VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ODEZÍRÁNÍ ZE RTŮ POMOCÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ

LIP READING WITH DEEP NEURAL NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JOSEF KADLEČEK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MICHAL HRADIŠ, Ph.D.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Kadleček Josef**
Program: Informační technologie
Název: **Odezírání ze rtů pomocí hlubokých neuronových sítí**
Convolutional Networks for Lip Reading
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy konvolučních sítí a rozpoznávání řeči.
2. Vytvořte si přehled o současných metodách rozpoznávání textu se zaměřením na Connectionist Temporal Classification a odezírání ze rtů.
3. Vyberte nebo navrhněte metodu aplikovatelnou na odezírání ze rtů.
4. Připravte si databázi vhodnou pro experimenty.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- J. S. Chung, A. Senior, O. Vinyals, A. Zisserman: Lip Reading Sentences in the Wild. IEEE Conference on Computer Vision and Pattern Recognition, 2017
- Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. CVPR, 2017.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 14. května 2019

Abstrakt

Tato práce se zabývá současnými metodami pro přepis řeči na text a odezírání ze rtů za pomoci neuronových sítí. Následně se zabývá podobností architektur neuronových sítí operujících nad zvukem a videem při rozpoznávání řeči a porovnává dostupné audiovizuální datové sady. Výsledkem práce je sada experimentů porovnávající různé zásahy do modelu a jejich dopad na výsledky. Dále je součástí implementace systému pro přepis řeči (CER: 12.6 %) a pro odezírání ze rtů (CER: 57,7 %). Architektury obou systémů jsou založeny na extrakci příznaků pomocí konvolučních vrstev. Za nimi následují rekurentní vrstvy LSTM, další řada konvolučních vrstev a hodnotící funkce CTC.

Abstract

This thesis deals with current methods for automatic speech recognition and lip reading via neural networks. Furthermore it deals with similarities in the architectures of neural networks for audio and visual data and available datasets in the field of audiovisual automatic speech recognition. The main contribution of this thesis is set of experiments comparing different changes in neural network architecture and its impact on results. The thesis includes an implementation of a system for automatic speech recognition from audio (CER: 12.6 %) and visual (CER: 57,7 %) data. The architectures of both systems are based on features extraction via convolutional networks followed by recurrent layers LSTM, another layer of convolutions and loss function CTC.

Klíčová slova

Odezírání ze rtů, přepis řeči, neuronové sítě, rekurentní neuronové sítě, konvoluce, počítačové vidění, sekvence na sekvenci, Enkodér-Dekodér, CTC, PyTorch, Python.

Keywords

Lip reading, speech recognition, neural networks, recurrent neural network, convolution, computer vision, sequence to sequence, Encoder-Decoder, CTC, PyTorch, Python.

Citace

KADLEČEK, Josef. *Odezírání ze rtů pomocí hlubokých neuronových sítí.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, Ph.D.

Odezírání ze rtů pomocí hlubokých neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Josef Kadleček
15. května 2019

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Michalu Hradišovi, PhD. za cenné rady, vedení práce a komponenty nezbytné pro realizaci práce. Další dík patří panu Robertu Cooperovi z BBC za poskytnutí přístupu k datovým sadám LRW, LRS2 a LRS3. Tato práce vznikla za podpory projektů CERIT Scientific Cloud (LM2015085) a CESNET (LM2015042) financovaných z programu MŠMT Projekty velkých infrastruktur pro VaVaI.

Obsah

Kapitola 1

Úvod

Přepis řeči na text se stal v dnešní době téměř samozřejmostí. Uživatel se s ním může setkávat denně v podobě různých virtuálních asistentů a jiných hlasem ovládaných aplikací – můžeme zmínit například rozmach hlasem ovládaných aplikací pro chytrou domácnost. V odborné sféře můžeme jako příklad využití uvést vyhledávání v audio, případně audiovizuálních datech a na něm založené filtry obsahu.
Žádná z těchto aplikací by však nebyla schopna fungovat bez možnosti kvalitně převádět mluvené slovo na text. Právě proto je přepis s co nejmenší chybovostí pro využití v komplexních systémech tolik kritický. Přestože v současné době je chybovost systémů přepisující řeč poměrně nízká, je stále možné, že systém vyhodnotí vstupní hlásku mylně a vznikne chyba. Ještě větší problém pak nastává, pokud je prostor zašumělý nebo mluví více lidí najednou [5]. Systém pro přepis řeči přirozeně není možné nasadit, chybí-li zvuková stopa úplně. Možným řešením proto může být do systému na přepis řeči zakomponovat vizuální informaci – při odezírání ze rtů dokážeme snáze přiřadit slova konkrétnímu řečníkovi (příkladem využití může být automatické generování titulků k videím s více řečníky [19]), při nekvalitním audiu můžeme doplnit zvukovou informaci o informaci vizuální a zvýšit tak procento úspěšnosti [11]. V neposlední řadě máme-li videozáznam mluvící osoby, který postrádá zvukovou stopu úplně (například záznam z bezpečností kamery, jež nedisponuje audio záznamem) můžeme se pokusit přepsat mluvené slovo pouze za pomoci vizuální informace.
Tato práce se zabývá porovnáním současných technologií na přepis řeči z audiovizuálních dat, dostupných audiovizuálních datových sad a následnou implementací systému schopného takovouto úlohu realizovat.

V kapitole 2 je hlouběji probrána problematika přepisu řeči na text, stručně zmíněny metody, které se využívaly dříve a metody, jež jsou v současnosti považovány za nejlepší možné řešení (*state of the art*). Kapitola 3 se zabývá popsáním dostupných audiovizuálních datových sad. Následující kapitola 4 popisuje architekturu systému implementovaného v rámci této práce. V kapitole 5 jsou potom zhodnoceny výsledky implementovaných systémů. Tyto výsledky jsou podrobeny analýze a srovnání s dalšími známými systémy realizující obdobnou úlohu. Na závěr jsou v kapitole 6 shrnuty výsledky, přínos této práce a navrhnuta další možná rozšíření do budoucna.

Kapitola 2

Přepis řeči a odezírání ze rtů

V této kapitole jsou stručně představeny systémy používané na počátku éry přepisu řeči a počátky odezírání ze rtů. Dále jsou v rámci kapitoly představeny metriky sloužící k porovnávání jednotlivých systémů a nastíněny nedostatky těchto metrik. V rámci kapitoly jsou představena a porovnána různá veřejně dostupná i komerční řešení jak na přepis řeči (myšleno z čistě zvukové informace), tak i odezírání ze rtů. V rámci systémů schopných odezírat ze rtů jsou také představeny systémy, jež dokáží kombinovat jak audio, tak vizuální informaci, a výhody z toho plynoucí. V neposlední řadě jsou v rámci kapitoly nastíněny další možnosti využití vizuální informace a systém, jenž se o ně opírá při rozlišování řečníků.

2.1 Automatický přepis řeči před nástupem moderních neuronových sítí

Tato sekce slouží k nastínění obtížnosti problému, popisu jeho vývoje a představení dostupných metod k řešení problematiky automatického přepisu řeči, nikoliv jako zevrubný popis všech existujících metod. Pokud by čtenář hledal komplexnější shrnutí s podrobným popisem jednotlivých metod, lze doporučit článek *Automatic Speech Recognition – A Brief History of the Technology Development* [21], z kterého zdejší přehled hojně čerpá.

Historie strojového přepisu řeči se začala psát již na počátku padesátých let minulého století v Bellových laboratořích, kde byl vyvinut systém [17] schopný přepisu číslic nula až devět vyřčených jedním řečníkem. Tento systém se tak dá označit za první systém pro automatický přepis řeči. O deset let později představilo IBM „Shoebox“ [1] – systém velikosti krabice od bot fungující obdobně, jako hlasem ovládaná kalkulačka: systém rozpoznával šestnáct slov – číslice od nuly do devíti a některé aritmetické operace. Převedená slova byla zadána do systému na bázi kalkulačky, který poté tiskl výsledky aritmetických výrazů. Tyto systémy byly založeny na bázi měření (případně odhadování) formantů jednotlivých samohlásek a následné klasifikaci. Přestože tyto systémy měly k obecnému přepisu řeči ještě daleko, jednalo se o prvopočátky nového vědního oboru – automatického přepisu řeči neboli ASR (z anglického *Automatic speach recognition*) a první ukázky toho, jak by v budoucnu mohlo být možné ovládat stroje pouhým hlasem.

Na počátku sedmdesátých let vstoupila na pole přepisu řeči tehdejší americká ARPA – od roku 1972 známá jako DARPA – Agentura ministerstva obrany pro pokročilé výzkumné projekty. Tato agentura přinesla finanční prostředky pro vývoj systémů pro ASR a založila SUR – *Speech Understanding Research*, volně přeloženo: výzkum pro porozumění řeči. Výsledkem tohoto výzkumu byl mimo jiné systém „Harpy“ [27], který byl natrénovaný na

datech nahraných čtyřmi řečníky a prokázal schopnost rozlišovat přes tisíc slov, přičemž testovací data byla nahrána stejnými řečníky pět měsíců po nahrání dat trénovacích. Ve stejné době probíhal výzkum v IBM, kde byla snaha o vytvoření záznamníku, který zvukový vstup přepisuje na text – systém měl do jisté míry automatizovat práci zapisovatelů v kancelářském prostředí. V Bellových laboratořích byl vyvíjen systém pro „hlasem ovládané telefonní ústředny“ – cílem tedy mimo jiné bylo, aby se jednalo o systém nezávislý na řečníkovi. Všechny tyto systémy přinesly cenné poznatky a představily metody používané dodnes – v první řadě skrytý Markovův model HMM [36] (*Hidden Markov Model*) , který definoval směřování přepisu řeči na další desetiletí, a jazykový model, díky kterému byl systém schopen produkovat přirozenější výsledky. Díky hojnému využívání HMM byly systémy konečně schopny realizovat úlohu přepisu řeči na velkém množství slov a zároveň nezávisle na řečníkovi [21]. Neuronové sítě začaly překonávat HMM až během posledních deseti let. V rámci přepisu řeči se také začala hojně používat extrakce MFC koeficentů [45] (*Mel-Frequency Cepstrum Coeficients)*, které jsou využívány i v moderních systémech popsaných v rámci této práce.
Chceme-li pokračovat v historii a představit neuronové sítě, je vhodné se ohlédnout za jejich počátky – již v roce 1943 neurofyziolog Warren McCulloch a matematik Walter Pitts představili systém [30] realizovaný pomocí elektronických obvodů, který simuloval funkci neuronů v mozku. Položili tím základ k dnešním moderním neuronovým sítím, které v současnosti ovládají nejen obor přepisu řeči.

2.2 Neuronové sítě pro přepis řeči a odezírání ze rtů

Teorie neuronových sítí v posledních letech zaznamenává obrovský pokrok a účelem této práce není obsáhnout celou teorii neuronových sítí – tato sekce jen velmi stručně shrnuje princip chování a učení neuronové sítě a odkazuje na další zdroje, kde čtenář může získat komplexnější představu o tom, co to neuronová síť je.

Učení neuronové sítě je hledání ideální funkce pro danou úlohu. V případě zde zmiňovaných modelů dopředných neuronových sítí určených k přepisu řeči probíhá učení na principu zpětné propagace chyby – *backpropagation*. Chyba, jež je vypočítána vhodnou hodnotící (*loss*) funkcí, je pak zpětně propagována modelem a upravuje zde váhy částí jednotlivých vrstev tak, aby optimalizovala výsledky daného modelu.

Stavební prvky neuronové sítě jsou krom jednotlivých modulů také

- Aktivační funkce, jež přispívají nelinearitou do modelu [34]. Z počátku byly využívány sigmoidální aktivační funkce [34], v současné době se však nejvíce používá aktivační funkce ReLU [4, 28] (*Rectified Linear Unit*), která řeší problém nasycených neuronů, které ztěžují či dokonce znemožňují zpětnou propagaci chyby. Obdobou ReLU je LeakyReLU [41], která za určitých podmínek urychluje trénování a přináší další výhody oproti klasické ReLU.

- Zahazovací vrstvy neboli *dropout* [37], které deaktivují některé náhodné neurony a přispívají tak ke snížení přetrénování.

- Normalizační vrstvy [24] zajišťují normalizaci dat, umožňují použití vyšší rychlosti učení (*learning rate*), zmenšují potřebu vhodné inicializace a do jisté míry přispívají k regularizaci modelu.

V případě audiovizuálního přepisu řeči jsou hojně využívány zejména tři základní typy vrstev:

- Plně propojené vrstvy, které jsou často používány jako poslední vrstva jednotlivých bloků neuronové sítě [5, 6], případně jako poslední vrstva části extrahující příznaky z vizuálních dat [12, 11, 13, 6].
- Konvoluční vrstvy, které jsou hojně používány pro zpracování signálů a u vizuálních dat k extrakci příznaků [5, 6, 12, 11, 13].
- Rekurentní vrstvy, které jsou vhodné pro úlohy v oblasti zpracování řeči. Díky pozorování širšího okolí klasifikovaného vstupu (v případě obousměrných rekurentních sítí v obou směrech) přenáší informaci získanou z jednotlivých slov dále do věty. Navíc jsou schopné částečně simulovat jazykový model a „hádat“ tak některá slova [31, 11, 5], čehož využívá zejména architektura Enkodér-Dekodér, která bude popsána v rámci této kapitoly.

2.3 Odezírání ze rtů

Odezírání ze rtů oproti klasickému přepisu zvuku přináší ve vstupních datech další dimenzi – zatímco vlnění zvuku lze diskretizovat a následně namapovat na jednodimenzionální vektor hodnot, vizuální informace je z podstaty dvoudimenzionální. Pro srovnání a nastínění obtížnosti úkolu odezírání ze rtů nejprve uveďme úspěšnost profesionálního odezírače ze rtů: dle dostupných zdrojů [18, 23] se úspěšnost pohybuje kolem 19 % správně klasifikovaných slov. Nejen z dat je patrné, že odezírání ze rtů je pro lidi náročný úkon a není možné vždy očekávat jednoznačný výsledek. S touto skutečností je třeba počítat i při implementaci systému na odezírání ze rtů, který je navíc ochuzen o další informace, jako je postoj řečníka, jeho gestikulace, kontext v rámci jednotlivých promluv a další vizuální informace, jež nejsou zachyceny v pohybu rtů. Jeden z problémů, který se u odezírání ze rtů objevuje nově, je lokalizace pole zájmu – ROI (*Region Of Interest*) – rtů. Většina systémů pro odezírání ze rtů má na vstupu pouze ústa a jejich okolí [46]. Je tedy nutné již v předzpracování ústa lokalizovat a udělat jejich výřez. Dalším problémem je extrakce obličejových rysů. Zatímco u zpracování zvuku se pro extrakci příznaků často používají MFC koeficienty, u zpracování obrazu nebylo dosaženo jedné konkrétní uznávané metody [46]. V přirozených podmínkách se navíc setkáváme s nestálostí držení úhlu hlavy a nemůžeme tedy předem předpokládat konstantní frontální postoj. *Visemes* jsou sady fonémů, které vypadají totožně při pozorování na rtech [46, 25]. Z toho vyplývá, že klasifikace ze rtů nemusí být vždy jednoznačná.

Shrneme-li informace podané k problematice odezírání ze rtů, je třeba dodat, že nedosahuje úspěšnosti přepisu řeči ze zvukové informace a navíc s sebou nese další nevýhody zvětšené velikosti záznamu, nutnosti detekce úst a frontální záběr na obličej mluvčího ideálně z plně frontálního pohledu. Z těchto důvodů je stále dominantní přepis řeči pouze ze zvukové informace a odezírání ze rtů se používá až v případě nedostupnosti zvukové stopy, v případě více řečníků mluvících najednou [19] nebo za účelem vylepšení výsledků získaných z přepisu ze zvuku, a nikoliv jako primární zdroj. Případnou další aplikací může být snazší rozlišování řečníků mluvících najednou, jak bude dále vysvětleno v představených systémech.

2.4 Metriky porovnávání úspěšnosti

Dříve, nežli budou představeny a porovnány konkrétní systémy pro přepis řeči a odezírání ze rtů, je vhodné ustanovit metriku na porovnávání jejich úspěšnosti a efektivity. Systémy jsou porovnávány pomocí dvou základních metrik – CER [11] (*Character Error Rate* neboli chybovost měřená v písmenech) a WER [11] (*Word Error Rate* neboli chybovost měřená ve slovech), které jsou považovány v oblasti přepisu řeči za standardní a většina publikovaných systémů uvádí aspoň jednu z nich, v lepším případě obě. Chybovost počítáme dle rovnice

$$ErrorRate = (S + D + I)/N, \tag{2.1}$$

uvedené například v článku *Lip Reading Sentences in the Wild* [11]. S je počet substitucí, D je počet odstranění a I je počet vložení, jež jsou nezbytná pro převedení řetězce vystupujícího ze systému na referenční výstup – GT (z anglického *Ground Truth*). V případě chybovosti udávané v písmenech jsou operace prováděny nad písmeny a analogicky při chybovosti ve slovech nad slovy. N je poté buď délka řetězce GT (u CER) nebo počet slov v GT řetězci (u WER).
Je nutné zmínit, že zcela zásadní pro měření chybovosti je složení testovací datové sady, na které chybovost počítáme. Ideální složení částí datové sady bude více rozebráno v kapitole 3. Zde budou zmíněny jen některé nejčastější faktory ovlivňující měření.

Nedostatečná generalizace je situace, kdy systém není dostatečně adaptovaný na typy řečníků mimo datovou sadu. Extrémní případ: pokud by byl v datové sadě pouze jeden řečník, nemusel by systém fungovat dostatečně dobře na další řečníky. Pokud by v testovací sadě nebyl jiný řečník, tento problém by nebyl odhalen. K této situaci může dojít ale i tehdy, pokud v testovací sadě je nedostatek řečníků, kteří budou unikátní právě pro testovací sadu.

Přetrénování neboli *overfitting* je situace, kdy systém je natolik adaptovaný na trénovací data, že není schopen stejně dobře rozlišovat data testovací. I z tohoto důvodu je nutné vhodné rozložení testovacích a trénovacích dat.

Vliv prostředí je další zásadní faktor pro kvalitní systém. Je tedy podstatný rozdíl mezi systémy trénovanými nad datovými sadami v laboratorních podmínkách oproti systémům trénovaným v přirozených podmínkách (*in the wild*).

Navíc je nutné brát v potaz celkovou kvalitu a způsob nahrávání a počítat se zvýšením chybovosti při započítání těchto faktorů. Kdyby hodnocení pomocí CER či WER poukazovalo na obdobnou kvalitu dvou systémů, je vhodné se podívat, nad jakými datovými sadami bylo prováděno. Ideálně vyhodnotit systémy na vlastní datové sadě. Problematika rozdělení datových sad na podčásti bude dále rozebrána v kapitole 3.

2.5 Architektura Enkodér-Dekodér

Architektura Enkodér-Dekodér [10], případně *Sequence to sequence modeling* [38], se v oboru přepisu a překladu řeči stává čím dál tím populárnější a její úspěšné použití můžeme vidět například v systému WLAS [11] (*Watch, Listen, Attend and Spell*), který je považován za *state of the art* v oblasti audiovizuálního zpracování řeči, případně v systému LAS [9] (*Listen, Attend and Spell*), který operuje pouze nad audio daty. Podrobný popis systému WLAS a současně ilustraci architektury Enkodér-Dekodér můžeme vidět na obrázku 2.2. V následujících odstavcích je rozebrán princip takovéto architektury, přičemž je pro ilustrační příklady hojně odkazováno na zmíněnou architekturu WLAS [11], kde již pojme-

nování jednotlivých částí napovídá jejich účelu: Watch (vizuální enkodér), Listen (audio enkodér) a Spell (dekodér).

Enkodér bývá složen z rekurentních vrstev – GRU [42] (*Gated Recurrent Unit*), nebo jako v případě systému WLAS z vrstev LSTM [11] (*Long Short Term Memory*). Účelem enkodéru je přijmout na vstupu jednotlivé sekvence a pomocí rekurentních jednotek je zakódovat do sekvence vektorů. V případě enkodéru *Watch* jsou vstupem obličejové příznaky vyextrahované pomocí série konvolučních vrstev zakončených vrstvou plně propojenou, v případě zvuku se jedná o koeficienty MFC [11]. Výstupem enkodéru je v obou případech vektor koncového stavu a sekvence vektorů výstupů jednotlivých rekurentních jednotek.

Attention, někdy také nazývaný *alignment*, je mechanismus přidávaný do architektur Enkodér-Dekodér. Jelikož pro enkodér je náročné propagovat informaci z počátku věty na konec, byl představen mechanismus *Attention* [7] . Zatímco Enkodér-Dekodér bez mechanismu *attention* přepisuje slova postupně a dochází tak k postupné ztrátě informace (slovo na začátku věty bude již na konci věty „zapomenuto“ nebo bude mít jen malý vliv), mechanismus *attention* usnadňuje přístup k již přepsaným klíčovým slovům a každé následující slovo tuto informaci může snáze využít. Příklad: mějme větu „Tu kost na zahradě zahrabal sousedův pes.“ Zatímco architektura bez *attention* mechanismu bude pouze velmi obtížně propagovat slova „kost“ a „zahrada“ na konec věty při klasifikaci slova „pes“ a nevyužije tak tato klíčová slova při rozhodování se, jestli šlo o slovo „les“, „pes“ nebo „ves“, *attention* mechanismus v architektuře modelu způsobí, že informace o předchozích slovech se dostanou až na konec věty a bude tak věta vnímána více jako celek. Přítomnost *attention* mechanismu je přinejmenším v systému WLAS kritická, neboť bez tohoto mechanismu model produkuje výsledky s WER větším než 100 % [11].

Dekodér v každém časovém okamžiku produkuje vlastní skryté stavy a výstupní vektory za pomoci: kontextových vektorů, výstupu a svých skrytých stavů z předchozího kroku.

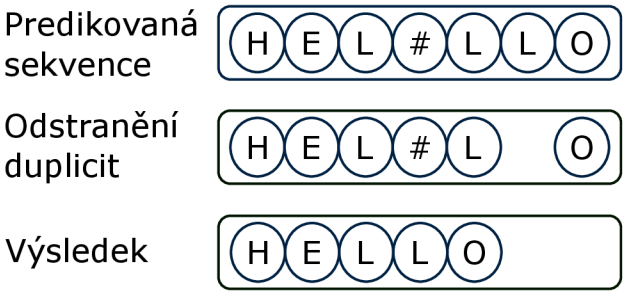
Za pomoci *attention* mechanismu je vytvořen *attention* vektor, jehož tvorba je popsána v práci od Bahdanau et al [7].

Zmiňovaný *attention* vektor (v případě WLAS dva vektory – zvlášť pro zvuk a zvlášť pro obraz) je zkombinován s výstupními stavy dekodéru, aby vyprodukoval kontextové vektory, které jsou nezbytné k vypočítání dalšího kroku. Celá tato operace je popsána v článku *Lip Reading Sentences in the Wild* [11] v sekci 2.3 rovnicemi 8 až 13. Systém WLAS je díky dvojitému *attention* mechanismu schopen extrahovat informace jak ze složky obrazové, tak zvukové. Zároveň je adaptovaný na situace, kdy jeden ze vstupů chybí.

2.6 Architektury s CTC

Druhým často používaným způsobem pro trénování sítí pro přepis řeči je architektura s hodnotící funkcí CTC [20] (*Connectionist Temporal Classification*) . Architektury využívající CTC jsou například DeepSpeech [5] (přepis řeči), nebo LipNet [6] (odezírání ze rtů). Použití této hodnotící funkce odstraňuje nutnost přesného zarovnání dat a jejich předsegmentace. Stejně jako architektury Enkodér-Dekodér tak umožňuje trénování na datech nezarovnaných s přepisy. Navíc ve většině případů předčí skryté Markovovy modely [20].

Model na bázi CTC typicky predikuje třídu (v případě ASR písmeno či slovo) pro každý časový úsek. Poté třídy, které jsou duplikované ihned za sebou, spojí do jedné. „Prázdný symbol“ (*blank label*) v obrázku 2.1 zobrazený jako „#“ zajišťuje možnost existence dvou stejných tříd (písmen) za sebou. Pokud by prázdný symbol chyběl, došlo by při spojování ke ztrátě zdvojeného písmena. Tato vlastnost je o poznání důležitější v angličtině, kde se



Obrázek 2.1: Obrázek ilustrující zarovnávání CTC inspirovaný animací z internetového článku *Sequence Modeling with CTC* [22]

Původ textu	Text
Referenční výstup	JUSTICE AND EVERYTHING ELSE
Enkodér-Dekodér	JUST GETTING EVERYTHING ELSE
Referenční výstup	IT'S TIME NOW FOR EVERYTHING IN BETWEEN
CTC	IT'S TIME OUW FOR EVERYTHING IN BE TWEN

Tabulka 2.1: Tabulka ilustruje výstupy systému WLAS [11] (Enkodér-Dekodér), a systému implementovaného v rámci této práce (CTC).

zdvojená písmena vyskytují podstatně častěji než v češtině. Následně funkce hledá optimální zarovnání těchto tříd. Slabinou CTC však je, že třídy predikované v čase jsou podmíněny pouze vstupem [11].

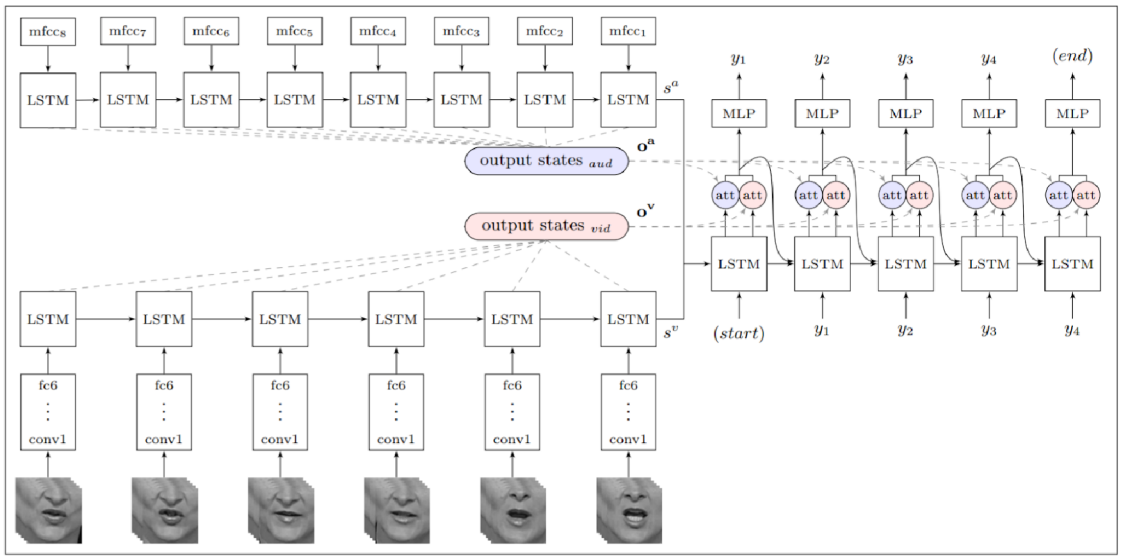
Porovnání CTC oproti architektuře Enkodér-Dekodér

Zatímco architektura Enkodér-Dekodér zakódovává vstupní sekvenci (promluvu) do vektoru, který následně dekóduje do výsledné sekvence, architektura postavená na CTC klasifikuje vždy úsek vstupní sekvence na úsek odpovídající výstupní sekvence.

Při porovnávání výsledků z architektur CTC a Enkodér-Dekodér si lze všimnout zajímavého trendu, který je ilustrován v tabulce 2.1. Architektura postavená na principu Enkodér-Dekodér se prakticky nedopouští pravopisných chyb, avšak při mylné klasifikaci může použít významově úplně jiné slovo. Oproti tomu architektura postavená na CTC se pravopisných chyb dopouští o poznání častěji namísto toho, aby užila jiného slova. Tento rozdíl se podepisuje i na zmíněných metrikách. Metrika CER nahrává architekturám s CTC, jelikož je mírnější na opravu pravopisných chyb a případných mezer uprostřed slova, naproti tomu metriku WER upřednostní architektury Enkodér-Dekodér, neboť by je metrika CER více potrestala za zmíněné „hádání“ slov.

2.7 Existující systémy pro audiovizuální přepis řeči

V této kapitole budou představeny jednotlivé systémy pro přepis řeči jen na základě audia (dále značené pouze A), pouze na základě vizuální informace (dále značené pouze V) a systémy, jež používají kombinaci obou (AV). Systémy budou porovnávány jak na zá-



Obrázek 2.2: Obrázek popisující architekturu systému WLAS [11]. Obrázek je převzatý z článku *Lip Reading Sentences in the Wild* [11], ve kterém byl systém představen.

kladě úspěšnosti (CER a WER), tak na základě architektury a využitých prvků (bude-li to možné).

Komerční řešení (A)

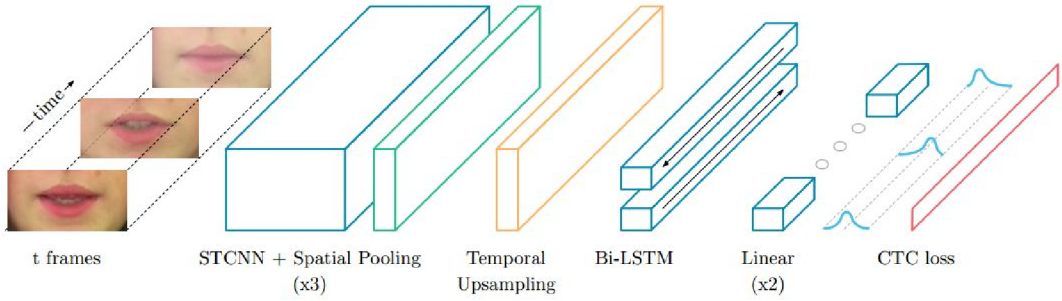
Asi nejznámější veřejně dostupná služba na převod hlasu na text je Google Speech API[1]. Běžný uživatel se setkává hlavně s hlasovým vyhledáváním nebo hlasovým zadáváním vstupu do překladače. Pro odborníky a komerční užití však Google poskytuje i řešení pomocí aplikačního rozhraní, které se může chlubit asi nejmenší chybovostí v oboru: WER 9 % předčí například i Microsoft [8]. Je nutné podotknout, že srovnání je měřeno na vlastní datové sadě a navíc je z roku 2017 – v současnosti lze očekávat výrazné zlepšení. Některé neoficiální zdroje [40] uvádějí chybovost pod 4,5 % WER. Dalším komerčním řešením je Amazon Transcribe, které dle dostupných zdrojů [32] zaostává za Google Speech API jen o nepatrné zlomky procent. Dalším velkým hráčem na poli ASR je IBM se svým systémem Watson a systém firmy Microsoft. Oba systémy se pohybují okolo hranice 5 % [39] a řadí se tak za Google a Amazon.

Dále existují řešení dalších, menších firem (zmínit můžeme například tuzemskou firmu Phonexia[2]). K těmto řešením je však pouze velmi omezené množství nezávislého srovnání .

Watch Listen Attend Spell (AV)

WLAS [11] je robustním systémem zaměřujícím se na celé věty a nikoliv izolovaná slova, jako jeho předchůdce [12]. Systém operuje nad audiovizuálními daty a je schopen přijímat pouze audio data, pouze video data a nebo je kombinovat. Navíc operuje nad datovou sadou v přirozených podmínkách a nad velkým počtem řečníků. Více o datové sadě v kapitole 3. Chybovost samotného systému pro odezírání ze rtů v rámci WLAS je CER: 59,9 % a WER:

[1]Google Speech API: https://cloud.google.com/speech-to-text/
[2]Phonexia: https://www.phonexia.com/



Obrázek 2.3: Obrázek popisující architekturu systému LipNet [6]. Obrázek je převzatý z článku *LipNet: End-to-End Sentence-level Lipreading* [6], ve kterém byl systém představen.

76,5 % avšak po zakomponování paprskového prohledávání, *curriculum learning* a *scheduled sampling* je chybovost o poznání lepší: CER: 39,5 % a WER: 50,2 % – v současnosti se tak jedná o nejlepší veřejně známý systém na odezírání ze rtů nad televizními daty. Jak již bylo zmíněno, systém je schopen kromě vizuální informace zakomponovat i audio informaci a dosáhnout tak v nejlepším případě chybovosti CER: 7,9 % a WER: 13,9 %. Systém byl také testovaný na datové sadě GRID [15], kde se osvědčil s úspěšností 3 %. Systém je ilustrován na obrázku 2.2.

Rozšíření systému WLAS autorů J.S. Chung a A. Zisserman [13] zajistilo lepší výsledky při odezírání ze rtů z profilu. Vylepšení sice nepatrně (o zhruba 1 %) zhoršilo odezírání ze rtů při čelním pohledu, avšak výrazně (u profilu až o 20 % WER) vylepšilo odezírání z úst řečníků částečně natočených do strany nebo z úplného profilu.

DeepSpeech2 (A)

Projekt DeepSpeech2 [5] představil mimo jiné množství architektur trénovaných na velkém množství dat. Většina prezentovaných architektur disponuje chybovostí okolo 8 % WER. Architektura většiny sítí se skládá z báze v podobě tří konvolučních vrstev, několika (až sedmi) rekurentních vrstev, *lookahead* konvoluce a nakonec plně propojené vrstvy. Architektury DeepSpeech využívají dříve představené CTC.

Listen Attend Spell (A)

Systém LAS [9] disponuje WER 14,1 %, aniž by používal jazykový model. Při využití jazykového modelu a paprskového prohledávání se úspěšnost zvyšuje na 10,3 % WER.

Systém je významným zástupcem architektur na principu Enkodér-Dekodér. Systém také odstranil potřebu využívat HMM. Vstupem do systému jsou akustické signály, které enkodér (*listener*) postavený na bázi pyramidové rekurentní sítě (za účelem zmenšení velikosti vstupních dat) zakóduje a dekodér (*speller*), který je v systému taktéž realizován pomocí rekurentní sítě, dekóduje na výstupní sekvenci.

LipNet (V)

Systém LipNet [6] trénovaný nad datovou sadou GRID [15] uvádí rozdíl mezi chybovostí nad předem neviděnými řečníky: CER 6,4 % a WER 11,4 %, a řečníky, již se vyskytovali v trénovací datové sadě: CER 1,9 % a WER 4,8 % Je však nutné podotknout, že datová

sada GRID je vytvořena v umělých podmínkách a obsahuje malé množství (34) řečníků. Více bude datová sada popsána v kapitole 3. Systém operuje pouze nad vizuálními daty, kde vstupem je tříkanálový (barevný) výřez rtů. Systém v první část extrahuje obličejové příznaky a zmenšuje objem dat pomocí *poolingu*. Systém je postaven na architektuře CTC a celý jeho popis vidíme na obrázku 2.3.

Looking to Listen at the Cocktail Party (AV)

Looking to Listen at the Cocktail Party [19] je článek, který představil systém, jehož primární úlohou není odezírání ze rtů, nýbrž izolace a filtrování promluv jednotlivých řečníků za pomoci vizuálních dat. Tento systém používá vizuální informaci primárně pro spárování řečníka a jemu příslušející zvukové stopy, sekundárně pak pro vylepšení přepisu zvuku. I tento systém používá konvoluci k prvnímu zpracování audiovizuálních dat, následně po fúzi audiovizuálních dat používá obousměrnou rekurentní (LSTM) vrstvu následovanou vrstvou plně propojenou.

Kapitola 3

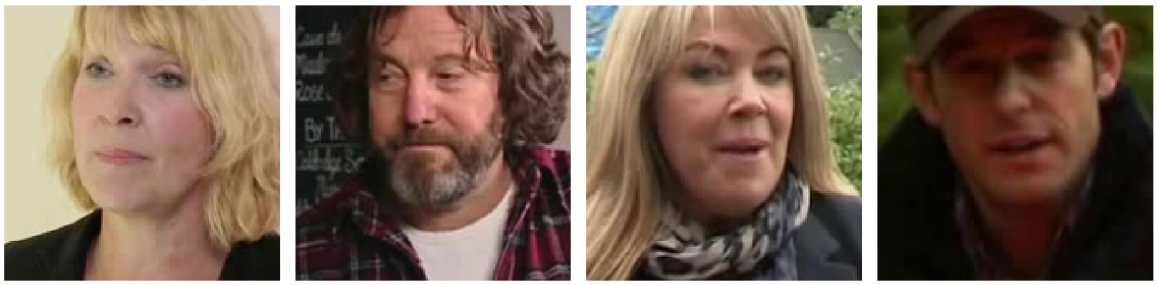
Dostupné audiovizuální datové sady

Neuronové sítě jsou podoborem strojového učení, z čehož vyplývá, že ke svému učení potřebují data. V případě přepisu řeči mluvíme o velkém množství dat. Tato kapitola popisuje , jak by měla taková data vypadat, co je nezbytnou součástí datové sady a co vítaným bonusem. Dále je zde popsáno dělení datové sady na podčásti a jejich úloha při trénování neuronové sítě. V rámci kapitoly jsou porovnány datové sady na základě velikosti a prostředí, ve kterém byly nahrávány. Nakonec jsou zmíněny některé způsoby částečného řešení nedostatečně velké datové sady. Většina datových sad pro přepis řeči je bohužel pouze zvukových. V této práci je však kladen důraz na obě složky mluveného projevu a tudíž jsou brány v potaz pouze datové sady obsahující i vizuální složku. Samostatná sekce je poté věnována datové sadě LRS2, která sloužila pro implementaci systému představeného v rámci této práce.

Rozdělení datové sady na část, na které probíhá trénování, a na část, která slouží k ověření výsledků modelu, je nezbytné pro správné využití datové sady. Díky vzorkům určeným pouze na testování je možné včas detekovat *overfitting* (situaci, kdy se model začne učit minoritní příznaky společné jen pro trénovací data). V ideálním případě mají všechny podčásti datové sady stejné statistické rozložení. Bohužel neexistuje jednoznačné pravidlo, jak velké by měly být podčásti datové sady, avšak validační část by měla být dostatečně velká, aby dokázala detekovat přetrénování modelu.

Velikost a rozložení jsou jedním z klíčových aspektů datové sady. Nejedná se pouze o počet promluv, ale i o jejich dostatečnou odlišnost. Zároveň je důležité i četnostní rozložení jednotlivých slov – v případě ideálně fungujícího systému vyžadujeme, aby rozložení datové sady odpovídalo rozložení mluvené řeči, respektive podmínkám užití – v případě obskurní, knižní, přehnaně odborné či naopak fádní či nízké mluvy nebude datová sada reflektovat reálné užití a její testovaná chybovost pak nebude odpovídat chybovosti při praktickém použití.

Počet a rozdílnost řečníků je zásadní pro správnou generalizaci (zobecnění) systému – při nedostatku řečníků se může stát, že systém rozpoznává pouze „již viděné“, respektive „již slyšené“ řečníky. Na tuto skutečnost je třeba dbát i při rozdělování datové sady na trénovací a testovací části, aby testování odhalilo případné nedostatky v generalizaci. Ze stejného důvodu jako množství řečníků je důležitá i jejich dostatečná odlišnost – ať už mluvíme o etnické, pohlavní či třeba geografické. Rozmanitost datové sady v těchto oblastech



Obrázek 3.1: Ukázka řečníků z datové sady LRS2 – téměř frontální postoj

Obrázek 3.2: Ukázka řečníků z datové sady LRS2 – částečně boční postoj a profil

umožňuje generalizaci nad větší množinou dat různých etnických skupin, nářečí či přízvuků a podepisuje se tak pozitivně na kvalitě celého systému.

Podmínky okolí jsou další zásadní okolností při strojovém učení. U zvuku mluvíme hlavně o okolním ruchu a šumu, ve vizuálních datech mluvíme zejména o nasvícení a pozadí (zdali člověk stojí před „zeleným plátnem“, nebo za ním je dav, který může potenciálně zhoršovat detekci). Rozlišujeme datové sady v umělých podmínkách, které jsou nahrávány zejména v laboratorních podmínkách v jedné místnosti s konstantní zvukovou ozvěnou, bez okolního ruchu, s konstantním nasvícením a zpravidla s malou množinou řečníků, a datové sady v přirozených podmínkách (*in the wild*), které jsou často pořizovány například z televizních záznamů.

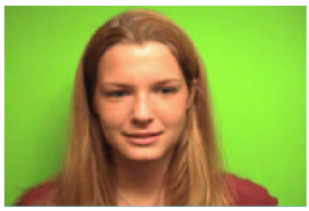
Augmentace dat se používá v případě, že datová sada není dostatečné veliká, případně jako prevence přetrénování. V případě ASR u audio dat mluvíme nejčastěji o zašumění nebo přidání hluku (*noise injection*), které navíc model více přiblíží realitě, ve které jsou ideální vstupní data zřídkavá. U vizuálních dat lze potom různě upravovat velikost výřezů, simulovat natočení nebo dopad světla pod různým úhlem.

Nedostačující velikost datové sady je častým problémem u tak náročných problémů, jako je přepis řeči nebo odezírání ze rtů. Běžně využívanými mechanismy, které částečně tento problém obchází, jsou *dropout* a *batch* normalizace.

3.1 Datová sada LRS2

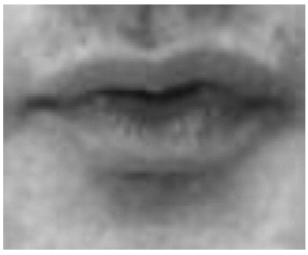
Tato datová sada [12] vznikla z televizních záznamů televize BCC z let 2010 až 2016. Primárním typem záznamu vyskytujícím se v datové sadě jsou různé typy zpráv. Autoři jako důvod uvádějí dva hlavní důvody – dostatečnou variabilitu řečníků a stálost záběru – není tak časté, že by kamera v průběhu jedné věty měnila cíl své pozornosti, a jsou tak častější kontinuální záběry na řečníky. Tento zdroj je z pohledu datové sady velmi vhodný, jelikož obsahuje velké množství řečníků v přirozených podmínkách, což lze vidět na obrázcích

(a) AVICAR



(b) CUAVE

(c) LILiR



(a) OuluVS

(b) GRID

(c) AVLetters

Obrázek 3.3: Ukázka jednotlivých řečníků ze zmiňovaných datových sad.

3.1 a 3.2, kde navíc vidíme, že řečníci nezaujímají vždy frontální postoj. Samotná datová sada je již předem rozdělena na čtyři části – trénovací, validační, testovací a předtrénovací. Pozornost si zaslouží zejména předtrénovací – tato část datové sady, jež se částečně překrývá s daty trénovacími, je specifická v tom, že jsou zde vyznačeny začátky a konce jednotlivých slov a jejich přepisy. Hlavním účelem předtrénovacích dat je *curriculum learning* – v rané fázi není vhodné model trénovat na složitých datech, ale je vhodné model připravit na trénování postupně. Obzvláště patrný je tento rozdíl u architektury Enkodér-Dekodér. Tvůrci datové sady uvádějí, že při užití jednotlivých slov pro předtrénování modelu se výsledný model zlepšil v o nezanedbatelných 15 % [11] (měřeno metrikou WER). Předchůdcem datové sady LRS2 byla datová sad LRW [12], která se však zaměřuje na jednotlivá slova a nikoliv celé věty, jako právě LRS2.

Za nedostatek datové sady by se dal označit přepis čísel. Zejména pro architekturu postavenou na CTC je důležité, aby vyřčená slova přesně odpovídala přepisu. V této datové sadě jsou však čísla většinou přepsána číslem, nikoli písmeny, která jsou na jejich vyslovení potřeba. To zvětšuje zátěž na model a potenciálně i chybovost.

Předtrénovací sada obsahuje přes dva miliony slov, zbytek datové sady přes 340 tisíc slov. Jednotlivé promluvy jsou rozděleny do složek dle data odvysílání. Samotná data jsou pak ve formátu mp4 se snímkovací frekvencí 25 fps a audiem o frekvenci 16 kHz.

3.2 Další audiovizuální datové sady

Následuje stručný přehled dalších audiovizuálních sad, který vychází primárně z článku „*A review of recent advances in visual speech decoding*“ [46] a z článků uvedených v rámci popisu jednotlivých datových sad. Datové sady nejsou vždy zaměřeny primárně na odezírání ze rtů, ale například na autentizaci pomocí obličeje nebo pouhou extrakci obličejových rysů. Tento souhrn zpravidla neuvádí technické specifikace, jako je snímkovací frekvence či formát uložení dat, neboť se očekává, že v případě potřeby je čtenář nalezne v originálních článcích.

AVICAR [26] je datovou sadou pořizovanou v jedoucím autě se sto různými řečníky, přičemž záznamy 86 z nich jsou dostupné. Záběry jsou simultánně natáčeny čtyřmi kamerami a sedmi mikrofony. Datová sada se věrně snaží napodobit přirozené podmínky do té míry, že jsou nahrávky pořizovány za různých rychlostí a se zavřenými i staženými okny, což způsobuje rozdílnou úroveň ruchu. Situace, kdy je automobil v pohybu, zajišťuje i rozdílné úhly nasvětlení, tudíž i vizuální data by se dala označit za „v přirozených podmínkách“, avšak tyto podmínky jsou natolik neměnné, že v celkovém souhrnu datová sada splňuje kritérium přirozených podmínek pouze částečně. Promluvy v datové sadě jsou ve formě izolovaných číslic, písmen, desetimístných telefonních čísel a několika vět.

AVLetters [29] a AVLetters2 [16] jsou datové sady, které dohromady obsahují patnáct řečníků, kteří každý třikrát izolovaně vyslovují písmena abecedy. Ústa řečníka jsou na začátku i konci promluvy zavřená. Tato datová sada je k pokročilému odezírání ze rtů nevhodná, neboť neobsahuje ani věty, ba dokonce ani slova.

CUAVE [35] je datová sada, která byla nahrána před zeleným plátnem, aby mohla být simulována široká škála pozadí. Sedmnáct řečníků a devatenáct řečnic předčítá číslice s různou polohou hlavy, což umožňuje studium obličejových rysů při nefrontálním postoji. Jelikož při odezírání ze rtů jsou řečníkova ústa vyříznuta, zelené plátno bohužel nepřináší téměř žádnou výhodu, přičemž jsou zachovány laboratorní podmínky, jako je odrážení zvuku či nasvícení.

GRID [15] je datová sada, jež byla využita při implementaci systému LipNet [6]. V datové sadě se vyskytuje 34 řečníků a každý předčítá tisíc frází. Fráze jsou ve formě: sloveso + barva + předložka + písmeno abecedy + číslice + příslovce, přičemž jsou zde použita pouze čtyři různá slovesa, barvy, předložky a příslovce. Číslice jsou v datové sadě od nuly do devíti. Vyjádříme-li tuto informaci v objemu slovní zásoby, znamená to $4 * 4 + 10 + 25 = 51$ různých slov. Výsledky modelu operujícího nad touto datovou sadou mohou být zkreslené, neboť má konstantní stavbu vět a mohlo by tak dojít k natrénování modelu, který vždy v čase vybírá pouze jednu z maximálně 25 tříd.

LILiR [2], celým jménem *The Language Independent Lip-Reading database*, se skládá z dvaceti řečníků vyslovujících 200 vět. Řečníci jsou nahráváni dvěma HD kamerami (ánfas a z profilu) a dalšími třemi kamerami umístěnými pod úhly 30°, 45°a 60°. Tato datová sada může být vhodná pro experimenty porovnávající zhoršení odezírání ze rtů při různých natočeních hlavy. Pro natrénování celého systému je vcelku malá jak z hlediska celkového trvání, tak slovní zásoby.

LRW [12] je předchůdcem datové sady LRS2 a tudíž se opět jedná o záznamy z BBC s promluvami mnoha různých řečníků. Tato datová sada je však o poznání menší. Jedná se o jednotlivá slova, kterých je v datové sadě pět set druhů.

LRS3 [3] datová sada vytvořená z tisíců promluv ze známé série konferencí TED a TEDx. Celková délka datové sady je asi čtyři sta hodin a jedná se tím pádem asi o největší známou veřejně dostupnou audiovizuální datovou sadu. Součástí datové sady je i rozdělení na části, které zajišťuje, že testovací sada je unikátní a nijak se nepřekrývá s předtrénovací nebo trénovací částí.

OuluVS [44] je poslední z uvedených databází pro audiovizuální přepis řeči. Skládá se z deseti běžně používaných anglických frází vyřčených dvaceti převážně mužskými řečníky. Každá fráze byla řečníkem zopakována až devětkrát.

Na závěr lze říci, že audiovizuálních datových sad je poměrně mnoho. Budeme-li ale uvažovat pouze datové sady s velkým množství řečníků, v přirozených podmínkách, s velikou slovní zásobou a dostatečnou velikostí, vyjdou nám jako nejlepší datové sady poskytované

	Počet řečníků	Počet promluv	Podmínky	Slovní zásoba	Velikost
GRID	34	33k	umělé	51	27,5h
AVICAR	86				
LILiR	20	200	umělé		
LRW			přirozené	500	173h
LRS2		140k+	přirozené	41k+	228,5h
LRS3	5k+	160k+	přirozené	51k+	438h

Tabulka 3.1: Tabulka popisující obsahy datových sad

Visual Geometry Group z University of Oxford – jedná se zejména o LRS2 a LRS3. Kvantitativní srovnání poskytuje tabulka 3.1.

Kapitola 4

Implementace vlastního systému pro přepis řeči

Tato kapitola se zabývá problematikou samotného vytváření systému pro audiovizuální přepis řeči a opírá se o konkrétní implementaci, jež je součástí této práce. Čtenář bude postupně proveden kroky, jako je výběr datové sady a její předzpracování. Poté bude obecně popsána architektura systému jako takového a popsány jeho jednotlivé části, jako například model neuronové sítě, třída pro načítání dat, třída pro vypisování a vykreslování statistik a další. V rámci popisu systému budou zmíněny možnosti jeho nasazení, případná změna nebo rozšíření datové sady, možnosti změny nastavení systému, jako je *batch size*, rychlost učení nebo symboly výstupní abecedy.

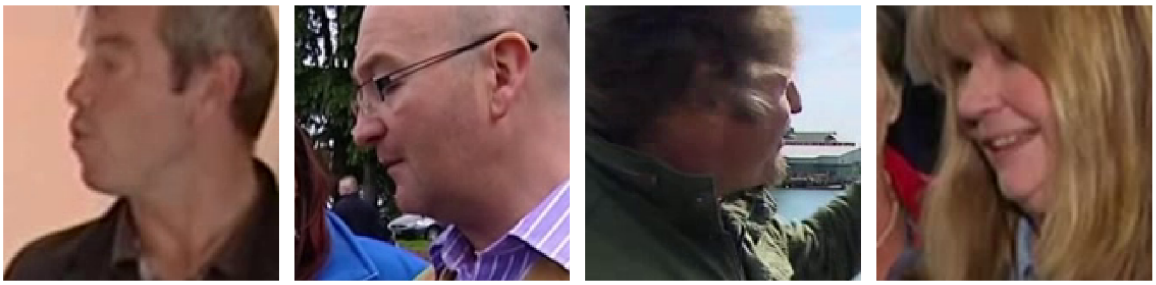
Pro začátek je vhodné podotknout, že systém je naprogramovaný v jazyce Python[1] za hojného využití frameworku PyTorch[2], ve kterém je impementována celá logika neuronové sítě zajišťující funkčnost systému. V následující kapitole pak budou popsány výsledky experimentů provedených nad tímto systémem a důsledky změn provedených v nastavení systému a architektuře modelu.

4.1 Příprava dat a jejich vstupní formát

Data poskytovaná v rámci datové sady LRS2 [11] jsou ve formátu mp4. Je tedy nutné data předzpracovat do formátu vhodného pro načítání a další zpracování v neuronové síti.

Předzpracování audia se skládá z prvotní extrakce zvukové stopy a převodu do nekomprimovaného formátu. Byl zvolen standardní formát WAV (*Waveform audio file format*), do nějž byla původní zvuková stopa převedena za pomoci utility *ffmpeg*[3]. Formát WAV je zároveň vhodný pro následnou extrakci MFCC (*Mel-frequency cepstrum coeficients*), která je prováděna za pomoci knihovny *Python Speech Features*[4], přičemž je ponecháno výchozí nastavení šířky okna a časový posuv. Vzorkovací frekvence je definována vstupním audiem – 16kHz. Výsledek v podobě MFC koeficientů je následně uložen ve formátu npy – *Numpy Array*[5], který zajištuje rychlé načítání, přičemž udržuje malou velikost dat.

[1] Programovací jazyk Python: https://www.python.org/
[2] Framework PyTorch: https://pytorch.org/
[3] Utilita ffmpeg: https://ffmpeg.org/
[4] Knihovna Python Speech Features: https://python-speech-features.readthedocs.io/en/latest/
[5] Knihovna NumPy: https://www.numpy.org/



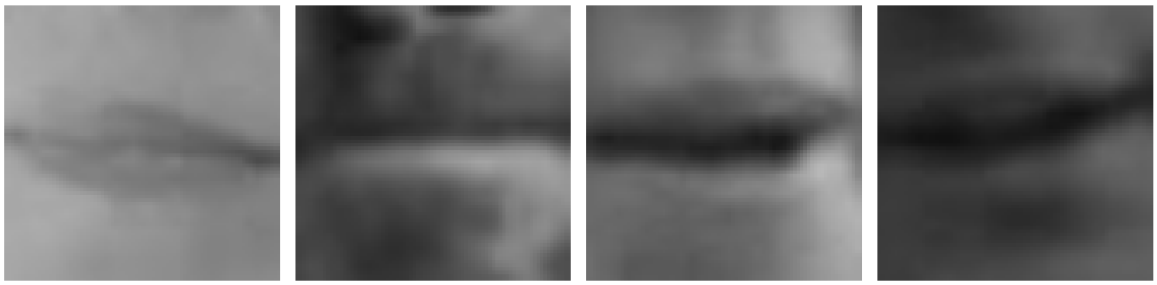
Obrázek 4.1: Ukázka řečníků z datové sady LRS2, již neprošli předzpracováním

Předzpracování videa je poněkud náročnější jak na realizaci, tak zejména časově. Jelikož datová sada obsahuje celou hlavu řečníka a okolí, je nejprve potřeba lokalizovat a vyřezat rty řečníka. K detekci rtů slouží systém implementovaný vedoucím této bakalářské práce Ing. Michalem Hradišem, Ph.D. Systém je inspirovaný prací autorů Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li a Yu Qiao [43]. Navíc je v detektoru jedna konvoluční neuronová síť, která má na vstupu detekci z MTCNN (*Multi-task Cascaded Convolutional Networks*) a odhaduje pozici 70 obličejových bodů. Tato síť byla natrénována na náhodných fotkách obličejů z Flickru[6], pro které *ground truth* pozice obličejových bodů vyprodukoval OpenPose [14]. Jelikož je datová sada v přirozených podmínkách a na následnou detekci rtů je použit jiný detektor než při její tvorbě, dochází přirozeně k chybným detekcím či nedetekování obličeje a úst. Tato situace není častá – stává se zhruba v 1 % případů. Příklady takovýchto vzorků vidíme na obrázku 4.1, jedná se zejména o velmi nekvalitní vzorky, natočení do extrémního profilu a nedostatečnou viditelnost očí (které slouží jako primární zdroj detekce obličeje). Tyto vzorky jsou poté vyřazeny z příslušných manifestů (soubory obsahující cesty k datům z datové sady), čímž je bohužel částečně zmenšena datová sada.

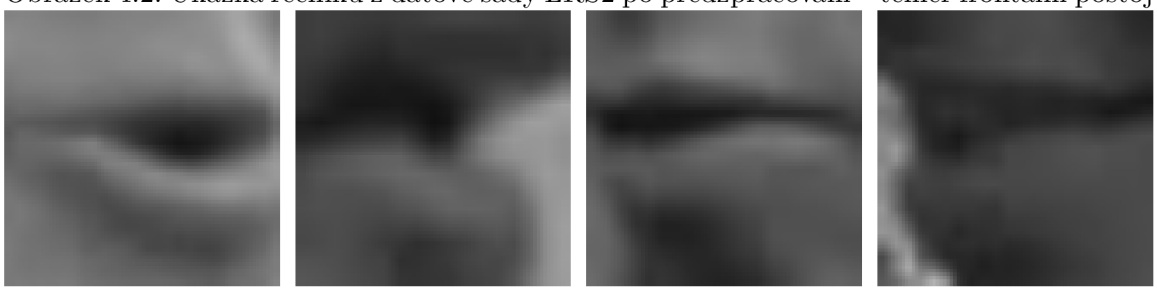
Další zúžení datové sady je způsobenou využitím architektury s CTC [20], neboť ta vyžaduje, aby délka vstupní sekvence byla delší než sekvence výstupní, aby mohlo proběhnout několik zarovnání a klasifikace na *blank labels*. Tato vlastnost není problémem u audio dat, neboť vzorkování probíhá každých 10 ms (100x za sekundu), ale u vizuálních dat, jež jsou snímkované na 25 fps (25x za sekundu) vzniká problém, kdy za jednosekundové video (25 snímků) zvládne rychle mluvící řečník vyřknout 25 znaků a mezer. CTC pak není schopná vyzkoušet zarovnání a v současné implementaci (PyTorch 1.0.1) vrací nekonečno a znehodnotí tak celý *batch*. Tyto vzorky byly tudíž také vyřazeny z trénovacího manifestu. Ve vzorcích, kde proběhla detekce úspěšně, je následně proveden čtvercový výřez rtů, který je převeden do černobílé. Otázkou zůstává, zdali je lepší tříkanálový barevný vstup, nebo jednokanálový – černobílý vstup a jestli je vhodnější vyřezávat čtvercové, nebo obdélníkové okolí. Systém WLAS implementovaný v rámci článku LRSW [11] pracuje se čtvercovým černobílým výřezem a systém LipNet [6] s obdélníkovým barevným výřezem. Ve své práci jsem zvolil metodu čtvercových jednokanálových výřezů se škálováním 80x80 pixelů, a to z následujících důvodů: při vyřezávání rtů vyslovujících písmena jako například „ó“ je výška výřezu srovnatelná se šířkou. Buď by tedy musel výřez zabírat širší okolí úst, nebo by deformoval rozměry úst. Přístup využívající širší okolí úst může nést další informaci, ale zvětšuje objem dat a ztěžuje tak detekci nejpodstatnějších částí.

Přístup využívající barevný vstup zvyšuje trojnásobně objem vstupních dat a značně tak zpomaluje trénování. Pro vývoj architektury bylo tedy vhodné použít černobílý vstup i za cenu ochuzení se o dodatečné vizuální informace. Experimenty s barevnými daty nebyly

[6]Flickr: https://www.flickr.com/



Obrázek 4.2: Ukázka řečníků z datové sady LRS2 po předzpracování – téměř frontální postoj



Obrázek 4.3: Ukázka řečníků z datové sady LRS2 po předzpracování – částečně boční postoj nebo profil

provedeny zejména z důvodu dlouhého předzpracování dat. Zůstává tedy otázkou, jak moc by barevnost napomohla ve výsledném modelu.

Na konci předzpracování jsou data opět uložena ve formátu npy ze stejného důvodu jako u audio dat.

Příklady dat po předzpracování můžeme vidět na obrázcích 4.2 a 4.3, jež zobrazují stejné řečníky, jací byli zobrazeni na obrázcích 3.1 a 3.2.

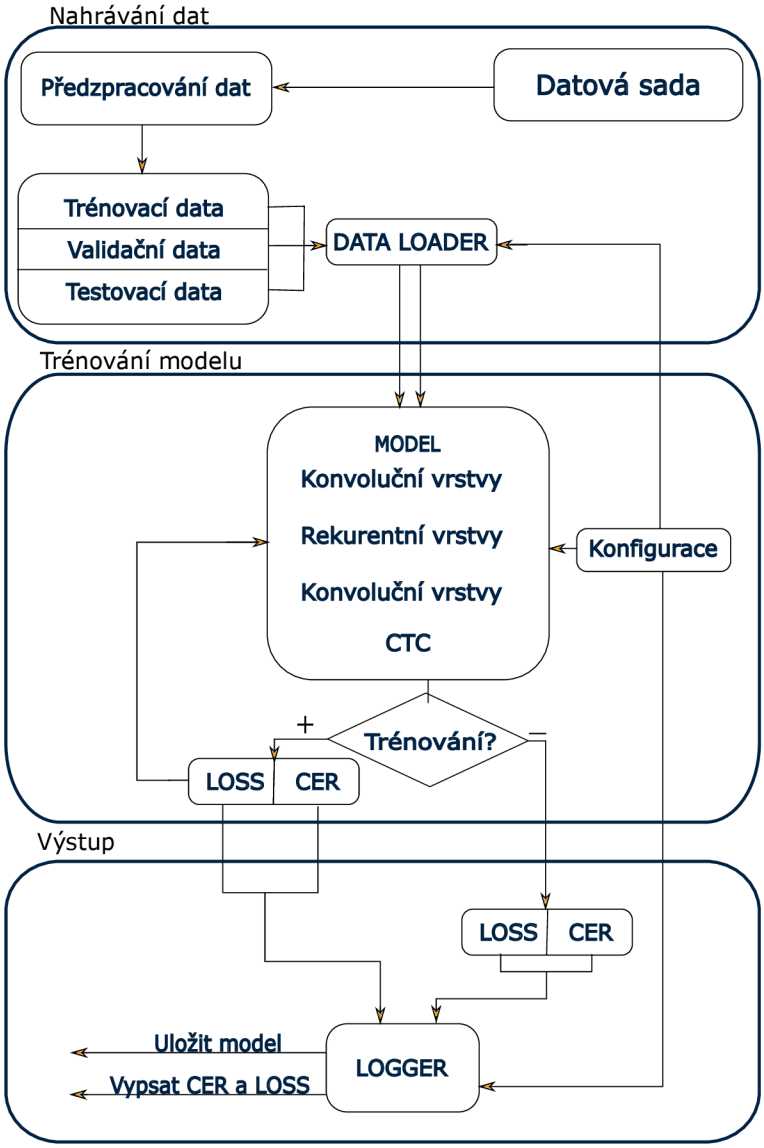
4.2 Architektura systému

Poté, co byla vstupní data předzpracována a uložena ve vhodném formátu, je čas na popsání architektury systému, jímž budou data procházet. Systém jako takový lze rozdělit do tří částí – nahrávání dat a případné předzpracování dat, samotné trénování a výpis statistik a ukládání modelu.

Konfigurace

Jelikož trénování probíhalo na mnoha různých strojích – od domácího počítače přes Collaboratory[7] až po Metacentrum[8], bylo nutné zajistit přenositelnost. Dále je nutné definovat proměnné jako je rychlost učení, *batch size*, výstupní abecedu, frekvenci logování a cestu k datové sadě a manifestům. Všechny tyto informace jsou pro jednoduchost shluknuty v třídě *Config*, která tak umožňuje jednoduchou přenositelnost mezi výpočetními stroji. Zároveň specifikuje složku, do které se mají ukládat výsledky experimentu, případně takovou složku vytváří v místě, kde je kořenová složka datové sady se jménem ve formě aktuálního data a času. Systém byl testován na operačních systémech Windows a Linux.

[7] Collaboratory: https://colab.research.google.com
[8] Metacentrum: https://metavo.metacentrum.cz



Obrázek 4.4: Architektura systému.

Nahrávání a předzpracování dat

Součástí systému je předzpracovávací skript, který z modulu `Config` vyčte cestu ke kořenu adresáře a cesty k jednotlivým manifestům. Jednoduše tak lze s pomocí zmíněného skriptu předzpracovat audiovizuální datové sady, ve kterých jsou data uložená ve formátu mp4. Nutností je samozřejmě u audio dat mít nainstalované příslušné utility a knihovny, jež zajišťují převod na formát WAV a extrakci MFCC (jak bylo zmíněno v předchozí kapitole). Skript poté vytvoří na stejném místě soubory s původním jménem a příponou „_lips.npy“ pro vizuální data a „_mfcc.npy“ pro audio data. Původní soubory jsou zachovány. Předzpracovávací skript je díky tomu spouštěn pouze jednorázově (případně pro každou novou datovou sadu).
Samotné načítání dat do systému tvoří třídy `AudioDataLoader`, `VideoDataLoader` a `PretrainDataLoader`, jež dědí ze třídy `DataLoader` implementované v rámci frameworku

PyTorch. Třídy zajišťují paralelní nahrávání dat na grafickou kartu a minimalizují tak čekání na data.
Zmiňované třídy `AudioDataLoader` a `VideoDataLoader` obsahují tři instance třídy `Dataset` – pro trénovací, validační a testovací data. Třída `PretrainDataLoader` byla implementovaná pouze pro audio data a obsahuje jednu instanci třídy `Dataset`, protože slouží pouze k trénování. Úkolem tříd `Dataset` je nahrání jednotlivých vzorků a následné vytvoření *batche*. Nahrání jednotlivých vzorků sestává z vytvoření čtyř tensorů – v případě audio dat se jedná o tensor MFC koeficientů, tensor obsahující délku promluvy, tensor obsahující GT (přepisy promluv) a tensor obsahující délku dané promluvy, u vizuálních dat jsou poté v prvním zmiňovaném tensoru uloženy výřezy rtů v čase, a v druhém jejich délky. Přepisy promluv a jejich délky jsou přirozeně shodné jako u audio dat. Následně je z B (velikost *batche*) vzorků vytvořen *batch*, přičemž probíhá zarovnání jednotlivých sekvencí na jednotnou délku (což je nezbytné pro následný vstup *batche* do sítě). Délky jednotlivých sekvencí jsou nyní rovny nejdelší sekvenci v *batchi*, proto je nutné zachovat jejich původní délky v dříve vytvořených tensorech. Poté, co je *batch* vytvořen, může vstoupit do modelu.

Model

Předtím, než bude popsáno, jak vypadá samotné trénování modelu, je vhodné specifikovat požadavky na model pro přepis řeči. Architektury postavené na *loss* funkci CTC se zpravidla drží schématu: „extrakce příznaků za pomoci několika konvolučních vrstev, několik rekurentních vrstev, konvoluční nebo plně propojená vrstva“. Liší se však v počtu jednotlivých vrstev, obousměrnosti rekurentních vrstev, počtů kanálů jednotlivých vrstev nebo například ve velikostech jader zmíněných konvolucí. V této práci je představena řada modelů, které všechny odpovídají schématu vyobrazenému na obrázku 4.5. Detaily jednotlivých modelů budou blíže rozebrány v rámci kapitoly zabývající se experimentováním s modelem, ale předem lze říci, že se jedná o variaci počtu rekurentních vrstev (až 7), variaci velikosti jader jednotlivých konvolucí či porovnání LSTM oproti GRU vrstvám.

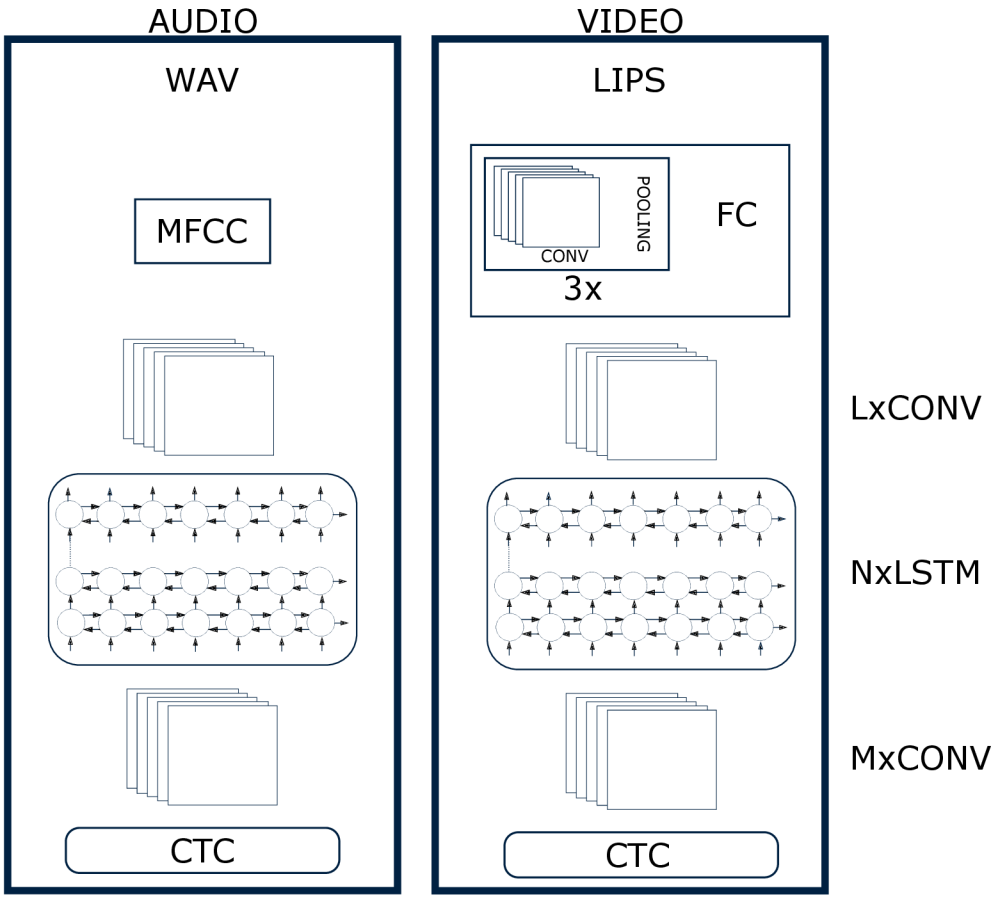
Co však musí mít bezpodmínečně všechny modely společného, je forma výstupu. Modely v rámci systému implementovaného v této práci mají výstup ve formě $L \times B \times C$, přičemž L je délka nejdelšího vzorku v čase, B je velikost *batche* a C je počet výstupních tříd – velikost abecedy. V rámci tohoto systému je velikost abecedy 44, neboť základní písmena jsou doplněna o znaky „’!?:,“ číslice a mezeru.

Trénování modelu

Před trénováním nejprve proběhne inicializace tříd `Config`, `DataLoader` a třídy `Logger`, nastavení konstantního *seed* (aby při experimentech nedocházelo k odchylkám z důvodu jinak nagenerovaných náhodných veličin) a nastavení rychlosti učení.

Samotné trénování modelu poté probíhá ve dvou vnořených cyklech – vnější, jenž realizuje epochy (projití všech dostupných trénovacích vzorků modelem) a vnitřní, jenž realizuje právě onen průchod jednotlivých vzorků. Celý algoritmus trénování je stručně popsán v algoritmu 1.

Právě proběhnutí jednoho vnitřního cyklu je jeden *update* a symbolizuje jedno zavolání funkce *backward* nad *loss*. Počet *update* bude v grafech jednotlivých experimentů znázorněn na ose x.



Obrázek 4.5: Schematický náčrt architektur systémů pro přepis řeči implementovaných v rámci této práce. Počty vrstev a velikosti kanálů se v rámci jednotlivých modelů liší.

Výstup

Stav sítě je průběžně vypisován do složky běžícího experimentu specifikované v konfiguračním modulu. Součástí jsou historie validační a trénovací *loss* a CER uložené ve formátu „.npy“ (z důvodu snadného opětovného nahrání), grafy *loss* a CER, podoba modelu v textové podobě, uložené modely (případně pouze nejlepší model) a výsledky modelu nad testovací sadou v lidem čitelné podobě (jelikož CER nemusí být vždy dostačující metrika, je vhodné vizuálně potvrdit, že síť plní svůj účel).

```
Inicializace;
while je trénování aktivní do
    vynulování historie loss a CER;
    while datová sada obsahuje další vzorky do
        přepni model do trénovacího režimu;
        propaguj batch modelem a získej výsledek;
        spočítej loss dle toho, jak moc se výsledek liší od GT;
        proveď zpětnou propagaci;
        přepni model do vyhodnocovacího režimu;
        propaguj batch modelem a získej výsledek;
        spočítej loss a CER a ulož je do historie;
    end
    přepni model do vyhodnocovacího režimu;
    spočítej loss a CER na validační a testovací sadě;
    vypiš výsledky trénování, testování a validace;
    ulož stav modelu;
end
```

Algoritmus 1: Trénování modelu neuronové sítě

Kapitola 5

Experimenty a testování

Součástí modelu jsou čtyři základní testy – dva pro modely a dva pro načítání dat. Testy pro načítání dat (jak audio, tak video) ověřují primárně správný formát dat poskytovaný třídou `DataLoader`, ale zachytí i chyby, jako je nepřítomnost manifestu. Lze je také nastavit na kontrolu dostupnosti a správnosti všech načítaných vzorků, tato možnost je ale ve výchozím nastavení vypnutá, neboť se jedná o časově velmi náročnou operaci. Testy ověřující model jsou schopné detekovat nesourodost jednotlivých vrstev (odhalí například chyby, kdy počet výstupních kanálů jedné vrstvy není roven vstupnímu počtu kanálů následující vrstvy, přičemž není zajištěno formátování dat). Tyto testy slouží jako ověření, že trénování bude vůbec možné provést a jsou obvykle spouštěny před nasazením nových experimentálních modelů na vzdálené trénovací zařízení.

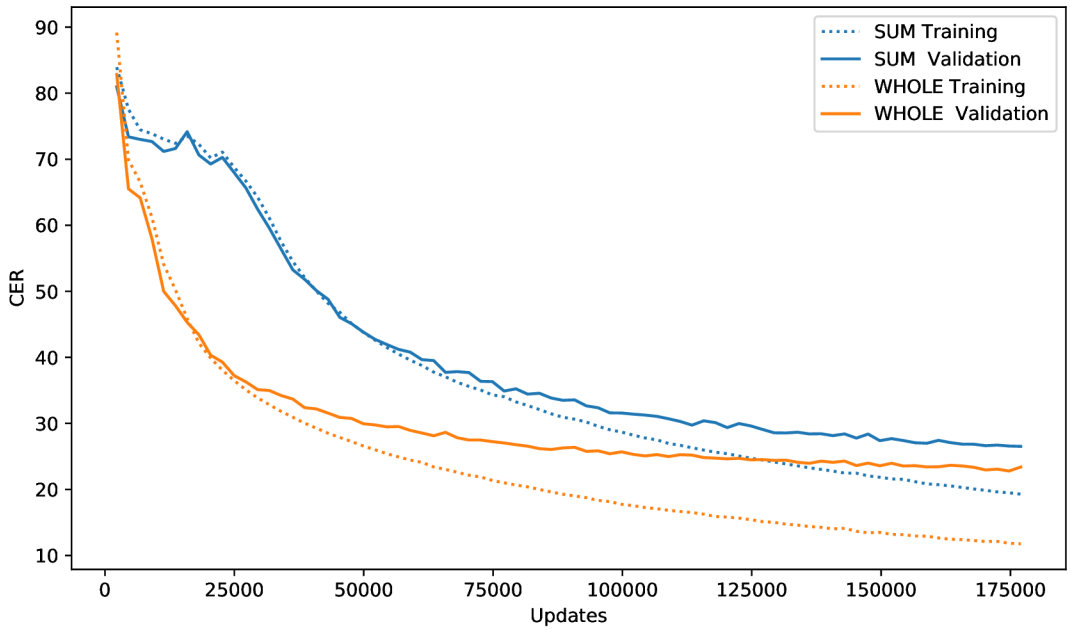
Poté, co byla ověřena schopnost modelu naučit se přepisovat řeč na malém počtu vzorků, byl systém trénován na Metacentru. V následujících podkapitolách jsou uvedeny výsledky a poznatky z vybraných experimentů. Experimenty probíhaly jak nad audio, tak video daty z důvodu porovnání jednotlivých modelů a rozlišení jejich schopnosti přepisovat řeč. Navíc experimenty nad audio daty probíhaly podstatně rychleji, jelikož se jedná o menší data.

V následujících sekcích jsou analyzovány jednotlivé experimenty, v rámci nichž jsou grafy zobrazující vývoj CER v průběhu trénování. Jelikož zdrojový kód celého systému včetně komentářů je v angličtině, jsou i popisky os a legendy v následující kapitole anglicky. Pro pochopení jsou však všechny popisky, které by mohly způsobit zmatení, dále vysvětleny pod jednotlivými grafy.

5.1 Zpracování výsledku obousměrné rekurentní vrstvy – Audio

Je několik možností jak přistoupit k rekurentním vrstvám v architektuře CTC. V architekturách prezentovaných v rámci publikace DeepSpeech2 [5] se autoři přiklánějí k jednosměrným rekurentním vrstvám, a například v LipNet [6] autoři volí obousměrné. Dalším z přístupů je sečtení výsledků jednotlivých směrů rekurentních vrstev, jak je vidět v jedné z implementací architektur DeepSpeech [33]. Další z možností je použití jednosměrných rekurentních vrstev s modifikací vstupu – jelikož v anglické větné stavbě jsou slova ke konci věty klíčově důležitější než slova na začátku, objevuje se přístup, kdy se vstupní sekvence zpracovává odzadu.

Na obrázku 5.1 vidíme graf porovnání metody sčítání výstupů rekurentních vrstev, tak jako je implementováno v jedné z dostupných verzí architektury DeepSpeech [33], a metody,



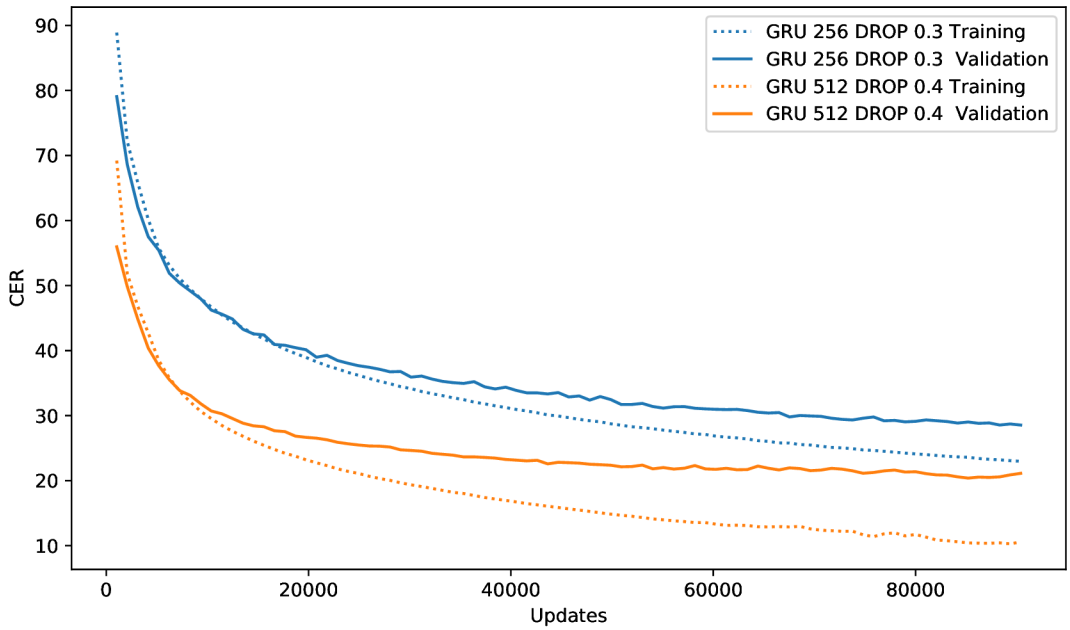
Obrázek 5.1: Porovnání vývoje CER dle nakládání s výsledkem obousměrné rekurentní vrstvy – SUM pro sčítání jednotlivých směrů rekurentních vrstev a WHOLE reprezentující přístup, kdy výstup každé jednotlivé rekurentní vrstvy je bez dalších modifikací použit jako vstup pro následující rekurentní vrstvu.

kdy celý výsledek obousměrné rekurentní vrstvy je vstupem další rekurentní vrstvy. V tomto příkladě se jedná o pět rekurentních vrstev typu GRU s *dropout* nastaveným na 0,3. Vstupní počet kanálů jednotlivých vrstev je 256 a výstupní v případě sčítání taktéž 256, v případě, že je celý výstup vložen do další rekurentní vrstvy, je výstupní počet kanálů poloviční – 128. Jak lze z grafu pozorovat, při sčítání výsledků trvalo modelu déle, než začal dávat přijatelné výsledky, avšak ani po delší době nedosahoval takových výsledků, jako přístup vkládání celého výsledku do další vrstvy. Na druhou stranu přístup, kde se výstupy sčítaly, měl menší tendence k přetrénování nežli přístup, kde s výsledky jednotlivých rekurentních vrstev nebylo nijak manipulováno.

V následujících experimentech je tedy vždy volen přístup, kdy se s výsledky obousměrné vrstvy nijak nemanipuluje.

5.2 Vliv počtu kanálů – Audio

Po prvotním experimentu, který napověděl jak nakládat s výstupem jednotlivých rekurentních vrstev, byl proveden experiment s počtem kanálů jednotlivých vrstev. Průběh CER lze pozorovat na obrázku 5.2. Dle očekávání byla větší vrstva schopna dosáhnout lepších výsledků, avšak za cenu většího a rychlejšího přetrénování, a to i s větší hodnotou parametru *dropout*. Větší *dropout* byl u větší vrstvy nastaven proto, že už u menších vrstev bylo pozorováno velmi rychlé přetrénování. Různé nastavení *dropout* může bohužel lehce zkreslovat výsledky experimentu, avšak při porovnávání dalších výsledků trénování je zřejmé, že



Obrázek 5.2: Porovnání vývoje CER na základě počtu kanálů rekurentních vrstev GRU.

tento trend se drží a že nastavení *dropout* na stejnou hodnotu by pouze prohloubilo rozdíly mezi přetrénováním modelů.

Tento experiment porovnává dva modely založené na rekurentních vrstvách GRU, přičemž v dalších experimentech budou představeny ještě další modely s větším počtem kanálů, které se nakonec ukázaly jako účinější, avšak při přesáhnutí určité meze se díky velkému počtu kanálů projevovalo větší přetrénování způsobené větší volností modelu.

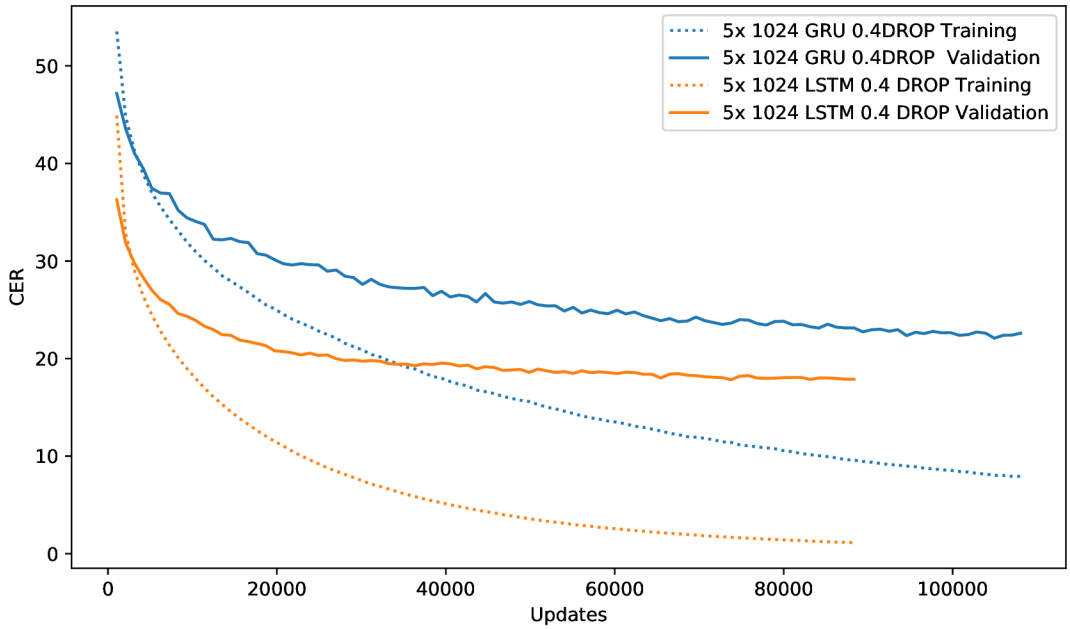
5.3 Vliv druhu rekurentní vrstev – Audio

Poté, co byl proveden experiment na počet kanálů rekurentních vrstev, vyvstala otázka, jaký druh rekurentní vrstvy vybrat, přičemž možnosti byly LSTM (*Long Short Term Memory*) a GRU (*Gated Recurrent Unit*). Je nutné zmínit, že vliv těchto dvou vrstev na výsledky modelu je poněkud obtížné porovnat z důvodu jejich charakteristik. Zatímco struktura GRU je poněkud jednodušší a menší, struktura LSTM je poněkud komplexnější. Tato vlastnost způsobuje, že GRU má menší schopnost naučit se důležité informace. Tato skutečnost bývá často vyvažována použitím větší či hlubší sítě při použití GRU vrstev. Na obrázku 5.3 lze pozorovat vývoj CER dvou totožných sítí s rozdílem použité rekurentní vrstvy. Obě mají hloubku rekurentní vrstvy 5 a počet kanálů 1024. Konvoluce před ani po rekurentních vrstvách se nijak neliší.

Z obrázku je patrné. že při použití LSTM je model schopen se naučit více informací rychleji (za méně *updates*), kdežto model postavený na bázi rekurentních vrstev GRU nebyl ani po delší době schopen dosáhnout takových výsledků.

Jako výsledek tohoto experimentu lze uvést, že výběr rekurentní vrstvy nemusí být vždy jednoznačný. Pokud se však rozhodneme použít GRU, bude muset být model podstatně

větší a je pravděpodobné, že se celkově prodlouží doba trénování. Právě proto je v této práci dávána přednost LSTM vrstvám, které se ukázaly jako výkonnější.

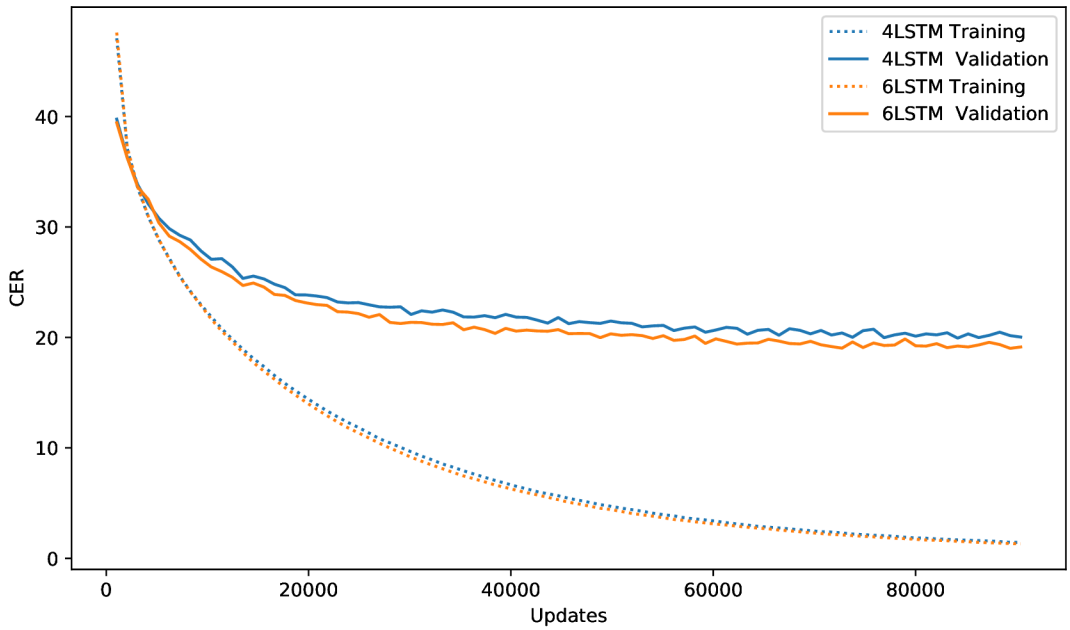


Obrázek 5.3: Porovnání vývoje CER na základě rozdílných rekurentních vrstev (GRU a LSTM). *Updates* jsou počítány v rámci trénovací sady, přičemž proběhlo předtrénování na předtrénovací sadě.

5.4 Vliv počtu rekurentních vrstev – Audio

Jelikož byly porovnány vlivy velikosti rekurentních vrstev a nakládání s jejich výstupy, přirozeně na povrch vyvstává otázka, kolik by rekurentních vrstev mělo být. Odpověď na tuto otázku se pokouší najít tento experiment. Porovnává dvě architektury založené na LSTM rekurentních vrstvách s počtem kanálů 1024 a nastavením *dropout* na 0,4. Jediný rozdíl je v počtu těchto vrstev. Jak lze vidět na grafu v obrázku 5.4 modely se chovají velmi obdobně. Rozdíl mezi šestivrstvou a čtyřvrstvou je v jednotkách procent ve prospěch šestivrstvé.

Je nutné zmínit, že zde bylo využito předtrénování na předtrénovací sadě – *update* na ose x jsou však počítány až pro data z trénovací sady. Přičtení *update* z předtrénovací sady by bylo zavádějící, neboť v předtrénovací sadě se vyskytují pouze izolovaná slova, nikoliv celé věty jako v sadě trénovací . Navíc je nutné podotknout, že trénování šestivrstvé architektury je výrazně pomalejší (exaktní měření nebylo provedeno, avšak dle počtu provedených *update* lze odhadovat zpomalení o necelých 30 %). Závěrem lze k tomuto experimentu říci, že jako kompromis mezi zlepšením o malá procenta a rychlostí trénování a s ním spojenou možnost vyzkoušení více architektur probíhala většina experimentů s pěti rekurentními vrstvami.



Obrázek 5.4: Porovnání vývoje CER na základě rozdílného počtu rekurentních vrstev. *Updates* jsou počítány v rámci trénovací sady, přičemž proběhlo předtrénování na předtrénovací sadě.

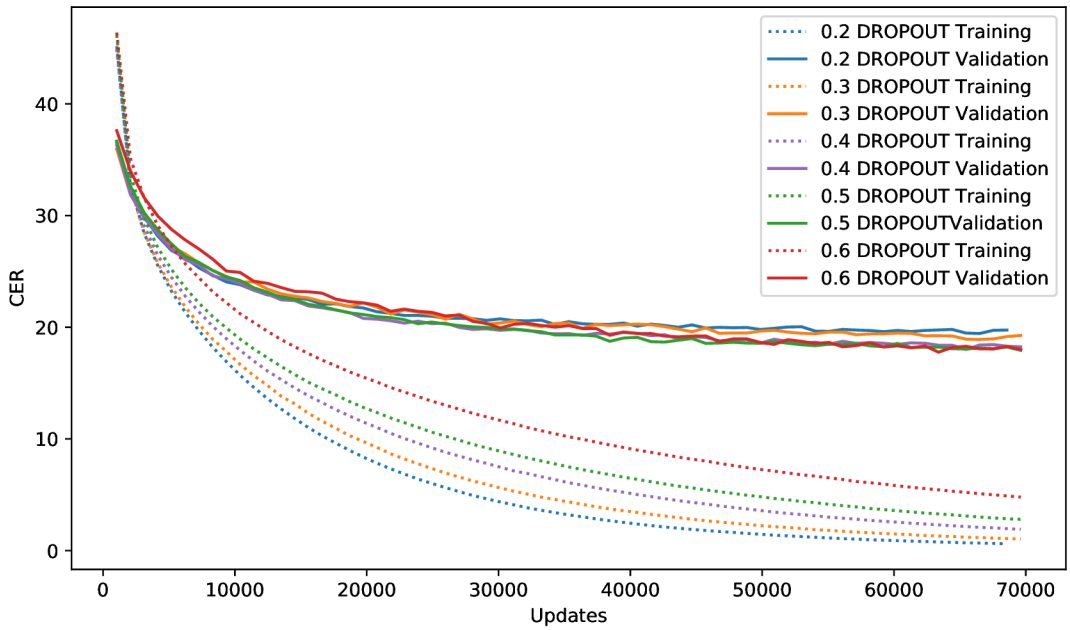
5.5 Vliv *dropout* mezi jednotlivými rekurentními vrstvami – Audio

Jelikož byl ve většině experimentů používán poměrně vysoký *dropout* mezi rekurentními vrstvami, proběhl experiment na zjištění jeho ideální hodnoty. Jak je z obrázku 5.5 vidět, regularizace pomocí *dropout* opravdu napomáhá zlepšit generalizaci modelu. Z průběhu CER jednotlivých modelů lze však pozorovat, že zvýšení hodnoty *dropout* nad 0,4 pouze zpomalilo trénování, ale nijak výrazně nezlepšilo výsledky daného modelu.

5.6 Shrnutí experimentů provedených nad audio daty

Představená sada experimentů v první řadě potvrdila schopnost modelu učit se produkovat relevantní výstupy a v druhé řadě přinesla řadu cenných poznatků co se týče velikosti, druhu a počtu jednotlivých rekurentních vrstev. Experimenty s konvolučními vrstvami před rekurentními vrstvami nebyly prezentovány z důvodu velké odlišnosti u odezírání ze rtů oproti přepisu řeči a budou více rozebrány u experimentů nad vizuálními daty.

Nejlepší model, který byl v rámci experimentů objeven, obsahoval 4 postupně se rozšiřující konvoluční vrstvy pro extrakci příznaků, všechny s velikostí jádra 3. *Dropout* mezi těmito vrstvami byl nastavený na 0,1. Následných konvolučních vrstev bylo 5 s počtem kanálů 1024, přičemž se jednalo o vrstvy LSTM s *dropout* mezi vrstvami 0,4. Po rekurentních vrstvách následovaly dvě konvoluční vrstvy, které snížily počet kanálů na velikost výstupní abecedy (44). I tyto vrstvy měly velikost konvolučního jádra 3, avšak neobsahovaly žádný



Obrázek 5.5: Porovnání vývoje CER na základě rozdílného *dropout* mezi jednotlivými rekurentními vrstvami. *Updates* jsou počítány v rámci trénovací sady, přičemž proběhlo předtrénování na předtrénovací sadě.

dropout. Úspěšnost tohoto modelu dosahovala na testovací sadě 12,6 % CER. Pro srovnání – model WLAS natrénovaný na stejné datové sadě pouze nad audio daty dosahuje úspěšnosti 10,4 % CER a Google Speech Api 17,6 % CER [11].

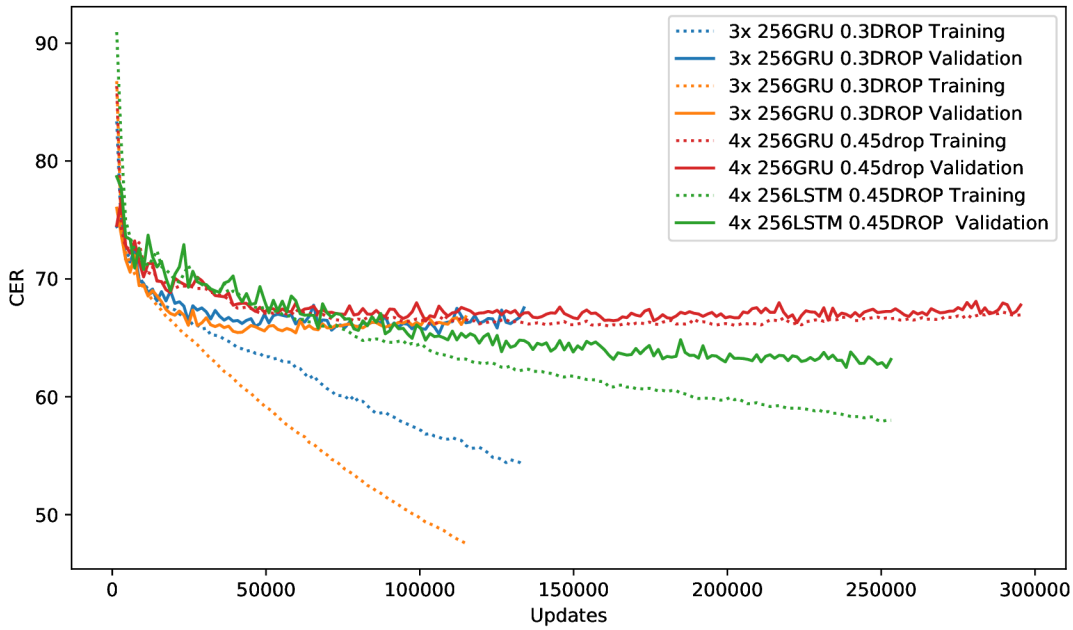
Jako poslední informaci je ještě záhodno uvést odlišnost validační a testovací CER. Ta byla v řádu jednotek procent a je dle mého soudu způsobena možnými chybami ve validační sadě (která na rozdíl od testovací sady nebyla manuálně kontrolována), případně většího počtu čísel přepsaných za pomoci číslic (přičemž model je přepsal slovně). V grafech je vždy uváděna validační CER, která byla průměrně o 5 % vyšší než výsledná testovací CER. Průběh růstu a klesání jednotlivých CER však vesměs odpovídal a není tak důvod k nedůvěře výsledkům jednotlivých experimentů.

5.7 Převod modelu na vizuální data

Poté, co byl přidán extraktor obličejových příznaků a byla předzpracována data, bylo možné spustit první experimenty nad vizuálními daty. Průběh trénování jednotlivých modelů lze pozorovat na obrázku 5.6.

Při porovnání s grafem obdobně rozsáhlých sítí nad audio daty (obrázek 5.2) je z experimentu patrné, že nedostatečná generalizace bude u vizuálních dat ještě větší problém než u audio dat. U všech testovaných modelů se testovací CER saturovala okolo 65 %, přičemž nejlepší výsledky poskytoval hlubší model založený na rekurentních vrstvách LSTM – nebyl sice schopen se naučit tolik jako mělčí model založený na GRU za stejný trénovací čas, ale vykazoval podstatně příznivější výsledky stran přetrénování. Výsledky tohoto experi-

mentu byly podnětem k vyzkoušení modelů s podstatně větším počtem kanálů a dalšímu otestování typu rekurentní sítě – na to bude kladen důraz v následujícím experimentu.



Obrázek 5.6: Prvotní experiment nad vizuálními daty zobrazující průběh CER nad modely s rozdílným počtem rekurentních vrstev, *dropout* a typu rekurentních vrstev.

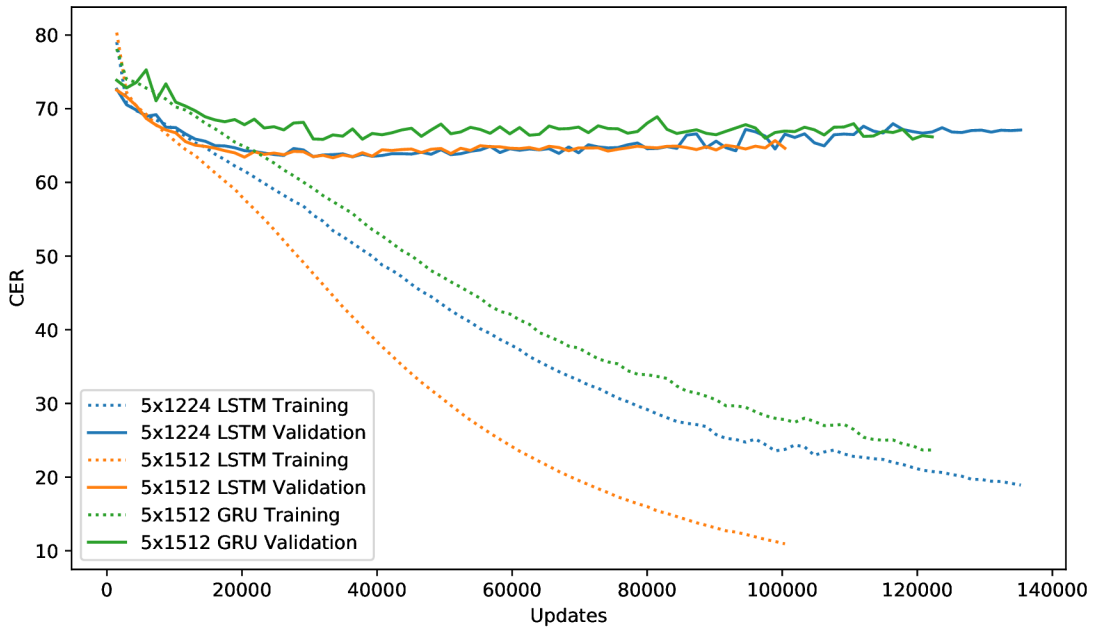
5.8 Typ a počet kanálů rekurentních vrstev – Video

Předchozí experiment nastínil větší výhodnost LSTM vrstev oproti GRU vrstvám. V tomto experimentu bude tato domněnka v rámci možností experimentálně potvrzena a bude kladen důraz na další zkoumání ideální hloubky sítě.

Nejprve k typu – již na obrázku 5.3 u audio dat a v předchozím experimentu na obrázku 5.6 byla nastíněna převaha LSTM vrstev nad vrstvami GRU. V reakci na to proběhl ještě jeden další experiment nad vizuálními daty, jehož průběh je zobrazený na obrázku 5.7, který trend potvrzuje. Nejen že vrstva GRU začala nedostatečně generalizovat data dříve než vrstva LSTM, ale navíc nebyla schopná se ani po delší době naučit nad trénovacími daty tolik, co vrstva LSTM.

Dále se tento experiment pokouší zjistit optimální počet kanálů. Průběh experimentu je zobrazen na obrázku 5.8. Opět zde můžeme vidět tři téměř totožné architektury, jejichž jediný rozdíl spočívá v různém počtu kanálů. Z obrázku je patrné, že vrstva s nejmenším počtem kanálů (1024) přešla poměrně brzy do stavu přetrénování a začala být velmi nestabilní. Tuto nestabilitu odstranilo (respektive oddálilo) zvýšení počtu kanálů, tentokrát na hodnotu 1224, avšak ani trénovací ani testovací CER stále nebyly dostatečně stabilní.

Jako dostatečný (z hlediska stability) se ukázal model s 1512 kanály. Jak se dalo očekávat, rapidně stoupla doba trénování modelu a jeho velikost. U všech modelů je však patrný trend nedostatečné generalizace – model se u testovacích dat zastaví na CER zhruba 65 %.

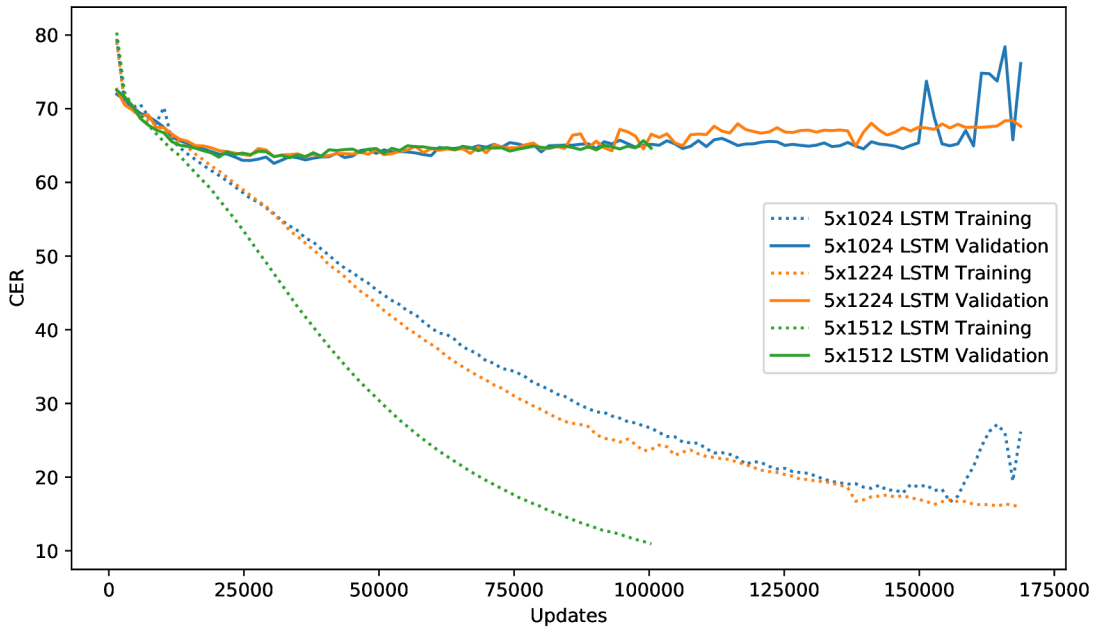


Obrázek 5.7: Vývoj CER modelů s rozdílnými typy rekurentních vrstev. Další nastavení krom počtu kanálů a typu rekurentní vrstvy zůstalo zachováno.

Jelikož tento neduh nebylo možné odstranit v rámci rekurentních vrstev, další experimenty se soustředí na konvoluční vrstvy a regularizaci dat.

5.9 Rozdílný počet konvolučních vrstev – Video

V tomto experimentu bude kladen důraz na rozdílné velikosti (jak hloubku, tak počet kanálů) konvolučních vrstev nacházejících se jak před rekurentními vrstvami, tak po nich. Nejprve proběhl pokus, který porovnával přidání konvoluční vrstvy před rekurentní vrstvy a přidání konvoluční vrstvy za rekurentní vrstvy. Jak můžeme vidět v grafu na obrázku 5.9, přidání další konvoluční vrstvy před vrstvy rekurentní (poté co jsou z jednotlivých obrázků extrahovány příznaky) zpočátku model nijak neovlivňovalo. V pozdější fázi trénování však způsobilo velkou nestabilitu, a to jak v trénovacích, tak testovacích datech. Oproti tomu přidání další konvoluční vrstvy za vrstvy rekurentní sice znatelně zpomalilo trénování modelu, avšak do jisté míry oddálilo jeho nedostatečnou generalizaci, přičemž celková schopnost naučit se přepisovat data na trénovací sadě zůstala zachována. Případné odebrání všech konvolučních vrstev po extrakci příznaků a před rekurentními vrstvami nijak nepomohlo k lepší generalizaci modelu a navíc zpomalilo a zhoršilo schopnost modelu učit se. Co se dalších hyperparametrů modelu týče: počet kanálu rekurentních vrstev: 1512, počet rekurentních vrstev: 5, velikosti konvolučních jader: 3, případně 3×3 při extrakci obličejových příznaků. Na základě tohoto experimentu byla přidána další konvoluční vrstva za rekurentní vrstvy.



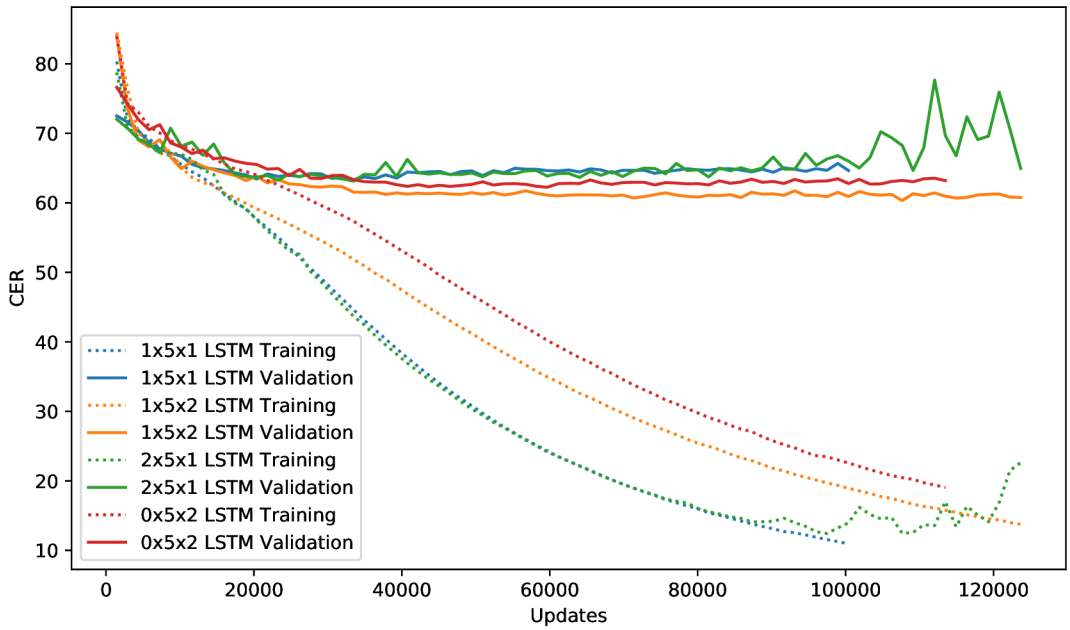
Obrázek 5.8: Vývoj CER u modelů s rekurentními vrstvami LSTM a hloubkou pět. Jediný rozdíl mezi modely je počet kanálů rekurentních vrstev.

5.10 *Dropout* při extrahování obličejových příznaků – Video

Jak předchozí experimenty ukazují, největším problémem modelu určenému k odezírání ze rtů je nedostatečná generalizace – model je schopen naučit se odezírat na trénovacích vzorcích, není však schopen aplikace nad neviděnými vzorky. Jelikož regularizace pomocí *batch* normalizace již byla aplikována, proběhl experiment s přidáním *dropout* do extraktoru obličejových příznaků. Výsledek tohoto experimentu lze pozorovat v grafu na obrázku 5.10. V grafu lze dobře vidět, že nastavení *dropout* na 0,1 nijak nepřispělo generalizaci, naopak v pozdější fázi trénování učinilo model podstatně nestabilnější. Tento trend lze o to více pozorovat při nastavení *dropout* na 0,15 – model je ještě méně stabilní a produkuje naprosto nesmyslné výsledky. Model pravděpodobně nezvládá situace, kdy z obličejových příznaků dostává neúplná data, a chová se nepredikovatelně. Přístup nastavení *dropout* ve vrstvách, které extrahují obličejové příznaky, je tedy naprosto nefunkční.

5.11 Shrnutí experimentů provedených nad vizuálními daty

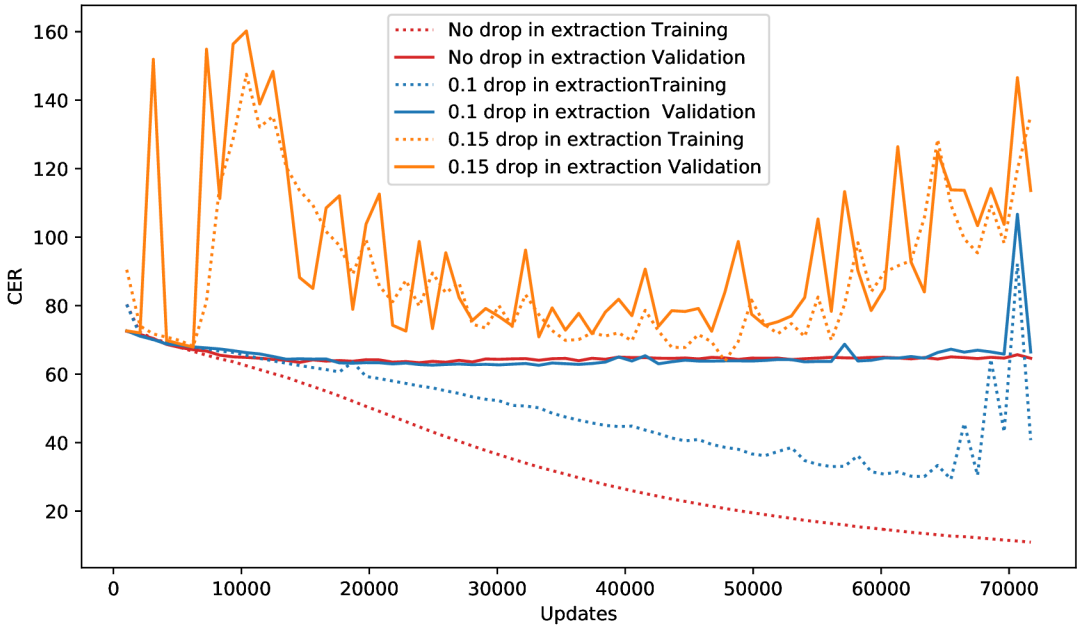
Představená sada experimentů prokázala, že při dostatečné velikosti je model schopen naučit se odezírat ze rtů, problémem u představených sítí však zůstává nedostatečná generalizace. V experimentech byly představeny různé přístupy k modelování sítě pro odezírání ze rtů, přičemž nejlepší modely produkovaly výsledky s CER 57,7 %. Model dokáže produkovat velmi dobré výsledky pro nejčastěji používaná slova (předložky, členy a některá zájmena). U některých častěji používaných slov dokáže lidský čtenář s trochou fantazie domyslet význam neúplných slov. U velmi neobvyklých slov model pouze odhaduje jedno či dvě



Obrázek 5.9: Vývoj CER u modelů s rozdílným počtem konvolučních vrstev. První číslo značí počet konvolučních vrstev po extrakci příznaků z každého jednoho obrázku, druhé počet rekurentních vrstev a třetí počet konvolučních vrstev po rekurentních vrstvách.

nejvýraznější písmena a mohl by tak být k užitku leda u velmi zašumělého audia jako poskytnutí dodatečné informace. Částečným řešením problému nedostatečné generalizace by mohlo být rozšíření použité datové sady o předtrénovací datovou sadu (ta z důvodu časové náročnosti předzpracování nebyla využita). Ideálním řešením by pak bylo vytvoření nové, větší datové sady operující nad českým jazykem.

Nejlepší model nelze s jistotou určit. Menší model (extrakce příznaků pomocí třech konvolučních vrstev proložených *poolingem* zakončených plně propojenou vrstvou, jedna konvoluční vrstva, pět BiLSTM vrstev s 512 kanály s *dropout* nastaveným na 0,4 a tři konvoluční vrstvy) sice dosáhl nejnižší CER, byl poměrně nestabilní a výsledky byly pro člověka hůře čitelné oproti většímu modelu. (Změna byla ve velikosti BiLSTM vrstev, které u většího modelu měly 1224 kanálů, a byla přítomna pouze jedna závěrečná konvoluční vrstva.)



Obrázek 5.10: Vývoj CER u modelů s rozdílným nastavením *dropout* v části modelu, která extrahuje obličejové příznaky.

Kapitola 6

Závěr

V této práci byla stručně shrnuta historie přepisu řeči, přičemž důraz byl kladen na neuronové sítě. V rámci neuronových sítí byly rozebrány dva nejpopulárnější přístupy – přístup využívající hodnotící funkci CTC a přístup využívající architekturu Enkodér-Dekodér. Tyto dva přístupy byly srovnány jak z hlediska postupu, tak výsledků a byly uvedeny ukázkové systémy pro každý z těchto přístupů. Byly detailně popsány metody porovnávání úspěšnosti – metriky CER a WER. Dále zde byly představeny systémy realizující přepis řeči jako takový, odezírání ze rtů nebo kombinaci obojího.

Dále byly představeny známé audiovizuální datové sady, které byly přehledově popsány a vybrané z nich kvantitativně shrnuty. Detailně byla rozebrána datová sada LRS2, která byla vyhodnocena jako nejvhodnější pro následující implementaci.

Byla zde slovně popsána implementace vlastního systému jak pro realizaci přepisu řeči z audio informace, tak pro odezírání ze rtů. Popis tohoto systému čtenáře schematicky provedl celým procesem od předpřípravy dat do požadovaného formátu přes jejich načítání a vstup do modelu až po vypisování statistik modelu a jeho ukládání. Jednotlivé modely byly hlouběji rozebrány v rámci jednotlivých experimentů a podrobeny rozborům, v rámci kterých byly srovnány množiny přístupů a porovnány jejich výsledky.

Výsledkem této práce je množina poznatků o neuronových sítích srovnávající různé velikosti, typy a množství jednotlivých vrstev a sada skriptů vhodných pro další experimentování nad jednotlivými modely. Dále je výsledkem této práce natrénovaný model schopen přepisu řeči (pouze audio data) s CER 12,6 % a model realizující odezírání ze rtů s chybovostí CER 57,7 %.

Možná rozšíření do budoucna zahrnují rozšíření datové sady, další experimenty postupně vylepšující oba modely, zejména odstranění nedostatečné generalizace u modelu realizující úlohu odezírání ze rtů. V neposlední řadě se nabízí možnost vytvoření datové sady nad českým jazykem a převod modelů nad český jazyk.

Literatura

[1] IBM Archives: IBM Shoebox. [Online; navštíveno 14.04.2019].
URL https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html

[2] LILiR. 2019, [Online; navštíveno 14.04.2019].
URL http://www.ee.surrey.ac.uk/Projects/LILiR/index.html

[3] Afouras, T.; Chung, J. S.; Zisserman, A.: LRS3-TED: a large-scale dataset for visual speech recognition. In *arXiv preprint arXiv:1809.00496*, 2018.

[4] Agarap, A. F.: Deep Learning using Rectified Linear Units (ReLU). 2018.

[5] Amodei, D.; Anubhai, R.; Battenberg, E.; aj.: Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *CoRR*, ročník abs/1512.02595, 2015.

[6] Assael, Y. M.; Shillingford, B.; Whiteson, S.; aj.: LipNet: Sentence-level Lipreading. *CoRR*, ročník abs/1611.01599, 2016.

[7] Bahdanau, D.; Cho, K.; Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. 2014.

[8] Bohouta, G.; Këpuska, V.: Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *Int. Journal of Engineering Research and Application*, ročník 2248-9622, 03 2017: s. 20–24.

[9] Chan, W.; Jaitly, N.; Le, Q. V.; aj.: Listen, Attend and Spell. *CoRR*, ročník abs/1508.01211, 2015.

[10] Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; aj.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR*, ročník abs/1406.1078, 2014.

[11] Chung, J. S.; Senior, A.; Vinyals, O.; aj.: Lip Reading Sentences in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Chung, J. S.; Zisserman, A.: Lip Reading in the Wild. In *Asian Conference on Computer Vision*, 2016.

[13] Chung, J. S.; Zisserman, A.: Lip Reading in Profile. 09 2017.

[14] CMU-Perceptual-Computing-Lab: *OpenPose*. [Online; navštíveno 15.04.2019].
URL https://github.com/CMU-Perceptual-Computing-Lab/openpose

[15] Cooke, M.; Barker, J.; Cunningham, S.; aj.: An audio-visual corpus for speech perception and automatic speech recognition (L). *The Journal of the Acoustical Society of America*, ročník 120, 12 2006.

[16] Cox, S.; Harvey, R.; Lan, Y.: The Challenge of multispeaker lip-reading. 01 2008.

[17] Davis, K. H.; Biddulph, R.; Balashek, S.: Automatic Recognition of Spoken Digits. *The Journal of the Acoustical Society of America*, ročník 24, č. 6, nov 1952: s. 637–642.

[18] Easton, R. D.; Basala, M.: Perceptual dominance during lipreading. *Perception & Psychophysics*, ročník 32, č. 6, Nov 1982: s. 562–570, ISSN 1532-5962.

[19] Ephrat, A.; Mosseri, I.; Lang, O.; aj.: Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Trans. Graph.*, ročník 37, 2018: s. 112:1–112:11.

[20] Graves, A.: *Connectionist Temporal Classification*. 02 2012, s. 61–93.

[21] H. Juang, B.; Rabiner, L.: Automatic Speech Recognition - A Brief History of the Technology Development. 01 2005.

[22] Hannun, A.: Sequence Modeling with CTC. 2017, doi:10.23915/distill.00008, https://distill.pub/2017/ctc.

[23] Hilder, S.; Harvey, R.; Theobald, B.-J.: Comparison of human and machine-based lip-reading. 2009.

[24] Ioffe, S.; Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, ročník abs/1502.03167, 2015.

[25] Koller, O.; Ney, H.; Bowden, R.: Deep Learning of Mouth Shapes for Sign Language. 12 2015.

[26] Lee, B.; Hasegawa-Johnson, M.; Goudeseune, C.; aj.: AVICAR: audio-visual speech corpus in a car environment. In *INTERSPEECH*, ISCA, 2004.

[27] Lowerre, B.: The Harpy speech understanding system. 12 1990: s. 576–586.

[28] Maas, A. L.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. 2013.

[29] Matthews, I.; Cootes, T.; Bangham, J. A.; aj.: Extraction of Visual Features for Lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 24, 2002: str. 2002.

[30] McCulloch, W. S.; Pitts, W.: A logical calculus of the ideas immanent in nervous activity. 1943. *Bulletin of mathematical biology*, ročník 52 1-2, 1988: s. 99–115; discussion 73–97.

[31] Mikolov, T.; Karafiát, M.; Burget, L.; aj.: Recurrent neural network based language model. In *INTERSPEECH*, editace T. Kobayashi; K. Hirose; S. Nakamura, ISCA, 2010, s. 1045–1048.

[32] Mohsin, A.: *How To Choose an ASR Service*. [Online; navštíveno 14.04.2019]. URL https://www.rev.ai/blog/how-to-choose-an-asr-service/

[33] Naren, S.: *Github*. [Online; navštíveno 15.04.2019].
URL https://github.com/SeanNaren/deepspeech.pytorch/blob/master/model.py

[34] Nwankpa, C.; Ijomah, W.; Gachagan, A.; aj.: Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *CoRR*, ročník abs/1811.03378, 2018.

[35] Patterson, E.; Gurbuz, S.; Tüfekci, Z.; aj.: CUAVE: A new audio-visual database for multimodal human-computer interface research. ročník 2, 01 2002.

[36] Rabiner, L. R.; Juang, B. H.: An introduction to hidden Markov models. *IEEE ASSp Magazine*, 1986.

[37] Srivastava, N.; Hinton, G.; Krizhevsky, A.; aj.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, ročník 15, č. 1, Leden 2014: s. 1929–1958, ISSN 1532-4435.

[38] Sutskever, I.; Vinyals, O.; Le, Q. V.: Sequence to Sequence Learning with Neural Networks. *CoRR*, ročník abs/1409.3215, 2014.

[39] Tung, L.: *IBM vs Microsoft: 'Human parity' speech recognition record changes hands again*. [Online; navštíveno 15.04.2019].
URL https://www.zdnet.com/article/ibm-vs-microsoft-human-parity-speech-recognition-record-changes-hands-again/

[40] Venturebeat: *Google's speech recognition technology now has a 4.9% word error rate*. [Online; navštíveno 15.04.2019].
URL https://venturebeat.com/2017/05/17/googles-speech-recognition-technology-now-has-a-4-9-word-error-rate/

[41] Xu, B.; Wang, N.; Chen, T.; aj.: Empirical Evaluation of Rectified Activations in Convolutional Network. *CoRR*, ročník abs/1505.00853, 2015.

[42] Zhang, J.; Du, J.; Dai, L.: A GRU-based Encoder-Decoder Approach with Attention for Online Handwritten Mathematical Expression Recognition. *CoRR*, ročník abs/1712.03991, 2017.

[43] Zhang, K.; Zhang, Z.; Li, Z.; aj.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, ročník 23, č. 10, Oct 2016: s. 1499–1503, ISSN 1070-9908.

[44] Zhao, G.; Barnard, M.; Pietikäinen, M.: Lipreading With Local Spatiotemporal Descriptors. *Multimedia, IEEE Transactions on*, ročník 11, 12 2009: s. 1254 – 1265.

[45] Zheng, F.; Zhang, G.; Song, Z.: Comparison of different implementations of MFCC. *Journal of Computer Science and Technology*, ročník 16, č. 6, Nov 2001: s. 582–589, ISSN 1860-4749.

[46] Zhou, Z.; Zhao, G.; Hong, X.; aj.: A Review of Recent Advances in Visual Speech Decoding. *Image and Vision Computing*, ročník 32, 06 2014: s. 590–605.

Příloha A

Obsah přiloženého DVD

Přibalené DVD obsahuje:

- **src** adresář obsahující zdrojové kódy implementovaných systémů
- **examples** adresář obsahující natrénované modely a ukázky jejich výstupu
- **thesis** adresář obsahující technickou zprávu a její zdrojové kódy.
- **video** video prezentující práci