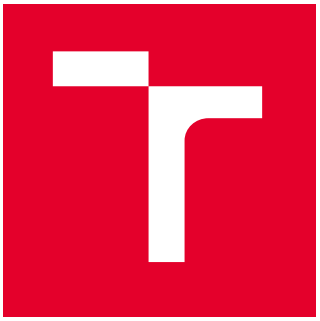


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

HLASOVÉ OVLÁDÁNÍ PRŮMYSLOVÝCH A MEDICÍNSKÝCH ZAŘÍZENÍ V RUŠNÝCH PROSTŘEDÍCH

VOICE CONTROL OF INDUSTRIAL AND MEDICAL DEVICES IN NOISY ENVIRONMENTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Lucie Vymětalíková**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Ladislav Dobrovský**

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Studentka:	**Bc. Lucie Vymětalíková**
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. Ladislav Dobrovský**
Akademický rok:	2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Hlasové ovládání průmyslových a medicínských zařízení v rušných prostředích

Stručná charakteristika problematiky úkolu:

Při práci na specializovaných a robotizovaných pracovištích v nemocnicích, laboratořích i průmyslových provozech může být výhodné hlasové ovládání pomocných zařízeních a robotů.
V případě práce s infekčními či toxickými látkami a jiným nebezpečným materiálem může být nevhodné požadovat od laboranta/operátora ovládání přes dotykový panel HMI nebo klávesnicí/myší.
Takový úkon vede na výměnu rukavic a další desinfekci u laboratorních boxů bez rukávů. Stojí tedy čas a materiál navíc.
Zároveň v takovýchto provozech bývá hluk a více pracovišť u sebe, kde laboranti/operátoři mohou mluvit mezi sebou a také chtít ovládat své pracoviště.
Například v projektu OpenTube2 byly čtyři hlavní zdroje hluku: odvětrávaný laboratorní box, řídící jednotka robotu, pohyby robotu, a automatická pipeta.

Cíle diplomové práce:

Rešerše dostupných řešení a modelů pro rozpoznávání řeči.
Volba otevřeného modelu pro analýzu řeči s možností transfer learningu.
Analýza možností odstranění hluku anebo přizpůsobení modelu pro jeho automatické potlačení.
Analýza vhodných slov a oslovování ovládaných pracovišť (předpokládáme více pracovišť vedle sebe) pro snížení nejasností (které pracoviště je ovládáno, možná záměna příkazů, ignorace běžné řeči operátorů mezi sebou).
Návrh a realizace postupu a SW pro kalibraci hluku pracoviště.
Návrh a realizace postupu a SW pro identifikaci operátorů.
SW pro rozpoznávání příkazů a dokumentace otevřeného API.
Návrh úpravy laboratorního boxu pro automatické otevírání.
Cíl navíc: prototyp automatického otevírání laboratorního boxu.

Seznam doporučené literatury:

HUANG, Xuedong, James BAKER a Raj REDDY. A historical perspective of speech recognition. Communications of the ACM [online]. 2014, 57(1), 94-103 [cit. 2022-10-10]. ISSN 0001-0782. Dostupné z: doi:10.1145/2500887.

LI, Jinyu. Recent Advances in End-to-End Automatic Speech Recognition. APSIPA Transactions on Signal and Information Processing [online]. 2022, 11(1) [cit. 2022-10-10]. ISSN 2048-7703. Dostupné z: doi:10.1561/116.00000050.

Automatic Speech Recognition with Transformer [online]. [cit. 2022-10-10]. Dostupné z: https://keras.io/examples/audio/transformer_asr/

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D. ředitel ústavu	doc. Ing. Jiří Hlinka, Ph.D. děkan fakulty

ABSTRAKT

Tato diplomová práce se zabývá hlasovým ovládáním průmyslových a medicínských zařízení v rušných prostředích. Porovnány jsou různé modely rozpoznávání řeči i metody pro odstraňování hluku z řečových signálů. Na základě rešerše i vlastních testování je sestaven vlastní systém hlasového ovládání. Systém je složen z modelu pro detekci vzbouzecí fráze a modelu pro rozpoznávání předem nadefinovaných příkazů. Implementována je v systému i audio odezva pro operátora a spouštění skriptů dle rozpoznaných příkazů. Navržena byla také úprava laboratorního boxu OpenTube2 pro automatické otevírání.

ABSTRACT

This diploma thesis deals with voice control of industrial and medical devices in noisy environments. Different speech recognition models and methods for noise supression in speech signals are compared. Based on the research and conducted testing, a custom voice control system is designed. The system consists of a wake word detection model and a model for the predefined commands recognition. An audio response for the operator and a script execution based on the recognized commands is also implemented in the system. A modification for automatic door opening of the OpenTube2 laboratory box was designed.

KLÍČOVÁ SLOVA

Hlasové ovládání, rozpoznávání řeči, ASR, potlačení hluku v řečovém signálu, beamforming, Porcupine, Whisper, automatické otevírání

KEYWORDS

Voice control, speech recognition, ASR, noise suppression in speech signal, beamforming, Porcupine, Whisper, automatic door opening

ÚSTAV AUTOMATIZACE A INFORMATIKY

2023

BIBLIOGRAFICKÁ CITACE

VYMĚTALÍKOVÁ, Lucie. *Hlasové ovládání průmyslových a medicínských zařízení v rušných prostředích*. Brno, 2023. Dostupné také z: `https://www.vut.cz/studenti/zav-prace/detail/149745`. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky, Vedoucí práce: Ing. Ladislav Dobrovský

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato diplomová práce je mým původním dílem, vypracovala jsem ji samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autorka uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků.

V Brně dne 24. 5. 2023 .....

Lucie Vymětalíková

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu práce Ing. Ladislavovi Dobrovskému za cenné rady a ochotný přístup při vedení této práce.

OBSAH

1 ÚVOD

Hlasové ovládání nejrůznějších rozhraní dosahuje v současné době vysoké popularity. Řeč je pro člověka nejpřirozenějším způsobem komunikace, proto je interakce člověka se zařízeními za použití hlasu velmi jednoduchá a efektivní. V posledních letech dochází k výraznému zlepšení kvality systémů rozpoznávání řeči, a to i v robustnosti vůči rušným prostředím.

Využitím hlasového ovládání pro ovládání průmyslových a medicínských zařízení je možno dosáhnout značného navýšení produktivity práce. Zároveň je v oblasti průmyslu a medicíny práce často vykonávána ve sterilních prostředích či je pracováno s toxickými látkami. Hlasovým ovládáním lze zamezit nutnosti časté výměny pracovních rukavic a desinfekci, tedy je docíleno i snížení spotřeby pracovního materiálu. Operátor má při ovládání pracoviště hlasem volné ruce i oči, dokáže se tak na vykonávanou práci lépe soustředit, čímž je spolu s omezením lidských chyb zajištěna i vyšší bezpečnost práce.

Systémy pro hlasové ovládání průmyslových a medicínských zařízení jsou dostupny převážně na zahraničním trhu, v České republice je hlasového ovládání využíváno především v logistice.

Práce je rozdělena do pěti hlavních kapitol. První kapitola uvádí přehled dostupných řešení hlasového ovládání na zahraničním i českém trhu. Druhá a třetí kapitola jsou rešeršního charakteru a věnují se modelům automatického rozpoznávání řeči a možnostem zvýšení kvality systémů rozpoznávání řeči v rušných prostředích. Ve čtvrté kapitole je navržen vlastní systém hlasového ovládání a kapitola pátá se zabývá úpravou laboratorního boxu OpenTube2 pro automatické otevírání.

Hlavním cílem práce bylo navrhnout systém hlasového ovládání, který je i v rušných prostředích schopen hlasové příkazy rozpoznávat. Při návrhu byl zohledněn předpoklad více ovládaných pracovišť vedle sebe, tedy systém má rozpoznat, které pracoviště je ovládáno, a i ignorovat běžnou řeč operátorů. Dalším cílem byl i návrh úpravy laboratorního boxu OpenTube2 pro automatické otevírání, které by spolu s hlasovým ovládáním přineslo značné usnadnění obsluhy pracoviště.

2 PŘEHLED DOSTUPNÝCH ŘEŠENÍ

Hlasové ovládání zařízení se spolu s rychlým rozvojem systémů rozpoznávání řeči v posledních letech pozvolna dostává i do oblasti průmyslu a medicíny. Důvodem je především usnadnění a zefektivnění práce. Kompletní řešení hlasového ovládání zařízení lze najít převážně na zahraničním trhu, v Česku je tato nabídka zatím dosti omezená.

2.1 Český trh

V průmyslu je v České republice ovládání hlasem využíváno především v logistice pro usnadnění práce skladníků. Tomuto použití se věnují například firmy EPG, Kvados a Ayes, které všechny využívají technologií převzatých od jiných společností.

Společnost EPG používá systém Lydia Voice Suite, který nabízí řadu systémů pro intralogistiku, výrobu, údržbu a řízení kvality [1]. Systémy jsou speciálně navrženy pro hlučná prostředí skladů a distribučních center. Jejich systém Pick by Voice například slouží k jednoduchému poskytnutí informací uživateli o poloze, množství a popisu produktů ve skladu. Systém Check by Voice provází uživatele pracovním procesem. Pracovník si hlasem postupně kontroluje plnění všech pracovních úkonů, přičemž se neustále může zrakem i rukama soustředit na prováděnou práci. Zajímavé je i řešení Lydia VoiceWear (viz obr. 1), ve kterém je místo klasického použití sluchátek s mikrofonem a malého výpočetního zařízení celý systém integrován do malé vesty. Lydia Voice slibuje i spolehlivou možnost použití s rouškou či jinou jednoduchou pokrývkou obličeje bez nutnosti jakýchkoliv úprav systému. [2]

Obr. 1: Vesta Lydia VoiceWear [3]

Kvados využívá technologie Honeywell Voice [4]. Hlasové ovládání opět přináší usnadnění práce skladníkům a zaručuje vyšší bezpečnost, efektivnost a přesnost

práce. Řešení Honeywell Voice je také připraveno pro hlučná prostředí a uvádí navýšení efektivity o více než 30 %, vyloučení až 80 % lidských chyb a až 20% navýšení bezpečnosti práce. [5]

Zaměřením společnosti Ayes jsou chytré brýle, které umožňují zobrazení pracovních postupů přímo před očima pracovníků, čímž zajišťují vyšší kvalitu práce a eliminují nutnost používání tištěných materiálů. Kromě zobrazování souborů lze pomocí chytrých brýlí sdílet, co pracovník vidí, je tedy možné s pracovníkem vzdáleně spolupracovat, což přináší výhody například servisním a údržbářským týmům. Brýle opět nachází využití i v logistice, nabízí pracovníkům zobrazení prostředí skladu pro navigaci i informace o zboží. Uživatelské rozhraní brýlí je možno plně ovládat hlasem. [6]

Obr. 2: Chytré brýle Ayes [7]

V oblasti medicíny v České republice hlasového ovládání využíváno není. Okrajově se k tomuto oboru blíží hlasem ovládaná polohovatelná lůžka společnosti LINAK, která jsou ale primárně určena pro běžné použití v domácnosti. LINAK vyrábí elektrické lineární pohony zajišťující plynulý pohyb v nejrůznějších oblastech, mezi které patří například zemědělství, vybavení kancelářských interiérů, průmyslová automatizace a zdravotnictví. Řešení pracuje s hlasovými asistenty Amazon Alexa či Google Assistant, kterým jsou pomocí aplikace Bed IoT přiřazeny speciální fráze, pomocí kterých uživatel lůžko ovládá. [8]

2.2 Zahraniční trh

V zahraničí je nabídka řešení hlasového ovládání výrazně rozsáhlejší, a to v oblasti průmyslu i medicíny.

Jedním z prvních řešení hlasového ovládání v průmyslu se stal systém Athena od společnosti iTSpeeX. Motivací k vytvoření tohoto systému bylo především zjednodušení ovládání strojů a možnost rychlého ovládání různých i operátorům neznámých strojů pomocí jednotných jednoduchých příkazů. Místo zdlouhavého nastavování konkrétního procesu a nutné přesné znalosti ovládání všech strojů, může operátor pro ovládání více strojů použít například stejný příkaz „Zahájit zahřívací cyklus.“ Systém kombinuje možnost zadávání příkazů hlasem, ale i pomocí klávesnice. [9]

Řešení vicCONTROL se zabývá aplikací hlasového ovládání v různých oblastech, tedy v průmyslu i v medicínských technologiích. VicCONTROL nabízí volbu jedinečného vzbouzecího slova pro systém rozpoznávání řeči. Zpracování řeči nevyžaduje posílání dat do externích služeb a probíhá offline přímo v zařízení, čímž je zajištěna bezpečnost zpracovávaných dat. Zajímavou vlastností systému je i využití algoritmu porozumění přirozenému jazyku, pomocí čehož lze systém dále rozšířit i pro možnost vedení dialogu se zařízením. Pro získání co nejlepších řečových signálů v hlučných prostředích je možné využít metody beamforming. Systém je připraven pro použití ve 30 jazycích a správa ovládaných zařízení a jednotlivých příkazů k jejich ovládání je umožněna pomocí přehledného webového rozhraní. [10]

Pro rozsáhlejší řešení vicCONTROL používá hardware od společnosti Spectra, která hlasové ovládání nazývá třetí rukou člověka. Mezi hlavní benefity ovládání hlasem Spectra řadí provádění akcí bez nutnosti použití rukou. Především v oblastech jako jsou medicína, chemie či farmacie, ve kterých je třeba dbát na zvýšenou hygienu a často je vyžadováno i použití rukavic, může být dosti obtěžující. [11]

Dalším řešením je Digi ConnectCore Voice Control, které nabízí vývojové sady a software, za pomocí kterých si uživatel sestaví vlastní aplikaci hlasového ovládání. Systém je opět podporován pro 30 jazyků a zpracování dat probíhá offline pouze v daném zařízení. Digi uvádí, že jazykový model je natrénován na 60 000 slov a také nabízí možnost volby vlastního vzbouzecího slova. Užití systému nevyžaduje žádná přídavná výpočetní zařízení. Mezi příklady použití systému jsou uvedeny hlasové ovládání medicínských zařízení pro jednoduché zjišťování stavu pacienta, hlasové ovládání parkovacích automatů, ale například i ovládání zemědělských zařízení. [12]

K zavedení hlasového ovládání už se přiklání například i známé firmy Siemens a Roche. Pro Siemens byla hlavní pohnutkou implementace hlasového ovládání práce s toxickými látkami a práce ve sterilním prostředí. Při těchto pracích je nutné použití rukavic a pro ovládání zařízení dotykovým panelem je tedy vyžadováno jejich sundávání a následně opětovné oblékání, které je neefektivní nejen časově, ale i vysokou spotřebou pracovních rukavic [13]. V Roche je hlavním důvodem vývoje hlasem ovládaného asistenta celkové usnadnění práce zaměstnanců. [14]

Jako příklad dalších firem zabývajících se hlasovým ovládáním lze uvést ArkX Laboratories, Paragon semvox a Concept Reply.

V medicínském prostředí se mimo hlasového ovládání zařízení zpracování řeči často vyskytuje k účelu automatického zápisu poznámek. Pro zvýšení přesnosti jsou modely přitrénovány na speciální medicínské výrazy. Záznamy z lékařských prohlídek následně mohou být přímo ukládány ve formě rozhovoru, nebo z nich může být generována lékařská zpráva. V obou případech tyto systémy slouží ke snížení času, který lékař musí věnovat administrativním činnostem a zároveň snižují riziko výskytu lidské chyby. Nejrozšířenějšími systémy s tímto účelem jsou DeepScribe a SpeechWrite.

3 AUTOMATICKÉ ROZPOZNÁVÁNÍ ŘEČI

Automatické rozpoznávání řeči (ASR) je jednou z hlavních řešených úloh oblasti strojového učení již od roku 1970 [15]. V posledních letech však poptávka po ASR systémech výrazně narůstá, a to především díky rozvoji chytrých zařízení a jejich využití v našich běžných životech. ASR nachází využití v různých aplikacích v mobilních telefonech, u hlasových asistentů nebo například i při ovládání automobilů. Řeč je pro člověka nejpřirozenějším způsobem komunikace, což je to, co ji činí tak snadno dostupnou pro použití k ovládání nejrůznějších rozhraní.

Úkolem systémů automatického rozpoznávání řeči je převod audio signálu mluvené řeči do její psané podoby. Vstupní signál je zpracován parametrizačním modulem, pomocí kterého je signál rozdělen na rámce, které jsou typicky 10 ms dlouhé, a následně jsou pro ně počítány vektory příznaků. Cílem extrakce příznaků je komprimovat řečový signál ve formě vlnění za minimální ztráty informace [16]. Nejčastěji používanými metodami extrakce příznaků je určování Mel-frekvenčních kepstrálních koeficientů (MFCC) a percepční lineární predikce (PLP) [17].

ASR systém tedy vstupní sekvenci příznaků $X = \{x_1, ..., x_T\}$ délky T přiřazuje výstupní sekvenci $L = \{l_1, ..., l_N\}$, která obecně představuje písmena, slova, nebo znaky, délky N. Všechny tyto možné výstupy jsou pro konkrétní systém uloženy ve slovníku ν. Pomocí $\nu*$ popisujeme množinu všech možných sekvencí vzniklých z obsahu slovníku ν. Cílem ASR je přiřadit dané vstupní sekvenci X výstupní sekvenci $\hat{L}$ s nejvyšší pravděpodobností. Formálně lze tuto problematiku zapsat vztahem (1). [15]

$$\hat{L} = \arg\max_{L \in \nu*} p(L|X) \tag{1}$$

Základním úkolem automatického rozpoznávání řeči je tedy vytvoření modelu, který dokáže přesně určovat pravděpodobnost $p(L|X)$. Modely lze dle jejich principů rozdělit do dvou základních kategorií, a to jsou modely založené na skrytých Markovových modelech (HMM) a tzv. end-to-end modely. [15]

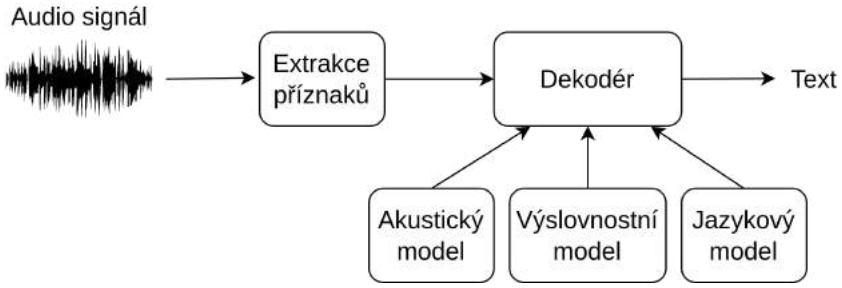
3.1 Modely založené na skrytých Markovových modelech

Modely založené na HMM byly dlouhou dobu modely dosahujícími nejlepších výsledků. Tyto modely lze rozdělit do tří na sobě nezávislých částí: akustický, výslovnostní a jazykový model.

Akustický model mapuje vektory příznaků na vektory krátkých akustických jednotek (typicky fonémů). Výslovnostní model, který je většinou sestaven profesionálními lingvisty, slouží k mapování fonémů na grafémy, tedy mapování nejmenších akustických jednotek na nejmenší jednotky psaného jazyka. Jazykový model udává

pravděpodobnost výskytu určité posloupnosti slov v daném jazyce, což slouží jako rozhodující faktor při výběru správné varianty z více podobně znějících slov o různých významech. [15]

Systém musí obsahovat ještě dekodér, který se s pomocí těchto tří modelů snaží nalézt optimální sekvenci $\hat{L}$ přes všechna řešení $\nu*$. Nejčastěji používaný algoritmus pro dekódování je Viterbiho algoritmus [16]. Schéma celkového modelu je znázorněno na obrázku 3.



Obr. 3: Schéma modelu založeného na HMM

HMM nachází hlavní využití v akustickém modelu, kde akustické jednotky (fonémy) představují stavy modelu a příznaky jsou jednotlivá pozorování. Užitím Bayesova vzorce, který udává, jak podmíněná pravděpodobnost jevu souvisí s opačnou podmíněnou pravděpodobností, lze vzorec (1) zapsat jako:

$$\begin{aligned}\hat{L} &= \arg\max_{L\in\nu*} p(L|X) \\ &= \arg\max_{L\in\nu*} \frac{p(L|X)}{p(X)} \\ &= \arg\max_{L\in\nu*} p(L,X) \\ &= \arg\max_{L\in\nu*} \sum_{S} p(L,S,X),\end{aligned} \tag{2}$$

kde $S = \{s_t \in \{1,...,J\} | t = 1,...,T\}$ je posloupnost stavů HMM a J je počet stavů modelu. Dále lze rovnici (2) rozepsat následovně:

$$\begin{aligned}\hat{L} &= \arg\max_{L\in\nu*} \sum_{S} p(L,S,X) \\ &= \arg\max_{L\in\nu*} \sum_{S} p(X|S,L)p(S,L) \\ &= \arg\max_{L\in\nu*} \sum_{S} p(X|S,L)p(S|L)p(L).\end{aligned} \tag{3}$$

Na základě hypotézy podmíněné nezávislosti aproximujeme $p(X|S, L) \approx p(X|S)$, čímž z rovnice (3) získáme výsledný tvar:

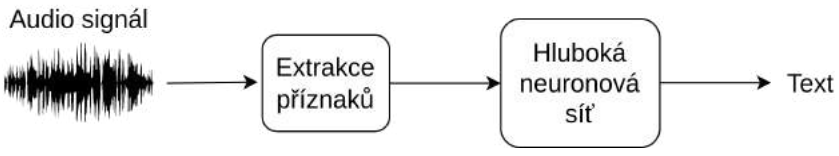
$$\hat{L} \approx \arg\max_{L \in \nu*} \sum_{S} p(X|S)p(S|L)p(L), \tag{4}$$

ve kterém $p(X|S)$ představuje akustický model, $p(S|L)$ výslovnostní model a $p(L)$ model jazykový.

HMM předpokládá nezávislost pozorování, což znamená, že pozorování v jakémkoliv čase je závislé pouze na skrytém stavu v daném čase. Pro určování pravděpodobností pozorování existují dvě různé architektury akustických modelů. První k výpočtům využívá směsi Gaussovských rozdělení (GMM) a následný model se označuje HMM-GMM, druhá využívá hlubokých neuronových sítí (DNN) a model je značen HMM-DNN. HMM-GMM byly dlouhou dobu používány v základních strukturách modelů rozpoznávání řeči. S rozvojem technologií DNN ale brzy HMM-DNN začaly dosahovat lepších výsledků a staly se tak ve své době state-of-the-art řešením pro rozpoznávání řeči. [15]

3.2 End-to-end modely

End-to-end (E2E) modely přímo mapují vstupní audio sekvence na sekvence slov nebo grafémů za použití jedné neuronové sítě. Oproti předchozím modelům založeným na HMM tedy není nutné trénovat zvlášť akustické, výslovnostní a jazykové modely. Je tedy výrazně zjednodušen proces trénování modelu i následný sytém rozpoznávání řeči (viz obr. 4). Hlavní výhodou E2E modelů je použití jedné účelové funkce k optimalizaci celé sítě, zatímco u modelů založených na HMM se jednotlivé části optimalizují zvlášť, což nemusí zaručit dosažení globálního minima. Díky tomu E2E modely prokazatelně překonávají výsledky modelů založených na HMM a jsou v současnosti state-of-the-art ASR systémů. [18]



Obr. 4: Obecné schéma end-to-end modelu

E2E modely lze dle použitých technik rozdělit do tří kategorií: modely založené na konekcionistické časově závislé klasifikaci (CTC), modely s attention mechanismem a modely využívající převodníku rekurentní neuronové sítě (RNN-T).

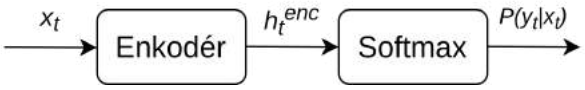
3.2.1 CTC modely

CTC je účelová funkce používaná pro klasifikaci sekvenčních dat, která nevyžaduje přesné zarovnání vstupních dat na výstupní třídy. Tato vlastnost je užitečná pokud pro určité časové rámce nejsou k dispozici výstupní třídy.

V řečovém signálu nemusí jednotlivé fonémy vždy spadat do samostatných rámců. V jednom rámci se mohou dva fonémy překrývat. Při trénování neuronové sítě ale i přes to dochází ke klasifikaci rámce jako jeden z fonémů, což má na výsledky trénování negativní vliv. CTC klasifikuje pouze rámce, u kterých je zřejmé, do které třídy patří, ostatní rámce jsou zařazeny do třídy *blank*. Poté jsou pomocí many-to-one mapování β odstraněny opakující se znaky a jsou odstraněny *blank* znaky [19]. Příklad mapování je uveden v rovnici (5), třída *blank* je znázorněna pomocí $-$.

$$\beta(--hh--e--ll-lll-ooo-) = \beta(-h-e-l-l-o-) = hello \qquad (5)$$

Modely využívající CTC se skládají z enkodéru a výstupní softmax vrstvy (viz obr. 5). Enkodéry obecně mapují vstupní sekvenci řečového signálu na příznaky vyššího řádu.



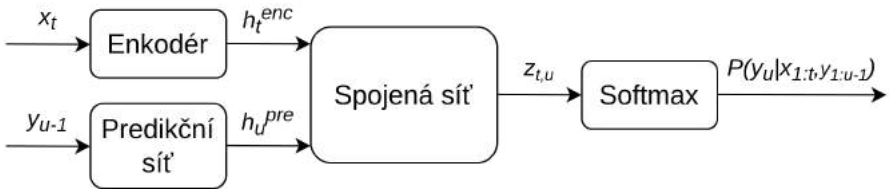
Obr. 5: Architektura modelu využívajícího CTC, upraveno z [18]

Vektor příznaků je v obr. 5 označen x_t, h_t^{enc} jsou příznaky vyššího řádu a $P(y_t|x_t)$ je pravděpodobnost, že za vstupní sekvence x_t je výstupní sekvence právě y_t.

3.2.2 Modely využívající RNN-T

RNN-T modely byly navrženy jako rozšíření k přístupu CTC a také využívají výstupní třídy *blank*. Hlavní změnou je přidání predikční rekurentní neuronové sítě (RNN), která generuje příznaky vyššího řádu na základě předchozích výstupů. Celkový model tedy obsahuje enkodér, predikční síť, spojenou síť a softmax vrstvu. [20]

Schéma modelu je zobrazeno na obrázku 6, kde y_{u-1} představuje předchozí výstup, h_u^{pre} příznaky vyššího řádu předchozího výstupu, $z_{t,u}$ je výstup spojené sítě, která kombinuje příznaky z enkodéru a předchozího výstupu a výsledná pravděpodobnost P, že výstupní sekvence je y_u, je podmíněná všemi dosud přijatými vstupy i všemi předchozími výstupy.



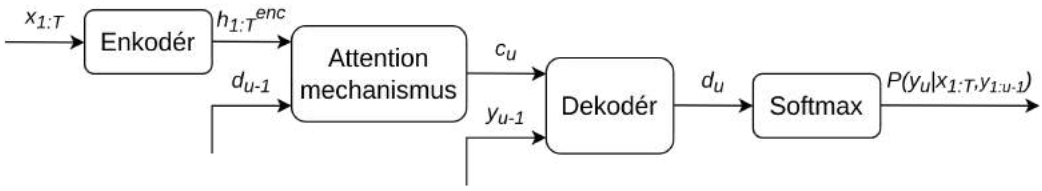
Obr. 6: Architektura RNN-T modelu, upraveno z [18]

3.2.3 Modely s attention mechanismem

V roce 2014 byla publikována práce o strojových překladech [21], jejíž cílem bylo řešit u modelů architektury enkodér-dekodér typický problém enkódování vstupního textu do vektorů fixní délky, což vedlo především u dlouhých vět k omezené schopnosti enkódování. V této práci byl vstupní text místo běžného jednoho vektoru enkódován do sekvence vektorů. Dekodér dále pomocí attention mechanismu neboli mechanismu pozornosti v jednotlivých krocích přiřazoval různé váhy každému z těchto vektorů, dle toho, za jak důležité je považoval (jak velkou jim věnovat pozornost - attention), přičemž výstupní vektor každého následujícího kroku bral ohled na předchozí vektor vah i na předchozí výstup.

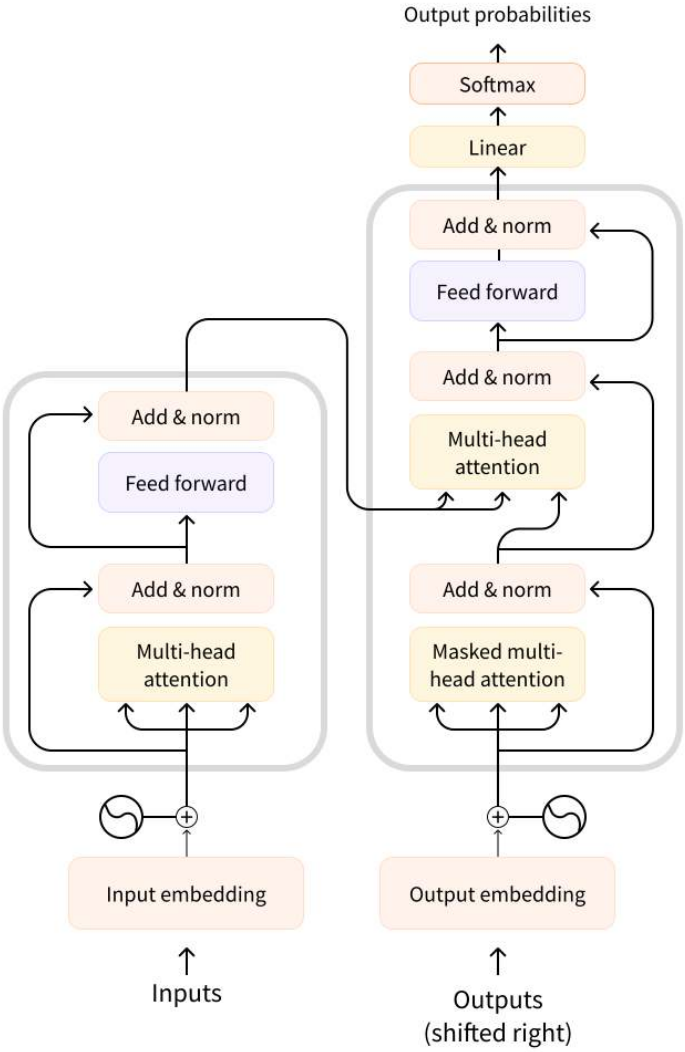
Rozpoznávání řeči lze v principu označit za úkol obdobný k strojovému překladu, jedná se totiž také o přepis vstupní sekvence na sekvenci výstupní. Tedy i pro ASR systémy je pomocí attention mechanismu dosaženo lepších výsledků při přepisu dlouhých vstupních sekvencí. Systémy s attention mechanismem se tak staly hlavním objektem zájmu end-to-end modelů rozpoznávání řeči. [15]

Architektura modelu je zobrazena na obrázku 7, c_u představuje kontextový vektor attention mechanismu a d_u příznaky získané z dekodéru.



Obr. 7: Architektura modelu s attention mechanismem, upraveno z [18]

V roce 2017 byla v článku *Attention Is All You Need* představena architektura modelu Transformer [22] (viz. obr 8) sloužícího opět pro strojový překlad. Jak již název článku napovídá, jedná se o architekturu enkodér-dekodér založenou pouze na attention mechanismu. Představený Transformer díky schopnosti „pochopení“ kontextu celé vstupní sekvence předčil výsledky předchozích řešení a položil základ současným jazykovým modelům, jako jsou například BERT, BART a GPT.



Obr. 8: Architektura modelu Transformer [23]

3.3 Dostupné modely rozpoznávání řeči

Otevřených (open-source) modelů rozpoznávání řeči je poměrně mnoho. Liší se od sebe především architekturou modelů, velikostí modelu, daty, na kterých byly modely trénovány, a požadovaným předzpracováním vstupních audio dat.

Jedním z prvních ASR softwarů byl Julius, jehož vývoj byl zahájen na Kjótské univerzitě již v roce 1991. Software byl dále ještě dlouhá léta vyvíjen a poslední verze byla vydána v roce 2019. Jedná se o HMM-DNN modely, které nabízí možnost přepisu angličtiny a japonštiny. [24]

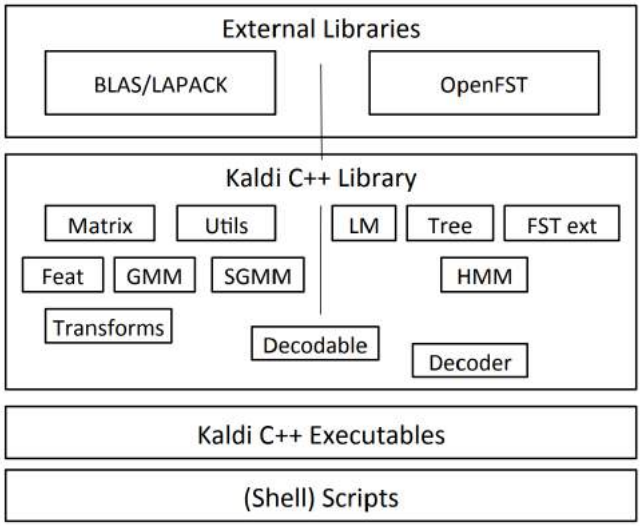
Dalším zajímavým ASR je end-to-end model projektu DeepSpeech od společnosti Mozilla, který byl poprvé představen v práci [25] v roce 2014. Trénovací dataset největšího DeepSpeech modelu byl sestaven z 7380 hodin audio dat, z nichž

většina byla tvořena čtením textu, zbytek byla řeč z běžných konverzací. Poslední verze modelu byla vydána v prosinci roku 2020 [26].

Dále budou podrobněji popsány a následně zhodnoceny další tři modely: Kaldi GigaSpeech ASR XL, wav2vec 2.0 a Whisper, které všechny nabízí možnost přitrénování na vlastních datech.

3.3.1 Kaldi GigaSpeech ASR XL

Kaldi je open-source toolkit navržený k výzkumu rozpoznávání řeči, jehož vývoj započal v roce 2009 [27]. Toolkit je psán v jazyce C++ a ovládání je umožněno pomocí shellových skriptů. Prvotní komponenty toolkitu byl navrženy pro modely založené na HMM a GMM [28], s postupným vývojem ale byla implementována i možnost tvorby E2E modelů [29]. Původní komponenty toolkitu Kaldi jsou zobrazeny na obrázku 9.



Obr. 9: Komponenty původního toolkitu Kaldi [28]

Mimo toolkit Kaldi představilo také několik svých modelů. Nejnovějšími je řada čtyř modelů Gigaspeech ASR, které jsou dle velikosti trénovacího datasetu pojmenovány S, M, L a XL. Všechny modely byly trénovány na datasetu Gigaspeech, respektive jeho částech. Model GigaSpeech ASR S byl trénován na 250 hodinách audio dat, model M na 1000 h, model L na 2500 h a největší model XL na 10 000 h. [30] Dataset Gigaspeech byl sestaven ze tří zdrojů audio dat, všechny jsou v datasetu téměř vyváženě zastoupeny. Jedná se audioknihy, podcasty a YouTube [31].

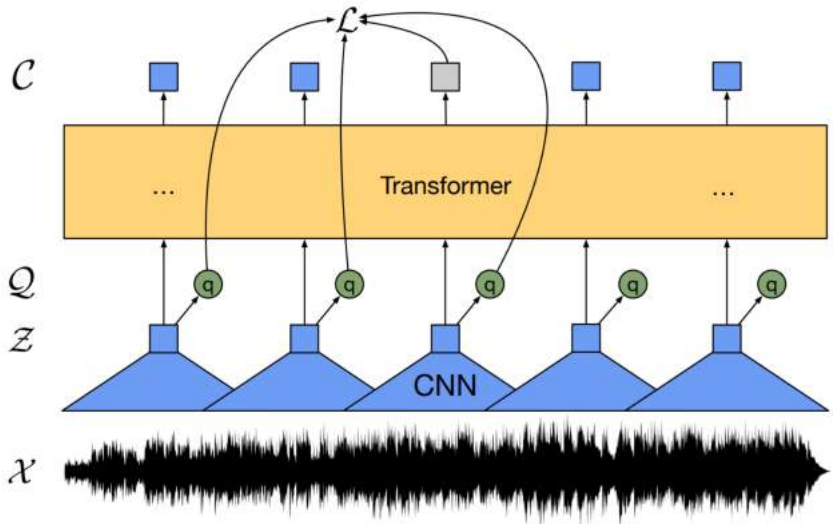
Nevýhodou Kaldi je nemožnost zpracování dlouhých audio souborů. Pro inferenci tedy musí být dlouhé soubory rozděleny do menších částí. Jako předzpracování audio dat je vyžadována pulzně kódová modulace (PCM) o vzorkovací frekvenci 16 kHz. Celkově se práce s Kaldi jeví být poměrně náročná.

3.3.2 Wav2vec 2.0

Wav2vec 2.0 je ASR framework představený Facebook AI v říjnu roku 2020 jako nástupce předchozího wav2vec. Wav2vec se od ostatních řešení liší především tím, že hlavní část trénování modelu probíhá učením bez učitele. Trénovací data jsou tedy pouze audio data bez jejich přepisů a model si sám vytváří reprezentace řeči. Po té je již učením s učitelem proveden fine-tuning modelu na audio datech včetně jejich přepisů, ve kterém se původní model naučí přiřazovat vlastní reprezentace řeči výsledným fonémům, nebo slovům. Pro tuto část je však potřeba mnohem menšího množství dat. Fine-tuning modelu využívá CTC a pro dekódování výsledku je třeba použití wav2vec 2.0 tokenizéru.

Výhodou celého tohoto trénovacího procesu je, že je snadné získat velké množství audio dat bez jejich přepisů pro trénování v první části. V práci [32] představující wav2vec 2.0 byl model předtrénován na 53 000 h dat a následně byl testován fine-tuning s různým množstvím dat. Při fine-tunování na 960 h dat dosahoval model četnosti chybných slov (WER) 1,8 % na testovacím datasetu čisté řeči a 3,3 % na řeči zašuměné. Pokud byl pro porovnání fine-tuning proveden pouze na 10 minutách, bylo dosaženo WER 4,8 % na čisté řeči a 8,2 % na zašuměné [32]. Je tedy vidět, že lze dosahovat velmi dobrých výsledků i za použití malého množství dat přepsané řeči, což může být dosti výhodné například pro použití na málo rozšířených jazycích.

Model přímo zpracovává audio signál o vzorkovací frekvenci 16 kHz. Učení řečových reprezentací je zobrazeno na obrázku 10, kde X představuje vstupní audio signál, Z jsou skryté řečové reprezentace vystupující z konvoluční sítě (CNN) a Q jsou kvantizované reprezentace. Model využívá Transformeru pro získání kontextu C z reprezentací řeči. L je chybová (loss) funkce.



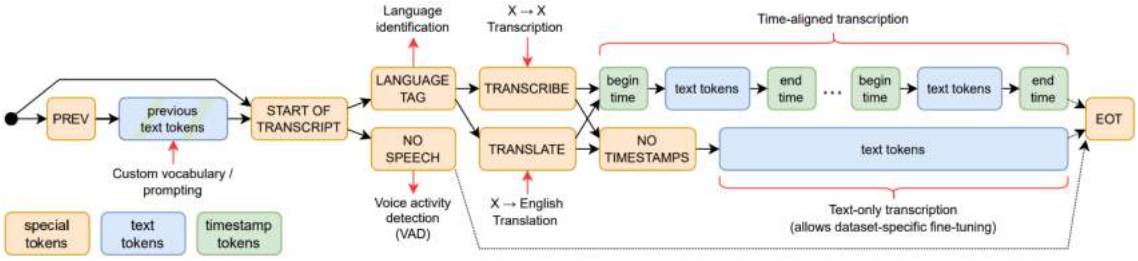
Obr. 10: Učení řečových reprezentací modelu wav2vec, upraveno z [32]

Hlavní nevýhodou modelu je, že byl trénován na datech z audioknih, což je vhodné pro rozpoznávání čisté či čtené řeči, ale při rozpoznávání řeči s hlukem z okolí nebo běžných konverzací může dosahovat horších výsledků.

3.3.3 Whisper

Whisper je nejnovějším ASR systémem. Představen byl v září roku 2022 společností OpenAI. Trénován byl učením s učitelem na 680 000 hodin multilingválních dat, která byla sesbírána z webu. Jelikož přepis audio dat nebyl lidsky kontrolován a může proto obsahovat i nepřesnosti, označuje společnost učení jako „weakly supervised“, tedy přechod mezi učením s učitelem a bez něj. 65 % trénovacích dat tvoří audio v anglickém jazyce s anglickým přepisem, přibližně 18 % je tvořeno neanglickými audio daty s anglickým přepisem a zbylých 17 % jsou neanglická data s přepisem v jimi odpovídajícím jazyce. Neanglická data reprezentují 98 různých dalších jazyků. Cílem tvorby modelu bylo prokázat, jak tak obrovský dataset o velké diverzitě dat vede k výraznému zlepšení robustnosti modelu, a to například vůči různým přízvukům a hluku z okolí. [33]

Jedná se o end-to-end model implementovaný jako Transformer typu enkodér/dekodér. Model nabízí nejen přepis řeči v mnoha jazycích, ale i překlad těchto jazyků do angličtiny. Zároveň je model schopný identifikovat vstupní jazyk, detekovat hlasovou aktivitu a přiřazovat k výstupnímu textu časové úseky. Schéma trénování takového multitasking modelu je zobrazeno na obrázku 11.

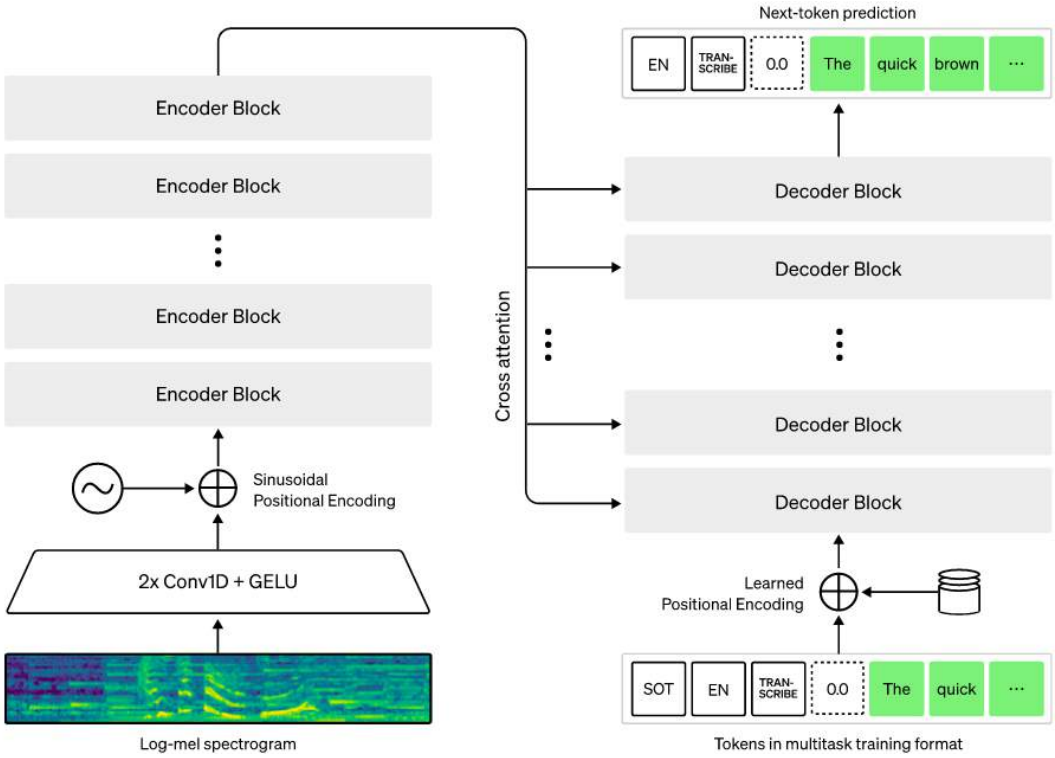


Obr. 11: Formát multitasking trénování modelu Whisper, upraveno z [33]

Zpracování audia před vstupem do enkodéru se skládá z rozdělení audia do 30 s dlouhých úseků a následné určení log-Mel spektrogramů pro tyto úseky. Audio opět vyžaduje vzorkování o frekvenci 16 kHz. Whisper do výstupního textu přidává i interpunkci a rozlišuje velká a malá písmena, zatímco předchozí Kaldi a wav2vec 2.0 vrací pouze posloupnosti slov psané velkými písmeny.

Whisper modely jsou dostupné v šesti verzích dle velikosti: tiny, base, small, medium, large a large-v2. čtyři nejmenší verze (tiny až medium) se dále dělí i na „English-only“ modely trénované pouze na anglických datech a na „Multilingual“ modely trénované na multilingválních datech.

Architektura modelu Whisper je vyobrazena na obrázku 12.



Obr. 12: Architektura modelu Whisper [34]

3.3.4 Benchmarking modelů

Benchmarking těchto tří modelů byl proveden v článku [35] od Deepgram. Whisper byl testován v anglické verzi o velikosti medium. Testovací dataset byl sestaven z interních datasetů Deepgramu. Vybráno bylo 50 souborů z pěti různých oblastí. Prvními z nich jsou konverzace o objednávání jídla mezi lidmi a chatboty. Nahrávky obsahují hluk i cizí konverzace v pozadí. Druhá část je tvořena telefonními hovory převážně z call center, jedná se tedy opět o ne zrovna kvalitní řečové audio obsahující další konverzace v pozadí. Třetí oblastí jsou videa, která naopak poskytují řeč v poměrně vysoké kvalitě. Další oblastí jsou meetingy a poslední jsou konferenční hovory, které obsahují převážně připravenou čtenou řeč, tedy řeč v dobré kvalitě.

Pro vyhodnocování pomocí WER se běžně provádí normalizace výstupního textu modelu a přepisu audio dat za účelem odstranění rozdílů ve formátování trénovacích a testovacích dat. Jelikož Whisper má vlastní normalizér, který kromě běžného převodu textu do malých písmen a odstranění interpunkce pracuje například i s různým mapováním číslic, adres a měn, bylo testování provedeno s tímto Whisper normalizérem i s běžnou normalizací.

Výsledky testování jsou zobrazeny v tabulce 1. Lze vidět, že Whisper jednoznačně dosahuje nejlepších výsledků napříč všemi oblastmi.

Tab. 1: WER modelů [35]

	Whisper normalizér			**Běžná normalizace**		
Dataset	**Kaldi**	**w2v 2.0**	**Whisper**	**Kaldi**	**w2v 2.0**	**Whisper**
Konverzace s AI	64,2	36,3	19,9	63,4	34,5	26,5
Tel. hovory	69,9	31,0	16,6	70,7	30,9	20,8
Meetingy	44,0	27,4	13,9	45,8	28,7	16,1
Konf. hovory	65,8	28,1	9,7	69,1	35,3	11,2
Videa	47,6	23,3	8,9	47,5	23,5	11,2

4 MOŽNOSTI ZVÝŠENÍ KVALITY ASR V RUŠNÝCH PROSTŘEDÍCH

Odstranění nežádoucího hluku je v oblasti zpracování audio signálu jedním z hlavních řešených úkolů již od vynálezu mikrofonu. Jedním z přístupů pro potlačení hluku a udržení řečového signálu je filtrování signálu již při jeho příjmu, a to pomocí metody beamforming. Další možností je zpracování již získaného signálu pomocí různých filtrů, které šum potlačují, nebo v dnešní době i pomocí neuronových sítí trénovaných k odstranění hluku z řečových signálů. Dále je ke zlepšení výsledků automatického rozpoznávání řeči možná i úprava jazykových modelů právě pro použití v rušných prostředích. Tyto tři přístupy budou dále podrobněji rozebrány.

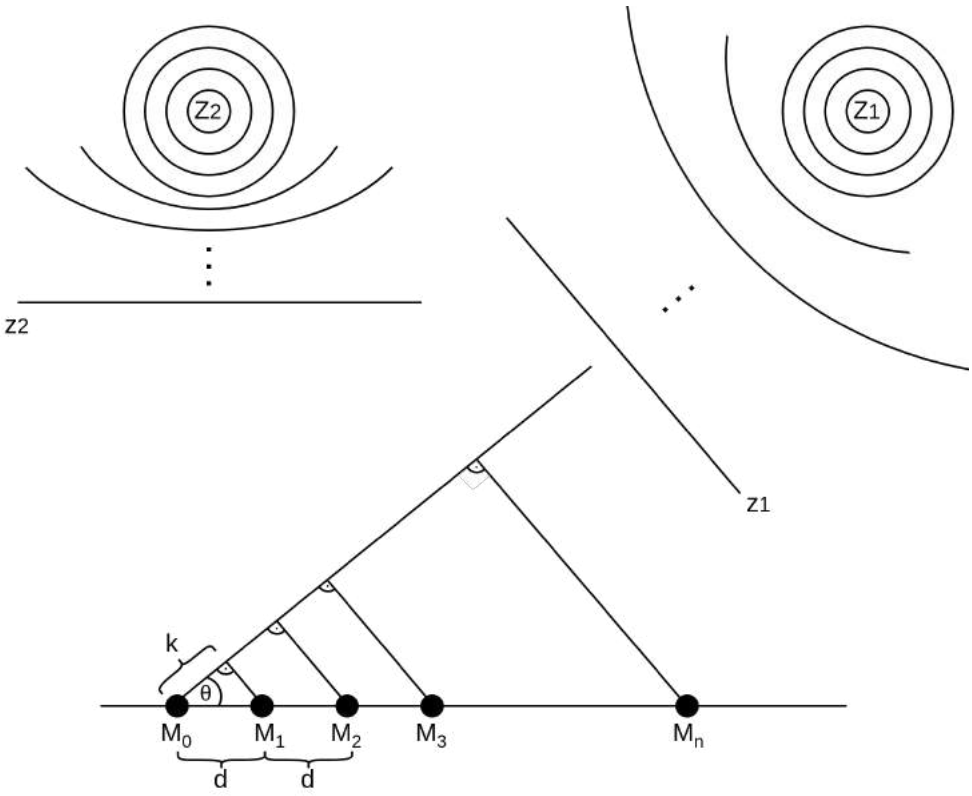
K porovnání úrovně požadovaného signálu s úrovní šumu v signálu se používá poměr signálu a šumu (SNR), který je definován jako poměr výkonu signálu k výkonu šumu. SNR se vyjadřuje v decibelech.

Mimo výše zmíněné běžné přístupy potlačování hluku se v současné době ke zvýšení kvality ASR vyvíjí také řešení kombinující ASR modely s modely, které řeč rozpoznávají na základě snímání pohybu rtů při mluvení. Takový systém se pak nazývá audiovizuální rozpoznávání řeči a jedno z řešení bylo představeno například v článku [36].

4.1 Beamforming

Technologie beamforming je v dostupných řešeních hlasového ovládání často využívána. Označována bývá také jako prostorové filtrování. Zjednodušeně se jedná o použití více mikrofonů, díky kterým lze získat informace o směru přicházejících signálů a nežádoucí signály okolního hluku odfiltrovat. Běžně se lze s touto technologií setkat i u mobilních telefonů, notebooků nebo například sluchátek, které mají více mikrofonů umístěných nedaleko od sebe.

Beamforming algoritmů je hned několik. Základním algoritmem je algoritmus Delay-And-Sum („zpoždění a součet“). Tento algoritmus uvažuje rovnou řadu n mikrofonů se stejnými vzdálenostmi d mezi nimi a využívá předpokladu, že zvukové vlny šířící se ze zdroje umístěného dostatečně daleko lze považovat za rovinné. Dle polohy zdroje zvuku dopadají zvukové vlny na mikrofony v různých časech. Mikrofony tedy snímají stejný signál, ale mírně posunutý v čase. Jelikož je uvažováno, že dopadající vlny jsou rovinné, dopadají na mikrofony pod stejným úhlem θ. [37] Schéma je zobrazeno na obrázku 13, ve kterém Z_1 a Z_2 představují dva zdroje zvuku, rovinné vlny po šíření jsou z_1 a z_2, M jsou mikrofony v řadě a k je vzdálenost, kterou musí vlna mezi jednotlivými mikrofony urazit navíc.



Obr. 13: Zvukové vlny dopadající na řadu mikrofonů

Pro získání signálu z požadovaného úhlu θ a potlačení signálů z ostatních úhlů je třeba vypočítat k jakému časovému zpoždění signálu T_d dojde při dopadech na jednotlivé mikrofony. Pro předem známé d a θ lze vypočítat k:

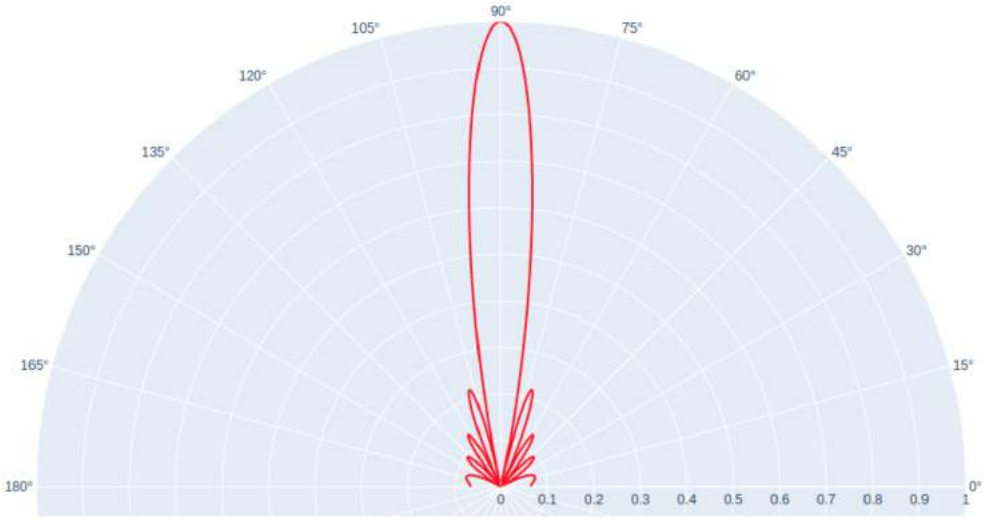
$$k = dcos(\theta). \tag{6}$$

Zpoždění závisí na k a rychlosti zvuku ve vzduchu c:

$$T_d = \frac{k}{c}. \tag{7}$$

Označíme například mikrofon M_0 jako referenční a signál z něj neupravujeme. Pro každý další mikrofon vypočteme zpoždění, ke kterému dojde při dopadu signálu z požadovaného úhlu θ, a určíme signál posunutý o toto zpoždění. Získáme tak ze všech mikrofonů stejný signál ve stejné fázi. Následně se provede součet referenčního signálu a všech posunutých signálů a výsledek se normalizuje vydělením počtem mikrofonů. Pokud signál dopadá z jiného než požadovaného úhlu, posunem signálů z mikrofonů o zpoždění vypočtené k požadovanému signálu získáme signály v různých fázích. Při následné sumaci a normalizaci dochází k destruktivní inferenci signálů. [37]

Na obrázku 14 je zobrazen příklad, jak může vypadat výsledek beamformingu, který je nastaven na propouštění signálů přicházejících z 90 °. Použito bylo 20 mikrofonů o vzdálenosti 8 centimetrů a signálem byla sinusoida o frekvenci 1000 Hz v rozmezí 0 až 180 °. Výsledek je vykreslen v polárních souřadnicích, vzdálenost od počátku udává zesílení získaného signálu.

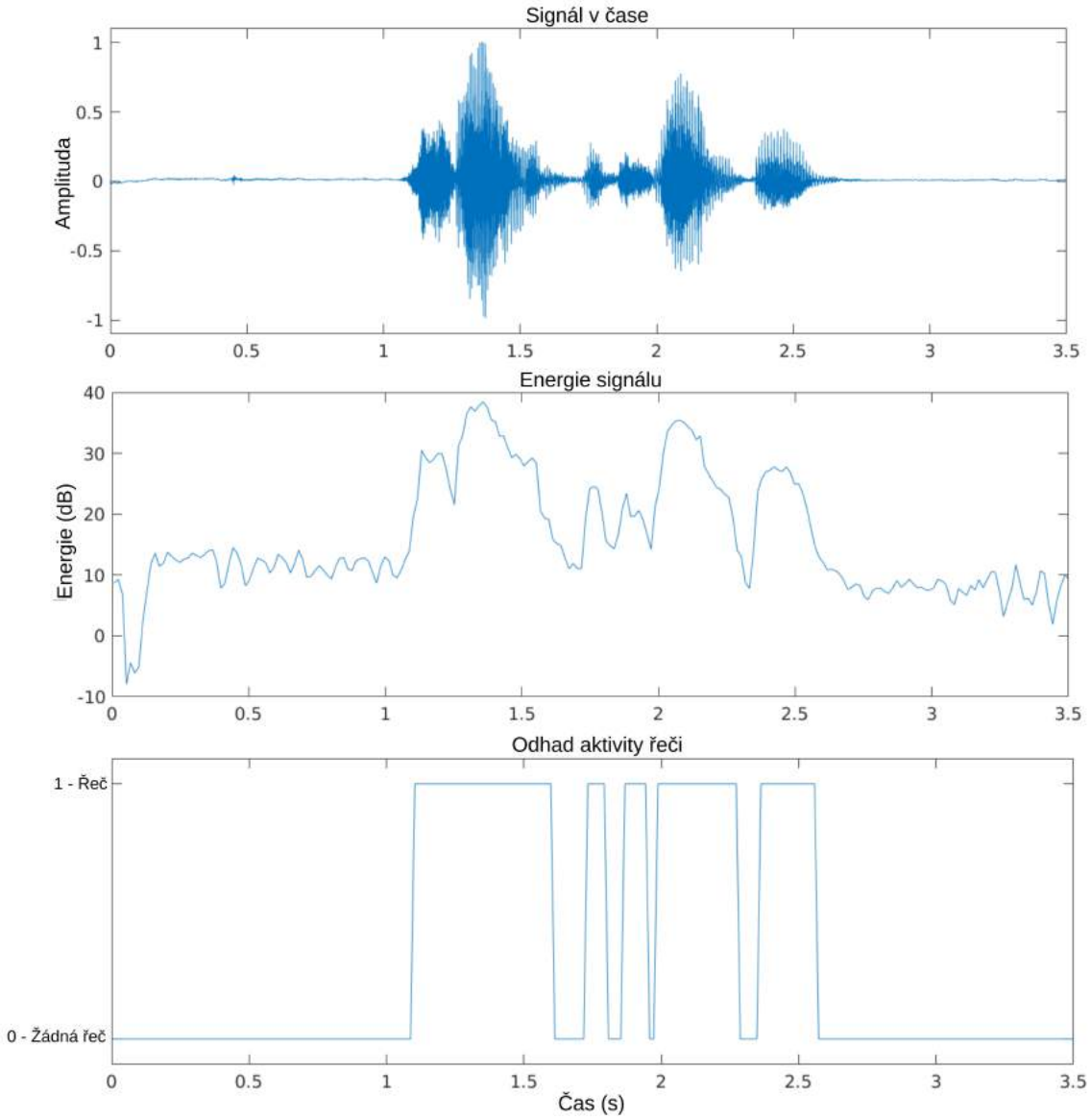


Obr. 14: Diagram citlivosti beamformingu [37]

4.2 Metody potlačení šumu v řečovém signálu

Pro potlačení šumu v řečovém signálu se nejčastěji využívají metody spektrálního odečítání, Wienerův filtr a neuronové sítě trénované k tomuto účelu. Jejich použití zvyšuje kvalitu signálu pro porozumění řeči lidmi, pro automatické rozpoznávání řeči je však jejich použití sporné. V některých případech můžou tyto metody výsledky ASR zlepšit, ale také na ně nemusí mít žádný vliv, nebo je můžou dokonce i zhoršit. Zhoršení může být způsobeno rozmazáním či deformací spektra řečového signálu, které může automatickému rozpoznávání řeči uškodit [38].

Většinou jsou metody založeny na adaptivním filtrování, což znamená, že se parametry filtru dle použitého algoritmu v čase mění. Odhady hluku v signálu se provádí v pauzách mezi úseky řeči. K detekci řeči se používá detektor řečové aktivity (VAD), který určuje, kdy je v signálu přítomna řeč a kdy pouze hluk. Detektor by měl dosahovat vysoké přesnosti i při nízkých hodnotách SNR, aby nedocházelo k znehodnocení řečových úseků v signálu. Detektor může být založen například na sledování velkých změn energie v mikrosegmentech signálu. [39] Detekce řeči v signálu je zobrazena na obrázku 15.



Obr. 15: Detekce řečové aktivity, upraveno z [40]

4.2.1 Spektrální odečítání

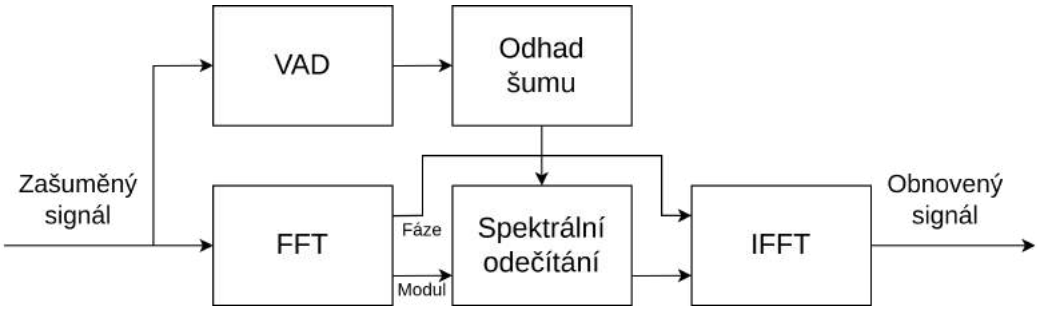
Spektrální odečítání bylo jednou z prvních technik odstraňování šumu v oblasti zpracování řečových signálů. Založeno je na myšlence odečtení šumu ze zašuměného signálu. Metoda předpokládá stacionární charakter šumu a nekorelovanost šumu s řečovým signálem [41]. Je výpočetně nenáročná a i přes svou jednoduchost velmi účinná. Principem spektrálního odečítání je odečtení spektra šumu od spektra řečového signálu.

Rychlou Fourierovou transformací (FFT) je vypočteno spektrum vstupního signálu. Od spektra vstupního signálu je odečteno spektrum hluku, jehož odhad se provádí v úsecích signálu bez přítomnosti řeči, čímž je získán odhad spektra nezašuměného signálu. Výpočet je popsán v rovnici 8, kde $\left|\hat{S}(\omega)\right|$ představuje výkonovou

spektrální hustotu obnoveného řečového signálu, $|Y(\omega)|$ výkonovou spektrální hustotu zašuměné řeči a $\left|\hat{D}(\omega)\right|$ odhad výkonové hustoty šumu. Zpětnou Fourierovou transformací (IFFT) odhadnutého spektra je následně vypočten časový průběh nezašuměného signálu. [39]

$$\left|\hat{S}(\omega)\right|^2 = |Y(\omega)|^2 - \left|\hat{D}(\omega)\right|^2 \tag{8}$$

Nedostatkem spektrálního odečítání je vznik tzv. hudebního šumu, ke kterému může dojít při nesprávném odhadu spektra hluku. V odhadnutém spektru pro obnovu řečového signálu pak vznikají nové spektrální složky, které se v signálu v časové oblasti projevují jako hudební tóny [41]. Schéma metody spektrálního odečítání je zobrazeno na obrázku 16.



Obr. 16: Schéma spektrálního odečítání, upraveno z [39]

4.2.2 Wienerův filtr

Principem Wienerova filtru je odhadnout čistý signál ze signálu zaneseného šumem. Založen je na minimalizaci střední kvadratické chyby odhadu. Filtr předpokládá stacionaritu a nekorelovanost šumu a řečového signálu. Přenosová funkce filtru ve frekvenční oblasti je:

$$H(\omega) = \frac{P_S(\omega)}{P_S(\omega) + P_V(\omega)}, \tag{9}$$

kde P_S představuje výkonové spektrum čistého signálu a P_V výkonové spektrum šumu. Nutné je předem znát hodnotu SNR, která je rovna:

$$SNR = \frac{P_S(\omega)}{P_V(\omega)}. \tag{10}$$

Dosazením rovnice 10 do rovnice 9 získáme přenosovou funkci ve formě [42]:

$$H(\omega) = \left[1 + \frac{1}{SNR}\right]^{-1}. \tag{11}$$

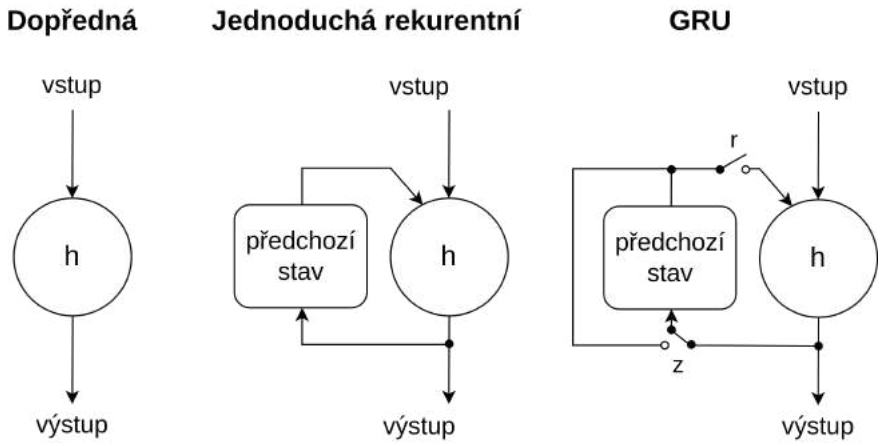
Adaptivní Wienerův filter v čase upravuje hodnoty SNR.

4.2.3 Neuronové sítě

Další z přístupů redukce šumu z řeči je založen na myšlence kombinace klasických přístupů zpracování signálů a hlubokých neuronových sítí. Lze takto vytvořit modely s vysokou rychlostí inference umožnující i použití v reálném čase, které jsou zároveň dostatečně malé, a tak pro výpočty není potřeba grafických karet.

Pro zpracování audio signálů je nejčastěji využíváno rekurentních neuronových sítí, které jsou navrženy pro zpracování sekvenčních dat namísto přiřazování výstupů ke vstupům po separátních nezávislých rámcích. Oproti dopředným neuronovým sítím obsahují navíc zpětný spoj. Výstup RNN v daném časovém kroku tedy závisí na aktuálním vstupu i na stavu z předchozího časového kroku. RNN byly dlouhou dobu limitovány neschopností udržet informace delší dobu a problémem mizejícího gradientu při trénování algoritmem zpětného šíření chyby. Tyto problémy byly potlačeny zavedením hradlových mechanismů, které síti pomáhají se efektivněji učit a udržovat dlouhodobé závislosti. Typickými sítěmi s hradlovými mechanismy jsou LSTM (Long Short-Term Memory) a GRU (Gated Recurrent Unit). [43]

Porovnání jednotek dopředné sítě, jednoduché rekurentní sítě a GRU je zobrazeno na obrázku 17. U GRU sítě jsou hradlové jednotky označeny z (update gate) a r (reset gate).

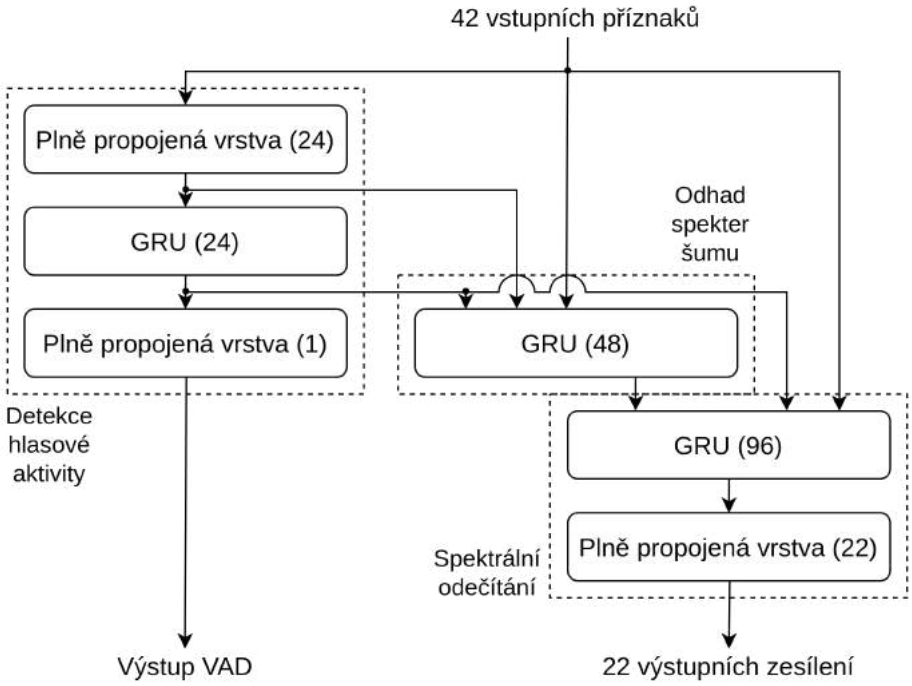


Obr. 17: Jednotky dopředné, jednoduché rekurentní a GRU sítě, upraveno z [43]

Nejznámější neuronová síť pro potlačení šumu z řečového signálu je RNNoise od společnosti Mozilla představená roku 2018 v práci [44]. Výstupem RNNoise jsou zesílení pro 22 frekvenčních pásem dle Barkovy stupnice, která jsou poté aplikována na vstupní signál. Vstupem do sítě je 22 Bark-frekvenčních kepstrálních koeficientů (BFCC), první a druhé derivace prvních šesti BFCC, perioda tónu, síla tónu a speci-

ální metrika nestacionarity pro pomoc detekce hlasové aktivity. Celkem je na vstupu 42 příznaků. [43]

Trénovací data RNNoise byla vytvořena zašuměním čistých řečových signálů. Jako zdroje hluku byly použity počítačové ventilátory, hluk z kanceláří, aut, letadel, vlaků, staveb a z davu lidí. [44] Architektura RNNoise je zobrazena na obrázku 18, u jednotlivých vrstev jsou v závorkách uvedeny počty neuronů těchto vrstev.



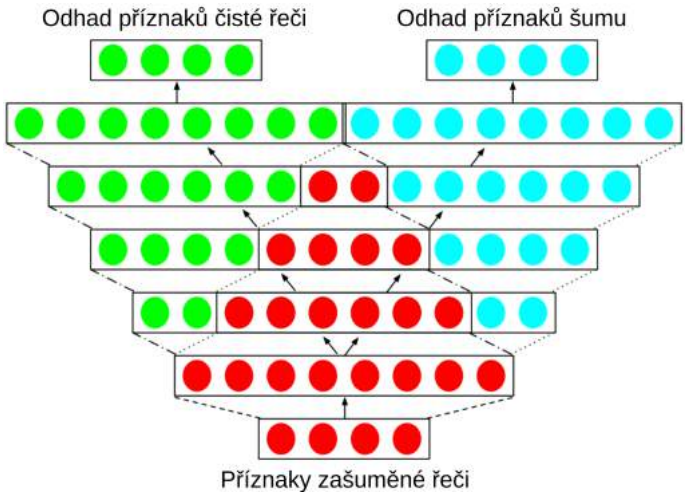
Obr. 18: Architektura RNNoise, upraveno z [44]

4.3 Úpravy jazykových modelů

Jak již bylo zmíněno v kapitole 3.3.2, trénování modelů pouze na čisté a srozumitelné řeči není příliš vhodné pro použití k rozpoznávání řeči s hlukem z okolí. Mnohem robustnější jsou vůči hluku modely trénované i na zašuměných datech. Pokud má být model používán v konkrétním prostředí s předem známým hlukem, je možné ke zvýšení kvality ASR model natrénovat na již zašuměných datech. Zašuměná data můžou být vytvořena buď uměle kombinací čisté řeči s hlukem, nebo mohou být přímo nasbírána z daného prostředí, což by ale bylo dosti časově náročné. Takový trénovací postup modelů se nazývá noise-aware training.

Zvýšení robustnosti modelu vůči hluku je dále možné pomocí multi-task trénování, při kterém je ASR model trénován spolu s klasifikátorem typu hluku, který poté může sloužit k odhlučnění řeči [45, 46]. Navržen byl i multi-task autoenkodér,

který z příznaků zašuměné řeči odhaduje zvlášť příznaky čisté řeči a příznaky šumu (viz obr. 19) [47].



Obr. 19: Multi-task autoenkodér, upraveno z [47]

I změna architektury modelu může změnit jeho robustnost vůči hluku. Výrazného zlepšení bylo po inspiraci ze zpracování obrazu dosaženo například pomocí použití vysokého počtu konvolučních vrstev neuronové sítě v modelu [48]. Pro lepší odolnost end-to-end modelů vůči hluku je vhodné využít i dlouhodobých závislostí v signálech pomocí rekurentních neuronových sítí s LSTM. V [49] bylo lepších výsledků ASR na zašuměné řeči dosaženo kombinací GRU, LSTM a jednoduchých rekurentních jednotek (SRU), které byly navíc pro zefektivnění výpočtů paralelizovány.

5 NÁVRH VLASTNÍHO SYSTÉMU

Vlastní systém navržený v práci má být schopen rozpoznávat předem definované ovládací příkazy i v hlučných prostředích. Uvažováno je umístění více hlasem ovládaných pracovišť vedle sebe, má být tedy dosaženo i rozlišení ovládání těchto pracovišť, aby nedošlo k záměně ovládacích příkazů mezi různými pracovišti. Dále v této kapitole bude popsán návrh jednotlivých částí systému.

Systém byl navržen pro ovládání v anglickém jazyce a předpokládá se použití headsetu, aby kromě mikrofonu pro snímání hlasu mohlo být využito i audio odezvy.

Testování navrženého systému bylo provedeno v prostředí robotického pracoviště OpenTube2, kde hlavním zdrojem hluku je odvětrávání laboratorního boxu. Měření intenzity hluku pracoviště bylo provedeno v místě, ve kterém by se při práci nacházel mikrofon operátora pracoviště, pomocí hlukoměru Sauter SU 130. Měření bylo provedeno v režimu Slow, který pro vyhodnocování využívá klouzavého průměru, a tedy lépe zachytí reálné hodnoty. Hlukoměr má tři režimy snímání. Režim A má stejnou citlivost jako lidské ucho, režim C je citlivější na hlučné okolní podmínky, jako jsou například stroje a motory, a režim F je vhodný pro oblasti s konstantní intenzitou hluku. V režimu A byla naměřena intenzita 64,0 dB, v režimu C 70,4 dB a v režimu F 72,2 dB. Pro představu intenzita hluku televizoru při běžné hlasitosti je 55 dB, hlasitý hovor dosahuje intenzity 60 dB, 70 dB je intenzita hluku silně frekventované ulice či strojovny a intenzita 80 dB je v tunelu metra [50].

Dále byl pro testování použit headset Jabra Evolve 20 MS. S pomocí headsetu byla vytvořena 30 s dlouhá nahrávka, která zachycuje vyslovení 4 příkazů a následně je snímán pouze zvuk pracoviště. Z nahrávky byla pomocí níže uvedené funkce stanovena hodnota SNR na 29,2 dB. K načtení audio nahrávky byla použita knihovna librosa.

Výpočet SNR z audio nahrávky

```
def SNR_dB(signal_arr, axis=0, ddof=0):
    signal_arr = np.asanyarray(signal_arr)
    m = signal_arr.mean(axis)
    sd = signal_arr.std(axis=axis, ddof=ddof)
    SNR_dB = -20*np.log10(abs(np.where(sd == 0, 0, m/sd)))
    return SNR_dB

signal, _ = librosa.load('/path/to/audio.wav')
SNR = SNR_dB(signal)
```

5.1 Model pro vzbouzecí frázi

První částí systému je model zachytávající pouze jednu frázi, a to frázi vzbouzecí, po které systém teprve začíná „poslouchat“ příkazy. Není totiž žádoucí, aby model pro rozpoznávání řeči nepřetržitě přepisoval vše, co slyší, a i výpočetně by byl tento přístup velmi náročný. Modely pro odchytávání jedné fráze jsou oproti ASR modelům výrazně menší a mohou být použity i v malých koncových zařízeních s takřka okamžitou inferencí. Pokud by navíc rozpoznávání příkazů pomocí ASR probíhalo online, je pomocí vzbouzecí fráze dosaženo i vyššího zabezpečení dat, jelikož nebude odesíláno a „odposloucháváno“ vše, co mikrofon nepřetržitě snímá.

Použití různých vzbouzecích frází pro jednotlivá pracoviště je také nejjednodušší a pravděpodobně nejefektivnější způsob pro odlišení ovládaných pracovišť. Pro odlišení ovládaných pracovišť by mohlo být pracováno i s rozlišením operátorů na základě jejich hlasů (tzv. speaker verification). To by ale znamenalo sestavování databáze operátorů a nahrávek jejich hlasů, trénování modelu pro jejich rozpoznání a i přihlašování operátora ke konkrétnímu pracovišti před zahájením práce. Tento přístup by byl velmi komplikovaný, nepraktický, pokud by měl pracoviště ovládat člověk, kterého systém zatím nezná, a také by mohl být nepřesný při změnách hlasu operátorů například při nemoci či použití ochranných roušek. V práci je tedy pro rozlišení pracovišť uvažováno použití odlišných vzbouzecích frází.

5.1.1 Vhodné vzbouzecí fráze

Základní vlastností dobré vzbouzecí fráze (nazývána také „wake word“ nebo „hotword“) je, aby se tato fráze nevyskytovala v běžné řeči. Cílem je, aby model ignoroval běžnou řeč, vhodné tedy může být i volit frázi, která není ani foneticky podobná frázím často využívaným během řeči. Doporučuje se spíše použití spojení více slov než slova samostatného, čímž lze snadno dosáhnout požadované neobvyklosti vzbouzecí fráze. Na druhou stranu by ale kvůli pohodlnosti použití vzbouzecí fráze neměla být ani příliš dlouhá. [51]

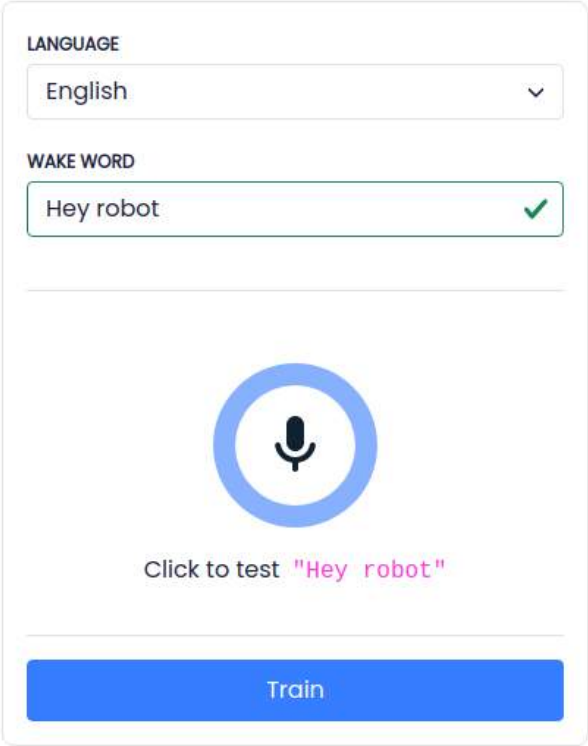
V angličtině je typické použití slov „Hey“ a „OK“ ve spojení se jménem ovládaného systému, které lze znát například z oslovení „OK Google“ a „Hey Siri“. V práci byla pro probuzení systému vybrána fráze „Hey robot“. Pro rozlišení jednotlivých pracovišť lze volit například různá jména pracovišť nebo na konec frází přidat čísla.

5.1.2 Porcupine

K detekci vzbouzecích frází byl společností Picovoice vyvinut engine Porcupine. Společnost uvádí, že tento engine dosahuje nejlepších výsledků a zároveň nejnižší zátěže CPU v porovnání s jinými dostupnými řešeními, jako jsou Snowboy či PocketSphinx. Pro zlepšení kvality modelu při použití v reálných prostředích jsou trénovací data

zašuměna na hodnotu SNR 10 dB. K zašumění je použit dataset DEMAND, který obsahuje nahrávky 18 různých zdrojů hluků. K rozlišení ostatní řeči od dané vzbouzecí fráze je při trénování využito datasetu LibriSpeech. [52]

Porcupine nabízí možnost trénování modelů pro vlastní fráze v 17 jazycích pomocí konzole na svých webových stránkách. Pro vytvoření modelu stačí zvolit jazyk, zadat frázi a následně zvolit platformu, na které bude model spuštěn. Dostupné jsou platformy Android, iOS, Linux, macOS, Windows, Raspberry Pi a další. Zdarma je možné si natrénovat tři modely měsíčně, pro častější použití je možno využít různých zpoplatněných členství. Trénovací konzole Porcupine je zobrazena na obrázku 20.



Obr. 20: Trénovací konzole Porcupine

Model je uložen ve formátu *.ppn*. Pro inferenci je třeba znát přístupový klíč, který je uživateli přidělen při registraci na stránkách Picovoice, a cestu k uloženému modelu. Registrace je nutná již pro samotné trénování modelu. V rámci jednoho programu lze použít více modelů, tedy reagovat na více frází. Pro snímání mikrofonu byla v práci použita knihovna pyaudio a dále je uvedena ukázka kódu pro inferenci Porcupine modelu.

Inference Porcupine modelu

```
engine = pvporcupine.create(
    access_key = 'my_access_key',
    keyword_paths=['/path/to/hey-robot_en_linux_v2_1_0.ppn']
)

p = pyaudio.PyAudio()

stream = p.open(
    format=pyaudio.paInt16,
    channels=1,
    rate=engine.sample_rate,
    input=True,
    frames_per_buffer=engine.sample_rate
)

while True:
    pcm = stream.read(engine.frame_length)
    pcm = struct.unpack_from('h' * engine.frame_length, pcm)

    keyword_idx = engine.process(pcm)
    if keyword_idx == 0:
        print('fráze "Hey robot" rozpoznána')

porcupine.delete()
```

5.1.3 Trénování vlastních modelů

Testovány byly i vlastní trénované modely. Využito bylo datasetu Multilingual Spoken Words Corpus, který obsahuje krátké nahrávky slov zvlášť roztříděné do adresářů. Dataset je vytvořen z datasetu Common Voice, který obsahuje audio nahrávky řeči s jejich přepisy. Dále byl při tvorbě Multilingual Spoken Words Corpus datasetu použit Montreal Forced Aligner, který každé nahrávce z Common Voice přiřazuje časové úseky pro jednotlivá slova v nahrávce. Slova jsou následně dle časových úseků vystřižena a roztříděna do samostatných adresářů. [53]

K trénování modelu byla třída pro vzbouzecí frázi *hey robot* vytvořena spojením a různými kombinacemi slov *hey* a *robot* z datasetu. Dále trénovací dataset obsahoval třídu pro hluk z okolí a třídu ostatní řeči, která byla vytvořena náhodnými výstřižky z datasetu Common Voice. Hluk z okolí byl vytvořen nastříháním

nahrávek třídy *_background_noise_* z datasetu Google Speech Commands. Za cílem dosažení vyváženého datasetu byla třída *hey robot* převzorkována.

Jelikož trénovaný model často dosahoval falešně pozitivních výsledků, byla dále přidána třída *x_robot* tvořená spojením náhodného slova či útržku řeči se slovem *robot* za cílem lepšího odlišení fráze *hey robot* od zbytku řeči. K mírnému zlepšení výsledků modelu došlo, i přes to je ale model Porcupine výrazně spolehlivější, a bude tak použit v celkovém systému.

Dále jsou uvedeny části kódu pro trénování modelu. [54]

Načtení datasetu

```
data_dir = pathlib.Path('path/to/dataset')

train_ds, val_ds = tf.keras.utils.audio_dataset_from_directory(
    directory=data_dir,
    batch_size=64,
    validation_split=0.2,
    seed=0,
    output_sequence_length=16000,
    subset='both'
)

def squeeze(audio, labels):
  audio = tf.squeeze(audio, axis=-1)
  return audio, labels

train_ds = train_ds.map(squeeze, tf.data.AUTOTUNE)
val_ds = val_ds.map(squeeze, tf.data.AUTOTUNE)

test_ds = val_ds.shard(num_shards=2, index=0)
val_ds = val_ds.shard(num_shards=2, index=1)
```

Vytvoření spektrogramů datasetů

```
def get_spectrogram(waveform):
  spectrogram = tf.signal.stft(waveform,
                               frame_length=255,
                               frame_step=128)
  spectrogram = tf.abs(spectrogram)
  spectrogram = spectrogram[..., tf.newaxis]
  return spectrogram
```

```
def make_spectrogram_ds(ds):
  return ds.map(
    map_func=lambda audio,label: (get_spectrogram(audio), label),
    num_parallel_calls=tf.data.AUTOTUNE)

train_spectrogram_ds = make_spectrogram_ds(train_ds)
val_spectrogram_ds = make_spectrogram_ds(val_ds)
test_spectrogram_ds = make_spectrogram_ds(test_ds)
```

Architektura modelu

```
norm_layer = layers.Normalization()
norm_layer.adapt(data=train_spectrogram_ds.map(
                map_func=lambda spec, label: spec))

model = models.Sequential([
    layers.Input(shape=input_shape),
    layers.Resizing(32, 32),
    norm_layer,
    layers.Conv2D(32, 3, activation='relu'),
    layers.Conv2D(64, 3, activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.25),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_labels),
])
```

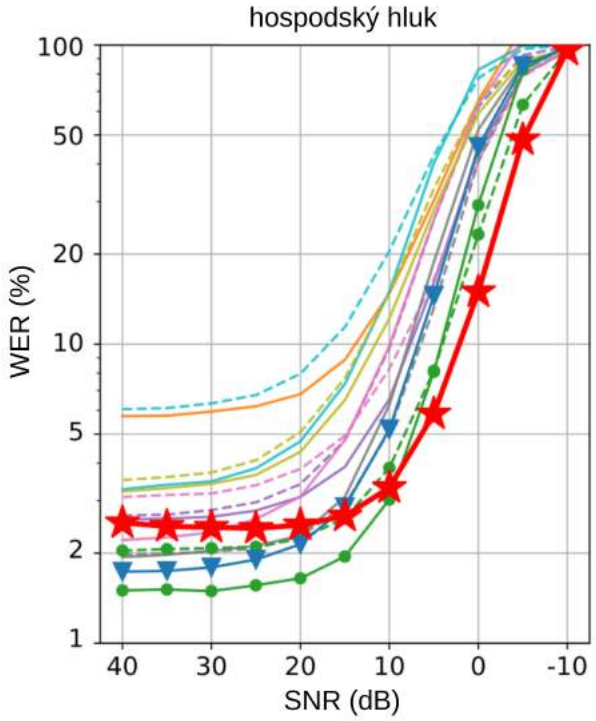
Konfigurace a trénování modelu

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(),
    loss=tf.keras.losses.SparseCategoricalCrossentropy(
                                        from_logits=True),
    metrics=['accuracy']
)
```

```
model.fit(
    train_spectrogram_ds,
    validation_data=val_spectrogram_ds,
    epochs=20,
    callbacks=tf.keras.callbacks.EarlyStopping(verbose=1,
                                               patience=2)
)
```

5.2 Model pro rozpoznávání řeči

Na základě hodnocení modelů v kapitole 3.3.4 byl dle nejvyšší přesnosti přepisu a diverzity trénovacích dat pro rozpoznávání příkazů vybrán model Whisper. Jelikož již je stanovena hodnota SNR v našem testovacím prostředí na 29,2 dB, lze určit předpokládanou přesnost modelu. V [33] bylo provedeno testování modelu Whisper v porovnání s jinými modely na LibriSpeech datasetu zašuměném hospodským hlukem o různých hodnotách SNR. Výsledky jsou zobrazeny na obrázku 21, model Whisper je zobrazen červenou čárou s hvězdičkami. Lze vidět, že pro SNR 29,2 dB nedochází ke zhoršení WER a i při ještě intenzivnějším hluku, tedy nižších hodnotách SNR, zůstávají výsledky ještě dlouhou dobu poměrně stabilní.



Obr. 21: WER modelu Whisperu o různých SNR, upraveno z [33]

Byť byl do práce díky nejlepším výsledkům vybrán model v anglické verzi, tedy verzi *English*, která byla trénována pouze na anglických datech, lze pomocí finetunování modelu na český dataset dosáhnout velmi dobrých výsledků i pro češtinu. Pokud porovnáme multilingvální verzi modelu přepisující anglický jazyk a model finetunovaný na češtinu, dosahují téměř shodných výsledků. V evaluaci různých velikostí multilingválního modelu na datasetu Common Voice 9.0 dosahuje model small WER 34,1 % pro češtinu a 14,5 % pro angličtinu, verze medium 18,8 % pro češtinu a 11,2 % pro angličtinu a verze large 17,1 % pro češtinu a 10,1 % pro angličtinu [33]. Po finetunování modelů na český dataset Common Voice 11.0 dosahují tyto modely při testování na tomto datasetu WER 18,5 % ve verzi small, 11,4 % ve verzi medium a 10,8 % ve verzi large [55, 56, 57].

Inference modelu Whisper může v systému hlasového ovládání probíhat buď na vzdáleném serveru, nebo přímo v koncovém zařízení. Pro vyšší kompaktnost systému a vyšší bezpečnost, jak již bylo dříve v práci zmíněno, je preferována inference přímo v zařízení. Jelikož jsou modely Whisper z hlediska paměťové náročnosti modely poměrně velké, byla do celkového systému provedena kvantizace modelu.

5.2.1 Kvantizace modelu

Kvantizace je technika redukce výpočetní a paměťové náročnosti při inferenci modelu pomocí převedení vah a aktivačních funkcí modelu z obvyklých 32 bitových floatů na méně precizní 8 bitové integery. Snížení počtu bitů vede k nižší spotřebě paměti, menší spotřebě energie a mnohem rychlejším maticovým výpočtům. [58]

Ke kvantizaci modelu bylo využito knihovny TensorFlow Lite, pomocí které lze TensorFlow modely jednoduše konvertovat do kvantizovaného formátu *.tflite*. Model Whisper je ve formátu TensorFlow dostupný pomocí *transformers* API platformy Hugging Face jako *TFWhisperModel*. Kód pro konvertování uloženého TF modelu do formátu *.tflite* je ukázán dále.

Generování tflite modelu

```
class GenerateModel(tf.Module):
  def __init__(self, model):
    super(GenerateModel, self).__init__()
    self.model = model

  @tf.function(
    input_signature=[
      tf.TensorSpec((1, 80, 3000), tf.float32,
                    name='input_features'),
    ],
  )

  def serving(self, input_features):
    outputs = self.model.generate(
      input_features,
      max_new_tokens=223,
      return_dict_in_generate=True,
    )
    return {'sequences': outputs['sequences']}

tf_model_dir = '/path/to/tf_whisper'
tflite_model_path = 'path/to/whisper.tflite'

generate_model = GenerateModel(model=model)
tf.saved_model.save(generate_model, tf_model_dir,
        signatures={'serving_default': generate_model.serving})

converter = tf.lite.TFLiteConverter.from_saved_model(tf_model_dir)
converter.target_spec.supported_ops = [
  tf.lite.OpsSet.TFLITE_BUILTINS,
  tf.lite.OpsSet.SELECT_TF_OPS]

converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()

with open(tflite_model_path, 'wb') as f:
    f.write(tflite_model)
```

V tabulce 2 jsou porovnány velikosti různých verzí modelů v původním *.h5* a kvantizovaném *.tflite* formátu.

Tab. 2: Velikosti modelů před a po kvantizaci v MB

	tiny-en	**base-en**	**small-en**	**medium-en**	**large**
.h5	151,0	291,0	967,0	3060,0	6170,0
.tflite	40,9	76,9	247,9	774,8	1450,0

K inferenci je kromě samotného modelu nutno použít WhisperProcessor pro konkrétní verzi modelu. WhisperProcessor v sobě kombinuje funkce tokenizéru (WhisperTokenizer) a extraktoru příznaků (WhisperFeatureExtractor) modelu. Dostupný je opět pomocí *transformers* API, ale může být i stažen a uložen v zařízení, čehož bylo v práci využito. Model použitý do celkového systému byl vybrán ve verzi tiny-en, tedy i WhisperProcessor byl použit v této verzi. Dále je uveden kód pro inferenci kvantizovaného modelu.

Inference tflite Whisper modelu

```
processor_path = '/path/to/whisper_processors/tiny-en'
tflite_model_path = '/path/to/tflite_models/tiny-en.tflite'

processor = WhisperProcessor.from_pretrained(processor_path)
interpreter = tf.lite.Interpreter(tflite_model_path)
tflite_generate = interpreter.get_signature_runner()

def transcribe(audio):
    input_features = processor(
                            [audio],
                            sampling_rate=16000,
                            return_tensors='tf').input_features
    generated_ids = tflite_generate(
                    input_features=input_features)['sequences']
    transcription = processor.batch_decode(
                                    generated_ids,
                                    skip_special_tokens=True)[0]
    return transcription

transcription = transcribe(audio_array)
```

5.3 Potlačení hluku

Na základě rešerše bylo jako nejvhodnější řešení zlepšení kvality ASR v hlučných prostředích vybráno použití metody beamforming, která aktivně potlačuje zvukové signály z jiných než požadovaných směrů, a volba dostatečně robustního ASR modelu.

Do systému hlasového ovládání je tedy doporučeno volit headset s více zabudovanými mikrofony a již implementovaným beamformingem. Headset použitý v práci pro testování ve svých specifikacích uvádí pouze užití state-of-the-art řešení pro potlačení hluku mikrofonu [59], ale například u jiných modelů značky Jabra je již uvedena metoda beamforming [60]. I z hodnoty SNR nahrávky vytvořené pomocí headsetu, která je rovna 29,2 dB, lze poznat, že je okolní hluk velmi dobře potlačen. I subjektivně je řeč v nahrávce zřetelná a dobře srozumitelná a hluk z okolí je slyšet pouze jako nepatrný šum.

Robustnost použitých modelů Porcupine a Whisper vůči hluku již byly rozebrány v předchozích kapitolách.

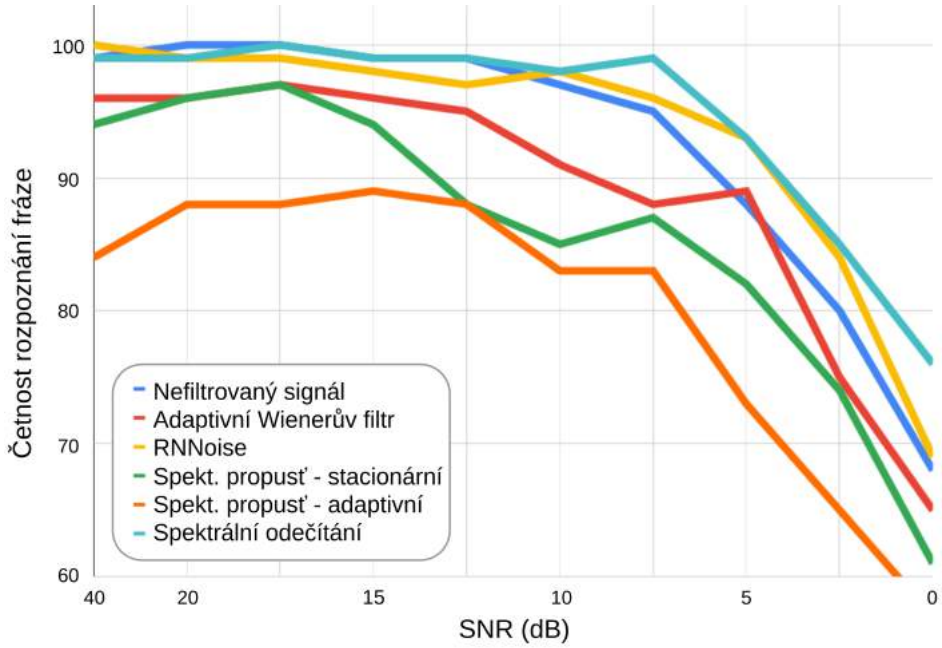
Jak bylo zmíněno v kapitole 4.2, různé filtrační metody pro odstranění hluku z řečového signálu mohou mít na výsledky ASR pozitivní i negativní vliv. Vybrané metody byly na modelech Porcupine a Whisper testovány.

Testována byla neuronová síť RNNoise [61], metoda spektrálního odečítání, adaptivní Wienerův filtr a spektrální propusť. Metoda spektrálního odečítání a Wienerův filtr byly testovány za využití knihovny Pyroomacoustics, která se zaměřuje na implementace známých algoritmů zpracování signálů [62, 63]. K testování spektrální propusti bylo využito knihovny noisereduce, která se zabývá potlačením šumu v signálech v časové oblasti [64]. Testována byla stacionární i adaptivní verze tohoto algoritmu.

Testování bylo provedeno umělým zašuměním řečových nahrávek a následnou aplikovací vybraných metod pro potlačení šumu na tyto nahrávky.

5.3.1 Testování na modelu Porcupine

Pro testování vlivu metod na efektivitu modelu Porcupine bylo vytvořeno 100 nahrávek vzbouzecí fráze *hey robot*, které byly následně zašuměny audio nahrávkou z výrobního prostředí na různé úrovně SNR. Vyhodnocována byla četnost rozpoznání frází po aplikaci jednotlivých metod při daných SNR. Výsledky jsou zobrazeny v grafu na obrázku 22. O hodnotě SNR 40 dB jsou vzorky nezašuměné.



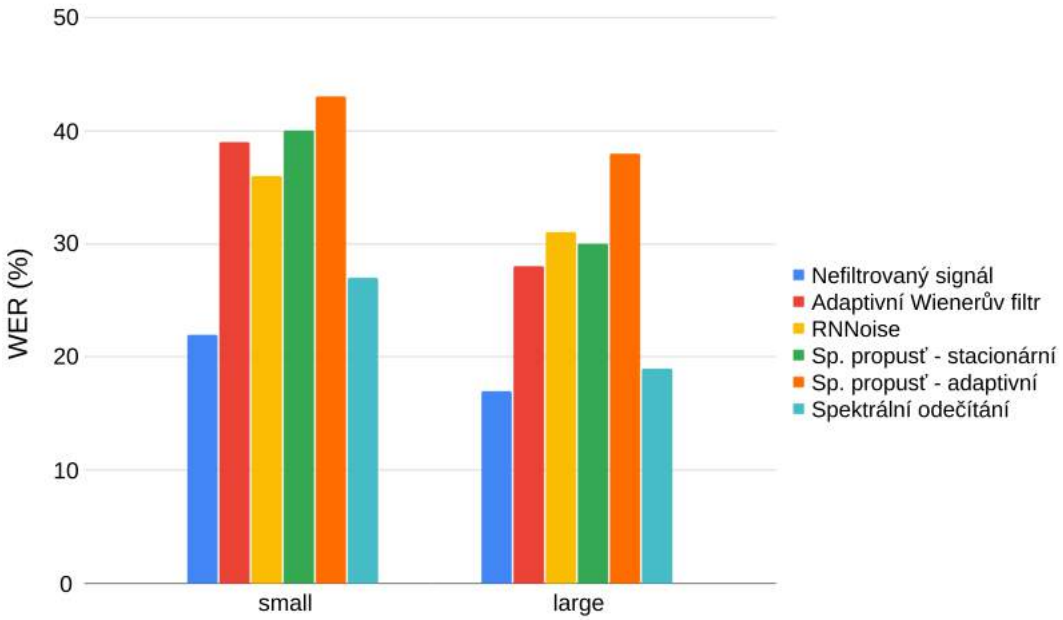
Obr. 22: Vyhodnocení vybraných metod na modelu Porcupine

Lze vidět, že zlepšení výsledků dosahuje až při vyšších intenzitách hluku metoda spektrálního odečítání. Filtrováním pomocí neuronové sítě RNNoise je také dosaženo mírného zlepšení při vyšších intenzitách hluku, při nižších ale dochází ke zhoršení. Ostatní metody schopnost rozpoznání fráze více či méně zhoršují. Do velmi hlučných prostředí, ve kterých řečové signály dosahují jen nízkých SNR, by tedy byla vhodná implementace metody spektrálního odečítání, případně by mohlo stát za zvážení i natrénování RNNoise na konkrétní hluk prostředí, díky čemuž by mohlo být dosaženo ještě lepších výsledků.

5.3.2 Testování na modelu Whisper

Pro testování na modelu Whisper bylo náhodně vybráno 3000 nahrávek datasetu Common Voice 11.0, které byly zašuměny stejnou audio nahrávkou jako v případě testování Porcupine. Zašumění tentokrát bylo provedeno pouze jednou, a to o hlasitosti, která v případě zašumění frází *hey robot* odpovídala hodnotě SNR 5 dB. Jelikož Common Voice obsahuje nahrávky řeči různě hlasité, či i mírně zašuměné, nelze hodnotu SNR jednoznačně stanovit.

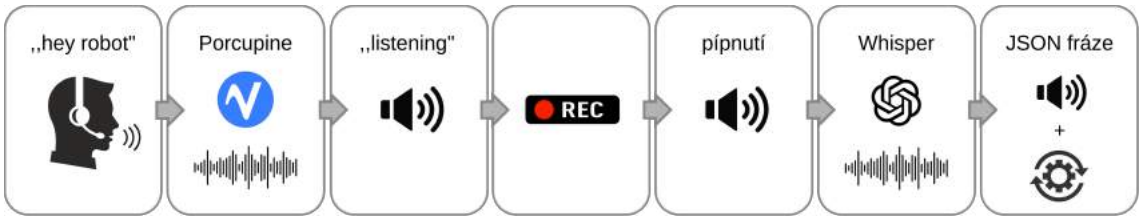
Vyhodnocováno bylo WER nahrávek odšuměných pomocí vybraných metod. Evaluace byla provedena na verzích modelu *small* a *large*. Výsledky jsou zobrazeny v grafu na obrázku 23, kde lze vidět, že pro model Whisper ani jedna z metod nepřináší zlepšení, a tedy pro rozpoznávání příkazů v práci nebylo žádné filtrování implementováno.



Obr. 23: Vyhodnocení vybraných metod na modelech Whisper

5.4 Celkový zhotovený systém

Schéma celého navrženého systému je zobrazeno na obrázku 24.



Obr. 24: Navržený systém hlasového ovládání

Operátor má headset s mikrofonem, který potlačuje hluk pomocí metody beamforming. Model Porcupine nepřetržitě vyhodnocuje signál z mikrofonu až do zachycení vzbouzecí fráze. Po rozpoznání fráze se operátorovi do sluchátek ozve „listening“, čímž systém operátora informuje, že frázi rozpoznal a má zadat příkaz. Následujících 2,5 s je nahráváno. Po uplynutí této doby je operátorovi konec nahrávání oznámen pípnutím do sluchátek. Nahrávka je dále zpracována modelem Whisper, z transkripce je odstraněna interpunkce, mezery a je přepsána malými písmeny, poté je vyhodnocena dle příkazů předem nadefinovaných v JSON souboru. Každý definovaný příkaz má přiřazenou frázi, která je opět uživateli přehrána po rozpoznání fráze, a výstupní skript, který je po rozpoznání fráze spuštěn. Výstupní skripty jsou uloženy ve vlastním adresáři *output_scripts*. Pro testovací potřeby práce

bylo v těchto výstupních skriptech provedeno pouze otevření obrázků s rozpoznanými frázemi, předpokládá se ale nahrazení skripty ovládajícími reálné zařízení. V případě, že je vyslovená fráze chybně rozpoznána, je operátorovi do sluchátek vyslovena fráze „Did not recognize the command.“

Dále je zobrazena ukázka zápisu příkazů a jejich výstupů v JSON souboru.

JSON soubor příkazů

```
{
    "start" : {
        "response" : "starting",
        "output" : "start.py"
    },
    "openthebox" : {
        "response" : "opening the box",
        "output" : "openthebox.py"
    },
    "putitdown" : {
        "response" : "putting it down",
        "output" : "putitdown.py"
    }
}
```

K audio odezvě do sluchátek bylo využito knihovny pyttsx3 zaměřující se na převod psaného textu na řeč („text-to-speech“). Zdrojové kódy celého systému jsou dostupné v příloze práce.

Systém byl testován na 100 příkazech, 85 z nich bylo úspěšně rozpoznáno, ve dvou případech nebyla zachycena vzbouzecí fráze a ve zbylých 13 případech byl vyslovený příkaz špatně rozpoznán a nebyl přiřazen žádné nadefinované frázi.

6 NÁVRH ÚPRAVY LABORATORNÍHO BOXU PRO AUTOMATICKÉ OTEVÍRÁNÍ

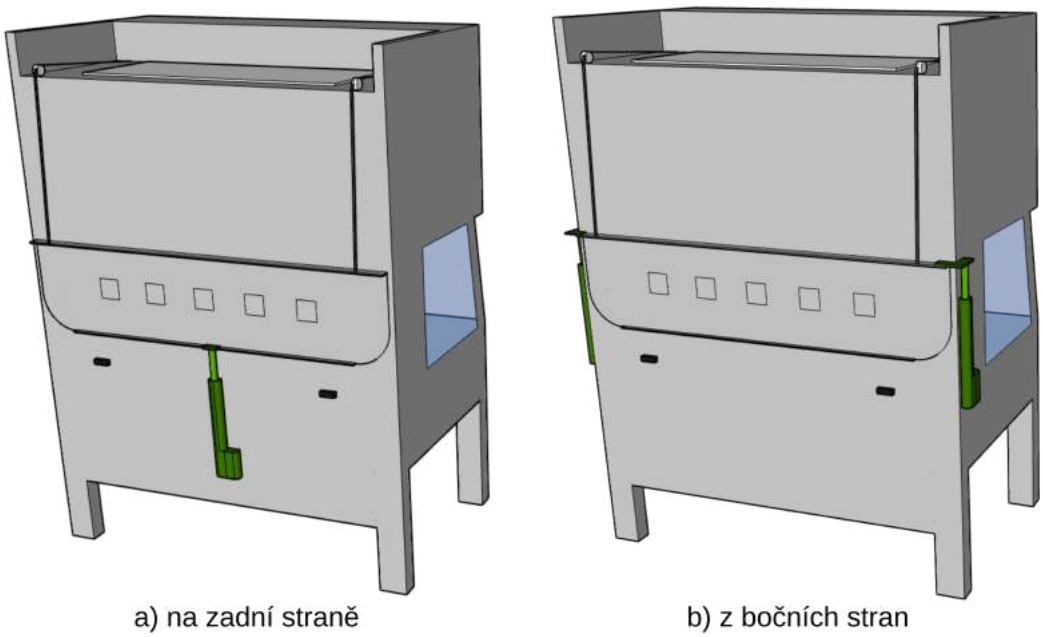
Po implementaci hlasového ovládání na pracoviště OpenTube2 by bylo přínosem přidání automatického otevírání laboratorního boxu pracoviště (viz obrázek 25). Pokud by totiž i otevírání boxu bylo ovládáno hlasem, nebyla by po operátorovi vyžadována desinfekce a výměna rukavic po každém otevření a zavření boxu při práci.

Obr. 25: Laboratorní box OpenTube2

Otevírání laboratorního boxu je realizováno skleněnými posuvnými dveřmi v přední části, které jsou lanky vedenými přes kladky na horní straně boxu připevněny k protizávaží na zadní straně. K tomuto protizávaží by mohly být připevněny lineární pohony a za přidání řídicí jednotky by bylo automatické otevírání implementováno. Zadní strana boxu bez podstavce je zachycena na obrázku 26.

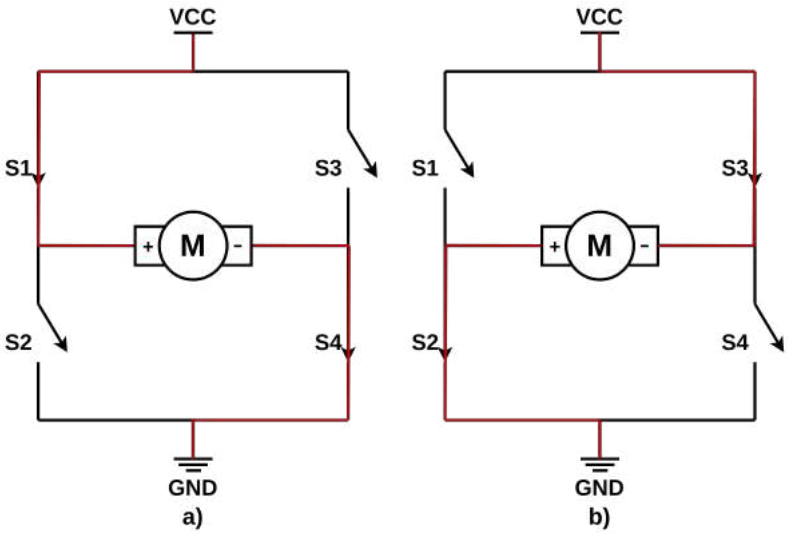
Obr. 26: Zadní strana laboratorního boxu

Podstavec laboratorního boxu je dostatečně vysoký, a tak by lineární pohon mohl být umístěn ze zadní strany podstavce. Zároveň jelikož z laboratorního boxu z bočních stran vedou kabely, a nepředpokládá se tedy, že by byl box ze stran k něčemu těsně umístěn, druhou možností je umístění dvou aktuátorů ze stran laboratorního boxu. Navržené možnosti přidání pohonů jsou představeny na obrázku 27, aktuátory jsou zobrazeny zeleně. Řídicí jednotka může být spolu s napájením pohonů přidána do boxu s ostatní elektronikou pracoviště.



Obr. 27: Možnosti přidání aktuátorů pro automatické otevírání

Pohybu lineárního pohonu oběma směry, tedy k otevírání i zavírání boxu, lze dosáhnout použitím H můstku, který umožňuje přepólování napájecího napětí, a tak způsobí pohyb motoru na opačnou stranu. Jedná se o elektrický obvod složený ze čtyř spínačů, které mohou být realizovány pomocí tranzistorů, či relé. Na obrázku 28 jsou zobrazeny dva stavy a) a b), ve kterých je motor pohonu zapojen s opačnou polaritou, tedy otáčí se opačným směrem. V případě, že by byly všechny spínače rozepnuty, proud obvodem neprochází. Zakázané stavy obvodu jsou při sepnutí všech spínačů, při sepnutí jakékoliv kombinace tří spínačů a při sepnutí spínačů S1 a S2, nebo S3 a S4, zdroj by byl ve zkratu.



Obr. 28: Stavy H můstku pro opačnou polaritu napájení

7 ZÁVĚR

Cílem této práce bylo sestavení systému hlasového ovládání do rušných prostředí včetně návrhu úpravy laboratorního boxu OpenTube2 pro automatické otevírání.

Po rešerši jazykových modelů rozpoznávání řeči byl vybrán model Porcupine pro detekci vzbouzecí fráze a model Whisper pro rozpoznávání předem definovaných příkazů. Model Whisper byl do systému kvantizován, čímž bylo dosaženo snížení paměťové a výpočetní náročnosti inference modelu, a byla tak umožněna offline inference v koncovém zařízení. Pro odlišení ovládání více pracovišť umístěných vedle sebe bylo navrženo natrénování Porcupine modelu na jinou vzbouzecí frázi pro každé pracoviště.

Na základě rešerše i po inspiraci již dostupnými řešeními byla jako nejefektivnější metoda potlačení hluku vybrána metoda beamforming, která použitím více nedaleko umístěných mikrofonů filtruje signály přijímané pouze z požadovaného směru. Dále bylo provedeno testování vlivu vybraných metod potlačení šumu z řečových signálů na efektivitu modelů Porcupine a Whisper. Testováním byl ale prokázán převážně negativní vliv těchto metod.

Navržený systém předpokládá použití headsetu s metodou beamforming pro potlačení hluku. Příkazy, které mají být systémem rozpoznány, jsou nadefinovány v JSON souboru spolu s frázemi, které jsou operátorovi přehrány jako audio odezva po rozpoznání příkazu, a se skripty, které jsou po rozpoznání příkazu spuštěny.

Systém byl úspěšně otestován v prostředí pracoviště OpenTube2, ve kterém je hlavním zdrojem hluku odvětrávání laboratorního boxu.

Pro laboratorní box OpenTube2 bylo pro možnost automatického otevírání navrženo připevnění lineárních aktuátorů k protizávaží otevíracích dveří, a to buď umístěním dvou pohonů ze stran boxu, nebo jedním pohonem ze zadní strany boxu.

V práci se dá pokračovat jednoznačně implementací navrženého hlasového ovládání na pracoviště OpenTube2 a může být realizováno i navržené automatické otevírání.

SEZNAM POUŽITÉ LITERATURY

[1] *EPG Hlasová řešení* [online]. [cit. 2023-04-22]. Dostupné z: https://www.epg.com/cs/hlasova-reseni.

[2] *Innovative voice picking system for intralogistics, production and quality assurance* [online]. [cit. 2023-04-22]. Dostupné z: https://www.lydia-voice.com/gb/.

[3] *Lydia VoiceWear* [online]. [cit. 2023-04-22]. Dostupné z: https://www.epg.com/gb/voice-solutions/voice-devices/voicewear.

[4] KVADOS.CZ. *Spojili jsme síly se společností Honeywell* [online]. [cit. 2023-04-22]. Dostupné z: https://www.kvados.cz/spojili-jsme-sily-se-spolecnosti-honeywell/.

[5] *Voice Picking Technology* [online]. [cit. 2023-04-22]. Dostupné z: https://sps.honeywell.com/us/en/software/productivity/order-picking-fulfillment/voice-picking.

[6] *Ayes Chytré brýle* [online]. [cit. 2023-04-22]. Dostupné z: https://www.ayes.cz/.

[7] *Vzdálená Spolupráce* [online]. [cit. 2023-04-22]. Dostupné z: https://www.ayes.cz/vzdalena-spoluprace/.

[8] *Hlasové ovládání* [online]. [cit. 2023-04-22]. Dostupné z: https://www.linak.cz/produkty/ovlada%C4%8De/voice-control/.

[9] *Athena works here.* [online]. [cit. 2023-04-22]. Dostupné z: https://athenaworkshere.com/.

[10] *Offline voice control* [online]. [cit. 2023-04-22]. Dostupné z: https://viccontrol.de/en/.

[11] *INDUSTRIAL VOICE CONTROL: spectra* [online]. [cit. 2023-04-23]. Dostupné z: https://www.spectra-austria.at/files/produkte/KA014821/web/spectra/Brochure-Industrial-Voice-Control.pdf.

[12] *Digi ConnectCore Voice Control* [online]. [cit. 2023-04-23]. Dostupné z: https://www.digi.com/solutions/by-technology/voice-control.

[13] *Hands full? No problem!* [online]. [cit. 2023-04-23]. Dostupné z: https://www.siemens.com/global/en/company/stories/industry/factory-automation/voiceassistant-automation-machinebuilding-fps.html.

[14] *Voice meets high tech: Controlling production plants by speech* [online]. [cit. 2023-04-23]. Dostupné z: https://careers.roche.com/global/en/voice-meets-high-tech-controlling-production-plants-by-speech.

[15] Wang, D., Wang, X. a Lv, S. An overview of end-to-end automatic speech recognition. *Symmetry*. MDPI. 2019, sv. 11, č. 8, s. 1018.

[16] Gales, M. a Young, S. The Application of Hidden Markov Models in Speech Recognition. *Foundations and Trends® in Signal Processing*. 2007, sv. 1, č. 3, s. 195–304. DOI: 10.1561/2000000004. ISSN 1932-8346. Dostupné z: http://www.nowpublishers.com/article/Details/SIG-004.

[17] Shrawankar, U. a Thakare, V. M. Techniques for feature extraction in speech recognition system: A comparative study. *ArXiv preprint arXiv:1305.1145*. 2013.

[18] Li, J. Recent Advances in End-to-End Automatic Speech Recognition. *APSIPA Transactions on Signal and Information Processing*. 2022, sv. 11, č. 1. DOI: 10.1561/116.00000050. ISSN 2048-7703. Dostupné z: http://www.nowpublishers.com/article/Details/SIP-2021-0050.

[19] Graves, A., Fernández, S., Gomez, F. a Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, s. 369–376.

[20] Rao, K., Sak, H. a Prabhavalkar, R. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In: IEEE. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2017, s. 193–199.

[21] Bahdanau, D., Cho, K. a Bengio, Y. Neural machine translation by jointly learning to align and translate. *ArXiv preprint arXiv:1409.0473*. 2014.

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. Attention is all you need. *Advances in neural information processing systems*. 2017, sv. 30.

[23] *How do Transformers work?* [online]. [cit. 2023-05-04]. Dostupné z: https://huggingface.co/learn/nlp-course/chapter1/4.

[24] *Julius-speech/julius: Release 4.5* [online]. 2019 [cit. 2023-05-05]. Dostupné z: https://doi.org/10.5281/zenodo.2530396.

[25] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G. et al. Deep speech: Scaling up end-to-end speech recognition. *ArXiv preprint arXiv:1412.5567*. 2014.

[26] *Deepspeech* [online]. [cit. 2023-05-05]. Dostupné z: `https://github.com/mozilla/DeepSpeech`.

[27] *History of the Kaldi project* [online]. [cit. 2023-05-05]. Dostupné z: `https://kaldi-asr.org/doc/history.html`.

[28] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O. et al. The Kaldi speech recognition toolkit. In: IEEE Signal Processing Society. *IEEE 2011 workshop on automatic speech recognition and understanding*. 2011, CONF.

[29] *Versions of Kaldi* [online]. [cit. 2023-05-05]. Dostupné z: `https://kaldi-asr.org/doc/versions.html`.

[30] *GigaSpeech ASR model* [online]. [cit. 2023-05-05]. Dostupné z: `http://kaldi-asr.org/models/m14`.

[31] Chen, G., Chai, S., Wang, G., Du, J., Zhang, W.-Q. et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *ArXiv preprint arXiv:2106.06909*. 2021.

[32] Baevski, A., Zhou, Y., Mohamed, A. a Auli, M. Wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*. 2020, sv. 33, s. 12449–12460.

[33] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C. et al. Robust speech recognition via large-scale weak supervision. *ArXiv preprint arXiv:2212.04356*. 2022.

[34] *Introducing Whisper* [online]. OpenAI [cit. 2023-05-06]. Dostupné z: `https://openai.com/research/whisper`.

[35] Seagraves, A. *Benchmarking Top Open Source Speech Recognition Models: Whisper, Facebook wav2vec2, and Kaldi* [online]. [cit. 2023-05-06]. Dostupné z: `https://blog.deepgram.com/benchmarking-top-open-source-speech-models/`.

[36] Li, D., Gao, Y., Zhu, C., Wang, Q. a Wang, R. Improving Speech Recognition Performance in Noisy Environments by Enhancing Lip Reading Accuracy. *Sensors*. MDPI. 2023, sv. 23, č. 4, s. 2053.

[37] Aleksanyan, H., Shmavonyan, H., Tonoyan, T. a Hovsepyan, A. *The Hard Side of Noise Reduction – Hardware Based Approach via Beamforming* [online]. [cit. 2023-05-11]. Dostupné z: `https://krisp.ai/blog/hardware-beamforming-noise-reduction/`.

[38] *The concept of noise reduction in speech recognition* [online]. [cit. 2023-05-12]. Dostupné z: `https://en.speechocean.com/Cy/531.html`.

[39] Poruba, J. a Matějíček, L. *Odfiltrování rušivých signálů ze zašumělé řeči* [online]. Ústav telekomunikací, FEKT, VUT Brno: [b.n.] [cit. 2023-05-12]. Dostupné z: `http://www.elektrorevue.cz/clanky/02047/index.html#[4]`.

[40] Bäckström, T. *Voice activity detection (VAD)* [online]. [cit. 2023-05-12]. Dostupné z: `https://wiki.aalto.fi/pages/viewpage.action?pageId=151500905`.

[41] Virtanen, T., Singh, R. a Raj, B. *Techniques for noise robustness in automatic speech recognition*. John Wiley & Sons, 2012.

[42] Padarti, V. K., Polavarapu, G. S., Madiraju, M., Naga Sai Nuthalapati, V., Thota, V. B. et al. A Study on Effectiveness of Deep Neural Networks for Speech Signal Enhancement in Comparison with Wiener Filtering Technique. In: *Advances in Speech and Music Technology: Computational Aspects and Applications*. Springer, 2022, s. 121–135.

[43] Valin, J.-M. *RNNoise: Learning Noise Suppression* [online]. Mozilla and Xiph.Org [cit. 2023-05-13]. Dostupné z: `https://jmvalin.ca/demo/rnnoise/`.

[44] Valin, J.-M. A hybrid DSP/deep learning approach to real-time full-band speech enhancement. In: IEEE. *2018 IEEE 20th international workshop on multimedia signal processing (MMSP)*. 2018, s. 1–5.

[45] Prasad, A., Jyothi, P. a Velmurugan, R. An investigation of end-to-end models for robust speech recognition. In: IEEE. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, s. 6893–6897.

[46] Pironkov, G., Dupont, S. a Dutoit, T. Multi-task learning for speech recognition: an overview. In: *ESANN*. 2016.

[47] Zhang, H., Liu, C., Inoue, N. a Shinoda, K. Multi-task autoencoder for noise-robust speech recognition. In: IEEE. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, s. 5599–5603.

[48] Qian, Y., Bi, M., Tan, T. a Yu, K. Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. IEEE. 2016, sv. 24, č. 12, s. 2263–2276.

[49] Ullah, R., Wuttisittikulkij, L., Chaudhary, S., Parnianifard, A., Shah, S. et al. End-to-End Deep Convolutional Recurrent Models for Noise Robust Waveform Speech Enhancement. *Sensors*. MDPI. 2022, sv. 22, č. 20, s. 7782.

[50] Bureš, J. *Příklady zvuků (intenzita hluku)* [online]. [cit. 2023-05-19]. Dostupné z: `http://www.converter.cz/tabulky/hluk.htm`.

[51] *Tips for Choosing a Wake Word* [online]. [cit. 2023-05-20]. Dostupné z: `https://picovoice.ai/docs/tips/choosing-a-wake-word/`.

[52] *Wake Word Benchmark* [online]. [cit. 2023-05-20]. Dostupné z: `https://picovoice.ai/docs/benchmark/wake-word/`.

[53] Mazumder, M., Chitlangia, S., Banbury, C., Kang, Y., Ciro, J. M. et al. Multilingual spoken words corpus. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[54] *Simple audio recognition: Recognizing keywords* [online]. [cit. 2023-05-20]. Dostupné z: `https://www.tensorflow.org/tutorials/audio/simple_audio`.

[55] *Whisper Small Czech CV11* [online]. [cit. 2023-05-20]. Dostupné z: `https://huggingface.co/mikr/whisper-small-cs-cv11`.

[56] *Whisper Medium Czech 2 CV11* [online]. [cit. 2023-05-20]. Dostupné z: `https://huggingface.co/mikr/whisper-medium-czech-cv11`.

[57] *Whisper Large Czech CV11* [online]. [cit. 2023-05-20]. Dostupné z: `https://huggingface.co/mikr/whisper-large-czech-cv11`.

[58] *Quantization* [online]. [cit. 2023-05-20]. Dostupné z: `https://huggingface.co/docs/optimum/concept_guides/quantization`.

[59] *Jabra Evolve 20 USB Stereo Headset* [online]. [cit. 2023-05-20]. Dostupné z: `https://www.headsetsdirect.com/product/jabra-evolve-20-usb-stereo-headset/`.

[60] *Jabra náhlavní souprava Evolve2 75 včetně stojánku, Link 380a MS, stereo, černá* [online]. [cit. 2023-05-20]. Dostupné z: `https://www.tonerpartner.cz/produkt-jabra-nahlavni-souprava-evolve2-75-vcetne-stojanku-link-380a-ms-stereo-cerna-112926cz/`.

[61] *Rnnoise* [online]. [cit. 2023-05-20]. Dostupné z: `https://github.com/xiph/rnnoise`.

[62] *Pyroomacoustics.denoise.spectral_subtraction module* [online]. [cit. 2023-05-20]. Dostupné z: `https://pyroomacoustics.readthedocs.io/en/pypi-release/pyroomacoustics.denoise.spectral_subtraction.html`.

[63] *Pyroomacoustics.denoise.iterative_wiener module* [online]. [cit. 2023-05-20]. Dostupné z: `https://pyroomacoustics.readthedocs.io/en/pypi-release/pyroomacoustics.denoise.iterative_wiener.html`.

[64] *Noisereduce 2.0.1* [online]. [cit. 2023-05-20]. Dostupné z: `https://pypi.org/project/noisereduce/`.

SEZNAM ZKRATEK A SYMBOLŮ

ASR automatické rozpoznávání řeči – Automatic speech recognition

HMM skrytý Markovův model – Hidden Markov model

GMM směs Gaussovských rozdělení – Gaussian mixture model

DNN hluboké neuronové sítě – Deep neural network

MFCC Mel-frekvenční kepstrální koeficient – Mel-frequency cepstral coefficient

E2E End-to-end

CTC konekcionistická časová klasifikace – Connectionist temporal classification

RNN rekurentní neuronová síť – Recurent neural network

RNN-T převodník rekurentní neuronové sítě – Recurent neural network transducer

PCM pulzně kódová modulace – Pulse-code modulation

WER četnost chybných slov – Word error rate

CNN konvoluční neuronová síť – Convolutional neural network

SNR poměr signálu a šumu – Sound to noise ratio

VAD detekce řečové aktivity – Voice activity detection

FFT rychlá Fourierova transformace – Fast Fourier transform

IFFT zpětná rychlá Fourierova transformace – Inverse fast Fourier transform

LSTM Long Short-Term Memory

GRU Gated Recurrent Unit

BFCC Bark-frekvenční kepstrální koeficienty – Bark-frequency cepstral coefficient

SRU jednoduchá rekurentní jednotka – Simple Recurrent Unit

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

DP_209297_prilohy.zip - Archiv obsahující zdrojové kódy navrženého systému