

POLITECHNIKA ŚLĄSKA W GLIWICACH  
WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ

Projekt

**Biometria**

**Analiza porównawcza skuteczności klasyfikacji  
emocji na podstawie nagrań audio za pomocą  
dwóch wybranych metod**

**Karolina Cicha, Adrianna Jagodzińska, Agata Momot, Julia Uzdowska**

Gliwice, 19 czerwca 2021

# Spis treści

|  |    |
|--|----|
| 1. Wstęp teoretyczny . . . . .                 | 1  |
| 1.1 Emocje . . . . .                           | 1  |
| 1.1.1 Czym są emocje? . . . . .                | 1  |
| 1.1.2 Emocje podstawowe Paula Ekmana . . . . . | 2  |
| 1.1.3 Teoria Emocji Plutchika . . . . .        | 4  |
| 1.1.4 Inne teorie emocji . . . . .             | 5  |
| 1.2 Analiza emocji . . . . .                   | 6  |
| 1.2.1 Cele analizy emocji . . . . .            | 6  |
| 1.2.2 Sposoby analizy emocji . . . . .         | 6  |
| 1.3 Baza danych <i>ravdness</i> . . . . .      | 8  |
| 1.4 Sieć neuronowa - <i>Bilstm</i> . . . . .   | 9  |
| 1.5 Maszyna wektorów nośnych (SVM) . . . . .   | 9  |
| 2. Sieć neuronowa . . . . .                    | 11 |
| 2.1 Implementacja . . . . .                    | 11 |
| 2.1.1 Rozszerzenie danych . . . . .            | 12 |
| 2.2 Ekstrakcja cech . . . . .                  | 13 |
| 2.2.1 Parametry sieci . . . . .                | 14 |
| 2.2.2 Ocena i testowanie . . . . .             | 16 |
| 2.3 Wyniki . . . . .                           | 17 |
| 3. SVM . . . . .                               | 18 |
| 3.1 Implementacja . . . . .                    | 18 |
| 3.2 Otrzymane wyniki . . . . .                 | 23 |
| 4. Wnioski . . . . .                           | 26 |
| 5. Podsumowanie . . . . .                      | 27 |
| Bibliografia . . . . .                         | 28 |

# 1. Wstęp teoretyczny

## 1.1 Emocje

### 1.1.1 Czym są emocje?

Emocje były koncepcją znaną już antycznym filozofom, rozważającym to co dziś nazywamy emocjami pod wieloma różnymi nazwami. Samo słowo emocja pochodzi z języka francuskiego, jednak jego pierwotne znaczenie związane było ze społecznym poruszeniem, nie natomiast stanem mentalnym. Słowo to w dzisiejszym znaczeniu weszło do użytku w XIX wieku. Przedmiotem zainteresowania badaczy pierwotnie był problem tego co jest potrzebne by dany stan mógł być uznany za emocje. Wśród komponentów związanych z reakcją emocjonalną człowieka (ewaluacyjnych, fizjologicznych, behawioralnych, fenomenologicznych oraz ekspresywnych) starano się wyizolować takie, które są niezbędne dla istnienia emocji. Rozważania te przyczyniły się do powstania szeregu teorii emocji, które można podzielić na trzy kategorie: uczuć, ewaluacji oraz motywacji [1].

- Teorie uczuć uznają to w jaki sposób odczuwane są emocje jako ich najważniejszą cechę. Emocje są definiowane jako oddzielne, świadome doświadczenia [1].
- Teorie ewaluacji jako najistotniejszy aspekt emocji postrzegają, to jak oddziałują one ze światem zewnętrznym. W tym przypadku emocje określa się jako ewaluacje istniejących okoliczności [1].
- Teorie motywacji postrzegają emocje jako oddzielne stany motywacji [1].

Większość teorii emocji, niezależnie od ich kategorii stara się odpowiedzieć na pytania [1]:

- Czym emocje różnią się między sobą i od tego co emocjami nie jest?
- Czy emocje motywują zachowanie i jeśli tak w jaki sposób?
- Czy emocje są zorientowane obiektowo i jeśli tak czy mogą być odpowiednie lub nieodpowiednie dla swojego obiektu?
- Czy emocje zawsze zawierają element subiektywnego doświadczenia i jeśli tak jakiego rodzaju doświadczenia?

### 1.1.2 Emocje podstawowe Paula Ekmana

Jedną z pierwszych teorii starających się opisać emocje człowieka została stworzona przez amerykańskiego psychologa Paula Ekmana w latach 70. XX wieku. Przedstawił on postulat, że istnieje sześć emocji, które są odczuwane uniwersalnie, bez względu na kulturę czy miejsce pochodzenia. Do listy tych emocji zaliczył szczęście, smutek, strach, wstręt, złość oraz zaskoczenie. Emocje te w sposób graficzny zostały przedstawione na rysunku 1.1. Teoria Ekmana w dalszych latach rozwijała się, a jego lista była rozszerzana o kolejne emocje [2]. Oprócz teorii emocji, Ekman był również pionierem w zakresie analizy mikroekspresji do rozpoznawania emocji oraz wykrywania kłamstw (jego badania były inspiracją dla stworzenia serialu *Magia Kłamstwa*) [3].



Rys. 1.1: Sześć emocji podstawowych wg. Paula Ekmana [4]

#### Szczęście

Szczęście jest emocją, której poświęca się wiele uwagi zarówno jako codzienne dążenie człowieka do osobistego szczęścia, jak również w ujęciu naukowym. Powstała nawet oddzielna dziedzina psychologii – psychologia pozytywna – zajmująca się poprawą kondycji zdrowotnej oraz społecznej poprzez analizę indywidualnej drogi do szczęścia. Samo szczęście jest definiowane jako przyjemny stan emocjonalny, charakteryzujący się odczuciem zadowolenia, radości, zaspokojenia, satysfakcji oraz dobrego samopoczucia. Do zewnętrznych wyrażeń tej emocji można zaliczyć wyraz twarzy (uśmiech), zrelaksowany język ciała oraz radosny ton głosu [2].

Mimo, że szczęście jako emocja jest bardzo zindywidualizowane, ludzkie postrzeganie tego, co je wywołuje jest w dużej mierze podatne na różne wpływy, takie jak otoczenie czy media. Z tego powodu wielokrotnie dąży się do szczęścia na sposoby, które mogą nie być odpowiednie dla danej osoby. Odczuwanie szczęścia nie jest jedynie związane z chwilową przyjemnością, ale wywiera ono także długoterminowe skutki na zdrowiu [2].

### *Smutek*

Smutek jest chwilowym stanem emocjonalnym, w którym odczuwane są zawód, żaloba, rozpacz, brak zainteresowania czy przytłumienie. Do zewnętrznych wyrazów smutku należą płacz, przygaszenie, ospałość, wyciszenie oraz izolowanie. Odczuwanie smutku jest naturalnym stanem człowieka, jednak jego długie i intensywne okresy mogą prowadzić lub być oznaką głębszych problemów psychologicznych, takich jak depresja [2].

### *Strach*

Strach jako emocja odpowiada za wyzwalanie instynktu walki lub ucieczki – podniesiony poziom adrenaliny pozwala na chwilową poprawę wydajności poprzez podniesienie tętna, przyspieszenia oddychania, poprawy szybkości myślenia oraz zwiększenia napięcia mięśni. Jest to przystosowanie ewolucyjne wspomagające przetrwanie w niebezpiecznych sytuacjach. Jednak strach może być również wyzwalany poza sytuacjami zagrożenia życia. Wraz z towarzyszącymi reakcjami może pojawiać się jako odpowiedź na oczekiwany lub nawet wyimaginowany atak. Może powodować to powstawanie schorzeń lękowych, jednak w niektórych przypadkach strach może wywoływać odczucie przyjemności, co jest przyczyną szukania okazji do pocucia zagrożenia np. poprzez uprawianie sportów ekstremalnych. Do zewnętrznych oznak strachu można zaliczyć wyraz twarzy taki jak rozszerzanie oczu, język ciała wskazujący na chęć ukrycia się lub ucieczki oraz reakcje fizjologiczne wymienione wcześniej w odniesieniu do reakcji walki lub ucieczki [2].

### *Obrzydzenie*

Odczucie odrazy może mieć wiele powodów zarówno wywołanych bodźcami zmysłowymi takimi jak zapach czy smak jak i umysłowymi, gdy uczucie odrzucenia powstaje na skutek moralnej reakcji na nieodpowiednie zachowanie. Wiodąca teoria na temat powstania odrazy głosi, że emocja ta została wykształcona jako zabezpieczenie przed szkodliwym bądź trującym pokarmem. Odraza może być wyrażana poprzez język ciała (odwracania się od źródła wzbudzenia emocji), reakcji fizycznych, takich jak mdłości lub wyrazów twarzy, np. marszczenia nosa [2].

### *Gniew*

Gniew jest kolejną emocją, która wywiera wpływ na instynkty zachowawcze człowieka. Odpowiedź na zagrożenie ma w tym przypadku postać walki. Emocja ta charakteryzuje się odczuciami wrogości, agresji i pobudzenia. Gniew wyrażany jest poprzez reakcje takie jak marszczenie brwi, krzyk lub szorstką mowę, czerwienienie, pocenie lub przemoc. Do pewnego momentu gniew może być przydatną emocją, pozwalającą między innymi na zwiększenie motywacji do działania, jednak niekontrolowany nadmierny lub wyrażany w nieakceptowalny sposób gniew wywiera negatywny wpływ na

otoczenie, a także zdrowie danej osoby, jest na przykład powiązany z chorobami układu krążenia [2].

### *Zaskoczenie*

Zaskoczenie jest krótkotrwałą emocją, będącą odpowiedzią na nieoczekiwane zdarzenie, które może być zarówno pozytywne lub negatywne, jak i neutralne. Zaskoczenie może objawiać się reakcjami fizycznymi jak odskoczenie, werbalnymi np. krzykiem lub mimiką twarzy – rozszerzenie oczu, otwarcie ust. Podobnie jak strach, zaskoczenie może spowodować wyzwolenie się instynktu walki lub ucieczki i podwyższenie poziomu adrenaliny. Badania naukowe wskazują duże znaczenie zaskoczenia w procesach poznawczych: niespodziewane wiadomości są bardziej zauważalne, zaskakujące argumenty bardziej przekonujące. Zaskakujące informacje są również łatwiej zapamiętywane [2].

### **1.1.3 Teoria Emocji Plutchika**

W 1980 roku amerykański psycholog Robert Plutchik opublikował swoją teorię emocji, według której istnieje osiem parami antagonistycznych emocji podstawowych. Parami tymi miały być:

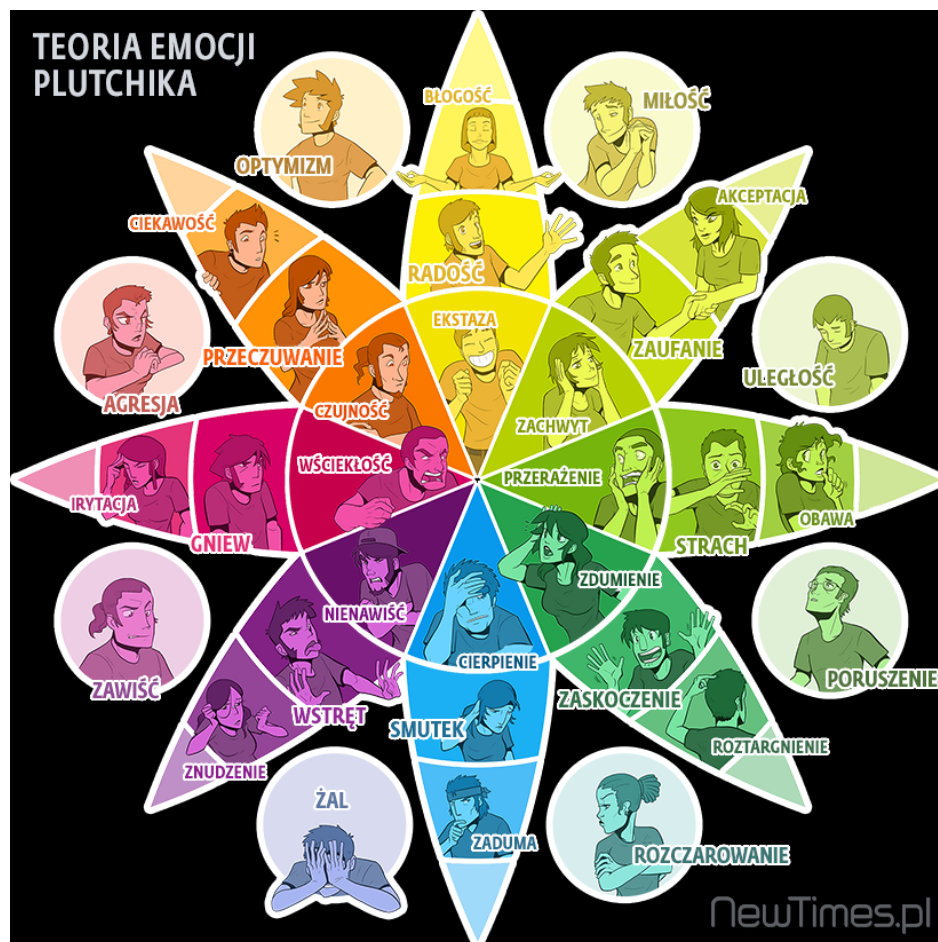
- radość i smutek,
- przeczuwanie i zaskoczenie,
- gniew i strach,
- zaufanie i wstręt.

Dla zilustrowania swojej idei emocji stworzył model w postaci stożka, który może być rozwinięty w koło emocji, przypominające koło barw. Koło emocji Plutchika zostało przedstawione na rysunku 1.2. Oprócz podstawowych emocji w modelu znajdują się emocje będące różnymi stopniami intensywności emocji podstawowej. Każda z emocji podstawowych jest przyjmowana jako średnia intensywność oraz są określone dla nich emocje o mniejszej oraz większej intensywności. Ostatnim elementem teorii Plutchika są kombinacje emocji: emocje takie jak miłość, uległość czy poruszenie będące połączeniem dwóch emocji podstawowych, między którymi znajdują się na kole emocji [5].

Teoria Plutchika opiera się na dziesięciu podstawowych założeniach:

- Emocje powstają w części mózgu, zwanej systemem limbicznym i podstawowe emocje są wspólne dla wszystkich ssaków.
- Na drodze ewolucji emocje człowieka ulegały zmianom, co pozwoliło na rozwinięcie emocji podstawowych w ponad 34000 różnych emocji.
- Emocje odgrywają ważną rolę w przetrwaniu.
- W każdej z ośmiu emocji podstawowych można wyróżnić pewne wspólne i rozpoznawalne wzorce, zwane prototypami.

- Istnieje osiem emocji podstawowych wspólnych dla wszystkich ssaków.
- Kombinacje emocji podstawowych stanowią oddzielne emocje.
- Emocje są hipotetycznymi konstruktami, które pozwalają opisać doświadczenia.
- Każda z emocji ma swoje przeciwieństwo.
- Między niektórymi emocjami istnieją pewne podobieństwa.
- Każda emocja ma różne stany intensywności [6].



Rys. 1.2: Koło uczuć Plutchika [7]

#### 1.1.4 Inne teorie emocji

Oprócz teorii Ekmana i Plutchika powstało wiele teorii opisujących emocje człowieka, różniące się między innymi liczbą emocji uznawanych za podstawowe. Wśród badaczy są tacy, którzy postulują, że podstawowych emocji jest mniej - dwie lub trzy,

w innych badaniach szacuje się liczbę podstawowych emocji na o wiele wyższą, np. w jednym z bardziej współczesnych liczbę tę określono na dwadzieścia siedem [2].

Inną popularną grupą teorii emocji jest hierarchiczność emocji. Według takich teorii większość podstawowych emocji może zostać rozbita na wiele emocji drugorzędnych, które z kolei mogą być dalej rozbite na emocje trzeciorzędne. Dla przykładu miłość może zostać rozłożona na uczucia przywiązania oraz pożądania, natomiast przywiązanie może zostać dalej rozłożone na sympatię, troskę, współczucie i czułość [2].

## 1.2 Analiza emocji

### 1.2.1 Cele analizy emocji

Automatyczna analiza emocji jest narzędziem o licznych zastosowaniach, których liczba ciągle wzrasta wraz z rozwojem technologii i cyfryzacją. Istotnymi obszarami, w których rozwój systemów rozpoznawania emocji są między innymi robotyka, marketing, edukacja oraz przemysł rozrywkowy [8].

Kluczowymi zagadnieniami analizy oraz rozpoznawania emocji wykorzystywanymi w robotyce są stworzenie modelu emocji, który mógłby zostać zastosowany jako część architektury poznawczej inteligentnych robotów i który mógłby odtwarzać odpowiednie wyrazy emocji typowe człowiekowi oraz rozpoznawanie emocji u ludzi w interakcji z robotem [9]. W edukacji analiza emocji jest użtkowana, aby znaleźć odpowiednie sposoby poprawy procesu uczenia, natomiast zarówno w marketingu jak i rozrywce przedmiotem badań jest ocena emocji konsumenta dla znalezienia odpowiedniego sposobu podejścia oraz zaproponowania produktu dopasowanego do niego [8].

### 1.2.2 Sposoby analizy emocji

#### *Mimika twarzy*

Metodą, która jest najbardziej charakterystyczna dla rozpoznawania emocji jest mimika twarzy. Proces ten w konwencjonalnych rozwiązaniach polega na detekcji regionów twarzy na obrazach i ich przetwarzania w taki sposób, aby dane mogły być wprowadzone do algorytmu uczenia maszynowego. W tym przypadku informacje teksturalne reprezentujące wygląd twarzy oraz informacje o lokacjach charakterystycznych punktów stanowią analizowane cechy. Dużym zainteresowaniem cieszą się również rozwiązania bazujące na głębokim uczeniu, których idea stosowania polega na uniknięciu uzależnienia się algorytmu od czynnika ludzkiego – osoby tworzącej lub wykorzystującej algorytm. Wyjątkowo skuteczne są konwolucyjne sieci neuronowe [9].

#### *Termowizyjne obrazy twarzy*

Zmiany stanów emocjonalnych mogą być rejestrowane z użyciem kamery termowizyjnej, dzięki powodowych przez nie zmian w rozprzestrzenianiu się krwi w naczyniach krwionośnych. Przyczyną tego są zjawiska rozszerzania i zwężania naczyń oraz pocenia skorelowane ze stanami emocjonalnymi. Kamery termowizyjne rejestrują zmiany



temperatury wynikające z tych zjawisk. Do rozpoznawania emocji rozważane są wybrane regiony zainteresowania takie jak czubek nosa, czoło, policzki oraz okolice oczu. W przypadku tego rodzaju analizy problematyczne jest pochodzenie ciepła z powodów niezwiązanych z emocjami [9].

### *Postawa ciała i kinematyka*

Język ciała jest jednym ze sposobów, które pozwalają rozpoznać emocje w sposób naturalny w interakcjach między ludzkich, co sugeruje, że takie sygnały mogą być również zastosowane w automatycznym rozpoznawaniu emocji przez komputer. Rozwiązania bazujące na ruchach ciała do modelowania pozycji ciała wykorzystują dane RGB lub map głębi. Na ich podstawie rozpoczynany jest proces analizy: detekcja osoby, szacowanie pozycji ciała, korzystając z wcześniej zdefiniowanego modelu np. szkieletu, który należy dopasować do rejestrowanego obrazu, a następnie klasyfikacja emocji – zazwyczaj z użyciem uczenia maszynowego lub głębokiego uczenia [9].

### *Aktywność mózgu*

Aktywność mózgu rejestrowana metodą EEG budzi zainteresowanie badaczy zajmujących się analizą i klasyfikacją emocji. Ta modalność jest bardzo atrakcyjną opcją, dzięki ominięciu potencjalnych zamierzonych lub nie zakłóceń wywoływanych przez osobę badaną, mogących zakłamywać dane docierające do systemu rozpoznawania. Typowymi cechami uzyskiwanymi z analizy EEG są czas, częstotliwość, wartości statystyczne, parametr Hjortha czy wymiar fraktalny. W badaniach prowadzonych w 2007 i 2014 roku stwierdzono, że klasyfikacja emocji, wykorzystująca aktywność elektryczną mózgu daje wysokie wyniki, nawet przy wykorzystaniu komercyjnej aparatury badawczej. Przez wiele lat najczęściej wykorzystywaną metodą analizy danych EEG był SVM, jednak obecnie coraz częściej sięga się do technik uczenia głębokiego [9].

### *Głos*

Kolejnym sposobem naturalnego rozpoznawania emocji w interakcjach między ludzkich jest głos, co jest powodem zainteresowania badaczy zastosowaniem jego analizy do rozpoznawania emocji. Automatyczne sposoby rozpoznawania emocji na podstawie danych akustycznych zazwyczaj skupiają się na parametrach takich jak wysokość, energia czy poziom głośności głosu, nie natomiast na znaczeniu wypowiedzianych słów. Cechy te mogą być rozważane jako całość lub rozważane po rozbiciu na krótsze odcinki. Tradycyjnie tego rodzaju klasyfikacja była wykonywana z użyciem ukrytych modeli Markova i SVM. Jednak również i tu w ostatnich latach popularnością cieszą się algorytmy uczenia głębokiego [9].

### *Peryferyjne odpowiedzi fizjologiczne*

Innym sposobem identyfikacji emocji jest wykorzystanie peryferyjnych sygnałów ciała, takich jak tętno, ciśnienie krwi, oddechu czy przewodnictwa skóry. Chociaż me-

toda wykazuje duży potencjał, zwłaszcza jako część większego multimodalnego systemu, rzadko jest wykorzystywana do szerokiej analizy emocji, a jedynie do ich pewnych składowych, takich jak poziom stresu bądź zmęczenia. Zastosowania takich danych do rozpoznawania emocji może być ułatwione, dzięki wzrostowi dostępności technologii przenośnych, osobistych urządzeń rejestrujących badane sygnały, np. inteligentnych zegarków [9].

### 1.3 Baza danych *ravdness*

Cała baza danych *RAVDESS* (ang. Ryerson Audio-Visual Database of Emotional Speech and Song) zawiera 7356 plików. Każdy z plików został oceniony 10 krotnie pod względem emocjonalnym, intensywności i autentyczności. Odnotowano wysoki poziom trafności emocjonalnej oraz wiarygodności. Dane walidacyjne są również ogólnodostępne i możliwe do pobrania.

Jak wspomniano audio-wizualna baza danych zawiera 7356 plików (całkowity rozmiar: 24,8 GB). Baza danych zawiera nagrania 24 profesjonalnych aktorów (12 kobiet, 12 mężczyzn), wypowiadających dwie zgodne leksykalnie wypowiedzi z neutralnym północnoamerykańskim akcentem. Mowa zawiera wyrazy spokoju, radości, smutku, złości, strachu, zaskoczenia i obrzydzenia, a piosenka zawiera emocje spokojne, radosne, smutne, złości i strachu. Każda ekspresja jest wykonana na dwóch poziomach intensywności emocjonalnej (normalny, silny). Wszystkie nagrania są dostępne w trybie tylko dźwięk (16 bitów, 48 kHz, rozszerzenie *.wav*), audio- wideo (720p, 48 kHz, rozszerzenie *.mp4*) i tylko wideo (bez dźwięku).

Podział plików:

- Pliki mowy audio (*Audio\_Speech\_Actors\_01-24.zip*, rozmiar: 215 MB) zawiera 1440 plików: 60 prób na aktora \* 24 aktorów = 1440 nagrań.
- Plik piosenek audio – 1012 nagrań.
- Pliki mowy video – 2880.
- Pliki piosenek video – 2024.

Sumarycznie baza danych *RAVDESS* obejmuje 7356 plików (2880+2024+1440+1012). W zrealizowanym projekcie wykorzystana została tylko część nagrań tzn. pliki mowy audio – 1440 plików.

Konwencja nazewnictwa plików Każdy z plików posiada unikalną nazwę pliku. Nazwa pliku składa się z 7-częściowego identyfikatora numerycznego (np. *02-01-06-01-02-01-12.mp4*). Definiują one kolejno charakterystykę nagrania:

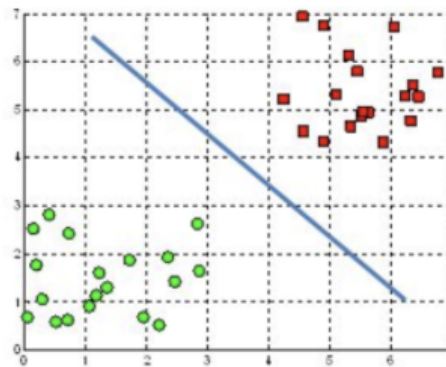
- Modalność (01 = audio-wideo, 02 = tylko wideo, 03 = tylko audio).
- Kanał wokalny (01 = mowa, 02 = piosenka).
- Emocje (01 = neutralny, 02 = spokojny, 03 = szczęśliwy, 04 = smutny, 05 = zły, 06 = przerażony, 07 = obrzydzony, 08 = zdziwiony).

- Intensywność emocjonalna (01 = normalna, 02 = silna).
- Zdanie wypowiedziane (01 = „*Kids are talking by the door*”, 02 = „*Dogs are sitting by the door*”).
- Powtórzenie (01 = 1 powtórzenie, 02 = 2 powtórzenie).
- Aktor (od 01 do 24. Aktorzy o numerach nieparzystych to mężczyźni, aktorzy o numerach parzystych to kobiety) [10].

## 1.4 Sieć neuronowa - *Bilstm*

## 1.5 Maszyna wektorów nośnych (SVM)

Maszyna wektorów zespolonych (SVM) jest prostym algorytmem uczenia maszynowego, cenionym dzięki swojej znacznej dokładności przy wymaganej stosunkowo niskiej mocy obliczeniowej, który może być stosowany zarówno w regresji jak i w klasyfikacji, najczęściej jednak w klasyfikacji. Celem algorytmu jest znalezienie hiperpłaszczyzny w przestrzeni o liczbie wymiarów równej liczbie cech, która pozwala na podział danych na klasy. Hiperpłaszczyzna to granica rozdzielająca punkty poszczególnych klas. Niektóre z punktów leżą blisko płaszczyzny i przez to wywierają wpływ na jej pozycję i orientację. Te punkty nazywane są wektorami nośnymi. Dzięki odpowiedniemu zastosowaniu tych wektorów możliwe jest uzyskanie takiej hiperpłaszczyzny, dla której odległości od punktów są jak największe. Na rysunkach 1.3 i 1.4 przedstawiono przykładowe hiperpłaszczyzny [11].



Rys. 1.3: Hiperpłaszczyzna dla dwóch cech [11]

Wartości funkcji dla SVM mieszczą się w przedziale od -1 do 1. Wartość równa 1 oznacza przynależność punktu do analizowanej klasy, natomiast -1 przynależność tego punktu do innej klasy. W celu minimalizacji marginesu pomiędzy punktami klas a hiperpłaszczyzną wykorzystuje się funkcję straty zawiasowej. Wartość funkcji straty

jest wyliczana jeśli wartość przewidziana oraz rzeczywista mają różny znak. Funkcja straty jest obliczana z wzoru [11]:

$$\min_w \lambda \|w\|^2 + \sum_{(i=1)}^n (1 - y_i < x_i, w >)_{+}$$

Na podstawie funkcji straty można obliczyć gradienty, które pozwalają na modyfikację wag. Gradienty są obliczane w następujący sposób [11]:

$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} (1 - y_i < x_i, w >)_{+} = 0 \text{ dla } y_i < x_i, w > \geq 1$$

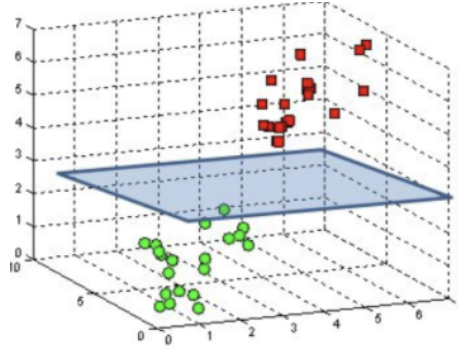
$$\frac{\delta}{\delta w_k} (1 - y_i < x_i, w >)_{+} = -y_i x_{ik} \text{ dla pozostałych}$$

Modyfikacja wektora wag może przebiegać na dwa sposoby w zależności od tego czy klasyfikacja przebiegła pomyślnie (klasa punktu została przewidziana poprawnie), czy też nie. W pierwszym przypadku zmodyfikowany wektor wag powstaje na podstawie wzoru [11]:

$$\omega = \omega - \alpha(2\lambda\omega)$$

W przeciwnym razie stosowany jest wzór [11]:

$$\omega = \omega + \alpha(y_i x_i - 2\lambda\omega)$$

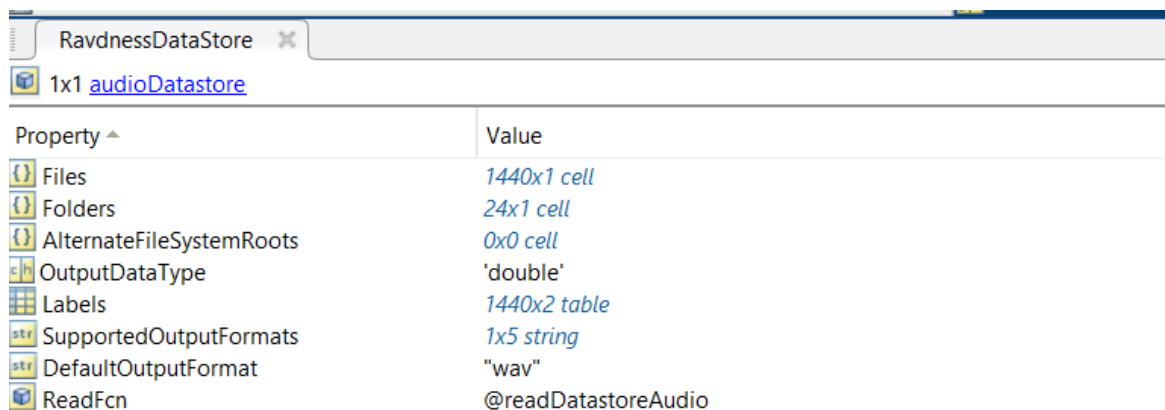


**Rys. 1.4:** Hiperpłaszczyzna dla trzech cech

## 2. Sieć neuronowa

### 2.1 Implementacja

Sieć neuronowa została zaimplementowana w Matlabie przy pomocy przygotowanych do tego celu funkcji. Jednak za nim do tego przystąpiono, należało dane odpowiednio przygotować. Zaczęto od zebrania wszystkich nagrań, które są rozdzielone na foldery, w jedną, wygodną w użytku zmienną. Do tego zadania świetnie nadała się struktura matlabowa **audioDatastore**, która została utworzona w celu zarządzania kolekcjami nagrań. Na rysunku 2.1 przedstawiono strukturę tej zmiennej.



| RavdnessDataStore                  |                     |
|------------------------------------|---------------------|
| 1x1 <a href="#">audioDatastore</a> |                     |
| Property ^                         | Value               |
| Files                              | 1440x1 cell         |
| Folders                            | 24x1 cell           |
| AlternateFileSystemRoots           | 0x0 cell            |
| OutputDataType                     | 'double'            |
| Labels                             | 1440x2 table        |
| SupportedOutputFormats             | 1x5 string          |
| DefaultOutputFormat                | 'wav'               |
| ReadFcn                            | @readDatastoreAudio |

Rys. 2.1: Struktura zmiennej audioDatastore

Dodatkowo wymagane było uzupełnienie właściwości „Label”, gdzie dla każdego nagrania przypisana miała zostać etykieta. Wybrano dwie etykiety: **speaker**, czyli która osoba wypowiadała się w danym nagraniu, i **emotion**, czyli jaką emocję dano nagranie miało reprezentować. Informacje te były zawarte w nazwie każdego pliku w konwencji, którą przedstawiono na rysunku 2.2. Etykiety najpierw zostały zmienione z napisu na zmienną kategorii, następnie przypisano je do właściwości „labels” w zmiennej audioDataStore

Table 1. Description of factor-level coding of RAVDESS filenames.

| Identifier | Coding description of factor levels   |
|------------|---|
| Modality   | 01 = Audio-video, 02 = Video-only, 03 = Audio-only  |
| Channel    | 01 = Speech, 02 = Song  |
| Emotion    | 01 = Neutral, 02 = Calm, 03 = Happy, 04 = Sad, 05 = Angry, 06 = Fearful, 07 = Disgust, 08 = Surprised |
| Intensity  | 01 = Normal, 02 = Strong  |
| Statement  | 01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door"                              |
| Repetition | 01 = First repetition, 02 = Second repetition   |
| Actor      | 01 = First actor, ..., 24 = Twenty-fourth actor   |

**Rys. 2.2:** Legenda, jakie informacje można wyciągnąć z nazw plików w bazie danych Ravdness

Podsumowanie liczbowe całych danych przedstawiono na rysunku 2.3. Pokazany jest na nim ile nagrań jest z danej kategorii w bazie

**Emotion:** 1440×1 categorical

Values:

|           |     |
|-----------|-----|
| Angry     | 192 |
| Calm      | 192 |
| Disgust   | 192 |
| Fearful   | 192 |
| Happy     | 192 |
| Neutral   | 96  |
| Sad       | 192 |
| Surprised | 192 |

**Rys. 2.3:** Podsumowanie liczbowe bazy danych Ravdness

### 2.1.1 Rozszerzenie danych

W bazie danych znajdowało się 1440 nagrań. Postanowiono rozszerzyć sztucznie liczbę tych danych przy pomocy narzędzia matlaba **audioDataAugmenter**. Pozwala on na utworzenie nowych plików na podstawie starych przy pomocy np.: dodawania szumu, przesuwania w czasie czy zmianę głośności. Wybrano rozszerzenie 30-krotne. Oznacza to, że z każdego pliku utworzono 30 nagrań. Sprawi to, że do dyspozycji będzie nie 1440 plików, ale 42 300 plików. Utworzono na podstawie tych plików nową zmienną

audioDatastore. Dodatkowo wykorzystano z poprzedniej etykiety. Każdą starą etykietę również należało powtórzyć 30 razy, żeby każde nowe nagranie miało odpowiednie oznaczenie.

Parametry ustawiane w zmiennej audioDataAugmenter i wybrane wartości:

- 'TimeStretchProbability' — 0,
- 'VolumeControlProbability' — 0,
- 'PitchShiftProbability' — 0.5,
- 'TimeShiftProbability' — 1,
- 'TimeShiftRange' — [-0.3,0.3],
- 'AddNoiseProbability' — 1,
- 'SNRRange' — -20,40,

## 2.2 Ekstrakcja cech

W matlabie do wyciąganie cech z plików audio również dostępna jest gotowa funkcja o nazwie audioFeatureExtractor. Na podstawie literatury wyodrębniono 4 cechy: *mffc*, *mffcdelta*, *mffcdeltadelta* i *pitch*. Wszystkie one były dostępne jako opcje w audioFeatureExtractor. Oprócz tego, należało zdefiniować okno, to znaczy na jakie ramki należy rozdzielić dany sygnał dźwiękowy. Z każdej takiej ramki otrzyma się osobno wartość danej cechy. Jako ramkę ustalono 0.3 sekundy, czyli  $0.3 \cdot \text{częstotliwość próbkowania}$  (w przypadku bazy danych Ravdness wynosi ona 48000Hz), czyli około 14 400 próbek na ramkę.

W zadaniach rozpoznawania, takich jak rozpoznawanie fonemów lub wykrywanie aktywności głosowej, klasyczną cechą wprowadzania są współczynniki cepstralne mel-częstotliwości (MFCC). Opisują one chwilowy, widmowy kształt obwiedni sygnału mowy. Jednak sygnały mowy są sygnałami zmiennymi w czasie i o stałym przepływie.

Powszechną metodą wyodrębniania informacji o takich przejściach jest wyznaczenie pierwszej różnicy cech sygnału, znanej jako delta cechy. W szczególności, dla cechy  $f_k$ , w chwili  $k$ , odpowiadająca jej delta jest zdefiniowana jako  $\Delta_k = f_k - f_{k-1}$ . Druga różnica, znana jako delta-delta, to odpowiednio  $\Delta\Delta_k = \Delta_k - \Delta_{k-1}$ .

Wysokość tonu (ang. *pitch*) obliczono za pomocą metody NFC (ang. *Normalized Filtered Correlation*).

Wyodrębnione cechy dla każdego nagrania zostały przedstawione w postaci macierzy komórkowej (42 300 wierszy), gdzie w każdej komórce była macierz o rozmiarach liczba cech na liczba ramek. Liczba cech wynosiła 40 (mfcc, mffcdelta i mffcdeltadelta po 13 i pitch). Liczba ramek jest różna dla każdego nagrania. Dla wszystkich cech obliczono również średnią i odchylenie standardowe.

### 2.2.1 Parametry sieci

Parametry sieci zostały ustawione za pomocą funkcji *trainingOptions*. Zwraca ona opcję w postaci jednej zmiennej, które można wykorzystać potem w innej funkcji stosowanej do utworzenia i trenowania sieci. Ustawiono następujące parametry:

- **"adam"** — tzw. solver dla sieci neuronowej, algorytm optymalizacji Adama jest rozszerzeniem stochastycznego spadku gradientu, który ostatnio zyskał szersze zastosowanie w zastosowaniach głębokiego uczenia w wizji komputerowej i przetwarzaniu języka naturalnego.
- **MiniBatchSize** — 512 — Ilość danych zawartych w każdej zmianie wagi podepokki jest znana jako *batchsize*. Na przykład w przypadku zestawu danych uczących zawierającego 1000 próbek pełny rozmiar partii będzie wynosił 1000, rozmiar *minibatch* będzie wynosił 500, 200 lub 100, a *batchsize* online będzie wynosił tylko 1,
- **InitialLearnRate** — 0.0005 — wielkość kroku w każdej iteracji, jednocześnie zbliżająca się do minimum funkcji straty (*loss function*)
- **LearnRateDropPeriod** — 15 — oprogramowanie mnoży globalną szybkość uczenia się przez współczynnik spadku za każdym razem, gdy minie określona przez ten parametr liczba epok
- **LearnRateSchedule** - „piecewise” — oprogramowanie aktualizuje szybkość uczenia się co określoną liczbę epok, mnożąc przez określony współczynnik.
- **MaxEpochs** — 30 - maksymalna liczba epok
- **Shuffle** — „every-epoch” — mieszanie danych treningowych przed każdą epoką treningową i mieszanie danych walidacyjne przed każdą walidacją sieci,
- **"Verbose"** — false — Wskaźnik do wyświetlania informacji o postępach treningu ,
- **"Plots"** — „Training-Progress” — Czy pokazywać wykres, który przedstawia utratę i dokładność *minibatch*, utratę i dokładność walidacji oraz dodatkowe informacje o postępach treningu.

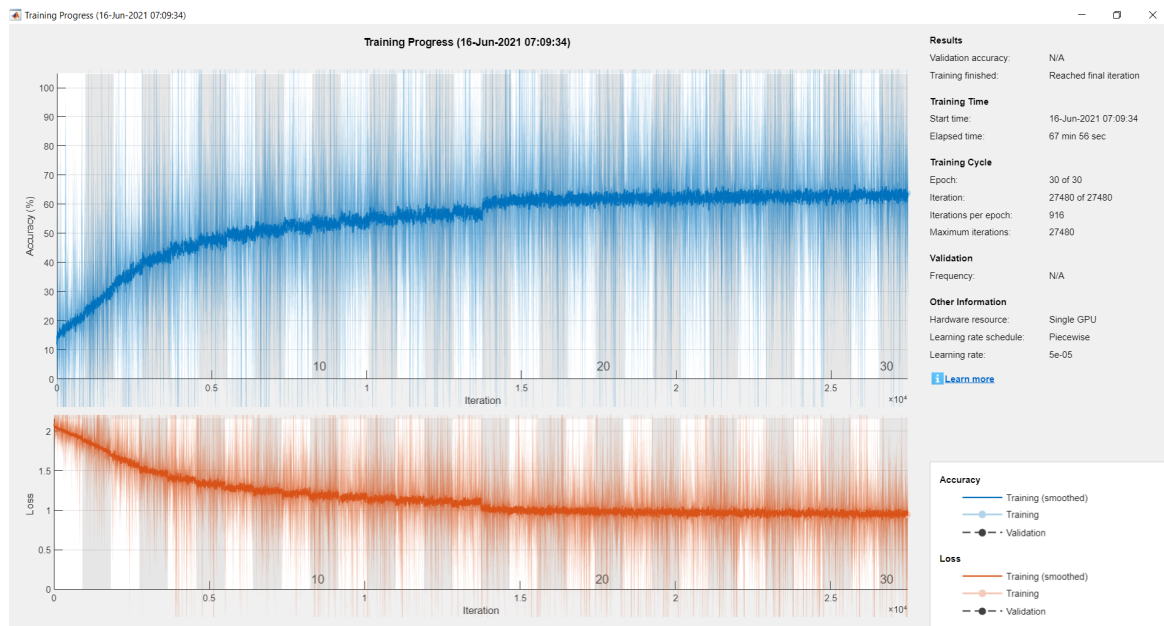
Kolejnym krokiem było ustalenie warstw sieci, które przedstawiono poniżej wraz z parametrami) :

- **sequenceInputLayer** — warstwa wejściowa,
- **dropoutLayer** (wartość prawdopodobieństwa) — Warstwa Dropout losowo ustawia jednostki wejściowe na 0 z częstotliwością na każdym kroku w czasie treningu, co pomaga zapobiegać nadmiernemu dopasowaniu. Ustawiono prawdopodobieństwo zerowania na 0.01



- **bilstmLayer**(Liczba jednostek ukrytych ,”OutputMode”,”last”) — Dwukierunkowa warstwa LSTM (BiLSTM) uczy się dwukierunkowych długoterminowych zależności między krokami czasowymi szeregów czasowych lub danych sekwencji. Te zależności mogą być przydatne, gdy sieć ma uczyć się na podstawie pełnych szeregów czasowych w każdym kroku czasowym. Liczba jednostek ukrytych (odpowiada ilości informacji zapamiętanych pomiędzy krokami czasowymi (stan ukryty). Stan ukryty może zawierać informacje ze wszystkich poprzednich kroków czasowych, niezależnie od długości sekwencji. Jeśli liczba ukrytych jednostek jest zbyt duża, warstwa może nadmiernie pasować do danych uczących. Wartość ta może wahać się od kilkudziesięciu do kilku tysięcy. „OutputMode” oznacza formę danych wyjściowych.
- **dropoutLayer**(wartość prawdopodobieństwa) - jak wyżej, wartość prawdopodobieństwa=0.01
- **fullyConnectedLayer**(liczba kategorii emocji) - ustawia opcjonalne parametry i inicjalizację, szybkość uczenia się i regularyzację oraz nazwy przy użyciu par nazwa-wartość.
- **softmaxLayer** - Funkcja Softmax to funkcja aktywacji, która zamienia liczby zwane logitami na prawdopodobieństwa sumujące się do jednego. Funkcja Softmax wyprowadza wektor, który reprezentuje rozkłady prawdopodobieństwa listy potencjalnych wyników
- **classificationLayer** — Warstwa klasyfikacji oblicza straty entropii krzyżowej dla zadań klasyfikacji i klasyfikacji ważonej z wzajemnie wykluczającymi się klasami.

Następnie przystąpiono do utworzenia i trenowania sieci. Przebieg trenowania sieci przedstawiono na rysunku [2.4](#).



Rys. 2.4: Przebieg trenowania sieci neuronowej.

Sieć została zapisana do późniejszego użytkowania w postaci pliku mat.

### 2.2.2 Ocena i testowanie

Aby zapewnić dokładną ocenę utworzonego modelu, zastosowanie trenowanie i sprawdzanie poprawności za pomocą walidacji krzyżowej z pozostawieniem części plików od jednego mówcy. W tej metodzie trenuje się sieć używając danych od mówców k-1, a następnie sprawdzana jest ona na pozostałych danych od Jednego mówcy. Tę procedurę powtarza się dla każdego mówcy. Ostateczna dokładność walidacji to średnia z wszystkich iteracji.

Jedną iterację można podzielić na następujące etapy:

1. Podzielenie pliku audioDataStore na zestawy uczące i walidacyjne i przekonwertowanie ich na tall arrays. Zestaw uczący do dane 23 mówców, zestaw walidujący pozostały jeden.
2. Wyodrębnienie cech z zestawu treningowego. Zmiana orientację funkcji tak, aby czas przebiegał wzdłuż wierszy, aby były zgodne z SequenceInputLayer.
3. Użycie zestawu uczącego, aby określić średnią i odchylenie standardowe każdej cechy. Na ich podstawie należy znormalizuj zestawy treningowe i walidacyjne.
4. Skonwertowanie sekwencji tak, aby każda sekwencja składała się z dwudziestu wektorów cech z nakładaniem się 10 wektorów cech.
5. Replikacja etykiet zestawów treningowych i walidacyjnych, aby odpowiadały jeden do jednego z sekwencjami.

6. Zdefiniowanie sieci neuronowej zgodnie z wcześniejszymi parametrami
7. Trenowanie sieci neuronowej na podstawie zbiorów treningowych
8. Ocena sieci zestawem walidującym

## 2.3 Wyniki

## 3. SVM

### 3.1 Implementacja

W ramach klasyfikacji wykorzystującej maszynę wektorów nośnych (SVM) posłużono się wyekstrahowanymi cechami dla sieci neuronowej. Cechy zapisano w macierzy *features*, natomiast znaczniki w wektorze *label*. W celu ułatwienia operacji zamieniono znaczniki słowne na liczbowe od 1 do 8 i zapisano je jako nowy wektor *label\_number*. Następnie przygotowano kod, który pozwala na wyekstrahowanie pożądaných znaczników emocji wraz z cechami im odpowiadającymi oraz zapisuje je w zmiennych *new\_label* i *new\_features*. Za pomocą wbudowanej funkcji *any* wykonano operację logiczną OR w celu usunięcia zbędnych zer z wektora i macierzy. Tak przygotowane dane podzielono losowo na dane treningowe – 70% oraz dane testowe – 30% wykorzystując w tym celu funkcję *randperm*. Funkcja ta zwraca wektor niepowtarzających się losowych liczb z zakresu 1 do  $n$ , gdzie  $n$  oznacza argument funkcji. Podzielone dane oraz znaczniki wpisano w zmienne reprezentujące dane treningowe i testowe. Kod realizujący niniejszą część przedstawia Listing. 3.1.

Na podstawie grupy treningowej zapisanej jako *y\_train* i *y\_train* wykonano operację krosvalidacji, wykorzystując do tego funkcję *cvpartition*. Funkcja ta przyjmuje za argumenty wektor operacji (*y\_train*) oraz parametr *KFold*, który określa typ walidacji krzyżowej. W tym wypadku zbiór podzielono na  $k$ -krotne podpróbki wybierane losowo, ale o mniej więcej takim samym rozmiarze. Domyślnie parametr  $k$  przyjmuje wartość 10 i taki zastosowano w niniejszym projekcie na podstawie doświadczenia, że mniejsza wartość skutkuje niższą wartością klasyfikacji.

Następnie wygenerowano model SVM:

- dla dwóch klas za pomocą funkcji *fitcsvm*. Funkcja przyjmuje za argumenty macierz predyktorów – *train data* oraz ich znaczniki – *test data*, ponadto przyjmuje parametr *KernelFunction*, który pozwala na wyznaczenie gęstości rozkładu zmiennej losowej na podstawie uzyskanych prób oraz parametr *rgf*, który zapewnia, że rozkład będzie Gaussowski.
- dla liczby klas powyżej dwóch za pomocą funkcji *fitcecoc*. Funkcja przyjmuje za argumenty macierz predyktorów –  $X$  oraz ich znaczniki –  $Y$ , parametr *Learner* określa z jakiego szablonu SVM funkcja skorzysta. W tym wypadku szablon utworzono funkcją *templateSVM*, który podobnie jak wyżej określa funkcję jądra *KernelFunction*, która pozwala na znormalizowanie predyktorów. Ponadto funkcja *fitcecoc* przyjmuje parametr *FitPosterior*, który określa czy przekształcić wynik w prawdopodobieństwo a posteriori, czyli takie które obliczane jest

na podstawie otrzymanych rezultatów. W parametrze *ClassNames* określa się nazywa klas, *CVPartition* jest parametrem pozwalającym określić krosvalidację, z kolei parametr *Verbose* pozwala na wyświetlanie informacji diagnostycznych na temat postępu uczenia. Parametr *options* pozwala na zastosowanie programu *Parallel Computing Toolbox* za pomocą parametru *UseParallel*, który pozwala na obliczenia równoległe w czasie.

Po utworzeniu modeli i przeprowadzeniu uczenia ostatnim krokiem była klasyfikacji przygotowanej grupy testowej *X\_test* oraz obliczenie na podstawie wyników klasyfikacji jej skuteczności *accuracy*. Uczenie w przypadku dwóch klas przedstawia Listing 3.2, natomiast w przypadku wielu klas Listing 3.3. Dodatkowo w celu przetestowania algorytmu na mniejszej liczbie cech zdecydowano się zastosować metodę PCA. Metoda ta na podstawie wektora wejściowego tworzy wektor o mniejszym wymiarze. W tym celu wyznacza się wektory własne oraz własności własne macierzy kowariancji. Następnie dokonuje się sortowania wektorów własnych zgodnie z odpowiadającymi im wartościami własnymi i odrzuca się te wektory, które mają najmniejsze znaczenie. Kryterium to określa się za pomocą parametru  $\lambda$ . Kos realizujący algorytm metody PCA przedstawia Listing 3.4.

**Listing 3.1:** Kod realizujący przygotowanie danych do uczenia maszynowego

```
1 %% Przygotowanie danych
load('LabelAllFeatures.mat');
vals = table2array(LabelAllFeatures);
label=vals(:,2);
5 load('cechy.mat')
features=allFeatures;
features=features';
wyniki=ones(15,1) %wektor wyników klasyfikacji
9
%% Zmiana na liczbowe znaczniki
label_number=zeros(length(label),1)
for i=1:length(label)
13     if(label(i)=='Neutral')
        label_number(i)=1;
    end
    if(label(i)=='Calm')
17         label_number(i)=2;
    end
    if(label(i)=='Happy')
        label_number(i)=3;
21     end
    if(label(i)=='Sad')
        label_number(i)=4;
    end
25     if(label(i)=='Angry')
```

```

        label_number(i)=5;
    end
    if(label(i)=='Fearful')
29         label_number(i)=6;
    end
    if(label(i)=='Disgust')
33         label_number(i)=7;
    end
    if(label(i)=='Surprised')
        label_number(i)=8;
    end
37 end
%% Wyciąganie tylko określonych cech

for i=1:length(label)
41     if(label_number(i)==5)
        new_label(i)=label_number(i);
        new_features(i,:)=features(i,:);
    end
45     if(label_number(i)==2)
        new_label(i)=label_number(i);
        new_features(i,:)=features(i,:);
    end
49 end
%% Usunięcie wierszy zawierających 0
new_label=new_label';
new_label = new_label(any(new_label,2),:);
53 new_features=new_features(any(new_features,2),:);

% Podział danych na dane treningowe i testowe
rand_num=randperm(3840);
57 % Treningowe 70%
X_train=new_features(rand_num(1:2688),:);
y_train=new_label(rand_num(1:2688),:);

61 % Testowe 30%
X_test=new_features(rand_num(2689:3840),:);
y_test=new_label(rand_num(2689:3840),:);

```

**Listing 3.2:** Kod realizujący uczenie dwóch klas

```

1 % Kroswalidacja
c=cvpartition(y_train,'k',10)

% SVM dla dwóch klas

```

```

5 Mdl = fitsvm(X_train,y_train,'KernelFunction','rbf','
    OptimizeHyperparameters','auto',...
    'HyperparameterOptimizationOptions',struct('
        AcquisitionFunctionName',...
        'expected-improvement-plus','ShowPlots',false));

9 % Obliczanie dokładności klasyfikacji
accuracy=(sum(predict(Mdl, X_test)==y_test)/length(y_test))
    *100
wyniki=accuracy

```

**Listing 3.3:** Kod realizujący uczenie wielu klas

```

t = templateSVM('Standardize',true,'KernelFunction','
    polynomial');
opts = statset('UseParallel',true);
3 % Krosvalidacja
c=cvpartition(y_train,'k',10)

% SVM dla dwóch klas
7 Mdl = fitcecoc(X_train,y_train,'Learners',t,'FitPosterior',
    true,...
    'ClassNames',{'1','2','3'},...
    'CVPartition',c,'Verbose',2,'options',opts);

11 % Obliczanie dokładności klasyfikacji
suma=0;
for j=1:length(Mdl.Trained)
    [pred score cost] = predict(Mdl.Trained{j,1}, X_test);
15 pred1 = str2double(pred);
    accuracy=(sum(pred1==y_test)/length(y_test))*100;
    suma=suma+accuracy;
end

19 suma=suma/length(Mdl.Trained)
wyniki=suma

```

**Listing 3.4:** Kod realizujący redukcję wymiarowości cech za pomocą metody PCA

```

%% Redukcja cech PCA
lambda=0.999
3 [NowyWektor,WW,DV] = PCA(features,lambda);
NowyWektor=NowyWektor'

```

```

%% Przypisanie nowego wektora cech do wektora cech
7 features=NowyWektor;

function [NowyWektor,WW,DV]=PCA(X, lambda)
% X-wektor cech
11 %      W kolumnach znajdują się kolejne cechy (zmienne)
%      W wierszach znajdują się kolejne osoby (obserwacje)
% lambda - próg istotności wybranych wektorów własnych

15 % NowyWektor - nowy wektor cech wyznaczony z użyciem metody
    PCA
% WW - wybrane wektory własne (wg. instrukcji powinny
    znajdować się w osobnych wierszach)
% DV - wybrane wartości własne

19 % Obliczenie wektora cech pomniejszonego o średnią
[n,c] = size(X);
X_sred = X - ones(n,1) * mean(X);

23 % Obliczenie kowariancji
kowariancja=X_sred'*X_sred/(n-1);

% TO DO
27 [V,D] = eig(kowariancja)
% DV=diag(D);
% WW=V;
D=diag(D);
31 WW=[];
DV=[];
    sumaPrzed=sum(D)
% Udział procentowy wybranych wartości własnych
35 prog=0;
% Wyszukujemy najważniejsze komponenty
while prog < lambda
    % TO DO
39    maxD=max(D)
    Wynik=V(D==maxD,: )
    WW=[WW; Wynik]
    Wynik2=D(D==maxD,: )
43 DV=[DV; Wynik2]
    V(D==maxD,: ) = [];
    D(D==maxD) = [];
    sumaPo=sum(DV)
47 prog=sumaPo/sumaPrzed

```



```

end

% Obliczamy nowe współczynniki
51 NowyWektor=WW*X_sred';

```

## 3.2 Otrzymane wyniki

W pierwszej kolejności przetestowano algorytm na pełnym zbiorze emocji, wynoszącym 8. Otrzymano wynik skuteczności klasyfikacji na poziomie 17%, wobec czego zdecydowano się na wykonanie testów ze względu na:

- rodzaj emocji,
- liczbę klas,
- liczbę cech.

Skuteczność klasyfikacji w ramach każdego z testów przeprowadzono poprzez 15-krotne powtórzenie algorytmu od momentu losowania danych treningowych i testowych dla 39 cech. Na podstawie otrzymanych wyników obliczono medianę oraz średnią. Zdecydowano się na przedstawienie wyników w postaci mediany wyrażonej procentowo w celu odrzucenia skrajnych wartości.

W Tab. 3.1 przedstawiono wyniki testów skuteczności klasyfikacji pomiędzy dwoma wybranymi klasami. Zdecydowano się na zbadanie skuteczności klasyfikacji wybierając pary skrajnych emocji jak radość (*ang. happy*) i smutek (*ang. sad*) czy strach (*ang. fearful*) i złość (*ang. angry*) oraz pary emocji, które są do siebie zbliżone jak spokój (*ang. calm*) i neutralność (*ang. neutral*) czy radość i zaskoczenie (*ang. surprised*). Jak można zaobserwować najwyższą wartość skuteczności otrzymała para *Calm-Neutral* w wysokości 74,5% (oznaczona kolorem zielonym), natomiast najniższą wartość 55,4% para *Calm-Angry* (oznaczona kolorem czerwonym).

**Tab. 3.1:** Wyniki testów skuteczności klasyfikacji pomiędzy dwoma klasami

| Para emocji         | Mediana, % |
|---------------------|------------|
| Calm - Neutral      | 74,5       |
| Calm - Angry        | 55,4       |
| Happy - Sad         | 62,1       |
| Happy - Surprised   | 64,8       |
| Happy - Disgust     | 60,8       |
| Fearful - Angry     | 62,8       |
| Fearful - Surprised | 61,9       |

Do dalszej analizy wpływu liczby cech na skuteczność klasyfikacji zdecydowano się wybrać najlepszy i najgorszy wynik, czyli parę emocji *Calm - Neutral* oraz *Calm -*

*Angry*. Wyniki przedstawiono w Tab. 3.2. Skuteczność klasyfikacji zbadano dla 39, 12 i 7 cech, których redukcję przeprowadzono za pomocą metody PCA. Jak można zaobserwować w przypadku pary *Calm–Neutral* najwyższą wartość skuteczności osiągnięto dla wyjściowej liczby cech – 39, natomiast w przypadku pary *Calm – Angry* dla 12 cech. Jednakże pomiędzy wartością skuteczności klasyfikacji dla pary *Calm–Angry* zaobserwowano niewielkie podwyższenie skuteczności, zaledwie o 0,6%.

**Tab. 3.2:** Wyniki skuteczności klasyfikacji dwóch par emocji ze względu na liczbę cech

| Liczba cech | Calm – Neutral<br>Mediana, % | Calm – Angry<br>Mediana, % |
|-------------|------------------------------|----------------------------|
| 39          | 74,5                         | 55,4                       |
| 12          | 71,3                         | 56                         |
| 7           | 67,5                         | 53                         |

Następnie po zbadaniu jak algorytm uczy się dwóch klas, zdecydowano się dołożyć dodatkowo trzecią klasę. Z racji najwyższej wartości skuteczności klasyfikacji dla pary *Calm–Neutral* zbadano jak będzie się zmieniać skuteczność po dołożeniu do niniejszej pary każdej z emocji ze zbioru – *Happy*, *Sad*, *Angry*, *Fearful*, *Surprised* i *Disgusted*. Wyniki badania przedstawiono w Tab. 3.3. Podobnie jak powyżej kolorem zielonym oznaczono najwyższą wartość skuteczności klasyfikacji, natomiast czerwonym najniższą. Jak można zaobserwować skuteczność klasyfikacji spadła o średnio 30% w porównaniu z wartościami jakie otrzymano w przypadku dwóch klas. Jednocześnie można zaobserwować, że wartości bez względu na dobór dodatkowej emocji do pary *Calm – Neutral* nie różnią się między sobą znacznie

**Tab. 3.3:** Wyniki skuteczności klasyfikacji dla trzech klas

| Mediana, %     | Happy | Sad  | Angry | Fearful | Surprised | Disgusted |
|----------------|-------|------|-------|---------|-----------|-----------|
| Calm – Neutral | 39,5  | 39,3 | 39,2  | 39,4    | 38,7      | 39,3      |

Po otrzymaniu wyników dla trzech klas sprawdzono dla trójki, która wykazała najwyższą wartość skuteczności, jak wpłynie na wynik zmiana liczby cech. Wyniki testów przedstawia Tab. 3.4. Podobnie jak podczas testowania pary *Calm – Angry* (Tab. 3.2) najwyższą wartość klasyfikacji otrzymano dla 12 cech, natomiast najniższą dla bazowej liczby 39 cech.

**Tab. 3.4:** Wyniki skuteczności klasyfikacji dla trzech emocji ze względu na liczbę cech

| Liczba cech | Calm – Neutral – Happy<br>Mediana, % |
|-------------|--------------------------------------|
| 39          | 39,5                                 |
| 12          | 45,8                                 |
| 7           | 42,8                                 |

Ze względu na wyraźny spadek skuteczności klasyfikacji po dołożeniu trzeciej emocji oraz na wynik skuteczności dla ośmiu klas na poziomie 17% zdecydowano się nie sprawdzać jak algorytm zachowa się po dokładaniu następnych cech. Zastosowana metoda nie mogłaby zostać wykorzystana jako jednoznaczny klasyfikator nawet w przypadku 2 cech. Średnia z wartości przedstawionych w Tab. 3.1 wynosi 63,2%. Powodami otrzymania takich wyników mogą być:

- niedostatecznie różnicujące cechy,
- nieodpowiednio dobrane parametry modelu SVM.

## 4. Wnioski

## 5. Podsumowanie

# Bibliografia

- [1] A. Scarantino and R. de Sousa, “Emotion,” in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta, ed.), Metaphysics Research Lab, Stanford University, Summer 2021 ed., 2021.
- [2] K. Cherry, “The 6 types of basic emotions and their effect on human behavior,” [www.verywellbeing.com](http://www.verywellbeing.com), 4 2021.
- [3] “The Truth Behind Lie To Me.” <https://www.paulekman.com/blog/truth-behind-lie/>, 6 2017.
- [4] “Emocje podstawowe i ich wpływ na ludzkie zachowanie.” <https://sentimenti.pl/blog/emocje-podstawowe-rodzaje-zachowanie-nauka/>, 6 2020.
- [5] “Plutchik’s Wheel of Emotions: Exploring the Emotion Wheel.” <https://www.6seconds.org/2020/08/11/plutchik-wheel-emotions/>.
- [6] “Robert Plutchik’s Wheel of Emotions.” <https://www.toolshero.com/psychology/wheel-of-emotions-plutchik/>.
- [7] “Emocje dla dzieci.” <https://www.dzieckiembadz.pl/2017/07/emocje-dla-dzieci.html>, 7 2017.
- [8] A. Dziedzickis, A. Kaklauskas, and V. Bucinskas, “Human emotion recognition: Review of sensors and methods,” *Sensors*, vol. 20, no. 3, 2020.
- [9] M. Spezialetti, G. Placidi, and S. Rossi, “Emotion recognition for human-robot interaction: Recent advances and future perspectives,” *Frontiers in Robotics and AI*, vol. 7, p. 145, 2020.
- [10] “RAVDESS.” <https://smartlaboratory.org/ravdess/>, 2021.
- [11] R. Gandhi, “Support vector machine — introduction to machine learning algorithms,” *towards data science*, 6 2018.