Bakalářská práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra teorie obvodů

Identifikace mluvčího na bázi hlubokých neuronových sítí

Martin Šubert

Vedoucí: Doc. Ing. Petr Pollák, CSc.
Obor: Komunikace, Multimédia a Elektronika
Studijní program: Multimediální technika
Květen 2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šubert** Jméno: **Martin** Osobní číslo: **457157**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra radioelektroniky**

Studijní program: **Komunikace, multimédia a elektronika**

Studijní obor: **Multimediální technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Identifikace mluvčího na bázi hlubokých neuronových sítí

Název bakalářské práce anglicky:

Speaker Identification Based on Deep Neural Networks

Pokyny pro vypracování:

1. Seznamte se s problematikou identifikace mluvčího a proveďte přehledovou rešerši aktuálně používaných technik na bázi GMM, i-vektorů a DNN včetně srovnání typicky dosahovaných výsledků.
2. Vybrané metody implementujte s nástroji KALDI a srovnejte přístupy na bázi DNN s identifikací na bázi GMM a i-vektorů.
3. V experimentální části ověřte funkčnost implementovaných metod a vyhodnoťte přesnost identifikace mluvčího na dostupných databázích mluvené řeči za různých akustických podmínek.
4. Na základě zavedených konvencí vytvořte pro vybrané metody a dostupné databáze skripty ('recepty') umožňující jejich budoucí využití na řešitelském pracovišti či odbornou komunitou.

Seznam doporučené literatury:

[1] J. Psutka, L. Müller, J. Matoušek, V. Radová. Mluvíme s počítačem česky. Academia 2006.
[2] D. Povey et al, The Kaldi Speech Recognition Toolkit. In Proc. of IEEE 2011 ASRU, Hawai, US, 2011.
[3] F. Richardson, D. Reynolds, N. Dehak: Deep Neural Network Approaches to Speaker and Language Recognition. IEEE Signal Processing Letters, Vol. 22, No. 10, October 2015.
[4] P. Matějka et al, Analysis of DNN Approaches to Speaker Identification. In Proc of ICASSP 2016, Shanghai, China, pp.5100-5104

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Petr Pollák, CSc., katedra teorie obvodů FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **01.02.2019** Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

doc. Ing. Petr Pollák, CSc.	prof. Mgr. Petr Páta, Ph.D.	prof. Ing. Pavel Ripka, CSc.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Mé poděkování patří panu doc. Ing. Petru Pollákovi, CSc., za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 24. května 2019

Abstrakt

Tato práce se zabývá metodami verifikace a identifikace řečníka. Hlavní pozornost je věnována především metodám založeným na bázi GMM resp. i-vektorů. Na teoretické úrovni jsou popsány metody využívající hlubokých neuronových sítí. Implementace byla vytvořena pro systém na bázi GMM resp. i-vektorů, včetně použití LDA a PLDA při výpočtu skóre pro zvýšení přesnosti identifikace resp. verifikace. Bylo využito nástrojů KALDI, které jsou přímo určeny pro úlohy rozpoznávání řeči a rozpoznávání řečníka.

Praktická část se zaměřuje především na otestování vlivu počtu a rozložení mluvčích a promluv v rámci jednotlivých trénovacích a testovacích množin. Testování bylo provedeno pro databázi GLOBALPHONE obsahující promluvy několika světových jazyků. Z výsledných hodnot testování lze říci, že s rostoucím počtem promluv použitých pro referenční a testovací množinu dochází k poklesu chyby při verifikaci a identifikaci mluvčího. Tato implementace je základem systému na bázi DNN, kdy velmi často používanou konfigurací je nepřímé použití neuronových sítí pro výpočet příznaků s následnou identifikací na bázi i-vektorů. Konkrétním výsledkem je vzorový skript (recept) dle konvence KALDI, který může být použitý pro navazující implementaci systému s DNN.

Klíčová slova: Rozpoznávání řečníka, Neuronové sítě, DNN, KALDI, GMM, i-vektor, UBM

Vedoucí: Doc. Ing. Petr Pollák, CSc.

Abstract

This work is focused on methods using in speaker recognition. The main attention is paid to methods based on GMM using i-vectors. At the theoretical level, methods using deep neural networks are described. Implementation was created for the GMM-based systems using i-vectors, including the use of LDA and PLDA. KALDI tools have been used that are directly designed for speech recognition and speaker recognition tasks.

The practical part focuses mainly on testing the influence of the number and distribution of speakers and utterances within individual training and testing sets. Testing was done for the GLOBALPHONE database containing several world languages. Based on the results, it can be said that with a higher number of reference and test utterances the identification and verification error decrease. This implementation is the basis of the DNN-based system. Often used configuration is the indirect use of neural networks to calculate the features followed by i-vector based identification. The concrete result is a script (recipe) according to the KALDI convention, which can be used for further implementation of the DNN system.

Keywords: Speaker recognition, Neural network, DNN, KALDI, GMM, i-vector, UBM

Title translation: Speaker Indentification Using Deep Neural Networks

Obsah

Obrázky

Tabulky

Kapitola 1

Úvod

Komunikace je proces, který je součástí našeho všedního života. Díky němu dochází ke sdělování informací, myšlenek, názorů či pocitů. Komunikace probíhá především verbálně (řeč) či pomocí textu, pokud se jedná o komunikace člověka s člověkem. V dnešní době dochází k rozvoji komunikace mezi člověkem a strojem. K tomuto typu komunikace dochází stále častěji, ať už se jedná o počítač, mobilní telefon či jiné stroje. Proto je úloha rozpoznávání řeči a rozpoznávání řečníka velmi rozvíjena.

Jako jednu z aplikací identifikace řečníka si lze představit kontrolní systém, který ověřuje osoby, které chtějí vstoupit do určitě budovy. Systém na základě promluvy analyzuje danou osobu a rozhodne, zda má či nemá povolení vstoupit. Další aplikací může být chytrá domácnost. Přes hlasového asistenta lze například spouštět oblíbenou hudbu, regulovat pokojovou teplotu nebo třeba ztlumit osvětlení na požadovanou úroveň. Pokud chce člověk vždy, když přijde domů, mít pokojovou teplotu 22°C a aby hrála jeho oblíbená hudba, tak místo toho, aby vždy znovu zadával své požadavky, může si uložit dané nastavení. V momentu, kdy asistent rozpozná jeho hlas, může aplikovat dané nastavení bez dalších pokynů. Takto by mohl mít pak nastavený svůj profil každý člen rodiny.

Přestože v dnešní době různé systémy dosahují již velmi vysokého procenta spolehlivosti a správnosti identifikace mluvčího, není to vždy 100%. Z tohoto důvodu je třeba při aplikaci rozlišovat, jak moc musí být systém "přísný". K tomu slouží verifikace, která na základě určitých parametrů ověřuje, zda se skutečně jedná o předpokládanou osobu.

Tato práce se zabývá realizací rozpoznávání řečníka se zaměřením na metody GMM, resp. i-vektorů a s využitím hlubokých neuronových sítí. První část se bude zabývat standardními metodami na základě statistických metod jako je GMM s využitím i-vektorů. Popsán bude celý proces od přípravy dat až k dané identifikaci a verifikaci řečníka. Druhá kapitola se bude věnovat principům a typickému použití neuronových sítí pro úlohu rozpoznávání řečníka.

V experimentální části dojde k implementaci vybraných metod a jejich otestování pro telefonní databázi GLOBALPHONE. Pro tuto implementaci budou použity nástroje KALDI, které jsou přímo určené pro úlohy rozpoznávání řeči a rozpoznávání řečníka. Testování bude zaměřeno především na zjištění vlivu rozložení a počtu mluvčích a promluv v jednotlivých množinách dat.

Kapitola 2

Rozpoznávání řečníka na bázi GMM a i-vektorů

V úloze rozpoznávání řečníka se řeší dva výstupy implementace - identifikace a verifikace. Při identifikaci dochází k přesnému určení řečníka. Výstupem je tedy jméno či ID příslušného mluvčího. Pomocí verifikace se pak pouze ověřuje, zda se jedná právě o daného řečníka. Tedy na základě určitých parametrů dostaneme výstup, který dává informaci o tom, zda se skutečně jedná o předpokládaného řečníka či nikoli.

Úlohu rozpoznávání řečníka lze dělit na dvě různé metody - subjektivní a objektivní metodu. Subjektivní metoda je taková, kdy záleží na subjektivním pohledu dané osoby či experta, který hodnotí podobnost určitých promluv řečníků. Tyto metody nelze jednoduše matematicky popsat a nejsou tedy vhodné pro implementaci v počítačové technice. Objektivní metoda má svou jasnou strukturu, kdy pro jeden konkrétní vstup je jednoznačný jeden výstup a jsou tedy vhodné pro implementaci v elektronice. Právě objektivní (automatickou) metodou se bude tato práce dále zabývat.

Kromě dělení na subjektivní a objektivní metodu lze rozdělit jednotlivé implementace také na textově závislé a textově nezávislé. U textově závislých realizací je předem stanovené slovo či věta, kterou musí identifikující osoba pronést. U porovnání promluvy je tedy zahrnut i její obsah. Tato metoda není využita při implementaci experimentální části této práce. Textově nezávislá realizace nehledí na obsah promluvy a bere v potaz pouze příznaky dané promluvy. Pro tuto metodu se mohou využívat právě příznaky MFCC, které jsou využity v implementaci práce.

2.1 Řečový signál a jeho vlastnosti

U komunikace je informace přenášena určitými elementy. Těm se u řeči říká *fonémy*. Řetězením těchto fonémů vzniká nekonečné množství slov či spojení. Z toho lze odvodit, že řeč je jednorozměrným signálem a lze jí tedy analyzovat pouze podle jejího časového vývoje. Fonémy jsou tedy chápány jako obecná reprezentace elementů řeči. Pro vytvoření konkrétního slova jsou zapotřebí určité fonémy, kterým se říká *hlásky*.

2.1.1 Fyzikální charakteristiky řeči

Jako fyzikální charakteristiky jsou chápány takové vlastnosti řeči, které byly člověku dány od narození a mění se bez většího vlivu dané osoby. Týká se to například hlasivkového tónu či formantových frekvencí, jelikož jsou závislé na vnitřní stavbě hlasového ústrojí. Jsou to charakteristiky, které nelze napodobovat jinou osobou. Z tohoto důvodu jsou fyzikální charakteristiky řeči velmi důležité při rozpoznávání řečníka.

U znělých oblastí řečového signálu lze detekovat určitou periodicitu, která je dána kmitáním hlasivek. Průměrná frekvence takového kmitání se označuje jako základní frekvence neboli hlasivkový tón. Základní frekvenci má každý člověk různou díky odlišnosti hlasivek. Tento fakt je důležitý pro analýzu a syntézu řeči.

Dutiny hlasového ústrojí jsou do určité míry flexibilní a mohou měnit svůj tvar a objem. Od toho se pak odvíjí frekvence, při které dutiny rezonují při průchodu vzduchu z plic. Tyto rezonančí frekvence se označují jako *formanty*. Ze stejného důvodu jako u hlasivkového tónu jsou i formanty různé pro různé lidi. Díky tomu lze formanty označit jako další specifickou vlastnost jednotlivých lidí, čehož se využívá opět při analýze a syntéze řeči.

2.1.2 Naučené charakteristiky řeči

Za naučené charakteristiky jsou považovány takové vlastnosti, které jsou získávány během života a mění se s velkým vlivem daného člověka. Můžou to být různé návyky či zvláštnosti, které člověk své řeči přidal. Přestože jsou tyto vlastnosti také většinou specifické pro každou osobu, lze je snáze napodobit a neklade se jim tedy takový důraz při aplikacích rozpoznávání řečníka.

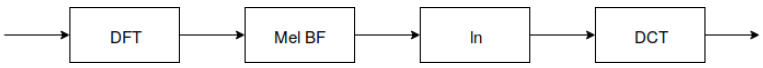
2.1.3 Zpracování řečového signálu

Pro implementaci rozpoznávání řečníka se využívají data, která jsou již v číslicové podobě. Analogový řečový signál je tedy nutné převést do digitální podoby. Prvním krokem zpracování řečového signálu je jeho digitalizace. Ta je závislá na dvou krocích - na diskretizaci a na kvantizaci. Nejdůležitějšími parametry jsou potom vzorkovací frekvence a počet bitů na jeden vzorek. Při vzorkování musí být dodržen Shannon (Nyquist) teorém, aby nedošlo k chybám jako je aliasing či jiné zkreslení signálu. U kvantování je důležitý počet bitů na jeden vzorek a kvantovací krok. Tyto údaje ovlivňují kvalitu výsledné digitální podoby řečového signálu. Pro úlohu rozpoznání mluvčího se snažíme mít co nejvyšší vzorkovací frekvenci a co největší počet bitů při kvantování, jelikož analýza zahrnuje vlastnosti a specifikace i pro vyšší kmitočty. Pro digitalizaci analogového signálu se používají různé metody. Jako nejznámější lze považovat pulsně kódovou modulaci (PCM). PCM je lineární modulace, kde nejdříve dochází ke vzorkování spojitého signálu a jeho následného kvantizování.

2.2 Výpočet kepstrálních příznaků

Aby došlo ke správnému porozumění řečových příznaků, je třeba si uvědomit, jak je lidská řeč tvořena. Řečový signál je generován výdechem plic, kdy vzduch prochází hlasovým a artikulačním ústrojím. V těchto oblastech je signál filtrován podle velikosti dutin a tvaru jazyku, zubů a podobně. Ideálně by tedy řečové příznaky měly co nejblíže specifikovat daný "filtr".

Výpočet příznaků dochází pomocí kepstrální analýzy. Proces výpočtu pomocí kepster lze vidět na obr. 2.1. Diskrétní kosinová transformace (DCT) nahrazuje pro reálný signál při výpočtu kepstra zpětnou Fourierovu transformaci (IFFT).



Obrázek 2.1: Blokové schéma výpočtu mel-kepstrálních koeficientů

Pro použití řečového signálu k rozpoznávání řečníka jsou po zpracování signálu do digitální podoby provedeny další dva kroky:

- Preemfáze - u lidské řeči lze pozorovat klesající úroveň amplitudy s rostoucí frekvencí. To je nutné kompenzovat a odstranit tento jev. Jako kompenzace se přidává filtr typu horní propust, kde jsou potlačeny dolní frekvence a srovná se tedy úroveň amplitud v celé šířce spektra.

- Krátkodobá spektrální analýza - po preemfázi následuje rozdělení signálu do časově omezených úseků pomocí váhovacích oken. Váhovací okno se posouvá po časové ose signálu a vybírá daný počet vzorků, kdy zbytek vzorků je vynulován. Na daném segmentu je pak prováděna spektrální či kepstrální analýza. Váhovací okno délky N se posouvá o N nebo méně vzorků.[1]

Při předpokladu, že jsou všechna data (audio signály) připraveny k úloze rozpoznávání řečníka, dochází k extrakci řečových příznaků. Řečové příznaky jsou takové příznaky, které charakterizují daný řečový signál. Při tomto procesu dochází k detekci obsahu promluvy, od které jsou odděleny informace jako například šum v pozadí či emoce promluvy a podobně. Z toho je patrné, že výpočet řečových příznaků se používá jak v úloze rozpoznání řečníka, tak v úloze rozpoznání řeči. Jednotlivé aplikace však využívají odlišné části těchto příznaků.

Mel Frequency Cepstral Coefficents (MFCCs) jsou koeficienty, které popisují dané řečové příznaky popsané výše. Extrakce těchto koeficientů z řečového signálu probíhá v následujících krocích:

1. preemfáze,
2. rozdělení signálu na segmenty, váhování segmentů,
3. výpočet FFT,
4. výpočet energie v pásmech pomocí banky filtrů,
5. logaritmus energie v pásmech,
6. výpočet DCT.

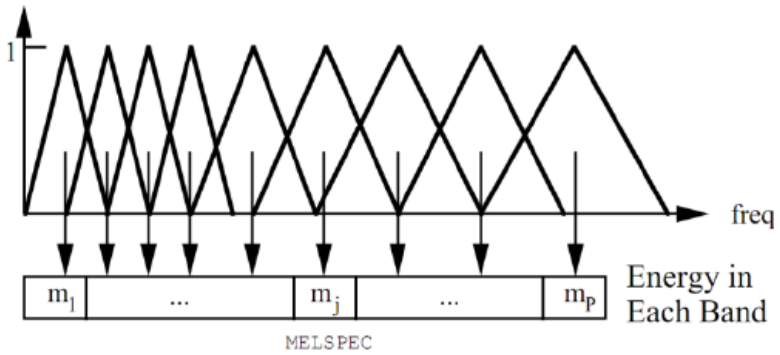
První dva kroky jsou vysvětleny na začátku kapitoly 2.2. Výpočtem FFT (Fast Fourier Transform) je získáno spektrum daného segmentu. V tomto okamžiku signál obsahuje velké množství informací, které jsou pro úlohu rozpoznávání řečníka či řeči nepodstatné. Jedna z věcí je neschopnost lidského ucha do určité míry nerozpoznat malou změnu kmitočtu u zvuku. Tento jev se zvětšuje se zvyšující se frekvencí. Tomuto jsou uzpůsobené i banky mel-filtrů.

Banky mel-filtrů jsou trojúhelníkové filtry, které jsou u nízkých kmitočtů velmi úzké a stávají se širšími s rostoucí frekvencí. Jak přesně rozložit filtry na dané spektrum říká mel stupnice. Ta je vypočtena dle vztahu

$$f_{mel} = 1125 \ln(1 + f/700)\,. \tag{2.1}$$

[1] Délka posunu záleží na volbě váhovacího okna. Například nástroje KALDI jako výchozí okno používají tzv. Povey okno, které je téměř stejné jako Hammingovo okno. U něho je překryv 50%.

Na signál jsou tedy aplikovány banky filtrů, kde jsou poté získány jednotlivé energie, jak lze vidět na následujícím obrázku.



Obrázek 2.2: Aplikace banky mel filtrů a vzetí jejich energií [1]

Čtvrtým krokem je aplikování logaritmů na jednotlivé energie. Tento krok vychází opět z definice kepstra.

2.3 VAD - Voice Activity Detection

VAD detekuje úseky, kde signál obsahuje řeč a kde nikoliv. V ideálním případě, kdy signál obsahuje pouze řeč a ticho, by byla implementace velmi jednoduchá. Nicméně to v reálném světě takřka nelze. Vždy bude signál obsahovat různé šumy, které ztěžují detekci řeči vůči tichu. Proces VAD probíhá v následujících krocích

1. rozdělení signálu na segmenty, váhování segmentů,
2. výpočet energie z krátkodobých segmentů.

První dva kroky jsou stejné jako u výpočtu MFCC. Následně dochází k extrakci charakteristik, které právě popisují, kde je řeč a kde není. Nejedná se o jednu charakteristiku, která by přesně charakterizovala řeč a ticho. Naopak je vzato určité množství charakteristik, jejichž kombinací lze s určitou přesností detekovat řeč v signálu.

2.4 Metody klasifikace na bázi UBM-GMM

Metody klasifikace založené na směsi Gaussovských hustotních funkcí, neboli GMM (Gaussian Mixture Model), jsou založeny na spojitém rozdělení pravděpodobnosti příznaků řeči, které nejlépe modelují řečníka či model pozadí [21].

GMM je použito pro porovnání vlastností řečových příznaků MFCC, kdy se hledá největší pravděpodobnost pro daný model. Jsou tedy natrénovány GMM modely pro každého řečníka v trénovací databázi, které jsou pak pomocí GMM porovnávány s modely testovaných řečníků.

Směs Gaussovských hustotních funkcí se skládá z mnoha Gaussovských distribucí. Gaussovské rozložení pravděpodobnosti o d-dimenzích pro vektor $x = (x^1, x^2, ..., x^d)^T$ je definováno jako

$$N\left(x \,|\, \mu_j, \Sigma_j\right) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}\,, \tag{2.2}$$

kde μ je vektor střední hodnoty a Σ kovarianční matice. Výpočet pravděpodobnosti shody referenčního modelu s modelem z databáze je proveden pomocí vztahu

$$p(x) = \sum_{j=1}^{K} w_j \cdot N\left(x|\mu_j, \Sigma_j\right), \tag{2.3}$$

kde K reprezentuje počet distribucí GMM a w_j je váha dané distribuce j a vztahují se na ní následující podmínky

$$\sum_{j=1}^{K} w_j = 1, \qquad w_j \geq 0\,. \tag{2.4}$$

UBM (Universal Background Model) je efektivní framework, který je nezávislý na mluvčím. Principiálně je to velké množství Gaussovských hustotních funkcí. Tento model je adaptován na jednotlivé řečníky pomocí tzv. MAP (a maximum a posteriori) schématu. UBM je trénován na velkém množství dat a následně se využívá při verifikaci, kdy se porovnává UBM-GMM model s jednotlivými GMM modely daných řečníků.

Pro správné vytvoření UBM modelu je důležité, aby byl model trénován na datech ve stejné doméně, jako budou data testovací. Doménou lze zahrnout velké množství skutečností. Pokud by měla být testovací data tvořena pouze z řečníků stejného pohlaví, UBM model by měl být trénován také na daném pohlaví. V doméně mohou být zahrnuty i jevy jako je vada řeči, typ mikrofonu, prostředí nahrávek řečníků a podobně. Při trénování UBM modelu je pak důležité, co nejvíce tyto podmínky přizpůsobit, aby odpovídaly testovacím datům.

Další důležitý parametr při vytváření UBM modelu je volba počtu GMM komponent. Tento parametr závisí na počtu dat, které máme k trénování modelu k dispozici. Čím více dat je použito při trénování, tím více GMM komponent by mělo být zvoleno. Zároveň je nutné si dát pozor na to, aby nedošlo k přetrénování modelu. V tomto případě by pak došlo k nepřesnostem ve výpočtu a ke zkreslení výsledných dat.

2.5 Použití i-vektorů jako reprezentace mluvčích

Pro reprezentaci mluvčího v UBM-GMM modelu se využívá tzv. supervektor. Ten je vytvořen pouze středními hodnotami μ_c všech komponent modelu, který vznikl z UBM-GMM modelu mluvčích. Dimenze supervektoru je vypočtena jako $C \cdot F$ kde C je počet komponent GMM modelu a F dimenze příznakového vektoru [16].

Tento supervektor obsahuje charakteristiky pro dané mluvčí, ale také velké množství informací, která není v tomto případě nijak důležitá. Proto se používají takové vektory, které mají menší dimenzi, ale stále obsahují všechny charakteristiky, které charakterizují daného řečníka.

Matematické vyjádření supervektoru je popsáno ve vztahu (2.5). Vztah je uvažován při zjednodušeném modelu faktorové analýzy. Ve vztahu (2.5) $m_{r,s}$ symbolizuje supervektor, kde index r znázorňuje danou promluvu a index s daného mluvčího. Parametr μ jsou střední hodnoty UBM modelu. Matice T zajišťuje redukci supervektoru a její dimenze je odvozena jako $C \cdot F \times D_{ivec}$, kde D_{ivec} je dimenze i-vektoru

$$m_{r,s} = \mu + T x_{r,s} \,. \tag{2.5}$$

Ze vztahu (2.5) pak lze vyjádřit výsledný i-vektor $x_{r,s}$, čímž vznikne výsledný vztah pro výpočet i-vektoru pro daného řečníka a jeho promluvu

$$x_{r,s} = T^{-1}(m_{r,s} - \mu) \,. \tag{2.6}$$

Trénování extraktoru probíhá pomocí Baum-Welch algoritmu, kdy jsou napočítané statistiky centrovány kolem vektoru μ [16]. Následně je vytvořen prostor celkové variability definovaný maticí T.

2.6 Výpočet skóre

Po extrakci vektorů může dojít k výpočtu skóre, které udává podobnost dvou daných mluvčích (jejich i-vektorů). Pro výpočet skóre se využívá kosinová vzdálenost dle vztahu

$$score_{ivec}(x_1, x_2) = \frac{\langle x_1, x_2 \rangle}{\|x_1\| \cdot \|x_2\|} \,, \tag{2.7}$$

kde x_1 je i-vektor referenčního mluvčího a x_2 vektor testovacího mluvčího.

2.6.1 Lineární diskriminační analýza LDA

Pro zvýšení schopnosti rozlišovat jednotlivé mluvčí lze využít LDA metodu, která spočívá v lineární transformaci i-vektorů. Další výhodou této metody

je snížení dimenze i-vektoru. Matematicky lze proces vyjádřit dle vztahu

$$x' = A^T x\,, \tag{2.8}$$

kde x je původní n-dimenzionální vektor a x' výsledný vektor s dimenzí m, kdy zároveň platí, že $m < n$. Parametr A^T je transformační matice o rozměru $m \times n$.

Dále se provádí optimalizace v závislosti na transformační matici A. Pro lepší pochopení této optimalizace je nutné pochopit pojem tříd. Daná třída obsahuje více promluv jednoho mluvčího. Rozptyl v rámci jedné třídy je tedy rozptyl mezi promluvami jednoho stejného mluvčího. Tento rozptyl je ideální snížit, aby mezi jednotlivými promluvami nebyl takový rozdíl. Naopak rozptyl mezi jednotlivými třídami je nutné zvětšit, aby došlo k lepšímu rozpoznání jednotlivých mluvčích. Tuto optimalizaci $J(A)$ lze matematicky vyjádřit dle vztahu

$$J(A) = tr((A^T \Sigma_w A)^{-1}(A^T \Sigma_B A))\,, \tag{2.9}$$

kde matice Σ_W je celková kovariance uvnitř tříd počítaná přes všechny třídy a Σ_B je kovariance mezi třídami.

2.6.2 Pravděpodobnostní lineární diskriminační analýza PLDA

PLDA metoda byla vytvořena jako alternativa k LDA, kde se využívá pravděpodobnosti a nedochází zde ke snížení dimenze i-vektoru. Metoda je vytvořena na základě faktorové analýzy a matematicky je vyjádřena pomocí vztahu

$$x_{r,s} = \mu + V y_s + U w_{r,s} + \epsilon_{r,s}\,, \tag{2.10}$$

kde $x_{r,s}$ je vstupní vektor mluvčího s. Parametr $\epsilon_{r,s}$ vyjadřuje variabilitu prostředí s rozložením $P(\epsilon_{r,s}) = N(0, \Sigma)$, kde Σ je diagonální kovarianční matice. Model má dvě složky

- s_s zastává charakteristiku mluvčího s a je konstantní pro všechny vlastní promluvy,
- $c_{r,s}$ zastává proměnné akustické podmínky prostředí a analogové části nahrávacího přístroje,

a lze je matematicky vyjádřit pomocí následujících vztahů

$$s_s = \mu + V y_s\,, \tag{2.11}$$

$$c_{r,s} = U w_{r,s} + \epsilon_{r,s}\,. \tag{2.12}$$

Trénování toho modelu probíhá pomocí EM algoritmu a jsou zde trénovány následující parametry:

- V - zátěžová matice pro model rozdílů mezi mluvčími,
- μ - vektor,
- U - zátěžová matice pro model akustických podmínek,
- Σ - matice.

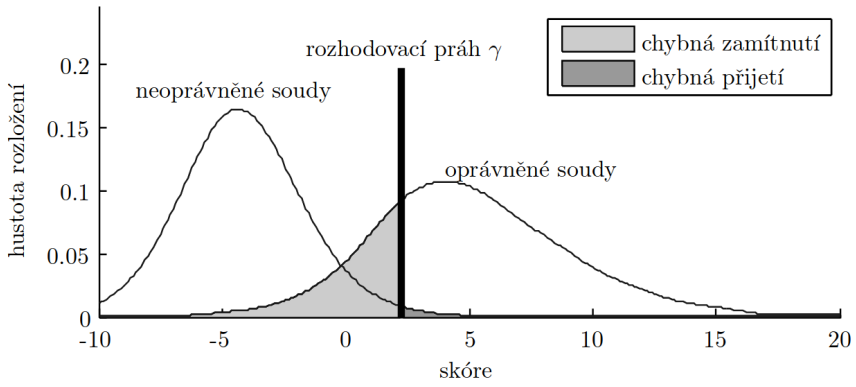
Tato metoda by měla být přesnější oproti například kosinové vzdálenosti. Její přesnost pak stoupá při vzrůstajícím počtu promluv daných mluvčích a při větší variabilitě akustického prostředí, ve kterém jsou nahrávky pořizovány.

2.7 Verifikace a její hodnocení

U kterékoli implementace rozpoznávání řečníka zatím přesnost identifikace není 100%. Z toho důvodu se zkoumá i přesnost verifikace mluvčího. Cílem verifikace je pouze ověření, zda předpokládaný mluvčí, je skutečně daná osoba. Vždy se tedy vezme dvojice mluvčích - předpokládaný mluvčí a referenční mluvčí a na základě vypočítaného skóre mezi nimi mohou nastat následující situace:

1. Správné přijetí (True Acceptance) - předpokládaný mluvčí byl správně verifikován jako referenční mluvčí.
2. Chybné přijetí (False Acceptance) - předpokládaný mluvčí byl verifikován jako referenční mluvčí, přestože se nejedná o stejnou osobu.
3. Chybné zamítnutí (False Rejection) - předpokládaný mluvčí byl chybně odmítnut, jelikož předpokládaný mluvčí je stejná osoba jako refereční mluvčí.
4. Správné zamítnutí (True Rejection) - předpokládaný mluvčí byl správně odmítnut, jelikož se nejedná o stejnou osobu jako je referenční mluvčí.

Tyto parametry velmi závisí na verifikačním prahu γ. Práh je určitá hodnota, podle které se určí, zda testovaný mluvčí bude verifikován či naopak. Pokud bude tedy hodnota skóre vetší než daný práh, dojde k přijetí mluvčího. Posuvem tohoto prahu lze ovlivnit množství přijetí či odmítnutí, jak je znázorněno na obrázku 2.3. Volba prahu se většinou liší dle aplikace.



Obrázek 2.3: Nastavení rozhodovacího prahu [16]

Míru chybného přijetí P_{FA}, nebo také FAR (False Acceptance Rate), při předpokladu rozhodovacího prahu γ, počtu chybných přijetí $N_{FA}(\gamma)$ a počtu neoprávněných soudů N_{NS}, lze zjistit dle vztahu

$$P_{FA}(\gamma) = \frac{N_{FA}(\gamma)}{N_{NS}}\,. \tag{2.13}$$

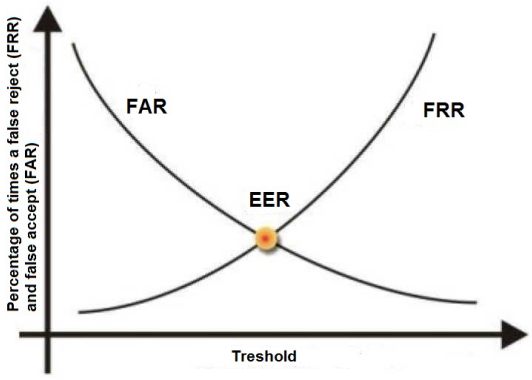
Pro výpočet míry chybného zamítnutí P_{FR}, nebo také FRR (False Rejection Rate), při předpokladu rozhodovacího prahu γ, počtu chybných zamítnutí $N_{FR}(\gamma)$ a počtu oprávněných soudů N_{OS}, lze využít vztah

$$P_{FR}(\gamma) = \frac{N_{FR}(\gamma)}{N_{OS}}\,. \tag{2.14}$$

Dále se zjišťuje tzv. EER (Equal Error Rate). EER je procentuální vyjádření, kdy hodnota FAR je stejná jako FRR, jak je zobrazeno na obrázku 2.4. Matematicky lze EER vyjádřit jako

$$EER = FAR = FRR\,, \tag{2.15}$$

kde je hodnota EER závislá především na volbě verifikačního prahu.



Obrázek 2.4: Equal Error Rate [17]

2.8 Identifikace a její hodnocení

Při identifikaci řečníka dochází k přesnému určení jména či ID dané osoby. Pro každého řečníka je tedy nalezena osoba s největším skóre napočítaným v předchozích krocích a dochází k identifikaci. Při implementaci identifikace mluvčího je nutné rozlišovat, o jaký typ množiny se jedná.

- **Uzavřená množina** - kdy jsou v databázi uloženy již vzorky jednotlivých mluvčích. Z těchto vzorků jsou vytvořeny jednotlivé statistické modely, pomocí kterých pak dochází k identifikaci mluvčího. Pro danou osobu je následně nalezena největší shoda (maximum) s modelem z databáze. Při správné implementaci by tedy dvojice mluvčích s největším skóre měla být ta samá osoba.
- **Otevřená množina** - kdy se předpokládá, že testované osoby nemusí mít již vytvořený model v dané databázi. V tomto případě může nastat situace, kdy testovanému mluvčímu je nalezena dvojice s největším skóre (největší podobností), ale nejedná se o stejnou osobu. Testovaný mluvčí totiž nemá žádnou shodu v databázi a měl by být správně odmítnut či neidentifikován. Z toho důvodu je vhodné použít verifikaci, která na základě verifikačního prahu ověří, zda daná dvojice je skutečně ten samý mluvčí.

Chybu přesnosti identifikace $P_{id_{err}}$, při předpokladu chybné identifikace n_{err} a celkovému počtu dvojic mluvčích n_{all}, lze zjistit dle vztahu

$$P_{id_{err}} = \frac{n_{err}}{n_{all}} \,. \tag{2.16}$$

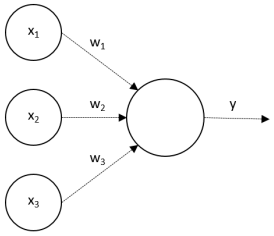
Kapitola 3

Rozpoznávání řečníka s použitím DNN

Během tohoto desetiletí se neuronové sítě začaly stále více využívat jako jeden z nástrojů pro strojové učení. K aplikaci sítí došlo i při implementacích úlohy identifikace řečníka. Neuronové sítě se zde mají přímé i nepřímé využití, kdy při přímém využití dochází k implementaci úlohy téměř jenom díky neuronovým sítím. Neuronové sítě při nepřímém využití pomáhají s částí implementace, kdy díky nim dochází například k výpočtu příznaků a lze je dále kombinovat se standardními metodami zmíněnými v předchozí kapitole.

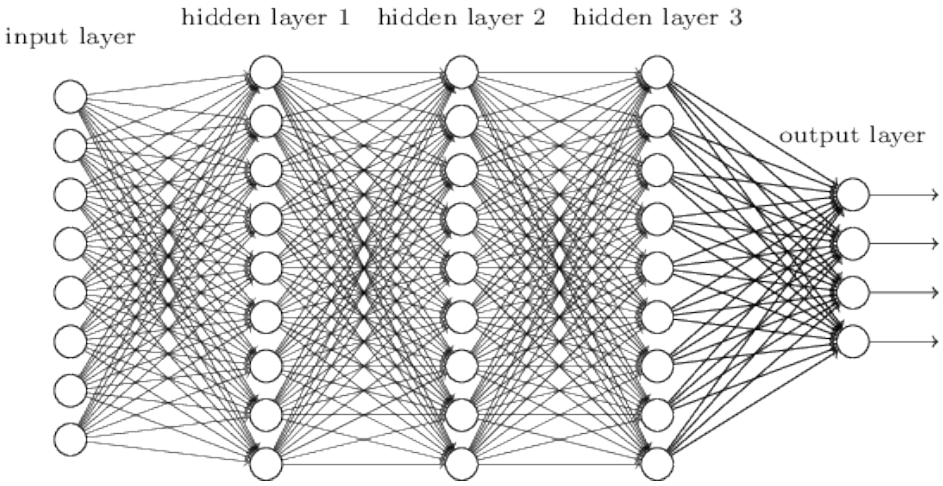
3.1 Umělé a hluboké neuronové sítě

Umělé neuronové sítě (ANN - Artificial Neural Networks) jsou v dnešní době stále více zkoumaným a aplikovaným nástrojem a staly se jedním z nejvíce úspěšných algoritmů strojového učení (Machine Learning). ANN jsou inspirovány fungováním neuronů v lidském mozku. Ten obsahuje asi 100 bilionů neuronů pospojovaných synapsemi. Jednotlivé neurony vysílají určitě impulsy, které se přenáší právě synapsemi. Tento proces je známý jako „myšlení“. Neurony v ANN jsou reprezentovány jako jednotlivé uzly (nodes) a synapse jako váhy (weight edges). Váhy jsou určité koeficienty, které jsou nastaveny tak, aby síť dosahovala nejlepších výsledků. Nastavení těchto koeficientů se většinou děje na základě určitého učení sítě. K tomu, aby byla implementace kvalitní a neuronová síť dosahovala chtěných výsledků, je typicky potřeba velké množství dat, což je jedna z největších překážek pro širší aplikace neuronových sítí. Jednoduchá neuronová síť je zobrazena na obrázku 3.1, kde jsou znázorněny vstupy x_n, přenásobeny vahami w_n a výstup y.



Obrázek 3.1: Jednoduchá neuronová síť [7]

Hluboké neuronové sítě DNN (Deep Neural Networks) jsou všechny neuronové sítě, které se skládají z více než dvou vrstev. K vstupní a výstupní vrstvě zde přibývají takzvané skryté vrstvy. Těchto vrstev může přibývat či ubývat dle druhu aplikace. Zároveň se může lišit počet uzlů či vah v jednotlivých vrstvách. Hluboká neuronová síť se dvěma skrytými vrstvami je zobrazena na obrázku 3.2.



Obrázek 3.2: Hluboká neuronová síť se skrytými vrstvami [12]

3.2 Trénování hlubokých sítí

Před použitím sítě v určité aplikaci je nutné síť natrénovat tak, aby byla vhodná pro danou implementaci. Trénování je proces, u kterého dochází k vytváření a korekci jednotlivých hodnot neuronů a vah mezi nimi. Pro tyto účely se využívají například metody cross-entropy trénování či sekvenční diskriminativní trénování. K trénování nemůže docházet náhodně, jelikož by se mohla zanést chyba do výsledné implementace a síť by negenerovala chtěné výsledky. Proto dochází k určité inicializaci sítě. Tento proces vytvoří výchozí hodnoty či nastavení pro následující trénování sítě. Pro inicializaci sítě se používají techniky jako například Restricted Boltzmann Machine (RBM) či Deep Belief Network (DBN).

3.2.1 Cross-entropy trénování

Jak bylo zmíněno v kapitole 3.2, DNN má více než tři vrstvy, kde jsou neurony pospojovány vahami. Váhy ovlivňují výsledný parametr neuronové sítě a k jejich výpočtu dochází trénováním. Na zjištění správných koeficientů vah se používá cross-entropy trénování. Z řeči lze extrahovat velké množství různých typů příznaků s tím, že každý se v řeči vyskytuje s různou pravděpodobností. Lze si například představit, že u dané osoby došlo ke zranění v oblasti úst a její řeč bude atypická. Toto není běžný jev a objevovat se může pouze málokdy. Naopak u malých dětí může být časté špatné vyslovení písmena R. Tyto dva jevy se u osob budou vyskytovat s jiným procentem a pravděpodobnost výskytu zde bude rozdílná. S tímto je výhodné pracovat i při návrhu neuronové sítě a z toho důvodu je použito cross-entropy trénování.

Při předpokladu hluboké neuronové sítě s neurony j a vstupy x_j, pak je skalární výstup y_j vypočten podle vztahu

$$y_j = logistic(x_j) = \frac{1}{1 + e^{-x_j}} \,. \tag{3.1}$$

Výpočet vstupu do daného uzle se pak vypočte dle vztahu

$$x_j = b_j + \sum_i y_i w_{ij} \,, \tag{3.2}$$

kde b_j je práh neuronu, i je index neuronu v předchozí vrstvě a w_{ij} je váha mezi neuronem j v aktuální vrstvě s neuronem i v předchozí vrstvě.

K výpočtu pravděpodobnosti třídy daného neuronu j se využívá nelinearity funkce. Funkce bere vektor reálných čísel a jako výstup dává vektor reálných čísel, které jsou v intervalu od 0 do 1. Pro výpočet pravděpodobnosti lze využít vztah

$$p_j = \frac{exp(x_j)}{\sum_k exp(x_k)} \,, \tag{3.3}$$

kde k je index přes všechny třídy.

Následně lze zjistit ztrátovou funkci (loss function) C pomocí křížové entropie mezi cílovou pravděpodobností[1] d a pravděpodobností získanou na bázi Softmax funkce p

$$C = -\sum_j d_j log p_j \,. \tag{3.4}$$

Natrénování sítě je výhodné dělat postupně. Tedy vzít určitou část trénovacích dat a natrénovat s nimi síť. Dále vzít další část, opět provést trénování a podle toho síť aktualizovat. Aktualizace vah se provádí pomocí vztahu

$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) - \epsilon \frac{\partial C}{\partial w_{ij}(t)} \,, \tag{3.5}$$

[1] Cílová pravděpodobnost je většinou volena jako 0 či 1. Závisí to na vstupních informacích, které slouží k trénování dané sítě.

kde α je momentum. Momentum je koeficient volený $0 < \alpha < 1$ a vyhlazuje změny při aktualizaci. Díky tomu je možné zlepšit tento proces.

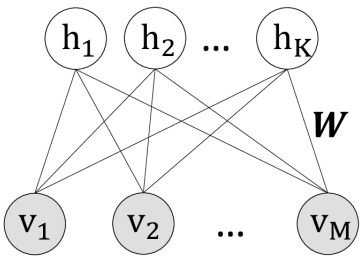
3.2.2 Sekvenční diskriminativní trénování

Cross-entropy trénování probíhá nezávisle na jednotlivých segmentech zvlášť. Nicméně na úlohu rozpoznávání řečníka je nutné nahlížet jako na sekvenční. Z toho důvodu se používají sekvenční diskriminativní techniky pro trénování, které jsou použity po cross-entropy trénování. Použitím těchto technik lze snížit chybovost sítě o 3 až 17 % [20]. Mezi nejznámější sekvenční diskriminativní techniky patří

- Maximum Mutual Information (MMI),
- Boosted MMI (BMMI),
- Minimum Phone Error (MPE),
- Minimum Bayes Risk (MBR).

3.2.3 RBM - Restricted Boltzmann Machine

RBM (Restricted Boltzmann Machine) je stochastická neuronová síť, která se často využívá při inicializaci neuronových sítí využívaných při úloze rozpoznávání řečníka. Je tvořena dvěma vrstvami neuronů - viditelnou a skrytou. Nedochází zde k žádnému spojení mezi neurony v jedné vrstvě. Pak tedy neuron ve viditelné vrstvě nebude nikdy spojen s jiným neuronem ve viditelné vrstvě. Naopak typicky je každý neuron z viditelné vrstvy spojen s každým ze skryté vrstvy, jak je vidět na obrázku 3.3. Tento typ architektury zajišťuje lehčí a rychlejší trénování. Zároveň je možné vytvářet rozsáhlejší a více komplexní sítě jako jsou Deep Belief Networks (DBN) nebo Deep Boltzmann Machines (DBM).



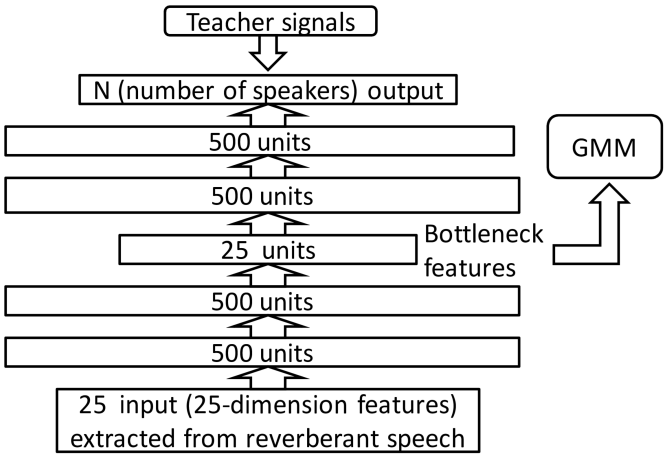
Obrázek 3.3: Příklad RBM sítě [18]

3.3 Metody rozpoznávání mluvčího na bázi DNN

Využití neuronových sítí pro rozpoznávání mluvčího je velmi široké a má uplatnění v různých částech implementace. Neuronová síť nemusí vždy nahradit klasické metody jako je GMM-UBM a podobně. Často jsou neuronové sítě použity pouze pro mezivýpočty v celém procesu. U pokročilejších implementací pak neuronová síť obstarává téměř celý proces rozpoznání mluvčího. Rozlišujeme pak mezi přímým a nepřímým použitím neuronových sítí v daných implementacích.

3.3.1 Bottleneck příznaky

Jedním z nepřímých použití neuronových sítí při implementaci rozpoznávání mlučího jsou Bottleneck příznaky. Kromě příznaků MFCC, o kterých pojednává kapitola 2.2, lze použít Bottleneck příznaky. Tyto příznaky jsou získávány z vícevrstvé sítě pomocí nelineární transformace. Použitím Bottleneck příznaků dochází až k 46,3 % redukci chyby oproti implementacím s využitím MFCC příznaků [10].



Obrázek 3.4: Extrakce Bottleneck příznaků z vrstvy neuronové sítě [10]

Pro výpočet Bottleneck příznaků se používá deep belief network (DBN), která se ukázala jako nejefektivnější pro použití Bottlenech příznaků. Při špatné volbě neuronové sítě může dojít i ke zvýšení chyby dané implementace. Proto je volba typu neuronové sítě důležitá. Bottleneck příznaky jsou extrahovány z jedné vnitřní vrstvy sítě pomocí vícevrstvého perceptronu neboli multi-layer perceptronu (MLP). Proces extrakce Bottleneck příznaků je znázorněn na obrázku 3.4. Vrstva, ze které jsou příznaky získány, má menší počet uzlů, než ostatní vrstvy. Díky tomu lze s touto vrstvou zacházet jako s

nízkorozměrnou nelineární funkcí vstupních příznaků. U MLP nedochází k žádnému trénovaní předem a je zde volena náhodná hodnota mezi -0,5 až 0,5. U DBN dojde nejdříve k nehlídanému trénování, ze kterého je získaná počáteční hodnota DBN. Po inicializaci dochází ke korigovanému diskriminační trénování. Následně je možné extrahovat Bottleneck příznaky a použít je pro trénování UBM-GMM modelu.

Kapitola 4

Implementace

Úloha je implementována pomocí nástrojů KALDI a skriptů, které jsou v nich dostupné. Úprava a tvorba nových skriptů byla psána převážně v programovacím jazyce Shell. KALDI toolkit je přizpůsoben především pro GNU/Linux, proto i celá implementace byla tvořena v tomto operačním systému.

4.1 KALDI nástroje

KALDI je toolkit pro rozpoznávání řeči napsaný v C++ a licencovaný pod Apache License v2.0. KALDI projekt začal v roce 2009, kdy se konal workshop na univerzitě Johns Hopkins. Někteří účastníci workshopu se dohodli na pokračování, kdy v roce 2010 hostovalo další ročník Vysoké učení technické v Brně. KALDI toolkit obsahuje implementaci většiny metod používaných při rozpoznávání řeči či řečníka a dochází k dalšímu vývoji a novým implementacím.

4.2 Implementace úlohy

Pro implementaci úlohy byly použity nástroje KALDI a skripty, připravené v rámci diplomové práce [21], která se zabývala rozpoznáváním řečníka na bázi GMM a i-vektorů. Z této práce byly převzaty především skripty (recepty), které zajišťovaly trénování modelů, extrakce i-vektorů, výpočet skóre a verifikaci. Nicméně se v dané implementaci pracovalo s jinými databázemi a příprava dat tedy musela být vytvořena nově.

Obrázek 4.1: Adresářová struktura složek a souborů

Na obrázku 4.1 lze vidět adresářovou strukturu pro daný projekt. Jako hlavní skript, ze kterého se spouštějí jednotlivé úseky (STAGES) úlohy, je zde *path.sh*. V tomto skriptu jsou nadefinovány základní proměnné a je vytvořena určitá struktura, podle které má daná úloha proběhnout. Z hlavního skriptu lze nastavit několik proměnných, který mají vliv na výslednou chybu dané implementace.

- **ENROLL_PART** - číselná hodnota, která znázorňuje procento mluvčích, které bude použito jako enroll data.
- **TRAIN_PART** - číselná hodnota, která znázorňuje procento mluvčích, které bude použito jako train data. Tato proměnná je vždy vypočtena jako zbylé procento oproti ENROLL_PART.
- **NUMUTT_TRAIN** - počet promluv jednotlivých mluvčích, které bude použito pro train data.
- **NUMUTT_ENROLL** - počet promluv jednotlivých mluvčích, které bude použito pro enroll data.
- **NUMUTT_TEST** - počet promluv jednotlivých mluvčích, které bude použito pro test data.

Hlavní složka dále obsahuje skripty *path.sh* a *cmd.sh*, kde jsou definovány cesty k databázím, ke KALDI nástroji a dalším frameworkům, které jsou použité pro správné fungování implementace.

Hlavní složka následně obsahuje podsložky *conf, local, steps* a *utils*. Ve složce conf jsou konfigurační soubory pro danou úlohu. Složka local obsahuje skripty vytvořené přímo pro danou implementaci s danou databází. Ve složce steps a utils lze najít skripty, které jsou dostupné v rámci distribuce KALDI nástrojů. Kromě těchto podsložek se po proběhnutí úlohy vytváří složky *data, mfcc, vad* a *exp*, ve kterých se ukládají průběžná data, příznaky MFCC a VAD a výsledné výstupy úlohy.

4.2.1 STAGE 0 - Příprava dat

První fáze se zabývá přípravou dat pro danou úlohu. Všechna data je potřeba připravit do nutného formátu pro pozdější zpracování. Dochází zde především k vytváření listů, ve kterých jsou vhodně kombinována data řečníků s jejich promluvami. K tomuto kroku byl vytvořen skript *spk_lists.sh* v podsložce *local*. Skript *spk_prep.sh* je spouštěn z hlavní skriptu *path.sh*, ze kterého jsou předávány kromě různých cest také důležité proměnné jako NUMUTT_TRAIN, NUMUTT_ENROLL, NUMUTT_TEST či ENROLL_PART.

Data lze tedy dělit na 3 skupiny:

- **train** - train data slouží k natrénování GMM a UBM modelu. Pro věrohodnou implementaci by tedy tato data neměla obsahovat data žádného z řečníků, které budou určeny pro enroll data.
- **enroll** - z enroll dat jsou vytvořeny GMM modely jednotlivých řečníků, které můžou být již součástí test dat, pokud chceme, aby byli v části verifikace rozpoznáni. V těchto datech tedy můžou být stejní mluvčí jako v test datech, nicméně by se měly lišit použité promluvy.
- **test** - test data jsou použita při verifikaci, kdy jsou srovnávána s enroll daty a při následném vyhodnocení, zda došlo k správnému rozpoznání řečníka. Tato množina by měla obsahovat řečníky z enroll dat, kteří by měli být při verifikaci rozpoznáni, a z množiny řečníků, kteří nejsou součástí enroll dat, aby při verifikaci nastala i chyba, kdy řečník není rozpoznán.

4.2.2 STAGE 1 - Výpočet příznaků

V této fázi dochází k výpočtu příznaků MFCC a VAD. Pro MFCC jsou získány základní koeficienty, které se pak používají v dalších fázích. Některé procesy vyžadují derivace těchto koeficientů (delta a delta delta), které jsou počítány vždy před danou fází.

Pro výpočet MFCC a VAD příznaků jsou použity recepty, které jsou součástí distribuce KALDI a nebylo tedy třeba tyto výpočty implementovat. Nastavení pro výpočet MFCC koeficientů lze modifikovat ze souboru *mfcc.conf* a jako výchozí nastavení jsou použity následující hodnoty:

- number of cepstrums: 20
- high cutoff frequency: -200 Hz
- sampling frequency: 16 kHz

VAD využívá logaritmus založený na výpočtu energii řečového segmentu. Energie je vždy počítána pro každý segment v průběhu výpočtu MFCC. Nastavení pro výpočet příznaků VAD lze modifikovat pomocí souboru *vad.conf*, kdy jeho výchozí nastavení je:

- energy threshold: 5,5
- energy mean scale: 0,5

4.2.3 STAGE 2 - Trénování UBM a i-vektor extraktoru

Výsledek trénování UBM lze ovlivnit z hlavního skriptu run.sh pomocí parametrů GMM_COMPONENTS (počet GMM komponent použitých při trénování UBM modelu) a IVECTOR_DIMENSION (dimenze i-vektorů). Proces trénování UBM a i-vektor extraktoru je proveden ve třech krocích:

- Vytvoření UBM s diagonální kovarianční maticí.
- Přetrénování na UBM s plnou kovarianční maticí.
- Natrénování extraktoru i-vektorů.

V tomto kroku jsou vytvořeny ve složce *exp/CZ* složky *extractor* a *full_ubm*, ve kterých jsou uloženy výsledné soubory. K natrénování UBM a i-vektor extraktoru jsou v KALDI dostupné již vytvořené recepty - train_diag_ubm.sh, train_full_ubm.sh a train_ivector_extractor.sh. Proto tato fáze úlohy nebude více rozepisována.

4.2.4 STAGE 3 - Extrakce i-vektorů

V tomto kroku jsou z MFCC koeficientů pomocí UBM vytvořeny statistiky (posteriors). Z těchto statistik se následně extrahují i-vektory pro jednotlivá train, enroll a test data. V tomto kroku se vytváří složka *ivectors* v podadresáři *exp/CZ*, kde jsou uloženy výsledné soubory. K tomuto procesu je využit skript *extract_ivectors.sh*, který je opět již implementován v nástrojích KALDI a není tedy třeba ho podrobněji popisovat.

4.2.5 STAGE 4 - Výpočet skóre

Výpočet skóre je možný třemi následujícími metodami:

- **Výpočet kosinové vzdálenosti** - pro výpočet skóre touto metodou je použit skript *cosine_scoring.sh* ve složce *local*. Zde dochází k zjištění podobnosti jednotlivých mluvčích pomocí i-vektorů napočítaných v předchozím kroku. Výpočtem skalárního součinu lze dostat hodnotu, která znázorňuje podobnost dvou daných mluvčích. Výběr dvojice mluvčích se provádí na základě listu *trials* vytvořeného při přípravě dat.
- **Výpočet kosinové vzdálenosti s pomocí LDA transformace** - u této metody dochází nejprve k výpočtu transformační matice. K tomu se využívá i-vektorů a listu *spk2utt* napočítaných pro train data. Po zjištění transformační matice lze transformovat i enroll a test data. Následně je postup stejný jako u výpočtu kosinové vzdálenosti pomocí skalárního součinu. Chybovost této metody lze ovlivnit vstupními proměnnými - dimenzí po transformaci a kovariančním faktorem.
- **Výpočet s použitím PLDA modelu** - tato metoda je obdobná LDA transformaci, ale místo transformační matice se zde natrénuje PLDA model na train datech a následně se použije pro enroll a test data.

V této části zároveň dochází k identifikaci mluvčích. Pro každého řečníka z test množiny je nalezena dvojice (řečník z množiny enroll), mezi kterými byla hodnota skóre největší. Tato dvojce je pak identifikována jako stejný mluvčí. Následně je vypočítána i chyba identifikace, kdy se podle ID daných řečníků ověřuje, zda bylo nejvyšší skóre skutečně mezi totožnou dvojicí či nikoli. Výpočtu této chyby je věnována kapitola 2.8.

4.2.6 STAGE 5 - Verifikace

Implementace verifikace je podobná jako u výpočtu chyby identifikace v předchozí fázi. Hlavním faktorem je zde volený práh. Tento práh udává informaci, od jaké hodnoty skóre by měla být daná dvojice považována za totožnou osobu. V ideálním případě by tedy všechny dvojice s hodnotou skóre nad daným prahem měly být totožné osoby a naopak všechny pod prahem rozdílné osoby. Pokud tomu tak není, došlo k chybné verifikaci.

Zde je vidět příklad výstupu, kde je zvolena hodnota prahu 80.

72,12 target - reject ERR

75,2 nontarget - reject OK

78,822 target - reject ERR

79,879 nontarget - reject OK

———————————————————————- verifikační práh 80

81,8 nontarget - accept ERR

88,7 nontarget - accept ERR

92,564 target - accept OK

93,56 target - accept OK

Posouváním toho prahu lze tedy najít určité optimální nastavení. Následně je vypočtena chybová míra EER, FAR a FRR, kterým je věnována kapitola 2.7.

Kapitola 5

Experimentální část

Experimentální část se zabývá implementací identifikace mluvčího na bázi GMM a i-vektorů pomocí MFCC a VAD příznaků. Pro tuto část byla použita jiná databáze mluvčích, než byla použita v [21], a tedy muselo dojít k změně přípravy dat.

Implementace byla upravena především v rámci prvních fází, které bylo třeba přizpůsobit databázi GLOBALPHONE. Dále byly upraveny recepty tak, aby byla větší možnost volby parametrů. Proces lze spouštět z příkazového řádku bez větší nutnosti zasahovat do jednotlivých receptů. Z příkazového řádku se spouští skript *prerun.sh*, a lze mu přidělit 9 parametrů:

1. volba jednotlivé fáze (STAGE) - lze spustit jednotlivé STAGE (0-5) nebo celý proces najednou (all)
2. volba jazyka - lze spustit jednotlivé jazyky zvlášť, případně více nebo všechny najednou. Dále je možnost spustit testování, kdy se použijí všichni řečníci a promluvy ze všech dostupných jazyků
3. volba procenta řečníků pro enroll data
4. volba počtu promluv použitých pro train data
5. volba počtu promluv použitých pro enroll data
6. volba počtu promluv použitých pro test data
7. threshold použitý pro metodu kosinová vzdálenost
8. threshold použitý pro metodu LDA
9. threshold použitý pro metodu PLDA

Ve složkách, do kterých jsou ukládany data a mezivýpočty (složky *data*, *mfcc*, *vad* a *exp*) jsou organizovány tak, že jsou zde vždy vytvořeny složky

podle jednotlivých nastavení parametrů ve formátu *LANG*_perc-*train_part*-*enroll_part*_utts-*numutt_train*-*numutt_enroll*-*numutt_test*, kde

- *LANG* je zkratka jazyka, pro který je test spuštěn
- *train_part* je procento mluvčích pro enroll množinu
- *enroll_part* je procento mluvčích pro train množinu
- *numutt_train* je počet promluv jednotlivých mluvčích použit v train množině
- *numutt_enroll* je počet promluv jednotlivých mluvčích použit v enroll množině
- *numutt_test* je počet promluv jednotlivých mluvčích použit v test množině

Složka pro jazyk CZ, kde je rozložení mluvčích mezi enroll a train daty 50:50, pro train množinu je použito 50 promluv od každého mluvčího a pro enroll a test je použita jedna promluva od každého mluvčího, pak bude mít jméno *CZ_perc-50-50_utts-50-1-1*.

Tato složka dále obsahuje složky jednotlivých množin train, enroll a test. Navíc je zde složka *local*, do které jsou generovány důležité listy, které dávají informace o rozložení řečníků a prolmuv do jednotlivých množin train, enroll a test.

5.1 Použité databáze

Pro testování byla použita databáze GLOBLAPHONE verze 3.5 [19]. Tato databáze obsahuje 20 světových jazyků včetně češtiny. Promluvy jednotlivých řečníků byly vytvořeny vždy v zemi, ve které je daný jazyk jako oficiální. V každé z těchto zemí bylo vybráno zhruba 100 rodilých mluvčích, aby přečetli přibližně 100 vět. Věty byly vybrány z různých světových novin v rozmezí 1995-2009 a u kterých slovní zásoba dosahuje 65000 slov. Všichni řečníci jsou dospělé osoby zahrnující mužské i ženské pohlaví.

5.2 Výchozí nastavení

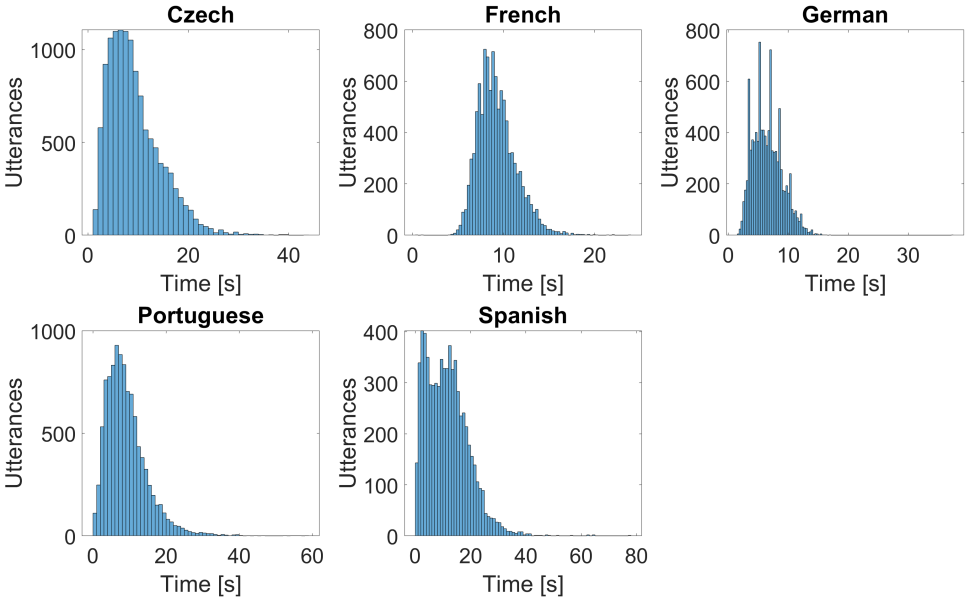
Hlavní skript *run.sh* obsahuje několik proměnných, kterými lze ovlivňovat výchozí nastavení implementace a výsledky. Tyto proměnné byly zmíněné v kapitole 4.2.

Testování bylo provedeno pro pět různých světových jazyků, které databáze GLOBALPHONE obsahuje.

- Čeština - obsahující 102 mluvčích a 12425 promluv.
- Němčina - obsahující 77 mluvčích a 10085 promluv.
- Francouzština - obsahující 100 mluvčích a 10478 promluv.
- Portugalština - obsahující 101 mluvčích a 10344 promluv.
- Španělština - obsahující 100 mluvčích a 6898 promluv.

Promluvy jednotlivých jazyků byly vytvořeny v podobných podmínkách a vždy rodilými osobami. Audio nahrávky jsou vždy vzorkovány frekvencí 16 kHz. Délky promluv jsou velmi rozdílné především v rámci daného jazyka.

V této práci dochází především k testování vlivu počtu mluvčích a počtu promluv dané databáze. Velký vliv zde ovšem hraje také délka jednotlivých promluv. Pro identifikaci mluvčího se často využívá pouze krátká několika sekundová promluva. Databáze GLOBALPHONE nicméně obsahuje promluvy velmi různých délek od krátkých promluv až po delší věty. Tento fakt není zahrnut v implementaci této práce, a proto může dojít ke zkreslení výsledků, kdy jsou použity promluvy různých délek a chybné rozpoznání může mít větší procento při použití dvou krátkých promluv oproti jedné delší.



Obrázek 5.1: Délky promluv jednotlivých jazyků

Na obrázku 5.1 lze vidět rozložení délek promluv pro jednotlivé jazyky. Z grafů lze vyčíst, že největší počet promluv je pro všechny jazyky mezi 5-10

sekundami. Španělština má téměř stejné procento i u promluv delších než 10 sekund. Dalo by se tedy předpokládat, že se výsledky pro španělský jazyk budou lišit od ostatních, jelikož má mírně odlišné rozložení délek promluv a navíc je pro tento jazyk dostupný nejmenší počet promluv (6898).

Tabulka 5.1 obsahuje nejkratší a nejdelší promluvu pro daný jazyk. Přestože z obrázku 5.1 je zřejmé, že těchto nahrávek není mnoho, mohou mít vliv na výsledky jednotlivých testování. Pokud by se jako enroll promluva zvolila právě ta nejkratší možná, bude výsledná chyba identifikace a verifikace velmi vysoká a zavádějící. Při vybrání nejdelší promluvy by byla situace opačná a chyba by byla naopak velmi malá.

	Minimální délka [s]	Maximální délka [s]
Čeština	1,1263	43,3725
Francouzština	0,5134	23,8734
Němčina	1,5354	37,3229
Portugalština	0,0014	58,7514
Španělština	0,2614	77,3114

Tabulka 5.1: Minimální a maximální délka jedné promluvy pro jednotlivé jazyky

Databáze GLOBALPHONE obsahuje různý počet mluvčích ženského a mužského pohlaví. Rozložení ženských a mužských řečníků se liší dle daného jazyka. Skript je tedy přizpůsoben tak, aby v množinách train a enroll bylo rozložení mužských a ženských mluvčích zhruba stejné a nedošlo tím ke zkreslení výsledků.

Výchozí nastavení některých parametrů bylo převzato z diplomové práce [21] a jejich hodnoty lze vidět v tabulce 5.2.

Počet GMM komponent	64
Dimenze i-vektorů	400
LDA zmenšení dimenze	250
LDA kovariační faktor	0,05

Tabulka 5.2: Výchozí nastavení parametrů

Optimalizací těchto parametrů se věnovala práce [21] a z toho důvodu se tato implementace nebude dále těmito parametry více zabývat. Větší důraz se bude klást na rozložení mluvčích a daných promluv.

5.3 Vliv rozložení promluv rámci množin train, enroll a test

Pro testování vlivu rozložení promluv jsou důležité tři proměnné

- PATTERN_TRAIN - počet promluv od každého řečníka použitých v množině train,
- PATTERN_ENROLL - počet promluv od každého řečníka použitých v množině enroll,
- PATTERN_TEST - počet promluv od každého řečníka použitých v množině test.

Rozložení promluv je vidět v tabulce 5.3. V tabulkách s výsledky je vždy rozložení prolmuv uvedeno ve sloupci **Rozložení** ve formátu *enroll-test*.

train	enroll	test
50	1	1
50	2	2
50	3	3
50	5	5

Tabulka 5.3: Rozložení promluv

Testování bylo provedeno vždy s rozložením mluvčích 50-50 *train-enroll*.

Výsledné hodnoty pro rozložení promluv

V tabulkách 5.4, 5.5 a 5.6 lze vidět výsledné hodnoty pro jednotlivé metody kosinová vzdálenost, LDA a PLDA pro rozložení promluv pro jednotlivé jazyky. Z výsledných hodnot lze říci, že zde není závislost na jazyku. Výrazněji horší výsledky jsou pouze pro španělský jazyk, kde chyba identifikace pro rozložení promluv 1-1 dosahuje 16 %. To se dá vysvětlit značně menším počtem řečníků a promluv, které jsou pro tento jazyk dostupné.

Sloupec **Identify error** je procentuální chyba při identifikaci mluvčího. Zde bylo předpokládáno, že při rostoucím počtu promluv použitých pro enroll a test data bude chyba klesat. Z hodnot lze vidět, že již při použití dvou promluv chyba identifikace prudce klesá. Je zajímavé, že při použití pěti promluv pro enroll a test data je chyba identifikace stejná nebo větší než při použití tří promluv. To by se dalo vysvětlit již zmíněnou rozdílností délek promluv, jelikož při testování ostatních jazyků tento jev nenastal.

EER vyjadřuje chybu při verifikaci jednotlivých mluvčích. Trend by zde měl být podobný jako při chybě identifikace. Velkou roli zde však hraje verifikační práh neboli threshold. Hodnota thresholdu se zvětšuje s rostoucím počtem použitých promluv. To je dané tím, že jsou jednotliví řečníci lépe charakterizováni a skóre mezi stejnými mluvčími je vyšší. Scoring EER, FRR a FAR je vždy napočítané pro uvedený threshold, který je vypočítán funkcí compute-err, která je dostupná v rámci distribuce KALDI. Tato funkce bere napočítané skóre pro jednotlivé metody a vypočte vhodný verifikační práh. FRR a FAR chyba by se pak dala modifikovat pomocí posunu verifikačního prahu.

Rozložení	$P_{id_{err}}$ [%]	**EER** [%]	**Threshold**	**FRR** [%]	**FAR** [%]
Čeština					
50-50-1-1	10	4	56,6	4	3,34
50-50-2-2	0	2	113,23	2	0,11
50-50-3-3	0	2	172,67	2	0
50-50-5-5	2	2	176,18	2	0,03
Francouzština					
50-50-1-1	2,04	2,041	81,9	4,08	0,24
50-50-2-2	0	2,041	141,34	4,08	0
50-50-3-3	0	2,041	192,15	2,04	0
50-50-5-5	0	0	207,9	0	0
Němčina					
50-50-1-1	7,89	5,26	49,82	7,89	3,84
50-50-2-2	0	2,632	110,85	5,26	0,03
50-50-3-3	0	0	139,23	2,63	0
50-50-5-5	0	0	205,56	0	0
Portugalština					
50-50-1-1	6	4	84,06	6	2,76
50-50-2-2	0	2	112,13	4	1,36
50-50-3-3	0	2	117,76	4	1,66
50-50-5-5	0	2	175,402	2	0,28
Španělština					
50-50-1-1	16	6	45,83	8	4,72
50-50-2-2	2	2	104,15	4	0,2
50-50-3-3	2	2	132,11	4	0,2
50-50-5-5	2	2	106,815	4	0,9
Multilanguage					
50-50-1-1	13,92	3,165	47,84	3,37	3,14
50-50-2-2	0,84	0,422	99,44	0,42	0,12
50-50-3-3	0,42	0,42	116,92	0,42	0,07
50-50-5-5	1,26	0,42	105,28	0,84	0,33

Tabulka 5.4: Výsledné hodnoty porovnání rozložení promluv pro metodu kosinová vzdálenost

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
1-1	10	4	62,8	6	1,96
2-2	0	2	88,95	4	0,73
3-3	0	2	122,572	4	0,07
5-5	2	2	136,92	2	0,03
Francouzština					
1-1	2	2,04	80,7	4,08	0,32
2-2	0	2,041	123,42	4,08	0
3-3	0	0	140,51	2,04	0
5-5	0	0	164,64	0	0
Němčina					
1-1	7,89	2,63	63,97	2,63	1,35
2-2	0	2,632	100,95	2,63	0,41
3-3	0	2,632	135,09	5,26	0,3
5-5	0	0	150,43	2,63	0
Portugalština					
1-1	6	4	53,35	6	3,58
2-2	0	4	106,92	4	0,08
3-3	0	2	87,14	4	1,18
5-5	0	2	128,98	2	0,04
Španělština					
1-1	16	8	45,71	8	3,41
2-2	2	2	72,91	4	0,9
3-3	2	2	89,38	2	0,42
5-5	2	2	111,18	4	0,14
Multilanguage					
1-1	13,92	2,321	45,85	2,1	2,21
2-2	0,84	0,844	89,78	0,84	0,13
3-3	0,42	0,42	96,67	0,42	0,07
5-5	1,26	0,42	100,03	0,42	0,26

Tabulka 5.5: Výsledné hodnoty porovnání rozložení promluv pro metodu LDA

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
1-1	10	6	-34,47	8	3,64
2-2	0	2	-28,87	2	0,55
3-3	0	2	-45,39	4	0,67
5-5	2	2	-79,12	4	1,42
Francouzština					
1-1	2	2,04	-0,02	4,08	0,08
2-2	0	0	1,88	0	0
3-3	0	0	-11,98	0	0
5-5	0	2,041	-50,69	2,04	0,43
Němčina					
1-1	7,89	4	-8,26	5,26	0,13
2-2	0	2,632	-11,95	5,26	0,03
3-3	0	2,632	-41,71	2,63	0,38
5-5	0	5,263	-74,03	5,26	2,07
Portugalština					
1-1	6	6	-3,31	6	3,44
2-2	0	2	-5,21	2	1,98
3-3	0	2	-0,65	4	0,6
5-5	0	2	-4,93	2	1,22
Španělština					
1-1	16	12	-31,74	14	10,04
2-2	2	2	-18,73	2	1,39
3-3	2	2	-20,31	2	0,68
5-5	2	4	-42,335	4	3,25
Multilanguage					
1-1	13,92	3,059	-24,12	3,37	3,02
2-2	0,84	1,266	-18,63	1,26	0,31
3-3	0,42	0,42	-25,11	0,42	0,21
5-5	1,26	0,63	-52,05	0,84	0,54

Tabulka 5.6: Výsledné hodnoty porovnání rozložení promluv pro metodu PLDA

5.4 Vliv rozložení promluv rámci množin train, enroll a test

Pro testování vlivu rozložení mluvčích jsou důležité dvě proměnné

- TRAIN_PART - procento mluvčích použitých pro train množinu,
- ENTOLL_PART - procento mluvčích použitých pro enroll množinu.

Množina test dat se skládá vždy ze všech dostupných řečníků pro daný jazyk, které obsahuje databáze GLOBALPHONE. Překrývá se tedy jak s množinou train tak s množinou enroll a je nutné, aby byly brány rozdílné promluvy řečníků, u kterých dochází k překryvu. Jinak by mohlo dojít ke zkreslení výsledných hodnot. Množiny enroll a train se nesmí překrývat u žádných z řečníků.

Testování bylo provedeno pro čtyři rozložení mluvčích v množinách train a enroll

1. 30 % mluvčích v train datech a 70 % mluvčích v enroll datech,
2. 50 % mluvčích v train datech a 50 % mluvčích v enroll datech,
3. 70 % mluvčích v train datech a 30 % mluvčích v enroll datech,
4. 90 % mluvčích v train datech a 10 % mluvčích v enroll datech,

kdy bylo vždy použity promluvy ve formátu 50-1-1 *train-enroll-test*. Rozložení množin je ve výsledných tabulkách uvedeno ve sloupci **Rozložení** ve formátu *train-enroll*.

Výsledné hodnoty pro rozložení mluvčích

V tabulkách 5.7, 5.8 a 5.9 lze vidět výsledné hodnoty pro jednotlivé metody kosinová vzdálenost, LDA a PLDA pro rozložení řečníků pro český jazyky a multilanguage. Pro český jazyk je dostupných 102 řečníků, což není mnoho. Proto z výsledků nelze jasně říci, které rozložení je výhodnější. Pokud v množině enroll bude pouze malý počet řečníků, výsledná chyba identifikace a verifikace bude velmi záležet už na jedné chybě identifikace či verifikace řečníka. Z výsledků pro multilanguage lze říci, že s rostoucím počtem řečníků v množině train se chyba identifikace i verifikace zmenšuje.

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
30-70	15,71	7,143	60,41	7,14	5,79
50-50	10	4	56,6	4	3,34
70-30	16,66	6,667	41,52	10	5,67
90-10	0	11,11	63,41	11,11	0,55
Multilanguage					
30-70	16,01	3,248	50,41	3,32	3,19
50-50	13,92	3,165	47,84	3,37	3,14
70-30	11,42	2,143	53,78	2,85	1,71
90-10	2,32	2,326	58,82	4,65	1

Tabulka 5.7: Výsledné hodnoty porovnání rozložení řečníků pro metodu kosinová vzdálenost

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
30-70	15,71	2,857	72,11	4,28	1,95
50-50	10	4	62,8	6	1,96
70-30	16,66	6,667	56,99	10	1,38
90-10	0	11,11	65,92	11,11	0,33
Multilanguage					
30-70	16,01	1,662	52,86	1,81	1,61
50-50	13,92	2,321	45,85	2,1	2,21
70-30	11,42	2,143	54,08	2,14	0,75
90-10	2,32	2,326	44,85	4,65	1,67

Tabulka 5.8: Výsledné hodnoty porovnání rozložení řečníků pro metodu LDA

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
30-70	15,71	4,286	-28,23	4,28	2,58
50-50	10	6	-34,47	8	3,64
70-30	16,66	3,33	-20,51	3,33	1,12
90-10	0	11,11	-20,51	3,33	1,12
Multilanguage					
30-70	16,01	3,021	-22,87	3,02	2,69
50-50	13,92	3,059	-24,12	3,37	3,02
70-30	11,42	2,857	-15,88	2,85	1,33
90-10	2,32	2,326	-11,81	4,65	0,86

Tabulka 5.9: Výsledné hodnoty porovnání rozložení řečníků pro metodu PLDA

5.5 Vliv vzorkovací frekvence u použitých dat

Výchozí vzorkovací frekvence audio nahrávek je 16 kHz. V této části testování došlo k převzorkování signálu z původních 16 kHz na 8 kHz. Převzorkováním na nižší kmitočet dochází ke ztrátě informací a dá se očekávat mírně zhoršené výsledky identifikace a verifikace oproti původním 16 kHz.

Testování bylo provedeno pro stejné rozložení promluv a mluvčích jako při testování vlivu rozložení promluv. Výsledné hodnoty jsou vypočítány pro český jazyka a multilanguage.

Výsledné hodnoty pro rozložení promluv se vzorkováním signálu 8 kHz

V tabulkách 5.10, 5.11 a 5.12 lze vidět výsledné hodnoty pro jednotlivé metody kosinová vzdálenost, LDA a PLDA pro rozložení promluv, kdy byl použit vzorkovací kmitočet 8 kHz. Při porovnání s měřením, kdy byly použity data se vzorkovacím kmitočtem 16 kHz, jsou výsledné chyby téměř stejné a rozdíly zanedbatelné. U některých hodnot jsou výsledné chyby identifikace a verifikace dokonce menší. Tyto odchylky jsou pravděpodobně způsobeny náhodným výběrem daných promluv a řečníků v jednotlivých množinách. Podvzorkování kmitočtem 8 kHz tedy nemělo velký vliv ani na chybu identifikace ani na chybu verifikace.

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
1-1	14	4	56,29	4	3,3
2-2	0	1	96,47	2	0,53
3-3	0	1	145,7	2	0,07
5-5	2	0,667	166,97	0	0,3
Multilanguage					
1-1	10,97	2,004	53,51	2,1	1,93
2-2	0,84	0,42	88,67	0,42	0,17
3-3	0,42	0,31	93,53	0,42	0,22
5-5	0,84	0,31	99,21	0	0,3

Tabulka 5.10: Výsledné hodnoty porovnání rozložení promluv pro metodu kosinová vzdálenost

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
1-1	14	4	61,96	6	1,6
2-2	0	2	111,73	4	0,05
3-3	0	1	124,38	0	0,03
5-5	2	0,667	119,29	1,33	0,19
Multilanguage					
1-1	10,97	1,68	53,83	2,1	0,97
2-2	0,84	0,42	95,22	0,84	0,16
3-3	0,42	0,42	96,11	0,84	0,13
5-5	0,84	0,42	95,35	0,84	0,29

Tabulka 5.11: Výsledné hodnoty porovnání rozložení promluv pro metodu LDA

Rozložení	$P_{id_{err}}$ [%]	EER [%]	Threshold	FRR [%]	FAR [%]
Čeština					
1-1	14	6	-42,59	6	4,77
2-2	0	2	-43,64	2	1,74
3-3	0	2	-45,31	2	0,69
5-5	2	4	-76,89	6	1,34
Multilanguage					
1-1	10,97	3,979	-23,93	4,21	3,1
2-2	0,84	0,94	-29,66	0,84	0,85
3-3	0,42	0,633	-36,91	0,84	0,59
5-5	0,84	0,84	-44,48	0,84	0,36

Tabulka 5.12: Výsledné hodnoty porovnání rozložení promluv pro metodu PLDA

5.6 Přesnost identifikace mluvčího s použitím DNN s bottleneck vrstvou

Z časových důvodů nedošlo k vlastní implementaci systému rozpoznávání řečníka pomocí neuronových sítí. Po dohodě s vedoucím práce byla nakonec upřednostněna ucelenější analýza implementace standardního UBM-GMM systému na bázi i-vektorů, neboť tento systém je základem pro nepřímé použití neuronových sítí pro úlohu rozpoznávání řečníka. V této části je však alespoň popsána implementace a dosahované výsledky jinými autory, např. [10], kde je využito právě nepřímé použití neuronových sítí a to pro výpočet Bottleneck příznaků.

sampling frequency	16 kHz
frame length	25 ms
frame shift	10 ms
feature space	25 dimensions with CMN (12 MFCCs + Δ + Δ power)
acoustic model	GMMs with 128 diagonal covariance matrices

Obrázek 5.2: Nastavení výchozích parametrů [10]

V [10] byly použity MFCC příznaky jako vstupní parametry DNN sítě, ze které byly následně extrahovány Bottleneck příznaky. Výchozí parametry testování lze vidět v tabule 5.2. Jako vstup neuronové sítě byla použita jedna část z 25 dimenzionálních MFCC příznaků. DNN síť byla tvořena 25 skrytými neurony v Bottleneck vrstvě a 500 neurony v ostatních skrytých vrstvách. Celkově se DNN skládala z 5 vrstev. Pro trénování sítě byl využit stochastický gradient, kdy vstupem do sítě bylo 100 vzorků. Trénování sítě bylo spuštěno 50 fází s hodnotou učení 0,1 pro všechny vrstvy. Pro doladění a zpřesnění sítě bylo následně spuštěno 1000 cyklů opět s hodnotou učení 0,1 pro všechny vrstvy.

impulse response no	room	RT60 (s)
(a) CENSREC-4 database for training		
1	Japanese style room	0.40
2	Japanese style bath	0.60
3	elevator hall	0.75
(b) RWCP database for test		
4	tatami-floored room	0.47
5	echo room (panel)	1.30

Obrázek 5.3: Nahrávací podmínky impulsové odezvy [10]

Při testování implementace byl kladen důraz na zjištění vlivu vzdálenosti mluvčího od snímacího zařízení (mikrofonu). Velký vliv zde tedy měla odezva dané místnosti. Podmínky jednotlivých místností při nahrávání lze vidět v tabulce 5.3. Bylo použito pět druhů vícekanálové impulsové odezvy z databáze Real World Computing Partnership (RWCP) pro testovací množinu a z databáze CENSREC-4 pro trénovací množinu, kde byla data konvolucí spojena s čistým audiem, aby došlo k umělému vytvoření odezvy místnosti. Dále byla použita rozsáhlá databáze Japanese Newspaper Article Sentence (JNAS) obsahující čisté audio promluvy řeči (bez odezvy místností). Pro trénovací množinu bylo použito 50 řečníků a 10 promluv od každého mluvčího. Testovací množina byla tvořena vždy 20 promluvami od každého řečníka. Průměrná délka trénovacích promluv byla stanovena na 3,9 sekund. Pro testovací data byla průměrná délka promluvy 5,63 sekund.

Test proběhl pro tři různé výpočty příznaků - pomocí MFCC, pomocí Bottleneck příznaků extrahovaných na základě MFCC bez předem trénované sítě (BF-MLP) a pomocí Bottleneck příznaků extrahovaných na základě MFCC s předem natrénovanou sítí (BF-DNN). Výsledné hodnoty lze vidět v tabulce 5.4.

Features	RT60 for training (s)			Ave.
	0.40	0.60	0.75	
(a) RT 60 for test = 0.47 s				
MFCC	90.4	76.5	87.3	84.7
BF-MLP	83.2	91.6	92.6	89.1
BF-DNNs	92.9	91.1	92.8	92.3
(b) RT 60 for test = 1.30 s				
MFCC	89.3	81.6	85.1	85.3
BF-MLP	78.2	91.3	94.4	88.0
BF-DNNs	90.9	90.6	93.3	91.6

Obrázek 5.4: Výsledné hodnoty pro testování MFCC a Bottleneck příznaků pro jednotlivé hodnoty odezev zvuku v místnosti [10]

Implementace pomocí BF-DNN má lepší výsledky ve všech možný uvedených místnostech oproti MFCC. Při srovnání BF-DNN a BF-MLP, BF-DNN má při zprůměrování všech hodnot lepší výsledky, nicméně BF-MLP má vyšší přesnost pro nastavení 0.60. Při trénování sítě navíc nebyly použity data z různých prostředí s různým dozvukem. Při jejich použití by mohlo dojít k dalšímu zlepšení výsledků implementace s využitím Bottleneck příznaků.

Kapitola 6

Závěr

Práce se zabývá problematikou identifikace mluvčího a popisuje vybrané metody využívané pro danou úlohu. První část je věnována standardním metodám založeným na statistickém modelování. Jedná se především o systém na bázi GMM resp. i-vektorů. Popsán je proces identifikace řečníka od přípravy dat, přes výpočet příznaků, až po vytváření modelů pro jednotlivé řečníky a výslednou identifikace, resp. verifikaci. Druhá část je věnována principům a typickému použití neuronových sítí pro úlohu rozpoznávání řečníka.

Pro implementaci úlohy byly využity nástroje KALDI, které jsou přímo určeny pro úlohy rozpoznávání řeči a rozpoznávání řečníka. Experimentální část obsahuje testování implementace pomocí metody GMM a i-vektorů pro databázi GLOBALPHONE. Z časových důvodů nedošlo k vlastní implementaci systému rozpoznávání řečníka pomocí neuronových sítí. Po dohodě s vedoucím práce byla upřednostněna ucelenější analýza implementace standardního UBM-GMM systému na bázi i-vektorů.

V rámci testování je práce zaměřena především na zjištění vlivu počtu a rozložení mluvčích a promluv v jednotlivých množinách, tedy v množině trénovacích, referenčních a testovacích dat. Výsledky byly vytvořeny pro pět jazyků, které byly pro databázi GLOBALPHONE dostupné. Výsledné hodnoty odpovídají předpokladu, že s rostoucím počtem použitých promluv pro referenční a testovací množinu, je výsledná chyba identifikace a verifikace menší. Zároveň je diskutován možný vliv délky jednotlivých promluv na výslednou chybu identifikace a verifikace. Díky těmto informacím se daly očekávat rozdílné chyby především u španělského jazyku, což se potvrdilo ve výsledném testování, kdy chyba identifikace při použití jedné promluvy pro enroll a test data byla 16 %. Vliv rozložení mluvčích šlo pozorovat především pro multilanguage, kdy byli využiti všichni řečníci a promluvy všech dostupných jazyků. Z výsledků bylo patrné, že s rostoucím počtem řečníků v množině train klesá chyba identifikace, resp. verifikace. Další testování ukázalo, že podvzorkování nahrávek z původních 16 kHz na 8 kHz nemá na výsledné chyby větší vliv.

Literatura

[1] Achraf Benba al et. Detecting patients with parkinson's disease using mel frequency cepstral coefficients and support vector machines. Červenec 2015.

[2] Luis Bermudez. Overview of neural networks. *https://medium.com/machinevision/overview-of-neural-networks-b86ce02ea3d1*, Listopad 2019.

[3] Edwin Chen. Introduction to restricted boltzmann machines. *http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines/*, 2011.

[4] Debarko De. Rnn or recurrent neural network for noobs. *https://hackernoon.com/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860*, Červen 2018.

[5] F.Richardson, D. Reynolds, N. Dehak. Analysis of dnn approaches to speaker identification. 2016.

[6] Luke Dormehl. What is an artificial neural network? here's everything you need to know. *https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/*, Leden 2019.

[7] Pavel Durčák. Neuronové sítě a princip jejich fungování. *https://www.napocitaci.cz/33/neuronove-site-a-princip-jejich-fungovani-uniqueidgOkE4NvrWuNY54vrLeM670eFNQh552VdDDulZX7UDBY/*, Září 2017.

[8] D. Povey et al. The kaldi speech recognition toolkit. 2011.

[9] P. Matějka et al. Deep neural network approaches to speaker and language recognition. 2015.

[10] Takanori Yamada, Longbiao Wang, Atsuhiko Kai. Improvement of distant-talking speaker identification using bottleneck features of

dnn. *https://hackernoon.com/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860*.

[11] Mojmír Lakosil. Detektor řečové aktivity na bázi dnn. Diplomová práce, České vysoké učení technické v Praze, Listopad 2017.

[12] Michael A. Nielsen. Neural networks and deep learning. 2015.

[13] Josef Psutka, Luděk Müller, Jindřich Matoušek, Vlasta Radová. *Mluvíme s počítačem česky*. ACADEMIA, Praha, 2006.

[14] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition.

[15] Lawrence R. Rabiner, Roland W. Schafer. *Introduction to Digital Speech Processing*. Foundations and Trends in Signal Processing Volume 1 Issue 1-2, 2007.

[16] Jan Silovský. Generativní a diskriminativní klasifikátory v úlohách textově nezávislého rozpoznávání a diarizace mluvčích. Autoreferát disertační práce, Technická univerzita v Liberci, Červen 2011.

[17] Steven Volkaert. Determining the accuracy of a biometric system. Září 2014.

[18] Hung Thanh Vu. Energy-based models for video anomaly detection. Srpen 2017.

[19] GmbH Co. KG XLingual. Globalphone: a multilingual text speech database. Srpen 2012.

[20] Dong Yu and li Deng. *Deep Neural Network Sequence-Discriminative Training*, pages 137–153. 11 2015.

[21] Michael Záruba. Moderní metody rozpoznávání mluvčího na bázi gmm a dnn. Diplomová práce, České vysoké učení technické v Praze, Leden 2017.

[22] Jan Uhlíř, Pavel Sovka, Petr Pollák, Václav Hanžl, Roman Čmejla. *Technologie hlasových komunikací*. Nakladatelství ČVUT, Praha, 2017.