Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Radioelektroniki i Technik Multimedialnych

Praca dyplomowa inżynierska

na kierunku Telekomunikacja
w specjalności Techniki Bezprzewodowe i Multimedialne

Projekt i realizacja systemu detekcji emocji poprzez analizę dźwięku za pomocą algorytmów uczenia maszynowego

Sandra Rojek
Numer albumu 318825

promotor
dr inż. Piotr Bobiński

WARSZAWA 2025

Pragnę serdecznie podziękować
Promotorowi Panu dr inż. Piotrowi Bobińskiemu
za życzliwość i otwartość na moje pomysły,
dzięki czemu zawsze czułam się wysłuchana i wspierana.

Niniejszą pracę inżynierską pragnę dedykować
mojej rodzinie, której wsparcie i miłość
pozwoliły mi zrozumieć, jak ważne są emocje,
które kryją się za każdym słowem.

Projekt i realizacja systemu detekcji emocji poprzez analizę dźwięku za pomocą algorytmów uczenia maszynowego

Streszczenie.

Praca inżynierska dotyczy projektu z zakresu uczenia maszynowego, którego celem było wykrywanie emocji w głosie, takich jak radość, smutek, zaskoczenie, złość i neutralność, przy użyciu konwolucyjnych sieci neuronowych. W pracy przeanalizowano wyniki działania modelu na zbiorze testowym wyodrębnionym z bazy *Emotional Speech Database* oraz autorskiej bazy danych, a także zaimplementowano model w aplikacji webowej.

Wstęp zawiera omówienie tła historycznego, znaczenia *Emotion AI* we współczesnym świecie oraz zastosowań w psychologii, bezpieczeństwie publicznym i interakcji człowiek-komputer. Rozdziały teoretyczne obejmują podstawy uczenia maszynowego, proces przekształcania plików dźwiękowych, charakterystykę modeli oraz szczegółowy opis sieci konwolucyjnych.

Końcowa część pracy przedstawia aplikację webową umożliwiającą wykrywanie emocji w próbkach dźwiękowych. Aplikacja oferuje także funkcje rejestracji i odsłuchu, co pozwala użytkownikowi ocenić przewidywania modelu.

Słowa kluczowe: uczenie maszynowe, sieci konwolucyjne, *Emotional Speech Database*, aplikacja webowa, *Emotion AI*

Design and Implementation of an Emotion Detection System through Sound Analysis Using Machine Learning Algorithms

Abstract.

The engineering thesis focuses on a machine learning project aimed at detecting emotions in speech, such as happiness, sadness, surprise, anger, and neutrality, using convolutional neural networks. The study analyzes the model's performance on a test set extracted from the *Emotional Speech Database* and an author-created dataset, as well as implements the model in a web application.

The introduction discusses the historical background, the importance of *Emotion AI* in the modern world, and its applications in psychology, public safety, and human-computer interaction. The theoretical chapters cover the fundamentals of machine learning, the process of transforming audio files, the characteristics of machine learning models, and a detailed description of convolutional neural networks.

The final part of the thesis presents a web application that enables the detection of emotions in audio samples. The application also offers recording and playback functionalities, allowing users to evaluate the model's predictions.

Keywords: machine learning, convolutional neural networks, *Emotional Speech Database*, web application, *Emotion AI*

Spis treści

1. Wstęp

Celem niniejszej pracy inżynierskiej jest stworzenie systemu do wykrywania emocji w dźwięku z wykorzystaniem algorytmów uczenia maszynowego, w tym głębokich sieci neuronowych. Dodatkowo praca obejmuje zapewnienie odpowiedniej jakości i ilości danych oraz świadome dobranie modelu i jego parametrów w celu uzyskania jak najwyższej dokładności na danych walidacyjnych i testowych. We wstępie przedstawiono ogólną charakterystykę sztucznej inteligencji oraz uczenia maszynowego, a także historię rozwoju koncepcji rozpoznawania emocji w głosie. Na zakończenie rozdziału omówiono dziedziny korzystające z detekcji emocji oraz potencjalne kierunki jej przyszłego zastosowania.

Fascynacja ludzkim mózgiem sprawiła, że wielu inżynierów zaczęło zgłębiać wiedzę na jego temat. Coraz częściej pojawiały się pomysły stworzenia struktury, która naśladowałaby działanie ludzkiego mózgu. W ten sposób z czasem narodziła się koncepcja sztucznej inteligencji wraz z jej nieodłącznym elementem: uczeniem maszynowym.

W miarę rozwoju sztuczna inteligencja i uczenie maszynowe znalazły zastosowanie w psychologii. AI zaczęto łączyć z ludzkimi emocjami, co doprowadziło do powstania nowej gałęzi w tej dziedzinie, zwanej *Emotion AI*.

Emocje stanowią integralną część życia każdego człowieka. Często działają jak wewnętrzny kompas, pomagając zrozumieć własne potrzeby. Od zarania dziejów towarzyszyły ludzkości, będąc fundamentem komunikacji na długo przed pojawieniem się mowy.

Sięgając czasów nowożytnych, eksploracją emocji zajmował się Charles Darwin, który w swojej publikacji *The Expression of the Emotions in Man and Animals*[1] stwierdził, że emocje są reakcją organizmu na czynniki zewnętrzne. Reakcje te przejawiają się między innymi w mimice, gestykulacji czy głosie. Badania Charlesa Darwina stały się podstawą do późniejszego rozumienia emocji [1].

W latach 70. XX wieku Paul Ekman prowadził badania nad mikroekspresjami, które pojawiają się na twarzy w przypływie silnych uczuć. Na tej podstawie wyodrębnił sześć podstawowych emocji, jakimi są: szczęście, smutek, złość, strach, zaskoczenie oraz wstręt. Stworzył teorię, że sposób wyrażania emocji jest uniwersalny na całym świecie [2].

Uczenie maszynowe pojawiło się w latach 50. XX wieku, a przełomowym wydarzeniem była konferencja w Dartmouth College w Hanoverze w 1956 roku. Organizator spotkania John McCarthy, wraz z zaproszonymi przez siebie naukowcami, uznał, że ze względu na szybki rozwój technologii maszyny mogą kiedyś osiągnąć inteligencję porównywalną z

[1] Polski tytuł: *Wyraz uczuć u człowieka i zwierząt*

ludzką. Na konferencji zaproponowano nazwę *sztuczna inteligencja* [3].

Koncepcja rozpoznawania obrazów przez komputery pojawiła się w latach 60. XX wieku. Trzydzieści lat później, wraz z większą dostępnością dużych baz danych, zaczęto intensywniej zgłębiać temat rozpoznawania obrazów, a także głosu. W latach 90. XX wieku rozpoczęto badania nad systemami umożliwiającymi rozpoznawanie emocji zarówno na podstawie wyrazu twarzy, jak i głosu. W próbkach głosu analizowano m.in. takie parametry, jak ton, tempo i intensywność mowy [4].

Na początku XXI wieku badacze z Uniwersytetu Południowej Kalifornii przeprowadzili ciekawy eksperyment dotyczący rozpoznawania emocji. Wyróżniono cztery emocje: smutek, radość, złość i stan neutralny. Najwyższą dokładność uzyskano dla smutku i radości (około 80%), natomiast dla złości i neutralnego głosu osiągnięto ponad 60% [5].

Przełomowym wydarzeniem w kontekście uczenia maszynowego i rozpoznawania emocji było wprowadzenie głębokiego uczenia, które zaczęto stosować już od lat 80. XX wieku. Zastosowanie modelu konwolucyjnego (CNN) znacząco zwiększyło precyzję rozpoznawania obrazów i wideo, o czym więcej w rozdziale 4. Przykładowo, w 2016 roku węgierscy badacze osiągnęli 95% dokładności w rozpoznawaniu emocji na podstawie mimiki twarzy w nagraniach filmowych[6][7].

Rozpoznawanie emocji za pomocą uczenia maszynowego ma długą historię i nieustannie się rozwija. Wraz z postępem technologii i udoskonaleniem algorytmów, dokładność rozpoznawania emocji rośnie, otwierając nowe perspektywy w tej dziedzinie. Poniżej zostały przedstawione dziedziny, w których *Emotion AI* cieszy się największym zainteresowaniem:

- psychologia
- bezpieczeństwo publiczne
- gry komputerowe
- interakcje człowiek-komputer

1. **Psychologia**

   Emotion AI znajduje zastosowanie w psychologii, wspierając diagnostykę i monitorowanie emocji pacjentów dzięki coraz szerszemu wykorzystaniu zaawansowanych technologii.

   Niestety, w dzisiejszych czasach coraz więcej osób cierpi na depresję. Szacuje się, że w Polsce dotyczy to około 1,2 miliona osób. Światowa Organizacja Zdrowia podaje,

że do roku 2030 depresja będzie najczęściej występującą chorobą na świecie[2]. Aby pomóc w walce z lękiem i samotnością, które są czynnikami wywołującymi stany depresyjne, powstaje coraz więcej aplikacji do monitorowania zdrowia psychicznego. Aplikacje te, poprzez szczegółową analizę, dostarczają wskazówek, jak dbać o swoje zdrowie psychiczne oraz przedstawiają czytelny obraz postępów użytkownika. Często są również wykorzystywane przez specjalistów.

Ciekawym przykładem aplikacji opartej na sztucznej inteligencji jest *Wysa: Anxiety, therapy chatbot*[3]. Została ona opracowana przez piętnastoosobowy zespół, w skład którego wchodzili psycholodzy, projektanci oraz programiści, a w głównej mierze opiera się na przetwarzaniu języka naturalnego. Twórcy stworzyli wirtualnego przyjaciela, który monitoruje zdrowie psychiczne użytkownika oraz dba o jego dobrą kondycję. Aplikacja ta oferuje szeroki zakres możliwości, od śledzenia nastroju po proponowanie praktyk medytacyjnych, które pomagają wyciszyć umysł i zadbać o jego zdrowie.

Drugim przykład aplikacji, która uwzględnia mowę użytkownika, jest apliakcja *VocalVue*[4], która w czasie rzeczywistym wykrywa emocje towarzyszące wypowiadanym słowom. Jest to rozwiązanie podobne do realizowanego projektu. Użytkownik dostarcza próbkę dźwięku, a aplikacja wykrywa dominującą emocję.

Innym zaskakującym przykładem wielowymiarowego wykorzystania *Emotion AI*, w tym rozpoznawania emocji w głosie, jest *SimSensei*. Projekt realizowany przez naukowców z Uniwersytetu Południowej Kalifornii stworzył wirtualnego terapeutę, który, wykorzystując przetwarzanie języka naturalnego oraz uczenie maszynowe, analizuje mowę, mimikę i gesty.
SimSensei prowadzi interakcje przypominające sesję terapeutyczną i dostarcza odpowiedzi zwrotne na temat emocji, co ma na celu lepsze samozrozumienie oraz wsparcie zdrowia psychicznego. W przyszłości projekt może przyczynić się do zwiększenia dostępności sesji terapeutycznych dla osób potrzebujących pomocy. Na dzień dzisiejszy projekt znajduje się w fazie rozwoju i badań, a jego zespół kontynuuje prace nad udoskonaleniem technologii i skuteczności interakcji [8].

Ponadto EAI pomaga dzieciom z zespołem Aspergera, które mają problem z rozpoznawaniem ludzkich emocji, poprzez specjalne aparaty słuchowe [5]. Urządzenia te analizują mimikę oraz ton głosu rozmówcy, co pomaga lepiej zrozumieć jego stan

[2] Dane dotyczące depresji w Polsce, według Narodowego Funduszu Zdrowia.
[3] Wysa: Anxiety, therapy chatbot, aplikacja dostępna w Google Play.
[4] VocalVue – Aplikacje w Google Play
[5] Aparaty słuchowe - projekt firmy Empath, o której więcej z następnym punkcie.

emocjonalny podczas rozmowy. Rozwiązanie to dodaje pewności siebie osobom z autyzmem oraz pozwala na trwalsze budowanie relacji.

Wraz ze wzrostem świadomości znaczenia zdrowia psychicznego, *Emotion AI* staje się coraz bardziej popularne. Nowoczesne aplikacje już dziś umożliwiają śledzenie kondycji emocjonalnej. W przyszłości niewykluczone, że będzie można korzystać z pomocy wirtualnych terapeutów bez konieczności długiego oczekiwania na specjalistyczną poradę.

2. **Interakcje człowiek-komputer**

W konsumpcyjnym świecie wielu firmom zależy na pozyskaniu jak największej liczby nowych klientów oraz utrzymaniu tych stałych. Algorytmy rozpoznawania emocji znalazły zastosowanie w call center, gdzie mierzone są parametry rozmowy telefonicznej, w tym rozpoznawanie emocji. Pozwala to na zapobieganie konfliktom i lepsze dostosowanie do potrzeb rozmówcy. Przykładem jest firma *Empath*, która obecnie posiada już 700 klientów w 50 krajach. Firma ta opracowała algorytmy, które pozwalają rozróżnić cztery emocje: radość, opanowanie, złość i smutek. Rozwiązanie zaproponowane przez firmę *Empath* nie analizuje treści wypowiadanych słów, ale skupia się na fizycznych właściwościach głosu[9].

Detekcja emocji zyskuje na popularności w nowoczesnych grach komputerowych, zwłaszcza dzięki wykorzystaniu technologii opartej na *Emotion AI*. Choć jeszcze nie jest szeroko rozpowszechniona, zdobywa rosnące zainteresowanie w projektach pilotażowych. Dzięki tej technologii, gry mogą reagować na emocje gracza w czasie rzeczywistym, dostosowując poziom trudności, narrację lub interakcję z postaciami w grze, co poprawia zaangażowanie i pozwala na bardziej personalizowane doświadczenie.
Dodatkowo EAI jest stopniowo wprowadzane do wirtualnych asystentów[6], gdzie analizowane są nie tylko treść komunikatu użytkownika, ale również zabarwienie emocjonalne wypowiedzi.

Podsumowując, w dziedzinie związanej z biznesem *Emotion AI* coraz częściej jest wdrażane do naszego codziennego życia. Istniejące rozwiązania, takie jak te proponowane przez firmę *Empath* [9], stają się coraz bardziej popularne, a projekty pilotażowe związane na przykład z przemysłem gier komputerowych, które obecnie są w fazie testów, w przyszłości mogą stać się rzeczywistością.

[6] Wirtualni asystenci - przykładem są : Google Assistant, Amazon Alexa czy Microsoft Cortana.

3. **Bezpieczeństwo publiczne**

   Dziedziną, w której sztuczna inteligencja emocjonalna prężnie się rozwija, jest bezpieczeństwo publiczne. Niestety, wiele krajów boryka się nadal z wysokim poziomem przestępczości, co można zaobserwować szczególnie w regionie Ameryki Łacińskiej. Chociaż wiele rozwiązań opartych na EAI znajduje się jeszcze w fazie testowej, powoli wdrażane są w komunikacji miejskiej i na lotniskach.

   Przykładem są systemy AI stosowane przez firmę Amazon w Wielkiej Brytanii. W ramach tych rozwiązań, emocje pasażerów są wychwytywane przez kamery CCTV, a zebrane dane przesyłane są do chmury. System ten jest obecnie używany na ośmiu stacjach kolejowych, co ma na celu zwiększenie bezpieczeństwa pasażerów oraz poprawę jakości obsługi[10].

   Dynamiczny rozwój projektów *Smart City* w Chinach poskutkował wzrostem zainteresowania *Emotion AI* w zakresie bezpieczeństwa publicznego. Aby zadbać o bezpieczeństwo obywateli i unikać niebezpiecznych sytuacji, w wielu miastach chińskich wprowadzono monitoring oparty na AI, służący do analizowania emocji przechodniów. Miliony kamer wspieranych przez technologie rozpoznawania twarzy umożliwiają identyfikację i szybkie reagowanie w sytuacjach zagrożenia[7] [11].

   W Polsce *Emotion AI* znalazło zastosowanie na Lotnisku Chopina w Warszawie oraz na lotnisku w Modlinie. Przy bramkach do odprawy pasażerów wykorzystywane są systemy rozpoznawania twarzy, które przyspieszają proces odprawy oraz zwiększają bezpieczeństwo.

   Te innowacje wskazują na rosnące znaczenie EAI w kontekście monitorowania i analizy zachowań, co może przyczynić się do szybszego reagowania na potencjalne zagrożenia.

Dynamiczny rozwój technologii sprawia, że trudno nadążyć za wprowadzanymi innowacjami, a *Emotion AI* już teraz odgrywa coraz większą rolę i może w przyszłości stać się integralnym elementem naszego życia. W przyszłości inteligentne systemy domowe mogłyby dostosowywać się do nastroju użytkowników, oferując spersonalizowane wsparcie. Szczególnie osoby starsze, zmagające się z samotnością, mogłyby skorzystać z wirtualnych asystentów zdolnych do rozpoznawania emocji na podstawie wyglądu, tonu głosu czy zachowania. Tego rodzaju technologia mogłaby pełnić funkcję nie tylko pomocnika, ale

[7] Więcej ciekawych informacji m.in. o wykorzytsaniu AI w Chinach można znaleźć na kanale *Wałkowanie Świata*, autorstwa Katarzyny Drelczuk

także towarzysza, wspierając użytkowników w codziennym życiu.

Inną dziedziną, w której *Emotion AI* może wprowadzić rewolucję, jest edukacja. Każdy uczeń jest inny, podobnie jak rośliny, które potrzebują różnej ilości wody i światła, tak samo uczniowie wymagają indywidualnego podejścia do nauki. Poprzez analizę emocji uczniów możliwe byłoby dostosowanie nauki, by stała się dla dzieci przyjemnością i dawała satysfakcję. Systemy oparte na AI mogłyby również pomóc nauczycielom w lepszym rozumieniu potrzeb uczniów i przygotowaniu odpowiednich materiałów. EAI mogłoby także rozwijać u dzieci świadomość emocji własnych oraz rówieśników.

Za kilka lat podobne zastosowania mogą stać się rzeczywistością także w środowisku pracy, gdzie każdy pracownik będzie traktowany indywidualnie, aby zwiększyć jego efektywność. Takie systemy mogłyby monitorować poziom stresu, zmęczenia lub frustracji, co pozwoliłoby na wczesną reakcję i dostosowanie warunków pracy.

Podsumowując, *Emotion AI* otwiera wiele dróg, stanowi szansę na rozwój ludzkości oraz lepsze rozumienie siebie i innych.

2. Uczenie maszynowe

Najprostszą definicją uczenia maszynowego jest uczenie się z danych. Algorytm, aby uzyskać wysoką wydajność potrzebuje ogromnej ilości danych. Jest on trenowany na zestawie uczącym (ang. *training set*), składający się z elementów nazywanych próbkami uczącymi, o czym więcej w rozdziale 3. Ważnym pojęciem jest również model, który jest częścią odpowiedzialną za uczenie, o czym więcej w rozdziale 4. Podsumowując, aby uzyskać sprawnie działający algorytm uczenia maszynowego potrzebujemy danych oraz modelu.

2.1. Rodzaje podejść w uczeniu maszynowym

Uczenie maszynowe można sklasyfikować pod wieloma kryteriami. Podstawowe kryteria to:

1. Sposób nadzorowania
    - Uczenie nadzorowane
    - Uczenie nienadzorowane
    - Uczenie półnadzorowane
    - Uczenie przez wzmacnianie
2. Możliowść uczenia w czasie rzeczywistym
    - Uczenie wsadowe
    - Uczenie przyrostowe
3. Zasada działania
    - Uczenie z przykładów
    - Uczenie z modelu

2.1.1. Sposób nadzorowania

Pierwszym kryterium jest sposób nadzorowania. Najpopularniejsze jest uczenie nadzorowane lub nienadzorowane. Dodatkowo istnieje również uczenie półnadzorowane oraz uczenie przez wzmacnianie.

Uczenie nadzorowane (ang. *supervised learning*) jest związane z pojęciem etykiet (ang. *labels*). Dane podlegają klasyfikacji (ang. *classification*) czyli podporządkowaniu do określonej klasy. Typowym zabiegiem w uczeniu nadzorowanym jest regresja, czyli przewidywanie wartości na podstawie cech wejściowych. W praktyce dana próbka może zostać podporządkowano do kilku docelowych wartości (ang. *target*).

Drugim typem jest **uczenie nienadzorowane** (ang. *unsupervised learning*), w którym dane nie posiadają etykiet. W tym typie uczenia korzysta się z analizy skupień. Dane łączą się w grupy o podobnych cechach. W uczeniu nienadzorowanym używa się również

algorytmów wizualizujących, gdzie dane stanowią reprezentacje określonych punktów na wykresie, dzięki tej metodzie można w prosty sposób odkryć cechy charakterystyczne dla danego obszaru. Cechą charakterystyczną uczenia nienadzorowanego jest łatwe wykrywanie anomalii (ang. *anomaly detection*), czyli znacznych odchyleń punktów od danego obszaru skupienia. Może być to szczególnie istotne w wykrywaniu usterek lub nieprawidłowego działania danego systemu.

Uczenie półnadzorowane (ang. *semisupervised learning*) jest podejściem hybrydowym, który łączy uczenie nadzorowane z nienadzorowanym. W rozwiązaniu tym, tylko część danych posiada etykiety, pozostałe dane są nieoznakowane. Podejście to pozwala zaoszczędzić czas i zasoby. Na początku dane ulegają podziałowi na grupy. Każdy element danej grupy zostaje oznaczony etykietą, która najczęściej pojawia się w tej grupie.

Na koniec warto wspomnieć o **uczeniu przez wzmacnianie** (ang. *reinforcement learning*). Jest to system różniący się od wymienionych powyżej. W uczeniu przez wzmacnianie wyróżniamy agenta oraz nagrody i kary. System uczący zwany agentem za wykonywane zadania odbiera nagrody (zazwyczaj punkty dodatnie) lub kary (punkty ujemne). W ten sposób agent uczy się strategii, zwanej polityką, aby zdobywać jak najwięcej punktów dodatnich. Przykładowo uczenie przez wzmacnianie jest wykorzystywane w grach komputerowych, takich jak szachy.

2.1.2. Możliowść uczenia w czasie rzeczywistym

Innym kryterium podziału systemów uczenia maszynowego jest możliwość uczenia modelu w czasie rzeczywistym. Wyróżniamy podział na uczenie przyrostowe i uczenie wsadowe.

Pierwsze z wymienionych **uczenie wsadowe** (ang. *batch learning*) które wiąże się z uczeniem offline (ang. *offline learning*), polega na wytrenowaniu modelu na dostępnych danych. Podejście to sprawdza się na danych, które nie ulegają szybkim zmianom, dlatego nie sprawdzi się do analizy danych pogodowych czy danych dotyczących ruchu drogowego. Jeśli zaistnieje potrzeba zaktualizowania systemu o nowe dane należy proces nauki zacząć od nowa, uwzględniając zarówno nowe jak i stare dane.

Częściej stosuje się **uczenie przyrostowe** (ang. *online learning*) gdzie system uczy się w czasie rzeczywistym, wraz z dostarczaniem nowych porcji danych nazywanych minipakietami (ang. *mini-batches*).

Uczenie przyrostowe charakteryzuje się większą podatnością na błędy, dlatego ważne jest zadbanie o dobrą jakość i adekwatność dostarczonych danych.

2.1.3. Zasada działania

Kolejnym kryterium podziału uczenia maszynowego jest sposób pracy, gdzie można wyróżnić uczenie z przykładów bądź uczenie z modelu.

W **uczeniu z przykładów** (ang. *instance-based learning*) niezbędne jest uwzględnienie miary podobieństwa między nowym elementem a elementem wcześniej zapamiętanym. Miara ta określa na ile dwa elementy są do siebie podobne, aby sklasyfikować je do tej samej grupy.

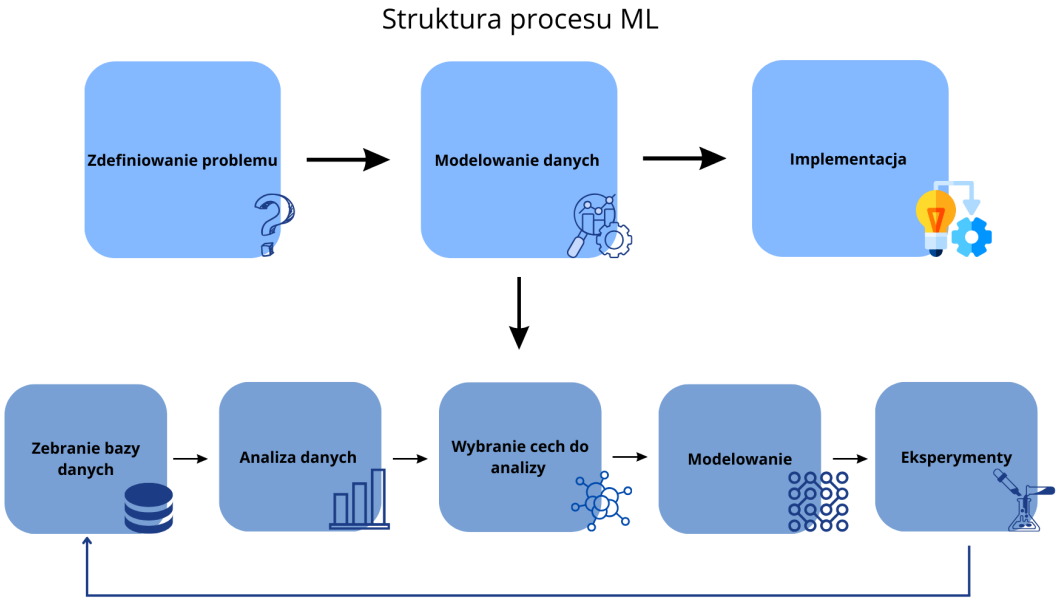
Uczenie z modelu (ang. *model-based learning*) to metoda, która polega na tworzeniu modelu na podstawie dostępnych danych. Model ma na celu przewidywanie wyników w przyszłości. Warto podkreślić, że w tym kontekście *model* nie odnosi się do konkretnego typu modelu, ale do procesu dobierania odpowiednich parametrów, które sprawiają, że model działa lepiej.

Aby ocenić, jak dobry jest stworzony model, używa się tzw. funkcji użyteczności, znanej także jako funkcja dopasowania. Funkcja ta pomaga zmierzyć skuteczność modelu - miarę efektywności. Na przykład w regresji liniowej używa się funkcji kosztu, która stara się zminimalizować różnicę między wartościami rzeczywistymi a tymi przewidywanymi przez model. Im mniejsza ta różnica, tym lepszy model[6].

2.2. Uczenie maszynowe w praktyce

Poprzez zgłębianie wiedzy na temat uczenia maszynowego dąży się do praktycznego wykorzystania zdobytych informacji. Wytrenowane modele, które potrafią wydobywać wzorce i klasyfikować nowe dane, są często wykorzystywane w aplikacjach w celu usprawnienia ich działania lub dostarczania użytkownikom konkretnych informacji. Tworzenie mniejszych modeli, takich jak te rozpoznające emocje na podstawie mimiki twarzy, głosu czy gestykulacji, może finalnie prowadzić do stworzenia kompleksowego rozwiązania, takiego jak *SimSensei* — wirtualnej terapeutki, o której wspomniano we wstępie pracy. Na rysunku 2.1 przedstawiono schemat ideowy[8], który obrazuje kolejne etapy realizacji procesu uczenia maszynowego, prowadzące do stworzenia funkcjonalnego rozwiązania[12].

[8] Wszytskie schematy zawarte w pracy zostały stworzone przy pomocy platformy Canva.



Rysunek 2.1. Struktura procesu ML

Realizacja uczenia maszynowego w praktyce obejmuje kilka kluczowych faz, począwszy od zdefiniowania problemu, aż po implementację, czyli wykorzystanie stworzonego rozwiązania w rzeczywistych zastosowaniach, np. w aplikacjach. Po ustaleniu specyfiki zadania, określeniu celów oraz kroków milowych, przechodzi się do etapu modelowania danych, który można podzielić na następujące fazy:

1. **Wybór danych**

Pierwszym krokiem w modelowaniu danych jest wybór odpowiedniej bazy danych. Jest to bardzo ważny etap, ponieważ jakość i adekwatność danych w znacznym stopniu wpływają na skuteczność modelu, który będzie na nich trenowany.

2. **Analiza danych**

Dostarczone dane są dokładnie analizowane, aby ocenić ich spójność, jakość, kompletność i podobieństwo. Celem tej analizy jest przygotowanie danych do dalszego przetwarzania.

3. **Wybór cech**

Po wstępnej analizie dokonuje się wyboru cech, na podstawie których model będzie uczył się wyodrębniać wzorce. Ten etap, zwany inżynierią cech, często obejmuje również tworzenie nowych zmiennych. Szczegółowe omówienie pierwszych trzech kroków procesu uczenia maszynowego zostanie przedstawione w rozdziale 3.

4. **Modelowanie**
Wybór odpowiedniego modelu uczenia maszynowego jest dostosowywany do specyfiki zadania oraz charakterystyki danych. Model ten może opierać się na różnych algorytmach, takich jak regresja, drzewa decyzyjne, sieci neuronowe itp. Szczegółowe omówienie procesu modelowania zostanie przedstawione w rozdziale 4.

5. **Eksperymentowanie**
W tej fazie model jest trenowany i testowany na wybranych danych. Eksperymenty pozwalają ocenić skuteczność modelu i identyfikować obszary wymagające poprawy. Wyniki eksperymentów służą do iteracyjnych modyfikacji wcześniejszych etapów, takich jak wybór cech czy przygotowanie danych.

Modelowanie danych to proces iteracyjny, w którym na podstawie wyników eksperymentów powraca się do wcześniejszych etapów w celu ich ulepszenia. Wprowadzone zmiany mogą obejmować np. korektę danych, dobór nowych cech lub zmianę algorytmu modelu.

Kiedy wyniki modelu są zadowalające, stworzone rozwiązanie może zostać wdrożone do rzeczywistego użytkowania. Wdrożenie może polegać na zintegrowaniu modelu z aplikacją lub innym systemem, gdzie będzie on wykorzystywany do podejmowania decyzji lub automatyzacji procesów.

2.3. Problem generalizacji

Generalizacja to zdolność modelu do poprawnego przewidywania wyników dla nowych, wcześniej niewidzianych danych. Aby model mógł skutecznie radzić sobie z danymi spoza zbioru treningowego, konieczne jest zapobieganie dwóm typowym zjawiskom występującym w uczeniu maszynowym, jakimi są **przeuczenie** (ang. *overfitting*) oraz **niedouczenie** (ang. *underfitting*).

2.3.1. Przeuczenie

Zjawisko nadmiernego dopasowania lub przetrenowania (ang. *overfitting*) występuję w przypadku zbyt złożonego modelu w porównaniu do ilości dostarczonych danych. Aby zapobiec zjawisku zbyt nadmiernego dopasowania danych do modelu należy w pierwszej kolejności postarać się uprościć model co zwie się regulacją (ang. *regularization*). Stopień regulacji kontroluje się poprzez dopasowanie hiperparmentrów czyli parametrów algorytmu uczącego. Hiperparametry wyznacza się przed procesem nauki modelu i nie ulegają one zmianom w trakcie uczenia. Innym rozwiązaniem jest dostarczenie większej ilości danych. Aby model jak najlepiej przewidywał przyszłe wyniki, istotne jest zachowanie

równowagi, czyli uzyskanie balansu między idealnym dopasowaniem do danych a prostotą modelu, co pozwoli na lepszą generalizację wyników w przyszłości.

2.3.2. Niedouczenie

Zjawiskiem przeciwstawnym jest niedotrenowanie (ang. *underfitting*) gdzie model jest zbyt prosty, aby uchwycić złożoność danych. Aby zapobiec temu zjawisko należy postarać się o bardziej złożony model, z większą liczbą parametrów. Dodatkowo można zadbać o dostarczenie większej ilości atrybutów w procesie inżynierii cech[6].

2.4. Podumowanie rozdziału

W rozdziale poświęconym uczeniu maszynowemu omówiono podstawowe pojęcia związane z ML, takie jak uczenie nadzorowane, nienadzorowane, przyrostowe oraz wsadowe. Przedstawiono również kluczowe kroki realizacji projektu ML w praktyce, co pozwoliło uzyskać ogólny wgląd w ideę zastosowania ML w rzeczywistych projektach. Na zakończenie poruszono dwa istotne problemy związane z uczeniem maszynowym: przeuczenie i niedouczenie.

3. Dane

3.1. Baza danych

Dostarczenie dużej ilości danych wysokiej jakości jest istotnym elementem uczenia maszynowego. W pierwszej kolejności należy dokładnie przeanalizować dane, aby upewnić się, że są one spójne z założeniami projektu. Warto oczyścić wybraną bazę danych z próbek, które zawierają braki lub znacznie odbiegają od pozostałych – takie próbki mogą negatywnie wpłynąć na trening modelu, generując zakłócenia i anomalie.

Porównując ludzki mózg do systemów uczących się, warto zwrócić uwagę, że człowiek potrzebuje jedynie jednej próbki, aby rozpoznać ją ponownie, podczas gdy model uczenia maszynowego wymaga setek próbek oraz wielu godzin treningu. Na przykład, aby człowiek ponownie rozpoznał twarz autora książki *The Expression of the Emotions in Man and Animals* – Charlesa Darwina, wystarczy jedno zdjęcie. Tymczasem model rozpoznawania obrazów potrzebowałby setek podobnych zdjęć, aby w przyszłości zidentyfikować tę osobę.

W kontekście analizy dźwięku należy zadbać o dostarczenie próbek najwyższej jakości, nagranych za pomocą profesjonalnego sprzętu w studyjnych warunkach. Równie ważna jest spójność danych – warto, aby wszystkie nagrania były wykonane w tym samym języku, z jednakową częstotliwością próbkowania oraz miały zbliżoną długość. Wszystkie te kryteria spełnia baza *Emotional Speech Database*, wykorzystana w niniejszym projekcie.

3.1.1. Baza danych ESD

Baza danych *Emotional Speech Database* składa się z 35 000 próbek, z czego połowa to próbki w języku angielskim, a druga połowa w języku mandaryńskim. Próbki obejmują pięć podstawowych emocji:

1. Radość
2. Złość
3. Smutek
4. Zaskoczenie
5. Neutralność

W realizacji projektu tworzenia bazy ESD wzięło udział 20 aktorów[9]. Każdy z aktorów został nagrany 350 razy dla każdej z emocji[10]. Treść wypowiadanych przez aktorów zdań jest przypadkowa, a zdania z określonym ładunkiem emocjonalnym są nagrane dla wszyst-

[9] Dwudziestu aktorów: 10 rodzimych użytkowników języka angielskiego i 10 rodzimych użytkowników języka mandaryńskiego.

[10] Łącznie na każdego aktora przypada 1750 próbek.

kich emocji. Przykładowo, zdanie: *I'm as bad as I can be* [11] zostało nagrane zarówno dla emocji radość, jak i złość.

Cechy bazy danych:

1. **Różnorodność językowa:**
   Jak wspomniano wcześniej, baza danych ESD jest wielojęzyczna, ponieważ zawiera próbki zarówno w języku angielskim, jak i mandaryńskim. Każdy język charakteryzuje się spójnością – wszystkie próbki w języku angielskim są w północnoamerykańskim angielskim, a próbki w języku chińskim to standardowy mandaryński. Zachowanie akcentów i niuansów językowych pozwala na uniknięcie zakłóceń wynikających z różnic dialektalnych[12].

2. **Różnorodność aktorów:**
   W obu grupach językowych zachowano różnorodność pod względem płci i wieku. Baza zawiera głosy zarówno żeńskie, jak i męskie, a aktorzy są w wieku od 25 do 35 lat. Nagrania nie zawierają elementów takich jak śmiech czy westchnienia, które mogłyby jednoznacznie wskazywać na określoną emocję.

3. **Środowisko nagraniowe:**
   Próbki zostały nagrane w warunkach studyjnych, co zapewniło wysoką jakość dźwięku[13]. Każda próbka została nagrana z częstotliwością próbkowania **16 kHz**.

4. **Organizacja struktury katalogów:**
   W głównym folderze bazy ESD każdy aktor posiada odrębny folder, którego numeracja zaczyna się od 0001, a kończy na 0020. W każdym z tych folderów znajdują się 5 podfolderów: *Angry*, *Happy*, *Sad*, *Neutral* oraz *Surprise*, w których przechowywane są próbki dźwiękowe odpowiadające poszczególnym emocjom. Autor bazy danych proponuje także podział na zbiory: treningowy, walidacyjny oraz testowy[14].

5. **Czas trwania:**
   Średnia długość wypowiedzi w bazie danych wynosi 2,76 s dla języka angielskiego i 3,22 s dla języka chińskiego. Frazy angielskie składają się średnio z 6,31 słów, a chińskie z 11,5 znaków, co odpowiada typowym zwrotom w komunikacji codziennej.

[11] ang. *Jestem tak zły, jak tylko mogę być.*
[12] W projekcie zostały wykorzystane tylko próbki w języku angielskim
[13] SNR (stosunek sygnału do szumu) powyżej 20 dB
[14] W realizowanym projekcie zastosowano podział zaproponowany przez autorów bazy danych ESD; dane zostały rozdzielone w proporcji 30/2/3.

6. **Zmienność leksykalna:**
   Dane z ESD zapewniają wysoką różnorodność leksykalną. Przykładowe zdania to:
   a) Neutralność: *That was his chief thought.*[15] (0020[16]_000358[17])
   b) Złość: *How I hate this foul pool!* [18] (0020_000386)
   c) Radość: *I think it'll encourage me.*[19] (0020_001083)
   d) Smutek: *This used to be Jerry's occupation.*[20] (0020_001083)
   e) Zakoczenie: *Tom now let our arrows fly!*[21] (0020_001470)

ESD jest jedną z największych dostępnych publicznie baz danych mowy emocjonalnej, wyróżniającą się szerokim zakresem leksykalnym, różnorodnością językową i mówców oraz kontrolą jakości nagrań. W realizowanym projekcie wykorzystano jedynie próbki w języku angielskim [13][14].

3.2. Zestawy danych

W uczeniu maszynowym najczęściej spotyka się podział danych na 3 zbiory:

1. **Zbiór trenigowy**(ang. *training set*)
   Zbiór treningowy, zwany także uczącym, służy do nauki zależności między danymi przez model. Model analizuje wzorce i struktury w danych, po czym dostosowuje swoje parametry wewnętrzne, aby poprawnie przewidywać wyniki. Zbiór treningowy jest największym zbiorem danych i często poddawany jest technikom augmentacji danych.

2. **Zbiór walidacyjny**(ang. *validation set*)
   Zbiór walidacyjny jest wykorzystywany podczas treningu modelu, ale sam model nie uczy się na próbkach z tego zbioru. Próbki są dla modelu nowe i służą do oceny jego jakości w trakcie trenowania. Wyniki uzyskane ze zbioru walidacyjnego wskazują, w jakim kierunku dostroić parametry, takie jak liczba warstw czy współczynnik uczenia. Zbiór walidacyjny pozwala również monitorować zjawiska, takie jak przeuczenie i niedouczenie modelu, co zostało omówione w rozdziale poświęconym ML.

3. **Zbiór testowy**(ang. *test set*)
   Zbiór testowy, niezależny od dwóch pozostałych, służy do końcowej oceny modelu. Jest wykorzystywany na danych, które są nieznane dla modelu, co pozwala obiektyw-

[15] pol. *To było jego główne myślenie.*
[16] nazwa folderu
[17] nazwa pliku
[18] pol. *Jak nienawidzę tego paskudnego stawu!*
[19] pol. *Myślę, że to mnie zachęci.*
[20] pol. *To było kiedyś zajęcie Jerry'ego.*
[21] pol. *Tom teraz niech wypuści nasze strzały!*

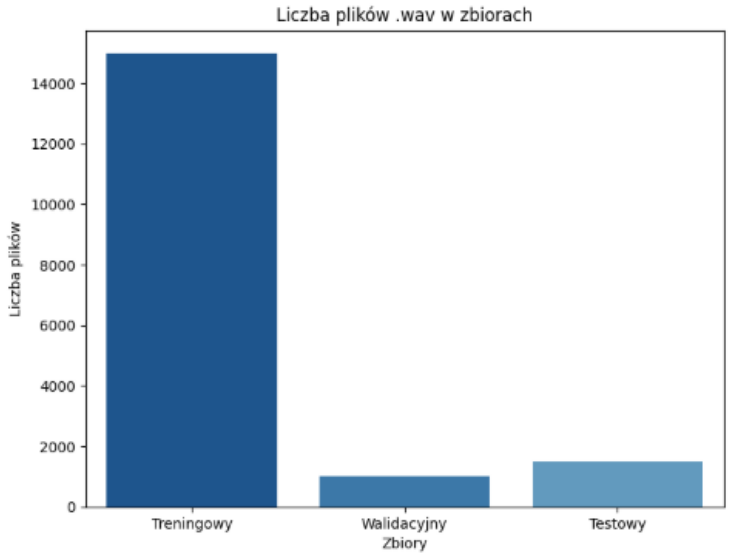
nie ocenić jego jakość. Wyniki uzyskane na zbiorze testowym pokazują, jak dobrze model generalizuje na nowych próbkach i jak skutecznie radzi sobie z danymi, które nie były używane podczas treningu ani walidacji[6].

Podział danych na zbiory treningowy, walidacyjny i testowy często odbywa się w proporcjach 80/10/10 lub 75/15/15. W realizowanym projekcie zastosowano jednak podział zaproponowany przez autora bazy danych, ponieważ po licznych próbach okazał się on najbardziej efektywny. Warto podkreślić, że wybór proporcji zależy od specyfiki realizowanego zadania oraz charakterystyki dostępnych danych. W przypadku bardzo dużych zbiorów, takich jak *Big Data*, na potrzeby zbioru testowego wystarczy przeznaczyć zaledwie 1–5% danych.

Tabela 3.1 oraz rysunek 3.1 przedstawiają rozkład ilościowy próbek dźwiękowych w poszczególnych zbiorach.

Tabela 3.1. Tabela z liczbą plików w zbiorach

Zbiór	**Liczba plików**
Treningowy	15 000
Walidacyjny	1 000
Testowy	1 500
Łącznie	**17 500**



Rysunek 3.1. Wykres słupkowy rozkładu plików .wav w poszczególnych zbiorach

Wykres został stworzony i jest dostępny w notebooku znajdującym się w środowisku *GitLab*[15]. Wykres przedstawia liczbę próbek w poszczególnych zbiorach: treningowym, walidacyjnym oraz testowym.

3.3. Wyodrębnanie cech

Po analizie zbioru danych kolejnym etapem jest wybór cech, które umożliwią przypisanie próbek do odpowiednich klas. Dobór tych cech zależy zarówno od charakteru danych, jak i specyfiki rozwiązywanego problemu. Na przykład zestaw cech potrzebnych do rozpoznawania płci głosu (męski lub żeński) różni się od tych wykorzystywanych w klasyfikacji dźwięków na emocje. W projekcie analizowane są sygnały czasowe, które są dzielone na ramki. W każdej ramce badane są cechy, takie jak:

1. **Współczynnik przejść przez zero (ZCR)**
   Do analizy dynamiki dźwięku wykorzystano współczynnik ZCR (ang. *Zero Crossing Rate*) [16], który mierzy liczbę zmian znaku sygnału w zdefiniowanej ramce czasowej. Współczynnik ten jest obliczany na podstawie poniższego wzoru:

$$\mathbf{ZCR}[\mathbf{k}] = \frac{1}{2N} \sum_{n=1}^{N-1} \left| \operatorname{sgn}(x_k[n]) - \operatorname{sgn}(x_k[n-1]) \right| \tag{1}$$

   gdzie:
   - k – numer ramki sygnału,
   - N – długość ramki sygnału,
   - $x_k[n]$ – wartość k-tej ramki sygnału w chwili n,
   - $\operatorname{sgn}(x)$ – funkcja signum.

   Funkcja signum ($\operatorname{sgn}(x)$) jest zdefiniowana jako:

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{dla } x > 0, \\ 0 & \text{dla } x = 0, \\ -1 & \text{dla } x < 0. \end{cases} \tag{2}$$

   Funkcja signum zwraca znak liczby x:
   - Wartość 1, gdy x jest dodatnia,
   - Wartość 0, gdy x jest równa zero,
   - Wartość -1, gdy x jest ujemna.

   Współczynnik ZCR przyjmuje wartości liczbowe w przedziale [0,1], gdzie 0 oznacza brak zmian, a 1 oznacza maksymalną liczbę zmian znaku w sygnale, co odpowiada sytuacji, w której każda próbka sygnału zmienia znak w stosunku do poprzedniej. Funkcja została zrealizowana przy pomocy *librosa.feature.zero_crossing_rate*[17].

2. **Średnia kwadratowa energia (RMS)**
   Drugim analizowanym parametrem dla próbek dźwiękowych był współczynnik RMS

(ang. Root Mean Square) [18], który określa średnią energię sygnału w poszczególnych ramkach czasowych. Parametr ten pozwala mierzyć zmiany energii sygnału w czasie, co umożliwia ocenę różnic w intensywności dźwięku dla poszczególnych emocji.

$$\text{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2} \tag{3}$$

Wyjaśnienie wzoru:

- $x[n]$: Wartości amplitudy sygnału w ramce czasowej n.
- N: Liczba próbek w ramce.
- $\sum_{n=1}^{N} x[n]^2$: Suma kwadratów wszystkich próbek w ramce.

Wyodrębnie współczynnika RMS zrealiozwano przy pomocy *librosa.feature.rms*[17].

3. **Współczynniki cepstralne Mel (MFCC)**
   Funkcja *librosa.feature.mfcc* [17] służy do obliczania MFCC (ang. *Mel-Frequency Cepstral Coefficients*). Nazwa pochodzi od zastosowania transformaty Fouriera, która analizuje widmo sygnału. Proces wyznaczania współczynników MFCC rozpoczyna się od filtracji preemfazy, która wzmacnia wyższe częstotliwości, redukując wpływ niskich częstotliwości. Następnie sygnał audio jest dzielony na ramki, z których każda jest analizowana osobno.
   Dzięki zastosowaniu dyskretnej transformaty Fouriera (DFT) sygnał przechodzi z dziedziny czasu do dziedziny częstotliwości, co umożliwia identyfikację dominujących częstotliwości w każdej ramce dźwięku. Po uzyskaniu widma dźwięku stosuje się bank filtrów Mel (o charakterystyce trójkątnej), który odwzorowuje ludzkie postrzeganie dźwięku. Skala Mel charakteryzuje się większą wrażliwością na różnice w niższych częstotliwościach i mniejszą w wyższych, co odpowiada naturalnemu działaniu ludzkiego słuchu.
   Następnie oblicza się logarytm energii w pasmach częstotliwości, co sprawia, że współczynniki stają się mniej podatne na szumy. Ostatnim krokiem jest zastosowanie dyskretnej transformaty kosinusowej (DCT) do logarytmów energii. Pozwala to na uzyskanie współczynników MFCC (zwykle od 12 do 20), które określają, jak bardzo określony zakres częstotliwości jest obecny w dźwięku [18].

4. **Szerokość pasma widmowego**
   Przy użyciu funkcji bazującej na *librosa.feature.spectral_bandwidth* [17] badano, jak szeroko rozłożone są częstotliwości w sygnale wokół jego środka ciężkości (centroidu). Funkcja ta analizowała stopień zróżnicowania dźwięku. Bardziej dynamiczne emocje

(np. złość) charakteryzują się szerszym widmem, co skutkuje wyższymi wartościami.

5. **Spektralny roll-off**
   Funkcja *librosa.feature.spectral_rolloff* [17] mierzy spektralny *roll-off*, czyli próg częstotliwości, poniżej którego znajduje się określony procent energii widma (domyślnie 85%). Umożliwia ocenę koncentracji energii sygnału w niższych częstotliwościach. Jeśli większość energii jest skumulowana w niższych częstotliwościach, wartość roll-off będzie niska.

6. **Wysokość tonu F0**
   Funkcja *librosa.core.piptrack*[22] służy do obliczania częstotliwości podstawowej (ang. *Fundamental Frequency, F0*). W widmie sygnału częstotliwość podstawowa to najniższa częstotliwość, w której sygnał powtarza się w sposób okresowy. Wyższa wartość F0 jest charakterystyczna dla dynamicznych emocji (np. radości, zdziwienia), natomiast niższa dla spokojniejszych emocji (np. smutku, neutralności).

7. **Energia**
   Przy użyciu biblioteki *NumPy* zaimplementowano funkcję *calculate_energy*, która umożliwia wyodrębnianie energii sygnału w kolejnych ramkach czasowych. Funkcja ta analizuje intensywność dźwięku w czasie, obliczając energię jako sumę kwadratów amplitud w każdej ramce.

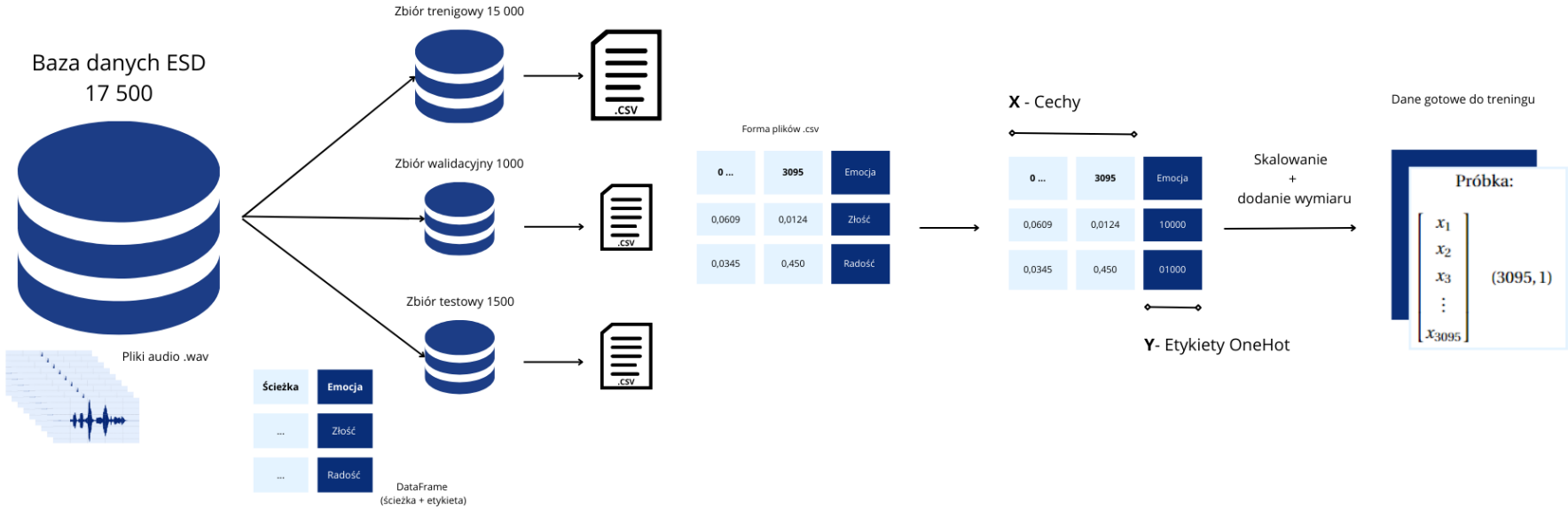
8. **Tempo**
   Przy użyciu funkcji *librosa.feature.tempo* zmierzono globalne tempo dla całego sygnału[17].

Na koniec, za pomocą funkcji *extract_audio_features*, wszystkie opisane wcześniej cechy zostały spłaszczone i połączone w jeden wektor, w którym każda cecha stanowi zestaw wartości. Taka integracja cech pozwoliła uzyskać bogatą reprezentację właściwości pojedynczej próbki dźwiękowej, co umożliwia klasyfikację emocji.

3.4. Proces przygotowania danych do modelowania

Model uczenia maszynowego wymaga odpowiednio przetworzonych danych wejściowych. Proces ten jest złożony i składa się z wielu etapów. Schemat 3.2 ilustruje kolejne fazy przetwarzania próbki dźwiękowej w formacie *.wav* na wektor cech o kształcie (3095,1), który zostanie użyty w procesie treningu modelu[12].

[22] Funkcja do obliczania wysokości tonu została zaimplementowana z wykorzystaniem biblioteki *librosa*.

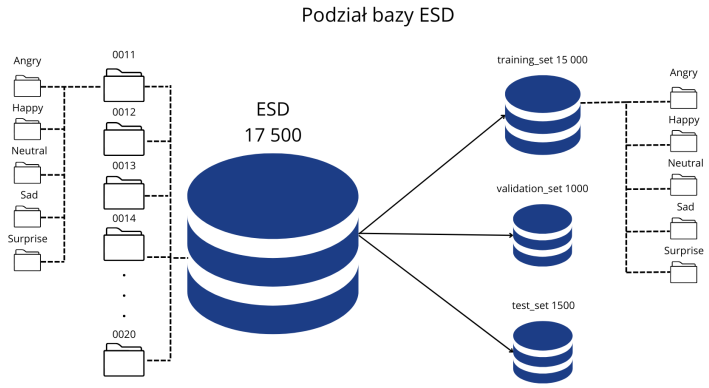


Rysunek 3.2. Proces przygotowania danych do modelowania

Schemat został stworzony przy pomocy platformy *Canva*[12]. Przedstawia kolejne fazy przetwarzania pliku dźwiękowego aż do uzyskania wektora składającego się z zestawu cech pozwalających sklasyfikować emocje.

Fazy przetwarzania próbki dźwiękowej:

1. **Podział głównej bazy danych**
   Baza danych ESD, składająca się z 17 500 próbek audio *.wav*, została podzielona na 3 podzbiory: zbiór treningowy (15 000 próbek), zbiór walidacyjny (1000 próbek) oraz zbiór testowy (1500 próbek). Podział został wykonany automatycznie w sposób kontrolowany przy użyciu języka Python, gdzie pierwsze 20 próbek każdego folderu zostało przeniesione do zbioru walidacyjnego, kolejne 30 próbek do zbioru testowego, a pozostałe do zbioru treningowego.
   W efekcie powstały 3 foldery (każdy zawiera 5 podfolderów odpowiadających poszczególnym emocjom) nazwane: *test_set*, *training_set*, *validation_set*[23]. Podział próbek dźwiękowych na 3 zbiory został przedstawiony na rysunku 3.3.



Rysunek 3.3. Schemat organizacji folderów.

Baza ESD składa się z 10 folderów, z których każdy reprezentuje jednego aktora. Każdy folder aktora zawiera 5 podfolderów odpowiadających poszczególnym emocjom. Baza została podzielona na trzy mniejsze zbiory: treningowy, walidacyjny i testowy. Każdy z tych zbiorów również składa się z 5 folderów przechowujących pliki *.wav* przypisane do odpowiednich emocji. Nazwy folderów poszczególnych emocji zachowały swoje oryginalne angielskie nazewnictwo od rodzica - bazy ESD. Schemat został stworzony przy pomocy platformy *Canva*[12].

2. **Przekształcanie folderów w *DataFrame***
   Przy użyciu biblioteki *pandas*, podzbiory w formie folderów zostają przekształcone do struktury tabelarycznej *DataFrame*, zawierającej dwie kolumny: ścieżkę do pliku oraz etykietę emocji.

3. **Przetworzenie zbiorów w pliki *.csv***
   Po wyodrębnieniu cech i ich zapisaniu, zebrane cechy dla plików audio zostają zapisane w plikach *.csv*. Powstają trzy pliki o nazwach: *train_emotions.csv, validation_emotions.csv, test_emotions.csv*. Pliki te zawierają próbki, które po przetworzeniu są ciągami liczb reprezentującymi wyodrębnione charakterystyczne cechy emocji. Łącznie dla każdego pliku wyodrębniono 3096 cech.

[23] Nazwy folderów są w języku angielskim, ponieważ całość nazewnictwa zmiennych w kodzie Python jest w języku angielskim.

Ponieważ pliki dźwiękowe były zróżnicowane, niezbędne było zastosowanie operacji wypełnienia brakujących wartości, gdzie wartości *NaN* zostały zastąpione zerami. Ta operacja pozwoliła zachować spójność danych niezbędną do prawidłowego trenowania modelu uczenia maszynowego, minimalizując ryzyko błędów wynikających z niekompletnych danych. Dzięki temu dane są jednolite i gotowe do dalszych etapów analizy i modelowania.

4. **Przekształcenie danych w zbiory: *X* i *Y***
   Omówione w poprzednim kroku pliki *.csv* wciąż zawierały etykiety w formie tekstowej, co wymagało przeprowadzenia operacji kodowania tych etykiet na reprezentację numeryczną. Kodowanie zostało zrealizowane za pomocą metody *OneHot*, która przekształca etykiety klas na wektory binarne, gdzie każda klasa jest reprezentowana jako wektor o długości równej liczbie wszystkich klas, zawierający wartość 1 na pozycji odpowiadającej danej klasie i 0 w pozostałych pozycjach.
   Etykiety, zakodowane w ten sposób, utworzyły zbiór *Y*, podczas gdy wyodrębnione cechy z próbek audio zostały przypisane do zbioru *X*. Następnie cechy zostały poddane procesowi skalowania za pomocą klasy *StandardScaler*, który przekształca dane w taki sposób, aby każda cecha miała średnią równą 0 oraz odchylenie standardowe równe 1. Skalowanie jest ważne, ponieważ zapewnia spójność danych wejściowych, co poprawia stabilność i wydajność modelu uczenia maszynowego, jednocześnie redukując wpływ różnic w wartościach numerycznych między cechami.
   Dodatkowo, do zbioru cech *X* został dodany dodatkowy wymiar za pomocą funkcji *np.expand_dims*, która umożliwia rozszerzenie kształtu danych poprzez dodanie nowej osi. Operacja ta była niezbędna, aby dostosować dane do wymaganego formatu wejściowego dla sieci konwolucyjnych (CNN), które oczekują tensorów wielowymiarowych o kształcie *(liczba próbek, liczba cech, liczba kanałów)*. W tym przypadku dodatkowy wymiar odpowiada liczbie kanałów, wynoszącej 1, co jest standardowe dla danych jednowymiarowych, takich jak cechy wyodrębnione z plików audio. Dzięki temu dane zostały odpowiednio sformatowane i przygotowane do dalszego etapu, jakim jest trenowanie modelu CNN.

Poniżej przedstawiono organizację pojedynczej próbki po przetworzeniu:

Próbka:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{3095} \end{bmatrix} \quad (3095, 1)$$

Objaśnienia oznaczeń:

- x_i – pojedyncza wartość cechy i-tej wyodrębniona z pliku audio, gdzie $i \in [1,3095]$.
- 3095 – liczba cech wyodrębnionych dla każdej próbki.
- 1 – liczba kanałów, charakterystyczna dla danych jednowymiarowych (np. cech akustycznych).
- **Kształt próbki:** $(3095,1)$ – każda próbka jest macierzą kolumnową zawierającą wyodrębnione cechy.

3.5. Podsumowanie rozdziału

W rozdziale poświęconym danym przedstawiono charakterystykę wybranej bazy danych ESD oraz jej najważniejsze cechy. W pracy inżynierskiej wykorzystano połowę bazy danych ESD zawierającą próbki w języku angielskim. Podział na zbiory treningowy, walidacyjny oraz testowy został zachowany zgodnie z propozycją autorów i wynosił odpowiednio 30:2:3. Ponadto przedstawiono krótką charakterystykę analizowanych cech dźwiękowych a na zakończenie opisano cały proces przetwarzania danych, od surowych próbek w formacie *.wav* aż po uzyskanie wektorów reprezentujących zestaw cech dla poszczególnych ramek.

4. Modele

W dniu dzisiejszym wykorzystuje się różne modele w uczeniu maszynowym. Wybór odpowiedniego modelu zależy od specyfiki zadania, rodzaju analizowanych danych oraz wcześniej założonych celów. Przed erą uczenia głębokiego i sztucznej inteligencji podstawę stanowiły techniki statystyczne, które operowały na małych zbiorach danych w formie tabelarycznej. Do przykładów należą:

K-Nearest Neighbors (KNN): Metoda stosowana do regresji i klasyfikacji. Do podstawowych parametrów należą:

- **Odległość** – wyliczana za pomocą metryki Euklidesowej lub Manhattan,
- **Liczba sąsiadów** (k) – liczba najbliższych punktów używana do przewidywania.

Drzewa decyzyjne (ang. Decision Trees): Podejście stosowane zarówno do klasyfikacji, jak i regresji. Nazwa modelu wiąże się ze strukturą drzewa, gdzie:

- **Każdy węzeł** reprezentuje pytanie o cechę danych,
- **Gałęzie** odpowiadają możliwym odpowiedziom,
- **Liście** zawierają przewidywania.

Drzewa decyzyjne są intuicyjne, ale mogą być podatne na przeuczenie.

Lasy losowe (ang. Random Forest): Rozszerzenie podejścia drzew decyzyjnych, które zapobiega przeuczeniu. Lasy losowe łączą wiele drzew decyzyjnych w celu poprawy dokładności. Charakterystyka tego podejścia:

- Każde drzewo działa niezależnie na losowo wybranym podzbiorze danych i cech,
- Wynik modelu jest agregowany poprzez **głosowanie większościowe** (dla klasyfikacji) lub **średnią** (dla regresji).

Chęć pracy na danych nieustrukturyzowanych stanowiła silny bodziec do badań nad uczeniem głębokim, które wykorzystuje sieci neuronowe. W odróżnieniu od uczenia tradycyjnego, opartego na technikach statystycznych i działającego na małych zbiorach danych przy niewielkich wymaganiach obliczeniowych, uczenie głębokie wykorzystuje sieci neuronowe złożone z wielu warstw. Sieci te operują na ogromnych zbiorach danych i wymagają dużej mocy obliczeniowej[6].

Rodzaje głębokich sieci neuronowych:

- **Sieci rekurencyjne (ang. Recurrent Neural Networks, RNN)** – Stosowane do przetwarzania języka naturalnego oraz szeregów czasowych.
- **Sieci transformerowe (ang. Transformers)** – Wykorzystywane w modelach takich jak GPT.

- **Sieci grafowe (ang. Graph Neural Networks, GNN)** – Analizują zależności i struktury, np. w badaniach nad strukturami chemicznymi czy połączeniami komunikacyjnymi.
- **Sieci konwolucyjne (ang. Convolutional Neural Networks, CNN)** – Służą do przetwarzania obrazów i dźwięku. Tematyka ta została omówiona w dalszej części rozdziału[19].

4.1. Model CNN

4.1.1. Historia CNN

Początki konwolucyjnych sieci neuronowych mają swoje początki w latach 60, kiedy to David Hubel i Torsten Wiesel prowadzili badania nad mózgiem kota. Przyczepili oni elektrody do głowy kota i w ten sposób badali aktywność jego mózgu. Zauważono, że podczas pokazywania zwierzęciu konkretnych kształtów na ekranie, aktywowała się określona grupa neuronów w korze wzrokowej. Naukowcy za swoje dokonanie otrzymali w 1981 Nagrodę Nobla.

Badania prowadzone przez Hubela i Torstena były podstawą do dalszego rozwoju. Dwie dekady później w latach 80 japoński informatyk Kunihiko Fukushima zaproponował hierarchiczną, wielowarstwową sztuczną sieć neuronową która służyła do rozpoznawania japońskich znaków, natomiast w 1989 roku YannLecun zastosował propagację wsteczną do modyfikacji współczynników jądra. Dokonanie to, było preludium stworzenia MNIST, bazy danych składającej się z kilkudziesięciu obrazów ręcznie pisanych cyfr, która służyła do trenowania i testowania algorytmów rozpoznawania obrazów.

Przełom miał miejsce w 2012, kiedy to zespół Alexa Krizhevsky wygrał w konkursie ImageNet. Model zaprezentowany w konkursie osiągnął niespotykaną do tej pory skuteczność, przekraczającą 80%. Zastosowano wówczas ogromną bazę danych składającą się z ponad 1.2 miliona obrazów[20].

Sieci konwolucyjne mają bogatą historię a pracę nad ich rozwojem trwają do dziś.

4.1.2. Zasada działania

Głównym zastosowaniem sieci neuronowych jest analiza danych nieustrukturyzowanych i klasyfikowanie ich do określonych klas. Każda próbka danych jest macierzą wartości. Do zrozumienia działania CNN niezbędne jest pojęcie jądra (filtru) oraz kroku, czyli wartości, o jaką będzie przesuwany filtr. W praktyce najczęściej wykorzystuje się wartość jeden. Filtr jest przesuwany po macierzy w celu wyodrębnienia powtarzalnych wzorców. Filtr charakteryzuje się określonym rozmiarem, czyli wymiarem przesuwanego okna nad danymi wejściowymi. Mniejsze jądra wyodrębniają więcej informacji, co w rezultacie

daje głębszą architekturę, ponieważ kolejne warstwy ulegają nieznacznemu zmniejszeniu. Duże filtry stosowane są do wyodrębniania większych elementów. W praktyce jądro jest mnożone przez fragmenty danych, a wyniki są sumowane, tworząc nową zredukowaną macierz nazywaną mapą cech. Wykorzystywana jest operacja splotu, konwolucja, która mierzy całkę z iloczynu punktowego funkcji.

- **Matematycznie:** Konwolucja (splot) mierzy całkę z iloczynu punktowego funkcji:

$$(f * g)(y) = \int_{-\infty}^{\infty} f(y-x)g(x)\,dx \tag{4}$$

- $f(x)$: Pierwsza funkcja (funkcja wejściowa).
- $g(x)$: Druga funkcja (filtr/jądro konwolucyjne).

Funkcja $g(x)$ przesuwana jest względem funkcji $f(x)$, gdzie z każdym przesunięciem mnożone są wartości obu funkcji, a następnie sumowane w celu uzyskania jednej wartości dla danego przesunięcia. Innymi słowy, konwolucja mierzy, jak bardzo filtr $g(x)$ pasuje do danych wejściowych $f(x)$ w celu analizy cech lokalnych.

Proces ten można powtarzać wielokrotnie, a każda nowa macierz, nazywana warstwą konwolucyjną, składa się z bardziej skomplikowanych cech.

Sieci konwolucyjne (CNN) składają się z następujących warstw:

- **Warstwa wejściowa (Input Layer):** Reprezentuje przetworzone dane wejściowe, np. obrazy lub próbki multimedialne.
- **Warstwy splotowe (Convolutional Layers):** Wykorzystują filtry, których wartości (wagi) są optymalizowane podczas treningu, aby wyodrębniać cechy z danych wejściowych.
- **Warstwy łączące (Pooling Layers):** Redukują rozmiar danych wejściowych, upraszczając obliczenia i zmniejszając zużycie pamięci, np. poprzez operację *max pooling*.

Dodatkowo z sieciami CNN związane jest pojęcie padding, czyli wypełniania, oraz funkcji aktywacji. Celem operacji wypełnienia jest dostarczenie takiego samego rozmiaru wyjścia jak wejścia. Jest to realizowane poprzez dodanie sztucznych wag na krawędziach danych, najczęściej o wartości zero. W sieciach CNN wykorzystuje się nieliniowe funkcje aktywacji, które zapewniają wielowarstwowość i dają możliwość realizacji skomplikowanych problemów. Stosowanymi funkcjami aktywacji są funkcje ReLU oraz Softmax. Ta pierwsza, Rectified Linear Activation, określana jest wzorem:

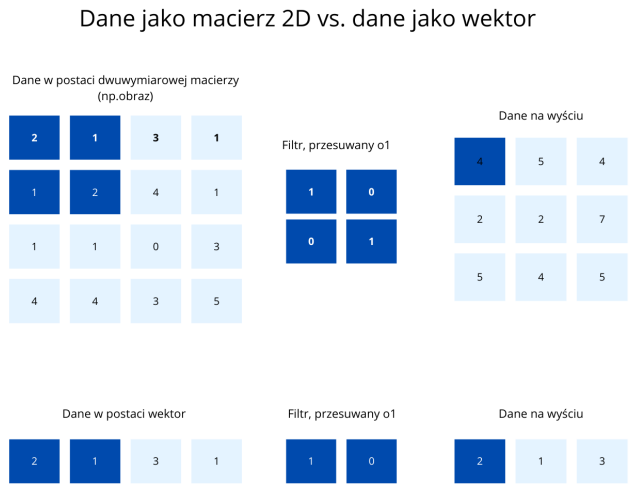
$$\mathrm{ReLU}(x) = \max\{x, 0\} \tag{5}$$

Ze wzoru wynika, że dla wartości wejściowej x, gdy x jest mniejsze lub równe zero,

funkcja zwraca 0, a w przeciwnym razie wartość x. Główną zaletą funkcji ReLU jest zapewnienie szybszego procesu uczenia dzięki prostocie obliczeń oraz rzadkim aktywacjom, ze względu na zwracanie zera dla ujemnych wartości.

Wracając do warstw, w strukturach CNN wyróżnia się również warstwy łączące (pooling layer), których głównym zadaniem jest zmniejszenie rozmiaru próbki wejściowej, co upraszcza obliczenia i zmniejsza zużycie pamięci. Charakterystyczną cechą warstw łączących jest to, że nie posiadają one wag. Każdy neuron w tych warstwach jest połączony z grupą neuronów z warstwy poprzedniej. Neurony te znajdują się w określonym obszarze, nazywanym polem recepcyjnym. Przykładem poolingu jest przekształcenie kilku wartości w jedeną. Często stosuje się max pooling, czyli wybieranie maksymalnej wartości, np. jeśli mamy obszar o rozmiarze 2×2 o wartościach $0, 5, 10, 20$, w max poolingu wybrana zostanie wartość 20 [21].

Jeśli dane wejściowe stanowią próbki obrazów, są one przekształcane w macierze dwuwymiarowe, a filtr stanowi róznież macierz o takim rozmierze. W realizowanym projekcie dane w postaci próbek dźwięku zostały początkowo przekształcone do formy wektora. Działanie CNN w tym przypadku jest analogiczne — filtr przesuwa się wzdłuż danych wejściowych reprezentowanych jako wektor. Różnice pomiędzy tymi przypadkami ilustruje rysunek 4.1.



Rysunek 4.1. Dane w kształcie macierzy dwuwymiarowej a dane w kształcie wektora

Liczby użyte w przedstawionym schemacie zostały dobrane losowo, ponieważ celem było jedynie zobrazowanie działania filtrów w zależności od danych wejściowych. Filtr jest stosowany na danych wejściowych, a wyniki operacji są sumowane i zapisywane w nowej macierzy. W obu przypadkach filtr przesuwa się o jeden krok (stride = 1). Schemat został stworzony przy pomocy platformy *Canva*[12].

4.1.3. Model CNN zastosowany w projekcie

Rysunek 4.2. Struktura modelu CNN

Powyższy schemat przedstawia strukturę modelu CNN zastosowanego w projekcie. Schemat został stworzony przy pomocy platformy *Canva*[12].

Przekształcone dane mające kształt wektorów, składających się z liczb, w których zakodowane są wcześniej wyodrębnione cechy takie jak MFCC czy ZCR, zostają dostarczone do warstwy wejściowej. Do warstwy wejściowej wprowadza się dane treningowe w formie (3095, 1), gdzie pierwsza liczba definiuje liczbę cech, a druga liczbę kanałów. Wartość jeden oznacza, że każda próbka jest traktowana jako jeden ciąg danych (jedna ścieżka informacji).

Kolejne warstwy konwolucyjne skanują dane w poszukiwaniu wzorców, przy czym pierwsze z nich szukają prostych zależności, a kolejne bardziej złożonych. Szukanie wzorców odbywa się przy użyciu filtrów. Filtr jest przesuwany po danych wejściowych, aby sprawdzić, gdzie pasuje. Każdy z filtrów (w pierwszej warstwie użyto ich 512) generuje mapę cech, czyli wartości liczbowe dla kolejnych fragmentów analizowanych danych, które pokazują, jak dobrze filtr pasuje do danych. Wysokie wartości wskazują na podobieństwo ze wzorcem, zaś niskie na brak takiego podobieństwa.

Normalizacja polega na przekształceniu danych tak, aby ich średnia wynosiła 0, a odchylenie standardowe 1. Zabieg ten przyspiesza uczenie modelu oraz stabilizuje wyniki.

Funkcja aktywacji ReLU zamienia wartości ujemne na 0, natomiast wartości dodatnie pozostawia bez zmian, co pomaga modelowi skupić się na istotnych wzorcach wyodrębnionych za pomocą filtrów.

Dodatkowo w modelu użyto warstwy Dropout, która losowo deaktywuje neurony podczas uczenia. Zabieg ten zapobiega przeuczeniu, czyli sytuacji, w której model uczy się „na pamięć” danych treningowych i nie potrafi poprawnie przewidywać nowych danych.

Warstwa spłaszczająca przekształca dane z formy wielowymiarowej 97×128 na jeden wektor o długości 12416. Zabieg ten pozwala połączyć dane z warstw konwolucyjnych z warstwami w pełni połączonymi (gęstymi). Warstwa gęsta to taka, w której każdy neuron jest połączony z neuronami z poprzedniej i następnej warstwy. W zastosowanym modelu warstwa w pełni połączona składa się z 512 neuronów, co pozwala zrozumieć globalne zależności między danymi.

W warstwie wyjściowej dokonuje się klasyfikacja na 5 klas, gdzie każda klasa odpowiada danej emocji.

4.2. Podsumowanie rozdziału

Niniejszy rozdział przybliża informacje na temat modeli wykorzystywanych w uczeniu maszynowym. Wybór odpowiedniego modelu zależy przede wszystkim od zdefiniowanego problemu oraz dostępnych danych. Ze względu na zastosowanie w pracy inżynierskiej głębokich sieci neuronowych typu CNN, przedstawiono ich historię oraz szczegółowo omówiono zasadę działania. Na zakończenie zilustrowano i opisano strukturę modelu CNN użytego w projekcie, w którym wykorzystano pięć warstw konwolucyjnych do rozpoznawania emocji: złości, radości, smutku, zaskoczenia oraz dźwięku neutralnego.

5. Wyniki

5.1. Narzędzia

Projekt został opracowany w środowisku programistycznym *Kaggle*, gdzie przeprowadzono podział oraz przetwarzanie plików *.wav*. W tym samym środowisku zrealizowano również proces treningu, którego celem było utworzenie pliku JSON zawierającego model oraz wytrenowane wagi w formacie HDF5, potrzebne do wdrożenia w aplikacji.

Całość kodu została napisana w języku *Python* z wykorzystaniem następujących bibliotek:

- *NumPy* – do obliczeń numerycznych i operacji na macierzach,
- *Pandas* – do przetwarzania i analizy danych,
- *Matplotlib* – do wizualizacji danych i tworzenia wykresów,
- *Librosa* – do ekstrakcji cech z plików audio,
- *TensorFlow* – do budowania i trenowania modeli głębokiego uczenia,
- *Keras* – wysokopoziomowa biblioteka zintegrowana z *TensorFlow*, umożliwiająca definiowanie, kompilowanie i trenowanie modeli w sposób bardziej przystępny i elastyczny.

Połączenie *TensorFlow* i *Keras* zapewnia zaawansowaną funkcjonalność przy jednoczesnym zachowaniu prostoty obsługi. Dzięki temu tworzenie modeli głębokiego uczenia maszynowego staje się zrozumiałe i dostępne nawet dla osób początkujących w tej dziedzinie.

5.2. Trening

Przetworzone dane stanowią podstawę do trenowania modelu. Trening polega na iteracyjnej optymalizacji wag, które są parametrami sieci neuronowej przypisanymi do każdego połączenia między neuronami. Wartość wagi determinuje, w jakim stopniu dana cecha wejściowa wpływa na kolejne neurony w sieci. Wagi regulują przepływ informacji w modelu, co bezpośrednio wpływa na wyniki wyjściowe. Precyzyjnie zoptymalizowane wagi podczas procesu treningu umożliwiają modelowi skuteczną generalizację na nowych, nieznanych danych.

Aby lepiej zrozumieć, na czym polega trening, warto zapoznać się z pojęciami epoki oraz współczynnika uczenia. Epoka (ang. *epoch*) oznacza jeden pełny przebieg po danych treningowych, podczas którego model przetwarza wszystkie dane treningowe i na podstawie błędów aktualizuje swoje wagi, aby poprawić swoją dokładność w kolejnych epokach. Współczynnik uczenia (ang. *learning rate*) jest parametrem kontrolującym, jak szybko model uczy się z danych, czyli jak duże zmiany są wprowadzane do wag. Im ten współczynnik jest mniejszy, tym model uczy się wolniej, co oznacza, że wprowadza mniejsze zmiany do wag.

Wydajność modelu oceniana jest na podstawie zbioru walidacyjnego gdzie monbitorowane są parmaetry takie jak dokładność i strata. Dokładność określa procent poprawnych przewidywań zaś statę można określić jako mairę błedu modelu czyli różnicę między przewidywaniem a rzeczywistością[6].

5.2.1. Trening w projekcie

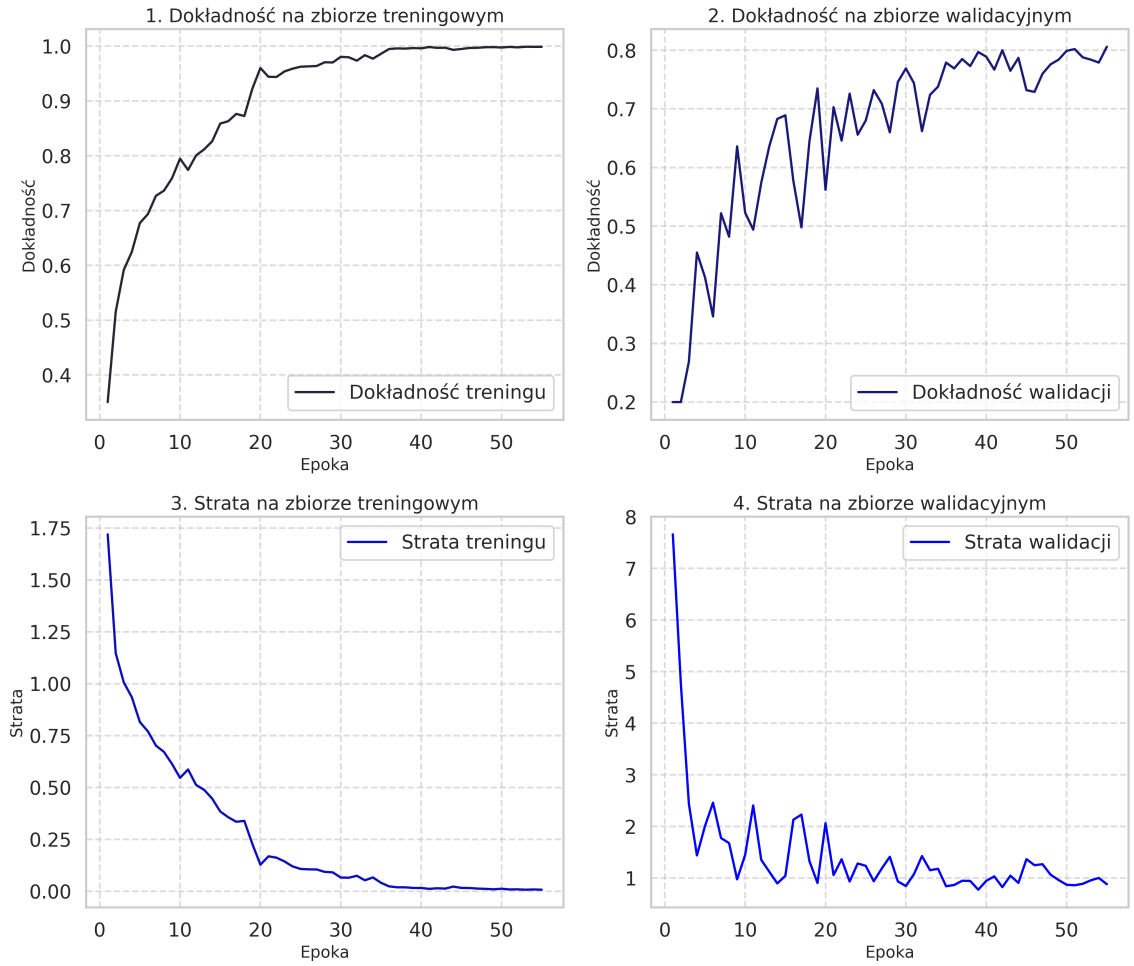
Model został wytrenowany przez 55 epok. W każdej epoce dane treningowe były dzielone na mniejsze porcje (batch), które model przetwarzał jedna po drugiej. Podczas tego procesu obliczane były błędy, na podstawie których model aktualizował swoje wagi. Po przetworzeniu całego zbioru treningowego w danej epoce oceniano wydajność modelu, mierząc dokładność i stratę na danych walidacyjnych. Jeśli model osiągał wyższą dokładność niż wcześniej zarejestrowana, zapisywano bieżące wagi. Współczynnik uczenia był sukcesywnie zmniejszany, jeśli model nie poprawiał się przez trzy kolejne epoki.

Tabela 5.1 zawiera wszystkie zebrane parametry dla 55 epok. Pierwsza kolumna pokazuje łączny czas trwania każdej epoki w sekundach, a kolejne zawierają: dokładność i stratę dla danych treningowych, dokładność i stratę dla danych walidacyjnych oraz wartość współczynnika uczenia, który zmniejszył się z początkowej wartości 0.001 do 0.0000625.

Tabela 5.1. Wyniki dla wszystkich epok (1–55)

Epoka	Czas (s)	Dokładność	Strata	Dokładność walidacji	Strata walidacji	Współczynnik uczenia
1	3173	0.3505	1.7191	0.2000	7.6581	0.0010
2	3160	0.5155	1.1467	0.2000	4.7617	0.0010
3	3077	0.5916	1.0062	0.2690	2.4345	0.0010
4	3112	0.6245	0.9351	0.4550	1.4391	0.0010
5	3120	0.6771	0.8155	0.4130	2.0071	0.0010
6	3155	0.6934	0.7704	0.3460	2.4598	0.0010
7	3156	0.7268	0.7018	0.5220	1.7758	0.0010
8	3148	0.7363	0.6713	0.4820	1.6764	0.0010
9	3210	0.7588	0.6135	0.6360	0.9755	0.0010
10	3179	0.7947	0.5462	0.5230	1.4495	0.0010
11	3064	0.7740	0.5867	0.4940	2.4070	0.0010
12	3070	0.8006	0.5121	0.5740	1.3532	0.0010
13	3095	0.8116	0.4887	0.6360	1.1236	0.0010
14	3050	0.8263	0.4464	0.6830	0.8963	0.0010
15	2960	0.8586	0.3844	0.6890	1.0425	0.0010
16	2972	0.8630	0.3566	0.5770	2.1299	0.0010
17	3001	0.8763	0.3347	0.4980	2.2296	0.0010
18	2965	0.8724	0.3391	0.6450	1.3266	0.0010
19	3009	0.9224	0.2278	0.7350	0.9049	0.0005
20	3014	0.9600	0.1280	0.5620	2.0640	0.0005
21	2817	0.9440	0.1683	0.7030	1.0590	0.0005
22	2822	0.9437	0.1619	0.6460	1.3640	0.0005
23	2848	0.9538	0.1436	0.7260	0.9333	0.0005
24	2812	0.9588	0.1199	0.6560	1.2825	0.0005
25	2846	0.9624	0.1075	0.6800	1.2367	0.0005
26	2846	0.9630	0.1056	0.7320	0.9380	0.0005
27	2858	0.9637	0.1050	0.7090	1.1860	0.0005
28	2805	0.9705	0.0934	0.6600	1.4115	0.0005
29	2803	0.9701	0.0911	0.7460	0.9345	0.0005
30	2806	0.9804	0.0660	0.7690	0.8451	0.0005
31	3021	0.9796	0.0653	0.7440	1.0749	0.0005
32	2987	0.9734	0.0745	0.6620	1.4253	0.0005
33	2986	0.9834	0.0528	0.7240	1.1522	0.0005
34	3018	0.9771	0.0667	0.7380	1.1795	0.0005
35	3021	0.9859	0.0409	0.7790	0.8419	0.00025
36	3008	0.9946	0.0237	0.7690	0.8657	0.00025
37	3009	0.9958	0.0192	0.7850	0.9473	0.00025
38	2990	0.9954	0.0190	0.7730	0.9444	0.00025
39	3001	0.9964	0.0160	0.7970	0.7742	0.00025
40	3008	0.9959	0.0159	0.7890	0.9464	0.00025
41	2981	0.9983	0.0116	0.7670	1.0321	0.00025
42	3032	0.9968	0.0144	0.8000	0.8238	0.00025
43	2907	0.9970	0.0129	0.7650	1.0456	0.00025
44	2905	0.9931	0.0226	0.7870	0.9058	0.00025
45	2905	0.9947	0.0162	0.7320	1.3659	0.00025
46	2909	0.9966	0.0157	0.7290	1.2495	0.00025
47	2911	0.9969	0.0130	0.7600	1.2679	0.00025
48	2957	0.9979	0.0116	0.7760	1.0656	0.000125
49	2950	0.9981	0.0095	0.7840	0.9619	0.000125
50	2949	0.9974	0.0122	0.7990	0.8696	0.000125
51	2956	0.9985	0.0088	0.8020	0.8605	0.000125
52	3003	0.9977	0.0093	0.7880	0.8886	0.000125
53	2968	0.9988	0.0075	0.7840	0.9553	0.000125
54	2971	0.9986	0.0087	0.7790	1.0024	0.0000625
55	2972	0.9987	0.0076	0.8060	0.8808	0.0000625

Dodatkowo, w celu lepszego zobrazowania procesu uczenia, na rysunku 5.1 zamieszczono wykresy przedstawiające zmiany parametrów wydajności, takich jak dokładność i strata, w kolejnych epokach dla danych treningowych i walidacyjnych.



Rysunek 5.1. Ewolucja strat i dokładności w procesie treningu na zbiorze trenigowym i walidacyjnym

Na zdjęciu przedstawiono cztery wykresy ilustrujące zmiany dokładności i strat podczas treningu na zbiorach treningowym i walidacyjnym. Wykres pierwszy ukazuje zmianę dokładności na zbiorze treningowym w miarę postępu epok, natomiast wykres drugi przedstawia zmianę dokładności na zbiorze walidacyjnym. Trzeci wykres obrazuje zmiany strat na zbiorze treningowym, a czwarty przedstawia zmiany strat na zbiorze walidacyjnym. Wykresy zostały stworzone w notebooku dostępnym na *GitLab*[15].

5.2.2. Wnioski dotyczące wyników treningu

Decydując się na zastosowanie głębokich sieci neuronowych CNN, gdzie dla pojedynczej próbki liczba cech wynosiła 3096, istotne było uwzględnienie, że proces treningu przy dużej liczbie epok będzie czasochłonny i będzie wymagał zapewnienia znacznej mocy obliczeniowej. Średni czas treningu jednej epoki wyniósł 2986,87 sekundy, co odpowiada około 50 minutom.

Stabilny wzrost dokładności na zbiorze treningowym świadczy o prawidłowym przebiegu procesu uczenia, w którym model finalnie osiągnął dokładność bliską 1. Malejąca wartość funkcji strat na zbiorze treningowym wskazuje na skuteczne minimalizowanie błędów przez model.

W przypadku danych walidacyjnych również zaobserwowano wzrost dokładności, która osiągnęła poziom ponad 80%. Porównując parametry wydajności dla zbiorów trenin-

gowego i walidacyjnego, zauważono większe fluktuacje w danych walidacyjnych. Wartość funkcji strat spadła z 7,6581 do 0,8808, co świadczy o skutecznej minimalizacji błędów. Jednak w porównaniu do strat na danych treningowych wskazuje to na możliwość dalszego doskonalenia modelu w przyszłości.

Podsumowując, model wykazał dobrą zdolność do uczenia się i generalizacji na danych walidacyjnych, osiągając dokładność na poziomie ponad 80%. Istniejąca rozbieżność między wynikami dla danych treningowych a walidacyjnych sugeruje jednak, że model można jeszcze udoskonalić, aby zwiększyć jego dokładność. W dalszej części stworzony model zostanie poddany ostatecznej ocenie na zbiorze testowym – danych, których model wcześniej nie widział. Wyniki tej oceny pozwolą określić, czy model jest gotowy do wdrożenia w aplikacji.

5.3. Wyniki dla zbioru testowego

Jednym ze sposobów oceny działania stworzonego modelu jest wizualizacja wyników przewidywań za pomocą macierzy pomyłek. Macierz ta przedstawia rzeczywiste klasy w wierszach oraz przewidywane klasy w kolumnach. Analizując macierz pomyłek, można posługiwać się następującymi pojęciami:

- **Prawdziwie pozytywne (PP)** – są to wszystkie wartości znajdujące się na przekątnej macierzy. Idealny klasyfikator miałby wyłącznie wartości niezerowe na przekątnej (przy idealnej klasyfikacji – 100% dla każdej klasy).
- **Prawdziwie negatywne (PN)** – suma wszystkich wartości spoza wiersza i kolumny danej klasy. Reprezentują poprawnie sklasyfikowane próbki jako nie należące do tej klasy.
- **Fałszywie pozytywne (FP)** – suma wartości w kolumnie danej klasy, ale poza przekątną. Są to błędnie sklasyfikowane próbki przypisane do danej klasy.
- **Fałszywie negatywne (FN)** – suma wartości w wierszu danej klasy, ale poza przekątną. Są to błędnie sklasyfikowane próbki, które powinny należeć do danej klasy.

W uczeniu maszynowym do oceny wydajności modelu wykorzystuje się dodatkowe miary, takie jak:

- **Precyzja** – dokładność pozytywnych przewidywań. Definiowana jest jako stosunek prawdziwie pozytywnych przewidywań (**PP**) do sumy prawdziwie pozytywnych przewidywań i fałszywie pozytywnych przewidywań (**FP**):

$$\text{Precyzja} = \frac{PP}{PP + FP} \tag{6}$$

  Miara ta określa, jak bardzo można ufać modelowi w jego pozytywnych decyzjach.

- **Pełność** - miara wskazująca jak dobrze model odnajduje wszytskie rzeczywiste przpadki pozytywne. Definiowana jest jako stosunek prawdziwie pozytywnych przewidywań (**PP**) do sumy prawdziwie pozytywnych przewidywań i fałszywie negatywnych (**FN**):

$$\text{Pełność} = \frac{PP}{PP + FN} \tag{7}$$

- **Średnia harmoniczna F1** – miara stanowiąca połączenie dwóch powyższych miar. Wysoką wartość wyniku F1 można uzyskać tylko w sytuacji, gdy precyzja i pełność są wysokie. Średnia harmoniczna precyzji i pełności definiowana jest jako:

$$F_1 = \frac{2}{\frac{1}{\text{Precyzja}} + \frac{1}{\text{Pełność}}} = 2 \cdot \frac{\text{Precyzja} \cdot \text{Pełność}}{\text{Precyzja} + \text{Pełność}} = \frac{PP}{PP + \frac{FN+FP}{2}} \tag{8}$$

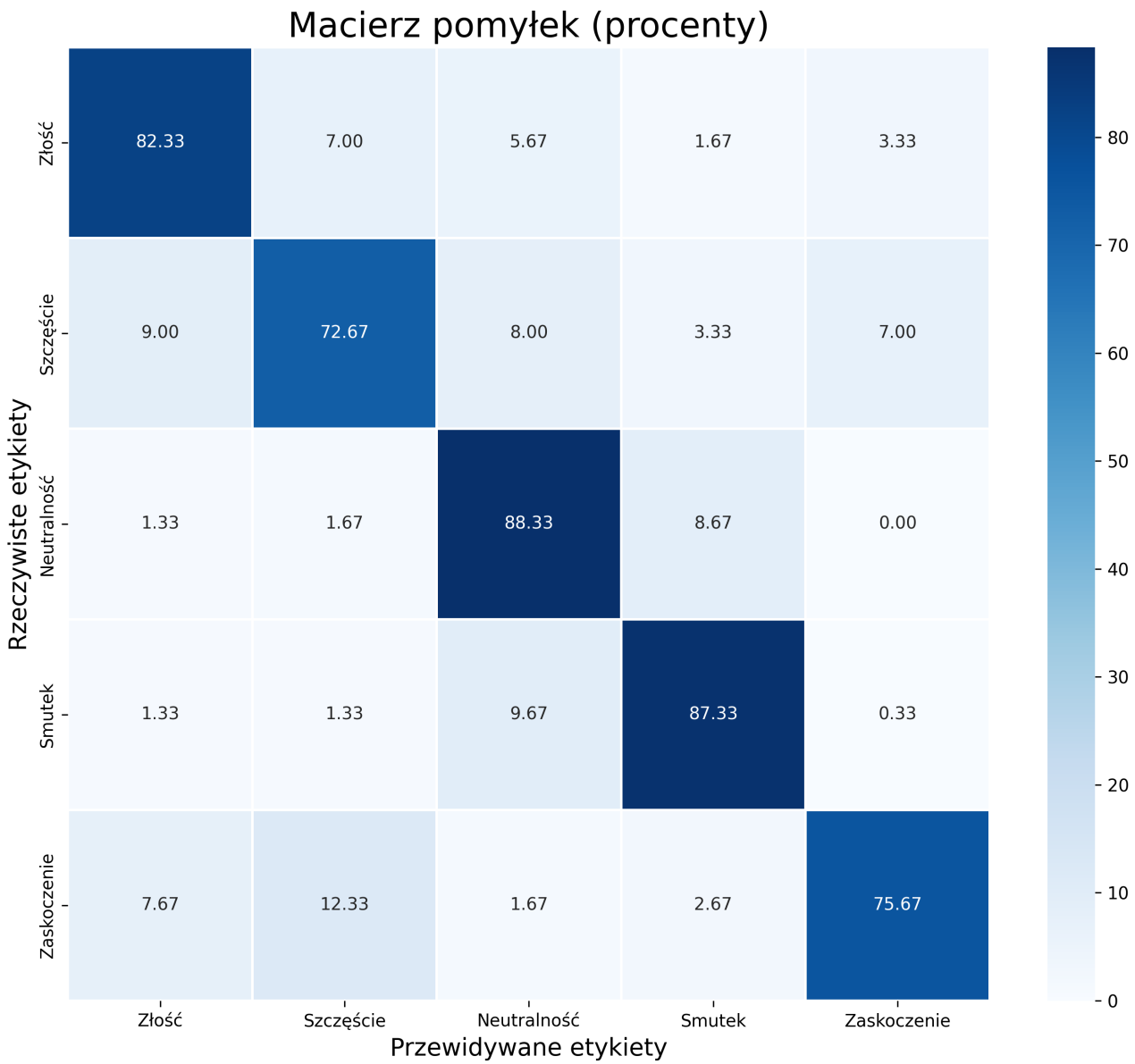
Tabela 5.2 przedstawia wyniki miar, takich jak precyzja, pełność oraz średnia harmoniczna, dla poszczególnych klas. Natomiast Tabela 5.3 prezentuje ostateczne wyniki dokładności oraz straty na zbiorze testowym.

Klasa	**Precyzja**	**Pełność**	**F1-miara**	**Próbki**
Złość	0.81	0.82	0.82	300
Szczęście	0.76	0.73	0.75	300
Neutralność	0.78	0.88	0.83	300
Smutek	0.84	0.87	0.86	300
Zaskoczenie	0.88	0.76	0.81	300
Dokładność		0.81		1500
Średnia arytmetyczna	0.81	0.81	0.81	1500
Średnia ważona	0.81	0.81	0.81	1500

Tabela 5.2. Ewaluacja modelu na zbiorze testowym.

Metryka	**Wartość**
Dokładność na zbiorze testowym	0.8127
Strata na zbiorze testowym	0.9271

Tabela 5.3. Ogólne wyniki modelu na zbiorze testowym.



Rysunek 5.2. Macierz pomyłek dla zbioru testowego

Macierz pomyłek przedstawia procentowy rozkład klasyfikacji modelu w poszczególnych klasach emocji. Wartości na przekątnej wskazują poprawne klasyfikacje, podczas gdy pozostałe pola ilustrują błędy w predykcjach. Macierz pomyłek została stworzona w notebooku dostępnym na *GitLab*[15].

5.3.1. Wnioski dotyczące wyników dla zbioru testowego

Stworzony model został przetestowany na zbiorze testowym. Zbiór testowy, jak wspomniano w poprzednich rozdziałach, jest zbiorem nieznanym dla modelu, w odróżnieniu od zbioru walidacyjnego, do którego parametry modelu są dostrajane. Wyniki ze zbioru testowego stanowią najbardziej wiarygodne źródło informacji o poprawności przewidywań modelu.

Na podstawie wyników ze zbioru testowego model był modyfikowany, a jego hiperparametry dostrajane w taki sposób, aby uzyskać jeszcze wyższą dokładność. Ostateczna dokładność na zbiorze testowym dla wszystkich klas wyniosła **81,27%**, co stanowi satysfakcjonujący wynik i daje możliwość dalszej pracy nad modelem w celu jego dalszego ulepszania.

Wyniki przypadków prawdziwie pozytywnych są przedstawione w tabeli 5.4:

Tabela 5.4. Przypadki prawdziwie pozytywne dla poszczególnych emocji (w procentach).

Emocja	**Prawdziwie pozytywne (%)**
Złość	82.33
Szczęście	72.67
Neutralność	88.33
Smutek	87.33
Zaskoczenie	75.67

Największe dokładności uzyskano dla emocji takich jak **Neutralność** i **Smutek** (blisko 90%). Nieco niższe wyniki dla emocji Szczęście (72,67%) i Zaskoczenie(75,67%), gdzie wciąż są wysokie, ale wskazują na możliwość dalszego udoskonalania modelu w przyszłości. Różnica w wynikach pomiędzy poszczególnymi klasami wynika z dużego podobieństwa próbek reprezentujących emocje Szczęście i Zaskoczenie. Po analizie bazy danych można zauważyć, że zaskoczenie w wielu przypadkach ma pozytywny charakter, co sprawiło, że model miał trudności z odróżnieniem tych emocji.

Dodatkowo baza danych jest pozbawiona kluczowych cech akustycznych, które mogłyby lepiej różnicować emocje, takich jak np. śmiech w przypadku radości czy szloch w przypadku smutku.

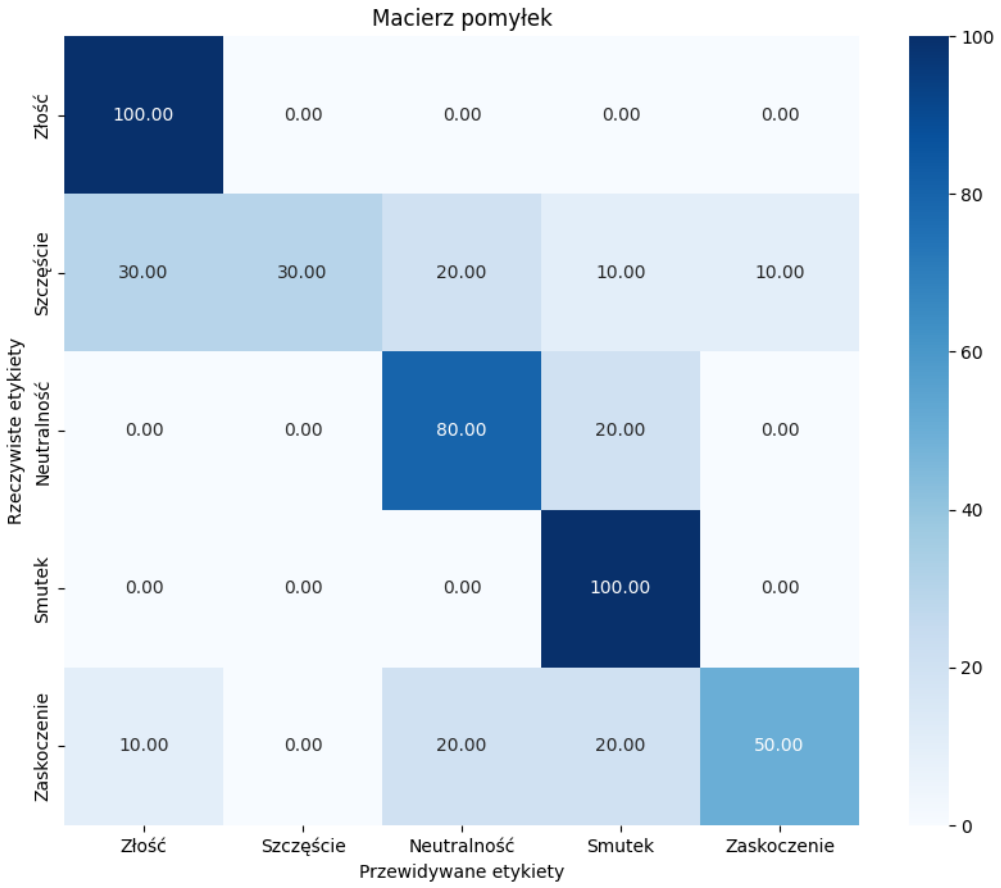
Analizując tabelę zawierającą miary wydajności, takie jak precyzja, pełność i miara F1 dla poszczególnych klas, uzyskano wysokie wyniki, w wielu przypadkach przekraczające 80%.

Podsumowując, wyniki uzyskane w projekcie są satysfakcjonujące, ponieważ w większości przypadków przekraczają **80%**. Otrzymane rezultaty pozwalają wdrożyć stworzony model do aplikacji, której szczegóły zostaną omówione rozdziale 6.

5.4. Wyniki dla zbioru testowego autora pracy

Dodatkowo przeprowadzono eksperyment na stworzonej bazie danych, składającej się z 50 plików audio, obejmujących po 10 nagrań w języku angielskim dla każdej emocji. W celu zachowania spójności z bazą danych *Emotional Speech Database*, każde nagranie trwało maksymalnie 3 sekundy i było rejestrowane z częstotliwością próbkowania 16 000 Hz. Nagrania zostały wykonane przy użyciu programu *Audacity*, który służy do nagrywania i edycji dźwięku.

Wyniki eksperymentu przedstawiono w formie macierzy pomyłek, która potwierdziła, że model skutecznie rozpoznaje emocje takie jak smutek, neutralność i złość. Niższe wyniki osiągnięto dla emocji szczęścia i zaskoczenia. Ogólna dokładność modelu wyniosła 72%.



Rysunek 5.3. Macierz pomyłek dla zbioru testowego autora pracy inżynierskiej

Macierz pomyłek została stworzona w notebooku dostępnym na *GitLab*[15].

5.5. Podsumowanie rozdziału

Opracowany model został stworzony w środowisku *Kaggle* i jest dostępny w repozytorium na platformie *GitLab* [15]. Model został przetrenowany przez 55 epok, osiągając dokładność na zbiorze testowym, składającym się z 1500 plików, na poziomie 81,27%.

Dodatkowo wyznaczono miary takie jak pełność, precyzja i F1 dla każdej z pięciu klas. Przeprowadzono również eksperyment na małej autorskiej bazie danych, składającej się z 50 plików. Wyniki eksperymentu potwierdziły regułę, że model gorzej radzi sobie z emocjami takimi jak zaskoczenie i szczęście, natomiast bardzo dobrze rozpoznaje emocje takie jak smutek, złość oraz neutralność.

6. Aplikacja

6.1. Wstęp

Ostatnim krokiem w procesie realizacji projektu ML, przedstawionego na początku pracy, było zaimplementowanie stworzonego modelu do wcześniej opracowanej aplikacji webowej opartej na frameworku Flask. Model został zapisany w formacie JSON, który przechowuje architekturę sieci, w tym informacje o warstwach, ich połączeniach oraz parametrach. Plik *CNN_model_55epoch.json* zawiera te dane. Dodatkowo, wytrenowane wagi modelu zapisano w formacie HDF5 pod nazwą *CNN_model_weights_55epoch.weights.h5*. Celem było stworzenie prostej i intuicyjnej aplikacji, której podstawowym zadaniem jest przewidywanie emocji na podstawie dostarczonych próbek dźwiękowych.

6.2. Technologie i biblioteki

Poniżej wypunktowano technologie i biblioteki potrzebne do stworzenia strony webowej, która została opracowana w środowisku programistycznym **Visual Studio Code**.

1. **Flask**: Framework w *Pythonie* do tworzenia aplikacji webowych. Mikroframework oparty na protokole **HTTP**, zawierający podstawowe funkcjonalności umożliwiające tworzenie prostych aplikacji.
2. **HTML/CSS**: Technologie frontendowe wykorzystywane do tworzenia interfejsu użytkownika, pozwalające na prostą i intuicyjną obsługę usług zaimplementowanych w backendzie, takich jak przetwarzanie plików dźwiękowych oraz przewidywanie w nich emocji.
3. **TensorFlow/Keras**: Biblioteki, które zostały wcześniej wykorzystane do stworzenia modelu, zaimplementowano również w aplikacji w celu korzystania z wytrenowanego modelu sieci neuronowych.
4. **Librosa**: Biblioteka umożliwiająca ekstrakcję cech z dostarczonych próbek dźwiękowych, zgodnie z metodami użytymi w procesie przygotowania danych treningowych.

6.3. Architektura aplikacji

Stworzona aplikacja składa się z kilku plików, które razem tworzą kompleksowe rozwiązanie umożliwiające wykrywanie dominującej emocji w zadanej próbce dźwiękowej.

1. **app.py**: Główny plik aplikacji zarządzający logiką działania oraz uruchamiający serwer **Flask**. W tym pliku ładowane są stworzone pliki zawierające architekturę modelu oraz wytrenowane wagi. Odpowiada również za ekstrakcję cech oraz przewidywanie emocji.
2. **predict.html**: Strona umożliwiająca przesyłanie plików do analizy. Pozwala na wyświetlanie przewidywanych emocji oraz odsłuchanie wgranego pliku.

3. **record.html**: Strona umożliwiająca nagrywanie dźwięku przez użytkownika bezpośrednio w przeglądarce oraz jego pobranie.
4. **instruction.html**: Strona zawierająca instrukcję obsługi, która wyjaśnia użytkownikowi, jak korzystać z funkcjonalności aplikacji.

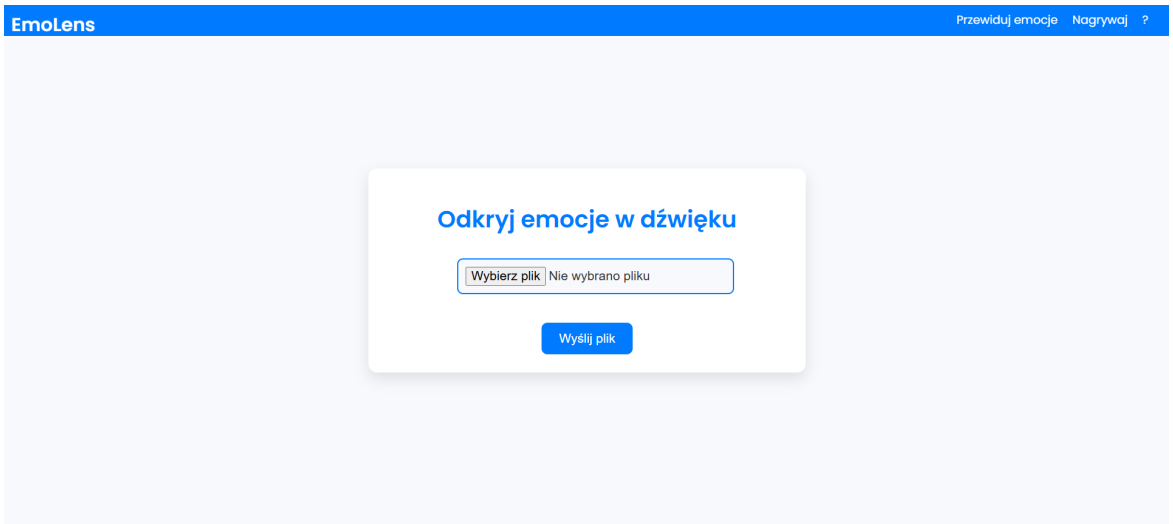
6.4. Prezentacja strony webowej

Aby uruchomić aplikację, należy uruchomić plik *app.py*, dostępny do pobrania w repozytorium na platformie *GitLab*[15], a następnie wpisać w przeglądarce adres: **http://127.0.0.1:5000**. Strona zawiera trzy główne zakładki:

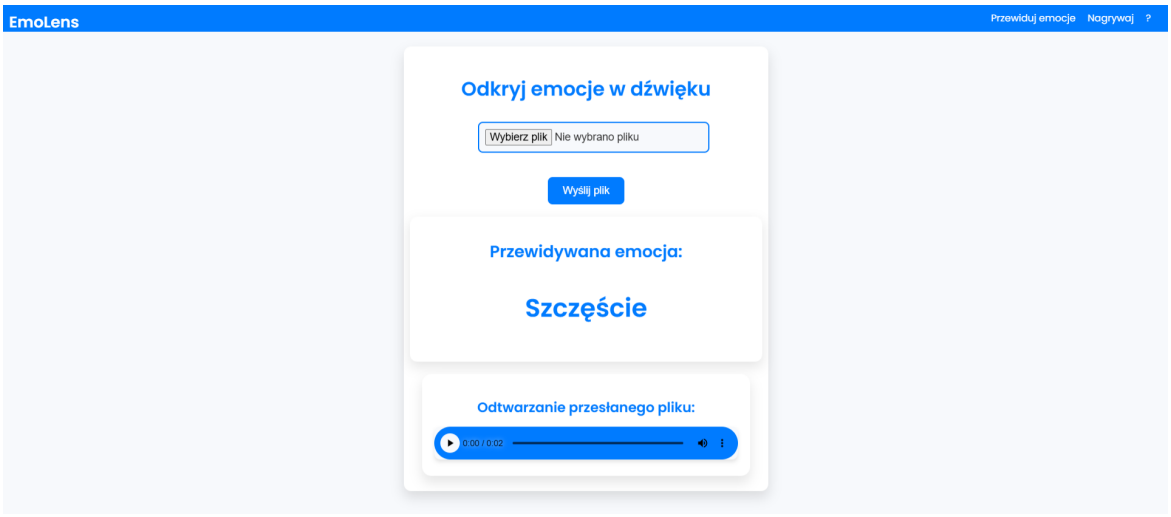
- Strona główna: *Przewiduj emocje*,
- Strona do nagrywania dźwięku: *Nagrywaj*,
- Instrukcja obsługi: *?*.

Aplikacja umożliwia wgrywanie plików *.wav*, które są analizowane w celu klasyfikacji do odpowiedniej grupy emocji. Jeśli użytkownik nie posiada pliku dźwiękowego, może go nagrać, korzystając z zakładki *Nagrywaj*. W przypadku trudności z obsługą aplikacji dostępna jest instrukcja pod ikoną znaku zapytania w prawym górnym rogu strony. Całość została zaprojektowana w kolorystyce niebieskiej, aby zapewnić przejrzystość i nowoczesny wygląd aplikacji.

6.4.1. Strona główna



Rysunek 6.1. Widok strony głównej po włączeniu.



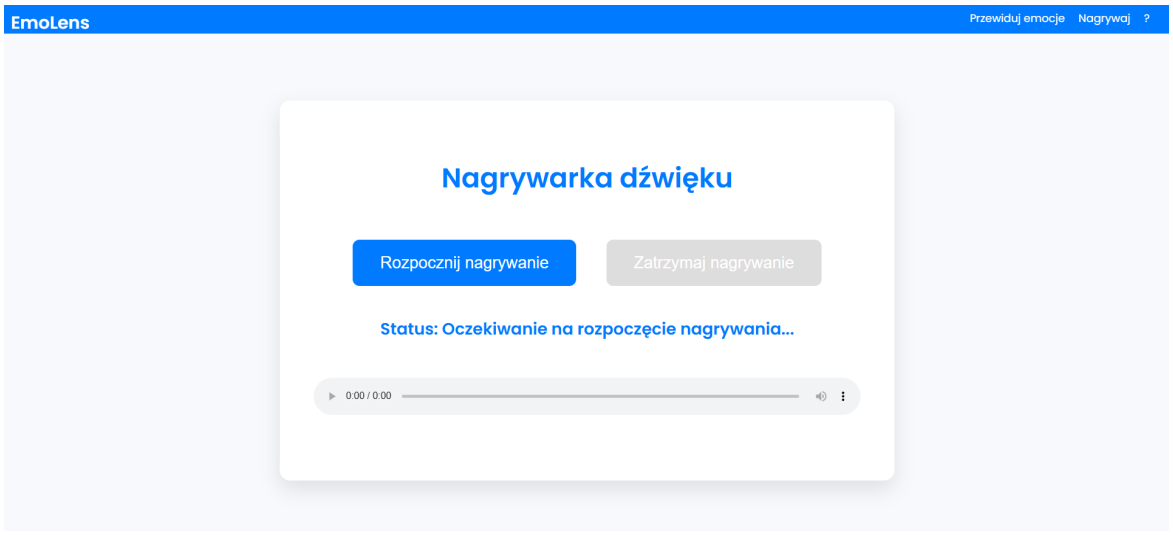
Rysunek 6.2. Widok strony głównej po wgraniu pliku dźwiękowego i przewidzeniu emocji.

Po uruchomieniu aplikacji wyświetla się okno, w którym użytkownik ma możliwość wgrania pliku dźwiękowego ze swojego urządzenia. Docelowo należy wgrać plik w formacie *.wav*, jednak jeśli użytkownik wgra plik o innym rozszerzeniu, zostanie on automatycznie przekonwertowany na format *.wav*. Po przesłaniu pliku do backendu aplikacji system przetwarza dostarczoną próbkę, wyodrębnia z niej cechy i na ich podstawie przewiduje emocję. Wynik przewidywań wyświetlany jest użytkownikowi. Możliwe emocje to:

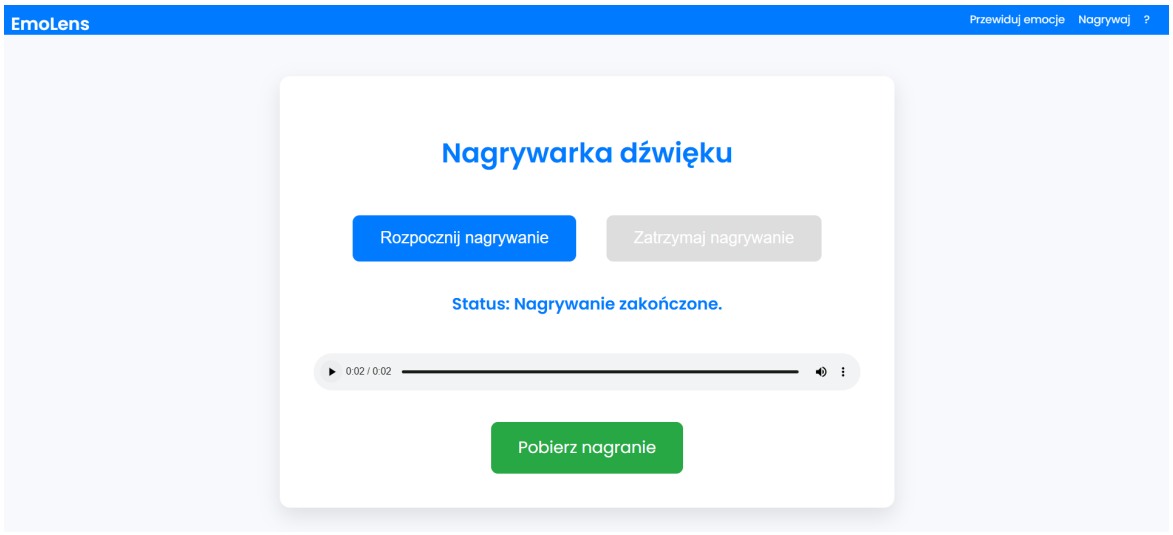
- Szczęście,
- Złość,
- Zaskoczenie,
- Smutek,
- Neutralność.

Dodatkowo użytkownik ma możliwość odsłuchania przesłanego dźwięku, co pozwala na porównanie wyniku przewidywań z subiektywnym wrażeniem.

6.4.2. Strona *Nagrywaj*



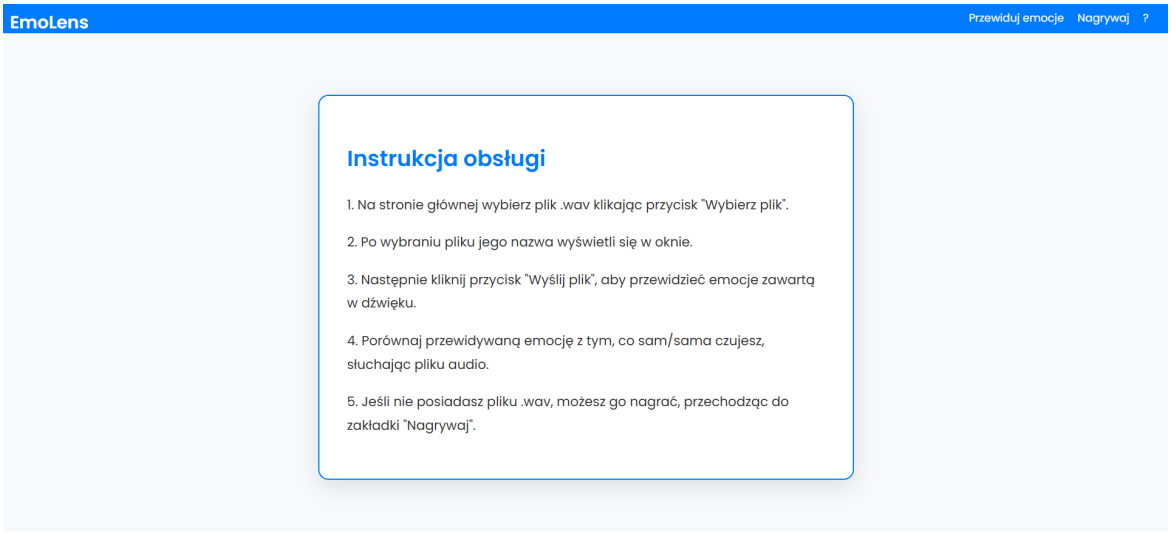
Rysunek 6.3. Widok strony przed nagrywaniem.



Rysunek 6.4. Widok strony po nagrywaniu.

W drugiej zakładce znajduje się strona obsługująca nagrywanie dźwięku. Użytkownik ma możliwość nagrywania własnych próbek dźwiękowych za pomocą wbudowanej nagrywarki. Proces nagrywania obsługiwany jest przez dwa przyciski: *Rozpocznij nagrywanie* oraz *Zatrzymaj nagranie*. Dodatkowo użytkownik może odsłuchać nagrany dźwięk, a jeśli spełnia on jego oczekiwania, ma możliwość pobrania pliku. Następnie, przechodząc na stronę główną, użytkownik może wykryć emocje zawarte w nagranym dźwięku.

6.4.3. Instrukcja obsługi



Rysunek 6.5. Zakładka *?* jako instrukcja obsługi.

Ostatnia zakładka zawiera instrukcję obsługi, która wyjaśnia, jak korzystać z przygotowanej strony internetowej. Dodanie intuicyjnych wskazówek na stronie jest istotne, ponieważ ułatwia nowym użytkownikom zorientowanie się w jej funkcjonalnościach. Jest to szczególnie ważne dla osób starszych, które często napotykają trudności w obsłudze stron internetowych.

6.5. Podsumowanie rozdziału

Etap stworzenia strony internetowej stanowił zakończenie projektu ML. Opracowany model został zaimplementowany w aplikacji webowej, zbudowanej przy użyciu mikroframeworka Flask. Głównym celem aplikacji jest wykrywanie emocji w dźwięku. Dodatkowo aplikacja oferuje funkcje nagrywania dźwięku oraz szczegółową instrukcję obsługi. Przy projektowaniu zadbano, aby interfejs był prosty, intuicyjny i przejrzysty, umożliwiając użytkownikom wygodne korzystanie z podstawowych funkcji. Wśród nich znajdują się przewidywanie emocji, odsłuchiwanie nagrań w celu porównania wyników z własnym wrażeniem oraz możliwość rejestracji nowych próbek dźwiękowych.

7. Podsumowanie

„Ze szczęściem czasami bywa tak, jak z okularami, szuka się ich, a one siedzą na nosie”

Phil Bosmans

Celem niniejszej pracy inżynierskiej było opracowanie systemu umożliwiającego wykrywanie emocji w głosie, co zostało osiągnięte na poziomie 81,27%, pozwalającym na wdrożenie modelu do aplikacji.

Emocje, będące nieodłącznym elementem ludzkiego życia, są skomplikowane, zmienne i czasami trudne do uchwycenia, co sprawia, że stanowią duże wyzwanie dla technologii. Ludzie posiadają wrodzoną zdolność rozpoznawania emocji u innych, ponieważ była to umiejętność kluczowa dla przetrwania gatunku ludzkiego już od czasów prehistorycznych. Dzięki wrodzonemu instynktowi oraz nauce zachowań innych ludzi od najmłodszych lat jesteśmy w stanie w sposób intuicyjny rozpoznawać emocje w ułamku sekundy. Nawet jeśli osoba stara się ukryć swoje prawdziwe emocje, nasze instynkty podpowiadają nam, jaka jest prawda.

Dla porównania, maszyna uczy się jedynie na dostarczonych przez nas danych i nie posiada szóstego zmysłu pozwalającego w sposób intuicyjny ocenić nastrój danej osoby. Potrzebuje ona kontrastowych cech w danych, aby wykrywać różnice pomiędzy emocjami. Dziedzina sztucznej inteligencji związana z rozpoznawaniem emocji nadal stanowi wyzwanie dla inżynierów, ponieważ każdy człowiek jest inny i wyraża swoje emocje w indywidualny sposób.

Testy przeprowadzone na bazie danych *Emotional Speech Database* pozwoliły osiągnąć satysfakcjonujące wyniki na poziomie 81,27% na zbiorze testowym. Warto jednak zauważyć, że testy wykonano na próbkach głosowych pochodzących od 10 aktorów, podczas gdy liczba ludzi na świecie wynosi obecnie 8 201 460 700[24]. Uzyskane wyniki na zbiorze testowym, stworzonym specjalnie na potrzeby tej pracy, osiągnęły poziom 72%. Potwierdziła się również reguła, że model ma trudności z rozpoznawaniem emocji takich jak szczęście, podczas gdy bardzo dobrze radzi sobie z dźwiękami mniej dynamicznymi, takimi jak smutek i neutralność, gdzie wyniki oscylują na poziomie bliskim 90%.

7.1. Napotkane trudności

Przedstawienie emocji w sposób zrozumiały dla maszyny stanowi duże wyzwanie ze względu na ich złożoność i trudność w jednoznacznym matematycznym zapisaniu. Próbka trwająca zaledwie 3 sekundy może być zapisana jako wektor składający się z 3096 wartości. Taka liczba cech sprawiła, że proces uczenia modelu był czasochłonny. Projekt realizowano w środowisku lokalnym, wykorzystując kartę graficzną Intel(R) Iris(R) Xe

[24] Dane z dnia 22.01.2025, godzina 21:30, według platformy *Worldmeter*

Graphics w laptopie, co powodowało, że trening jednej epoki trwał aż 50 minut. W przyszłości, przy realizacji projektów związanych z uczeniem maszynowym i wykorzystaniem sieci splotowych, warto byłoby skorzystać ze środowiska obliczeniowego udostępnionego przez uczelnię. Takie rozwiązanie mogłoby znacząco skrócić czas treningu, umożliwiając elastyczne dostosowywanie parametrów modelu oraz przeprowadzenie większej liczby eksperymentów.

Drugą napotkaną trudnością była ograniczona dostępność baz danych dotyczących emocji w głosie. Zastosowana baza danych, *Emotional Speech Database*, stanowiła największą znalezioną bazę, jednak jej próbki nie były idealne. Przykładem może być podobieństwo między próbkami związanymi z emocjami szczęścia i zaskoczenia, gdzie zaskoczenie często miało charakter pozytywny. Dodatkowo baza danych nie zawierała charakterystycznych cech emocjonalnych, takich jak śmiech czy krzyk.

Przeprowadzone testy na połączonych różnych bazach danych nie przyniosły satysfakcjonujących efektów. Było to spowodowane brakiem spójności między danymi – różniły się one częstotliwością próbkowania, czasem trwania oraz językiem.

Podsumowując, największymi wyzwaniami podczas realizacji projektu były czas trwania treningu każdej epoki oraz ograniczona dostępność wysokiej jakości baz danych związanych z emocjami w głosie.

7.2. Możliwość rozwoju

Każdy projekt z zakresu uczenia maszynowego można rozwijać, na przykład poprzez optymalizację modelu w celu zwiększenia jego wydajności lub rozszerzenie liczby klas rozpoznawanych przez system. Projekt rozpoznawania emocji w głosie, skoncentrowany na klasyfikacji pięciu emocji: radości, zaskoczenia, smutku, złości oraz neutralności, stanowił solidną bazę dla dalszych badań i eksperymentów w obszarze uczenia maszynowego. Jest to szczególnie istotne, ponieważ dziedzina ta, choć relatywnie młoda, rozwija się w szybkim tempie, oferując szerokie możliwości zastosowań.

Rozbudowa projektu mogłaby obejmować integrację dodatkowych komponentów, takich jak rozpoznawanie emocji na podstawie mimiki twarzy czy gestykulacji. Taka rozbudowa umożliwiłaby stworzenie zaawansowanego systemu, podobnego do *SimSensei* – wirtualnego terapeuty, który w czasie rzeczywistym analizuje mowę, mimikę oraz gesty, oferując holistyczne podejście do rozpoznawania emocji.

Podsumowując, zrealizowany projekt był wartościowym doświadczeniem, które umożliwiło zgłębienie podstaw uczenia maszynowego i otworzyło możliwości do dalszej eksploracji tego zagadnienia. Projekt stanowi solidny fundament do realizacji przyszłych przedsięwzięć w zakresie zaawansowanych systemów rozpoznawania emocji.

7.3. GitLab

Wszystkie pliki stworzone na potrzeby zrealizowania pracy inżynierskiej zostały umieszczone w środowisku *GitLab* [15]. Repozytorium zawiera trzy foldery:

- **Aplikacja_webowa** – zawiera wszystkie pliki związane z stworzoną aplikacją, pozwalającą użytkownikowi wykrywać emocje w próbce dźwiękowej.
- **Autorska_baza_danych** – zawiera plik *.zip* ze stworzoną bazą danych składającą się z 50 plików wraz z subskrypcją.
- **Notebook** – folder zawierający notebook pobrany ze środowiska Kaggle, w którym został stworzony model rozpoznawania emocji.

Dokładna instrukcja pozwalająca uruchomić aplikację znajduje się w pliku *README.md*.

Bibliografia

[1] C. Darwin, *The Expression of the Emotions in Man and Animals*, English. United Kingdom: John Murray, 1872, First edition.

[2] P. Ekman, *Emotions Revealed: Understanding Faces and Feelings*. Hachette UK, 2012, s. 304, Oryginalne wydanie: April 7, 2003, ISBN: 1780225504, 9781780225500.

[3] N. J. Nilsson, *John McCarthy: A Biographical Memoirs*. United States: National Academy of Sciences, 2012. adr.: https://www.nasonline.org/wp-content/uploads/2024/06/mccarthy-john.pdf.

[4] L. G. Shapiro, *Computer Vision: The Last Fifty Years*. adr.: https://homes.cs.washington.edu/~shapiro/vision.pdf.

[5] A. Metallinou, S. Lee i S. Narayanan, *Audio-Visual Emotion Recognition using Gaussian Mixture Models for Face and Voice*, Los Angeles, CA 90089-2560. adr.: https://ict.usc.edu/pubs/Audio-Visual%20Emotion%20Recognition%20using%20Gaussian%20Mixture%20Models%20for%20Face%20and%20Voice.pdf.

[6] A. Geron, *Uczenie maszynowe z użyciem Scikit-Learn, Keras i TensorFlow*, polski, III, tłum. K. Sawka. Gliwice, Polska: Wydawnictwo Helion, 2023, s. 776, Data premiery: 2023-07-04, okładka miękka, ISBN: 1388727725.

[7] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal i T. Shimamura, "Facial Emotion Recognition Using Transfer Learning in the Deep CNN", *Electronics*, 2021. DOI: 10.3390/electronics10091036.

[8] L.-P. Morency, G. Stratou, D. DeVault i in., "SimSensei Demonstration: A Perceptive Virtual Human Interviewer for Healthcare Applications", w *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII)*, Los Angeles, California, 2017.

[9] N. Innowacje, *Maszyny odczytują emocje. Czy w przyszłości będą potrafiły rozpoznać nastrój człowieka?*, Dostęp online: 2024-11-12, 2024. adr.: https://innowacje.newseria.pl/news/maszyny-odczytuja-emocje,p873573058.

[10] B. Update, "UK train stations trial Amazon emotion recognition on passengers", Dostęp online: 12.11.2024, 2024. adr.: https://www.biometricupdate.com/202406/uk-train-stations-trial-amazon-emotion-recognition-on-passengers.

[11] K. Drelczuk, *Sztuczna Inteligencja w Chinach. Oprawca czy wybawca*, 2023. adr.: https://www.youtube.com/watch?v=F2FsRqWJDvc.

[12] Canva, *Canva: Design anything. Create stunning graphics and documents*, https://www.canva.com, Dostępne na stronie internetowej.

[13] K. Zhou, B. Sisman, R. Liu i H. Li, *Emotional Speech Dataset (ESD)*. adr.: https://hltsingapore.github.io/ESD/download.html.

[14] K. Zhou, B. Sisman, R. Liu i H. Li, "Emotional voice conversion: Theory, databases and ESD", *Speech Communication*, t. 137, s. 1–18, 2022, ISSN: 0167-6393.

[15] S. Rojek, *Praca Inżynierska - Wykrywanie emocji w głosie*, 2025. adr.: https://gitlab-stud.elka.pw.edu.pl/srojek/praca_inzynierska.

[16] A. Król-Nowak i K. Kotarba, *Podstawy uczenia maszynowego*, A. Król-Nowak, red. Kraków: Wydawnictwa AGH, 2022, ISBN: 978-83-67427-05-0. Pod redakcją Aleksandry Król-Nowak. adr.: https://winntbg.bg.agh.edu.pl/skrypty4/0599/podstawy_uczenia.pdf.

[17] "librosa.feature: Audio Feature Extraction Documentation", spraw. tech. adr.: https://librosa.org/doc/main/feature.html.

[18] K. Ł. i Józef Kotus, "Akustyka mowy: Parametryzacja sygnału mowy. Perceptualne skale częstotliwości", Katedra Systemów Multimedialnych, spraw. tech. adr.: https://sound.eti.pg.gda.pl/student/amowy/AM_05_parametryzacja.pdf.

[19] M. Ziaja, *Przekrojowe wprowadzenie do uczenia maszynowego*, https://nowy.kmim.wm.pwr.edu.pl/pl/event/2024/03/przekrojowe-wprowadzenie-do-uczenia-maszynowego/przekrojowe-wprowadzenie-do-uczenia-maszynowego.pdf, Seminarium Dynamiki PWR, marzec 2024. POLSL, Katedra Algorytmiki i Oprogramowania | KP Labs, dział uczenia maszynowego., 2024.

[20] M. Mamczur, *Jak działają konwolucyjne sieci neuronowe (CNN)?*, Blog o data science. adr.: https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/.

[21] K. Bogus, *Sieci splotowe (CNN)*, Prezentacja, 2022. adr.: https://prac.im.pwr.edu.pl/~bdyda/sem2022zimowy/ml/Sieci_splotowe_09_11_2022.pdf.

Wykaz symboli i skrótów

AI – ang. *Artificial Intelligence*
CNN – ang. *Convolutional Neural Networks*
EAI – ang. *Emotion AI*
EiTI – Wydział Elektroniki i Technik Informacyjnych
ESD – ang. *Emotional Speech Database*
F0 – ang. *Fundamental Frequency*
MFCC – ang. *Mel Frequency Cepstral Coefficients*
ML – ang. *Machine Learning*
PW – Politechnika Warszawska
RMS – ang. *Root Mean Square*
ZCR – ang. *Zero Crossing Rate*

Spis rysunków

Spis tabel