każdego osobno). Przy podejściu *user-independent* następuje próba budowania uniwersalnych, wspólnych dla wszyst-

kich użytkowników klasyfikatorów. W pierwszym przypadku na każdego użytkownika przypadało około 25 obrazów dla każdej emocji (jak opisano w poprzednim punkcie). Przy drugim podejściu tworzony był jeden wektor, zawierający ponad 300 obrazów. Podzielono go na dane treningowe i testowe w takim samym stosunku jak dla rozwiązań indywidualnych. Wyznaczona skuteczność stanowi stosunek ilości poprawnych klasyfikacji do wszystkich podanych danych.

W testach wykorzystano dostępne implementacje algorytmów, o których wspomniano w poprzednim podrozdziale. Parametry klasyfikatorów ustalono empirycznie tak, aby zwracały możliwe najlepsze wyniki. Ich głębsza analiza nie została objęta testami. Parametry wybrano w następujący sposób:

- **NN** (sieć neuronowa):
  - konfiguracja sieci: 16-7-7 (klasy w kodzie 1 z n),
  - funkcja aktywacji: sigmoidalna, unipolarna,
  - liczba epok: 1000,
  - współczynnik uczenia : 0.01,
- **SVM** (maszyna wektorów nośnych):
  - funkcja jądrowa: RBF,
  - parametr C: 10,
- **k-NN** (k-najbliższych sąsiadów):
  - parametr k: 3.

Zespół klasyfikatorów (**Ensemble**) zbudowano na podstawie algorytmów z przedstawionymi parametrami. Do podejmowania decyzji wykorzystano prostą metodę głosowania większościowego.

Dodatkowo dla jednego użytkownika przeprowadzono badanie poziomu wsparcia sieci neuronowej dla poszczególnych emocji w trakcie sekwencji wideo. Test tego typu pozwala na uzyskanie ogólniejszego obrazu efektywności działania całego systemu, co zostało poruszone w kolejnym podrozdziale, zawierającym analizę uzyskanych wyników.

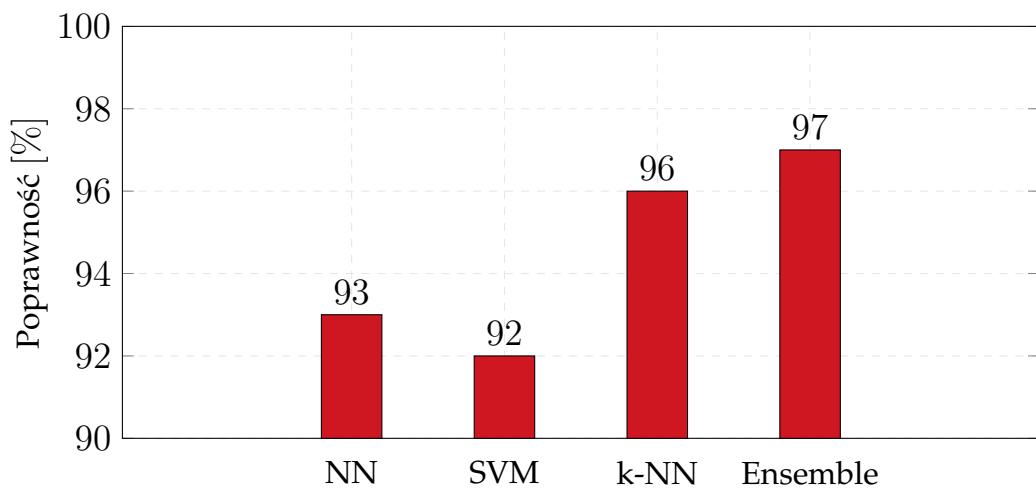
### 5.3 Wyniki

W kolejnych punktach zamieszczone zostały wyniki poszczególnych eksperymentów wraz z ich podstawową analizą. Ostateczna ocena skuteczności rozwiązania jest przeprowadzana w następnym podrozdziale. W prezentowanych wynikach emocje oznaczono ich trzema pierwszymi literami, np. *Neutralny - Neu*, *Zniesmaczenie - Zni*. Wynika to z potrzeby zachowania czytelności.

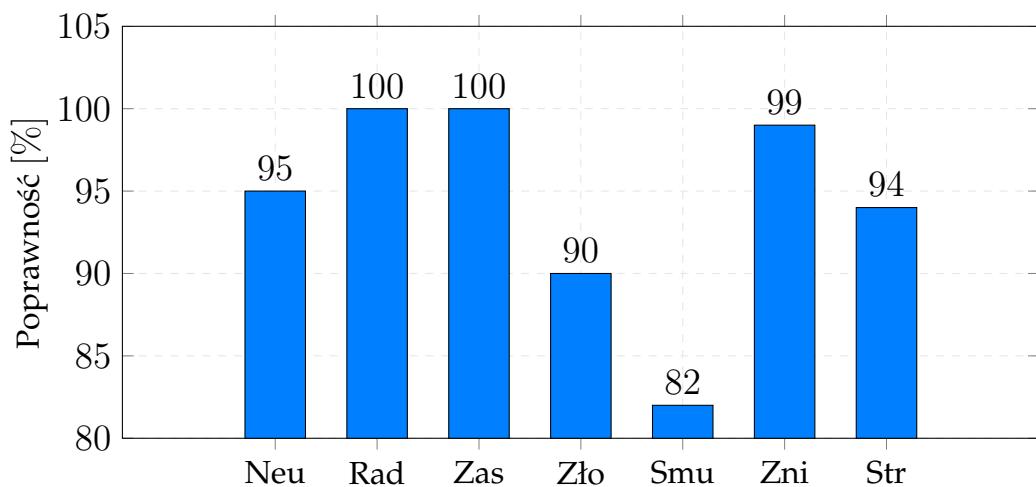
### 5.3.1 Podejścia *user-dependent* i *user-independent*

Tabela 5.1: Uzyskane średnie wyniki testów *user-dependent* dla różnych algorytmów i emocji

—	Neu	Rad	Zas	Zło	Smu	Zni	Str
<b>NN</b>	90%	100%	100%	87%	80%	100%	100%
<b>SVM</b>	97%	100%	100%	80%	80%	100%	90%
<b>k-NN</b>	93%	100%	100%	100%	90%	97%	93%
<b>Ensemble</b>	100%	100%	100%	93%	90%	100%	93%



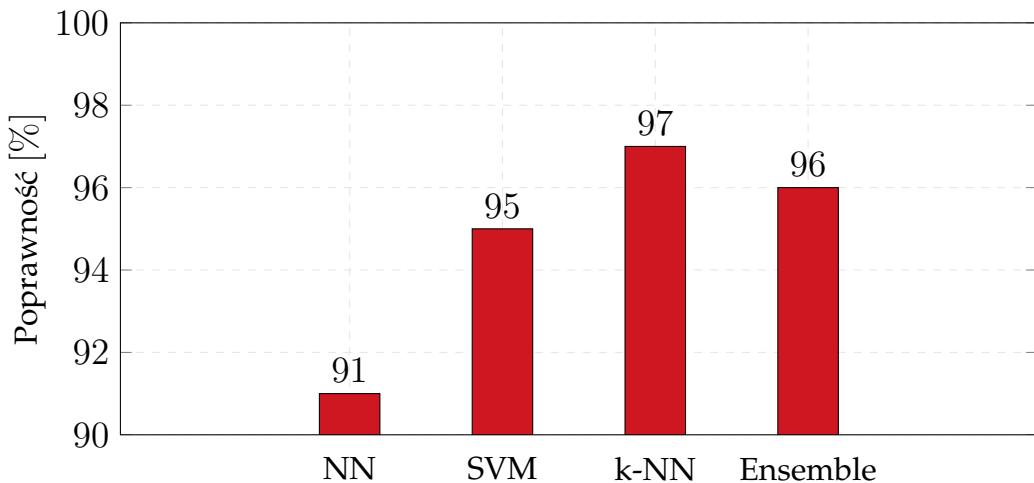
Rysunek 5.2: Średnie wyniki testów *user-dependent* dla poszczególnych klasyfikatorów



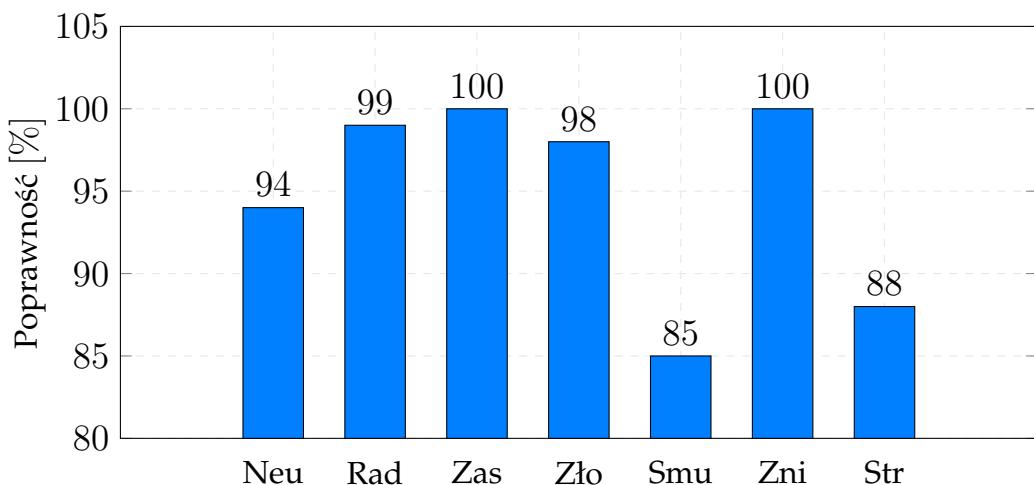
Rysunek 5.3: Średnie wyniki testów *user-dependent* dla poszczególnych emocji

Tabela 5.2: Uzyskane średnie wyniki testów *user-independent* dla różnych algorytmów i emocji

—	Neu	Rad	Zas	Zło	Smu	Zni	Str
<b>NN</b>	83%	97%	100%	97%	70%	100%	93%
<b>SVM</b>	97%	100%	100%	97%	90%	100%	80%
<b>k-NN</b>	97%	100%	100%	100%	90%	100%	93%
<b>Ensemble</b>	97%	100%	100%	97%	90%	100%	87%



Rysunek 5.4: Średnie wyniki testów *user-independent* dla poszczególnych klasyfikatorów



Rysunek 5.5: Średnie wyniki testów *user-independent* dla poszczególnych emocji

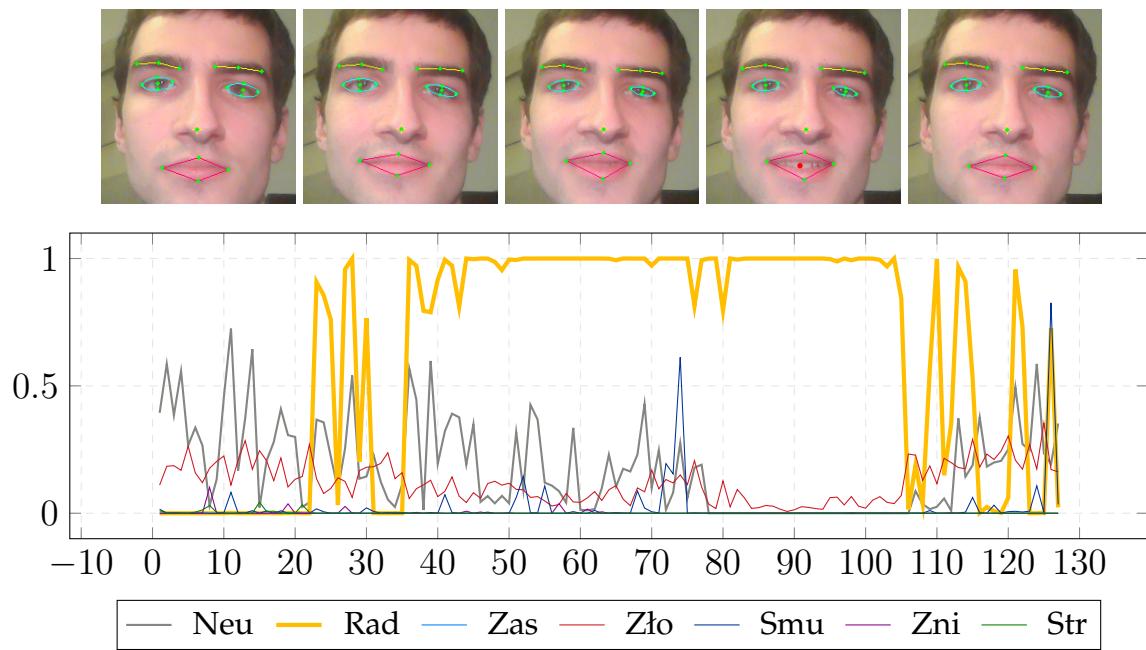
Zaprezentowane wyniki z pewnością można uznać za wysokie. Dla obu rodzajów testów uzyskano 97% skuteczności. Najlepiej spośród czterech klasyfikatorów dla obu rodzajów testów wypadły algorytmy **k-NN** i **zespoł**. Warto zaznaczyć, że pierwszy z nich to najprostszy pod względem mechaniki algorytm, zaś drugi najbardziej rozbudowany. Sieć neuronowa i SVM uzyskały nieco gorsze wyniki. Wciąż jednak były one

na wysokim poziomie – powyżej 90%. Wprowadzenie podejścia *user-independent* praktycznie nie wpłynęło na zdolność systemu do rozpoznawania emocji użytkowników. Pokazuje to, że algorytmy bardzo dobrze radzą sobie z generalizacją danych. Prawdopodobnie możliwe jest uzyskanie lepszych wyników dla sieci neuronowej. Wymagałoby to przeprowadzenia dodatkowych badań z różnymi parametrami uczenia i liczbą epok. Widoczne jest także, że w niektórych przypadkach (głównie dla *user-dependent*) można wyróżnić algorytmy lepiej radzące sobie z poszczególnymi emocjami, np. wykrywanie stanu neutralnego najlepiej wykonywane było przez zespół, złości przez k-najbliższych sąsiadów, zaś strachu przez sieć neuronową.

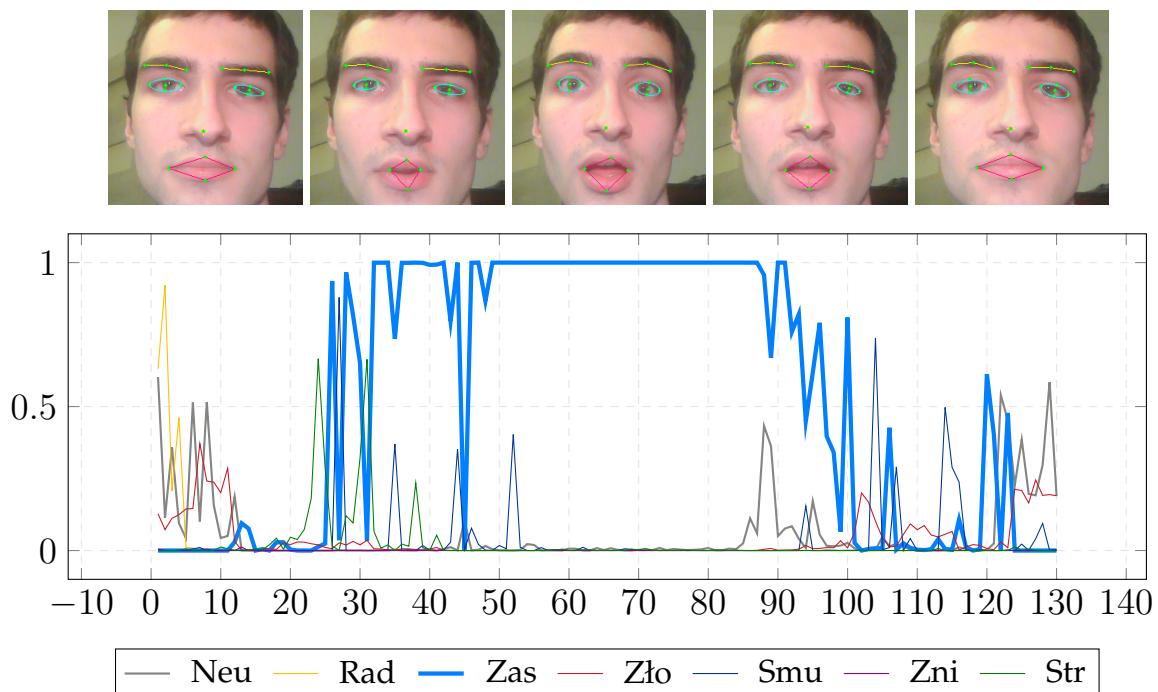
Interesująco prezentuje się zestawienie dla poszczególnych emocji. Najtrudniejszymi do rozpoznania okazały się smutek (82% i 85%) oraz strach (94% i 88%). W pierwszym przypadku wynika to z naturalnej trudności z wymuszonym wyrażaniem smutku. Strach natomiast był czasami mylony z zaskoczeniem, co ma uzasadnienie w opisanej teorii emocji. Jeśli brać pod uwagę wzgłydy teoretyczne, to system dobrze sobie radził z rozróżnianiem złości i zniesmaczenia. Zauważalny jest duży wzrost efektywności w rozpoznawaniu złości przy przejściu z podejścia *user-dependent* na *user-independent* (kolejno 90% i 98%). Okazuje się w tym przypadku, że zróżnicowanie informacji nie jest utrudnieniem, a wręcz przeciwnie, umożliwia zbudowanie bardziej wiarygodnych klasyfikatorów. Najlepiej rozpoznawane były emocje o wyraźnej ekspresji i łatwe do zaprezentowania, tj. radość (100% i 99%) oraz zaskoczenie (100% w obu przypadkach).

### 5.3.2 Testy dla sekwencji obrazów

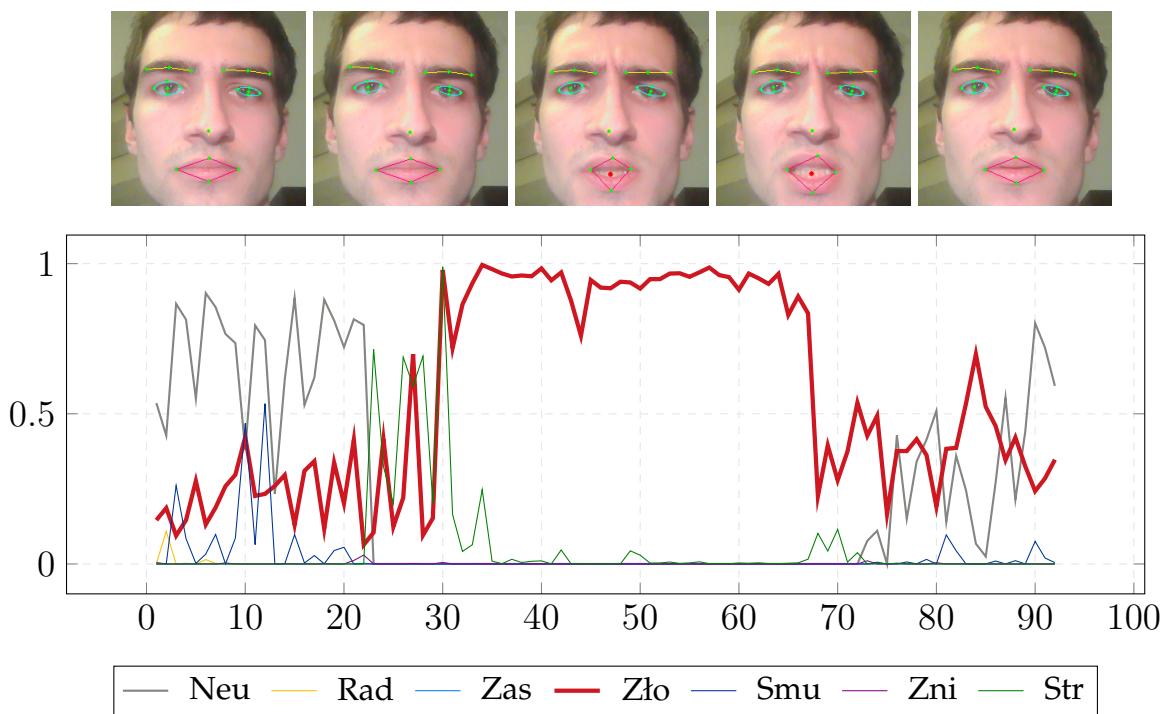
W prezentowanych wynikach badania sekwencji zamieszczone zostały pojedyncze klatki w miejscach mniej więcej odpowiadających momentowi ich pobrania (**osi X** to numer klatki). Wykresy obrazują poziom mierzonego wsparcia (**osi Y**) sieci neuronowej dla poszczególnych klas. Ze względu na czytelność pominięto podpisy osi. Wartości dla docelowej emocji zostały odpowiednio pogrubione.



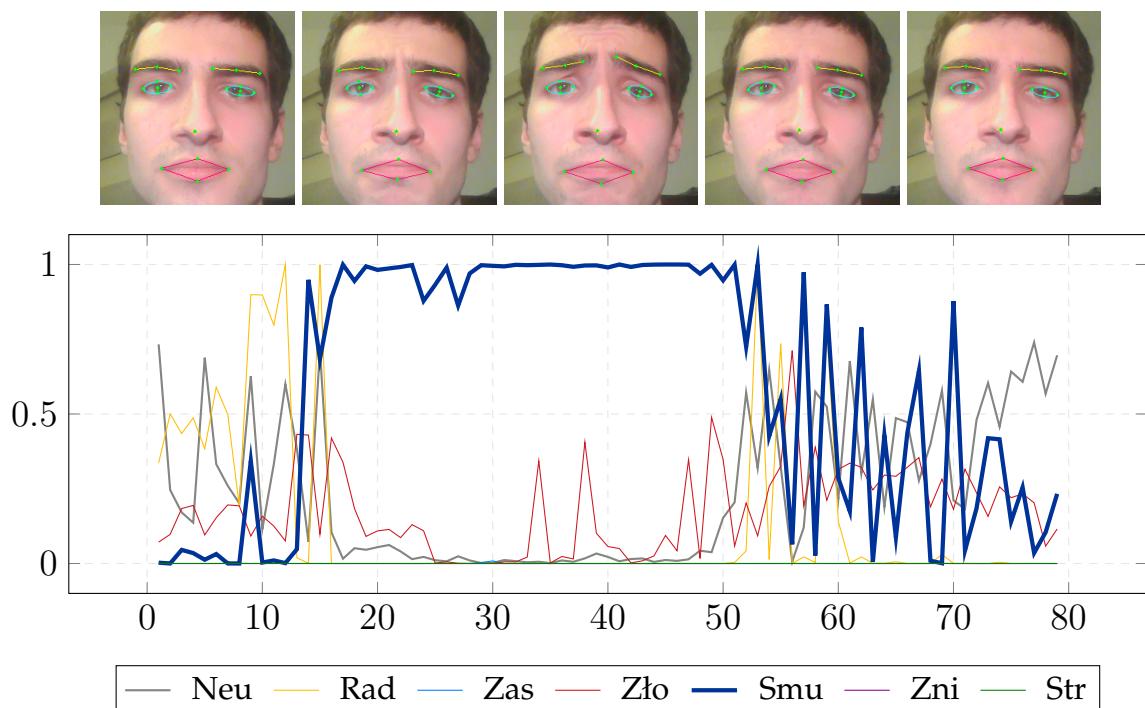
Rysunek 5.6: Poziom wsparcia dla każdej emocji w sekwencji z ekspresją radości



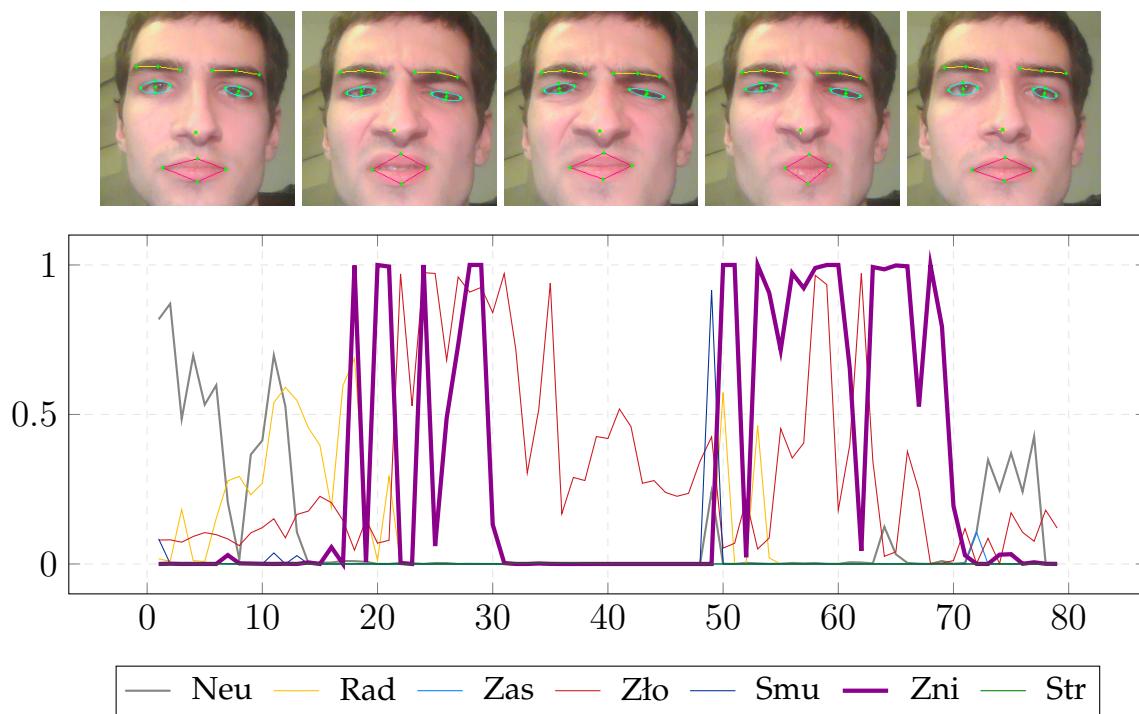
Rysunek 5.7: Poziom wsparcia dla każdej emocji w sekwencji z ekspresją zaskoczenia



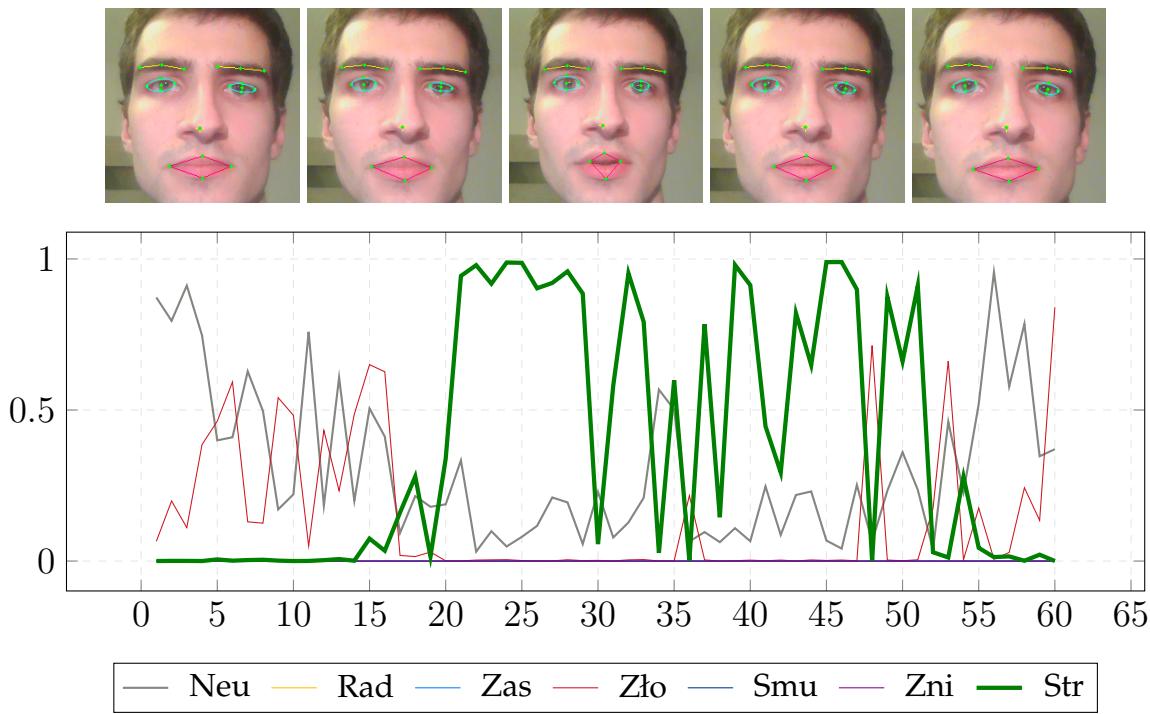
Rysunek 5.8: Poziom wsparcia dla każdej emocji w sekwencji z ekspresją złości



Rysunek 5.9: Poziom wsparcia dla każdej emocji w sekwencji z ekspresją smutku



Rysunek 5.10: Poziom wsparcia dla każdej emocji w sekwencji z ekspresją **zniesmaczenia**



Rysunek 5.11: Poziom wsparcia dla każdej emocji w sekwencji z ekspresją **strachu**

Powyższe wykresy niosą ze sobą kluczowe informacje o charakterze działania systemu oraz są pewnym obrazem naturalnych zależności pomiędzy niektórymi emocjami. Z racji tego, że owe testy zostały przeprowadzone tylko dla jednej osoby z użyciem

wyłącznie sieci neuronowej, wszelkie wyciągnięte wnioski mają ograniczony poziom ogólności. Można je jednak w pewnym stopniu generalizować ze względu na fakt użycia modeli klasyfikatorów typu *user-independent*, a więc wspólnych dla wszystkich.

Zauważalna jest stabilność rozpoznania dla emocji łatwych do wyraźnego zaakcentowania (radość, zaskoczenie, złość) – to konieczna z funkcjonalnego punktu widzenia cecha. Przy osłabionej intensywności ekspresji – momenty narastania i opadania – widoczne są często wahania o wysokiej amplitudzie i podwyższonej częstotliwości. Dzieje się tak w przypadku np. radości (klatki 20-40) czy zaskoczenia (25-35). Poziom wsparcia dla emocji w takich momentach może zmieniać się diametralnie i ciężko uznać takie wyniki za wiarygodne. Nieco lepiej prezentują się momenty kończące wyrażanie danej emocji. Część pomyłek wynika z nieprecyzyjnej detekcji lub poważnych błędów w jej zakresie. Inne związane są z **inherentną cechą samych ekspresji** – stają się one do siebie podobne w miarę słabnięcia. Wrażliwość systemu jest więc ograniczona. Ze względu na swoje częściowo naturalne podłożę, ciężko jednoznacznie traktować to jako wadę techniczną.

Pewnym mankamentem systemu jest stosunkowo niski poziom pewności dla stanu neutralnego. Utrzymuje się on przeważnie na poziomie 50%, co sprawia, że jest on podatny na skorelowane z nim emocje. Takie powiązanie widać np. ze złością i w mniejszym stopniu ze smutkiem. Ich opis geometryczny jest podobny – brwi na średnim lub niskim poziomie, niewielkie zmiany w stopniu otwarcia oczu, zamknięte usta (umiarowana złość). Problem podobieństwa emocji jest kluczowym powodem mniejszej stabilności systemu dla ekspresji zniesmaczenia i strachu, nawet przy ekstremum ich okażywania. Zniesmaczenie może zostać łatwo pomycone ze złością, w szczególności z łagodniejszą jej formą. Widać to na rysunku 5.10. Poziom złości niemal przez cały czas utrzymywany jest na wysokim poziomie, a pomiędzy 30 i 50 klatką jest ona dominującą emocją, co powoduje błędne rozpoznanie. W przypadku dość subtelnego strachu co prawda nie pokazała się inna konkuruje emocja, ale w praktyce łatwo o pomyłenie jej przez system z zaskoczeniem. Tak jak w przypadku stanu neutralnego, wynika to z naturalnych podobieństw pomiędzy wspomnianymi stanami. Było to komunikowane we wprowadzeniu teoretycznym dokumentu. Problem ten jest w dużej mierze problemem koncepcyjnym, dotyczącym określenia rozróżnialnych emocji. Możliwości technicznego rozwiązania takiego problemu są ograniczone.

Warto zaznaczyć, że wspomniane zmiany związane z błędami mają często charakter pseudo impulsów, tzn. trwają stosunkowo krótko (np. 45 klatka przykładu dla zaskoczenia, 34 klatka dla strachu). Przekłada się to na wyniki pomiarów efektywności rozpoznawania. Skoro bowiem pomyłki rzadko mają charakter ciągły i względnie trwały (jak w przykładzie zniesmaczenia), to przy badaniu pewnej liczby klatek stanowią one zdecydowaną mniejszość. Ma to znaczenie praktyczne, ale jest mniej widoczne ze statystycznego punktu widzenia. Pewnym sposobem na opanowanie tego typu błędów byłoby podejście z zakresu przetwarzania sygnałów jednowymiarowych. Mogłoby ono polegać na wprowadzeniu pewnej inercji zmian, np. poprzez wprowadzenie filtrów uśredniających.

Badaniom tego typu należałyby poświęcić zdecydowanie więcej czasu i uwagi. Są one bardzo istotnym źródłem wiedzy zarówno na gruncie czysto technicznym (możliwość zmiany spojrzenia na problem) jak i teoretycznym (perspektywa głębszej analizy zależności pomiędzy sposobami wyrażania różnych emocji). Ponadto pozwalają one na lepszą ocenę całego rozwiązania.

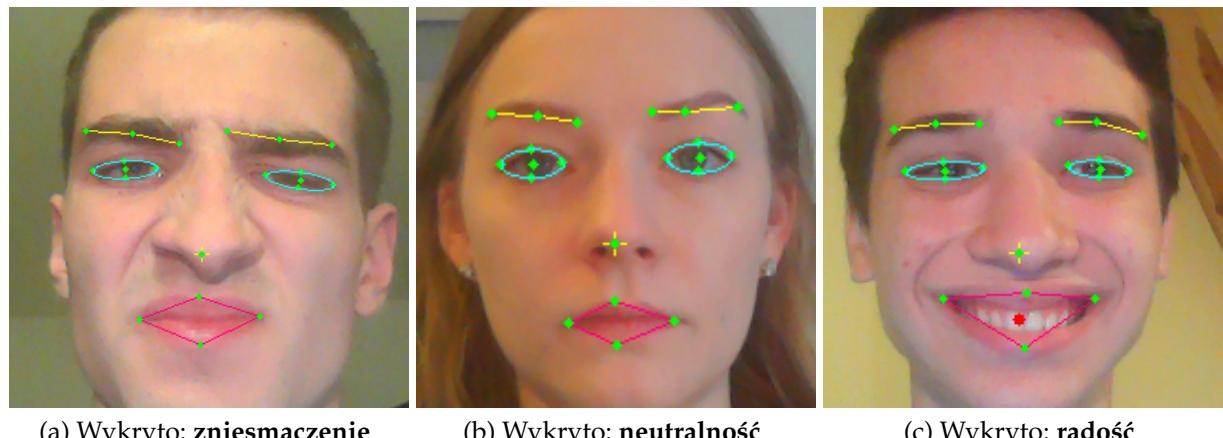
## 5.4 Ocena skuteczności

Choć uzyskane wyniki mogą napawać optymizmem, należy ostrożnie podejść do ich szerszej interpretacji. Kluczem do poprawnego zrozumienia uzyskanych rezultatów jest **kontekst badań**, z których one pochodzą. Dotyczy to głównie charakteru dostarczonych danych.

„To nie do końca tak, jak wygląda”

Byłoby naiwnym sądzeniem, że stworzony system można uznać za uniwersalne i praktycznie kompletne rozwiązanie problemu. Z pewnością nie pozwala na to skala przeprowadzonych badań. Do pierwszych dwóch eksperymentów użyto danych pochodzących od trzech osób. Ekspresje przez nie wyrażane miały być wyraźnie zarysowane tak, aby w przybliżeniu zgadzały się z ogólnymi szablonami przedstawionymi w systemie FACS. Ponadto detekcja punktów musiała być mniej więcej poprawna, a to nie zawsze ma miejsce. Jak już wspomniano w poprzednim punkcie, wynikała z tego konieczność pewnego dostosowania użytkowników do ograniczeń samego ekstraktora punktów – głównie w zakresie ułożenia głowy i unikania ekstremów pozycji niektórych punktów. Drugi rodzaj badań daje bardziej ogólną информацию na temat systemu, ale wciąż dotyczy on tylko jednej osoby, okazującej emocje w podobnych warunkach. Jakie są więc ogólne wnioski?

Można z pewnością stwierdzić, że system jest w stanie rozpoznawać emocje z wysoką skutecznością, jeśli panują dobre warunki oświetleniowe, głowa użytkownika znajduje się w odpowiedniej pozycji przed kamerą oraz emocja okazywana jest w wyraźny sposób z uwzględnieniem ograniczeń modułu wykrywania punktów twarzy. Czynniki te zwiększą prawdopodobieństwo poprawnej detekcji owych punktów, co przekłada się na jakość klasyfikacji.

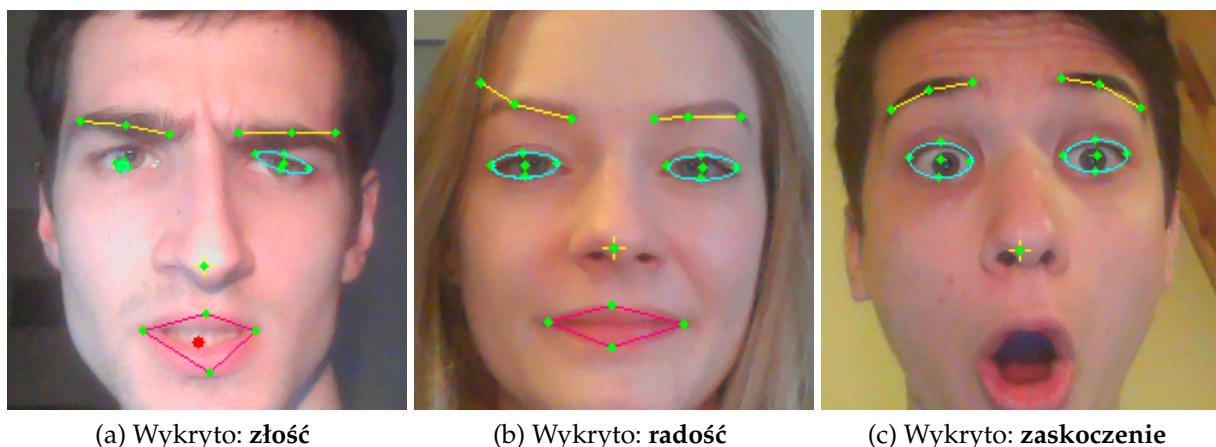


Rysunek 5.12: Przykładowe rozpoznawanie emocji w odpowiednich warunkach (w tych i kolejnych przykładach odpowiedzi pochodzą od zespołu klasyfikatorów)

Jeśli wspomniane warunki nie są spełnione, wykrywanie punktów charakterystycznych twarzy przestaje być precyzyjne, co może skutkować błędą klasyfikacji (patrz: rozdział 5.5). Efekt ten można by zredukować, biorąc część przypadków z błędą detekcją do zbioru uczącego. Niektóre mogą być bowiem niepoprawne z czysto teoretycznego punktu widzenia i jednocześnie przydatne pod względem praktycznym.

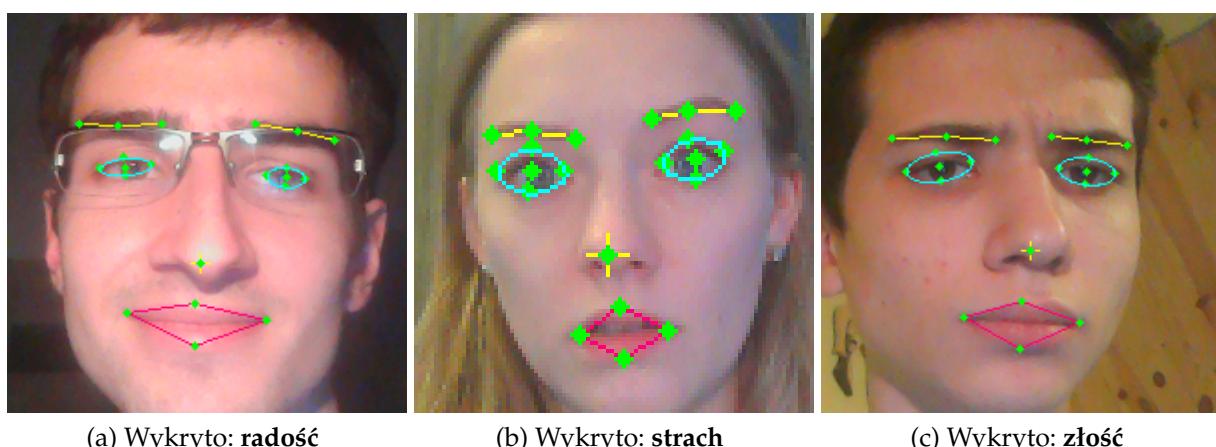
„Ale wciąż jest bardzo dobrze”

Wskazuje na to m.in. zauważona przez autora dobra generalizacja wewnętrznie wytworzzonego modelu emocii systemu. Czym ona się objawia? Klasyfikatory wykazują zaskakującą wręcz odporność i związaną z tym zdolność do rozpoznawania emocji nawet przy nieprecyzyjnej detekcji punktów. Dzieje się tak pomimo nieuwzględnienia ich w zbiorze uczącym.



Rysunek 5.13: Przykłady rozpoznawania emocji w utrudnionych warunkach

Ograniczenie zróżnicowania zbioru w dwóch pierwszych testach wynikało z założenia autora, że konieczne będzie narzucenie mocniejszych warunków przy jego kompletowaniu tak, aby umożliwić klasyfikatorom dostatecznie dobry proces uczenia. Szczególnie dla przypadku *user-independent* przewidywano znacznie słabszą generalizację. Jak się okazało, system działa lepiej niż to pierwotnie zakładano. Sam detektor detektor również wykazuje pewne nieprzewidziane wcześniej silne strony, jak np. zdolność do detekcji oczu użytkownika noszącego okulary czy utrzymywanie poprawnej ekstrakcji dla małych rotacji bocznych głowy. Działa on dosyć skutecznie także dla nieco mniejszych rozdzielczości.



Rysunek 5.14: Działanie systemu dla użytkownika z okularami i przy innym oświetleniu (po lewej), przy rozdzielczości 320 x 240 (środek), przy małej rotacji głowy w bok (po prawej)

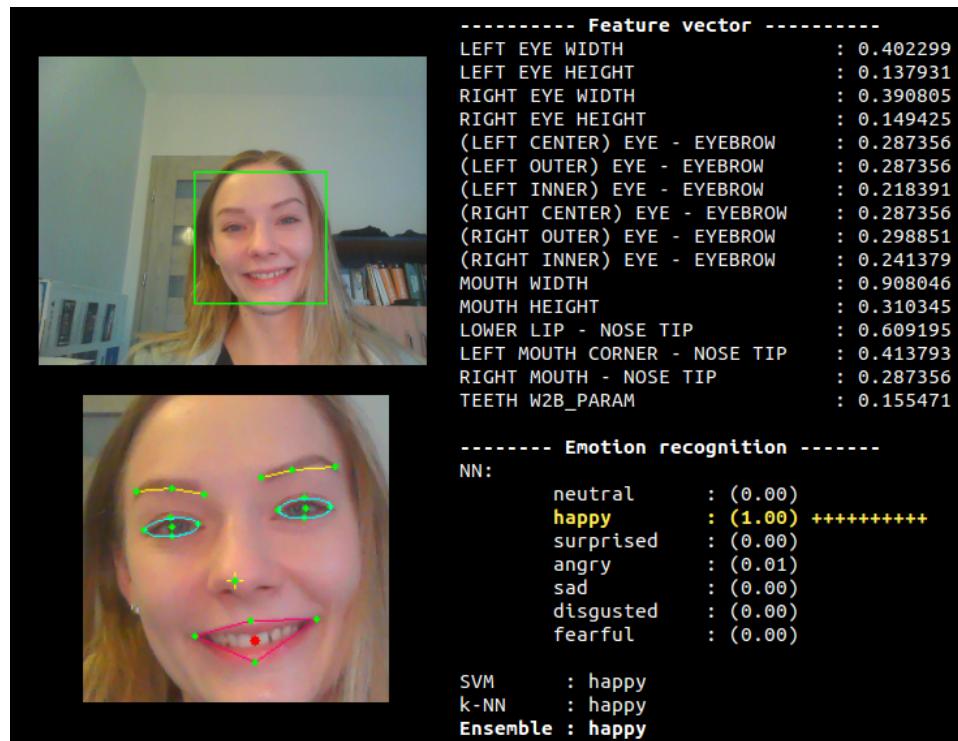
Uzyskane rezultaty i opisane spostrzeżenia otwierają drogę do treningu i testów w mniej sterylnych warunkach. **Wykonane testy to tak naprawdę pierwszy etap badań, które należałoby wykonać.** Bezpośrednie porównanie z innymi dostępnymi rozwiązaniami byłoby możliwe po przeprowadzeniu testów na zewnętrznych bazach ekspresji twarzy lub uruchomieniu tychże rozwiązań na zebranej i ewentualnie rozszerzonej przez autora bazie. Wiążałoby się to z dodatkowymi nakładami pracy i prawdopodobnie lepszym dopasowaniem pracy modułu ekstraktora. Nie podjęto się tego zadania ze względu na ograniczenia czasowe projektu. Ponieważ detekcja nie zawsze jest wystarczająco poprawna, sensownym byłoby również przeprowadzenie dodatkowych badań z wykorzystaniem sekwencji wideo z losowo wybranymi klatkami. Pojedyncze próby testowania na wspomnianych już sekwencjach wideo pokazują w każdym razie, że system ma potencjał i być może potrzebne są jedynie niewielkie poprawki i dodatkowe badania.

Patrząc na problem z innej perspektywy, nie można być też przesadnie krytycznym odnośnie aktualnych wyników. Możliwości wielu realizowanych systemów tego typu są związane z takimi samymi ograniczeniami jak przy proponowanym przez autora systemie. Dobre oświetlenie czy odpowiednia pozycja głowy to często pojawiające się warunki. Oczywiście bardziej wysublimowane rozwiązania są zorientowane na minimalizację ich wpływu, przykładem może być wspomniany Intraface. Ostatecznie nie wiadomo jednak jak poradziłyby sobie inne, potencjalnie ogólniejsze rozwiązania ze zbiorem przypadków autora. Wstępne, wykonane przez autora próby z takimi bibliotekami jak flandmark [62] czy modułem biblioteki dlib [63] pozwalają mu stwierdzić, że niekoniecznie osiągnęłyby one lepsze rezultaty.

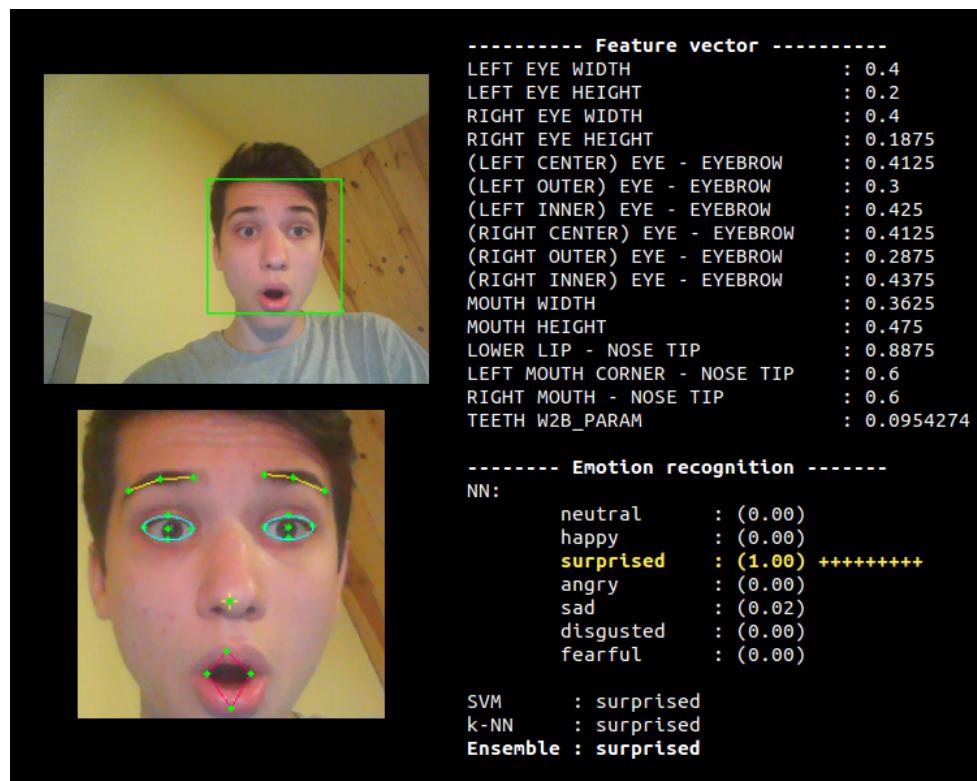
Mniej sceptycznie należy też traktować konieczność wyraźnego i czasami przerywanego, nie do końca naturalnego wyrażania emocji. Po pierwsze ekspresja twarzy nie jest jedynym źródłem wyrażania emocji. Jak już wspomniano, to około 60% informacji. Rozpoznawanie bardziej subtelnych i naturalnych emocji wyłącznie na podstawie obserwacji twarzy jest dużą trudnością nawet dla człowieka. Kluczowe znaczenie mogą mieć słowa, sposób ich wypowiadania czy postawa całego ciała (patrz: rozdział 1.2). Nie jest więc przypadkiem, że wiele zewnętrznych baz ekspresji twarzy ma podobny charakter jak zbiory autora, np. [61].

## 5.5 Przykłady

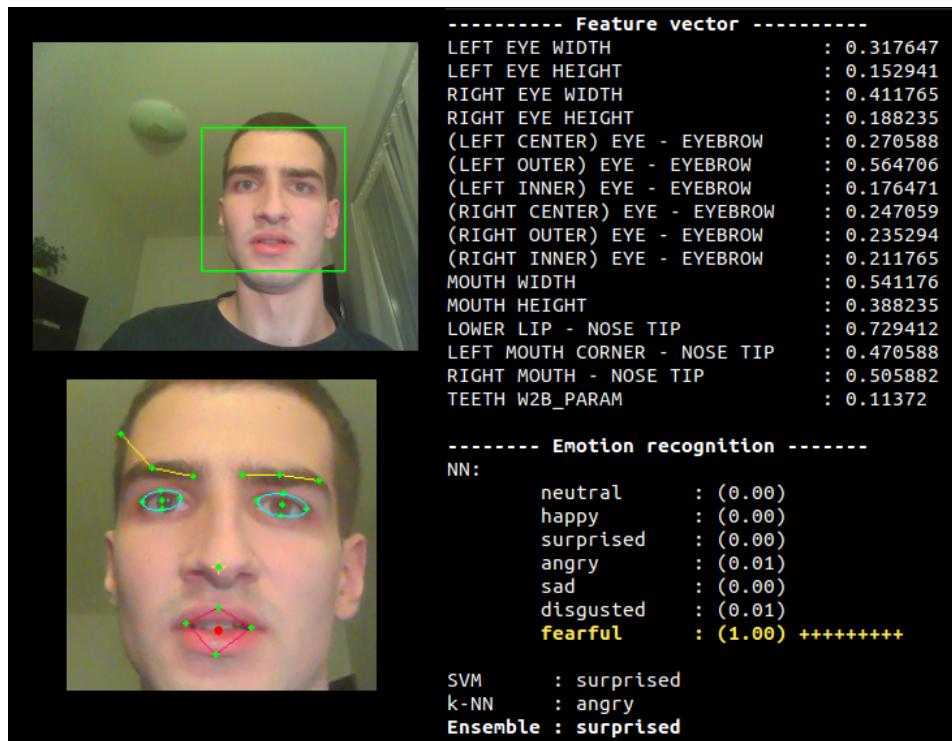
Na każdy przykład działania systemu składają się trzy części: wizualizacja wykrytych punktów twarzy, obliczony wektor cech oraz decyzje podjęte przez poszczególne klasyfikatory. Sieć neuronowa umożliwiła sensowne odwzorowanie stopnia pewności dla wybranej klasy. Wynik jej działania prezentowany jest w postaci słupków w zakresie [0-1]. Za najważniejszy wynik uznaje się wskazanie zespołu (Ensemble, na samym dole).



Rysunek 5.15: Przykład poprawnego rozpoznania radości



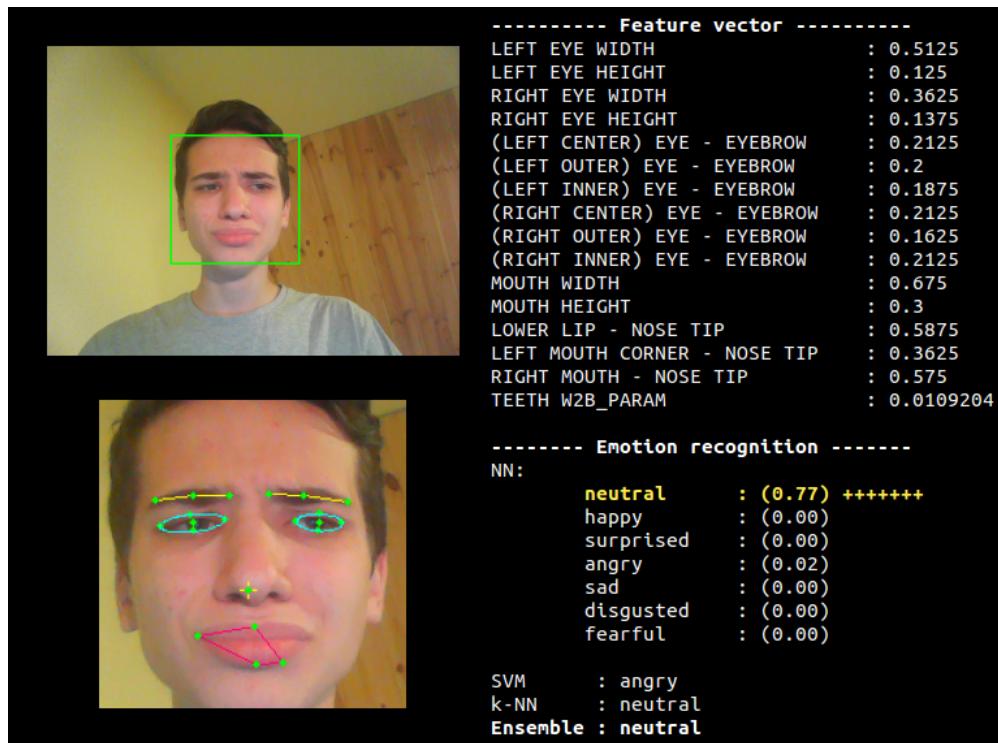
Rysunek 5.16: Przykład poprawnego rozpoznania zaskoczenia



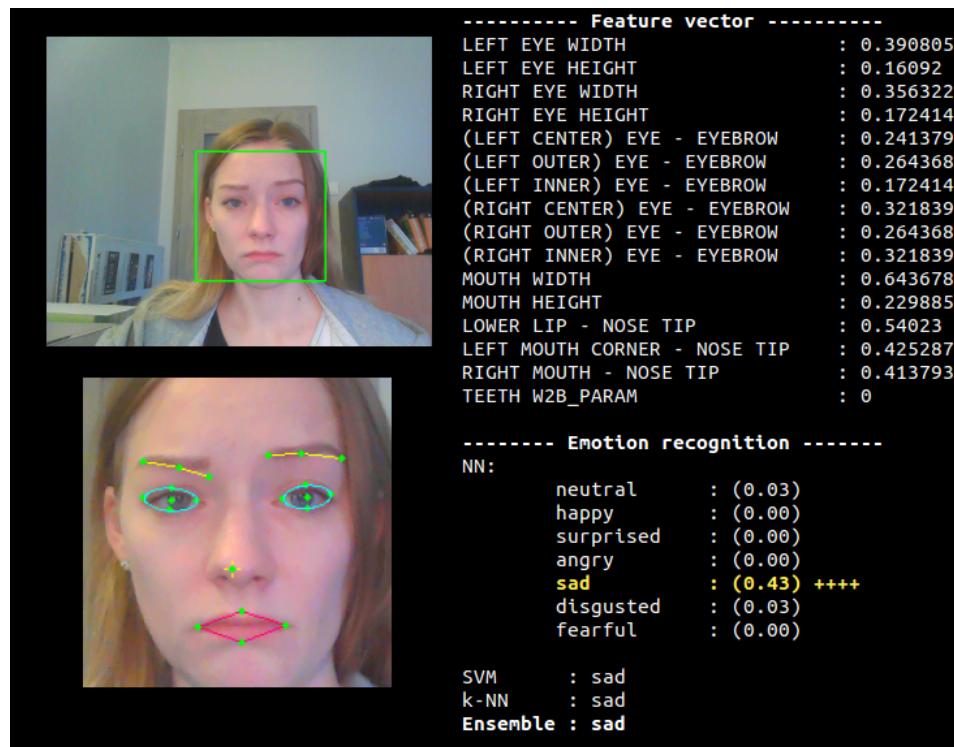
Rysunek 5.17: Przykład **błędnego** rozpoznania zaskoczenia. Lewa brew nie została poprawnie wykryta i system nie poradził sobie w tym przypadku. Sieć neuronowa poprawnie zaklasyfikowała emocię, jednak w głosowaniu każdy klasyfikator wskazał inną odpowiedź, przez co ostateczny wybór został ustalony losowo



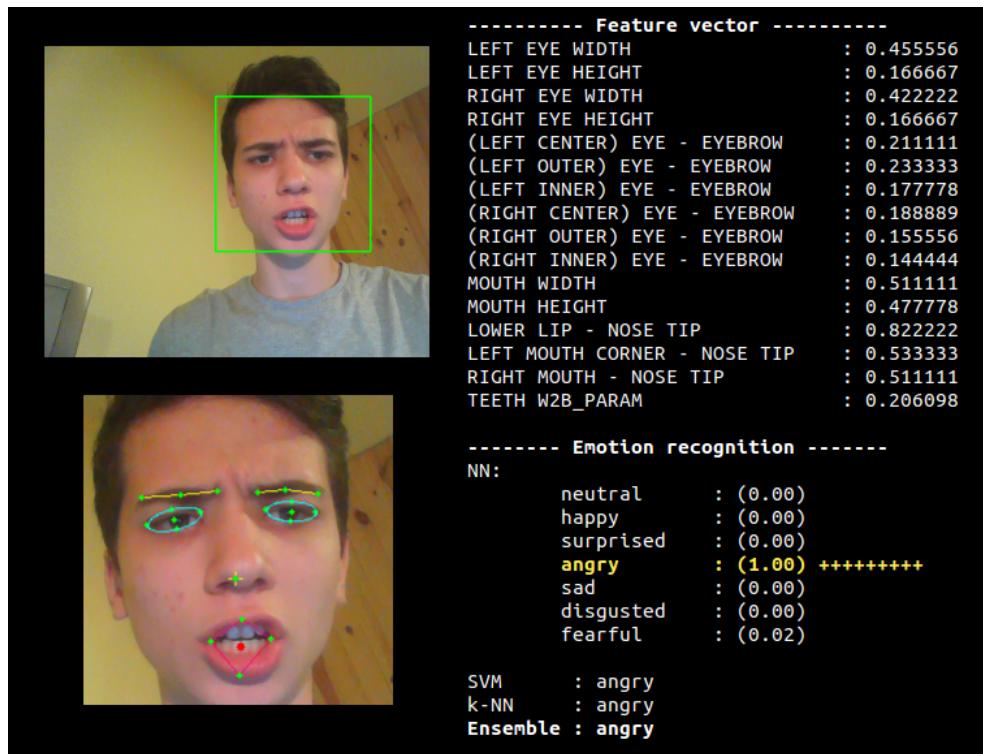
Rysunek 5.18: Przykład poprawnego rozpoznania zniesmaczenia. W tym przypadku to sieć neuronowa popełniła błąd, a głosowanie pozytywne wpłynęło na wynik klasyfikacji



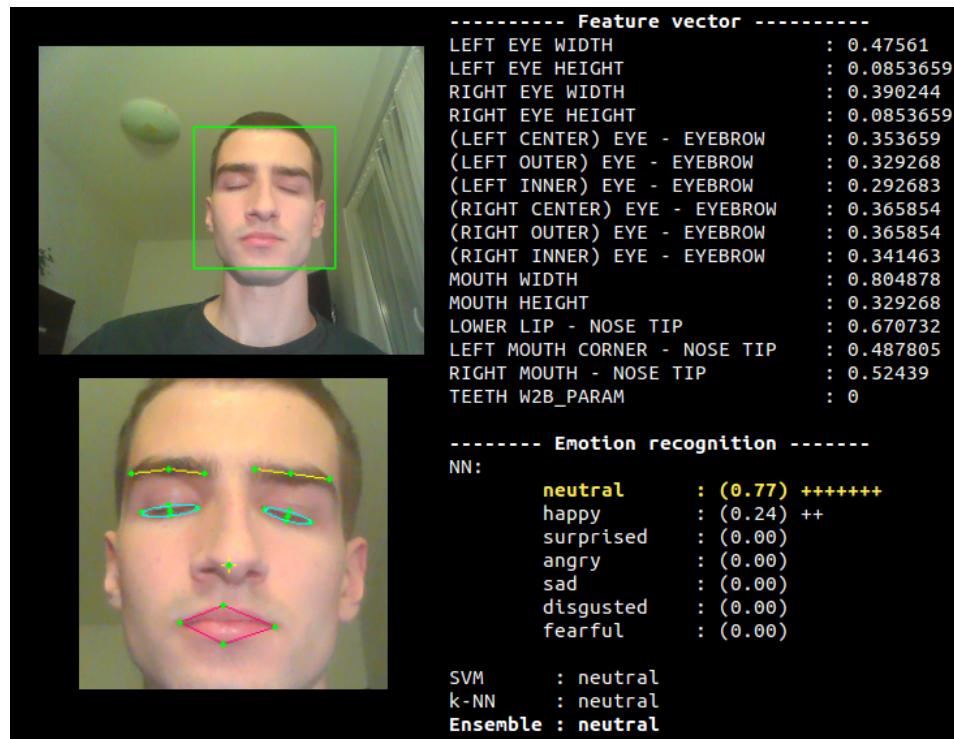
Rysunek 5.19: Przykład **błędnego** rozpoznania stanu neutralnego. Pomyłka wynikła z dwóch przyczyn: błędu detektora, z którym nie poradziły sobie klasyfikatory oraz niewystarczająco wyraźniej ekspresji smutku na poziomie oczu i brwi



Rysunek 5.20: Przykład poprawnego rozpoznania smutku pomimo nie do końca poprawnej detekcji punktów lewej brwi



Rysunek 5.21: Przykład poprawnego rozpoznania złości



Rysunek 5.22: Przykład poprawnego rozpoznania stanu neutralnego w momencie mrugnięcia

# Rozdział 6

## Podsumowanie

Niniejsze podsumowanie zamyka prezentowaną pracę inżynierską. Zamieszczone zostały w nim konkluzje dotyczące stworzonego rozwiązania oraz potencjalne drogi dalszego rozwoju.

### 6.1 Aktualny stan

#### Skuteczność

Zaproponowany system jest w stanie poprawnie i z wysoką skutecznością rozpoznawać emoce na podstawie ekspresji twarzy. Dobrze działa on zarówno przy podejściu *user-dependent* jak i *user-independent*. Najlepsze uzyskane wyniki to 97% w obu przypadkach, przy czym należy mieć na uwadze warunki narzucone na wykorzystany zbiór.

#### Warunki

Efektywność działania systemu jest zależna przede wszystkim od warunków oświetleniowych i intensywności ekspresji użytkownika. Dopuszczalne są małe odchylenia twarzy od pozycji frontalnej. Ekstraktor cech systemu posiada pewne ograniczenia i czasami może okazać się niemożliwe poprawne wykrycie np. bardzo nisko opuszczałych brwi bądź równie szeroko otwartych ust.

#### Odporność

Ze względu na dobrą generalizację klasyfikatorów w trakcie uczenia, wspomniane ograniczenia nie są dotkliwie krępujące. System jest w stanie poprawnie rozpoznawać emocje w wielu przypadkach z nieprecyzyjną ekstrakcją punktów charakterystycznych twarzy.

#### Intensywność ekspresji

Rezultaty uzyskane przy badaniu sekwencji wideo oraz pojedyncze przykłady w trudniejszych warunkach rozpoznawania pozwalają sądzić, że system poradziłby sobie również z przypadkami bardziej spontanicznymi. Niewątpliwie jednak jego wrażliwość jest w pewnym stopniu ograniczona. Cechuje go stabilność dla wyraźnych eks-

presji i często występujące wahania przy mniej intensywnych emocjach. Wynika to zarówno z niedoskonałości technicznych jak i natury samego problemu.

### Adaptowalność

Przeprowadzone badania umożliwiają w pewnym stopniu stwierdzenie, że system jest w stanie adaptować się do różnych użytkowników i w mniejszym stopniu do małych zmian charakteru oświetlenia. Konieczne jest przeprowadzenie dodatkowych testów na rozszerzonym zbiorze i/lub zewnętrznych bazach spełniających wymogi systemu. Mogłoby się to wiązać z koniecznością lepszego dopasowania modułu ekstraktora do większej grupy użytkowników.

### Szybkość

System działa w czasie rzeczywistym z szybkością **8 FPS** dla użytego laptopa przeciętnej klasy.

## 6.2 Perspektywy rozwoju

Za podstawowe zadanie do wykonania należy uznać przetestowanie systemu na zewnętrznych bazach i dopracowanie ekstraktora lub lepsze skalibrowanie jego parametrów tak, aby rozszerzyć spektrum możliwych do analizowania przypadków. Konieczne jest przede wszystkim zmniejszenie uzależnienia systemu od oświetlenia i osłabienie ograniczeń na swobodę wyrażania emocji, co jest równoważne swobodzie w ruchach twarzy.

Zwiększenie wysokości obszaru twarzy zwracanego przez detektor mogłoby umożliwić szersze otwieranie ust przy wyrażaniu zaskoczenia. Poszczególne metody detekcji punktów twarzy można uzupełnić o zależności geometryczne (elipsa ust) i dokładniejsze heurystyki (adaptacyjna rozbudowa ciemnego obszaru brwi nad okiem).

Znaczące ograniczenie wpływu samego oświetlenia wymagałoby rozważenia zupełnie odmiennego podejścia do ekstrakcji punktów twarzy. Możliwe jest np. wykorzystanie analizy twarzy w świetle podczerwonym. Wiążałoby się z utratą informacji o kanałach RGB i koniecznością pracy z obrazem wejściowym w skali szarości. Autor sugerowałby zastosowanie aktywnych modeli kształtów (ASM), które – zgodnie z różnymi opublikowanymi badaniami – dobrze radzą sobie z tego typu zadaniami.

Umożliwienie bardziej spontanicznych zachowań użytkowników wiążałoby się nie tylko z udoskonaleniem samego ekstraktora, ale również i ulepszeniem procesu wytwarzania klasyfikatorów. Nawet dla aktualnego stanu ich wydajność można jeszcze zwiększyć, wykorzystując metody selekcji i redukcji dla danych ze zbioru uczącego. Ponadto sprawdzony został zaledwie ułamek możliwości kombinowania algorytmów ze sobą w ramach zespołów oraz ustalenie ich parametrów. Istnieje szansa, że dokładniejsza analiza możliwych rozwiązań w połączeniu ze wspomnianą selekcją i udoskonaleniem ekstraktora wpłynęłyby na polepszenie efektywności działania systemu w zakresie wykrywania spontanicznych lub mniej intensywnych emocji. Dobrym sposobem na sprawdzenie tego, byłyby testy na zaproponowanych już sekwencjach wideo.

# Bibliografia

- [1] Card, Stuart K., Thomas P. Moran, Allen Newell, *The keystroke-level model for user performance time with interactive systems*, [w:] Communications of the ACM, Vol. 23, No. 7, 1980, s. 396–410.
- [2] D. Te'eni, J. Carey and P. Zhang, *Human Computer Interaction: Developing Effective Organizational Information Systems*, John Wiley & Sons, Hoboken, 2007.
- [3] F. Karay, M. Alemzadeh, J. A. Saleh and M. Nours *Human-Computer Interaction: Overview on State of the Art*, [w:] The International Journal on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, 2008.
- [4] G. Fanelli, J. Gall, L. Van Gool, *Hough Transform-based Mouth Localization for Audio-Visual Speech Recognition*, 2009.
- [5] A. Jaimes, N. Sebe, *Multimodal human computer interaction: a survey*, Computer Vision and Image Understanding, Vol. 108, No. 1-2, 2007, s. 116-134.
- [6] Strona główna Affectiva SDK: [www.affectiva.com](http://www.affectiva.com), ostatni dostęp: 7.12.2015.
- [7] Strona główna EmoVu SDK: [www.emovu.com](http://www.emovu.com), ostatni dostęp: 7.12.2015.
- [8] W. D. Ross, *Aristotelis Ars Rhetorica*, Clarendon Press, Oxford, 1959.
- [9] Ch. Darwin, P. Eckman, P. Prodger, *The Expression of Emotion in Man and Animals*, 3. edycja, Harper Collins, Londyn, 1998.
- [10] R. Plutchik, *Emotion: A Psychoevolutionary Synthesis*, Harper and Row, Nowy Jork, 1980.
- [11] P. Ekman, W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Consulting Psychologists Press, Palo Alto, 1978.
- [12] Termin „inteligencja emocjonalna”: [https://en.wikipedia.org/wiki/Emotional\\_intelligence](https://en.wikipedia.org/wiki/Emotional_intelligence), ostatni dostęp: 7.12.2015.
- [13] Y. Liu, O. Sourina, M. K. Nguyen *Real-time EEG-based Emotion Recognition and its Applications*, Nanyang Technological University, Singapur, 2011.
- [14] A. Mehrabian, *Silent Messages*, Wadsworth Publishing Company, Belmont, 1971.
- [15] Definicja terminu „afekt”: <https://pl.wikipedia.org/wiki/Afekt>, ostatni dostęp: 7.12.2015.

- [16] C. C. Chibelushi, F. Bourel, *Facial Expression Recognition: A Brief Tutorial Overview*, 2008.
- [17] A. Azcarate, F. Hageloh, K.van de Sande, R. Valenti, *Automatic facial emotion recognition*, Universiteit van Amsterdam, 2005.
- [18] H. F. Huang and S.Ch. Tai, *Facial Expression Recognition Using New Feature Extraction Algorithm*, National Cheng Kung University, 2011.
- [19] M. S. Islam, S. Auwatanamongkol *A Novel Feature Extraction Method for Facial Expression Recognition*, School of Applied Statistics, Bangkok, 2013.
- [20] K.Ch. Huang, S. Y. Huang, Y.H. Kuo, *Emotion recognition based on a novel triangular facial feature extraction method*, National Cheng Kung University, 2010.
- [21] H. Tao, T.S. Huang. *Connected vibrations: a modal analysis approach to non-rigid motion tracking*, CVPR (IEEE), 1998, s. 735-740.
- [22] Strona główna Insight SDK: [www.sightcorp.com](http://www.sightcorp.com), ostatni dostęp: 7.12.2015.
- [23] X. Xiong, F. De la Torre, *Supervised Descent Method and its Applications to Face Alignment*, Carnegie Mellon University, 2013.
- [24] W. S. Chuy, F. De la Torrey, J. F. Cohnyz *Selective Transfer Machine for Personalized Facial Action Unit Detection*, Carnegie Mellon University, 2013.
- [25] Strona główna IntraFace: [www.humansensing.cs.cmu.edu/intraface/](http://www.humansensing.cs.cmu.edu/intraface/), ostatni dostęp: 7.12.2015.
- [26] T. Cootes, *An Introduction to Active Shape Models*.
- [27] Strona główna STASM: [www.milbo.users.sonic.net/stasm/](http://www.milbo.users.sonic.net/stasm/), ostatni dostęp: 7.12.2015.
- [28] S. Milborrow, *Locating Facial Features with Active Shape Models*, University of Cape Town, 2007.
- [29] Strona główna Emotient: [www.emotient.com](http://www.emotient.com), ostatni dostęp: 7.12.2015.
- [30] Strona główna FaceReader: [www.facereader-online.com](http://www.facereader-online.com), ostatni dostęp: 7.12.2015.
- [31] Strona główna nVisio: [www.nviso.ch](http://www.nviso.ch), ostatni dostęp: 7.12.2015.
- [32] P. A. Kowalski, *Procedura ekstrakcji cech z obrazu twarzy dla potrzeb systemu biometrycznego*, w: Czasopismo Techniczne. Automatyka, Wydawnictwo Politechniki Krakowskiej im. Tadeusza Kościuszki, Kraków, 2012.
- [33] V. Bevilacqua, A. Cicciarri, I. Leone, G. Mastronardi *Automatic Facial Feature Points Detection*, Bari University of Technology, 2008.
- [34] Biblioteka OpenCV 3.0: <http://opencv.org/opencv-3-0.html>, ostatni dostęp: 7.12.2015.
- [35] P. Viola, M. J. Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features*, CVPR (IEEE), 2001, s. 511-518.
- [36] C. Papageorgiou, M. Oren, and T. Poggio. *A general framework for object detection*, CVPR (IEEE), 1998, s. 555-562.

- [37] F. C. Crow, *Summed-area tables for texture mapping*, ACM SIGGRAPH Computer Graphics, Vol. 18, 1984, s. 207-212.
- [38] Analiza ilości możliwych cech Haara w oknie o wymiarach 24x24, <http://stackoverflow.com/questions/1707620/viola-jones-face-detection-claims-180k-features>, ostatni dostęp: 7.12.2015.
- [39] J. B. Gomez-Mendoza, *A contribution to mouth structure segmentation in images aimed towards automatic mouth gesture recognition*, Universidad Nacional de Colombia, Lyon, 2012.
- [40] L. G. Farkas and I. R. Munro, *Anthropometric Facial Proportions in Medicine*, Springfield, Illinois, 1987.
- [41] J. L. Moreira, A. Braun, S. R. Musse, *Eyes and Eyebrows Detection for Performance Driven Animation*, Pontificia Universidade Católica do Rio Grande do Sul, 2010.
- [42] M. Betke , W. J. Mullally , J. J. Magee, *Active Detection of Eye Scleras in Real Time*, CVPR (IEEE), 2000.
- [43] V. Vezhnevets, V. Sazonov, A. Andreeva, *A Survey on Pixel-Based Skin Color Detection Techniques*, Faculty of Computational Mathematics and Cybernetics, Moskwa, 2003.
- [44] Różne operacje transformujące w OpenCV:  
[http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous\\_transformations.html](http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html), ostatni dostęp: 7.12.2015.
- [45] Operacja dylacji i erozji w OpenCV:  
<http://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html>, ostatni dostęp: 7.12.2015.
- [46] Analiza strukturalna i deskryptory kształtu w OpenCV:  
[http://docs.opencv.org/2.4/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html](http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html), ostatni dostęp: 7.12.2015.
- [47] S. Baskan, M. M. Bulut, V. Atalay, *Projection based method for segmentation of human face and its evaluation*, Middle East Technical University, Ankara 2001.
- [48] L. Florea and R. Boia, *Eyebrows Localization for Expression Analysis*, ICCP (IEEE), 2011, s. 281-284.
- [49] Wyrównywanie histogramu, [http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram\\_equalization/histogram\\_equalization.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html), ostatni dostęp: 7.12.2015.
- [50] Metody filtracji w OpenCV: <http://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html>, ostatni dostęp: 7.12.2015.
- [51] M. Pantic, M. Tomc and L. J.M. Rothkrantz, *A Hybrid Approach to Mouth Features Detection*, Systems, Man and Cybernetics (IEEE), vol.2, 2001, s. 1188-1193.
- [52] J. Pan, Y. Guan, S. Wang, *A New Color Transformation Based Fast Outer Lip Contour Extraction*, [w:] Journal of Information & Computational Science, Vol.9, No.9, 2012, s. 2505–2514.

- [53] M. Kurzyński, *Klasyfikacja – wykład*, Politechnika Wrocławskiego, 2015.
- [54] P. Michel, *Support Vector Machines in Automated Emotion Classification*, Churchill College, 2003.
- [55] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, S. Narayanan, *Analysis of Emotion Recognition using Facial Expressions, Speech and Multimodal Information*, University of Southern California, 2004.
- [56] C. C. Lee, E. Mower, C. Busso, S. Lee, S. Narayanan, *Emotion Recognition Using a Hierarchical Binary Decision Tree Approach*, University of Southern California, 2009.
- [57] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, Massachusetts, 2005.
- [58] M. Law, *A Simple Introduction to Support Vector Machines*, wykład dla CSE 802, Michigan State University.
- [59] Biblioteka sieci neuronowej FANN: <http://leenissen.dk/fann/wp/>, ostatni dostęp: 7.12.2015.
- [60] Algorytmy uczenia maszynowego w OpenCV: <http://leenissen.dk/fann/wp/>, ostatni dostęp: 7.12.2015.
- [61] Strona bazy MMI: <http://mmifacedb.eu/>, ostatni dostęp: 7.12.2015.
- [62] Strona biblioteki flandmark: <http://cmp.felk.cvut.cz/~uricamic/flandmark/>, ostatni dostęp: 7.12.2015.
- [63] Strona modułu detekcji punktów w bibliotece dlib: <http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>, ostatni dostęp: 7.12.2015.

# Spis rysunków

1.1	Przykłady zaawansowanej interakcji człowiek-komputer . . . . .	4
1.2	Koło Plutchika . . . . .	5
1.3	Oznaczenia niektórych mięśni twarzy i akcje mimiczne . . . . .	6
1.4	Reguła 55-38-7 Mehrabiana dotycząca procentowego udziału poszczególnych źródeł informacji w komunikacji emocjonalnej . . . . .	7
1.5	Podstawowy schemat przetwarzania ekspresji twarzy . . . . .	8
1.6	Śledzenie ruchów twarzy przy pomocy metody PBVD . . . . .	9
1.7	Idea działania proponowanej metody STM personalizującej generyczny klasyfikator SVM i działanie aplikacji . . . . .	11
1.8	Przykładowe dopasowanie modelu ASM przy pomocy biblioteki STASM	12
2.1	Przykład poprawnego i niepoprawnego obrazu wejściowego . . . . .	16
2.2	Przykładowe macierze bazowe oraz wizualizacja idei obrazów integralnych . . . . .	17
2.3	Dwie pierwsze cechy wybrane przez algorytm AdaBoost w trakcie treningu . . . . .	18
2.4	Przykłady działania detektora dla różnych parametrów . . . . .	19
2.5	Przykłady działania detektora dla ustalonych parametrów i z wyborem największego proponowanego obiektu twarzy . . . . .	20
2.6	Liczba przetwarzanych klatek na sekundę (FPS) w zależności od rozdzielczości obrazu . . . . .	20
3.1	Ogólny schemat ekstrakcji cech z obrazu twarzy . . . . .	21
3.2	Przykładowe obszary wyznaczone przy pomocy detektorów . . . . .	23
3.3	Ustalone obszary ROI na wykrytej twarzy . . . . .	24
3.4	Liczba przetwarzanych klatek na sekundę (FPS) w zależności od rozdzielczości obrazu i zastosowanej metody ustalania ROI . . . . .	25
3.5	Przykłady nałożenia statycznych obszarów dla różnych osób i przypadków . . . . .	25
3.6	Obraz wejściowy, obraz po wykonaniu operacji mapowania, obraz mapy ze zwiększym kontrastem dla lepszej wizualizacji . . . . .	27

3.7	Obraz po binaryzacji, obraz po wykonaniu operacji czyszczenia krawędzi, obraz mapy po operacji dylacji . . . . .	28
3.8	Przykłady detekcji punktów oka dla różnych stanów i osób . . . . .	28
3.9	Kolejne etapy wstępniego przetwarzania brwi . . . . .	29
3.10	Przykład działania funkcji liniowo czyszczącej krawędzie okna brwi, obraz po binaryzacji, obraz oczyszczony przy pomocy filtru medianowego . . . . .	30
3.11	Przykłady detekcji punktów brwi dla różnych stanów i osób . . . . .	31
3.12	Obraz wejściowy ust, obraz po mapowaniu omówioną transformacją, obraz po wzmacnieniu kontrastu i oczyszczeniu krawędzi okna . . . . .	32
3.13	Obraz ust po binaryzacji, obraz po operacji dylacji, dopasowana do segmentu elipsa i para punktów wertykalnych ust . . . . .	32
3.14	Detekcja prawego kącika ust . . . . .	33
3.15	Przykłady detekcji punktów ust dla różnych stanów i osób . . . . .	33
3.16	Obraz zębów po mapowaniu ust, zanegowany obraz po czyszczeniu krawędzi, obraz po binaryzacji . . . . .	34
3.17	Przykłady detekcji zębów dla różnych stanów i osób. . . . .	34
3.18	Przykłady detekcji nosa dla różnych stanów i osób . . . . .	34
3.19	Przykłady pełnej detekcji punktów twarzy dla różnych stanów i osób .	35
3.20	Wykryte punkty na etapie ekstrakcji . . . . .	35
4.1	Ogólny schemat zadania klasyfikacji . . . . .	37
4.2	Model pojedynczego neuronu . . . . .	39
4.3	Wizualizacja sposobu działania algorytmu propagacji wstecznej dla fragmentu sieci . . . . .	41
4.4	Prosty przykład wyznaczonej hiperpłaszczyyny $H$ , separującej dwie klasy w przestrzeni 2-wymiarowej . . . . .	42
4.5	Idea rozwiązywania problemów nieliniowych w algorytmie SVM. Źródło: <a href="http://www.reddit.com">www.reddit.com</a> . . . . .	44
5.1	Przykładowe obrazy treningowe zabrane przez użytkowników, kolejno: U1, U2, U3 . . . . .	48
5.2	Średnie wyniki testów <i>user-dependent</i> dla poszczególnych klasyfikatorów	50
5.3	Średnie wyniki testów <i>user-dependent</i> dla poszczególnych emocji . . . .	50
5.4	Średnie wyniki testów <i>user-independent</i> dla poszczególnych klasyfikatorów . . . . .	51
5.5	Średnie wyniki testów <i>user-independent</i> dla poszczególnych emocji . . . .	51
5.6	Poziom wsparcia dla każdej emocji w sekwencji z ekspresją <b>radości</b> . .	53
5.7	Poziom wsparcia dla każdej emocji w sekwencji z ekspresją <b>zaskoczenia</b>	53

---

5.8 Poziom wsparcia dla każdej emocji w sekwencji z ekspresją <b>złości</b> . . . . .	54
5.9 Poziom wsparcia dla każdej emocji w sekwencji z ekspresją <b>smutku</b> . . . . .	54
5.10 Poziom wsparcia dla każdej emocji w sekwencji z ekspresją <b>zniesmaczenia</b> . . . . .	55
5.11 Poziom wsparcia dla każdej emocji w sekwencji z ekspresją <b>strachu</b> . . . . .	55
5.12 Przykładowe rozpoznawanie emocji w odpowiednich warunkach . . . . .	57
5.13 Przykłady rozpoznawania emocji w utrudnionych warunkach . . . . .	58
5.14 Działanie systemu dla użytkownika z okularami i przy innym oświetleniu, przy rozdzielczości 320 x 240, przy małej rotacji głowy w bok . . . . .	58
5.15 Przykład poprawnego rozpoznania radości . . . . .	60
5.16 Przykład poprawnego rozpoznania zaskoczenia . . . . .	60
5.17 Przykład błędного rozpoznania zaskoczenia . . . . .	61
5.18 Przykład poprawnego rozpoznania zniesmaczenia . . . . .	61
5.19 Przykład błędного rozpoznania stanu neutralnego . . . . .	62
5.20 Przykład poprawnego rozpoznania smutku . . . . .	62
5.21 Przykład poprawnego rozpoznania złości . . . . .	63
5.22 Przykład poprawnego rozpoznania stanu neutralnego w momencie mrugnięcia . . . . .	63

# Spis tabel

3.1	Lewe górne narożniki obszarów ROI oraz ich wysokości i szerokości . . . . .	24
3.2	Wartości parametrów $m$ i $w$ dla poszczególnych krawędzi okna brwi. . . . .	30
5.1	Uzyskane średnie wyniki testów <i>user-dependent</i> dla różnych algorytmów i emocji . . . . .	50
5.2	Uzyskane średnie wyniki testów <i>user-independent</i> dla różnych algorytmów i emocji . . . . .	51