

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Automatická dátová augmentácia pre
predikciu PD z hlasu**

Diplomová práca

2024

Bc. Martin Obrcian

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Automatická dátová augmentácia pre
predikciu PD z hlasu**

Diplomová práca

Študijný program: Informatika
Študijný odbor: 9.2.1. Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: prof. Ing. Peter Drotár PhD.

Košice 2024

Bc. Martin Obrcian

Abstrakt v SJ

Táto práca sa zaoberá problematikou zovšeobecnenia modelov neurónových sietí pre diagnostiku Parkinsonovej choroby z hlasu s ohľadom na rôzne jazykové prostredia. V práci sú preskúmané metódy na zvýšenie zovšeobecniteľnosti, najmä prostredníctvom dátových augmentácií a automatizácie ich aplikácie. Navrhovaný prístup aplikuje augmentačné techniky priamo na zvukové vzorky a následne ich prevádza do obrázkovej formy ako Mel spektrogramy. Výsledky experimentov potvrdili, že je možné navrhnuť program, ktorý dokáže automaticky aplikovať špecifikované augmentačné techniky s príslušnými magnitúdami nezávisle od seba i súčasne. Tento prístup umožňuje identifikovať najlepšie augmentačné techniky pre každú verifikovanú dátovú sadu, čo umožňuje vytvoriť robustnejší model. Navrhnuté riešenie má potenciál na aplikáciu aj v ďalších úlohách rečovej analýzy.

Klíčové slová v SJ

neurónové siete, analýza rečových dát, doménové zovšeobecnenie, automatická dátová augmentácia, dátové augmentácie, Parkinsonova choroba

Abstract

This thesis addresses the problem of generalizing neural network models for the diagnosis of Parkinson's disease from voice with respect to different language environments. Methods to increase the generalization are explored in this work, in particular through data augmentation and their automated application. The proposed approach applies the augmentation techniques directly to audio samples and then converts them into image form as Mel spectrograms. Experimental results confirm that it is possible to design a program that can automatically apply the specified augmentation techniques with the appropriate magnitudes independently and simultaneously. This approach allows the best augmentation techniques to be identified for each verified dataset, allowing a more robust model to be developed. The proposed solution has the potential for application in other speech analysis tasks.

Keywords

neural networks, speech data analysis, domain generalization, automatic data augmentation, data augmentations, Parkinson's disease

Bibliografická citácia

OBRCIAN, Martin. *Automatická dátová augmentácia pre predikciu PD z hlasu*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2024. ??s.
Vedúci práce: prof. Ing. Peter Drotár PhD.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

**ZADANIE
DIPLOMOVEJ PRÁCE**

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

Automatická dátova augmentácia pre predikciu PD z hlasu
Automatic Data Augmentation for PD Detection from Voice

Študent: **Bc. Martin Obrcian**

Školiteľ: **prof. Ing. Peter Drotár, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. Navrhnuť prístup na báze neuronových sietí pre diagnostiku PD z hlasu.
2. Overiť zovšeobecniteľnosť tohto prístupu na viacerých dátových sadách.
3. Navrhnuť a experimentálne vyhodnotiť metódy pre zovšeobecniteľnosť modelov.
4. Zhodnotiť a spracovať výsledky.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 19.04.2024

Dátum zadania diplomovej práce: 31.10.2023



L. Vokorokos

prof. Ing. Liberios Vokorokos, PhD.

dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som záverečnú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 17.04.2024

.....

Vlastnoručný podpis

Podakovanie

Na tomto mieste by som rád poďakoval svojmu vedúcemu práce za jeho čas a odborné vedenie počas riešenia mojej záverečnej práce.

Rovnako by som sa rád poďakoval svojim rodičom a priateľom za ich podporu a povzbudzovanie počas celého môjho štúdia.

Obsah

Úvod	1
1 Analýza patologického hlasu	3
1.1 Typy hlasových porúch	3
1.2 Metódy analýzy patologického hlasu	3
1.3 Možnosti predikcie Parkinsonovej choroby z hlasu	6
2 Neurónové siete	8
2.1 Základy neurónových sietí	8
2.2 Konvolučné neurónové siete	16
2.3 Prenosové učenie	17
3 Doménové zovšeobecnenie	19
3.1 Význam doménového zovšeobecnenia	19
3.2 Dátové augmentácie	21
3.3 Automatické augmentácie	23
4 Zovšeobecnenie pomocou augmentácií zvukového signálu	28
4.1 Dátové sady pre detekciu Parkinsonovej choroby z reči	28
4.2 Architektúra klasifikačného modelu	29
4.3 Grafická reprezentácia zvukových nahrávok	32
4.4 Techniky augmentácie zvukových dát	32
4.4.1 Zmena rýchlosti	34
4.4.2 Zmena výšky tónu	34
4.4.3 Časový posun	35
4.4.4 Pridanie šumu	36
4.5 Automatická dátová augmentácia	36
4.6 Implementácia riešenia	39

5	Výsledky experimentov	42
5.1	Analýza prístupu na jednej samohláske	43
5.1.1	Trénovanie na sade SP	43
5.1.2	Trénovanie na sadách SP a EN	44
5.2	Analýza prístupu na spoločných úlohách	46
5.2.1	Trénovanie na sade SP	46
5.2.2	Trénovanie na sade CZ	48
5.2.3	Trénovanie na sade EN	49
5.2.4	Trénovanie na sade IT	50
5.3	Analýza prístupu na všetkých úlohách	51
	Záver	53
	Literatúra	55
	Zoznam príloh	60

Zoznam obrázkov

1.1	Zvukový signál z pohľadu časovej domény	4
1.2	Zvukový signál z pohľadu frekvenčnej domény	4
1.3	Zvukový signál v podobe spektrogramu	5
1.4	Porovnanie Mel (naľavo) a MFCC (napravo) spektrogramov	5
2.1	Neurón - vstupy a výstupy	8
2.2	Prehľad aktivačných funkcií	9
2.3	Plne prepojená neurónová sieť so vstupnou, výstupnou a dvomi skrytými vrstvami	10
2.4	Problémy pri učení neurónovej siete	14
2.5	Konvolúcia a združovanie	17
3.1	AutoAugment - prehľadávaný priestor	25
3.2	RandAugment - ukážka augmentácií	26
3.3	TrivialAugment - ukážka použitia	26
3.4	Porovnanie techník automatickej augmentácie	27
4.1	Architektúra modelu použitého na experimenty v tejto práci	30
4.2	Porovnanie modelov s rôznymi počtami vrstiev a neurónov	31
4.3	Príklad vzorky vo forme spektrogramu	33
4.4	Augmentácia zvukového signálu zrýchlením	34
4.5	Augmentácia zvukového signálu zvýšením tónu	35
4.6	Augmentácia zvukového signálu časovým posunom	35
4.7	Augmentácia zvukového signálu pridaním šumu	36
4.8	Diagram navrhovaných metód zovšeobecnenia aplikovaním aug- mentácie na zvukový signál	37

Zoznam tabuliek

4.1	Porovnanie obsiahnutých úloh a počtov účastníkov v použitých dátových sadách	29
4.2	Výsledky najlepšieho základného modelu	32
5.1	Výsledky tréningovania na SP (scenár 1)	44
5.2	Výsledky tréningovania na SP+EN (scenár 1)	45
5.3	Výsledky tréningovania na SP (scenár 2)	47
5.4	Výsledky tréningovania na CZ (scenár 2)	48
5.5	Výsledky tréningovania na EN (scenár 2)	49
5.6	Výsledky tréningovania na IT (scenár 2)	50
5.7	Výsledky tréningovania na SP (scenár 3)	51

Zoznam skratiek

CZ CzechPD.

EN RMIT-PD.

FN False Negatives.

FP False Positives.

FPR False Positive Rate.

HC Healthy Controls.

IT ItalianPVS.

MAE Mean Absolute Error.

MSE Mean Squared Error.

PD Parkinson's Disease.

SP PC-GITA.

TN True Negatives.

TP True Positives.

TPR True Positive Rate.

Úvod

V súčasnej dobe je v oblasti spracovania hlasu kladený dôraz na využitie techník umelej inteligencie na rozličné úlohy od rozpoznávania reči až po identifikáciu zdravotného stavu z rečových dát. Jedným z týchto zdravotných stavov je Parkinsonova choroba, progresívne neurodegeneratívne ochorenie, ktoré ovplyvňuje aj mnoho aspektov reči. Voľba témy práce pramení z autorovej afinity k medicínskym aplikáciám strojového učenia a zároveň voľne nadväzuje na jeho bakalársku prácu, v ktorej sa otvorili príležitosti na vylepšenie s ohľadom na potenciálne nasadenie modelu na predikciu Parkinsonovej choroby.

Cieľom tejto práce je skúmať a implementovať metódy automatickej dátovej augmentácie pre účely predikcie Parkinsonovej choroby z hlasu. Dátová augmentácia je spôsob rozšírenia existujúcich dát, často pomocou aplikácie rôznych transformácií, so zámerom zlepšiť výkonnosť modelov strojového učenia. Medicínska oblasť z pohľadu strojového učenia je známa menším množstvom dostupných dát na učenie modelov, preto má zmysel uvažovať o doménovom zovšeobecnení ako o prostriedku na zvýšenie robustnosti neurónových sietí v rôznych prostrediach. Práca sa zameriava na zovšeobecnenie medzi štyrmi dátovými sadami na predikciu Parkinsonovej choroby z hlasu v rôznych jazykoch.

V práci sú monitorované metódy analýzy hlasu, zároveň sú priblížené aktuálne spôsoby augmentačných techník i automatizovaných prístupov na dosiahnutie doménového zovšeobecnenia - zlepšenia schopnosti modelov rozpoznávať a interpretovať určité vlastnosti, špecificky hlasové, ktoré môžu byť indikátorom Parkinsonovej choroby. Práca sa preto zaoberá teoretickým pozadím, metodikou aj praktickými aplikáciami automatickej dátovej augmentácie.

Na základe vykonanej analýzy bude navrhnutý prístup na báze neurónových sietí pre diagnostiku Parkinsonovej choroby z hlasu. Ďalšími krokmi budú návrh a realizácia metód pre zovšeobecniteľnosť tohto modelu s použitím získaných znalostí postupov automatickej augmentácie. Splnenie úloh práce bude experimentálne overené, pričom na základe výsledkov bude vyhodnotená efektívnosť navrhnutého prístupu.

Formulácia úlohy

Táto práca sa zameriava na jeden z problémov modelov na predikciu Parkinsonovej choroby z rečových dát - nízka efektivita na iných dátových sadách oproti sade použitej na učenie modelu. Cieľom práce je navrhnúť a vytvoriť riešenie problému založené na automatickej dátovej augmentácii. Prvou úlohou je navrhnutie prístupu na báze neurónových sietí pre diagnostiku Parkinsonovej choroby z hlasu. Následne je potrebné overiť zovšeobeciteľnosť tohto prístupu na viacerých dátových sadách. Súčasťou riešenia má byť návrh, ako aj experimentálne vyhodnotenie metód pre zovšeobeciteľnosť modelov. Posledným cieľom je zhodnotenie a spracovanie výsledkov.

1 Analýza patologického hlasu

Rečový prejav poskytuje v medicínskom výskume možnosti na stanovenie diagnózy, resp. vyhodnotenie stavu pacienta. Analýza hlasu prebieha s využitím pokročilých techník a metód analýzy zvukového signálu na monitorovanie rôznych hlasových porúch a ochorení. Pojem **patologický hlas** je používaný na charakteristiku hlasu s rôznymi abnormalitami spôsobenými určitými podmienkami alebo poruchami.

1.1 Typy hlasových porúch

Hlasové poruchy môžu byť rozdelené do troch kategórií, uvádza Martin [1]:

- **funkčné** - hlasový aparát je normálny, ale pacient má problém s jeho použitím, napr. v dôsledku nedostatočného ovládania svalov hlasiviek,
- **organické** - narušená štruktúra hlasiviek alebo pľúc v dôsledku výrastkov, či neurologických porúch,
- **psychogénne** - môžu vznikajúť dôsledkom depresie a úzkosti, sú zriedkavé.

Bežné poruchy hlasu bývajú príznačné hlasom, ktorý znie drsne, chrapľavo, dýchavo, trasľavo, s prestávkami alebo pauzami vo zvuku. Taktiež sem patrí hlas, ktorý znie príliš vysoko, či príliš nízko.

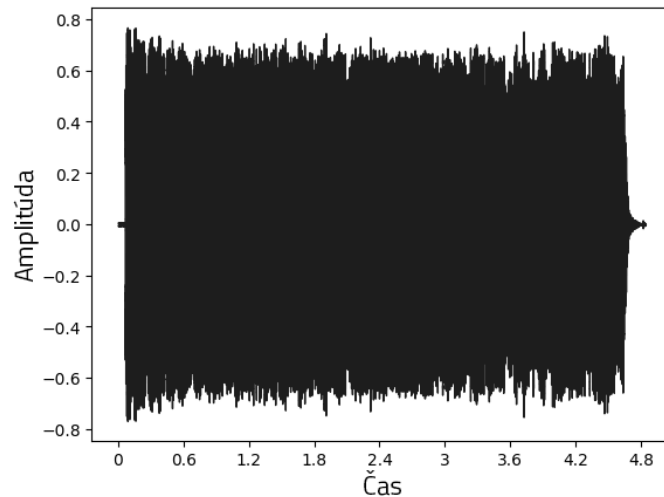
Diagnostika hlasových porúch zvyčajne zahŕňa dôkladné vyšetrenie s ohľadom na anamnézu pacienta. Tiež môžu byť využité zobrazovacie vyšetrenia napr. na detekciu výrastkov.

1.2 Metódy analýzy patologického hlasu

Prvým pohľadom na možnosti analýzy zvukového signálu sú tzv. domény, ktoré znázorňujú vlastnosti signálu vo forme obrázku. Pre ilustráciu rozdielov medzi jednotlivými doménami je uvedený rovnaký zvukový signál - vyslovenie samohlásky **a** po dobu niekoľkých sekúnd.

1. Časová doména

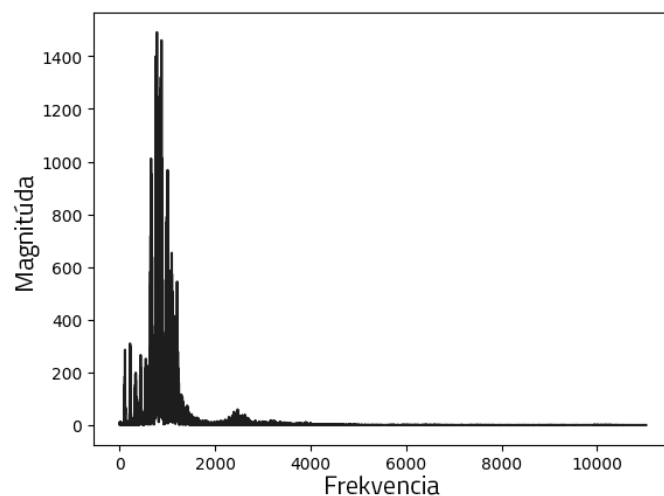
Z grafu na obrázku 1.1 je možné odčítať vývoj hodnoty amplitúdy v čase. Takýto pohľad však poskytuje len limitované možnosti na zachytenie zložitých vlastností zvukového signálu pre analýzu hlasu, preto je žiadúca transformácia na frekvenčnú doménu.



Obr. 1.1: Zvukový signál z pohľadu časovej domény

2. Frekvenčná doména

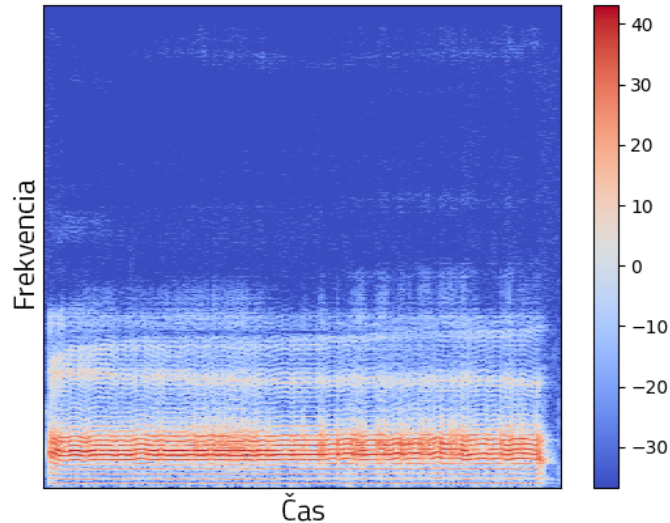
Využitím Fourierovej transformácie je možné z časovej domény získať frekvenčnú doménu, ktorá znázorňuje spektrum magnítúd jednotlivých frekvencií, ako napr. na obrázku 1.2. Znovu však neposkytuje ucelený obraz o signáli pre analýzu hlasu, napr. z dôvodu dynamiky hlasu.



Obr. 1.2: Zvukový signál z pohľadu frekvenčnej domény

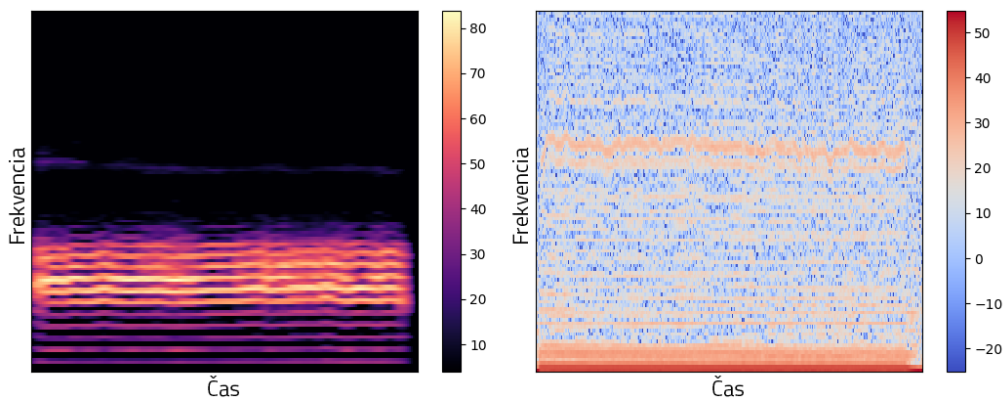
3. Časovo-frekvenčná doména

Ponúka informácie o výskyte individuálnych frekvencií v čase a zároveň o ich intenzite. Aplikovaním krátkodobej Fourierovej transformácie vzniknú *spektrogramy*, ako na obrázku 1.3, ktoré ponúkajú viac informácií ako predchádzajúce prístupy.



Obr. 1.3: Zvukový signál v podobe spektrogramu

Prítomnosť určitých prvkov, napr. prasklín v zvuku môže naznačovať prítomnosť poruchy hlasu. I keď niektoré abnormality môžu byť na prvý pohľad evidentné z grafu signálu, zvyčajne je vhodnejšie (aj z dôvodu automatizácie) zapojiť strojové učenia na detekciu a klasifikáciu porúch na báze tréningových dát. Súčasné prístupy využívajú najmä konvolučné neurónové siete na automatickú extrakciu príznakov a následnú klasifikáciu. Na účely tréningovania modelov neurónových sietí je najčastejšie používaná časovo-frekvenčná doména.



Obr. 1.4: Porovnanie Mel (naľavo) a MFCC (napravo) spektrogramov

Medzi najčastejšie používané zobrazenia pre hlboké učenie patria Mel spektrogramy a MFCC (z angl. *mel-frequency cepstral coefficients*) spektrogramy. Ako uvádza Zhang [2], Mel spektrogramy využívajú prevod do tzv. Mel-frekvenčnej škály a tak je možné zvýrazniť detaily frekvencií signálu v čase. MFCC je možné získať dodatočným aplikovaním diskkrétnej kosínusovej transformácie. Obrázok 1.4 ilustruje rozdiely medzi týmito dvoma typmi spektrogramov.

1.3 Možnosti predikcie Parkinsonovej choroby z hlasu

Vzhľadom na to, že jedným z cieľov tejto práce je preskúmať prístupy na zlepšenie predikcie Parkinsonovej choroby z rečových dát, sa v nasledujúcej časti nachádzajú východiská práce. Tieto zahŕňajú aktuálny stav problematiky, detailný popis problému a uvádzanie relevantných prístupov k riešeniu podobných úloh.

Parkinsonova choroba je druhé najrozšírenejšie neurodegeneratívne ochorenie. Zasahuje časť mozgu zodpovednú za tvorbu dopamínu. Parkinsonova choroba sa v rôznych štádiách prejavuje motorickými aj rečovými abnormalitami [3]. Keďže v súčasnosti neexistuje laboratórne vyšetrenie krvného obrazu schopné diagnostikovať Parkinsonovu chorobu, tradične je používaný komplexný postup zahŕňajúci anamnézu pacienta a neurologické vyšetrenie. Aby bolo možné tento proces zjednodušiť, boli navrhnuté riešenia založené na modeloch neurónových sietí, ktoré diagnostikujú Parkinsonovu chorobu na základe analýzy rukopisu, sledovania pohybových vlastností alebo rečových dát. Posledný spomínaný prístup je typicky realizovaný pomocou konvolučných neurónových sietí, ktoré spracovávajú rečové dáta prevedené do obrázkovej podoby využitím napr. Fourierovej analýzy.

Jedným z problémov, ktorým čelí nielen hlboké učenie za pomoci neurónových sietí, ale aj strojové učenie ako celok, je pretrénovanie. Tento pojem znamená, že model nie je schopný presnej predikcie pri doposiaľ nevidenej vzorke, t.j. vzorke, ktorá nebola použitá pri učení modelu. Pri neurónových sieťach zameraných na analýzu biosignálov, prípadne analýzu reči, je možné pozorovať, že model zas môže dosahovať vynikajúcu presnosť na dátovej sade, na ktorej bol trébovaný, ale jeho výkon na inej dátovej sade, ktorá sa zameriava na riešenie rovnakého problému, len obsahuje rozdiely v niektorých charakteristikách, je neuspokojivý. Všeobecne sa tento problém nazýva doménový posun (z angl. *domain shift*) a proces riešenia sa nazýva doménová generalizácia/doménové zovšeobecnenie (z angl. *domain generalization*) [4].

Pre ilustráciu analyzovanej problematiky diagnostiky Parkinsonovej choroby

z rečových dát je možné uviesť nasledujúci príklad. Pokiaľ je model natrénovaný na dátovej sade obsahujúcej vzorky získané od španielsky hovoriacich pacientov, je očakávané, že dosahuje najlepšiu presnosť pri predikcii na podobných vzorkách. Pokiaľ sa vzorky na predikciu začnú odlišovať, či už regiónom, prízvukom, alebo až iným jazykom, presnosť predikcie prudko klesá. Toto potvrdzuje aj článok od autorov Hireša, Drotára a ďalších [5].

Výsledné komplikácie sú z nemalej časti spôsobené skutočnosťou, že dát pre tréning daných modelov je pomerne málo. Malé množstvo dát na učenie je známym nedostatkom neurónových sietí.

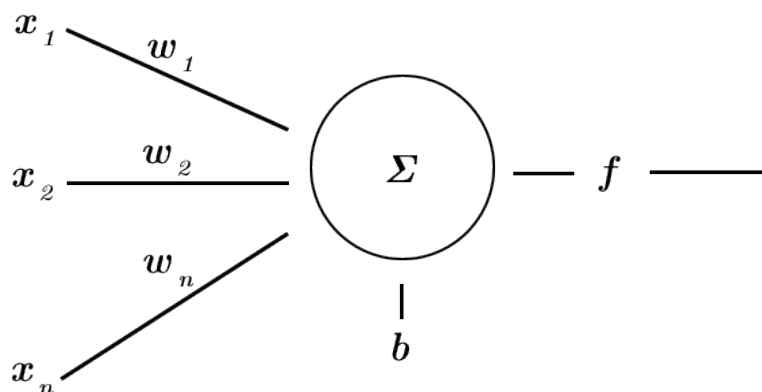
Hlavným zámerom tejto práce je objaviť adekvátne riešenie pre generalizáciu modelu určeného na diagnostiku Parkinsonovej choroby z rečových dát. Existujú rôzne prístupy k riešeniu podobných problémov v iných oblastiach. Jedným z nich je technika premietnutia obrázkov na povrch gule, ktorá sa používa pri segmentácii snímok očnej sietnice, ako uvádza Lyu [6]. Ďalší postup, navrhnutý Pandeyom [7], zahŕňa posun rámcov krátkodobej Fourierovej transformácie v kontexte spracovania reči.

2 Neurónové siete

Neurónová sieť v oblasti hlbokého učenia je model strojového učenia zložený z vrstiev obsahujúcich neuróny. Pôvod tohto označenia odkazuje na biologický centrálny nervový systém. Neurón ako bunka zodpovedá za prenos informácie vo forme elektrického signálu, uvádza Kittner [8]. Tento princíp je uplatnený aj v prípade neurónov umelých neurónových sietí (ďalej len neurónové siete). Prvé neurónové siete začali vznikať v 50-tych rokoch 20. storočia pod vedením F. Rosenblatta na Cornell University, uvádza vo svojom článku Fradkov [9]. Prvá neurónová sieť bola nazvaná „perceptron“, čo znamená stroj schopný vnímania a rozpoznávania.

2.1 Základy neurónových sietí

Ako bolo naznačené v úvode kapitoly, základnou jednotkou každej neurónovej siete je neurón.



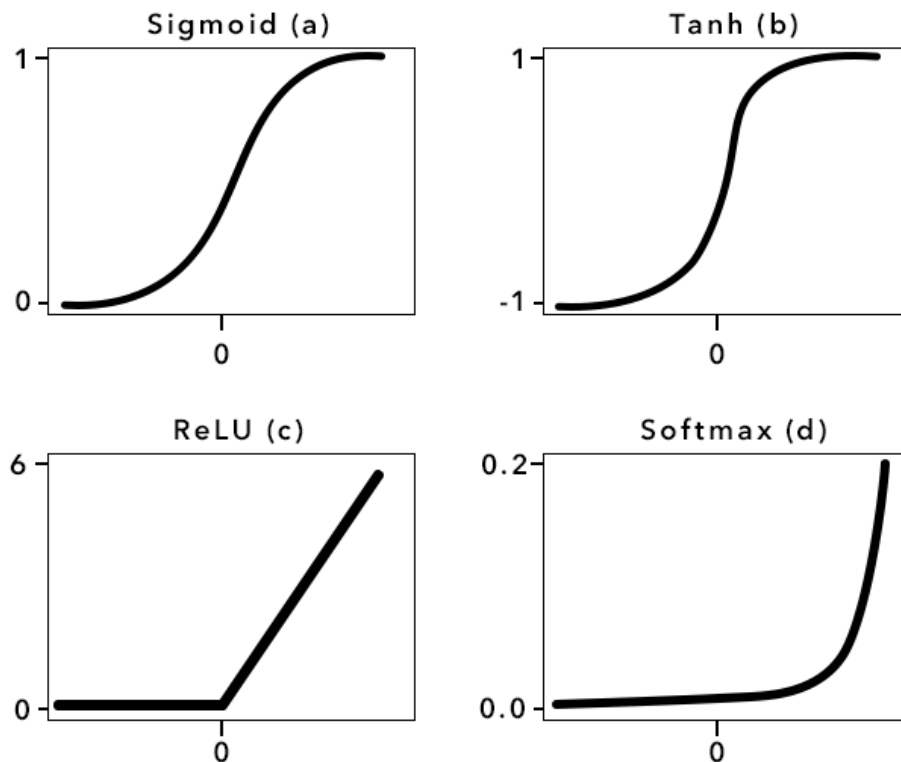
Obr. 2.1: Neurón - vstupy a výstupy

Obrázok 2.1 ilustruje prípad, kde do neurónu vstupujú vstupné dáta (x), pričom každé spojenie má individuálnu váhu (w). V niektorých prípadoch je žia-

dúce hodnotu neurónu posunúť pomocou špeciálnej váhy, nazvanej *bias* (b).

$$y_k = f \left(\sum_{i=0}^n w_{ki} x_i \right) \quad (2.1)$$

Funkcia 2.1 je matematickým popisom správania takéhoto neurónu, známym tiež ako *lineárna kombinácia*. Vypočítava súčet (Σ) jednotlivých vstupov (x), ktoré sú násobené príslušnými váhami (w). Posledným krokom pred posunutím výslednej hodnoty je prechod tzv. aktivačnou funkciou (f).



Obr. 2.2: Prehľad aktivačných funkcií

Na obrázku 2.2 sú zobrazené typické aktivačné funkcie:

- **Sigmoid** (a),

Jej rozsah je od 0 do 1, preto je vhodná na použitie pri určovaní pravdepodobnosti, napr. aj pri binárnej klasifikácii.

- **Tanh** (b),

Tvar tejto funkcie je podobný ako sigmoid (typu s), no rozsah hodnôt je od -1 do 1.

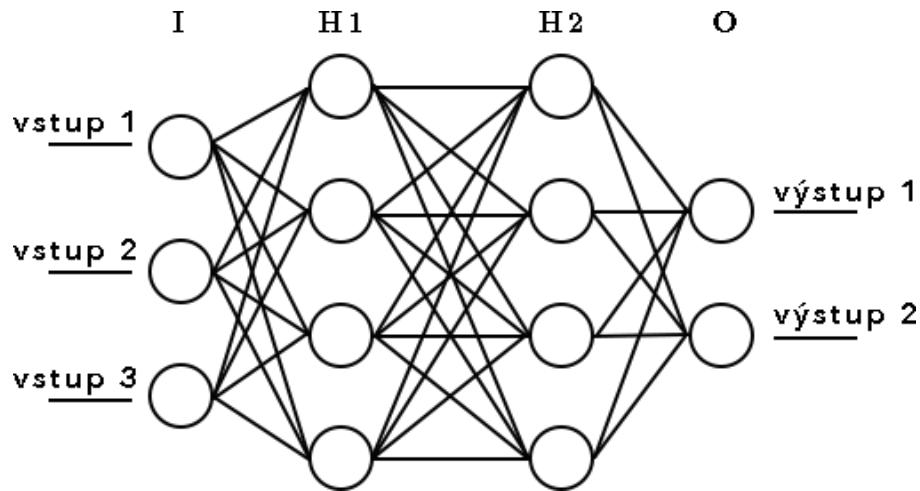
- **ReLU** (*Rectified Linear Unit*) (c),

Odrezáva záporné hodnoty a jej rozsah je od 0 do nekonečna.

- **Softmax** (d).

Používa sa pri zaradení do jednej z viacerých (aspoň troch) tried.

Architektúra neurónových sietí



Obr. 2.3: Plne prepojená neurónová sieť so vstupnou, výstupnou a dvomi skrytými vrstvami

V sieti sú neuróny usporiadané vo vrstvách, ako znázorňuje obrázok 2.3. Keďže každý neurón je prepojený s každým v ďalšej vrstve, označuje sa takáto vrstva ako *plne prepojená*. Prvá vrstva sa nazýva **vstupná** (I), kde počet neurónov korešponduje s počtom sledovaných príznakov (angl. *features*) v dátach, ako napríklad rozmer obrázka alebo počet stĺpcov tabuľky v dátovej sade.

Údaje zo vstupnej vrstvy následne prechádzajú do **skrytých** vrstiev (H1, H2...), ktoré umožňujú zmeny vstupných dát potrebné na určenie výstupu. Tieto vrstvy sú do veľkej miery zodpovedné za schopnosť siete naučiť sa zložité úlohy a dosahovať výbornú efektivitu, preto je dôležité ich správne nastavenie. Ich početnosť závisí od zložitosti riešeného problému, pri jednoduchších úlohách zväčša postačí jedna až dve skryté vrstvy.

Poslednou je **výstupná** vrstva (O), ktorá produkuje výsledok pre dané vstupy. Jednotlivé neuróny reprezentujú konkrétny výstup, na ktorého predikciu je sieť navrhnutá. Sieť klasifikujúca desať tried (napr. identifikácia rukou písaných čífer) bude mať výstupnú vrstvu s desiatimi neurónmi. Spomedzi nich neurón s najvyššou hodnotou indikuje predikovanú triedu. Výber aktivačnej funkcie pri tejto vrstve závisí od typu úlohy. Pokiaľ sa jedná o binárnu klasifikáciu, väčšinou je využitá funkcia sigmoid, v prípade klasifikácie s väčším počtom tried softmax.

Učenie neurónových sietí

Predikcia výsledku neurónovej siete spočíva v prechode dát cez sieť jednotlivými vrstvami. Jednotlivé neuróny sú prepojené spojeniami s váhami a práve hodnoty týchto váh sú rozhodujúce. Ich optimalizácia je podstatou učenia neurónovej siete. Pri inicializácii má podstatný dopad aj určenie počiatočných hodnôt váh. Toto nastavenie má vplyv na rýchlosť učenia siete ako aj na samotnú kvalitu finálneho naučeného modelu, preto existuje niekoľko spôsobov, ako k tomu pristúpiť.

Jeden z najrozšírenejších prístupov je náhodné nastavenie. Pri tomto spôsobe sú váhy nastavené na veľmi malé náhodné hodnoty, čo má za následok narušenie symetrie a zaručuje, že neuróny sa učia rozlišné veci. Tiež je dôležité, aby hodnoty boli malé, pretože v opačnom prípade môže dôjsť k tomu, že neuróny budú spomaľovať proces učenia.

Ďalším spôsobom je Xavierova/Glorotova inicializácia pomenovaná po jej autorovi [10]. V tomto prípade sa pri nastavovaní hodnôt berie do úvahy veľkosť predchádzajúcej vrstvy. Hodnoty váh sú vybrané z rozdelenia, ktorého stredná hodnota je 0 a rozptyl je závislý od počtu vstupných a výstupných neurónov, čo pomáha správne šíreniu dát cez sieť.

Podobnou technikou je aj He inicializácia, navrhnutá v článku He, Zhanga a ďalších [11]. Jej princíp je podobný ako Glorotovo nastavenie, len využíva rozdielny rozsah pre rozptyl hodnôt. Vychádza z funkcie ReLU, ktorá vynuluje záporné hodnoty a rozptyl je tak vo výsledku polovičný. Taktiež existujú aj prístupy, keď sú všetky váhy počiatočne nastavené na rovnakú hodnotu, najčastejšie na 0 prípadne 1, no preferované sú vyššie uvedené.

Po prechode vzorky dát sieťou je predikovaný výsledok porovnaný s očakávaným. Kvantifikácia ich rozdielu je charakterizovaná stratovou funkciou (angl. *loss function*). Voľba konkrétnej stratovej funkcie závisí od typu úlohy. Pri neurónových sieťach riešiacich regresné úlohy je zvolená jedna z nasledujúcich.

- **Stredná kvadratická chyba (MSE)**

Vyjadruje odchýlku ako priemer rozdielov skutočnej (y) a predikovanej (\hat{y}) hodnoty v druhej mocnine, podľa vzorca 2.2.

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2 \quad (2.2)$$

Takto získaná stratová funkcia je citlivá na výskyt tzv. *outliers* - odľahlých hodnôt s výrazným odstupom od zvyšku hodnôt. Pokiaľ sa takáto hodnota vyskytne v priebehu tréningu modelu, mocnina na ňu zväčší dôraz.

- **Stredná absolútna chyba (MAE)**

Podobne vyjadruje priemer rozdielov skutočných a predikovaných hodnôt, no miesto druhej mocniny sa používa absolútna hodnota. Výpočet stratovej funkcie je realizovaný podľa vzorca 2.3.

$$MAE = \frac{1}{n} \sum |y - \hat{y}| \quad (2.3)$$

Oproti MSE práve naopak ignoruje odľahlé hodnoty.

- **Huberova strata**

Kombinuje predchádzajúce funkcie tak, že pokiaľ je strata menšia ako určitá hodnota δ , použije sa MSE, v opačnom prípade je použitá MAE. Matematický zápis tejto funkcie je uvedený vo vzorci 2.4.

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{pre } |y - f(x)| \leq \delta, \\ \delta \cdot (|y - f(x)| - \frac{1}{2}\delta), & \text{inak.} \end{cases} \quad (2.4)$$

Je menej citlivá na *outliers* než kvadratická chyba, uvádza [12].

V prípade riešenia klasifikačných úloh sú používané najmä:

- **binárna krížová entropia,**

Používa sa pri binárnej klasifikácii (rozdelenie na 2 triedy), je nazývaná tiež logaritmická strata. Je vypočítaná pomocou upravených predikovaných pravdepodobností, resp. ich logaritmov, ako je uvedené vo vzorci 2.5. Upravenie predikovanej pravdepodobnosti (p) zodpovedá za zistenie nesprávnej kategorizácie vzorky. Pokiaľ bola zaradenie vzorky patriacej do triedy 1 napríklad 0,33 (zaradenie do triedy 0), upravená pravdepodobnosť ($1 - p$) by bola 0,67.

$$bCE = -\frac{1}{n} \sum y \cdot \log(p(y)) + (1 - y) \cdot \log(1 - p(y)) \quad (2.5)$$

- **kategorická krížová entropia,**

Jej využitie je podobné ako v prípade binárnej krížovej entropii, no je používaná na učenie modelov klasifikujúcich do vyššieho (tri a viac) počtu tried. Vzorec 2.6 reprezentuje zjednodušenú formu výpočtu stratovej funkcie.

$$cCE = - \sum y \cdot \log(p(y)) \quad (2.6)$$

- **maximálna pravdepodobnosť,**

Všeobecnejší koncept stratovej funkcie, ktorý nevyužíva logaritmickú stupnicu. Výpočet je realizovaný vzorcom 2.7.

$$Prob = -\frac{1}{n} \sum y \cdot (p(y)) + (1 - y) \cdot (1 - p(y)) \quad (2.7)$$

Na výber konkrétnej zo skupiny stratových funkcií neexistuje štandardizovaný postup, najlepšia pre danú úlohu býva zvolená na základe experimentov. Sieť sa následne učí nájsť minimum tejto funkcie, k čomu smeruje nastavenie váh. K tomu je používaný algoritmus spätného šírenia (angl. *backpropagation*), ktorý opakuje štyri základné kroky.

1. **Dopredný prechod**

Vzorky dát prechádzajú vrstvami siete, aby bol získaný výsledok.

2. **Výpočet straty**

Aplikácia stratovej funkcie pre určenie diskrepancie medzi predikciou siete a očakávaným výsledkom.

3. **Spätný prechod**

Algoritmus podľa gradientu stratovej funkcie určí smer úprav jednotlivých váh neurónov. Gradient reprezentuje sklon funkcie.

4. **Aktualizácia váh**

Po výpočte gradientov sú jednotlivé váhy upravené využitím niektorého z optimalizátorov.

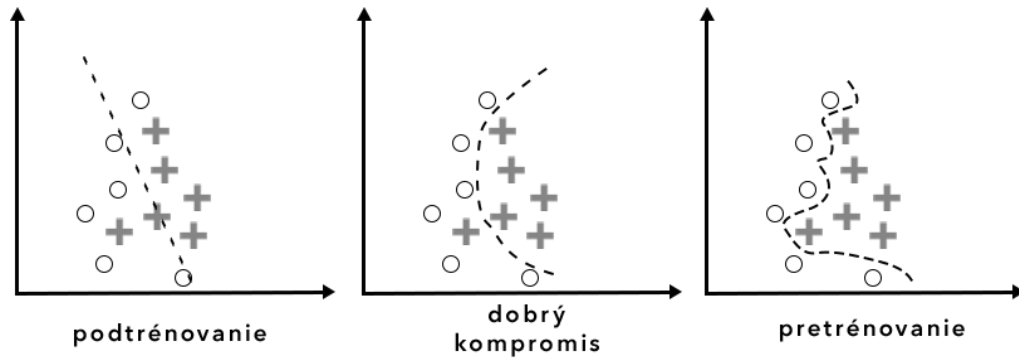
Počas tohto procesu sa vykonáva predvolený počet iterácií, prípadne môžu byť využité techniky na predčasné zastavenie učenia (napr. pokiaľ sa strata nezlepšuje po určitom počte iterácií).

Spomenuté optimalizátory sú algoritmy meniace vlastnosti neurónových sietí ako napr. rýchlosť učenia - veľkosť kroku zmien parametrov siete. Niektoré optimalizátory používajú konštantnú hodnotu rýchlosti učenia a iné sú zas *adaptívne* - rýchlosť učenia menia počas tréningu. V prehľade optimalizačných algoritmov Ruder [13] spomína:

- **gradientný zostup** (angl. *gradient descent*),
- **stochastický gradientný zostup** (angl. *stochastic gradient descent*),

- Adam [14],
- AdaAgrad [15],
- Nesterovov zrýchlený gradient [16].

Pri učení neurónových sietí môže dôjsť ku dvom hlavným problémom, zobrazeným na obrázku 2.4 - podtrénovaniu a pretrénovaniu.



Obr. 2.4: Problémy pri učení neurónovej siete podľa [17]

Podtrénovanie znamená, že model je príliš jednoduchý, nedokáže sa efektívne naučiť trénovacie dáta a jeho výkonnosť je následne nízka. Medzi riešenia tohto problému patria:

- zvýšenie zložitosti modelu (napr. pridaním vrstiev alebo neurónov),
- zvýšenie počtu sledovaných príznakov (*features*),
- predĺženie doby učenia zvýšením počtu epoch.

Opačným extrémom je **pretrénovanie**. Tento stav znamená, že model sa pri trénovaní učí zo šumu a nepresností v dátach, potom predikcia modelu je nepresná na nových dátach. Pretrénovanie je možné riešiť nasledujúcimi spôsobmi:

- **zníženie zložitosti modelu,**
- **L1 a L2 regularizácia,**
Regularizačné techniky pridávajú penalizačný člen do výpočtu stratovej funkcie, čím odkláňajú model od nadbytočnej zložitosti.
- **dropout** - ignorovanie náhodne zvolených neurónov,
- **dátová augmentácia,**
Na dáta sa aplikujú náhodné transformácie, napr. orezanie, otočenie, či zmena veľkosti v prípade obrázkových dát.
- **predčasné zastavenie učenia** (angl. *Early Stopping*).

Vyhodnotenie efektivity neurónových sietí

Počas učenia siete, ale aj po jeho skončení je možné sledovať efektívnosť siete, čo je možné vypočítať pomocou viacerých funkcií nazývaných **metriky**. V prípade sietí riešiacich regresné úlohy sa používajú napr. aj MAE a MSE spomínané v predchádzajúcej sekcii. Rozdiel medzi stratovou funkciou a metriku výkonnosti spočíva v skutočnosti, že zatiaľ čo metriky nevstupujú do procesu učenia neurónovej siete, stratová funkcia je kritickým faktorom pre učenie a priamo doň vstupuje, keďže sieť sa snaží nájsť jej minimum.

Pri klasifikačných úlohách je možné sledovať metriky (pozn. často je možné sa stretnúť aj s anglickými názvami, preto sú tiež uvedené) na základe matice zámen - tabuľky obsahujúcej informácie o počte správnych a nesprávnych klasifikácií. Pre binárnu klasifikáciu táto tabuľka obsahuje štyri údaje:

- počet správnych zaradení prvkov ako negatívnych (TN),
- počet nesprávnych zaradení prvkov ako pozitívnych (FP),
- počet nesprávnych zaradení prvkov ako negatívnych (FN),
- počet správnych zaradení prvkov ako pozitívnych (TP).

Z týchto hodnôt je možné vypočítať nasledujúce metriky:

- **accuracy** (presnosť),

$$AC = \frac{TP + TN}{TP + FP + TN + FN}$$

- **precision** (precíznosť),

$$PR = \frac{TP}{TP + FP}$$

- **recall** (úplnosť), alebo aj **sensitivity** (citlivosť, senzitivita),

$$RE = SE = \frac{TP}{TP + FN}$$

- **specificity** (špecifickosť),

$$SP = \frac{TN}{TN + FP}$$

- **F1 skóre**,

$$F_1 = \frac{2 * PR * RE}{PR + RE}$$

- **AUC - area under curve** (oblasť pod krivkou).

Senzitivitu je možné označiť aj ako TPR - mieru správnej pozitivity a jej doplnok ($1 - TPR$) ako FPR - mieru falošnej pozitivity. Pokiaľ sú na grafe vykreslené hodnoty FPR na osi x a hodnoty TPR na osi y, vznikne tzv. ROC krivka (angl. *Receiver Operating Characteristic*, operačná charakteristika prijímača). Z nej je možné získať číselnú hodnotu AUC, teda oblasť pod ňou.

2.2 Konvolučné neurónové siete

Konvolučná neurónová sieť je typ neurónovej siete, ktorá je primárne používaná na rozpoznávanie vzorov v obrázkových dátach. Ako uvádza O'Shea [18], vo svojej podstate sú konvolučné siete analogicky podobné tradičným neurónovým sieťam v tom, že rovnako obsahujú neuróny schopné automatickej optimalizácie počas učenia. Tieto siete však navyše obsahujú konvolučné vrstvy schopné identifikovať od hrán a rovných čiar až po rozličné tvary na obrázku, preto sú ideálne na riešenie klasifikačných problémov v prípadoch s obrázkovými dátami.

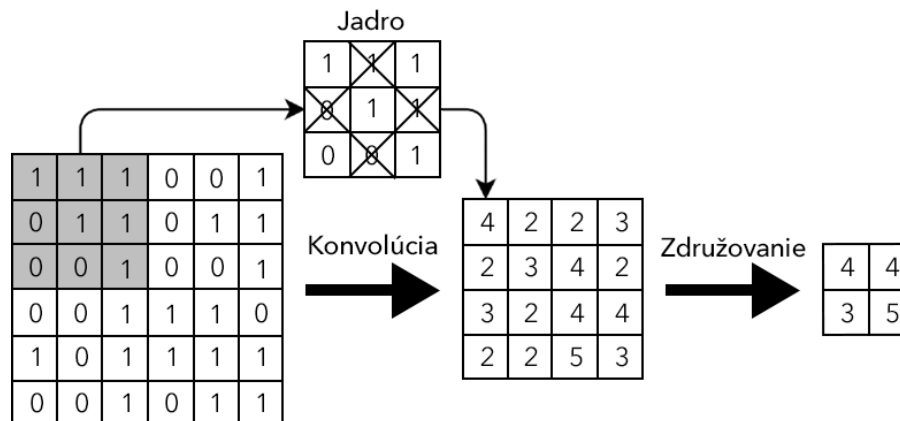
Spomínané konvolučné vrstvy fungujú na princípe posúvania konvolučného jadra alebo tiež konvolučného filtra (matice určitej veľkosti) cez vstupný obrázok, čo vytvára mapu príznakov, ktorá je potom vstupom pre ďalšie vrstvy. Samotná operácia konvolúcie je skalárnym súčinom matíc medzi jadrom a príslušnou časťou obrázka. Konvolučné vrstvy sú tiež ladené počas učenia siete tak, aby zvýraznili podstatné vlastnosti, či zložité tvary na obrázku. Hlavnými parametrami pre konvolučné vrstvy sú:

- **počet filtrov** - každý filter extrahuje rôzne príznaky,
- **veľkosť konvolučného jadra**,
- **krok posunu filtra**,
- **výplň** (angl. *padding*)

Aby nedošlo k strate dát na okrajoch obrázku, sú obrázky rozšírené ďalším pixelmi, zvyčajne s nulovými hodnotami.

Okrem samotných konvolučných vrstiev sú do modelov zahrňané aj združovacie (angl. *pooling*) vrstvy. Ich úlohou je zmenšiť dimenzionalitu dát z konvolučných vrstiev a tak zrýchliť výpočty. Fungujú tak, že z jednotlivých oblastí vyberajú jednu hodnotu, najčastejšie najväčšiu alebo priemernú. Na rozdiel od konvolučných, či plne prepojených vrstiev, združovacie vrstvy neobsahujú parametre, ktoré by sa aktualizovali počas učenia, uvádza Raschka [17].

Princíp týchto dvoch hlavných operácií ilustruje obrázok 2.5. Po vstupnej matici je posúvané konvolučné jadro s veľkosťou 3×3 a krokom posunu 1. Výsledok konvolúcie je v tomto prípade súčet čísel na pozíciách, ktoré nie sú v jadre preškrtnuté. Nasleduje združovanie typu *max* (volí najväčšiu hodnotu) veľkosti 2×2 a krokom posunu 2.



Obr. 2.5: Konvolúcia a združovanie

2.3 Prenosové učenie

Prenosové učenie (angl. *transfer learning*) je technika pre využitie znalostí naučených z jednej úlohy alebo dátovej sady pri inej úlohe. Vedomosti naučené zo zdrojovej domény sú v procese prenosového učenia aplikované na cieľovú doménu. Ako uvádza Tan, tento prístup má pozitívny efekt na mnoho domén, pri ktorých je ťažké zlepšiť výkonnosť pre nedostatok dát na učenie [19]. Okrem toho patrí medzi nepochybné benefity prenosového učenia aj šetrenie výpočtových zdrojov. Použitie predtrénovaného modelu znamená, že sieť sa nemusí učiť rozpoznávať napr. základné tvary (obsiahnuté v zdrojovej doméne), a tak dochádza k úspore času.

Model využívajúci prenosové učenie sa skladá z dvoch častí. Prvou je samotný predtrénovaný model, ktorý disponuje znalosťou domény a typicky bol učený na obrovskej dátovej sade. Keď dáta prejdú týmto modelom, prechádzajú do druhej časti - klasifikátora, ktorý je navrhnutý na riešenie cieľového problému. Aby učenie bolo efektívne, dochádza k jemnému ladeniu (angl. *fine-tuning*), čo znamená, že váhy predtrénovaného modelu sú „zmrazené“, t.j. nemenia sa počas učenia a jedine váhy klasifikátora sú menené na základe dátovej sady.

V prípade konvolučných neurónových sietí je možné aplikovať prenosové učenie na klasifikáciu obrázkov, kde je predtrénovaný model naučený napr. na sade

ImageNet¹ alebo pri úlohách na detekciu objektov, či sémantickú segmentáciu. V praxi môže prenosové učenie vyzeráť napríklad nasledovne. Je k dispozícii malá množina fotografií jednotlivých druhov zvierat a cieľom je natrénovať klasifikátor, ktorý ich má rozoznávať. V tomto prípade existuje možnosť použiť predtrénovaný model, ktorý bol trénovaný na veľkej sade obrázkov znázorňujúcich rôzne objekty. Tento model je možné jemne ladiť na konkrétnu riešenú úlohu (identifikácia zvierat), čo umožňuje dosiahnuť dobré výsledky aj s obmedzeným množstvom dát.

Prenosové učenie síce prináša mnoho výhod, no nesmie sa aplikovať bezhlavo, pretože môže spôsobiť určité komplikácie. Jedným z takýchto problémov je riziko prenesenia skreslenia zdrojovej domény na cieľovú doménu. To znamená, že ak sú dáta v zdrojovej doméne nepresné alebo skreslené, môže to ovplyvniť aj výkon modelu na cieľovej doméne. Preto je výber zdrojovej domény kľúčový, aby čo do najväčšej miery korešpondovala s cieľovou doménou. Ako ilustráciu je možné uviesť predpoklad, že pokiaľ model bol trénovaný na klasifikáciu mačiek a psov, prenos jeho váh do modelu na detekciu áut a dopravného značenia s veľkou pravdepodobnosťou nepovedie k lepším výsledkom.

¹ImageNet je databáza vyše 14 miliónov označených obrázkov navrhnutá na rozpoznávanie objektov [20].

3 Doménové zovšeobecnenie

Problematika doménového zovšeobecnenia predstavuje kľúčový koncept v oblasti hlbokého učenia a prenášania poznatkov. Aj keď zovšeobecnenie je schopnosťou, ktorá je prirodzená pre ľudí, pre stroje je jej replikácia náročná. Ako uvádza Wang [21], doménové zovšeobecnenie je schopnosť modelu transformovať svoje poznatky a naučené zručnosti na nové, neznáme údaje alebo domény, ktoré sú odlišné od tých, na ktorých bol model trénovaný. V súčasnosti je tento koncept intenzívne študovaný, pretože jeho úspešné zvládnutie môže viesť k vytváraniu univerzálnejších a robustnejších modelov.

Okrem doménového zovšeobecnenia je možné stretnúť sa aj s pojmom doménová adaptácia. Je preto potrebné zdôrazniť, že napriek tomu, že dané koncepty spolu súvisia, nie sú identické. Zatiaľ čo doménová adaptácia (*angl. domain adaptation*) dokáže využiť neoznačené cieľové dáta (bez labels), doménové zovšeobecnenie (*angl. domain generalization*) touto schopnosťou nedisponuje, uvádza Ghifary [22].

3.1 Význam doménového zovšeobecnenia

Doménové zovšeobecnenie má význam predovšetkým v oblastiach ako medicína, autonómne riadenie a robotika. V týchto odvetviach sú častokrát modely trénované na špecifických dátových sadách, ktoré sa v praktickom nasadení môžu líšiť. Napríklad výkon modelu trénovaný na obrázkoch magnetickej rezonancie z jednej kliniky sa môže zhoršiť, ak sa použije na obrázky z iného zariadenia s odlišnou kalibráciou. Doménové zovšeobecnenie sa snaží umožniť modelom efektívne prispôbienie sa novým podmienkam a následne zvýšenú spoľahlivosť výsledkov v rôznych prostrediach.

Na dôležitosť doménového zovšeobecnenia dáva dôraz aj štúdia odborníkov z talianskej univerzity Politecnico di Torino [23]. Tento systematický prehľad literatúry analyzoval viac ako 300 článkov publikovaných v rokoch 2018 až 2022, ktoré sa zameriavali na používanie techník dátových augmentácií v medicínskom

zobrazovaní. Poskytol rozsiahlu kvalitatívnu a kvantitatívnu analýzu, ako aj porovnanie s predchádzajúcimi prieskumami a experimentálnymi štúdiami. Okrem toho prehľad poukázal na súčasné medzery vo výskume a smery budúceho zlepšenia, čím poukázal na potenciálne možnosti budúceho výskumu v tejto živej oblasti. Zo zistení prehľadu vyplýva, že stratégie rozšírenia údajov majú významný potenciál na zlepšenie úloh analýzy medicínskych obrazov a v budúcnosti si vyžadujú ďalšie skúmanie a zdokonaľovanie.

Doménové zovšeobecnenie rieši podstatné výzvy strojového učenia, ako napríklad:

- **Zdrojové a cieľové domény**

Rozdielne vlastnosti a štruktúry zdrojových a cieľových domén môžu spôsobiť, že modely založené na doménovom zovšeobecnení budú mať nedostatočnú výkonnosť. Ideálny model by mal byť schopný sa prispôbiť širokej škále domén a zdrojových úloh.

- **Nízka dostupnosť dátových sád**

Často dochádza k problému nedostatku reprezentatívnych dátových sád pre rôzne prostredia alebo domény, čo môže obmedziť schopnosť modelu prispôbiť sa novým podmienkam.

- **Nízka dostupnosť označených cieľových dát**

Nie vždy je možné získať dostatok označených cieľových dát pre cieľovú úlohu.

- **Citlivosť na zmeny v dátach**

Modely môžu byť veľmi citlivé na malé zmeny v dátach.

V oblasti doménového zovšeobecnenia sa často využívajú nasledujúce techniky a prístupy, ktoré umožňujú modelom efektívne sa prispôbiť na nové podmienky a domény:

- **Prenosové učenie**

Využitie poznatkov z jednej úlohy na riešenie iných úloh v iných prostrediach. Napríklad model trénovaný na rozpoznávanie tváří na fotografiách môže byť použitý na rozpoznávanie tváří v záznamoch z bezpečnostných kamier. Bližší popis prenosového učenia z technického hľadiska sa nachádza v predchádzajúcej kapitole (2.3).

- **Regularizácia**

Táto technika sa snaží minimalizovať fenomén pretrénovania. Medzi najčastejšie používané metódy patrí L1 a L2 regularizácia, ktoré sú schopné obmedziť veľkosť váh modelu a minimalizovať efekt pretrénovania.

- **Učenie pomocou komisií** (*angl. ensemble learning*)

Trénovanie viacerých modelov a kombinovanie ich výstupov umožňuje vytvárať robustnejšie a spoľahlivejšie predikcie. Napríklad, model trénovaný na rozpoznávanie tvárí pomocou neurónových sietí môže byť kombinovaný s modelom trénovaným na rozpoznávanie tvárí pomocou rozhodovacích stromov.

- **Dátové augmentácie**

Tento prístup spočíva v generovaní ďalších tréningových príkladov z existujúcich dát pomocou rôznych transformácií, čo umožňuje modelom sa prispôbiť na nové podmienky. V ďalšej sekcii je uvedená detailná analýza augmentačných techník.

3.2 Dátové augmentácie

Modely konvolučných neurónových sietí môžu zlyhať v úlohe zovšeobecnenia nad rámec prostredia, v ktorom sa učia, uvádza Raileanu [24]. Väčšina spôsobov na zovšeobecnenie uvedených v prechádzajúcej sekcii sa zameriava na architektúru samotného modelu neurónovej siete. Augmentovanie dát rieši problém od základov - tréningovej dátovej sady. Dátová augmentácia (rozšírenie) je technika na zväčšenie sady na učenie vytváraním upravených kópií existujúcich dát. Nová sada môže vznikáť tak, že existujúce vzorky prejdú určitým typom transformácie, alebo na základe nich generatívna sieť vytvorí syntetické dáta.

Xu a kol. uvádzajú niekoľko dôvodov na aplikáciu augmentácií (konkrétne na obrázkové dáta) [25]. Prvý z nich je zvýšenie variability dát, s cieľom simulovať rôzne napr. svetelné podmienky fotografických vzoriek. Ďalším dôvodom je nevyvážený počet vzoriek v klasifikačných triedach na základe nerovnomerného výskytu objektov. Benefit použitia augmentačných techník je evidentný aj v prípade doménového posunu. Keď je model používaný v inej doméne, než na akej bol trénovaný, je možné využiť dátové augmentácie.

Prínosy augmentácií boli tiež experimentálne potvrdené. Shorten uvádza príklad, keď použitie augmentácií spôsobilo zlepšenie výkonnosti modelu na sade

CIFAR-100 zo 66 % na 73 % [26]. Existuje množstvo augmentačných techník, pričom ich použitie závisí od typu dát.

Zvukové augmentácie

V prípade zvukových dát sú možné nasledujúce úpravy. Prostredníctvom vloženia šumu sa do zvukového signálu pridá umelo vygenerovaný šum, čo môže pomôcť ku zvýšeniu robustnosti modelu voči šumu v reálnom prostredí. Ďalej je možné posunúť vzorku doprava/dolava a tak simulovať drobné oneskorenia. Následne je možné zmeniť rýchlosť signálu, buď zrýchliť, alebo spomaliť. Táto úprava môže mať vplyv na dynamiku a intonáciu. Rovnako je možné upraviť melodické črty zvukového záznamu zmenou výšky tónu, t.j. modifikáciou frekvencie zvukových vln. Posledné dve techniky - normalizácia a kompresia dynamiky realizujú vyrovnanie amplitúdy, resp. zmenšenie rozdielov medzi jej najvyššou a najnižšou hodnotou, čím sa dosiahne konzistentná hlasitosť.

Textové augmentácie

Pokiaľ model pracuje s textovými vzorkami, je možné preusporiadať slová alebo vety. Pokiaľ je technika správne nastavená, môže viesť k vytvoreniu nových formulácií, pričom sa zachováva obsah pôvodného textu. Nahradenie slov ich synonymami spočíva v zámene určitých slov vo vzorke ich významovo ekvivalentnými slovami. Tento postup môže zvýšiť rozmanitosť slovníka a zlepšiť pochopenie textu. Nadstavbou je parafrázovanie vety, čo znamená vyjadrenie rovnakého obsahu vety inými slovami alebo vetou s odlišnou štruktúrou. Táto úprava môže pomôcť vytvoriť varianty vety so zachovaním pôvodného významu. Vloženie alebo odstránenie náhodného slova je postup ovplyvňujúci gramatiku a sémantiku textu, pričom zároveň zvyšuje jeho variabilitu.

Obrázkové augmentácie

Široký priestor augmentačných techník pre obrázkové dáta zhŕňa Wei [27], najčastejšie sem patria rôzne geometrické transformácie, ako náhodné obrátenia, orezania, otočenia a zmena veľkosti obrázkov. Takto sa môže dosiahnuť robustnosť voči odlišným pohľadom, či perspektívam. Zmeny farebného priestoru spočívajú v úpravách jedného, či viacerých farebných kanálov RGB, jas a kontrastu. Týmto spôsobom je možné zmeniť farebný tón a dynamiku obrázka. Zvýšenie ostroti a rozmazanie sú techniky, ktoré ovplyvňujú ostrosť hraníc a detaily v obrázku. Umelé pridanie filtrov rozmazania simuluje napríklad rozmazanie, ktoré vzniká

pohybom kamery, či rýchlo pohybujúceho sa objektu. Odstránenie časti obrázku umožňuje vytváranie variantov obrázkov bez určitých častí. Táto úprava môže byť užitočná pri odstraňovaní rušivých či irelevantných informácií z obrázkových dát.

Pred použitím augmentácií je potrebné zvážiť niektoré dôležité aspekty. Jedným z nich je presnosť a dôveryhodnosť syntetických dát vytvorených pomocou generatívnych modelov. Je kľúčové zabezpečiť, aby syntetické dáta boli presné a dôveryhodné, pretože ich nízka kvalita či nepresnosť môžu negatívne ovplyvniť výkon modelu. Druhým aspektom je výpočtová náročnosť tréningu generatívnych modelov a vytvárania syntetických dát. Tieto procesy môžu byť veľmi náročné na výpočtové prostriedky, preto je dôležité zvážiť, či je vhodné investovať do vytvorenia syntetických dát alebo radšej uprednostniť iné metódy dátovej augmentácie. Posledným aspektom je obmedzené zlepšenie presnosti a výkonu modelov konvolučných neurónových sietí pomocou dátovej augmentácie. Hoci využitie augmentačných techník môže v niektorých prípadoch zvýšiť úspešnosť modelov, zlepšenie môže byť aj obmedzené. Preto je dôležité zvážiť aj iné metódy na zlepšenie zovšeobecnenia, ako sú napríklad techniky regularizácie, multi-task learning alebo už spomínané prenosové učenie.

3.3 Automatické augmentácie

Aplikácia dátových augmentácií, uvedených v predchádzajúcich častiach, môže priniesť zlepšenie výkonnosti modelov neurónových sietí, no pokiaľ je vykonávaná manuálne, ide o zložitú a časovo náročnú úlohu, hlavne pri veľkých dátových sadách. Preto má význam uvažovať o metódach, ktoré umožňujú tento proces automatizovať, a tak šetriť čas i ľudské úsilie pri príprave dát. Okrem toho, automatická augmentácia otvára možnosti na dosiahnutie presnejšieho a rozšíreného prehľadávania priestoru dátových transformácií. Manuálne aplikované augmentačné techniky sú náchylné k obmedzeniu na známe a tradičné metódy, zatiaľ čo automatické metódy môžu viesť k objaveniu nových a zložitejších transformácií. Tieto potom majú potenciál viesť k ešte lepším výsledkom.

Ako uvádza Yang [28], automatické augmentácie sú založené na automatizovaných algoritmoch, ktoré hľadajú najlepšie stratégie pre transformáciu dát, aby sa zlepšila výkonnosť modelov. Tieto algoritmy sa snažia najlepšie využiť dostupné dáta a rozšíriť kvalitu dostupných dátových sád prostredníctvom rôznych transformácií. Konkrétne transformácie sú závislé od typu spracovaných dát, príklady sú uvedené v predošlej časti. Tieto procesy sú zamerané na zlepšenie schopnosti modelov na generalizáciu a nájdenie skrytých vzorov v dátach.

História vývoja metód na automatizáciu dátových augmentácií začína v roku 2017, keď bol navrhnutý systém TANDA (*angl. Transformation Adversarial Networks for Data Augmentations*) [29], ktorý bol prvým pokusom o automatické zostrojenie a následné ladenie politík dátových augmentácií na základe poskytnutých dát. Prístup využíva generatívne siete GAN (*angl. Generative Adversarial Networks*) a využíva dve siete. Jedna neurónová sieť sa učí generovať transformácie na pôvodných dátach, ktoré budú použité na augmentáciu a druhá sieť klasifikuje výstup prvej siete, či patrí do pôvodnej triedy dát. Týmto spôsobom sa generujú také transformácie, ktoré sú relevantné pre rozšírenie dátovej sady, pričom zároveň udržiavajú súvislosť s pôvodnými dátami. V ďalších častiach sú popísané tri efektívne techniky na automatickú augmentáciu.

AutoAugment

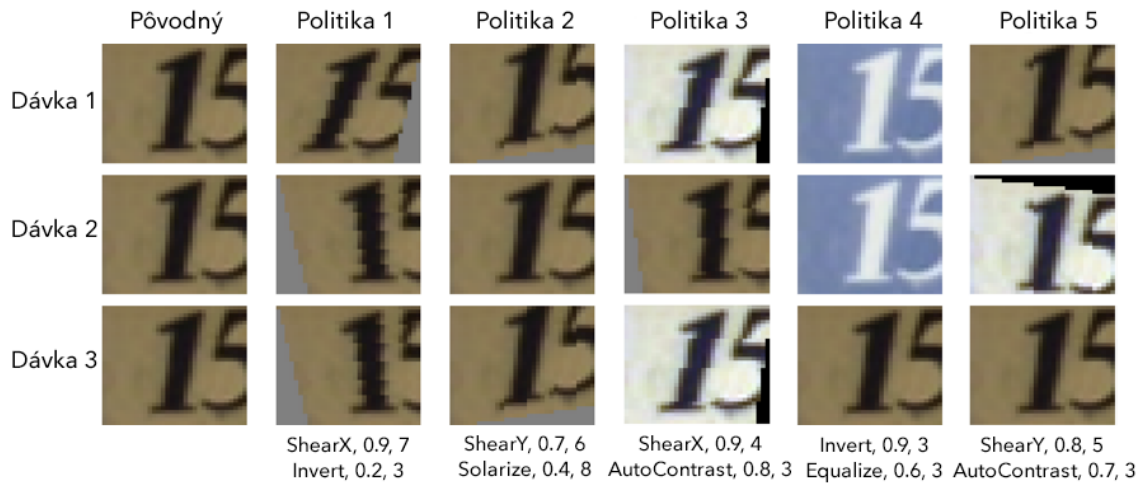
Prístup algoritmu TANDA inšpiroval návrh vyhľadávacieho priestoru v algoritme AutoAugment [30]. Táto metóda formuluje problém hľadania najlepšej politiky augmentácie ako diskretný vyhľadávací problém. AutoAugment sa skladá z dvoch komponentov: vyhľadávacieho algoritmu a vyhľadávacieho priestoru.

Vyhľadávací algoritmus, implementovaný ako rekurentná neurónová sieť, rozhoduje o tom, ktorú transformáciu použiť, pravdepodobnosť použitia pre každú dávku dát, ako aj aká magnitúda techniky bude použitá. Iteračným prístupom sú zvolené (spočiatku náhodné) stratégie a ich výsledky sú použité na aktualizáciu rekurentnej siete slúžiacej ako ovládač algoritmu. Pre každú dávku je vytvorených päť politík, ktoré sú vzájomne porovnané.

Vyhľadávací priestor je tvorený 16-imi operáciami obrázkového spracovania (napr. zmena jas, kontrastu, otočenie, atď.). Každá z nich môže nadobúdať magnitúdu z desiatich hodnôt (s rovnomerným rozmiestnením). Rovnako pravdepodobnosti pre jednotlivé operácie sú volené z 11 hodnôt. Takýto priestor tvorí približne 2.9×10^{32} rôznych možností.

Obrázok 3.1 ilustruje príklad vytvoreného prehľadávaného priestoru pre AutoAugment. Pre každú z politík je definovaná dvojica augmentácií, pričom každá augmentácia má vlastnú pravdepodobnosť aplikácie (prvé číslo) a magnitúdu (druhá hodnota) - tá je pre danú politiku vždy fixná. V závislosti od konkrétnych pravdepodobností môže byť na vzorku aplikovaná: žiadna operácia, jedna z operácií alebo obe operácie.

AutoAugment predstavuje inováciu v oblasti augmentácie dát, ktorá automaticky hľadá najlepšie augmentačné politiky. Táto metóda zlepšuje výkonnosť neurónových sietí prostredníctvom efektívnej manipulácie s dátami a demonštruje



Obr. 3.1: AutoAugment - prehľadávaný priestor podľa [30]

svoju schopnosť zlepšovať výkonnosť na rôznych dátových zoznamoch, čo ju činí cenným nástrojom pre vývojárov a výskumníkov v oblasti počítačového videnia.

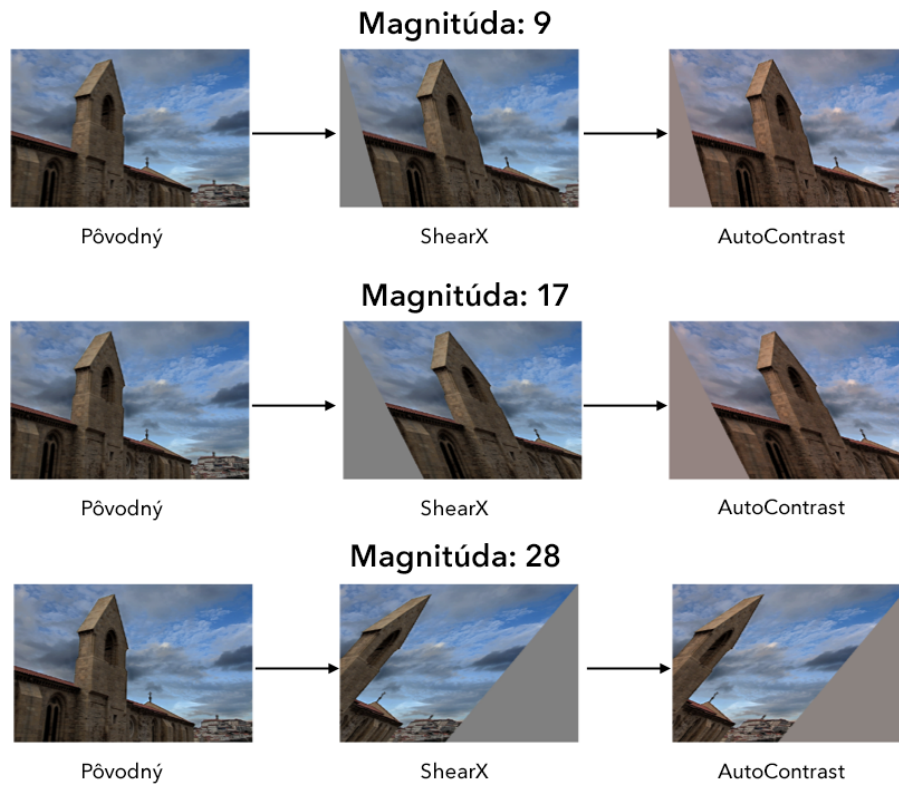
RandAugment

Ďalším automatizovaným prístupom k augmentácii dát je RandAugment, ktorý zjednodušuje proces hľadania najlepších politík augmentácie. Tento prístup bol popísaný v článku od Cubuka a kol. [31], ktorí sú autormi aj predchádzajúceho algoritmu. RandAugment je založený na náhodnom výbere a aplikácii operácií na obrázky, čo umožňuje efektívne rozšírenie dát bez nutnosti hľadania optimálnej politiky.

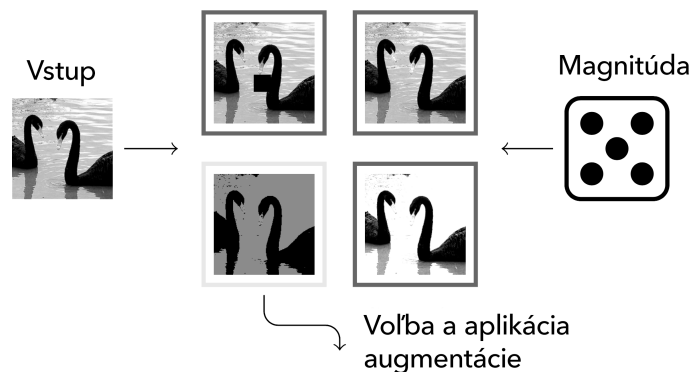
Zatiaľ čo AutoAugment využíva učenie posilňovaním (*angl. reinforcement learning*) na hľadanie najlepšej augmentačnej politiky, RandAugment predstavuje zjednodušený prístup. Jeho funkcionálnosť je založená na náhodnom výbere daného počtu operácií z definovaného súboru a následnej aplikácii na obrázky (operácia má vždy rovnakú magnitúdu). Keďže odpadá potreba využitia učenia posilňovaním, dochádza k podstatne nižšej výpočtovej zložitosti. Príklad aplikácie augmentácií algoritmom RandAugment je uvedený na obrázku 3.2.

TrivialAugment

Jedným z najnovších prístupov je TrivialAugment [32], ktorý sa dá označiť ako minimalistický. TrivialAugment aplikuje zakaždým iba jednu operáciu na každý obrázok. Ďalším rozdielom od RandAugment-u je tiež skutočnosť, že TrivialAugment určuje náhodnú silu zvolenej operácie pre každú vzorku. Týmto dosahuje ešte nižšiu výpočtovú náročnosť ako RandAugment.



Obr. 3.2: RandAugment - ukážka augmentácií podľa [31]



Obr. 3.3: TrivialAugment - ukážka použitia podľa [32]

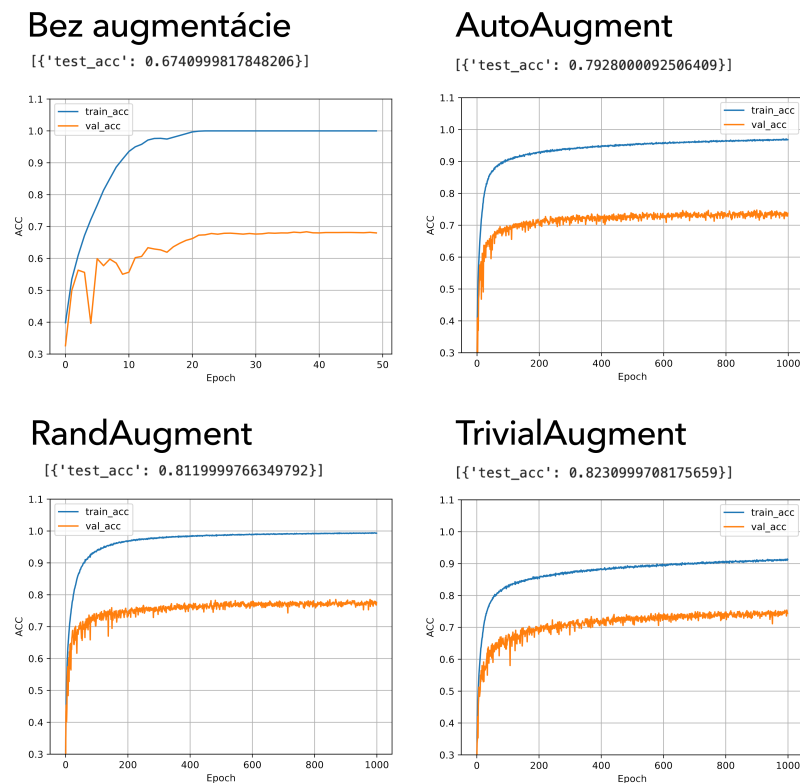
Zhrnutie

Rozdiely medzi jednotlivými algoritmi je možné zhrnúť nasledovne:

- **AutoAugment:** Hľadá augmentačnú politiku, čo mu umožňuje dynamicky kombinovať rôzne techniky augmentácie. Je to efektívny prístup, no môže byť náročný na výpočtové zdroje.
- **RandAugment:** Zjednodušený prístup, ktorý premení výber augmentačných techník na problém hľadania hyper-parametrov. Je tiež efektívny a jednoduchý na implementáciu.

- **TrivialAugment**: Aplikuje len jednu augmentáciu na obrázok so silou, ktorá je určená náhodne. Je to najjednoduchší prístup, ale je efektívny a nevyžaduje žiadne ďalšie nastavenia.

Efektivitu jednotlivých prístupov vzájomne porovnal Raschka [33] na sade CIFAR-10¹. Obrázok 3.4 znázorňuje priebeh evaluačných experimentov. Vo výsledkoch uvádza, že na testovanej dátovej sade došlo pri použití AutoAugment-u ku zlepšeniu presnosti o 12 %, jeho nástupca RandAugment zlepšuje výkonnosť o ďalšie dva percentuálne body a TrivialAugment dosahuje presnosť o dodatočné percento vyššiu ako RandAugment.



Obr. 3.4: Porovnanie techník automatickej augmentácie podľa [33]

Okrem algoritmov, ktoré boli spomínané, existujú aj ďalšie, napr. **AugMix**², či **RandomAffine**³. Tieto prístupy boli tiež súčasťou uvedeného porovnania s AutoAugment, RandAugment a TrivialAugment, no dosiahli horšie výsledky.

¹CIFAR-10 je dátová sada obsahujúca 60 000 obrázkov rozdelených do desiatich tried, ktoré sa používajú na trénovanie a testovanie klasifikačných modelov počítačového videnia [34].

²AugMix kombinuje viacero obrázkov a augmentácií do jedného obrázka [35].

³Technika implementovaná ako súčasť balíka pytorch [36].

4 Zovšeobecnenie pomocou augmentácií zvukového signálu

Na zvýšenie zovšeobecniteľnosti modelu na predikciu Parkinsonovej choroby z rečových dát bol navrhnutý prístup na báze neurónovej siete. Po potvrdení hypotézy o slabom výkone na iných sadách bol navrhnutý súhrn metód, ktorý aplikuje augmentačné techniky na zvukové signály.

4.1 Dátové sady pre detekciu Parkinsonovej choroby z reči

V tejto sekcii sú opísané štyri dátové sady použité pri tréňovaní a neskoršom vyhodnocovaní modelu neurónovej siete. Pre zjednodušenie neskoršej orientácie medzi jednotlivými sadami sú zavedené aj skratky.

PC-GITA

Táto databáza (v ďalších častiach bude označená *SP*) obsahuje vzorky od 100 španielsky hovoriacich účastníkov, pričom 50 je zdravých (značené ako *HC*) a 50 sú pacienti s Parkinsonovou chorobou (označenie *PD*). Zvuk bol zaznamenaný do nahrávok so vzorkovacou frekvenciou 44,1 kHz a hĺbkou 16 bitov. Dostupná časť obsahuje záznam úlohy vyslovenia samohlások **a, e, i, o, u** po dobu niekoľkých sekúnd, pričom sú k dispozícii 3 vzorky od každého z účastníkov. Úplná databáza obsahuje aj ďalšie úlohy, pričom ich podrobnejší popis je možné nájsť v [37].

CzechPD

Táto sada (neskôr pod skratkou *CZ*) obsahuje vzorky 32 česky hovoriacich účastníkov - 16 zdravých a 16 pacientov s Parkinsonovou chorobou. Vzorkovacia frek-

vencia nahrávok je 48 kHz a bitová hĺbka 16 [38]. K dispozícii sú záznamy vyslovenia samohlások **a, i, u** trvajúce niekoľko sekúnd.

RMIT-PD

Sada RMIT-PD (ďalej v texte pod skratkou *EN*) obsahuje vzorky 41 anglicky hovoriacich účastníkov, z ktorých je 13 zdravých a 28 pacientov s Parkinsonovou chorobou. Nahrávky sú vzorkovacej frekvencie 48 kHz a hĺbky 16 bitov [39]. Databáza zahŕňa dlhé vyslovenie samohlások **a, o** a slabík **mmm, naa**.

ItalianPVS

ItalianPVS (ďalej len *IT*) zahŕňa nahrávky 50 taliansky hovoriacich účastníkov. Zdravú skupinu tvorí 22 a pacientov s Parkinsonovou chorobou je 28. Vzorky sú o frekvencii 16 kHz s 16 bitovou hĺbkou, pričom zvuk obsahuje výrazný šum prostredia v pozadí [40]. Na nahrávkach je zaznamenané vykonanie úlohy vyslovenia samohlások **a, e, i, o, u** po dobu niekoľkých sekúnd.

Vzájomné porovnanie dostupných dátových sád je uvedené v tabuľke 4.1.

Dátová sada	Úlohy	Počet HC	Počet PD
SP	a, e, i, o, u	50	50
CZ	a, i, u	16	16
EN	a, o, mmm, naa	13	28
IT	a, e, i, o, u	22	28

Tabuľka 4.1: Porovnanie obsiahnutých úloh a počtov účastníkov v použitých dátových sádach

4.2 Architektúra klasifikačného modelu

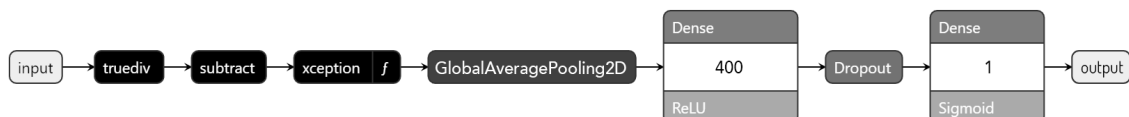
Vhodný návrh architektúry modelu je podstatným krokom pri riešení problému diagnostiky Parkinsonovej choroby z reči. Ako bolo už spomínané, zvukový signál je prevedený do obrázkovej formy a teda do úvahy prichádzajú konvolučné neurónové siete, ktoré sa štandardne používajú pri úlohách s obrázkami.

Pre zmenšenie množstva parametrov na tréningovanie siete, a tiež z dôvodov uvedených v časti o prenosovom učení (2.3), bol pri voľbe základného modelu braný do úvahy tento princíp. V rámci úsilia pri hľadaní vhodného modelu boli vykonané pokusy s použitím modelov na báze ResNet152V2 [41] a Xception [42].

Trénovanie neurónovej siete prebiehalo na sade SP, keďže obsahuje najväčší počet vzoriek, čo je dobrý predpoklad na kvalitnejšie natrénovanie siete, t.j. lepšie naučenie rozdielov medzi zdravými vzorkami a vzorkami pacientov. Použitá sada je tvorená 300 vzorkami (150 HC a 150 PD) obsahujúcimi úlohu dlhého vyslovenia samohlásky **a**. Tieto vzorky boli rozdelené do trénovacej a overovacej (*angl. validation*) sady v pomere 80:20. Výsledný model má nasledovnú architektúru:

1. **vrstva na predspracovanie vstupov,**
2. **model Xception,**
 - využíva váhy ImageNet,
 - vrchné (klasifikačné) vrstvy nie sú zahrnuté
3. **združovacia vrstva,**
4. **hlboko-prepojená vrstva,**
 - 400 neurónov,
 - aktivačná funkcia ReLU
5. **Dropout vrstva,**
 - hodnota parametra 0,2
6. **predikčná vrstva.**
 - 1 neurón,
 - aktivačná funkcia typu sigmoid

Použitým optimalizátorom je Adadelta s rýchlosťou učenia nastavenou na 0,01. Architektúra modelu je ilustrovaná na obrázku 4.1, schéma bola vytvorená s pomocou voľne dostupného nástroja Netron¹.



Obr. 4.1: Architektúra modelu použitého na experimenty v tejto práci

¹<https://netron.app>

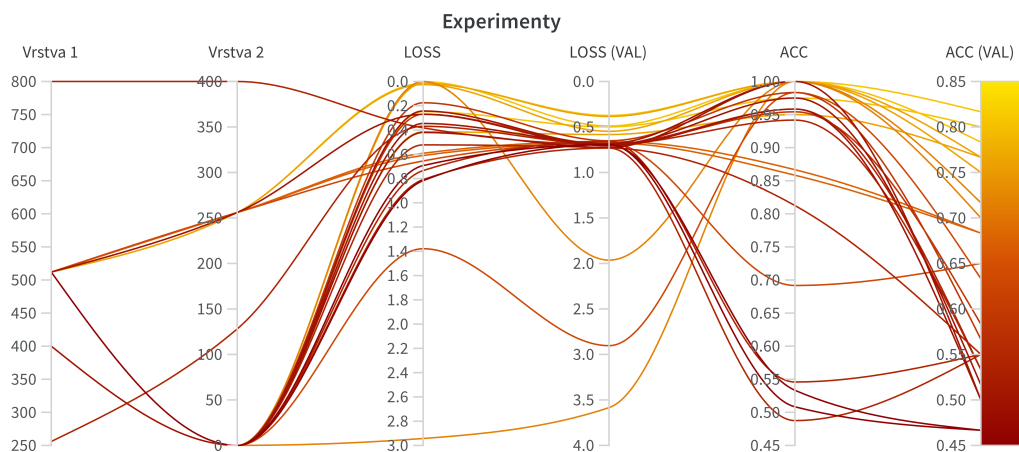
Táto konfigurácia je výsledkom experimentov, počas ktorých boli ladené parametre neurónovej siete, primárne rôzne počty hlboko prepojených vrstiev s rôznymi množstvami ich neurónov. Taktiež bola odskúšaná aplikácia Dropout vrstvy, viacerých možností združovacej (pooling) vrstvy a použitie iných optimalizátorov.

Spoločným menovateľom pre všetky experimenty bola konfigurácia pozostávajúca z týchto nastavení:

- **100 epoch** - maximálny počet tréningových iterácií,
- **veľkosť dávky 32**,
- **binárna krížová entropia** - použitá stratová funkcia.

Súčasne bola použitá aj technika Early Stopping, ktorá umožňuje skoršie ukončenie tréningovania, pokiaľ sa hodnota sledovanej metriky nezlepšuje pri určitom počte epoch, v tomto prípade bola nastavená trpezlivosť na 15 epoch. Pri hľadaní najlepšieho modelu boli porovnávané hodnoty metrík presnosti (accuracy), AUC a stratovej funkcie (LOSS), pričom sledované boli výsledky na tréningovej aj overovacej časti.

Výsledné porovnanie modelov je ilustrované na obrázku 4.2. Prvé dva stĺpce symbolizujú počet neurónov v prvej a druhej hlboko prepojenej vrstve a zvyšné uvádzajú dosiahnuté hodnoty metrík (označenie *VAL* znamená metriku pre overovaciu časť sady).



Obr. 4.2: Porovnanie modelov s rôznymi počtami vrstiev a neurónov

SP*			CZ			EN			IT		
LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
0,43	0,91	0,97	0,61	0,78	0,8	0,66	0,63	0,63	0,68	0,59	0,63

Tabuľka 4.2: Výsledky najlepšieho základného modelu. V tejto aj v ďalších tabuľkách je tréningová sada označená symbolom *.

Najlepšie výsledky z vykonaných experimentov dosiahol model s architektúrou na báze modelu Xception. Výkonnosť tohto modelu vrátane výkonnosti na zvyšných dátových sadách je zaznamenaná v tabuľke 4.2. Je evidentné, že model dosahuje najlepšie výsledky pri sade, na ktorej bol trénovaný, pričom však výkonnosť na iných sadách je výrazne zhoršená.

4.3 Grafická reprezentácia zvukových nahrávok

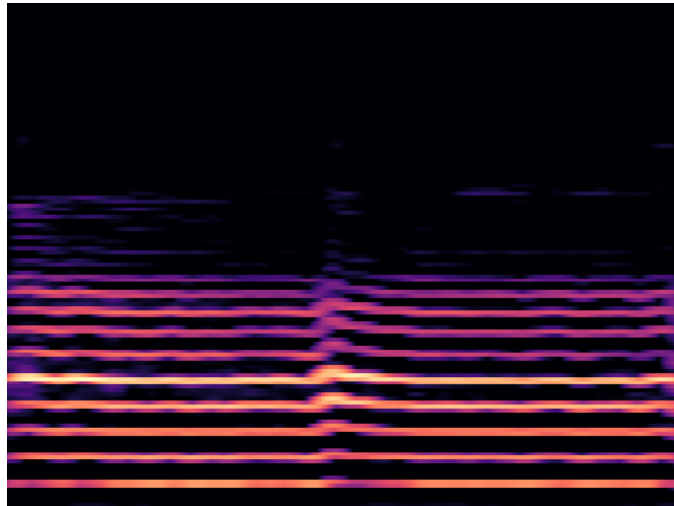
Keďže použitý model je konvolučnou neurónovou sieťou, zvukové vzorky sú prevedené do obrázkovej podoby. V rámci prípravy vzoriek boli všetky zvukové súbory upravené tak, že z nich boli odrezané prázdne miesta na začiatku a na konci, aby spektrogram obsahoval len relevantné údaje.

Pri otvorení súboru je vzorkovacia frekvencia znížená na 22050 Hz. Tento krok má za výsledok kratší čas spracovania vzoriek, pričom nedochádza k výraznej strate informácie vo vzorkách, keďže sa jedná o analýzu hlasu. Keby však vzorky obsahovali hudbu, tak by zníženie frekvencie mohlo viesť ku strate informácie a následne k zhoršeniu kvality analýzy. Počet Mel pásiem je nastavený na 128, pričom táto hodnota je kompromisom medzi prehľadnosťou vzorky (t.j. neurónová sieť je z nej schopná sa naučiť potrebné príznaky) a príliš vysokou hodnotou, ktorá by zvýšila čas predspracovania vzoriek aj čas trénovania siete. Výsledkom je spektrogram, ktorý je možné použiť pri trénovaní modelu, ako napr. na obrázku 4.3.

Experimentálne bol overený aj prevod zvukových vzoriek na MFCC spektrogramy, no v tom prípade model neurónovej siete dosahoval horšie výsledky, preto sa v ďalej uplatňuje konverzia iba na Mel spektrogramy.

4.4 Techniky augmentácie zvukových dát

V kontexte diagnostiky Parkinsonovej choroby z rečových dát je dôležité vytvoriť spoľahlivý a robustný predikčný model. Hlavná výzva spočíva v tom, že do-



Obr. 4.3: Príklad vzorky vo forme spektrogramu

stupné trérovacie dáta sú v rôznych jazykoch a ich reprezentácie sa vzájomne líšia v prízvuku, tempe a ďalších prozodických vlastnostiach na základe regionálnych rozdielov. Preto sú používané augmentačné techniky na rozšírenie trérovacích dát a zvýšenie ich variability.

Zvolený prístup využíva aplikovanie zvukových augmentácií na vzorky ešte pred prevedením do obrázkovej podoby ako spektrogram. Medzi dôvody patrí zvýšená implementačná flexibilita a taktiež predpoklad, že takto natrénovaný model má lepšie využitie v reálnom prostredí, kde vzorky môžu obsahovať šum, či zvuky v pozadí. Rovnako sa predpokladá, že zvukové augmentácie majú priaznivejší dopad na zvýšenie robustnosti modelu s ohľadom na rôzne jazyky.

Pri výbere jednotlivých augmentácií bola jedným z rozhodujúcich faktorov možnosť priamej manipulácie so signálom prečítaným z formátu *.wav*. Takúto možnosť ponúka aj balík *librosa*, ktorý už je použitý na konverziu signálu na spektrogram. Použitím tohto balíka sa jednak dosiahne nižšia závislosť od ďalších knižníc a tiež dôjde k nižšiemu času prípravy dátovej sady pre použitie neurónovou sieťou. Prípravou sa v tomto prípade rozumie aplikácia augmentácie a prevod na spektrogram. Síce boli vykonané aj experimenty s ďalšími augmentáciami mimo balíka *librosa*, no tie mali negatívny dopad najmä na spomínaný čas spracovania, čo pri väčšom počte vzoriek je výrazné negatívum. Súčasťou implementácie sú však aj techniky, ktoré pracujú so signálom ako poľom číselných hodnôt.

Je dôležité zdôrazniť, že veľký vplyv na vzhľad výsledného spektrogramu (a teda aj na schopnosť modelu naučiť sa z neho) má voľba magnitúdy/sily danej augmentácie. Pokiaľ sa pri nastavovaní augmentačnej techniky na to neberie ohľad, môže dôjsť k strate informácie zo signálu do takej miery, že model nebude schopný učiť sa z poskytnutých skreslených spektrogramov.

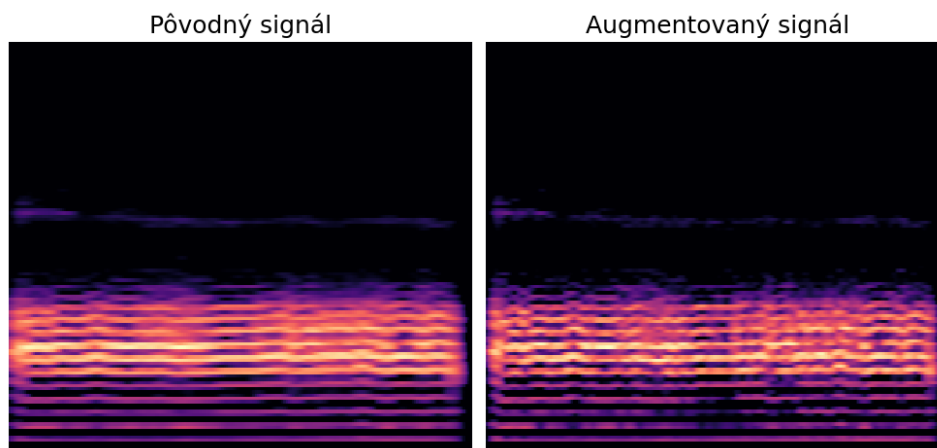
4.4.1 Zmena rýchlosti

Táto technika zahŕňa zmenu rýchlosti zvukového signálu za zachovania pôvodnej frekvencie vzorky, čo ma za výsledok zvýšenie alebo zníženie tempa reči.

Augmentácia je nastaviteľná číselným parametrom, ktorý rozhoduje o miere zmeny rýchlosti, pričom parameter môže nadobúdať hodnoty z intervalov:

- $(0, 1)$ - zrýchlenie signálu, skrátenie vzorky,
- $(1, \infty)$ - spomalenie signálu, predĺženie vzorky.

Na obrázku 4.4 je ilustrovaný vplyv augmentácie na zvukový signál, konkrétne ide o príklad zrýchlenia vzorky.



Obr. 4.4: Augmentácia zvukového signálu zrýchlením

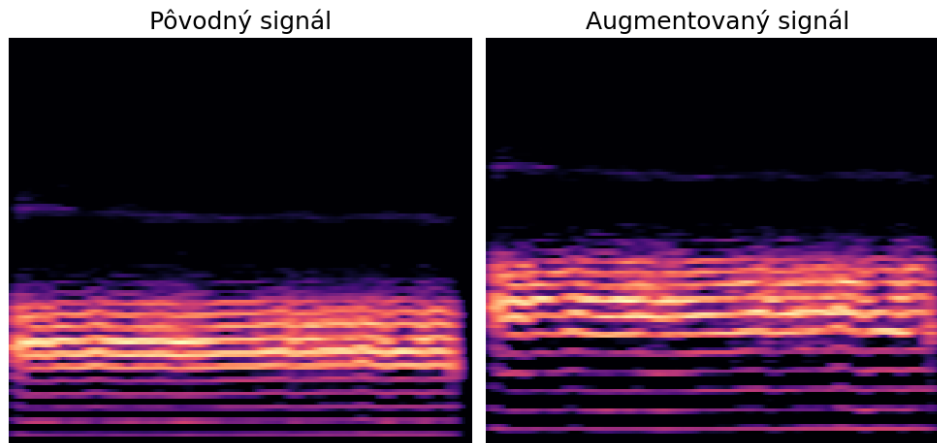
4.4.2 Zmena výšky tónu

V porovnaní s predchádzajúcou technikou, táto mení výšku tónu (resp. frekvenciu) vzorky, pričom nie je ovplyvnená dĺžka signálu. Takto upravené dáta môžu modelu umožniť naučiť sa rôzne hlasové tóny a modulácie.

Funkcionalitu posunu tejto augmentácie je možné špecifikovať nastavením parametra, ktorý môže nadobúdať hodnoty:

- **záporné** - zníženie tónu o daný počet poltónov,
- **kladné** - zvýšenie tónu o daný počet poltónov.

Jeden z týchto prípadov - zvýšenie tónu je znázornené na obrázku 4.5.



Obr. 4.5: Augmentácia zvukového signálu zvýšením tónu

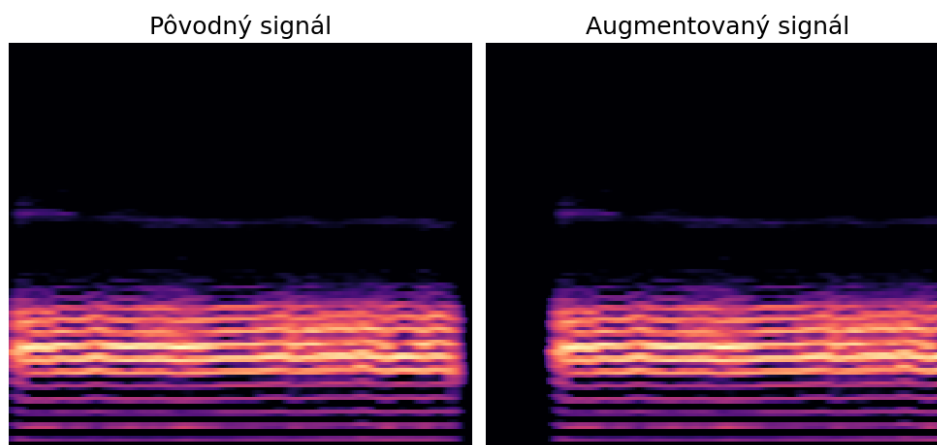
4.4.3 Časový posun

Časový posun znamená posunutie zvukového signálu doprava alebo doľava. Takto sa dosiahne zmena pozície hlasu v čase.

Konfigurácia techniky prebieha parametrom symbolizujúcim čas (v sekundách), o ktorý je signál posunutý. V tomto prípade pojem vzorka (*angl. sample, frame*) súvisí so vzorkovacou frekvenciou zvukového súboru, nie s dátovou jednotkou pre učenie siete. Je možné dosiahnuť posun:

- **doprava** - hodnota parametra je kladné číslo,
- **doľava** - záporná hodnota parametra.

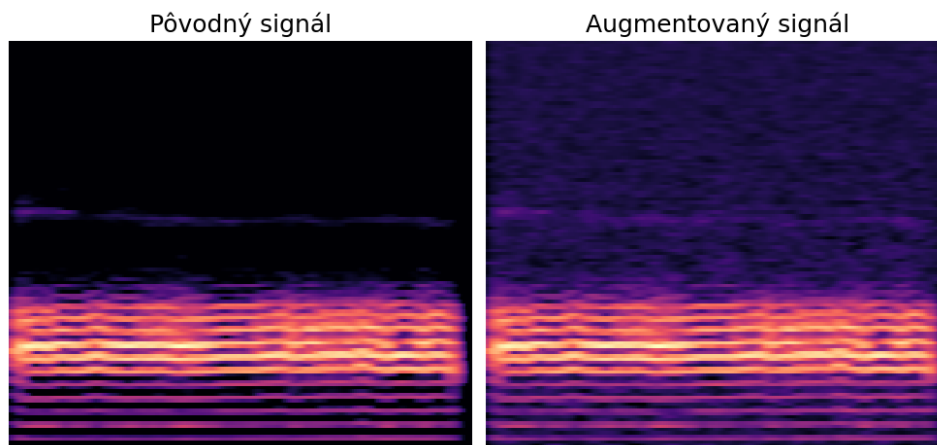
V oboch prípadoch sa na výsledný spektrogram dostáva prázdne miesto, ako ilustruje obrázok 4.6.



Obr. 4.6: Augmentácia zvukového signálu časovým posunom

4.4.4 Pridanie šumu

V rámci tejto augmentácie je do vzorky umelo pridaný šum, čo zvyšuje variabilitu modelu pri použití vzoriek, ktoré nie sú dokonalo čisté (napr. nevznikli v štúdiových podmienkach). Mieru pridania šumu determinuje kladné číslo, ktoré znamená faktor určujúci množstvo šumu. Vplyv šumu je možné pozorovať na obrázku 4.7.



Obr. 4.7: Augmentácia zvukového signálu pridaním šumu

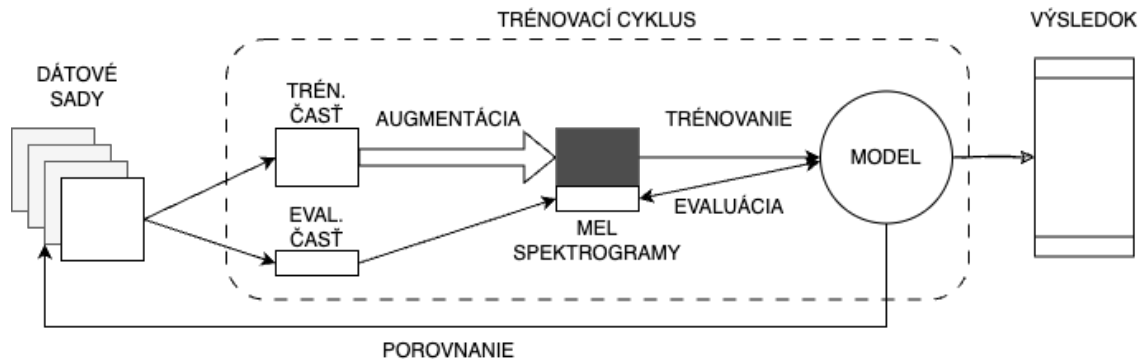
4.5 Automatická dátová augmentácia

Keďže hlavnou úlohou tejto práce je navrhnúť prístup automatickej augmentácie pre predikciu Parkinsonovej choroby z hlasu, bude práve to obsahom tejto časti. Pred samotným návrhom boli preskúmané existujúce možnosti automatickej augmentácie (3.3). Priama aplikácia niektorého z týchto algoritmov však nie je možná, keďže fungujú na báze obrázkových augmentácií a navrhovaný prístup v rámci tejto práce sa zameriava na zvukové augmentácie pôvodného signálu. Napriek tomu bol najmä AutoAugment výraznou inšpiráciou v princípe vyhodnocovania efektivity použitých augmentačných techník.

Navrhovaný prístup ilustrovaný na obrázku 4.8 pozostáva z nasledujúcich krokov.

1. Predspracovanie zvukových dát

Prvým krokom je príprava zvukových dát. Na začiatku sa vytvoria potrebné adresáre pre tréningovú a evaluačnú sadu s pomenovaním v závislosti od názvu



Obr. 4.8: Diagram navrhovaných metód zovšeobecnenia aplikovaním augmentácie na zvukový signál

augmentácie a parametra. Následne sa vstupné dáta rozdelia na tréningovú a evaluačnú sadu v pomere 90:10.

Na vzorky označené ako tréningové je najprv aplikovaná zvolená augmentácia, potom sú prevedené do podoby Mel spektrogramu a nakoniec sú uložené do cieľového adresára vo formáte *.png*.

Vzorky z evaluačnej časti prechádzajú podobným procesom s tou výnimkou, že na ne augmentácia nie je aplikovaná. Rozdelenie dátovej sady na tréningovú a evaluačnú časť je dôležité pre overenie a validáciu výkonu modelu. Evaluačná časť slúži na posúdenie výkonnosti modelu na doposiaľ nevidených dátach. Tento krok je kritický pre zabezpečenie toho, že model sa dobre zovšeobecnil na nové údaje a nie je pretrénovaný na tréningových dátach.

Aplikácia augmentácií len na tréningové dáta zabezpečuje, že evaluačná časť sady obsahuje len pôvodné dáta bez akejkoľvek manipulácie. To umožňuje presné posúdenie výkonnosti modelu na netréningovaných dátach, ktoré reprezentujú reálne podmienky. Ak by boli augmentácie aplikované aj na evaluačnú časť, model by bol tréningovaný na augmentovaných a netréningovaných dátach, čo by mohlo viesť k pretrénovaniu a nesprávnym odhadom výkonnosti na nových dátach.

V prípade zvolenia možnosti aplikácie viacerých augmentácií na vzorky je namiesto spomínaného procesu využitá postupná aplikácia všetkých zvolených augmentácií na každú zo vzoriek v tréningovej časti. Obrázkové dáta sú dodatočne prevedené do spracovateľnej formy pre neurónovú sieť.

2. Vytvorenie modelu

Následne prebieha vytvorenie modelu pre každú z použitých augmentácií. Každý cyklus je vykonávaný na novej inštancii modelu. Architektúra modelu je založená na predtrénovanom modeli Xception, ktorý je upravený tak, aby bol vhodný

pre konkrétnu úlohu. Tento krok zvyšuje rýchlosť a efektívnosť tréovania a umožňuje modelu efektívne sa prispôbiť na nové podmienky a domény.

Konkrétna architektúra modelu je výsledkom experimentov detailne opísaných v sekcii 4.2. Konfigurácia parametrov je zadefinovaná v globálnom slovníku, ktorý je inicializovaný na začiatku kódu. Dôvodom na oddelenie týchto hodnôt je ich potenciálne viacnásobné využitie, ako napr. v prípade, keď sa jednotlivé tréovacie cykly sledujú vo vizualizačnom nástroji, tak tieto hodnoty sú nápomocné pri ich odlišení.

3. Evaluácia výkonnosti

Posledným krokom je vyhodnotenie výkonnosti a teda aj prínos danej augmentácie. Po inicializácii prázdneho slovníka s výsledkami pre jednotlivé dátové sady sa postupne vyhodnocujú výsledky pre každú augmentáciu. Po získaní metrík sa vykoná porovnanie s výsledkami predchádzajúcich augmentácií. Na základe vopred definovaného kritéria je vybraná najlepšia augmentácia. Na výber sú kritériá stratová funkcia, presnosť (accuracy) a AUC.

Pri porovnávaní výsledkov je potrebné zohľadniť odlišnú charakteristiku jednotlivých kritérií. Vyššia hodnota presnosti a AUC je považovaná za priaznivejšiu, zatiaľ čo nižšia hodnota stratovej funkcie je indikátorom lepšej výkonnosti modelu. Preto je logika porovnania výsledkov prispôbená tomuto faktoru.

V inicializačnom kroku slovníka s najlepšimi výsledkami je dôležité nastaviť počiatočné hodnoty tak, aby akákoľvek hodnota výsledku bola považovaná za priaznivejšiu pri prvom porovnaní. To zabezpečuje, že prvý získaný výsledok bude automaticky považovaný za najlepší a následné porovnania budú vyhodnocovať, či sú nové výsledky lepšie alebo horšie ako tento prvý referenčný výsledok.

Po vykonaní porovnania a výbere najlepšej augmentácie na základe stanoveného kritéria sa získané výsledky ukladajú do slovníka s najlepšimi výsledkami. Tento proces sa opakuje pre každú dátovú sadu, aby sa zabezpečilo, že na konci bude zvolená augmentácia, ktorá dosahuje najlepšie výsledky zo všetkých prípadov.

Zhrnutie postupu

Celý proces prebieha v 5 krokoch, ktoré sa opakujú pre každú zo zadaných augmentácií, pričom je vykonaný nasledovný postup:

1. Príprava trénovacích a evaluačných dát

V tomto kroku prebehne vytvorenie obrázkovej dátovej sady s príslušnou augmentáciou a prevod do podoby vyžadovanej neurónovou sieťou.

2. Vytvorenie predikčného modelu

Pre každú augmentáciu je vytvorený samostatný model.

3. Trénovanie modelu

Počas tejto fázy prebieha samotné učenie modelu na trénovacích dátach a následné vyhodnotenie na evaluačnej časti.

4. Vyhodnotenie výkonnosti

Poslednými krokmi sú vyhodnotenie naučeného modelu na ostatných dátových sádach, porovnanie cieľových metrík a zaznamenanie najlepších dosiahnutých výsledkov.

Po poslednej augmentácii program poskytne ako výsledok údaje o najefektívnejších augmentáciách pre každú z dátových sád vo forme slovníka. Pokiaľ však používateľ zvolí možnosť aplikácie viacerých augmentácií naraz, tieto výsledky nie sú dostupné.

4.6 Implementácia riešenia

Pri vývoji programu bolo použité prostredie PyCharm². Návrh je realizovaný v programovacom jazyku Python 3.10.5 [43], pre ktorý je dostupná široká ponuka pomocných knižníc pre prácu s neurónovými sieťami, ako aj pre spracovanie zvukových signálov. Na implementáciu modelov neurónovej siete bola použitá knižnica TensorFlow [44] (verzia 2.9.0) pre lepšiu možnosť vizualizácie procesu učenia aj pre existujúce skúsenosti s jej ovládaním. Učenie neurónovej siete prebiehalo na počítači s grafickou kartou NVIDIA GeForce RTX 4070 s grafickou pamäťou 12 GB. Na porovnanie výkonnosti jednotlivých konfigurácií modelov

²<https://www.jetbrains.com/pycharm/>

bol použitý nástroj Weights & Biases [45], ktorý umožňuje vizualizáciu sledovaných metrík v reálnom čase aj vzájomné grafické porovnanie. Na prácu so zvukovými súbormi (predspracovanie a aplikácia augmentačných techník), ako aj na konverziu vzorky na spektrogram bola použitá knižnica librosa [46], konkrétne verzia 0.10.1. Implementácia riešenia je realizovaná s využitím prostredia Jupyter Notebook [47], ktoré umožňuje reprodukovateľné a jednoducho nastaviteľné vývojové podmienky.

Spustenie programu

Pred spustením sa od používateľa vyžaduje vynaloženie úsilia jedine na definíciu priestoru prehľadávaných augmentácií. Notácia pre augmentácie sa skladá z dvoch častí:

1. Volanie funkcie z triedy `Augmentation`,
2. Špecifikácia funkcie v podobe slovníka.

Popis v druhej časti slúži na špecifikovanie parametra augmentačnej funkcie a tiež na pomenovanie (pre prehľadnosť v logovacích súboroch). Duplicita hodnôt pre danú funkciu je odôvodnená jednoduchším implementačným spracovaním, keďže parameter použitý pri výpisoch má jeden názov (v parametri *value*) a každá z augmentácií očakáva iný názov konfiguračnej premennej (ďalší parameter, napr. *n_steps*). Zároveň je možné aj definovať augmentáciu ako *None*, čiže žiadna augmentácia nebude pre daný cyklus aplikovaná. Výsledky tohto cyklu môžu byť použité ako referenčná hodnota pre porovnanie efektivity ďalších augmentácií.

Následne stačí zvoliť porovnávacie kritérium a po ukončení sú údaje o najviac efektívnych augmentáciách pre individuálne dátové sady uložené do premennej. Čas trvania celého procesu je determinovaný z najväčšej časti časom učenia modelu, ktorý závisí do výraznej miery od dostupného výkonu počítača (najmä výkonnosť grafickej karty).

Po spustení programu sa minimalizuje potreba interakcie zo strany používateľa, keďže program nepretržite poskytuje informácie o svojom stave a postupe vykonávania. Tieto informácie sú systematicky zaznamenávané a následne poskytované prostredníctvom **výpisov do konzoly**, kde sú zreteľne zobrazené aktuálne vykonávané operácie. Takýto prístup umožňuje používateľovi sledovať postup práce programu v reálnom čase a mať tak aktuálny prehľad o tom, čo sa deje.

Po úplnom dokončení vykonávania programu je používateľovi k dispozícii **logovací súbor**, ktorý obsahuje podrobnejšie záznamy o vykonaných operáciách spolu s príslušnými časovými značkami. Tento logovací súbor zabezpečuje možnosť následného prehľadávania a analyzovania informácií o priebehu vykonávania programu. Okrem toho je názov tohto súboru špecificky navrhnutý tak, aby obsahoval aj časovú značku spustenia programu. Tento prístup zvyšuje prehľadnosť a organizovanosť v prípade, že program bol spustený viackrát a umožňuje jednoduché identifikovanie a archiváciu relevantných logovacích informácií.

5 Výsledky experimentov

Overenie navrhnutého prístupu bolo vykonané pomocou troch testovacích scenárov, pričom každý z nich je ekvivalentný s potenciálnym prípadom použitia implementovaného programu. Scenáre sa odlišujú použitými časťami dátových sád (t.j. pridaním iných úloh do spoločnej dátovej sady) a niektoré z nich obsahujú aj experimenty (experimentom sa rozumie natrénovanie modelu na dátach s/bez augmentácie a vyhodnotenie na všetkých dostupných dátových sadách), pri ktorých boli zakaždým nastavené iné dátové sady ako tréningové a porovnanie výkonnosti prebehlo na všetkých štyroch sadách. Pre každý z nich bol najprv vykonaný cyklus bez použitia akejkoľvek augmentačnej techniky, ktorého výsledky slúžia ako referenčné hodnoty pre ďalšie porovnania. Zvyšok cyklov je už s využitím niektorej augmentácie, pričom pred samotným spustením programu boli zvolené všetky dostupné augmentačné techniky (z časti 4.4). Všetky augmentácie boli zvolené viacnásobne s rozdielom v hodnote parametra, aby došlo ku korektnému overeniu prínosu rôznych nastavení - napr. pri zmene rýchlosti má zmysel overovať zvýšenie aj zníženie. Súčasne boli použité aj rôzne hodnoty parametrov pre rovnaké augmentácie medzi jednotlivými scenármi, čím vznikla širšia škála pre porovnanie efektivity daných augmentačných techník. Po ukončení experimentov program vráti najúspešnejšie augmentácie pre jednotlivé dátové sady, spolu s uvedením hodnoty nastavenej sledovanej metriky - v tomto prípade **LOSS** (hodnota stratovej funkcie). Pre kvantifikáciu prínosu sú vypočítané percentuálne zlepšenia dosiahnutej hodnoty metriky najlepších augmentačných techník podľa uvedeného vzorca.

$$\text{zlepšenie}(\%) = \frac{\text{pôvodná hodnota} - \text{nová hodnota}}{\text{pôvodná hodnota}} \times 100$$

Dodatočne je vypočítané aj priemerné zlepšenie pre všetky dátové sady. V nasledujúcich sekciách sú uvedené detailné informácie o individuálnych scenároch, výsledky relevantných experimentov a následná interpretácia.

5.1 Analýza prístupu na jednej samohláske

V rámci tohto scenára boli do použitých dátových sád zahrnuté zvukové vzorky obsahujúce úlohu vyslovenia samohlásky *a* s trvaním niekoľkých sekúnd. Tento výber je odôvodnený tým, že daná samohláska je obsiahnutá vo všetkých dátových sádach (SP, EN, CZ, IT) a zároveň počet týchto vzoriek je najväčší. V tomto prípade je overovaná výkonnosť modelu na čo najpodobnejších vzorkách. Množstvo vzoriek klasifikačných tried jednotlivých sád je uvedené v tabuľke:

Dátová sada	Počet vzoriek HC	Počet vzoriek PD
SP	150	150
CZ	16	16
EN	13	28
IT	44	56

5.1.1 Trénovanie na sade SP

Prvá sada experimentov využíva na trénovanie dátovú sadu PC-GITA (SP), obsahuje augmentácie uvedené v nasledujúcej tabuľke. Stĺpec *Použité parametre* uvádza postupne jednotlivé nastavenia parametra pre danú augmentáciu, čiže štyri hodnoty oddelené čiarkou znamenajú štyri samostatné experimenty s rovnakou augmentačnou technikou a odlišným nastavením.

Typ augmentácie	Použité parametre
Zmena rýchlosti	1.1, 1.5, 2, 0.7
Časový posun	2, 8, 16, 100
Pridanie šumu	1, 2, 5
Zmena výšky tónu	4
MULTI	Zmena rýchlosti 0.7 + časový posun 4 + pridanie šumu 5

Posledný experiment zahŕňa aplikáciu viacerých augmentačných techník naraz, jednotlivé parametre boli zvolené na základe výsledkov predošlých experimentov, s ohľadom na najlepší výkon na viacerých sádach.

EXPERIMENT	SP*			CZ			EN			IT		
~	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
Bez augmentácie	0,368	0,844	0,916	0,549	0,75	0,805	0,668	0,625	0,486	0,953	0,406	0,238
Zmena rýchlosti 1.1	0,39	0,875	0,914	0,648	0,656	0,775	0,711	0,594	0,461	0,833	0,594	0,377
Zmena rýchlosti 1.5	0,476	0,875	0,835	0,607	0,656	0,815	0,679	0,719	0,479	1,089	0,469	0,369
Zmena rýchlosti 2	0,522	0,812	0,841	0,715	0,594	0,779	0,922	0,531	0,219	0,818	0,531	0,302
Zmena rýchlosti 0.7	0,476	0,75	0,873	0,508	0,812	0,844	0,804	0,438	0,287	0,861	0,406	0,408
Časový posun 2	0,093	0,938	1	0,505	0,719	0,84	0,702	0,719	0,486	0,982	0,406	0,298
Časový posun 8	0,085	1	1	0,529	0,656	0,854	0,798	0,688	0,457	1,225	0,438	0,341
Časový posun 16	0,259	0,969	0,996	0,488	0,75	0,867	0,754	0,594	0,318	0,942	0,344	0,222
Časový posun 100	0,181	0,906	0,977	0,591	0,656	0,762	0,761	0,594	0,426	1,125	0,344	0,277
Pridanie šumu 1	0,669	0,625	0,671	0,662	0,625	0,736	0,687	0,625	0,573	0,651	0,688	0,701
Pridanie šumu 2	0,693	0,562	0,478	0,707	0,5	0,602	0,844	0,312	0,405	0,725	0,469	0,643
Pridanie šumu 5	0,699	0,531	0,514	0,713	0,562	0,41	0,668	0,719	0,621	0,652	0,656	0,7
Zmena výšky tónu 4	0,585	0,625	0,744	0,717	0,625	0,682	0,792	0,625	0,355	0,912	0,531	0,352
MULTI	0,691	0,562	0,494	0,714	0,562	0,379	0,708	0,5	0,552	0,709	0,469	0,424

Tabuľka 5.1: Výsledky tréovania na SP (scenár 1)

Tabuľka 5.1 uvádza zaznamenané výsledky jednotlivých experimentov. Kvôli prehľadnosti sú najpriaznivejšie hodnoty pre všetky metriky (nielen LOSS) zvýraznené tučným písmom. Po ukončení boli programom vyhodnotené ako najlepšie (na základe metriky **LOSS**) tieto z nich:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	časový posun (8)	0,085	76,9
CZ	časový posun (16)	0,488	11,1
EN	pridanie šumu (5)	0,668	0
IT	pridanie šumu (1)	0,651	31,7

Priemerné zlepšenie je 29,9 %. Na základe týchto výsledkov je možné konštatovať, že v tomto prípade sa nepodarilo nájsť augmentáciu s konkrétnym parametrom, ktorá by bola univerzálne použiteľná na všetky dátové sady a mala by pozitívny prínos, no časový posun a pridanie šumu (i keď s rôznymi parametrami) majú spoločne po dve dátové sady, ktorým pomohli. Je tiež evidentné, že pokus spojiť viaceré augmentácie a aplikovať ich naraz neprinesol výrazné zlepšenie.

5.1.2 Tréovanie na sadách SP a EN

V druhej sade experimentov boli spojené vzorky zo sád PC-GITA (SP) a RMIT-PD (EN) a následne boli použité na tréovanie. Táto spojená sada obsahovala 163 HC a 178 PD vzoriek, výsledky boli opäť porovnávané s jednotlivými pôvodnými

sadami. Obsiahnuté augmentácie sú:

Typ augmentácie	Použité parametre
Zmena rýchlosti	0.75, 0.65
Časový posun	3, 4, 10, 15
Pridanie šumu	2, 5
Zmena výšky tónu	3, 5
MULTI	Zmena rýchlosti 4 + zmena výšky tónu 5

Znovu viacnásobná augmentácia na záver bola zvolená na základe predpokladu najpriaznivejších spoločných augmentácií z predchádzajúcich výsledkov získaných po vykonaní prvej sady iterácií.

EXPERIMENT	SP*			CZ			EN*			IT		
~	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
Bez augmentácie	0,124	0,938	0,994	0,666	0,656	0,709	0,108	0,938	0,991	1,434	0,344	0,219
Zmena rýchlosti 0.75	0,402	0,844	0,917	0,685	0,594	0,674	0,39	0,844	0,909	1,097	0,469	0,285
Zmena rýchlosti 0.65	0,628	0,688	0,763	0,674	0,562	0,602	0,572	0,719	0,667	0,893	0,344	0,337
Časový posun 3	0,072	1	1	0,686	0,625	0,699	0,111	0,969	0,994	1,017	0,469	0,287
Časový posun 4	0,177	0,969	0,986	0,601	0,719	0,773	0,105	0,938	1	1,071	0,469	0,392
Časový posun 10	0,201	0,906	0,98	0,524	0,719	0,815	0,056	1	1	0,944	0,438	0,439
Časový posun 15	0,09	0,969	1	0,596	0,688	0,766	0,093	0,938	1	1,025	0,375	0,292
Pridanie šumu 2	0,697	0,531	0,64	0,687	0,5	0,543	0,658	0,688	0,591	0,683	0,531	0,565
Pridanie šumu 5	0,687	0,531	0,582	0,695	0,5	0,625	0,702	0,406	0,645	0,733	0,281	0,341
Zmena výšky tónu 3	0,448	0,844	0,901	0,575	0,688	0,777	0,478	0,75	0,881	1,361	0,281	0,268
Zmena výšky tónu 5	0,498	0,812	0,851	0,591	0,688	0,752	0,455	0,781	0,841	1,059	0,375	0,235
MULTI	0,605	0,625	0,782	0,603	0,594	0,697	0,437	0,75	0,87	1,09	0,406	0,325

Tabuľka 5.2: Výsledky tréovania na SP+EN (scenár 1)

Z experimentov, ktorých výsledky sú uvedené v tabuľke 5.2, program vybral ako najlepšie tieto:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	časový posun (3)	0,072	41,9
CZ	časový posun (10)	0,524	21,3
EN	časový posun (10)	0,056	48,2
IT	pridanie šumu (2)	0,683	52,4

Priemerné zlepšenie nadobúda hodnotu 41 %. V tomto prípade je evidentný najväčší prínos jednej augmentácie - časového posunu, aj s konkrétnym parametrom (10), ktorý mal najpriaznivejší efekt na dve z overovaných sád - CZ a EN.

Opätovne bol pozorovaný fakt, že súčasná aplikácia viacerých augmentácií nemala výrazný prínos, preto v nasledujúcich scenároch tento prístup už nebol použitý.

Uvedené časti sú dostatočné na získanie relevantných poznatkov o vplyve navrhovaného prístupu na danú úlohu. V prípade zahrnutia viacerých kombinácií tréningovej sady by došlo ku zvýšeniu zložitosti analýzy a interpretácie výsledkov.

5.2 Analýza prístupu na spoločných úlohách

V ďalšom scenári je experimentálne overovaný prístup na dátových sadách tvorených úlohami, ktoré sú spoločné pre všetky dostupné dátové sady. Výsledné dátové sady sú rozšírené oproti prvému scenáru ďalšou úlohou a tvoria ich vyslovenie samohlások **a** a **u**. Použité sady tak obsahovali nasledovné počty vzoriek:

Dátová sada	Počet vzoriek HC	Počet vzoriek PD
SP	300	300
CZ	32	32
EN	26	56
IT	44	56

V tomto scenári už je variabilita vzoriek vyššia oproti predchádzajúcemu scenáru, a preto boli vykonané štyri sady experimentov, pričom pri každej z nich bola využitá iná z dostupných dátových sád na tréning modelu neurónovej siete.

5.2.1 Tréning na sade SP

Prvá zo sád experimentov využíva ako zdrojovú dátovú sadu na tréning sadu PC-GITA (SP) a overuje augmentácie:

Typ augmentácie	Použité parametre
Zmena rýchlosti	0.6, 1.5, 2.5
Časový posun	4, 8, 12
Pridanie šumu	3, 6, 1
Zmena výšky tónu	2, 6, 7

Výsledky po aplikácii jednotlivých augmentačných techník sú zaznamenané

v tabuľke 5.3. Výsledný výpis najlepších augmentácií pre jednotlivé sady obsahuje:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	bez augmentácie	0,062	0
CZ	bez augmentácie	0,621	0
EN	pridanie šumu (1)	0,647	50,1
IT	pridanie šumu (1)	0,694	38,5

EXPERIMENT	SP*			CZ			EN			IT		
	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
~												
Bez augmentácie	0,062	1	1	0,621	0,719	0,81	1,297	0,5	0,162	1,129	0,438	0,416
Zmena rýchlosti 0.6	0,717	0,656	0,782	0,669	0,594	0,669	0,804	0,5	0,439	1,374	0,281	0,161
Zmena rýchlosti 1.5	0,385	0,812	0,898	0,828	0,656	0,627	1,063	0,531	0,352	1,001	0,562	0,51
Časový posun 2.5	0,535	0,719	0,816	0,86	0,594	0,676	0,969	0,594	0,356	0,955	0,594	0,306
Časový posun 4	0,175	0,938	0,984	0,877	0,594	0,615	0,995	0,688	0,18	1,135	0,438	0,412
Časový posun 8	0,125	0,938	0,996	0,833	0,531	0,567	1,183	0,438	0,239	1,334	0,312	0,301
Časový posun 12	0,287	0,875	0,981	0,633	0,656	0,8	1,365	0,531	0,177	1,31	0,312	0,32
Pridanie šumu 3	1,146	0,625	0,387	1,216	0,5	0,639	2,24	0,156	0,449	1,463	0,5	0,578
Pridanie šumu 6	1,09	0,469	0,478	0,985	0,5	0,416	0,672	0,688	0,45	0,827	0,594	0,48
Pridanie šumu 1	0,649	0,562	0,798	0,71	0,531	0,557	0,647	0,656	0,482	0,694	0,562	0,643
Zmena výšky tónu 2	0,383	0,844	0,892	0,785	0,781	0,697	0,727	0,688	0,366	1,25	0,312	0,359
Zmena výšky tónu 6	0,423	0,812	0,875	0,659	0,656	0,68	0,699	0,656	0,561	1,169	0,406	0,261
Zmena výšky tónu 7	0,542	0,75	0,837	0,679	0,688	0,696	0,805	0,531	0,497	1,056	0,406	0,372

Tabuľka 5.3: Výsledky tréningu na SP (scenár 2)

Tentokrát najlepší výsledok bol dosiahnutý s pridaním šumu (1) - pre sady EN a IT. Na zvyšných dvoch sadoch boli dosiahnuté najlepšie výsledky bez akejkoľvek aplikovanej augmentácie, keďže na sade SP došlo k pretrénovaniu - presnosť dosiahla maximálnu hodnotu 1. V poradí druhou najlepšou technikou je pre ňu časový posun (8). K takémuto záveru je možné dospieť aj bez dodatočne pripravenej tabuľky a to vďaka súhrnnému výpisu, ktorý je dostupný po ukončení vykonania programu a umožňuje analyzovať dosiahnuté hodnoty metrík. Priemerné zlepšenie bolo v tomto prípade 22,1 %.

5.2.2 Tréovanie na sade CZ

Ďalšia časť experimentov využíva na tréovanie sadu CzechPD (CZ), pričom medzi prehľadávané augmentačné techniky patria:

Typ augmentácie	Použité parametre
Zmena rýchlosti	0.65, 2.5
Časový posun	3, 6
Pridanie šumu	8, 1
Zmena výšky tónu	5, 9

EXPERIMENT	SP			CZ*			EN			IT		
	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
~												
Bez augmentácie	0,725	0,406	0,433	0,722	0,438	0,492	0,723	0,406	0,648	0,77	0,406	0,33
Zmena rýchlosti 0.65	0,706	0,469	0,577	0,724	0,406	0,32	0,673	0,625	0,575	0,641	0,719	0,837
Zmena rýchlosti 2.5	0,656	0,656	0,686	0,658	0,688	0,792	0,672	0,688	0,692	0,709	0,438	0,422
Časový posun 3	0,731	0,406	0,575	0,724	0,406	0,532	0,725	0,344	0,487	0,735	0,375	0,456
Časový posun 6	0,667	0,656	0,515	0,716	0,5	0,508	0,635	0,719	0,705	0,689	0,562	0,472
Pridanie šumu 8	0,743	0,344	0,452	0,707	0,406	0,68	0,712	0,438	0,628	0,693	0,562	0,417
Pridanie šumu 1	0,664	0,781	0,765	0,695	0,469	0,522	0,653	0,75	0,775	0,698	0,5	0,439
Zmena výšky tónu 5	0,747	0,375	0,325	0,727	0,406	0,385	0,718	0,375	0,659	0,742	0,375	0,321
Zmena výšky tónu 9	0,679	0,5	0,694	0,7	0,5	0,447	0,692	0,469	0,659	0,675	0,656	0,7

Tabuľka 5.4: Výsledky tréovania na CZ (scenár 2)

Spomedzi experimentov v tabuľke 5.4 boli najlepšie:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	zmena rýchlosti (2.5)	0,657	13,5
CZ	zmena rýchlosti (2.5)	0,658	8,9
EN	časový posun (6)	0,635	12,2
IT	zmena rýchlosti (0.65)	0,641	16,8

V tomto prípade bolo priemerné zlepšenie 12,8 %, pričom najlepšie výsledky mala zmena rýchlosti (2.5) pre dve overované sady - SP a CZ. Je zaujímavé, že rovnaká augmentácia, no s parametrom 0.65, má priaznivý vplyv na sadu IT. Pokiaľ by sa na túto sadu použil parameter 2.5, príde k rapidnému poklesu výkonnosti.

5.2.3 Tréovanie na sade EN

V tretej sade boli porovnávané modely tréované na dátovej sade RMIT-PD (EN). Prehľadávaný priestor je tvorený týmito augmentáciami:

Typ augmentácie	Použité parametre
Zmena rýchlosti	0,8, 2,2
Časový posun	10, 5
Pridanie šumu	2, 4
Zmena výšky tónu	5, 9

EXPERIMENT	SP			CZ			EN*			IT		
	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
~												
Bez augmentácie	0,801	0,5	0,494	0,843	0,5	0,49	0,393	0,781	0,961	0,704	0,5	0,852
Zmena rýchlosti 0.8	0,696	0,562	0,657	0,867	0,438	0,528	0,468	0,719	0,923	0,721	0,562	0,587
Zmena rýchlosti 2.2	0,665	0,688	0,423	0,842	0,469	0,414	0,582	0,625	0,819	0,759	0,531	0,456
Časový posun 10	0,769	0,531	0,563	0,809	0,469	0,522	0,6	0,656	0,777	0,676	0,531	0,788
Časový posun 5	0,725	0,531	0,606	0,68	0,5	0,795	0,438	0,75	0,987	0,575	0,656	0,842
Pridanie šumu 2	0,638	0,688	0,47	0,75	0,5	0,408	0,637	0,719	0,275	0,716	0,562	0,381
Pridanie šumu 4	0,735	0,469	0,437	0,683	0,562	0,55	0,609	0,719	0,688	0,68	0,562	0,546
Zmena výšky tónu 5	0,676	0,656	0,472	0,773	0,562	0,496	0,545	0,594	0,937	0,671	0,656	0,548
Zmena výšky tónu 9	0,86	0,438	0,403	0,729	0,469	0,584	0,478	0,781	0,803	0,621	0,625	0,804

Tabuľka 5.5: Výsledky tréovania na EN (scenár 2)

Z experimentov zaznamenaných v tabuľke 5.5 boli najlepšie:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	pridanie šumu (2)	0,638	20,4
CZ	časový posun (5)	0,68	19,3
EN	bez augmentácie	0,393	0
IT	časový posun (5)	0,575	18,3

Na základe týchto výsledkov je možné skonštatovať, že najlepšou augmentačnou technikou pre dve zo sád bol časový posun (5). Priemerné zlepšenie bolo na úrovni 14,5 %.

5.2.4 Trénovanie na sade IT

V poslednej časti je použitá na trénovanie dátová sada ItalianPVS (IT) a použité augmentácie sú rovnaké ako v jednom z predošlých prípadov (5.2.2), teda:

Typ augmentácie	Použité parametre
Zmena rýchlosti	0.65, 2.5
Časový posun	3, 6
Pridanie šumu	8, 1
Zmena výšky tónu	5, 9

EXPERIMENT	SP			CZ			EN			IT*		
	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
~												
Bez augmentácie	0,674	0,625	0,649	0,671	0,594	0,567	0,679	0,594	0,48	0,478	0,906	0,976
Zmena rýchlosti 0.65	0,687	0,469	0,523	0,763	0,375	0,565	0,514	0,75	0,86	0,599	0,625	0,842
Zmena rýchlosti 2.5	0,619	0,719	0,69	0,76	0,562	0,565	0,692	0,656	0,565	0,434	0,719	0,938
Časový posun 3	0,748	0,469	0,377	0,721	0,438	0,504	0,625	0,719	0,612	0,547	0,75	0,876
Časový posun 6	0,751	0,406	0,353	0,742	0,469	0,445	0,622	0,594	0,6	0,431	0,844	0,901
Pridanie šumu 8	0,66	0,625	0,582	0,658	0,594	0,764	0,733	0,375	0,246	0,709	0,438	0,332
Pridanie šumu 1	0,7	0,469	0,447	0,74	0,406	0,48	0,679	0,562	0,525	0,703	0,531	0,482
Zmena výšky tónu 5	0,446	0,812	0,899	0,753	0,531	0,51	0,517	0,781	0,766	0,446	0,812	0,899
Zmena výšky tónu 9	0,757	0,375	0,431	0,743	0,531	0,37	0,604	0,719	0,593	0,337	0,938	0,996

Tabuľka 5.6: Výsledky trénovania na IT (scenár 2)

Z tabuľky experimentov 5.6 boli vybraté najlepšie:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	zmena rýchlosti (2.5)	0,619	8,2
CZ	pridanie šumu (8)	0,658	1,9
EN	zmena rýchlosti (0.65)	0,514	24,3
IT	zmena výšky tónu (9)	0,337	29,5

V rámci tejto sady augmentácií nie je možné zvoliť jednu, ktorá by mala pozitívny vplyv na väčšinu dostupných sád, no priemerné zlepšenie dosiahlo 16 %.

5.3 Analýza prístupu na všetkých úlohách

Posledný scenár overuje dátové sady s najvyššou variabilitou. Do spracovávaných dátových sád sú zahrnuté všetky úlohy z každej dátovej sady, ktorá je dostupná (aj keď nie všetky úlohy sa vyskytujú pri všetkých sádach). Spolu sa tak naprieč dátovými sadami spolu vyskytujú úlohy vyslovenie samohlások **a, e, i, o, u** po dobu niekoľkých sekúnd aj vyslovenie slabík **mmm, naa**. Celkové počty vzoriek pre jednotlivé triedy na klasifikáciu boli nasledovné:

Dátová sada	Počet vzoriek HC	Počet vzoriek PD
SP	750	750
CZ	48	48
EN	52	105
IT	110	140

Scenár bol vykonaný s využitím sady PC-GITA (SP) ako trénovacej a s nasledovnými augmentáciami:

Typ augmentácie	Použité parametre
Zmena rýchlosti	0.7, 2
Časový posun	2, 6
Pridanie šumu	3, 1
Zmena výšky tónu	2, 3

Keďže množstvo vzoriek bolo v tomto scenári výrazne väčšie, odrazilo sa to aj na čase vykonania programu. Z tohto dôvodu bol pre overovanie tohto scenára zvolený menší počet augmentácií.

EXPERIMENT	SP*			CZ			EN			IT		
~	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC	LOSS	ACC	AUC
Bez augmentácie	0,172	0,938	0,996	0,908	0,688	0,653	1,158	0,531	0,346	1,64	0,438	0,353
Zmena rýchlosti 0.7	0,252	0,906	0,965	0,911	0,594	0,588	1,478	0,469	0,364	1,584	0,375	0,381
Zmena rýchlosti 2	0,521	0,781	0,853	1,384	0,562	0,412	0,897	0,625	0,286	1,012	0,5	0,466
Časový posun 2	0,145	0,906	0,992	0,878	0,656	0,651	1,405	0,531	0,286	1,676	0,469	0,416
Časový posun 6	0,01	1	1	1,125	0,531	0,637	0,755	0,594	0,563	1,716	0,375	0,32
Pridanie šumu 3	1,596	0,594	0,544	2,022	0,5	0,646	1,293	0,688	0,65	2,124	0,562	0,331
Pridanie šumu 1	1,299	0,5	0,551	1,408	0,469	0,527	1,036	0,656	0,394	1,431	0,531	0,494
Zmena výšky tónu 2	0,364	0,938	0,929	0,86	0,594	0,658	1,396	0,438	0,23	1,46	0,438	0,277
Zmena výšky tónu 3	0,653	0,719	0,747	0,913	0,625	0,62	1,445	0,406	0,131	1,279	0,469	0,402

Tabuľka 5.7: Výsledky trénovania na SP (scenár 3)

Všetky uvedené experimenty v tejto časti sú zhrnuté v tabuľke 5.7. Program vrátil ako najlepšie nasledovné:

Dátová sada	Najlepšia augmentácia	LOSS	Zlepšenie (%)
SP	časový posun (6)	0,01	94,2
CZ	zmena výšky tónu (2)	0,860	5,3
EN	časový posun (6)	0,755	34,8
IT	zmena rýchlosti (2)	1,012	38,3

V poslednom prípade priniesol najlepšie výsledky časový posun (6), no je nutné podotknúť, že v tomto prípade dosiahla presnosť hodnotu 1 pre sadu SP, čo naznačuje pretrénovanie modelu na danej sade. Priemerné zlepšenie pre všetky dátové sady v tomto scenári nadobúda hodnotu 43,1 %.

V súhrnných výsledkoch experimentov naprieč scenármi sa najčastejšie (desaťkrát) vyskytoval časový posun ako najefektívnejšia augmentácia. Ďalšie v poradí boli pridanie šumu (zlepšenie v siedmich prípadoch) a zmena rýchlosti (prínos v šiestich prípadoch). Zmena výšky tónu spôsobila najväčší prínos iba dvakrát, zároveň niekoľko experimentov dosiahlo najlepšie výsledky bez použitia akejkoľvek augmentácie.

Záver

Hlavným cieľom práce bolo realizovať návrh riešenia problému zovšeobeciteľnosti modelov neurónových sietí pre diagnostiku Parkinsonovej choroby z hlasu. V rámci analýzy boli preskúmané dostupné možnosti využívané na zvýšenie zovšeobeciteľnosti v strojovom učení najmä z kategórie dátových augmentácií a automatizácie ich aplikácie. Súčasne bolo zhrnuté aj teoretické pozadie prístupov na analýzu rečových dát a neurónových sietí.

V prvom kroku bola vytvorená konvolučná neurónová sieť slúžiaca na overenie hypotézy nízkej zovšeobeciteľnosti predikcie Parkinsonovej choroby z rečových dát s ohľadom na odlišné jazykové prostredie. Výsledkom prvotných experimentov bola skutočnosť, že model natrénovaný na jednej dátovej sade mal dobrú výkonnosť na danej sade, no schopnosť korektnej predikcie na zvyšných dátových sadách bola rapídne znížená, čím sa podarilo potvrdiť pôvodný predpoklad.

Následne bol navrhnutý a implementovaný prístup využívajúci metódy automatickej dátovej augmentácie. Tento prístup zahŕňa iteračné vytvorenie modelu, aplikácie jednej augmentácie (príp. viacerých augmentácií súčasne) na vzorky zdrojovej dátovej sady, naučenie modelu a porovnanie výsledkov na všetkých dostupných dátových sadách. Výstupom programu je zoznam najlepších augmentačných techník pre jednotlivé sady podľa zvolenej metriky.

Keďže analyzované existujúce prístupy využívajú výhradne obrázkové augmentácie, hlavnou výzvou bol návrh špecifického postupu pre zvukové signály. Riešením je vytvorenie vlastného prístupu, ktorý najprv aplikuje augmentačnú techniku priamo na zvukové vzorky a následne ich prevádza do obrázkovej formy ako Mel spektrogramy potrebné pre predikčný model. Ďalšou výzvou bolo zvoliť vhodný spôsob poskytnutia získaných výsledkov používateľovi, preto bol zahrnutý výpis do konzoly aj zápis do logovacieho súboru. Takto používateľ disponuje aktuálnymi informáciami o prebiehajúcich operáciách v reálnom čase aj ex post s príslušnými časovými značkami.

Poslednými krokmi bolo experimentálne vyhodnotenie v troch scenároch. Počas nich bola vyhodnotená široká škála rôznych nastavení možných kombinácií

augmentačných techník pre rozličné úlohy, ktoré boli súčasťou dostupných dátových sád. Pre každú sadu experimentov boli získané najúspešnejšie augmentačné techniky, ktorých prínos bol percentuálne kvantifikovaný pre jednotlivé dátové sady aj v priemere. Po vyhodnotení jednotlivých scenárov je možné konštatovať, že sa podarilo navrhnuť program, ktorý dokáže automaticky aplikovať špecifikované augmentačné techniky s príslušnými magnitúdami nezávisle od seba i súčasne. Následne dokáže indikovať najlepšie z nich pre každú verifikovanú sadu, čo umožňuje vytvoriť robustnejší model pre konkrétnu požadovanú štruktúru použitých dát, či už sa jedná o jednoduchšiu úlohu, ako vyslovenie jednej samohlásky, cez úlohy s odlišnými zvukovými prejavmi až po dlhšie slabiky, prípadne vety. Preto je možné vyhodnotiť navrhnutý prístup pre zovšeobecniteľnosť modelov ako úspešný. Z uvedených experimentov bola najčastejšie (až v desiatich prípadoch) vyhodnocovaná ako najvýhodnejšia augmentačná technika časového posunu vzorky s rôznymi magnitúdami.

Navrhnuté riešenie, ktoré je výsledkom práce, má potenciál byť cenným príspevkom v oblasti doménového zovšeobecnenia pri spracovaní zvukových signálov aj v ďalších úlohách, napr. identifikácii emócie z reči alebo pri detekcii únavy z hlasu. Okrem toho boli identifikované potenciálne oblasti na dodatočné vylepšenia a rozšírenia zvoleného prístupu. Jedným z nich môže byť zväčšenie priestoru dostupných augmentácií nad rámec implementovaných štyroch. Ďalším zlepšením by bolo náhodné určenie magnitúdy augmentácií, čo by viedlo k jednoduchšiemu nastaveniu.

Literatúra

1. MARTIN, David; FITCH, James; WOLFE, Virginia. Pathologic voice type and the acoustic prediction of severity. *Journal of Speech, Language, and Hearing Research*. 1995, roč. 38, č. 4, s. 765–771.
2. THORNTON, BZJLS. Audio recognition using mel spectrograms and convolution neural networks. 2019.
3. BLOEM, Bastiaan R; OKUN, Michael S; KLEIN, Christine. Parkinson's disease. *The Lancet*. 2021, roč. 397, č. 10291, s. 2284–2303.
4. LI, Da; YANG, Yongxin; SONG, Yi-Zhe; HOSPEDALES, Timothy. Learning to generalize: Meta-learning for domain generalization. In: *Proceedings of the AAAI conference on artificial intelligence*. 2018, zv. 32. Č. 1.
5. HIREŠ, Máté; DROTÁR, Peter; PAH, Nemuel Daniel; NGO, Quoc Cuong; KUMAR, Dinesh. *Strengths and Limitations of Computerized PD Diagnosis from Voice*. Rochester, NY, 2023. Dostupné z DOI: 10.2139/ssrn.4327662.
6. LYU, Junyan; ZHANG, Yiqi; HUANG, Yijin; LIN, Li; CHENG, Pujin; TANG, Xiaoying. AADG: Automatic Augmentation for Domain Generalization on Retinal Image Segmentation. 2022. Dostupné z DOI: 10.1109/TMI.2022.3193146.
7. PANDEY, Ashutosh; WANG, Deliang. On Cross-Corpus Generalization of Deep Learning Based Speech Enhancement. *IEEE/ACM Transactions on Audio Speech and Language Processing*. 2020, roč. 28, s. 2489–2499. ISSN 23299304. Dostupné z DOI: 10.1109/TASLP.2020.3016487.
8. KITTNAR, Otomar. *Lékařská fyziologie*. Praha: Grada, 2011. ISBN 978-80-247-3068-4.
9. FRADKOV, Alexander L. Early history of machine learning. *IFAC-PapersOnLine*. 2020, roč. 53, č. 2, s. 1385–1390.

10. GOROT, Xavier; BENGIO, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In: TEH, Yee Whye; TITTERINGTON, Mike (ed.). *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, zv. 9, s. 249–256. Proceedings of Machine Learning Research. Dostupné tiež z: <https://proceedings.mlr.press/v9/glorot10a.html>.
11. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. arXiv, 2015. Dostupné z DOI: 10.48550/arXiv.1502.01852. arXiv:1502.01852 [cs].
12. ESMAEILI, Ashkan; MARVASTI, Farokh. A novel approach to quantized matrix completion using huber loss measure. *IEEE Signal Processing Letters*. 2019, roč. 26, č. 2, s. 337–341.
13. RUDER, Sebastian. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*. 2016.
14. KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014.
15. DUCHI, John; HAZAN, Elad; SINGER, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*. 2011, roč. 12, č. 7.
16. NESTEROV, Yurii. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In: *Doklady an ussr*. 1983, zv. 269, s. 543–547.
17. RASCHKA, Sebastian; MIRJALILI, Vahid. *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Third edition. Birmingham Mumbai: Packt, 2019. Expert insight. ISBN 9781789955750.
18. O'SHEA, Keiron; NASH, Ryan. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*. 2015.
19. TAN, Chuanqi; SUN, Fuchun; KONG, Tao; ZHANG, Wenchang; YANG, Chao; LIU, Chunfang. A survey on deep transfer learning. In: *Artificial Neural Networks and Machine Learning—ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III* 27. 2018, s. 270–279.

20. DENG, Jia; DONG, Wei; SOCHER, Richard; LI, Li-Jia; LI, Kai; FEI-FEI, Li. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*. 2009, s. 248–255.
21. WANG, Jindong; LAN, Cuiling; LIU, Chang; OUYANG, Yidong; QIN, Tao; LU, Wang; CHEN, Yiqiang; ZENG, Wenjun; YU, Philip. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*. 2022.
22. GHIFARY, Muhammad; BALDUZZI, David; KLEIJN, W Bastiaan; ZHANG, Mengjie. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*. 2016, roč. 39, č. 7, s. 1414–1430.
23. GARCEA, Fabio; SERRA, Alessio; LAMBERTI, Fabrizio; MORRA, Lia. Data augmentation for medical imaging: A systematic literature review. *Computers in Biology and Medicine*. 2023, roč. 152, s. 106391.
24. RAILEANU, Roberta; GOLDSTEIN, Max; YARATS, Denis; KOSTRIKOV, Ilya; FERGUS, Rob. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*. 2020.
25. XU, Mingle; YOON, Sook; FUENTES, Alvaro; PARK, Dong Sun. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*. 2023, roč. 137, s. 109347.
26. SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. A survey on image data augmentation for deep learning. *Journal of big data*. 2019, roč. 6, č. 1, s. 1–48.
27. WEI DI Anurag Bhardwaj, Jianing Wei. *Deep Learning Essentials*. Packt, 2018. ISBN 978-1-78588-036-0.
28. YANG, Zihan; SINNOTT, Richard O; BAILEY, James; KE, Qihong. A survey of automated data augmentation algorithms for deep learning-based image classification tasks. *Knowledge and Information Systems*. 2023, roč. 65, č. 7, s. 2805–2861.
29. RATNER, Alexander J; EHRENBERG, Henry; HUSSAIN, Zeshan; DUNN-MON, Jared; RÉ, Christopher. Learning to compose domain-specific transformations for data augmentation. *Advances in neural information processing systems*. 2017, roč. 30.
30. CUBUK, Ekin D; ZOPH, Barret; MANE, Dandelion; VASUDEVAN, Vijay; LE, Quoc V. Autoaugment: Learning augmentation policies from data. *arXiv 2018. arXiv preprint arXiv:1805.09501*. 1805.

31. CUBUK, Ekin D; ZOPH, Barret; SHLENS, Jonathon; LE, Quoc V. Randaugment: Practical automated data augmentation with a reduced search space. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, s. 702–703.
32. MÜLLER, Samuel G; HUTTER, Frank. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, s. 774–782.
33. RASCHKA, Sebastian. *Comparing Different Automatic Image Augmentation Methods in PyTorch*. 2023. Dostupné tiež z: <https://sebastianraschka.com/blog/2023/data-augmentation-pytorch.html>.
34. KRIZHEVSKY, Alex; HINTON, Geoffrey et al. Learning multiple layers of features from tiny images. 2009.
35. HENDRYCKS, Dan; MU, Norman; CUBUK, Ekin D; ZOPH, Barret; GILMER, Justin; LAKSHMINARAYANAN, Balaji. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*. 2019.
36. PASZKE, Adam; GROSS, Sam et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, s. 8024–8035. Dostupné tiež z: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
37. OROZCO-ARROYAVE, Juan Rafael; ARIAS-LONDONO, Julián David; VARGAS-BONILLA, Jesús Francisco; GONZALEZ-RÁTIVA, María Claudia; NOTH, Elmar. New Spanish speech corpus database for the analysis of people suffering from Parkinson's disease. In: *LREC*. 2014, s. 342–347.
38. RUSZ, Jan; CMEJLA, Roman; TYKALOVA, Tereza; RUZICKOVA, Hana; KLEMPER, Jiri; MAJEROVA, Veronika; PICMAUSOVA, Jana; ROTH, Jan; RUZICKA, Evzen. Imprecise vowel articulation as a potential early marker of Parkinson's disease: Effect of speaking task. *The Journal of the Acoustical Society of America*. 2013, roč. 134, č. 3, s. 2171–2181.
39. VISWANATHAN, R; KHOJASTEH, Parham; ALIAHMAD, Behzad; ARJUNAN, Sridhar Poosapadi; RAGNAV, S; KEMPSTER, P; WONG, Kitty; NAGAO, Jennifer; KUMAR, DK. Efficiency of voice features based on consonant for detection of Parkinson's disease. In: *2018 IEEE Life Sciences Conference (LSC)*. 2018, s. 49–52.

40. DIMAURO, Giovanni; DI NICOLA, Vincenzo; BEVILACQUA, Vitoantonio; CAIVANO, Danilo; GIRARDI, Francesco. Assessment of speech intelligibility in Parkinson's disease using a speech-to-text system. *IEEE Access*. 2017, roč. 5, s. 22199–22208.
41. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Identity mappings in deep residual networks. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. 2016, s. 630–645.
42. CHOLLET, François. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, s. 1251–1258.
43. VAN ROSSUM, Guido; DRAKE, Fred L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.
44. ABADI, Martin; BARHAM, Paul et al. Tensorflow: A system for large-scale machine learning. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, s. 265–283.
45. BIEWALD, Lukas. *Experiment Tracking with Weights and Biases*. 2020. Dostupné tiež z: <https://www.wandb.com/>. Software available from wandb.com.
46. MCFEE, B; METSAI, A; MCVICAR et al. librosa/librosa: 0.9. 1 (2022). URL <https://doi.org/10.5281/ZENODO>. [B.r.], roč. 6097378.
47. KLUYVER, Thomas; RAGAN-KELLEY, Benjamin et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (ed.). *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. 2016, s. 87–90.

Zoznam príloh

Príloha A CD médium – záverečná práca v elektronickej podobe