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

APLIKACE PRO ODEZÍRÁNÍ MLUVENÉHO SLOVA

APPLICATION FOR VISUAL SPEECH RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **MATÚŠ PESTUN**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. TOMÁŠ GOLDMANN, Ph.D.**
SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

164849

Ústav:	Ústav inteligentních systémů (UITS)
Student:	**Pestun Matus**
Program:	Informační technologie
Název:	**Aplikace pro odezírání mluveného slova**
Kategorie:	Umělá inteligence
Akademický rok:	2024/25

Zadání:

1. Seznamte se s problematikou odezírání mluveného slova ze rtů. Zjistěte, do jaké míry lze z pohybu úst rozlišit, o jaká písmena se jedná.
2. Nastudujte metody strojového učení, které se používají nebo lze použít k rozpoznávání pohybů úst.
3. Navrhněte algoritmus, který bude zaznamenávat pohyb úst a predikovat vyslovovaná slova.
4. Navržené řešení implementujte v programovacím jazyce Python a vytvořte k němu jednoduché uživatelské rozhraní.
5. Proveďte experimenty zaměřené na vyhodnocení přesnosti predikce mluveného slova.

Literatura:

- THEIN, Thein; SAN, Kalyar Myo. Lip movements recognition towards an automatic lip reading system for Myanmar consonants. In: *2018 12th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 2018. p. 1-6.
- DATTA, Soumyya Kanti; JIA, Shan; LYU, Siwei. Exposing Lip-syncing Deepfakes from Mouth Inconsistencies. *arXiv preprint arXiv:2401.10113*, 2024.

Při obhajobě semestrální části projektu je požadováno:
Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Goldmann Tomáš, Ing., Ph.D.**
Vedoucí ústavu:	Kočí Radek, Ing., Ph.D.
Datum zadání:	1.11.2024
Termín pro odevzdání:	14.5.2025
Datum schválení:	31.10.2024

Abstrakt

Cieľom práce bolo navrhnúť systém na rozpoznávanie hovorených slov na základe pohybov pier bez zvukového vstupu a overiť jeho využiteľnosť v reálnej aplikácii. Práca sa zaoberá vizuálnym rozpoznávaním reči, ktoré má potenciál využitia napríklad v asistívnej komunikácii. Navrhnuté riešenie zahŕňa kompletný proces spracovania videí z datasetu LRS2 vrátane detekcie tváre, extrakcie oblasti úst a prípravy dát na trénovanie. Model kombinuje 3D konvolučnú neurónovú sieť, obojsmerné GRU a dekódovanie pomocou CTC a mechanizmu pozornosti. Systém bol nasadený v jednoduchej webovej aplikácii, avšak dosiahnuté výsledky (napr. chybovosť znakov – Character Error Rate – približne 60 %) zatiaľ neumožňujú jeho praktické využitie. Napriek tomu práca predstavuje pevný a funkčný základ pre ďalší výskum. Prínosom je najmä vytvorenie kompletnej architektúry, na ktorej možno ďalej stavať.

Abstract

This thesis aimed to design a system capable of recognising spoken words based solely on lip movements, without relying on audio input. The goal was not only to build such a system but also to test its potential use in a real-world application, such as assistive communication. The solution includes a complete processing pipeline for LRS2 video data, covering face detection, mouth region extraction, and data preparation for model training. The core of the system is a neural network combining 3D convolutions, bidirectional GRUs, and decoding through CTC and attention mechanisms. Although the system was successfully integrated into a simple web application, the achieved performance – characterised by a Character Error Rate of around 60 % – is not yet sufficient for practical use. Still, the work lays a solid foundation for future improvements and provides a complete architecture to build upon.

Kľúčové slová

čítanie z pier, analýza pohybu pier, strojové učenie, počítačové videnie, neurónové siete, konvolučné neurónové siete, rekurentné neurónové siete, kaskádový attention-CTC dekóder, detekcia oblasti úst, LRS2 dataset

Keywords

lip reading, lip movement analysis, machine learning, computer vision, neural networks, convolutional neural networks, recurrent neural networks, cascaded attention-CTC decoder, mouth region detection, LRS2 dataset

Citácia

PESTUN, Matúš. *Aplikace pro odezírání mluveného slova.* Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann, Ph.D.

Aplikace pro odezírání mluveného slova

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Tomáša Goldmanna, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Matúš Pestun
14. mája 2025

Poďakovanie

Rád by som poďakoval vedúcemu mojej bakalárskej práce, Ing. Tomášovi Goldmannovi, PhD., za odbornú pomoc, cenné rady a konštruktívne pripomienky, ktoré mi poskytol počas celého riešenia práce. Poďakovanie patrí aj infraštruktúre MetaCentrum za poskytnutie výpočtových prostriedkov, ktoré boli nevyhnutné na trénovanie neurónových sietí a testovanie výsledného riešenia.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Kapitola 1

Úvod

V súčasnosti sa čoraz častejšie stretávame s technológiami, ktoré umožňujú komunikáciu medzi človekom a počítačom pomocou rôznych zmyslov. Hoci sa väčšina z nás pri komunikácii spolieha najmä na sluch, zrak zohráva nemenej dôležitú úlohu – vďaka nemu napríklad dokážeme odčítať hovorené slová z pohybu pier. Tento jav, známy ako čítanie z pier, je pre ľudí prirodzený, no zároveň náročný. Preto je potrebné navrhnúť počítačový model, ktorý túto výzvu prekoná a bude schopný čítať z pier.

Práve tejto výzve sa venuje predkladaná práca, ktorá sa zaoberá rozpoznávaním hovorených slov výlučne na základe vizuálneho pohybu pier, bez použitia zvukového vstupu. Táto oblasť je významná nielen z pohľadu výskumu, ale aj z praktického hľadiska – môže výrazne pomôcť osobám so sluchovým postihnutím, zlepšiť zabezpečenie systémov pomocou biometrie či umožniť tichú komunikáciu v špecifických podmienkach. Pre mňa osobne je táto téma zaujímavá vďaka prepojeniu medzi obrazovým spracovaním a strojovým učením, ktoré patria medzi moje hlavné študijné záujmy.

Technológia čítania z pier má dlhú históriu – od tradičných metód založených na spracovaní obrazu a sledovaní pohybu pier až po moderné prístupy, ktoré využívajú hlboké neurónové siete. V posledných rokoch sa ukázalo, že práve modely založené na konvolučných a rekurentných neurónových sieťach prinášajú najpresnejšie výsledky. Preto sa práca zameriava predovšetkým na tieto moderné riešenia – ich architektúru, výhody a obmedzenia.

Cieľom práce je vytvoriť model schopný rozpoznávať hovorené slová na základe obrazových sekvencií pier, bez potreby zvukového vstupu. K dosiahnutiu tohto cieľa vedie niekoľko krokov – zber a predspracovanie obrazových dát, kde rečník rozpráva, návrh a trénovanie neurónovej siete, hodnotenie jej presnosti a analýza výsledkov. Súčasťou práce je tiež porovnanie tradičných a moderných metód a diskusia o možných aplikáciách v reálnom svete.

V nasledujúcej kapitole 2 je predstavená samotná problematika čítania z pier – hlavné výzvy, existujúce riešenia a potenciálne aplikácie. Kapitola 3 sa venuje základom strojového učenia a podrobne opisuje konvolučné a rekurentné neurónové siete. Kapitola 4 je zameraná na návrh a implementáciu riešenia; podrobne sa v nej rozoberá predspracovanie vizuálnych dát, architektúra výsledného modelu, spôsob generovania výstupu na základe pohybov pier, ako aj opis finálnej aplikácie, v ktorej je možné riešenie prakticky otestovať. Nasleduje kapitola 5, ktorá prezentuje výsledky testovania finálne navrhnutého modelu a porovnanie rôznych prístupov. Prácu uzatvára kapitola 6, kde sú zhrnuté hlavné prínosy a navrhnuté možnosti ďalšieho vývoja.

Kapitola 2

Problematika čítania z pier

Čítanie z pier patrí medzi zaujímavé a náročné výzvy modernej technológie. Ide o schopnosť rozpoznať hovorené slová iba na základe pohybu pier, bez akéhokoľvek zvukového vstupu. Aj keď sú technológie ako umelá inteligencia a počítačové videnie čoraz vyspelejšie, čítanie z pier zostáva komplikovanou úlohou. Rôznorodosť jazykov, podobnosť pohybov pier pri rôznych zvukoch a ďalšie faktory výrazne sťažujú dosiahnutie vysokej presnosti.

V tejto kapitole sa rozoberá, čo všetko zahŕňa čítanie z pier. Je analyzované, do akej miery je možné rozpoznať hovorené slová len z vizuálnych informácií a identifikujú sa hlavné prekážky, ktoré je potrebné prekonať. Predstavujú sa existujúce riešenia využívané v tejto oblasti, pričom sú zdôraznené aj tradičné metódy, ktorých pochopenie poskytuje dôležitý historický kontext a ukazuje postupný vývoj tejto oblasti. Nakoniec sú naznačené oblasti, v ktorých by technológie čítania z pier mohli nájsť praktické využitie, čím by významne prispeli k zlepšeniu prístupu k informáciám a komunikácii v rôznych sférach života.

2.1 Výzvy a obmedzenia

Ľudská schopnosť čítať slová výlučne na základe pohybu pier je mimoriadne obmedzená [11]. Z tohto dôvodu je dôležité vyvíjať modely schopné presne identifikovať a analyzovať jednotlivé hovorené slová. Miera, do akej je možné hovorené slová rozpoznať, závisí od schopnosti modelov spracovať rôzne výzvy a obmedzenia. Najvýznamnejšie výzvy a obmedzenia vychádzajú zo štúdií [10] a [12].

Jednou z hlavných výziev sú visémy a fonémy. V mnohých jazykoch, vrátane angličtiny, existujú visémy, skupiny fonémov s rovnakým vizuálnym prejavom. Fonémy sú základné zvukové jednotky jazyka, ktoré sa síce zvukovo odlišujú, no pri ich vyslovovaní je tvar a pohyb pier rovnaký. Napríklad fonémy v anglickom jazyku, „*p*“, „*b*“ a „*m*“ patria do jednej visémovej skupiny, pretože ich vyslovovanie spôsobuje identický pohyb pier – 2.1.

Tabuľka 2.1: Štandardná sada visémov špecifikovaná v MPEG-4 a súvisiacich fonémach. Tabuľka popisuje rozdelenie anglických fonémov do visémových skupín, prevzaté z [40].

Viseme Class	**phonemes**	**examples**
0	none	na
1	p, b, m	put, bed, mill
2	f, v	far, voice
3	T, D	think, that
4	t, d	tip, doll
5	k, g	call, gas
6	tS, dZ, S	chair, join, she
7	s, z	sir, zeal
8	n, l	lot, not
9	r	red
10	A:	car
11	e	bed
12	I	tip
13	Q	top
14	U	book

Ďalším podobným obmedzením sú homofóny – celé slová, ktorých zvuk je rozdielny, ale ich vizuálny prejav je identický (vid. obrázok vyššie, stĺpec „example“). Napríklad anglické slová *„mark“*, *„bark“* a *„park“* patria, rovnako ako fonémy, do jednej visémovej skupiny.

Okrem visémov a homofónov presnosť modelov výrazne ovplyvňujú aj faktory ako rýchlosť reči, kvalita artikulácie, prízvuk či vizuálne podmienky vrátane osvetlenia, rozlíšenia obrazu alebo šumu.

2.2 Moderné a tradičné metódy

V oblasti čítania hovorených slov z pier dominujú moderné technológie, predovšetkým hlboké neurónové siete, ktoré svojou schopnosťou spracovávať obraz a chápať kontext reči dosahujú vysokú presnosť [21]. Tradičné metódy, hoci boli v posledných rokoch nahradené týmito pokročilými technológiami, si zaslúžia zmienku nielen pre svoj historický význam, ale aj pre pochopenie základov a vývoja tejto oblasti [17].

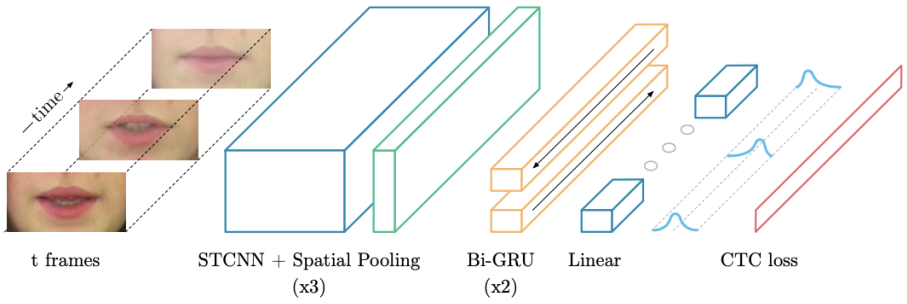
Moderné metódy založené na hlbokých neurónových sieťach

Keďže sa táto práca primárne zameriava na riešenie problému čítania hovorených slov z pier pomocou neurónových sietí, ktoré sa ukázali ako mimoriadne úspešné a efektívne, je vhodné spomenúť existujúce riešenia založené na tejto technológii. Tieto riešenia využívajú moderné prístupy, ktoré kombinujú pokročilé techniky spracovania obrazu a analýzy sekvencií na dosiahnutie vysokej presnosti pri rozpoznávaní hovorených slov.

LipNet

LipNet [6] je prvý end-to-end model navrhnutý špeciálne na čítanie viet z pier pomocou neurónových sietí. Tento model spracováva výlučne vizuálne dáta a nevyžaduje žiadny zvukový vstup.

Fungovanie modelu je založené na kombinácii spatiotemporálnych konvolučných neurónových sietí (STCNN) a obojsmerných GRU (Bi-GRU), čo umožňuje zachytiť priestorové aj časové súvislosti v pohyboch pier. Pri rozpoznávaní sekvencií využíva stratovú funkciu Connectionist Temporal Classification (CTC), ktorá eliminuje potrebu manuálneho zarovnania dát. To znamená, že model nemusí vedieť, ktoré časti videa zodpovedajú konkrétnym slovám – všetko zvládne automaticky. Tento prístup zjednodušuje učenie a zvyšuje presnosť predikcie, vid. popísanú architektúru nižšie 2.1.



Obr. 2.1: Architektúra modelu LipNet zahrňujúca STCNN, Bi-GRU a CTC stratovú funkciu [6].

Model LipNet bol trénovaný na GRID datasete, ktorý obsahuje jednoduché vety s obmedzenou slovnou zásobou. Aj v takto kontrolovanom prostredí model dosiahol výnimočnú presnosť 95,2 %, čím prekonal nielen staršie technológie (86,4 %), ale aj ľudských expertov, ktorí dokázali správne identifikovať len 52,3 % slov.

Jednou z najväčších výhod LipNet je jeho schopnosť spracovávať celé vety, čo minimalizuje nejednoznačnosť medzi visémami. Okrem toho model nevyžaduje žiadne manuálne predspracovanie dát, čo výrazne urýchľuje jeho nasadenie a zjednodušuje používanie.

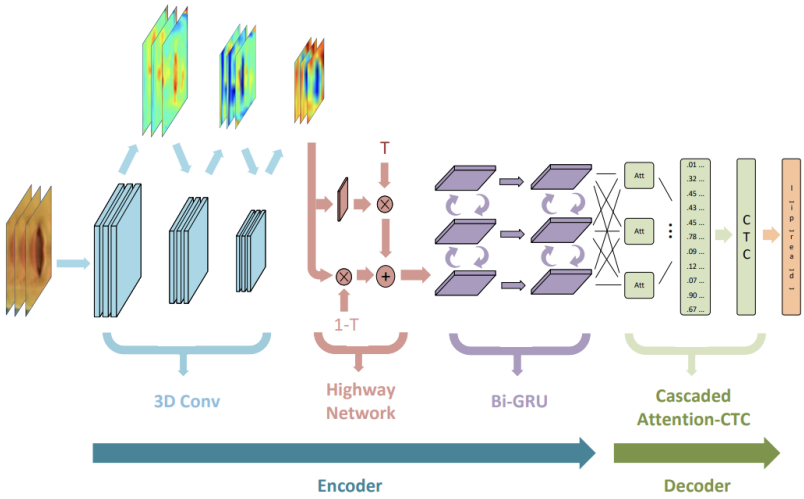
Na druhej strane, model LipNet čelí výzvam. Jeho tréning bol obmedzený na GRID datasete, ktorý má veľmi jednoduchú slovnú zásobu. To môže spôsobiť problémy pri aplikovaní modelu na zložitejšie jazyky alebo reálne konverzačné situácie, kde je potrebné pracovať s oveľa širším spektrom slov a výrazov.

Model LipNet priniesol zásadný prelom v oblasti čítania z pier a položil základ pre vývoj ďalších pokročilých riešení, ktoré stavajú na jeho architektúre a stratégiách učenia.

LCANet

LCANet [38] je moderný end-to-end model na čítanie slov z pier, ktorý využíva výhradne vizuálne dáta a nepotrebuje zvukový vstup. Používa 3D konvolučné neurónové siete (3D-CNN) na zachytenie priestorovo-časových čŕt v pohyboch pier, highway sieť na efektívny prenos informácií medzi jednotlivými vrstvami a obojsmerné GRU (Bi-GRU) na pochopenie dlhodobých časových súvislostí.

Jednou z najväčších výhod LCANet je jeho kaskádový dekodér, ktorý využíva mechanizmus pozornosti a stratovú funkciu Connectionist Temporal Classification (CTC). Tento prístup umožňuje modelu pracovať s kontextom celej sekvencie, čo ho odlišuje od modelov ako LipNet, ktoré analyzujú pohyby pier izolovane. Výsledkom je lepšie rozpoznávanie slov a minimalizovanie chýb.



Obr. 2.2: Architektúra modelu LCANet zahrňujúca 3D-CNN, highway sieť, Bi-GRU a kaskádový dekodér s mechanizmom pozornosti v kombinácii so stratovou funkciou CTC. Obrázok prevzatý z [38].

Model bol testovaný na GRID datasete, ktorý obsahuje vety s obmedzenou slovnou zásobou. Napriek tomu LCANet dosiahol výnimočné výsledky: 1,3 % chybovosť pri znakoch (CER) a 2,9 % chybovosť pri slovách (WER). V porovnaní s LipNet-om priniesol LCANet viac ako 30 % zlepšenie v oboch metrikách.

Medzi hlavné prínosy modelu patrí presnosť a efektívnosť spracovania obrazu. Highway sieť zlepšuje prenos dát medzi vrstvami a kaskádový dekodér dokáže lepšie pochopiť pohyby pier v širšom kontexte. Tento prístup výrazne znižuje chyby spôsobené nesprávnym zarovnaním video snímok.

Napriek týmto výhodám je LCANet vypočtovo náročný, čo môže byť pre niektoré aplikácie obmedzujúce. Napriek tomu predstavuje tento model významný krok vpred v oblasti čítania z pier a nastavuje nový štandard pre budúce riešenia.

Tradičné metódy

Pred nástupom moderných neurónových sietí sa čítanie slov z pier realizovalo tradičnými metódami, ktoré využívali základné techniky spracovania obrazu a sledovania pohybov pier. Hoci tieto prístupy boli nahradené modernejšími metódami, ich pochopenie poskytuje dôležitý historický kontext a ukazuje postupný vývoj tejto oblasti. Na ilustráciu sú uvedené dve konkrétne tradičné riešenia, ktoré reprezentujú typické princípy týchto prístupov.

Prvá tradičná metóda

Táto metóda využíva kombináciu farebnej transformácie, prahovania a algoritmov na sledovanie obrysov pier [33]. Postup zahŕňa tri hlavné kroky:

1. Lokalizácia a segmentácia pier (obr. 2.3) – každý obrázok, reprezentujúci snímku z videa, je spracovaný na identifikáciu oblasti pier. Segmentácia umožňuje izolovať pery od zvyšku obrazu.

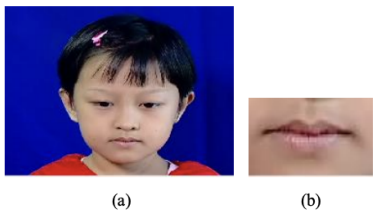


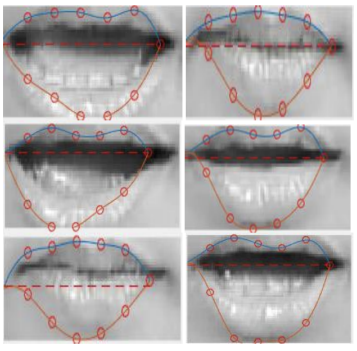
Fig. 2. (a) Original image, (b) Segmented lip region.

Fig. 3. (a) to (j) Number of selected frames for utterance of Tha (one syllable consonants).

Obr. 2.3: Lokalizácia a segmentácia pier (obr. 2) a výsledky segmentácie pier v jednotlivých snímkach videa (obr. 3). Obrázok prevzatý z [33].

2. Prahovanie a izolácia obrysov (obr. 2.4) – farebné transformácie a prahovacie algoritmy sa aplikujú na izolovanú oblasť pier, aby sa zvýraznil ich obrys. Tento obrys sa následne zdokonaľuje pomocou algoritmov, ktoré zlepšujú presnosť identifikácie.

Obr. 2.4: Izolovaný a prahovaný obrys pier [33].



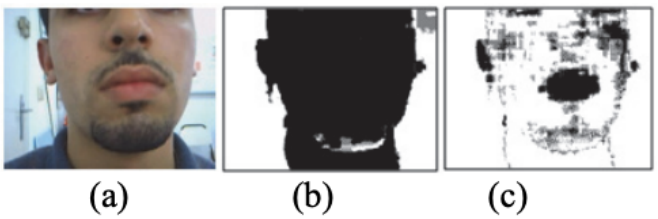
Obr. 2.5: Zdokonalené obrysy pier v snímkach, kde rečník vyslovuje jednu spoluhlásku [33].

3. Extrahovanie vizuálnych čŕt a klasifikácia – zo zdokonalených obrysov pier sa extrahujú kľúčové črty, ako výška pier, šírka pier, a ďalšie parametre. Tieto vlastnosti sa následne použijú ako vstup do lineárneho SVM klasifikačného modelu, ktorý predpovedá vyslovené slovo.

Druhá tradičná metóda

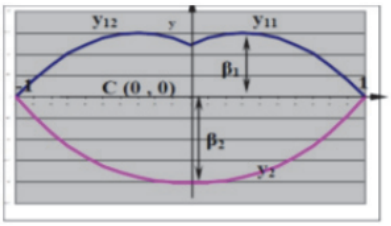
Táto metóda využíva geometrický model pier v kombinácii s prahovaním a časticovými filtrami na sledovanie pohybov pier [18].

1. Lokalizácia a segmentácia pier (obr. 2.6) – obrázky sú spracované pomocou dvoch prahovacích algoritmov. Prvý krok identifikuje oblasť tváre, druhý sa zameriava na presnú lokalizáciu pier v rámci tváre.



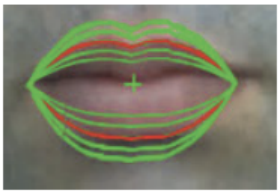
Obr. 2.6: Detekcia tváre (a), prahovanie tváre (b) a lokalizácia pier (c). Obrázok prevzatý z [18].

2. Vytvorenie geometrického modelu (obr. 2.7) – po lokalizácii sa vytvorí geometrický model pier, ktorý je vytvorený z troch kriviek reprezentujúce obrysy horných a dolných pier. Model je flexibilný a umožňuje prispôsobenie tvaru pier pomocou parametrov, ako sú výška, šírka, zakrivenie alebo asymetria pier.



Obr. 2.7: Geometrický model pier vytvorený z troch kriviek [18].

3. Sledovanie pohybu pier (obr. 2.8) – na sledovanie pohybu pier sa využíva časticový filter. Tento filter predikuje nové pozície pier na základe pohybu ich rohov. Najpresnejšia častica aktualizuje stav modelu podľa podobnosti s pôvodným geometrickým modelom. Táto metóda je účinná v dynamických podmienkach a umožňuje presnú lokalizáciu pier.



Obr. 2.8: Sledovanie častíc (zelené) okolo aktuálneho modelu pier (červené). Obrázok prevzatý z [18].

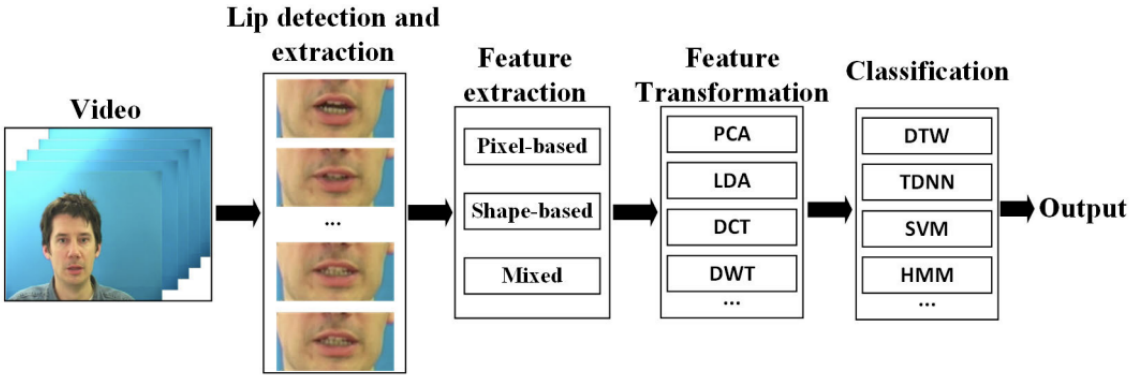
4. Extrahovanie vizuálnych čŕt a klasifikácia – z aktuálnych geometrických modelov v čase t, sa extrahujú dôležité vizuálne črty, ako sú výška pier, šírka pier, stupeň otvorenia úst, celková plocha pier a podobne. Tieto črty sú následne použité ako vstup do lineárneho SVM klasifikátora, ktorý predpovedá, čo bolo rečníkom povedané.

2.3 Porovnanie metód

Všetky metódy riešiace problematiku čítania hovorených slov z pier majú spoločný cieľ – extrahovať vizuálne črty pohybu pier, ktoré umožňujú modelu predikovať a určiť, aké slovo bolo rečníkom povedané. Rozdiel medzi týmito jednotlivými metódami spočíva v spôsobe, akým sú tieto črty extrahované, reprezentované a využívané na finálnu klasifikáciu hovorených slov [4]. Porovnanie tradičných a moderných metód je obsiahnuté v článku [17], z ktorého bola čerpaná väčšina informácií.

Tradičné metódy

Tradičné metódy čítania z pier sa spoliehali na ručne navrhnuté algoritmy a techniky spracovania obrazu. Tieto prístupy zahŕňali lokalizáciu oblasti pier, extrakciu dôležitých vizuálnych čŕt, ako sú pohyby alebo tvar pier, a následnú analýzu týchto čŕt pomocou jednoduchých klasifikačných modelov. Typickým krokom bola redukcia dimenzionality extrahovaných dát s cieľom zjednodušiť ďalšie spracovanie. Celý proces je ilustrovaný na obrázku nižšie 2.9.



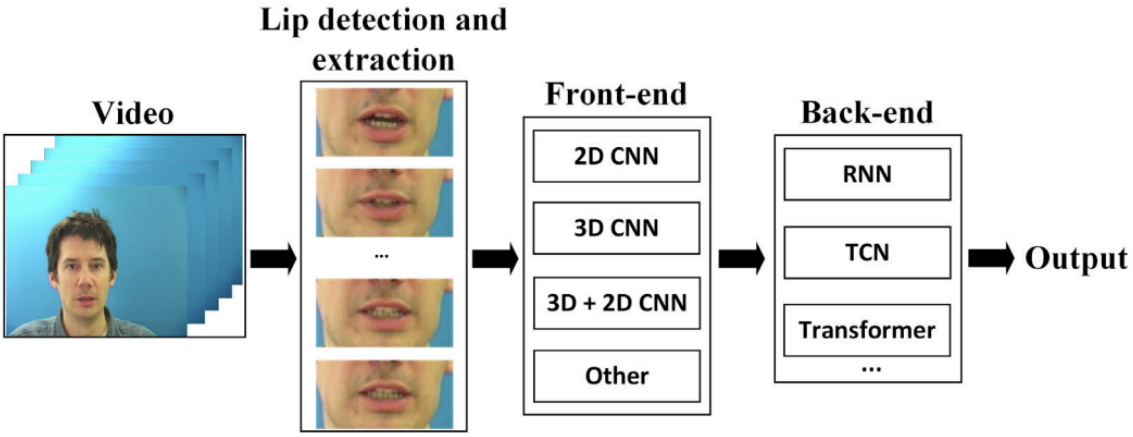
Obr. 2.9: Proces tradičného čítania z pier: Lokalizácia a extrakcia pier, extrakcia dôležitých vizuálnych čŕt z obrazu pier, následná redukcia dimenzionality týchto čŕt a nakoniec klasifikácia pomocou vhodného klasifikátora [17].

Tieto metódy boli limitované manuálnym návrhom algoritmov a nízkou schopnosťou prispôsobiť sa rôznym podmienkam, ako sú zmeny osvetlenia, variácie pohybov pier alebo rozlíšenie videa. Napriek týmto obmedzeniam položili základy pre neskorší vývoj modernejších techník, pričom ich historický význam spočíva v poskytovaní prvých riešení na spracovanie vizuálnej reči.

Moderné metódy založené na neurónových sieťach

Moderné metódy čítania z pier priniesli veľký pokrok v presnosti a spoľahlivosti rozpoznávania hovorených slov z pier. Tieto metódy využívajú hlboké neurónové siete, ako CNN (konvolučné neurónové siete, viac v sekcii 3.3) a RNN (rekurentné neurónové siete, viac v sekcii 3.4), na automatickú extrakciu a analýzu komplexných vizuálnych čŕt priamo z obrazových sekvencií. Na rozdiel od tradičných metód nevyžadujú manuálne návrhy algoritmov na extrakciu čŕt a dokážu analyzovať kontext celej vety. Táto vlastnosť umožňuje minimalizovať chyby pri rozpoznávaní podobných visém, napríklad zvukov „*p*“, „*b*“ a „*m*“.

Tieto modely zahŕňajú proces detekcie tváre a lokalizácie pier, kde sú extrahované črty následne analyzované a použité na klasifikáciu hovorených slov. Pokročilé stratové funkcie, ako Connectionist Temporal Classification (CTC), zabezpečujú, že model nemusí manuálne zarovnávať jednotlivé časti videa s konkrétnymi slovami, čo šetrí čas a zvyšuje efektivitu. Celý proces je ilustrovaný na obrázku nižšie 2.10.



Obr. 2.10: Proces moderného čítania z pier: Najskôr sa lokalizujú a extrahujú pery, následne predná časť modelu (*front-end*) extrahuje časové a priestorové črty. Tieto črty sú potom použité ako vstup do zadnej časti modelu (*back-end*), kde sa spájajú v jednotlivých časových krokoch pre finálnu klasifikáciu. Obrázok prevzatý z [17].

Najväčšou výhodou týchto metód je ich flexibilita. Dokážu sa prispôsobiť rôznym datasetom a podmienkam, ako je šum na pozadí, nekonzistentné osvetlenie alebo rôzne spôsoby pohybov pier. Vďaka tejto univerzálnosti už našli uplatnenie v mnohých oblastiach a majú potenciál rozšíriť svoje využitie v budúcnosti.

2.4 Praktické aplikácie technológií čítania z pier

Modely na čítanie hovorených slov z pier majú široké uplatnenie v rôznych oblastiach, kde je potrebné rozpoznať reč výhradne na základe vizuálnych vstupov.

Informačná bezpečnosť

Tieto technológie môžu výrazne zlepšiť úroveň zabezpečenia. Modely sa dajú využiť na overovanie totožnosti osôb pomocou rozpoznávania pohybu pier, čo posilňuje ochranu zariadení alebo prístupových systémov [25], [36]. Taktiež môžu byť nasadené v kamerových systémoch na analýzu výpovede osôb, čo je užitočné pri vyšetrovaní alebo identifikácii potenciálnych hrozieb [36]. V špecifických situáciách, ako sú vojenské operácie či núdzové stavy, umožňujú tichú komunikáciu, eliminujúc tak zvukové stopy a minimalizujúc riziko odhalenia.

Zdravotníctvo

V zdravotníctve majú modely čítania z pier významný prínos pre osoby s rečovými a sluchovými poruchami:

- Osoby s rečovými poruchami môžu pomocou týchto modelov komunikovať prostredníctvom pohybov pier bez potreby zvukového prejavu, čo nahrádza tradičné zariadenia na generovanie hlasu [28].
- Sluchovo postihnutí môžu využiť tieto technológie ako doplnok na pochopenie hovoreného slova, najmä tam, kde nie je možné použiť znakovú reč. Tieto modely môžu tiež zabezpečiť titulkovanie v reálnom čase v médiách, vzdelávacích inštitúciách alebo na pracovisku [1].

Okrem toho sa modely uplatňujú v rehabilitáciách, kde sledujú pokrok u pacientov so zhoršenou artikuláciou alebo sluchom. Poskytujú presnú spätnú väzbu, ktorá pomáha pacientom zlepšiť kvalitu ich reči alebo schopnosť interpretácie vizuálnej komunikácie.

Vzdelávanie

V školách môžu modely pomôcť deťom so sluchovými alebo rečovými poruchami získať lepší prístup k informáciám. Tieto technológie podporujú inkluzívne vzdelávacie prostredie, kde všetci študenti majú rovnaké možnosti učiť sa a komunikovať.

2.5 Zhrnutie

Čítanie z pier predstavuje fascinujúcu, no náročnú výzvu pre moderné technológie. Táto kapitola sa zaoberala problematikou rozpoznávania hovorených slov na základe pohybu pier a poukázala na hlavné prekážky, ktoré s tým súvisia. Medzi najväčšie prekážky patrí existencia visémov a homofónov, rýchlosť reči, kvalita artikulácie a nepriaznivé vizuálne podmienky, ako sú slabé osvetlenie alebo šum.

V tejto kapitole boli taktiež predstavené niektoré tradičné metódy, ktoré sa zaoberajú touto problematikou. Hoci tieto metódy položili základ pre vývoj tejto technológie, ich možnosti boli obmedzené manuálnymi algoritmami a slabou adaptáciou na rôzne podmienky. Naopak, moderné metódy založené na hlbokých neurónových sieťach, ako sú CNN a RNN,

však priniesli výrazne zlepšenia. Ich schopnosť automaticky extrahovať a analyzovať komplexné vizuálne črty výrazne zlepšila presnosť a spoľahlivosť pri rozpoznávaní hovorených slov.

Kapitola ďalej analyzovala modely LipNet a LCANet, ktoré demonštrujú silu moderných technológií v tejto oblasti. Rovnako zdôrazňovala aj praktické aplikácie čítania z pier, ktoré majú význam v oblastiach, ako sú informačná bezpečnosť, zdravotníctvo a vzdelávanie.

Kapitola 3

Strojové učenie

Čítanie z pier je úzko spojené so strojovým učením, kde kľúčovú úlohu zohrávajú neurónové siete. Tieto siete umožňujú analyzovať zložité vzťahy medzi vstupmi a výstupmi a následne predikovať alebo klasifikovať dáta. Pre úlohu čítania slov z pier bez zvukových informácií sú najvhodnejšie konvolučné a rekurentné neurónové siete, pretože dokážu efektívne spracovávať obrazové a sekvenčné dáta.

Táto kapitola stručne vysvetľuje základné princípy strojového učenia a jeho hlavné kategórie. Následne sa zameriava na neurónové siete, ich štruktúru a spôsob fungovania. Pri konvolučných sieťach je vysvetlené, ako tieto siete pomocou konvolučných a pooling vrstiev extrahujú dôležité vizuálne črty z obrazových dát, napríklad zo snímok pier. Pri rekurentných sieťach sa objasňuje ich schopnosť spracovávať sekvencie a uchovávať informácie o predchádzajúcich krokoch, čo umožňuje modelu pochopiť celkový význam sekvencie, napríklad pohyby pier. Bližšie sú predstavené varianty týchto sietí – LSTM a GRU, ktoré sú vhodné na prácu s dlhými sekvenciami, keďže lepšie zvládajú dlhodobé závislosti.

Na záver kapitoly sú popísané dáta, s ktorými sa bude pracovať, vrátane predtréningovej, tréningovej, validačnej a testovacej množiny, spolu s vysvetlením ich významu. Zdôrazňuje sa dôležitosť správneho predspracovania dát a predstavujú sa techniky augmentácie, ktoré zohrávajú kľúčovú úlohu pri zvyšovaní presnosti a spoľahlivosti modelu.

3.1 Základy strojového učenia

Základy strojového učenia boli prevzaté z kníh [34] a [8]. Strojové učenie (*Machine learning*) je podmnožinou umelej inteligencie, ktorá umožňuje počítačom „učiť sa“ z dát a riešiť úlohy bez potreby explicitného naprogramovania každého kroku. Počítače využívajú algoritmy a matematické modely na analýzu dát, hľadanie vzorcov a predpovedanie budúcich výsledkov na základe už dostupných informácií.

Jednoducho povedané, cieľom strojového učenia je vytvoriť systémy schopné samostatne získavať poznatky z dát a následne tieto poznatky použiť na predpovede alebo rozhodovanie. Existujú štyri hlavné kategórie strojového učenia:

- Učenie s učiteľom (*Supervised learning*) – model sa učí na označených dátach, kde každý vstup má priradený výstup. Cieľom je správne predikovať výstupy pre nové vstupy. Do tohto typu učenia patria aj neurónové siete, ktoré sú ďalej popísané v 3.2.
- Učenie bez učiteľa (*Unsupervised learning*) – model pracuje s neoznačenými dátami a snaží sa nájsť vzory alebo štruktúry bez vopred definovaných výstupov.

- Kombinácia učenia s učiteľom a bez učiteľa (*Semi-supervised learning*) – kombinuje malú časť označených dát s veľkou časťou neoznačených dát, čím zlepšuje efektivitu učenia.
- Učenie formou odmeňovania (*Reinforcement learning*) – model, tzv. agent, sa učí vykonávať akcie na základe odmien a trestov, pričom cieľom je maximalizovať získané odmeny.

3.2 Neurónové siete

Základná teória neurónových sietí je spracovaná na základe literatúry [34], [3]. Neurónové siete tvoria základ mnohých moderných aplikácií umelej inteligencie a strojového učenia. Ich princíp je inšpirovaný fungovaním ľudského mozgu.

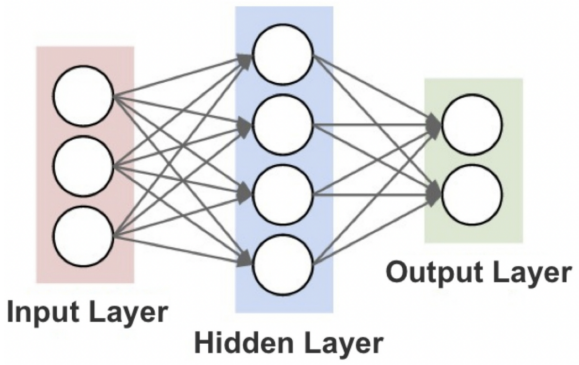
Táto sekcia sa zameriava na vysvetlenie základných pojmov neurónových sietí, ako sú umelé neuróny, aktivačné a stratové funkcie, pričom zdôrazňuje ich význam. Ďalej sa venuje procesu trénovania týchto modelov, dôležitosti hyperparametrov a metódam regularizácie na zlepšenie ich schopnosti generalizovať.

Neurónové siete sú kľúčové pre riešenie problému čítania slov z pier na základe pohybu, preto je nevyhnutné objasniť ich princípy a základné koncepty.

Vrstvy neurónových sietí

Základným stavebným prvkom neurónových sietí sú ich vrstvy. Neurónové siete pozostávajú z troch základných vrstiev (vid. obrázok 3.1):

1. Vstupná vrstva – prijíma dáta v surovom formáte, napríklad obrázky alebo texty.
2. Skryté vrstvy – identifikujú a spracovávajú vzory vo vstupných dátach.
3. Výstupná vrstva – generuje konečné výsledky, ako sú predikcie alebo klasifikácie.



Obr. 3.1: Tri základné vrstvy neurónových sietí [34]. Krúžky reprezentujú neuróny v jednotlivých vrstvách. Šipky reprezentujú vstupy, kde každý vstup ma svoju váhu.

Umelý neurón

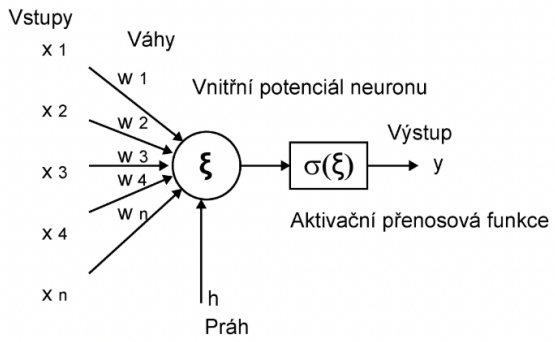
Umelý neurón je základná stavebná jednotka neurónových sietí. Jeho úlohou je prijímať a spracovávať vstupy. Výsledkom spracovania je hodnota, ktorá sa prenáša do ďalších neuró-

nov v sieti. Tento mechanizmus umožňuje neurónom učiť sa, rozpoznávať vzory a prispôsobovať sa rôznym úlohám, ako sú klasifikácia, predikcia či rozpoznávanie obrazov.

Každý neurón pozostáva z troch hlavných častí: vstupy a váhy, lineárna kombinácia vstupov a aktivačná funkcia. Každý vstup $(x_1, \cdots, x_n)$ sa násobí príslušnou váhou $(w_1, \cdots, w_n)$, ktorá určuje jeho dôležitosť, k výsledku sa taktiež pridá aj prahová hodnota (h). Výsledok lineárneho výpočtu

$$\xi = \sum_{i=1}^{n} w_i x_i + h \tag{3.1}$$

sa spracuje aktivačnou funkciou (σ), ktorá rozhoduje, či sa neurón aktivuje. Výstup neurónu (y), označovaný ako aktivácia, je výsledkom aplikácie aktivačnej funkcie.



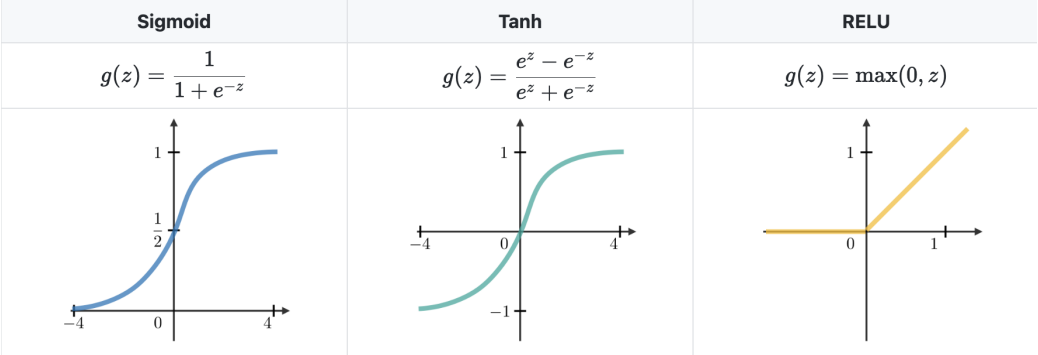
Obr. 3.2: Model umelého neurónu [24].

Aktivačné funkcie

Aktivačné funkcie sú neoddeliteľnou súčasťou neurónových sietí, pretože pridávajú nelinearitu, ktorá umožňuje modelu učiť sa komplexné vzory v dátach.

Bez použitia nelineárnych aktivačných funkcií by všetky vrstvy siete vykonávali iba jednoduché lineárne transformácie. To by zásadne obmedzilo schopnosť siete zachytiť zložité vzťahy medzi vstupnými a výstupnými dátami. Medzi najčastejšie používané a najzákladnejšie aktivačné funkcie patria (vid. aj obrázok 3.3):

- Sigmoida (*sigmoid*) – transformuje výstupy do rozsahu (0, 1), čo je užitočné pre interpretáciu pravdepodobností
- Hyperbolický tangens (*tanh*) – normalizuje výstupy do rozsahu (-1, 1), čo je vhodné pri dátach obsahujúcich kladné aj záporné hodnoty.
- ReLU (Rectified Linear Unit) – je jednoduchá a efektívna funkcia často používaná v hlbokých sieťach. Vracia 0 pre záporné vstupy a pôvodnú hodnotu pre kladné.



Obr. 3.3: Najčastejšie používané aktivačné funkcie [5].

Trénovanie neurónových sietí

Trénovanie neurónových sietí je iteratívny proces, ktorého cieľom je nastaviť váhy tak, aby výstupy modelu čo najpresnejšie zodpovedali očakávaným hodnotám pre danú úlohu. Tento proces zahŕňa nasledujúce kľúčové kroky:

1. Inicializácia váh – váhy medzi neurónmi sa na začiatku nastavujú náhodnými hodnotami. Výber vhodnej inicializácie je dôležitý, aby sa predišlo problémom ako sú vanishing alebo exploding gradienty.

2. Forward pass – vstupy prechádzajú sieťou, kde sa postupne aplikujú váhy, prahové hodnoty a aktivačné funkcie na výpočet výstupov pre jednotlivé vrstvy a nakoniec celej siete.

3. Výpočet chyby – výsledky modelu sa porovnajú s očakávanými výstupmi pomocou stratovej funkcie. Viac o stratových funkciách je v podsekcii 3.2.

4. Backpropagation – pomocou backpropagation sa vypočítajú gradienty chyby smerom od výstupnej vrstvy ku vstupnej pre každú váhu siete a pre každý práh.

5. Aktualizácia váh – váhy sa upravujú pomocou gradientného zostupu alebo jeho variantov (napr. Adam, RMSprop), aby sa minimalizovala chyba.

6. Opakovanie procesu – kroky 2 až 5 sa iteratívne opakujú, kým model nedosiahne požadovanú presnosť alebo kým hodnota chyby prestane klesať.

Stratové funkcie

Stratové funkcie sú nevyhnutnou súčasťou trénovania neurónových sietí, pretože určujú, ako dobre sa výstupy modelu zhodujú s očakávanými hodnotami. Miera chyby vyjadrená stratovou funkciou určuje, ako dobre model plní danú úlohu. Nižšia hodnota stratovej funkcie znamená lepšie predikcie. Výber stratovej funkcie závisí od typu úlohy a povahy dát. Najčastejšie používané stratové funkcie boli prevzaté z [15]:

- Mean Squared Error (MSE) – používa sa v regresných úlohách.
- Mean Absolute Error (MAE) – rovnako ako MSE sa používa pri regresných úlohách. Je menej citlivá na veľké odľahlé hodnoty než MSE.

- Categorical Cross-Entropy – používa sa v klasifikačných úlohách s viacerými triedami.
- Huber Loss – kombinuje výhody MSE a MAE, čím je robustnejšia voči odľahlým hodnotám. Chyba je kvadratická pre malé odchýlky a lineárna pre veľké odchýlky.

Hyperparametre

Hyperparametre [30] sú nastaviteľné parametre, ktoré sa určujú pred začiatkom trénovania a významne ovplyvňujú presnosť modelu a jeho schopnosť generalizovať. Na rozdiel od váh sa počas trénovania nemenia.

- Rýchlosť učenia (*Learning rate*) – určuje veľkosť krokov, ktorými sa aktualizujú váhy počas trénovania. Vysoká hodnota môže viesť k nestabilite, nízka k pomalému učeniu. Bežnou praxou je začať s relatívne vysokou rýchlosťou učenia a postupne ju znižovať.
- Počet epoch – počet kompletných prechodov dát cez sieť. Príliš veľa epoch vedie k pretrénovaniu, málo k podtrénovaniu (viac informácií o pretrénovaní a podtrénovanie je v podsekcii 3.2).
- Veľkosť dávky (*Batch size*) – udáva, koľko vstupných dát sa spracováva naraz pri jednom tréningovom alebo validačnom kroku. Bežne používané hodnoty sú 32, 64, 128 alebo 256.
- Počet skrytých vrstiev a neurónov – určuje hĺbku a šírku siete. Viac vrstiev a neurónov môže zvýšiť presnosť, ale zároveň riziko pretrénovania.

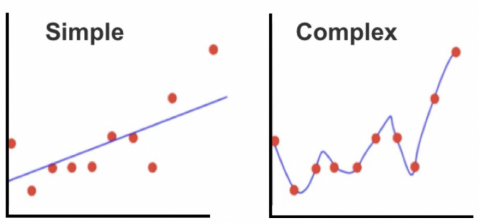
Pri trénovaní neurónových sietí je experimentovanie a optimalizácia hyperparametrov kľúčové pre dosiahnutie robustného a presného modelu.

Bias/Variance problém

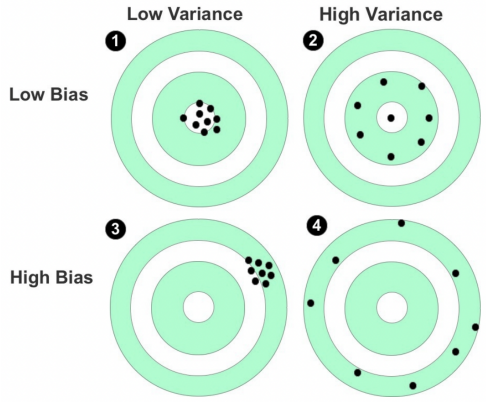
Pri trénovaní modelov je dôležité nájsť rovnováhu medzi pretrénovaním a podtrénovaním. Tieto pojmy vyjadrujú schopnosť modelu generalizovať, teda správne predikovať výsledky na nových dátach.

Pretrénovanie (*overfitting*) nastáva, keď sa model príliš prispôsobí trénovacím dátam, vrátane šumu a irelevantných vzorov, čo znižuje jeho schopnosť generalizovať. Taký model má vysokú presnosť na trénovacích dátach, no jeho výkon na nových dátach je slabý. Pretrénovaný model má nízky bias, ale vysokú varianciu.

Podtrénovanie (*underfitting*) znamená, že model je príliš jednoduchý na zachytenie vzorov, čo vedie k nízkej presnosti na trénovacích aj validačných dátach. Podtrénovaný model má vysoký bias a nízku varianciu.



Obr. 3.4: Podtrénovanie (naľavo) a pretrénovanie (napravo) [34].



Obr. 3.5: Terče ilustrujúce rozdiely medzi bias a varianciou [34].

Cieľom je, aby model dosiahol nízky bias aj nízku varianciu, čo mu umožní správne generalizovať. To je možné dosiahnuť nasledujúcimi metódami:

- Regularizácia – obmedzuje pretrénovanie pridaním penalizácií do tréningu modelu (viac v podsekcii 3.2).
- Optimalizácia modelu – výber vhodnej architektúry a správne nastavenie hyperparametrov.
- Zväčšenie datasetu – viac dát umožňuje modelu naučiť sa relevantné vzory a zlepšiť generalizáciu.

Regularizácia

Regularizácia je kľúčová technika používaná na zlepšenie generalizácie modelu tým, že znižuje riziko pretrénovania. Regularizácia pridáva do trénovania obmedzenia, ktoré podporujú vytváranie jednoduchších a robustnejších modelov. Bežne používané metódy regularizácie zahŕňajú:

- Dropout – táto metóda náhodne vypína (nastavuje na nulu) určité neuróny počas trénovania. Dropout núti model nespoliehať sa iba na konkrétne neuróny alebo ich kombinácie, čím vytvára robustnejšie reprezentácie dát. Tento prístup znižuje riziko pretrénovania a zlepšuje generalizáciu modelu.
- L1 a L2 regularizácia – tieto techniky penalizujú veľké váhy, keďže veľké váhy sú väčšinou dôsledkom pretrénovania.
- Predčasné zastavenie (*Early stopping*) – táto metóda sleduje výkon modelu počas trénovania a zastaví učenie, keď presnosť modelu prestane rásť alebo začne klesať. Predčasné zastavenie pomáha zabrániť pretrénovaniu a zbytočnému zvyšovaniu zložitosti modelu.

3.3 Konvolučné neurónové siete

Koncept konvolučných neurónových sietí vychádza prevažne z literatúry [3]. Konvolučné neurónové siete (CNN, z angl. *Convolutional Neural Networks*) patria medzi najpoužívanejšie

typy hlbokých neurónových sietí, navrhnuté na efektívne spracovanie dát s priestorovou štruktúrou, ako sú obrazy a videá. Ich hlavnou výhodou je schopnosť automaticky extrahovať významné vizuálne črty bez potreby ručného výberu príznakov.

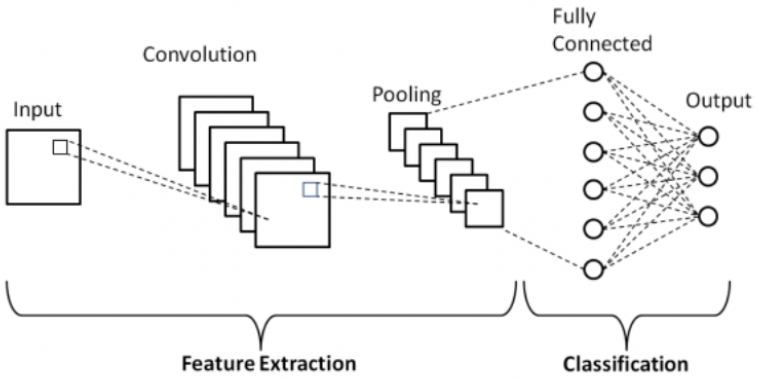
Táto sekcia popisuje základnú architektúru CNN, zahŕňajúcu konvolučné vrstvy s operáciami (stride, padding), poolingové vrstvy a plne prepojené vrstvy. Stručne je predstavený aj koncept 3D-CNN a známe architektúry CNN.

Vzhľadom na to, že práca sa zameriava na rozpoznávanie slov z pier bez zvukových informácií, CNN zohráva kľúčovú úlohu pri riešení tohto problému. Na správne rozpoznanie slov je potrebné extrahovať vizuálne črty, ako sú pohyby a tvar pier.

Základná architektúra konvolučných neurónových sietí

Architektúra CNN sa skladá z dvoch hlavných častí (ilustrovaná na obrázku 3.6):

1. Konvolučná časť – slúži na extrakciu vizuálnych čŕt zo vstupných dát. Túto extrakciu zabezpečujú nasledujúce vrstvy:
   - Konvolučná vrstva 3.3 – aplikuje sadu filtrov (*kernels*) na vstupné dáta s cieľom extrahovať špecifické lokálne črty, ako sú hrany alebo textúry. Táto vrstva zahŕňa operácie ako padding alebo stride.
   - Pooling vrstva 3.3 – znižuje rozmery výstupu konvolučnej vrstvy a tým redukuje počet parametrov siete.
2. Plne prepojená vrstva 3.3 – pozostáva z neurónov, ktoré prijímajú výstup z konvolučnej časti vo forme vektora a vytvárajú finálnu predikciu alebo klasifikáciu.



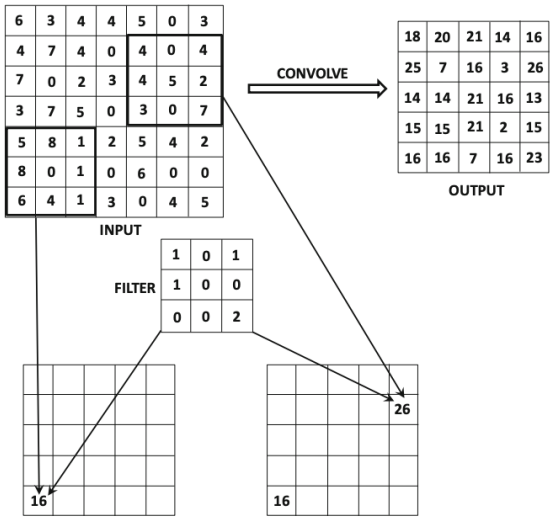
Obr. 3.6: Architektúra konvolučných neurónových sietí [29].

Konvolučná vrstva

Konvolučná vrstva (*Convolutional layer*) zabezpečuje extrakciu vizuálnych čŕt z obrazu. Táto vrstva využíva filtre (*kernels*), ktorými sa vykonáva konvolučná operácia (vid. 3.7) na vstupné dáta, čím dochádza k identifikácii dôležitých prvkov obrazu. Okrem toho používa techniky, ako sú padding a stride. Padding zabraňuje strate dôležitých informácií na okrajoch obrazu, zatiaľ čo stride umožňuje efektívne zmenšenie výstupu a tým znižuje výpočtovú náročnosť spracovania.

Po aplikácii konvolučnej operácie sa typicky používa ReLU (*Rectified Linear Unit*) ako aktivačná funkcia, ktorá nelineárne transformuje výstupy. ReLU zavádza do modelu nelinearitu, čím umožňuje učenie zložitejších vzťahov v dátach.

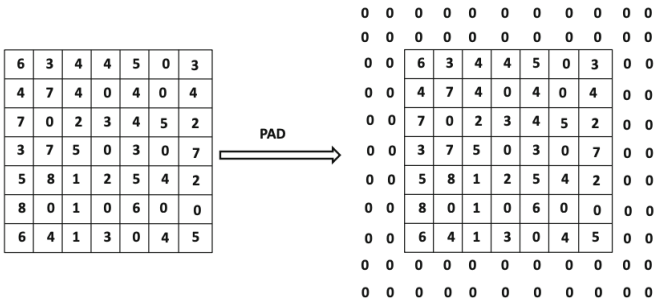
V konvolučnej časti siete sa zvyčajne nachádza niekoľko takýchto vrstiev, ktoré spolu tvoria základnú štruktúru CNN.



Obr. 3.7: Príklad konvolučnej operácie medzi vstupom s rozmermi $7 \times 7 \times 1$ a filtrom s rozmermi $3 \times 3 \times 1$. Rozmer filtra alebo vstupu je definovaný ako $f \times f \times c$, kde c predstavuje počet kanálov a $f \times f$ je veľkosť filtra alebo vstupu. V tomto príklade je počet kanálov 1, čo znamená, že obraz je šedotónový. Pre RGB obraz by bol počet kanálov 3. Obrázok prevzatý z [3].

Padding

Padding je technika, pri ktorej sa pridávajú nulové hodnoty okolo okrajov vstupných dát, aby sa zabránilo strate informácií na hranách obrazu počas konvolučnej operácie. Použitím paddingu sa zabezpečí, že aj črty blízko okrajov obrazu budú spracované rovnako ako črty v jeho strede. Obrázok nižšie 3.8 ilustruje túto techniku.



Obr. 3.8: Príklad techniky padding, kde padding je 2 [3].

Stride

Stride určuje veľkosť kroku, o ktorý sa filter posúva po obraze počas konvolučnej operácie. Štandardná hodnota stride je 1, čo znamená, že filter sa posúva o jeden pixel naraz. Zvýšením hodnoty stride sa výstupný rozmer zmenší, čo vedie k zníženiu výpočtovej náročnosti siete.

Použitie väčšej hodnoty stride je vhodné v prípadoch, keď je potrebné zrýchliť spracovanie údajov alebo zmenšiť počet parametrov siete bez výraznej straty informácií. Typické hodnoty stride sú 1 alebo 2, v závislosti od požadovaného rozlíšenia výstupu.

Pooling vrstva

Pooling vrstva nasleduje po konvolučnej vrstve a jej úlohou je zmenšiť rozmery výstupu pri zachovaní najdôležitejších informácií. Tým sa znižuje počet parametrov modelu, zvyšuje výpočtová efektivita, znižuje riziko pretrénovania a zlepšuje generalizácia na nové dáta. Medzi najčastejšie používané techniky pooling-u patria:

- Max-pooling – vyberá maximálnu hodnotu z každej oblasti vstupu, čím zachováva najvýraznejšie črty.
- Average-pooling – počíta priemernú hodnotu z každej oblasti vstupu. Používa sa menej často, pretože nezachováva ostré črty tak efektívne ako max-pooling.

Pooling vrstva nevyžaduje žiadne trénovacie parametre a slúži len na zjednodušenie výstupu z predchádzajúcej vrstvy. Pri návrhu CNN sa odporúča, aby počet konvolučných vrstiev bol rovnaký alebo väčší ako počet pooling vrstiev.

Plne prepojená vrstva

Plne prepojená vrstva (*Fully connected layer*) prijíma výstup z konvolučnej časti vo forme jednorozmerného vektora, ktorý následne spracováva prostredníctvom neurónov, spojených so všetkými neurónmi predchádzajúcej vrstvy. Úlohou tejto vrstvy je kombinovať extrahované črty a vytvoriť konečný výstup, napríklad vo forme predikcie alebo klasifikácie. Na transformáciu výstupov neurónov sa v plne prepojenej vrstve používajú rôzne aktivačné funkcie, ako softmax, sigmoida alebo lineárna funkcia.

Aj keď plne prepojené vrstvy poskytujú vysokú flexibilitu, ich použitie znamená veľký počet parametrov, čo môže zvýšiť riziko pretrénovania modelu. Na zmiernenie tohto problému sa často používajú techniky regularizácie 3.2.

V krátkosti o 3D-CNN

3D konvolučné neurónové siete (3D-CNN) [16] sú rozšírením klasických CNN, kde filtre vykonávajú operácie v troch rozmeroch (šírka, výška a hĺbka). Tento prístup umožňuje spracovávať dáta s priestorovo-časovou štruktúrou, ako sú videá alebo medicínske 3D snímky (napr. MRI, CT). Vďaka tretej dimenzii dokážu 3D-CNN lepšie identifikovať komplexné vzory a jemné detaily, ktoré by klasická CNN mohla prehliadnuť. To vedie k vyššej presnosti pri analýze dát, kde čas alebo hĺbka zohrávajú kľúčovú úlohu.

3D-CNN sa často využívajú v oblastiach ako analýza videí, autonómne riadenie alebo medicínske zobrazovanie. Nevýhodou 3D-CNN oproti klasickým CNN je ich vyššia výpočtová náročnosť, čo zahŕňa náročnejšiu tvorbu modelu, dlhší čas tréningu a vyššie požiadavky na hardvér.

Známe architektúry konvolučných neurónových sietí

Niekoľko známych architektúr konvolučných neurónových sietí významne prispelo k pokroku v oblasti spracovania obrazových dát. Medzi najdôležitejšie patrí:

- LeNet – jedna z prvých CNN, navrhnutá na rozpoznávanie rukou písaných číslic. Skladá sa zo striedania konvolučných a pooling vrstiev a niekoľkých plne prepojených vrstiev. Preukázala, že CNN dokážu automaticky extrahovať črty bez ručného návrhu. Napriek úspechu bola vhodná len na jednoduchšie úlohy a dáta s nižším rozlíšením.
- AlexNet – víťaz súťaže ILSVRC 2012, ktorá spopularizovala hlboké CNN a využitie GPU na rýchly tréning. Zaviedla ReLU ako aktivačnú funkciu a dropout na regularizáciu. AlexNet umožnila výrazne lepšiu presnosť pri rozpoznávaní farebných obrazov.
- ResNet – architektúra využívajúca reziduálne bloky, ktoré umožňujú efektívny tréning veľmi hlbokých sietí (150+ vrstiev) bez problému miznúceho gradientu. Skip connections (priamy prenos vstupu k výstupu) zlepšili konvergenciu modelu a umožnili dosiahnutie výsledkov porovnateľných s ľudskou presnosťou pri klasifikácii obrazov.

Tieto architektúry zohrali kľúčovú úlohu pri vývoji CNN a sú základom moderných modelov používaných v súčasných aplikáciách strojového učenia.

3.4 Rekurentné neurónové siete

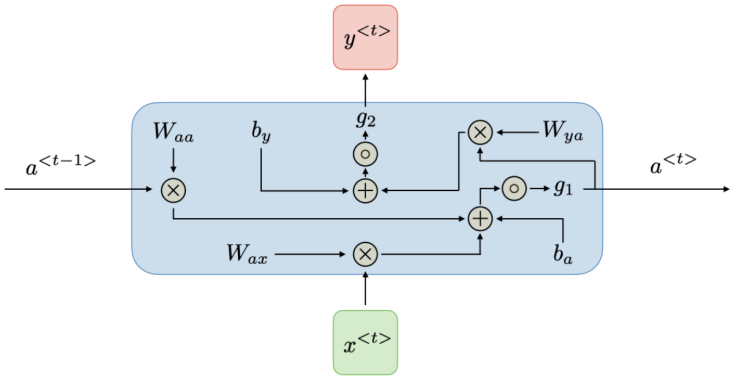
Základy rekurentných neurónových sietí vychádzajú prevažne z [3], [5]. Rekurentné neurónové siete (RNN, z angl. *Recurrent Neural Networks*) sú navrhnuté na spracovanie sekvenčných údajov, kde je potrebné zachovať informáciu o poradí jednotlivých vstupov – záleží na poradí v sekvencii. Používajú sa pri úlohách ako spracovanie prirodzeného jazyka (preklad, generovanie textu), rozpoznávanie reči, transkripcia hovoreného slova či predikcia časových radov.

Táto sekcia popisuje základy RNN architektúr vrátane rekurentných neurónov, typov RNN a metód zakódovania slov. Stručne je vysvetlený význam obojsmerných RNN a predstavené sú varianty LSTM a GRU, ktoré riešia problémy miznúcich a explodujúcich gradientov.

Keďže cieľom tejto práce je čítanie z pier bez zvukových informácií, spracovanie sekvencií obrazových dát zohráva dôležitú úlohu. Každý pohyb pier tvorí časť sekvencie, ktorá vytvára celé slovo alebo vetu. Význam pohybu je možné pochopiť len v súvislosti s predchádzajúcimi a nasledujúcimi pohybmi, preto je potrebné spracovávať nielen jednotlivé snímky, ale aj dynamické prechody medzi nimi, čo zlepšuje presnosť rozpoznania celej sekvencie.

Rekurentný neurón

Základným prvkom RNN je rekurentný neurón (vid. 3.9), ktorý spracováva sekvenčné údaje tým, že si medzi jednotlivými časovými krokmi odovzdáva informácie. Vďaka tomu dokáže model uchovať kontext predchádzajúcich vstupov, čo je kľúčové pri úlohách, kde záleží na poradí prvkov v sekvencii. Rekurentný neurón pozostáva z troch hlavných častí: vstupy, výpočet a výstupy.



Obr. 3.9: Rekurentný neurón [5].

Vstupy

Rekurentný neurón prijíma dva vstupy: aktuálny prvok sekvencie ($x^{<t>}$) a skrytý stav z predchádzajúceho kroku ($a^{<t-1>}$). Aktuálny prvok sekvencie predstavuje vstupný údaj pre časový krok t, pričom ide o zakódované slovo zo slovníka (viac o zakódovaní slov v podsekcii 3.4). Skrytý stav z predchádzajúceho kroku nesie informácie o všetkých predchádzajúcich vstupoch.

Výpočet

V každom časovom kroku sa vypočíta nový skrytý stav ($a^{<t>}$) kombináciou aktuálneho vstupu ($x^{<t>}$) a skrytého stavu z predchádzajúceho kroku ($a^{<t-1>}$). Výstup ($y^{<t>}$) sa následne vypočíta na základe nového skrytého stavu.

$$a^{<t>} = g1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$
$$y^{<t>} = g2(W_{ya}a^{<t>} + b_y)$$

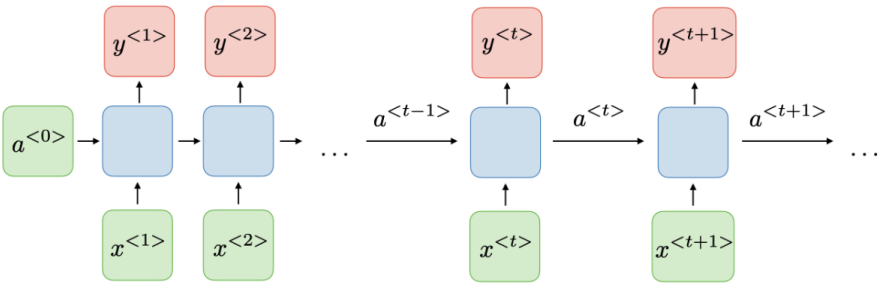
kde W_{aa}, W_{ax} a W_{ya} sú váhové matice, b_a a b_y sú prahové hodnoty, a $g1$, $g2$ sú nelineárne aktivačné funkcie (napr. tanh, ReLU, softmax).

Výstupy

Výstupom rekurentného neurónu je nový skrytý stav ($a^{<t>}$), ktorý sa posúva ako vstup do ďalšieho časového kroku, a finálna predikcia ($y^{<t>}$), reprezentovaná ako vektor pravdepodobností, z ktorého sa vyberie najpravdepodobnejšie slovo.

Základná architektúra RNN

Základná architektúra RNN pozostáva z niekoľkých rekurentných neurónov, kde počet neurónov závisí od dĺžky spracovávanej sekvencie. Obrázok nižšie zobrazuje tuto architekturu 3.10.

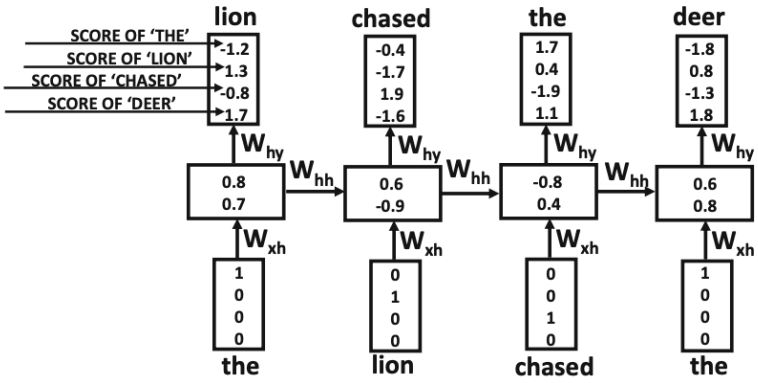


Obr. 3.10: Architektúra RNN [5].

Na lepšie pochopenie tohto princípu uvažujme anglickú vetu:

The lion chased the deer.

Táto veta pozostáva zo štyroch jedinečných slov: {„the“, „lion“, „chased“, „deer“}. Model spracováva vetu po jednotlivých slovách, pričom sa v každom kroku snaží predpovedať nasledujúce slovo. Na obrázku 3.11 je znázornený proces, kde každý neurón prijíma zakódované slovo, vypočíta pravdepodobnosti pre všetky slová v slovníku a vyberie slovo s najvyššou pravdepodobnosťou ako ďalší výstup. Tento postup sa opakuje, kým nie je spracovaná celá sekvencia.



Obr. 3.11: Architektúra RNN pre sekvenciu „The lion chased the deer“ [3].

Typy rekurentných neurónových sietí

RNN sa delia na štyri hlavné typy podľa počtu vstupov a výstupov siete [14]:

1. One-to-One – najjednoduchší model, kde jeden vstup zodpovedá jednému výstupu, napríklad pri klasifikácii obrázkov.
2. One-to-Many – model prijíma jeden vstup a generuje viacero výstupov. Využíva sa pri úlohách, ako je generovanie popisov obrázkov.
3. Many-to-One – sieť prijíma sekvenciu vstupov a vytvára jeden výstup. Používa sa pri analýze sentimentu, kde model dostane vetu a určí, či je pozitívna, negatívna alebo neutrálna alebo pri klasifikácii textu.

4. Many-to-Many – prijíma sekvenciu vstupov a generuje sekvenciu výstupov. Príkladom je strojový preklad, kde každé slovo vstupnej sekvencie vedie k vytvoreniu výstupnej sekvencie v inom jazyku.

Zakódovanie slov

Aby RNN modely mohli spracovávať slová, je potrebné ich previesť do číselnej podoby. Na tento účel sa používa tzv. zakódovanie slov – technika, ktorá každému slovu v slovníku priradí číselnú reprezentáciu. V tejto práci bola zvolená tokenizácia, pretože ide o jednoduchý, výpočtovo nenáročný a priamočiary spôsob, ktorý dobre funguje s rekurentnými architektúrami. Navyše je ľahko aplikovateľný na vstupné dáta reprezentované ako sekvencia čísel, čo zjednodušuje následné spracovanie.

Samozrejme, okrem tokenizácie existujú aj iné prístupy, napríklad one-hot encoding, kde je každé slovo reprezentované dlhým binárnym vektorom, alebo word embeddings, ktoré slová mapujú na husté vektory so zachytenými významovými súvislosťami. Tieto metódy dokážu vystihnúť vzťahy medzi slovami, no sú náročnejšie na pamäť, výpočtový výkon alebo si vyžadujú samostatný tréning.

Tokenizácia

Jednou z najjednoduchších metód zakódovania slov je priradenie unikátneho číselného identifikátora (tzv. tokenu) každému slovu alebo znaku v slovníku [32]. Tento prístup sa nazýva tokenizácia alebo indexové zakódovanie. Každé slovo alebo znak je nahradené celým číslom, ktoré reprezentuje jeho pozíciu v slovníku.

Napríklad pre slovník obsahujúci slová „pero“, „fixka“ a „ceruzka“ by priradenie tokenov mohlo vyzerať takto:

- pero: 1,
- fixka: 2,
- ceruzka: 3.

Tokenizácia je efektívna a dobre škáluje pri práci s veľkým množstvom dát. Hoci samotné číselné reprezentácie nenesú žiadnu informáciu o význame slov, model si môže významové vzťahy osvojiť počas tréningu. V kombinácii s ďalšími vrstvami neurónovej siete sa tak aj z obyčajných tokenov môže naučiť zachytávať sémantiku a kontext.

V krátkosti o obojsmerných RNN

Obojsmerné rekurentné neurónové siete (Bi-RNN, z angl. *Bidirectional Recurrent Neural Networks*) sú vylepšenou verziou klasických RNN, ktoré dokážu spracovávať sekvencie oboma smermi – nielen od začiatku ku koncu, ale aj opačne, teda od konca k začiatku.

To umožňuje modelu lepšie pochopiť kontext, pretože význam prvku v sekvencii môže závisieť nielen od predchádzajúcich, ale aj od nasledujúcich prvkov. Preto sa Bi-RNN často využívajú pri úlohách, ako je strojový preklad, rozpoznávanie reči alebo analýza sentimentu.

Nevýhodou tejto metódy je vyššia výpočtová náročnosť, pretože model musí spracovať každú sekvenciu dvakrát – raz vpred a raz vzad.

Problémy RNN a ich varianty

Obyčajné rekurentné neurónové siete čelia dvom hlavným problémom:

1. Miznúce gradienty (*Vanishing gradients*) – počas trénovania môžu gradienty klesnúť na veľmi malé hodnoty, čo bráni efektívnej aktualizácii váh vo vzdialených časových krokoch a sťažuje učenie dlhodobých závislostí – problém s dlhodobou pamäťou.

2. Explodujúce gradienty (*Exploding gradients*) – gradienty môžu naopak rýchlo narásť na extrémne hodnoty, čo destabilizuje trénovanie a vedie k zlyhaniu modelu.

Tieto problémy sťažujú učenie najmä pri dlhých sekvenciách, kde je potrebné uchovávať informácie z väčšej časovej vzdialenosti. Na riešenie týchto nedostatkov boli vytvorené pokročilejšie varianty RNN – LSTM 3.4 a GRU 3.4, ktoré lepšie zvládajú dlhodobé závislosti a zaisťujú stabilnejšie trénovanie.

LSTM

Long Short-Term Memory (LSTM) je varianta RNN, ktorá bola navrhnutá na riešenie problému miznúcich gradientov.

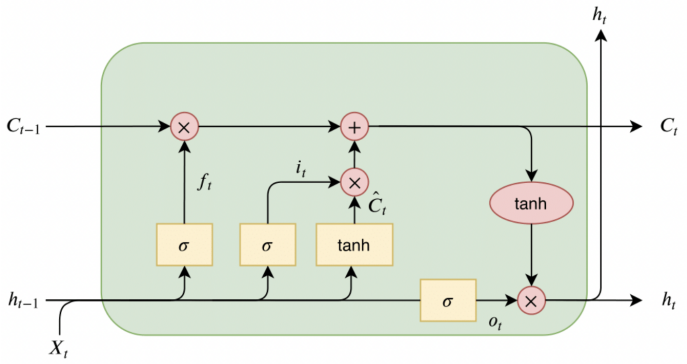
LSTM tento problém rieši pomocou špeciálnej architektúry, ktorá obsahuje brány na riadenie toku informácií. Tieto brány modelu pomáhajú rozhodnúť, ktoré informácie si má zapamätať, ktoré ignorovať a ktoré posunúť ďalej. Existujú tri typy brán:

1. Forget gate (f_t) – rozhoduje, ktoré informácie z predchádzajúcich krokov sú už nepotrebné a môžu sa zahodiť.

2. Input gate (i_t) – určuje, ktoré nové informácie sú dôležité a majú sa pridať do pamäte. Najskôr vypočíta kandidátne hodnoty (možné nové informácie) a z nich vyberie tie najrelevantnejšie.

3. Output gate (o_t) – rozhoduje, ktoré informácie sa použijú ako výstup a zároveň sa posunú do nasledujúceho časového kroku.

LSTM pracuje s dvoma pamäťovými stavmi: dlhodobou pamäťou (c_t) a krátkodobou pamäťou (h_t). Vďaka tejto architektúre (vid. 3.12) dokáže efektívne spracovávať dlhé sekvencie a zachovávať dôležité informácie, čo ho robí vhodným pre úlohy ako spracovanie prirodzeného jazyka, rozpoznávanie reči alebo predikcia časových radov.



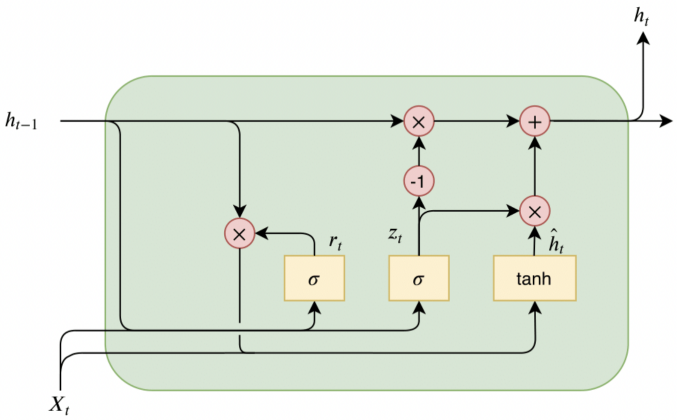
Obr. 3.12: LSTM bunka [35].

GRU

Gated Recurrent Unit (GRU) je zjednodušená verzia LSTM, ktorá tiež rieši problém miznúcich gradientov. Na rozdiel od LSTM používa GRU len jeden pamäťový stav (h_t), ktorý kombinuje funkcie krátkodobej a dlhodobej pamäte. Táto jednoduchšia architektúra (vid. 3.13) znižuje výpočtovú náročnosť a zrýchľuje trénovanie, pričom výsledky sú často porovnateľné s LSTM.

GRU obsahuje dve hlavné brány:

1. Reset gate (r_t) – určuje, akú časť informácií z predchádzajúceho kroku je ešte dôležité zachovať.
2. Update gate (z_t) – rozhoduje, ktoré informácie z predošlého kroku možno zahodiť a ktoré nové informácie treba pridať.



Obr. 3.13: GRU bunka [35].

Vďaka nižším výpočtovým nárokom je GRU vhodné pre aplikácie, kde je dôležitá rýchlosť a efektivita, napríklad pri preklade textu, rozpoznávaní reči alebo predikcii časových radov.

Mechanizmus pozornosti v rekurentných sieťach

Aj keď LSTM a GRU dokážu spracovať dlhé sekvencie a zapamätať si dôležité informácie z predchádzajúcich krokov, niekedy to jednoducho nestačí. Pri čítaní z pier môže byť dôležité sústrediť sa na konkrétne snímky, napríklad tie, kde je pohyb úst najvýraznejší, a iné naopak ignorovať. Na to slúži mechanizmus pozornosti.

Pozornosť (angl. *attention*) [7] umožňuje modelu dynamicky rozhodovať, ktoré časti sekvencie sú v danom momente najdôležitejšie. Namiesto toho, aby sa model spoliehal na jeden spoločný súhrn všetkých informácií, vie sa vrátiť späť k jednotlivým krokom v sekvencii a rozhodnúť, ktoré z nich sú práve teraz najdôležitejšie. Každému kroku potom priradí väčšiu alebo menšiu váhu podľa jeho významu.

Vďaka tomuto prístupu dokáže model oveľa presnejšie pracovať aj s dlhými a zložitejšími vstupmi. V prípade čítania z pier môže model vďaka pozornosti napríklad správne určiť, že niektoré pohyby pier sú kľúčové pre rozlíšenie medzi dvoma podobnými slovami, a sústrediť sa práve na ne. Mechanizmus pozornosti sa často využíva v spojení s rekurentnými sieťami

ako doplnok nad výstupmi z Bi-GRU alebo Bi-LSTM. V modernejších architektúrach sa pozornosť používa aj samostatne – napríklad v transformeroch.

3.5 Manipulácia a práca s dátami

Táto sekcia opisuje formát vstupných dát, ich rozdelenie na trénovaciu, validačnú a testovaciu množinu, predspracovanie a augmentáciu, ktoré sú nevyhnutné pre úspešné testovanie modelu na čítanie slov z pier. Správna príprava dát výrazne ovplyvňuje kvalitu tréningu a presnosť výsledného modelu.

Formát dát

V rámci úlohy čítania z pier sa pracuje s videami, ktoré zaznamenávajú osobu vyslovujúcu jednotlivé slová alebo vety bez zvuku. Každé video pozostáva zo sekvencie snímok (*frames*), pričom každá snímka obsahuje tvár osoby so zameraním na oblasť úst.

Na spracovanie modelom je potrebné konvertovať videá na jednotlivé snímky. Každé video je reprezentované sekvenciou obrázkov s pevnou dĺžkou (konštantný počet snímok na jedno video). Snímky sa následne normalizujú na jednotné rozmery, napríklad 64×64 pixelov. Obrázky sa často konvertujú do čiernobieleho formátu (*grayscale*) s cieľom minimalizovať výpočtové nároky a redukovať množstvo nepotrebných farebných informácií. Použitie grayscale však nie je povinné – v niektorých prípadoch, v závislosti od architektúry modelu, sa môžu zachovávať aj pôvodné farebné (RGB) snímky, ak farebné informácie zlepšujú výkonnosť modelu.

Trénovacia, validačná a testovacia množina

Celý dataset potrebný na vytvorenie modelu na čítanie z pier je rozdelený do niekoľkých častí [31]:

- Trénovacia množina – dáta používané na natrénovanie modelu, pričom model sa na nich učí vzory a závislosti.
- Validačná množina – dáta používané na validáciu modelu počas trénovania, pomáhajú predchádzať pretrénovaniu (*overfittingu*) a nastavovať hyperparametre.
- Testovacia množina – dáta používané na finálne nezávislé overenie výkonnosti modelu po ukončení tréningu.

Rozdelenie dát sa môže vykonať náhodne, pričom sa zabezpečí rovnomerné zastúpenie všetkých tried vo všetkých množinách. V niektorých prípadoch však môže byť dataset už vopred rozdelený autorom, pričom sú trénovacie, validačné a testovacie množiny jasne určené.

Typicky býva trénovacia množina najväčšia (napr. 70 % všetkých dát), zatiaľ čo validačná a testovacia množina sú menšie a zvyčajne rovnako veľké (napr. po 15 % dát).

Predspracovanie dát

Pred začatím tréningu a tvorby modelu je potrebné vykonať niekoľko krokov predspracovania dát [12]:

1. Normalizácia – všetky obrázky sú normalizované do rozsahu od 0 do 1. Tento krok zabezpečí jednotný rozsah hodnôt a rýchlejšiu konvergenciu počas trénovania modelu. Práca s menšími číslami zrýchľuje výpočty a zlepšuje stabilitu tréningu, keďže hodnoty pixelov sa štandardne pohybujú v rozmedzí od 0 do 255.

2. Detekcia tváre a extrakcia oblasti úst – z každej snímky sa detekuje oblasť úst, ktorá je následne zmenšená na požadované rozmery.

3. Konverzia z RGB na odtiene sivej (*greyscale*) – farebné obrázky sa môžu konvertovať na čiernobielu, čím sa znižuje počet kanálov. Tento krok však nie je vždy povinný, ak farebné informácie (RGB) prispievajú k lepšiemu rozpoznávaniu.

4. Zabezpečenie jednotnej dĺžky sekvencie – pri videách s rôznou dĺžkou sa buď doplnia prázdne snímky (*padding*), alebo sa sekvencia oreže na požadovanú dĺžku.

Techniky augmentácie dát

Na zvýšenie robustnosti modelu a prevenciu pretrénovania sa využívajú nasledujúce techniky augmentácie. Dôležité je zabezpečiť, aby augmentácia neovplyvnila viditeľnosť a kvalitu obrazu pier [23], [10], [29], [12].

- Náhodné otočenie (*Random rotation*) – snímky sa náhodne otáčajú v rozsahu uhla, zvyčajne $-15°$ od $+15°$ alebo od $-5°$ do $+5°$.

- Náhodné horizontálne prevrátenie (*Random horizontal flipping*) – snímky sa náhodne prevracajú horizontálne s určitou pravdepodobnosťou (zvyčajne 50 %)

- Náhodné priblíženie (*Random zoom*) – zväčšenie snímky o 10–20 % pôvodnej veľkosti. Pery musia zostať viditeľné.

- Náhodné posunutie (*Random shift*) – posun snímky o určitý počet pixelov horizontálne alebo vertikálne, pričom pery musia byť stále na zábere.

- Náhodná zmena jasu (*Random brightness adjustment*) – úprava jasu v rozsahu od -10 % do +10 %.

- Pridanie šumu (*Gaussian noise*) – pridanie šumu s nulovou strednou hodnotou a nízkou varianciou na zvýšenie odolnosti modelu voči šumu v reálnych dátach.

- Časové natiahnutie (*Time wrapping*) – mierne natiahnutie alebo stlačenie časovej osi videozáznamu, ktoré simuluje zmeny v rýchlosti reči a zvyšuje odolnosť modelu voči variabilite tempa hovorenia.

3.6 Zhrnutie

V predchádzajúcej kapitole sa vysvetlilo, ako moderné technológie na čítanie z pier využívajú neurónové siete a ich rôzne typy. Hlavná pozornosť v tejto kapitole bola venovaná základným princípom fungovania týchto sietí a ich najdôležitejším konceptom.

Prebrali sa dva hlavné typy sietí – konvolučné a rekurentné. Konvolučné siete pomáhajú extrahovať kľúčové vizuálne črty zo snímok videí, kde človek rozpráva. Stručne sa vysvetlila ich architektúra a základné časti, ako sú konvolučné vrstvy, padding a stride operácie,

pooling vrstvy a plne prepojené vrstvy. V krátkosti sa spomenuli aj 3D-CNN a známe modely založené na tejto technológii.

Rekurentné siete sú nevyhnutné na spracovanie sekvencií, keďže umožňujú modelu uchovávať a analyzovať význam celej sekvencie na základe informácií z predchádzajúcich krokov. Rozobrali sa základné vlastnosti rekurentných neurónov a ich praktické využitie. Dôležitou súčasťou bolo aj kódovanie slov, pretože počítač ich musí vnímať ako čísla. Vysvetlila sa technika tokenizácie, ktorá umožňuje efektívne spracovanie slov vo forme číselných sekvencií bez náročného tréningu vektorových reprezentácií.

Predstavil sa aj mechanizmus pozornosti, ktorý umožňuje modelu sústrediť sa na kľúčové časti sekvencie pri predikcii výsledku. Vďaka tomu je možné zvýšiť presnosť rozpoznávania slov z pier tým, že model dynamicky vyhodnocuje, ktoré snímky alebo pohyby pier sú pre správne rozpoznanie najdôležitejšie.

Na záver sa spomenuli aj obojsmerné RNN a ich varianty, ako sú LSTM a GRU, ktoré budú kľúčové pri samotnej implementácii čítania z pier.

Kapitola sa končí opisom dát, ktoré budú použité – konkrétne pôjde o snímky extrahované z videí. Spomenuli sa aj rôzne dátové množiny (trénovacia, validačná a testovacia) a zdôraznila sa potreba správneho predspracovania dát na efektívny tréning modelu. Na záver sa rozoberali aj techniky augmentácie dát, ktoré sú nevyhnutné pre dosiahnutie čo najpresnejších výsledkov.

Kapitola 4

Návrh a implementácia

Po teoretickom úvode do strojového učenia a predstavení kľúčových typov neurónových sietí, ako sú konvolučné (CNN) a rekurentné siete (RNN), nasleduje praktická časť tejto práce – návrh a implementácia modelu LipDesciphNet. Cieľom tohto modelu je čo najpresnejšie určiť, čo anglicky hovoriaca osoba vyslovuje iba na základe pohybov pier.

Táto kapitola pokrýva celý proces vytvárania systému. Najprv je predstavený použitý dataset a spôsoby spracovania dát – od detekcie oblasti úst cez normalizáciu textových transkripcií až po tvorbu tokenizačného slovníka. Nasleduje fáza predspracovania obrazových vstupov (oblastí úst), vrátane techník augmentácie, ktoré zvyšujú robustnosť modelu.

Ťažiskom kapitoly je samotná architektúra modelu LipDesciphNet, ktorá spája viacero moderných prvkov: 3D konvolučné siete na extrakciu priestorovo-časových čŕt, sieť typu Highway na prenos dôležitých informácií, obojsmernú GRU na analýzu pohybov pier a kaskádový dekóder využívajúci mechanizmus pozornosti v kombinácii so stratovou funkciou CTC. Vysvetlené sú aj spôsoby dekódovania výstupu pomocou greedy a beam search algoritmov. Model bol implementovaný pomocou frameworku PyTorch [27], ktorý poskytuje flexibilné a výkonné prostredie pre návrh hlbokých neurónových sietí.

Na záver je predstavená výsledná webová aplikácia, pozostávajúca z frontendovej časti, ktorá pomocou webkamery zaznamenáva oblasti úst, a backendovej časti, ktorá využíva trénovaný model LipDesciphNet na prevod pohybov pier na text v reálnom čase.

4.1 Popis datasetu

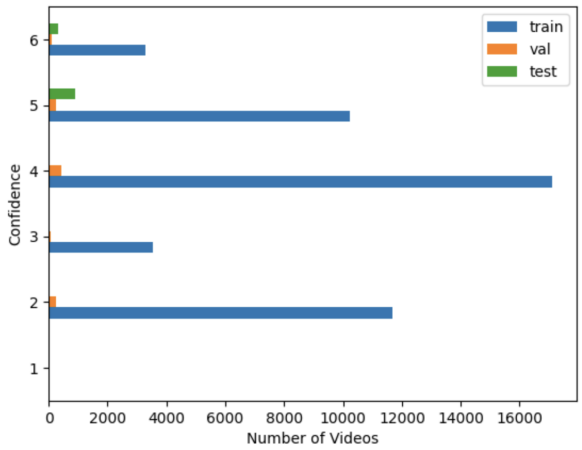
Na realizáciu tejto práce bol použitý dataset LRS2 (Lip Reading Sentences 2) [2], ktorý patrí medzi najväčšie dostupné datasety pre úlohu čítania z pier. Dataset obsahuje videá anglicky hovorených viet pochádzajúcich z talk show programov BBC spolu s textovými transkripciami, ktoré udávajú, čo bolo v každom videu povedané. Autori datasetu LRS2 rozdelili dáta do štyroch množín:

- predtrénovacia množina – 96 318 videí, 41 427 unikátnych slov; pri každom videu sú k dispozícii aj *alignmenty*, ktoré obsahujú informáciu o časovom zarovnaní slov vo videu,
- trénovacia množina – 45 839 videí, 17 660 unikátnych slov, bez dostupných alignmentov,
- validačná množina – 1 082 videí, 1 984 unikátnych slov, bez dostupných alignmentov,

- testovacia množina – 1 243 videí, 1 698 unikátnych slov, bez dostupných alignmentov,

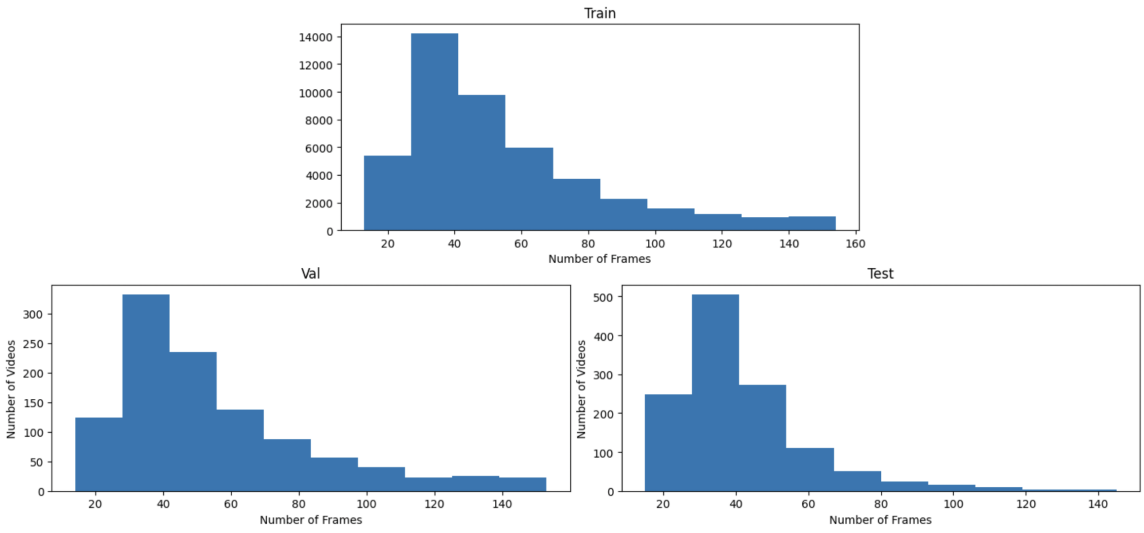
Predtrénovacia množina obsahuje menej kvalitné videá – napríklad so zlým uhlom záberu alebo nepresnou transkripciou. Z tohto dôvodu bude v tejto práci vynechaná a ďalej sa nebude nevyužívať.

Každá transkripcia je sprevádzaná hodnotou *confidence*, ktorá udáva mieru istoty správnosti prepisu. Hodnoty sa pohybujú v rozsahu od 1 do 6, pričom hodnota 1 predstavuje najnižšiu a hodnota 6 najvyššiu mieru istoty správnosti transkripcie. Graf nižšie 4.1 znázorňuje, koľko videí z jednotlivých množín dosahuje jednotlivé hodnoty confidence.



Obr. 4.1: Distribúcia hodnoty confidence v jednotlivých množinách.

Všetky množiny obsahujú videá rôznej dĺžky – niektoré trvajú 3 sekundy, iné 4 sekundy a podobne. Tento fakt predstavuje problém pri implementácii, keďže je potrebné, aby všetky videá v dávke mali rovnakú dĺžku, teda rovnaký počet snímok (viac v sekcii predspracovania dát 4.3). Dĺžka videa závisí od počtu snímok a snímkovej frekvencie. Snímková frekvencia všetkých videí v datasete je 25 snímok za sekundu. Graf nižšie 4.2 znázorňuje rozloženie počtu snímok vo videách jednotlivých množín.



Obr. 4.2: Histogram dĺžky videí v jednotlivých množinách.

4.2 Príprava dát

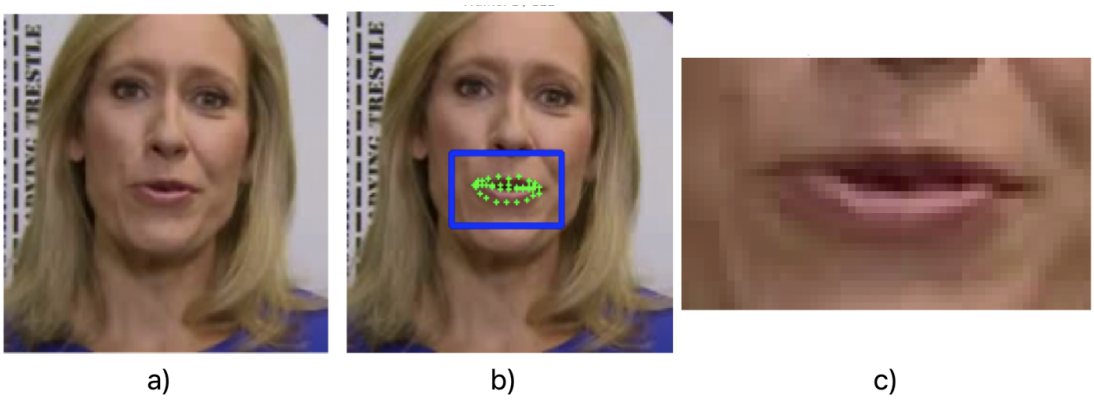
Pred trénovaním modelu je potrebné pripraviť dáta. Tento proces zahŕňa načítanie videí spolu s ich transkripciami. Z jednotlivých videí sa extrahuje oblasť úst zo snímok, zatiaľ čo transkripcie sa normalizujú do finálnej podoby. Následne sa vytvorí charakterovo založený slovník, ktorý reprezentuje jednotlivé znaky anglickej abecedy a ďalšie špeciálne znaky ako apostrof a podobne. Každý z týchto krokov je kľúčový na zabezpečenie konzistentného a vhodného vstupu pre následné trénovanie modelu.

Detekcia tváre a extrakcia oblasti úst

V prvom kroku je potrebné z každého videa detekovať tvár a z nej izolovať iba oblasť úst, keďže ostatné časti tváre sú pre čítanie z pier irelevantné. Zo sekvencie snímok videa, v ktorom osoba rozpráva, sa preto z každého snímku extrahuje len oblasť úst, ktorá sa ďalej používa v ďalšom spracovaní.

Hoci by sa dali použiť iba súradnice kľúčových bodov pier (zelené body na obrázku (b) 4.3), táto práca uprednostňuje extrakciu celej oblasti úst. Dôvod je jednoduchý – obrázky zachytávajú omnoho viac detailov než len pozície bodov. Okrem tvaru pier sú viditeľné aj pohyby pokožky, zuby, bradu, jazyk či tiene – všetko to, čo môže pomôcť pri rozlíšení foném alebo slov. Tieto vizuálne detaily by pri samotných bodoch chýbali, čo by mohlo znížiť presnosť modelu. Práve preto sa pracuje s obrázkami, ktoré umožňujú modelu učiť sa bohatšie a presnejšie reprezentácie.

V tejto fáze ešte tieto oblasti úst neprešli žiadnym predspracovaním – sú v pôvodnej farebnej schéme RGB, nemajú jednotnú veľkosť a videá nemajú rovnaký počet snímok. Ide teda o surové dáta, kde jediná vykonaná úprava bola izolácia oblasti úst zo snímok.



Obr. 4.3: Ukážka extrakcie oblasti úst z pôvodného snímku. Na obrázku (a) je zobrazený pôvodný snímok z náhodného videa z datasetu LRS2. Obrázok (b) znázorňuje detekciu pier pomocou bodov (zelené body) a detekciu oblasti úst (modrý ohraničujúci rámček). Obrázok (c) ukazuje výslednú extrahovanú oblasť úst. Táto oblasť je zatiaľ nespracovaná, v pôvodných farbách RGB, bez zmeny veľkosti alebo ďalšieho predspracovania.

Na detekciu tváre a extrakciu oblasti úst je použitá knižnica MediaPipe [22] s využitím modelu FaceMesh, ktorý z obrázku získava kľúčové body tváre vrátane pier (zelené body na obrázku (b) 4.3). Na základe týchto bodov sa určí ohraničujúci obdĺžnik pomocou extrémov

súradníc na osiach x a y bodov pier, pričom rámček bol rozšírený o malý okraj, aby zachytil aj blízke okolie pier (modrý rámček na obrázku (b) a výsledok na obrázku (c) 4.3).

MediaPipe FaceMesh je zvolený z dôvodu vysokej rýchlosti, presnosti detekcie a efektívneho fungovania aj na procesoroch. Vďaka hustote 468 bodov umožňuje presnú lokalizáciu pier, čo je kľúčové pre úlohy čítania z pier. V porovnaní s inými metódami ponúka optimálnu rovnováhu medzi presnosťou a výpočtovou náročnosťou.

Normalizácia transkripcií

Každé video v datasete LRS2 obsahuje transkripciu hovoreného obsahu, teda textový zápis toho, čo bolo vo videu povedané. Tieto transkripcie však nie sú v pôvodnej forme priamo použiteľné na ďalšie spracovanie, a preto prechádzajú procesom normalizácie. Jej cieľom je zabezpečiť jednotný a konzistentný textový formát, vhodný na spracovanie modelmi strojového učenia.

V rámci normalizácie sa aplikujú viaceré štandardné kroky. Všetky znaky sa prevádzajú na malé písmená (tzv. *lowercasing*), čím sa zabezpečí jednotnosť zápisu bez ohľadu na pôvodnú veľkosť písmen. Z textu sa zároveň odstraňujú nadbytočné medzery, prázdne znaky a iné nerelevantné symboly, ktoré by mohli znižovať kvalitu vstupných dát.

Dôležitou súčasťou procesu je aj konverzia číselných údajov do slovnej podoby a podobná transformácia dátumov, skratiek či špeciálnych znakov do ich štandardizovaných textových foriem. Takto upravený text je výrazne vhodnejší na ďalšie spracovanie v jazykových modeloch, pretože redukuje šum a variabilitu vstupných údajov.

Príklad pôvodnej a normalizovanej verzie:

- Pôvodný text: „Temperature today is −5 degrees“
- Normalizovaný text: „temperature today is minus five degrees“

Na konverziu číselných údajov do slovnej podoby bol použitý nástroj Text Normalizer z frameworku NeMo od spoločnosti NVIDIA [26]. Tento nástroj je špeciálne navrhnutý na tento účel a poskytuje vysokú presnosť pri normalizácii textu.

Tvorba slovníku

Na reprezentáciu transkripcií bol vytvorený tokenizačný slovník 3.4, ktorý priraďuje každej jednotke výstupu jedinečný celočíselný token. V prípade tejto práce bol zvolený charakterovo založený slovník, teda každý znak v texte má svoje jedinečné číselné zastúpenie.

Charakterový slovník sa ukázal ako vhodná voľba pri spracovaní sekvencií hovoreného jazyka, najmä pre modely využívajúce stratovú funkciu CTC. Tento prístup umožňuje presné modelovanie aj zriedkavých alebo predtým nevidených slov a zároveň znižuje veľkosť výstupného priestoru, keďže pracuje iba s obmedzeným počtom znakov namiesto celých slov.

Keďže sa práca zameriava na čítanie pier v anglickom jazyku, slovník obsahuje nasledujúce znaky:

- všetky malé písmená anglickej abecedy od „a“ po „z“,
- medzeru ako znak oddeľujúci jednotlivé slová, ktorý je označený znakom „|“,
- apostrof (’) na zachovanie významu skratiek ako napríklad u slov *it’s*, *don’t*, *you’re*,
- špeciálny token `<BLANK>`, používaný stratovou funkciou CTC 4.4,

Pred každým trénovaním modelu sa normalizovaná transkripcia videa konvertuje na postupnosť tokenov, kde každý token reprezentuje jeden konkrétny znak transkripcie. Pre lepšiu predstavu, ako tento proces funguje – od normalizácie textu po jeho tokenizáciu, je uvedený príklad:

- Pôvodný text: „I have 2 cats“
- Normalizovaný text: „i have two cats“
- Tokenizáčny slovník:
  - i: 1,
  - |: 2,
  - h: 3,
  - a: 4,
  - v: 5,
  - e: 6,
  - t: 7,
  - w: 8,
  - o: 9,
  - c: 10,
  - s: 11
- Tokenizovaný text: $[1^{(\text{i})}, 2^{(\text{whitespace})}, 3^{(\text{h})}, 4^{(\text{a})}, 5^{(\text{v})}, 6^{(\text{e})}, 2^{(\text{whitespace})}, 7^{(\text{t})}, 8^{(\text{w})}, 9^{(\text{o})}, 2^{(\text{whitespace})}, 10^{(\text{c})}, 4^{(\text{a})}, 7^{(\text{t})}, 11^{(\text{s})}]$

4.3 Predspracovanie dát

Po získaní surových dát – teda sekvencií snímok, kde každý snímok obsahuje iba oblasť úst – a po normalizácii textových transkripcií, nasleduje fáza predspracovania. Jej cieľom je pripraviť vizuálne dáta tak, aby mohli byť efektívne spracované neurónovou sieťou. To zahŕňa úpravu veľkosti oblastí úst, aby mali všetky jednotný rozmer, a zároveň aj zarovnanie dĺžky sekvencií, aby boli všetky videá v dávke rovnako dlhé. V rámci tejto fázy sa tiež aplikujú augmentačné techniky na zvýšenie rozmanitosti trénovacích dát.

Rovnaká veľkosť oblasti úst

Jedným z problémov je rôzna veľkosť extrahovaných oblastí úst. Preto sú všetky snímky zjednotené na rovnakú veľkosť 86×138 pixelov. Táto veľkosť je zvolená ako kompromis – nie je ani príliš malá na to, aby sa stratili dôležité detaily, ani príliš veľká, čo by zvyšovalo výpočtovú náročnosť.

Rovnaká dĺžka videí

Ďalšou výzvou je rôzna dĺžka videí – niektoré trvajú 3 sekundy, iné napríklad 5 sekúnd. Keďže modely vyžadujú, aby všetky vstupy v jednej dávke mali rovnakú dĺžku, je potrebné tento rozdiel eliminovať.

Na elimináciu tohto problému je použitá technika paddingu, pri ktorej sa všetky kratšie videá dopĺňajú (tzv. napaddujú) prázdnymi snímkami na dĺžku najdlhšieho videa v dávke. Prítomnosť paddingu (prázdnych snímok) však neovplyvňuje výsledok učenia, pretože použitá stratová funkcia CTC 4.4 správne pracuje s informáciou o skutočnej dĺžke sekvencií a padding automaticky ignoruje.

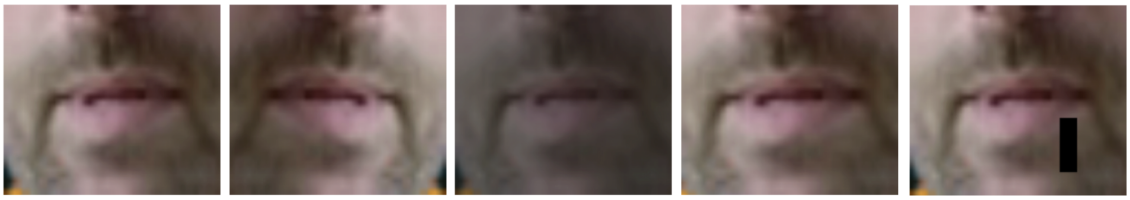
Augmentácia dát

Aby sa zvýšila robustnosť modelu voči rôznym variáciám vo vizuálnej oblasti úst a zároveň sa zlepšila jeho generalizácia, je použitá technika augmentácie dát (taktiež spomínaná v sekcii 3.5).

Hoci v niektorých experimentoch nebola augmentácia použitá, pri iných sa ukázala ako prínosná. V prípade videodát, kde každé video predstavuje sekvenciu snímok oblasti úst, je dôležité zachovať konzistenciu v rámci jednotlivých videí. Preto sú augmentačné transformácie aplikované nie na jednotlivé snímky nezávisle, ale konzistentne na celú sekvenciu – teda všetky snímky v jednom videu sú transformované rovnakým spôsobom. Tým sa zachováva temporálna súdržnosť pohybu pier.

Na trénovacie videá sú náhodne aplikované nasledujúce augmentačné techniky, ktoré sú taktiež vizuálne zobrazené na obrázku nižšie 4.4:

- Náhodné horizontálne prevrátenie – simuluje variácie pri rôznych uhloch záberu kamery alebo symetriu pohybov pier.
- Náhodné zmeny jasu a kontrastu – simulácia rôznych svetelných podmienok v prostredí, kde bolo video nahraté.
- Náhodné priblíženie – model sa učí extrahovať dôležité informácie aj z rôzne orezaných alebo mierne zväčšených oblastí úst.
- Cutout (náhodné zakrytie časti obrazu) – zakrytie malej náhodnej oblasti úst (napr. tieň, zub, vráska), čo vedie k vyššej robustnosti pri reálnom použití.



Obr. 4.4: Obrázky znázorňujúce dátové augmentácie spomenuté vyššie. Na obrázku a) je pôvodný snímok oblasti úst. Obrázok b) ukazuje horizontálne prevrátenie snímku, c) zmenu jasu a kontrastu, d) mierne priblíženie a e) náhodné zakrytie časti obrazu prázdnym (čiernym) obdĺžnikom.

Tieto augmentácie sú aplikované s určitými pravdepodobnosťami, tzn. že každé video môže byť modifikované jednou alebo viacerými z uvedených techník, alebo nemusí byť modifikované vôbec. Týmto spôsobom sa zvyšuje variabilita trénovacích dát bez toho, aby došlo k strate významových prvkov potrebných na čítanie z pier, pričom sa zároveň zachováva prirodzená dynamika pohybu úst v čase.

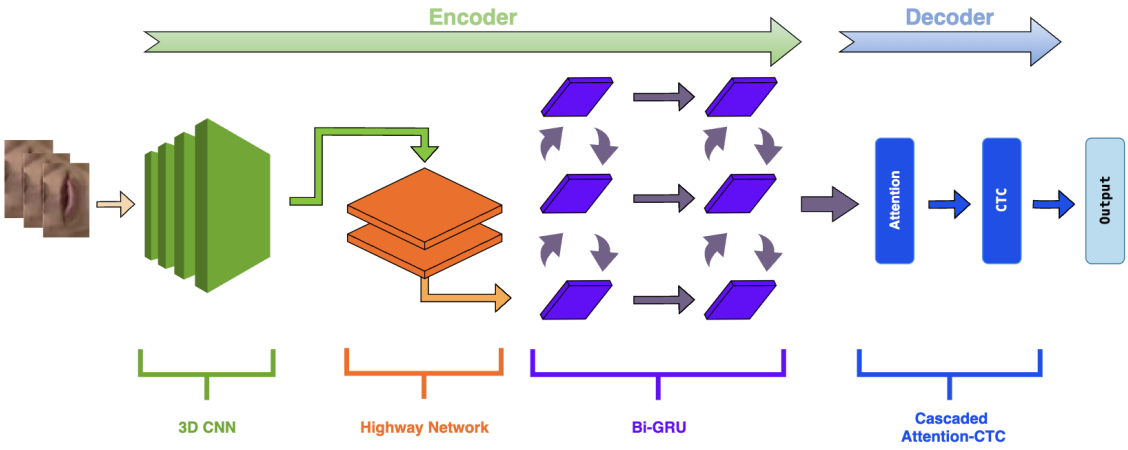
4.4 Architektúra modelu LipDesciphNet

Model LipDesciphNet je navrhnutý na úlohu automatického čítania z pier – teda na prevod pohybov úst vo videu na zodpovedajúci textový výstup. Inšpiruje sa existujúcim systémom

LCANet 2.2, ktorý kombinuje výhody stratovej funkcie CTC a mechanizmu pozornosti v rámci tzv. kaskádového dekódovania.

LipDesciphNet preberá hlavnú myšlienku LCANet, v rámci ktorej sa pozornosť využíva na extrakciu kontextovo dôležitých častí sekvencie. Zároveň je model trénovaný pomocou CTC straty, ktorá nevyžaduje presné časové zarovnanie medzi vizuálnym vstupom a textovým výstupom. Táto kombinácia zvyšuje stabilitu trénovania aj presnosť rozpoznávania.

Na rozdiel od pôvodného LCANet, ktorý taktiež využíva 3D-CNN, model LipDesciphNet používa vlastnú 3D-CNN architektúru. Tá je síce výpočtovo náročnejšia, no zároveň lepšie optimalizovaná na extrakciu priestoro-časových čŕt z oblasti úst.



Obr. 4.5: Architektúra modelu LipDesciphNet.

Architektúra modelu (vid. vyššie 4.5) pozostáva zo štyroch hlavných komponentov:

- 3D-CNN extraktor priestorovo-časových čŕt 4.4 – táto časť modelu spracúva video ako trojrozmerné dáta (šírka, výška, čas) a zo sekvencie pohybov pier extrahuje dôležité vizuálne informácie.
- Highway sieť 4.4 – zabezpečuje, aby sa extrahované informácie nestratili pri prechode do ďalších vrstiev modelu. Pomáha zachovať podstatné črty, ktoré by inak mohli zaniknúť.
- Bi-GRU enkóder 4.4 – dvojvrstvová obojsmerná GRU sieť, ktorá sleduje pohyby pier v čase.
- Kaskádový attention-CTC dekóder 4.4 – posledná časť modelu, ktorá pomocou mechanizmu pozornosti vyberá najdôležitejšie časti sekvencie a postupne generuje textový výstup. Celý model sa trénuje pomocou CTC straty, ktorá umožňuje naučiť sa správne zarovnávať video s textom.

3D-CNN extraktor priestorovo-časových čŕt

Prvou zložkou architektúry je 3D-CNN sieť, ktorá slúži ako extraktor priestorovo-časových čŕt z oblasti úst. Na rozdiel od tradičných CNN, ktoré pracujú len s priestorom, 3D-CNN dokáže spracovávať aj časovú dimenziu, vďaka čomu efektívne zachytáva pohyby pier naprieč snímkami.

Aby model dokázal naplno využiť informácie o pohybe pier, architektúru je potrebné navrhnúť bez akéhokoľvek downsamplingu v časovej dimenzii – teda počet časových krokov T zostáva nezmenený počas celej 3D-CNN časti modelu. Zachovanie tejto časovej rezolúcie je zásadné, pretože každá snímka môže niesť dôležité informácie o artikulačných pohyboch, ktoré by sa pri zmenšení mohli stratiť. Navrhnutá architektúra 3D-CNN extraktora je zobrazená v tabuľke nižšie 4.1.

Tabuľka 4.1: Architektúra 3D-CNN extraktoru modelu LipDesciphNet. Všetky snímky majú jednotnú veľkosť 86×138, kde B označuje veľkosť dávky a T dĺžku sekvencie videí v dávke.

Názov vrstvy	Kernel	Stride	Padding	Veľkosť výstupu
Vstup	-	-	-	$B \times 3 \times T \times 86 \times 138$
3d-conv1	(3, 5, 5)	(1, 2, 2)	(1, 2, 2)	$B \times 32 \times T \times 43 \times 69$
bn/relu				$B \times 32 \times T \times 43 \times 69$
3d-conv2	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	$B \times 32 \times T \times 43 \times 69$
bn/relu/drop				$B \times 32 \times T \times 43 \times 69$
max-pool1	(1, 2, 2)	(1, 2, 2)		$B \times 32 \times T \times 21 \times 34$
3d-conv3	(3, 5, 5)	(1, 1, 1)	(1, 2, 2)	$B \times 64 \times T \times 21 \times 34$
bn/relu				$B \times 64 \times T \times 21 \times 34$
3d-conv4	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	$B \times 64 \times T \times 21 \times 34$
bn/relu/drop				$B \times 64 \times T \times 21 \times 34$
max-pool2	(1, 2, 2)	(1, 2, 2)		$B \times 64 \times T \times 10 \times 17$
3d-conv5	(3, 5, 5)	(1, 1, 1)	(1, 2, 2)	$B \times 128 \times T \times 10 \times 17$
bn/relu				$B \times 128 \times T \times 10 \times 17$
3d-conv6	(3, 3, 3)	(1, 1, 1)	(1, 1, 1)	$B \times 128 \times T \times 10 \times 17$
bn/relu/drop				$B \times 128 \times T \times 10 \times 17$
max-pool3	(1, 2, 2)	(1, 2, 2)		$B \times 128 \times T \times 5 \times 8$

Highway sieť a Bi-GRU enkóder

Po extrakcii priestorovo-časových čŕt pomocou 3D-CNN prechádzajú tieto dáta cez tzv. sieť typu Highway, ktorá modelu pomáha uchovávať dôležité informácie. Slúži ako „inteligentný filter“, ktorý sám rozhoduje, ktorým častiam vstupu umožní prejsť ďalej bez zmeny a ktoré upraví. Vďaka tomu sa nestratia ani jemné, ale významné znaky pohybov pier.

Na túto vrstvu nadväzuje dvojvrstvová obojsmerná GRU sieť (Bi-GRU), ktorá dokáže spracovať sekvenciu pohybov úst v oboch smeroch – z minulosti aj z budúcnosti. Získava tak bohatší časový kontext, čo je pri rozpoznávaní slov veľmi dôležité, keďže niektoré pohyby pier dávajú zmysel až v spojení s nasledujúcimi. Výsledkom je sekvencia vektorov, ktorá veľmi presne reprezentuje celkový priebeh artikulácie.

Kaskádový attention-CTC dekóder

Poslednou súčasťou architektúry LipDesciphNet je tzv. kaskádový attention-CTC dekóder, ktorý kombinuje mechanizmus pozornosti (mechanizmus pozornosti je bližšie popísaný v sekcii 3.4) a stratovú funkciu CTC (z angl. *Connectionist Temporal Classification*). Návrh tohto dekodéra priamo vychádza z práce LCANet [38]. Jeho úlohou je premeniť sekvenciu vektorov (získanú z Bi-GRU) na výsledný text – teda rozpoznať, aké slová boli vyslovené na základe pohybov pier.

LCANet ukázal, že samotný attention mechanizmus je veľmi efektívny pri učení zarovnania medzi vstupmi a výstupmi, najmä keď sa význam výstupu odvíja od rôznych častí vstupnej sekvencie. V prípade čítania z pier to znamená, že model sa dokáže „sústrediť“ na konkrétne momenty vo videu, ktoré sú pre rozpoznanie dôležité.

Naopak, CTC je obzvlášť vhodné pri trénovaní modelov, kde nie je známe presné časové zarovnanie medzi vstupom a výstupom – čo presne zodpovedá problému čítania z pier [13]. Umožňuje modelu učiť sa iba z finálneho textu bez potreby rámček-po-rámček zarovnania s videom.

V tomto modeli dekóder pre každý výstupný krok vytvára váženú reprezentáciu vstupu, pričom namiesto bežne používanej *cross-entropy* straty sa trénuje výhradne pomocou CTC straty. Tým sa spájajú výhody oboch prístupov: attention pomáha modelu extrahovať relevantné informácie, zatiaľ čo CTC zabezpečuje presné a stabilné dekódovanie bez potreby ručného zarovnávania.

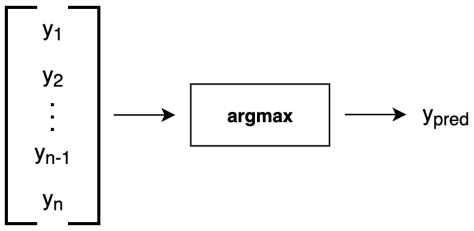
Viac detailov o kaskádovom attention-CTC dekóderi možno nájsť v práci LCANet, ktorá tento efektívny prístup navrhla a podrobne opísala.

Dekódovanie výstupu

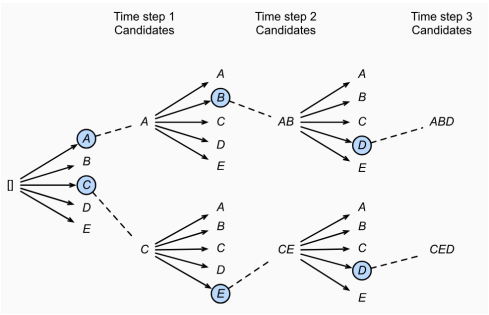
Výstupom modelu LipDesciphNet je sekvencia pravdepodobností pre jednotlivé znaky v každom časovom kroku. Tieto pravdepodobnosti je potrebné premeniť na výsledný text – teda dekódovať predikciu modelu. V tomto procese sa využívajú dve hlavné metódy [20]:

1. Greedy search (vid. prvý obrázok 4.6) – najjednoduchší spôsob dekódovania, pri ktorom sa v každom časovom kroku vyberie znak s najvyššou pravdepodobnosťou. Tento prístup je veľmi rýchly, no často nemusí viesť k najlepšiemu možnému výsledku, pretože neberie do úvahy globálny kontext.

2. Beam search (vid. druhý obrázok 4.6) – múdrejší spôsob dekódovania, ktorý si počas predikcie neuchováva len jednu možnosť ako greedy search, ale sleduje viacero najpravdepodobnejších ciest naraz (tzv. hypotézy). Vďaka tomu dokáže lepšie nájsť správny text aj vtedy, keď v niektorých časových krokoch najpravdepodobnejší znak nie je ten správny v celkovom kontexte. Preto býva beam search vo výsledku presnejší ale za to pomalší než greedy search.



Obr. 4.6: Obrázok znázorňuje výber najpravdepodobnejšieho znaku pomocou greedy search. V každom časovom kroku t model vygeneruje pravdepodobnosti $y_1 \dots y_n$ pre všetky znaky vrátane tokenu `<BLANK>`, kde n je veľkosť slovníka. Funkcia `argmax` vyberie index s najvyššou pravdepodobnosťou – teda predikovaný token y_{pred}, ktorý sa následne premapuje na znak zo slovníka.



Obr. 4.7: Obrázok znázorňuje výber najpravdepodobnejšieho znaku pomocou beam search [39]. V každom časovom kroku sa zachová viacero najpravdepodobnejších sekvencií (hypotéz), ktoré sa postupne rozširujú o nové znaky. Na konci sa vyberie najpravdepodobnejšia cesta.

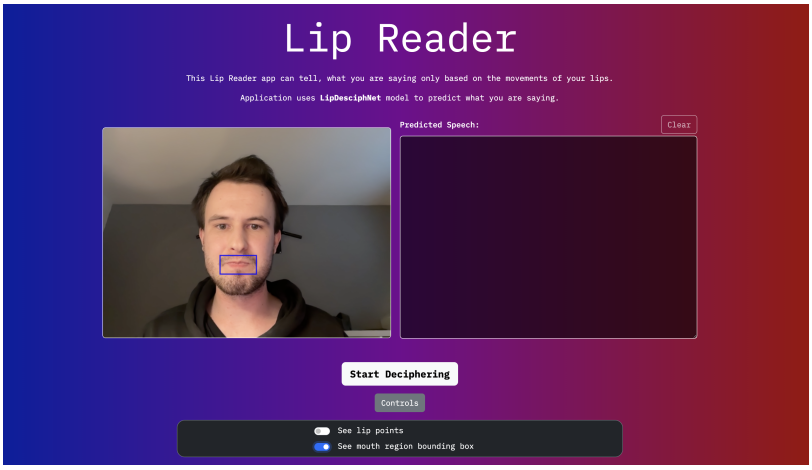
V praxi sa v LipDesciphNet používa nasledovné nastavenie:

- počas trénovania sa používa iba greedy search, keďže slúži najmä na orientačné sledovanie vývoja presnosti. Je užitočné vedieť, ako sa modelu darí bez pomoci sofistikovanejšieho algoritmu beam search,
- počas validácie, testovania a aj pri finálnom nasadení modelu sa využíva beam search, pretože dosahuje výrazne lepšie výsledky.

Použitá implementácia CTC beam search dekódera [37] pochádza z knižnice torchaudio, ktorá poskytuje optimalizovaný a flexibilný nástroj na dekódovanie CTC výstupov bez nutnosti zarovnania vstupu a výstupu. Táto metóda predstavuje kompromis medzi výpočtovou náročnosťou a presnosťou, čo ju robí ideálnou vo fáze inferencie.

4.5 Výsledná aplikácia

Výsledkom tejto práce je interaktívna webová aplikácia, ktorá v reálnom čase rozpoznáva, čo používateľ hovorí, a to len na základe pohybov jeho pier. Využíva pritom webkameru, model LipDesciphNet (model vysvetlený vyššie 4.4) a intuitívne používateľské rozhranie. Pri návrhu aplikácie bol kladený dôraz na jednoduchosť, zrozumiteľnosť a čo najmenšie oneskorenie medzi pohybom pier a zobrazeným textom. Obrázok nižšie (vid. 4.8) ukazuje, ako celková aplikácia vyzerá.



Obr. 4.8: Ukážka výslednej aplikácie.

Frontend aplikácie

Grafické používateľské rozhranie je postavené na technológii React, pričom na vizuálne prvky bol použitý CSS framework Bootstrap. Používateľ má k dispozícii jednoduché ovládacie prvky, pomocou ktorých môže:

- povoliť alebo zakázať prístup ku kamere,
- zapnúť alebo vypnúť zobrazenie detekovaných bodov na perách a orámovanie oblasti úst,
- spustiť alebo zastaviť proces rozpoznávania hovoreného slova,
- sledovať výsledok rozpoznávania v textovom poli v reálnom čase.

Aplikácia využíva knižnicu MediaPipe (FaceMesh) na detekciu bodov na tvári – konkrétne na perách. Na základe týchto bodov sa vypočíta ohraničujúci rámček okolo úst, z ktorého sa každých 40 ms (čo zodpovedá snímkovej frekvencii 25 snímok za sekundu) extrahuje aktuálna oblasť úst (extrakcia oblasti úst v sekcii 4.2).

Tieto oblasti úst sa následne ukladajú do kruhového zásobníka s kapacitou 75 snímok, čo predstavuje 3 sekundy videa. Keď je zásobník plný, celý tento úsek sa odošle na server na spracovanie. Po odoslaní sa zásobník „posunie“ o 25 snímok dopredu (t.j. o jednu sekundu) a proces pokračuje ďalej. Znamená to, že prvý odoslaný úsek obsahuje snímky 0 až 74, nasledujúci 25 až 99, potom 50 až 124, a tak ďalej. Takýmto spôsobom sa nové dáta nepretržite odosielajú bez potreby čakať na úplné vyprázdnenie zásobníka.

Backend aplikácie

Serverová časť aplikácie je vytvorená v jazyku Python s využitím moderného webového frameworku FastAPI, ktorý umožňuje jednoducho a efektívne budovať rýchle API služby. Hlavnou úlohou backendu je spracovať sekvenciu oblastí úst, ktoré prichádzajú z frontendu, a pomocou predtrénovaného modelu LipDesciphNet predpovedať, čo používateľ práve povedal. Celý proces funguje nasledovne:

1. Klient pošle požiadavku, ktorá obsahuje 75 snímok oblastí úst vo formáte Base64 – ide o výrezy z videa, v ktorom používateľ rozpráva.

2. Každý obrázok je na serveri najskôr dekódovaný, prevedený na numerický tenzor a upravený tak, aby zodpovedal rozmerom a formátu požadovanému modelom.

3. Všetkých 75 snímok sa následne odošle do modelu LipDesciphNet, ktorý predpovedá, čo používateľ povedal iba na základe pohybu pier

4. Model vráti pravdepodobnostnú sekvenciu znakov, teda predikciu, čo mohlo byť vyslovené.

5. Na premenu tejto sekvencie na bežný text sa použije dekóder CTC beam search z knižnice torchaudio (vid. 4.4). Ten zabezpečí čo najpresnejší výsledok.

6. Výsledná predikcia – teda rozpoznaná veta – sa odošle späť klientovi vo forme JSON odpovede.

Aby bola zabezpečená bezproblémová komunikácia medzi frontendom (napr. bežiacim na localhost:5173) a backendom (na localhost:8000), server je nastavený tak, aby podporoval CORS (Cross-Origin Resource Sharing). Vďaka tomu môžu obe časti aplikácie spolu komunikovať aj keď bežia na rôznych portoch.

4.6 Zhrnutie

V tejto kapitole bol predstavený celý proces vytvárania systému na čítanie z pier – od výberu a spracovania dát až po návrh modelu LipDesciphNet a jeho nasadenie do reálnej aplikácie. Najskôr bol popísaný použitý dataset LRS2, ktorý obsahuje videá hovoriacich v angličtine a ich transkripcie. Ukázalo sa, že nie všetky časti datasetu sú vhodné na použitie – napríklad predtrénovacia množina bola pre nižšiu kvalitu dát vynechaná.

Ďalej bola detailne rozobratá príprava dát. Z videí sa pomocou MediaPipe FaceMesh extrahovala iba oblasť úst, keďže práve tá je kľúčová pre rozpoznávanie hovorených slov. Zároveň boli transkripcie normalizované a prevedené na číselnú formu pomocou tokenizácie. Tieto kroky boli nevyhnutné na to, aby dáta mohli byť spracované neurónovou sieťou.

Nasledovalo predspracovanie, pri ktorom sa zabezpečila jednotná veľkosť obrázkov aj dĺžka sekvencií. V prípade kratších videí sa sekvencie dopĺňali prázdnymi snímkami, aby mohli byť spracované naraz. Spomenuté boli aj techniky augmentácie, ktoré pomáhajú modelu lepšie sa vyrovnať s rôznorodosťou reálneho sveta.

Dôležitou časťou kapitoly bola architektúra modelu LipDesciphNet. Tento model spája viacero moderných prístupov – 3D konvolučné vrstvy zachytávajú pohyby pier v čase, Highway sieť napomáha prenášaniu dôležitých informácií, obojsmerná GRU spracúva sekvenciu z oboch smerov a kaskádový dekóder s mechanizmom pozornosti a stratou CTC sa stará o presné dekódovanie výsledného textu. Popísané boli aj stratégie dekódovania, ktoré model využíva pri predikcii.

Na záver bol predstavený praktický výstup – webová aplikácia postavená na technológii React, v ktorej môže používateľ rozprávať pomocou kamery a systém mu v reálnom čase prepisuje hovorené slová na text. Táto aplikácia je dôkazom, že celý systém funguje nielen teoreticky, ale aj v praxi.

Kapitola 5

Testovanie

Po navrhnutí a implementácii modelu LipDesciphNet nasleduje dôležitá fáza overovania jeho funkčnosti – testovanie. Cieľom tejto kapitoly je zhodnotiť, ako dobre si model vedie pri rozpoznávaní hovoreného textu na základe pohybov pier, a najmä ako dokáže generalizovať na nové, doposiaľ nevidené dáta.

Kapitola začína popisom konfigurácie experimentov, použitých výpočtových zdrojov a konkrétnych hyperparametrov, ktoré boli pri jednotlivých testovaniach skúmané. Nasleduje predstavenie metrík, ktoré slúžia na objektívne vyhodnotenie kvality predikcií, vrátane najčastejšie používaných ukazovateľov v úlohách rozpoznávania reči – WER, CER a Exact Match.

Dôležitou súčasťou kapitoly je prezentácia výsledkov testovania modelu LipDesciphNet v rôznych konfiguráciách. Sledované sú najmä verzie modelu označené ako A a B, ktoré dosiahli najstabilnejšie výsledky v porovnaní s ostatnými. Súčasťou analýzy sú aj grafy trénovacej a validačnej straty, tabuľky s konkrétnymi metrikami a porovnanie rôznych nastavení učenia.

Na záver kapitola obsahuje diskusiu o výkonnosti modelu, identifikáciu jeho limitácií a návrhy na ďalšie zlepšenie v budúcnosti.

5.1 Konfigurácia testovaní

Testovanie modelu bolo realizované s cieľom overiť jeho schopnosť generalizácie na nevidených dátach. Výpočty prebiehali na grafických kartách NVIDIA A40 48 GB a NVIDIA Tesla T4 16 GB prostredníctvom GPU servera, ktorý poskytla výpočtová infraštruktúra MetaCentrum [9].

Na trénovanie modelu boli použité trénovacia a validačná množina už spomínaného datasetu LRS2 4.1. Trénovacia množina slúžila na optimalizáciu parametrov modelu, zatiaľ čo validačná množina bola využívaná počas tréningu na sledovanie vývoja metrík a validačnej straty s cieľom predchádzať pretrénovaniu. Záverečné vyhodnotenie bolo vykonané na testovacej množine tohto datasetu.

Vo všetkých testovaných konfiguráciách boli použité augmentované trénovacie dáta. Počiatočné experimenty s neaugmentovanými dátami ukázali výrazné pretrénovanie modelu už po niekoľkých epochách, čo viedlo k zlej generalizácii. Z tohto dôvodu bola do trénovacieho procesu zaradená dátová augmentácia (augmentácie, ktoré boli aplikované, sa preberajú v sekcii 4.3.

Počas testovania boli experimentálne porovnávané rôzne konfigurácie modelu, pričom sa sledovali tieto variácie hyperparametrov (hyperparametre vysvetlené v sekcii 3.2):

- Batch size – 32 a 64.
- Learning rate – $1 \cdot 10^{-4}$ a $5 \cdot 10^{-4}$.

Optimalizácia prebiehala pomocou optimalizátora AdamW, pričom na všetky experimenty bol aplikovaný mechanizmus early stopping, gradient clipping a weight decay bol nastavený na hodnotu $1 \cdot 10^{-3}$. Learning rate sa adaptívne upravoval na základe validačnej straty pomocou plánovača (*scheduleru*) ReduceLROnPlateau, ktorý pri stagnácii tejto straty znižoval hodnotu learning rate o polovicu.

5.2 Hodnotiace metriky

Na vyhodnotenie presnosti a spoľahlivosti modelu boli použité viaceré metriky, ktoré reflektujú kvalitu predikovaných transkripcií – to, čo rečník v skutočnosti povedal [19]:

- Word Error Rate (WER) – pomer chybne rozpoznaných slov vzhľadom na celkový počet slov v referenčnej (správnej) vete. Táto metrika sa štandardne využíva v úlohách rozpoznávania reči:
$$\text{WER} = \frac{S + D + I}{N} \tag{5.1}$$
kde S je počet substitúcií (nesprávne rozpoznané slová), D počet vynechaných slov, I počet vložených slov a N počet slov v referenčnej vete. Platí, že $N = S + D + C$, kde C je počet správne rozpoznaných slov.
- Character Error Rate (CER) – podobná metrika ako WER, ale pracuje na úrovni znakov. Je obzvlášť vhodná pri hodnotení modelov, ktoré generujú výstupy po jednotlivých znakoch, alebo pri krátkych slovách:
$$\text{CER} = \frac{S + D + I}{N} \tag{5.2}$$
kde význam S, D, I a N je rovnaký ako pri WER.
- Presné zhody (Exact Match) – podiel viet, ktoré boli predikované úplne bez chyby, teda sú identické s referenčnou transkripciou. Ide o najprísnejšiu metriku.

Tieto metriky umožňujú komplexne posúdiť výkonnosť modelu z rôznych hľadísk – od presnosti rozpoznávania jednotlivých znakov až po správnosť celých viet.

5.3 Výsledky testovaní

Počas testovania boli vyhodnotené viaceré verzie modelu, ktoré sa líšili konfiguráciou hyperparametrov a použitím augmentácie (ako je uvedené v sekcii 5.1). V tejto časti sú prezentované iba tie verzie, ktoré dosahovali najsľubnejšie výsledky z hľadiska generalizačnej schopnosti modelu.

Menej úspešné experimenty – napríklad verzie bez augmentácie alebo s výrazne vyššou hodnotou learning rate $5 \cdot 10^{-4}$ – viedli k zníženej výkonnosti modelu, a preto neboli ďalej podrobne analyzované. V niekoľkých prípadoch sú však tieto horšie výsledky uvedené ilustratívne, aby demonštrovali konkrétne dôvody zlyhania alebo nestability učenia.

Najviac sľubné verzie modelu boli označené písmenami A a B a ich konfigurácie sú uvedené v tabuľke 5.1:

Tabuľka 5.1: Konfigurácie jednotlivých verzií modelu

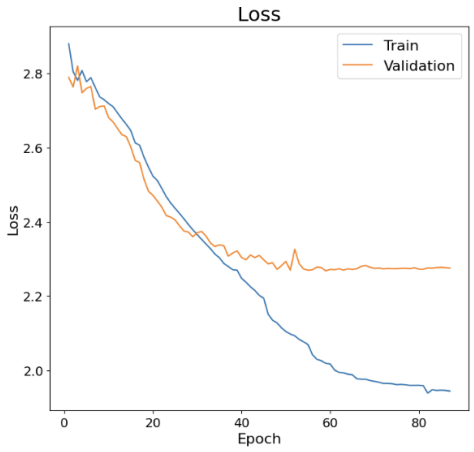
Verzia	**Batch size**	**Learning rate**	**Weight decay**
A	64	$1 \cdot 10^{-4}$	$1 \cdot 10^{-3}$
B	32	$1 \cdot 10^{-4}$	$1 \cdot 10^{-3}$

Výsledky počas trénovania

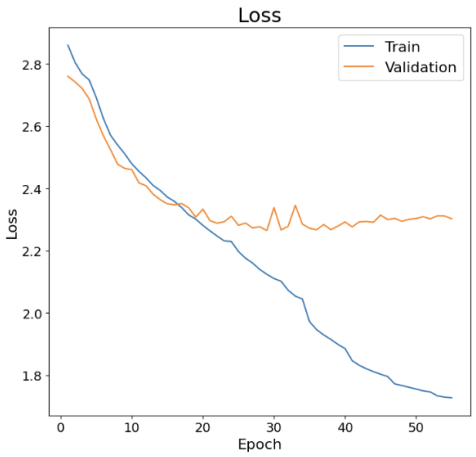
Pre každú verziu modelu boli počas trénovania sledované trénovacie a validačné metriky:

- celková strata (*loss*),
- WER,
- CER a,
- Exact Match.

V nasledujúcich grafoch je zobrazený priebeh hodnoty straty po jednotlivých epochách na trénovacej aj validačnej množine, aby bolo možné sledovať, ako jednotlivé modely konvergovali.



Obr. 5.1: Priebeh straty modelu verzie A.



Obr. 5.2: Priebeh straty modelu verzie B.

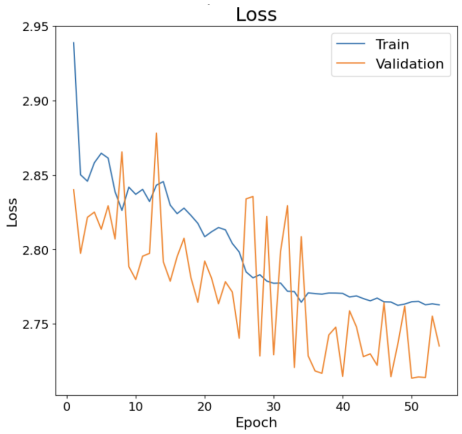
Z grafov na obrázku 5.1 možno vidieť, že v prípade oboch verzií modelu došlo počas trénovania k postupnému znižovaniu trénovacej CTC straty, čo naznačuje, že modely sa učili efektívne. V určitom bode však validačná strata prestala vykazovať zlepšenie – validačná strata sa ustálila a začala oscilovať okolo stabilnej hodnoty. Tento jav nenaznačuje klasický overfitting, keďže nedošlo k výraznému zhoršeniu validačnej straty, ale skôr k stagnácii učenia. V niektorých prípadoch môže validácia mierne kolísať smerom nahor, no celkovo sa

strata pohybuje v približne rovnakej oblasti. Tabuľka nižšie 5.2 uvádza konkrétne hodnoty strát a metrík u jednotlivých verzií modelov:

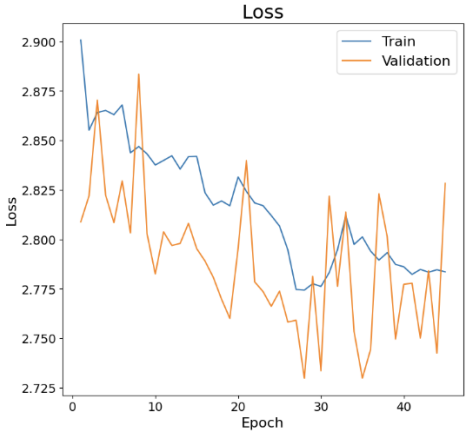
Tabuľka 5.2: Hodnoty metrík na trénovacej a validačnej množine pre jednotlivé verzie modelov.

Verzia	**Strata**		**WER**		**CER**		**Exact Match**		**Epocha**
	Train	Val	Train	Val	Train	Val	Train	Val	
A	1,9444	2,2758	96,65 %	101,55 %	56,29 %	61,65 %	0,19 %	0,1 %	87
B	1,7279	2,3027	91,36 %	101,04 %	49,48 %	60,26 %	0,45 %	0,1 %	55

Na druhej strane, pri niektorých verziách modelu, ktoré nie sú zahrnuté medzi finálne analyzované (napríklad pri learning rate $5 \cdot 10^{-4}$), bol priebeh učenia výrazne nestabilnejší. Validačná strata, ako ukazujú grafy nižšie 5.3, pri týchto konfiguráciách neustále kolísala, často veľmi prudko – bez náznaku konvergencie alebo ustálenia.



Obr. 5.3: Priebeh učenia verzie s veľkosťou dávky 64, augmentáciou, weight decay $1 \cdot 10^{-3}$ a rýchlosťou učenia $5 \cdot 10^{-4}$, kde možno pozorovať silne nestabilnú validáciu.



Obr. 5.4: Priebeh učenia verzie s veľkosťou dávky 32, augmentáciou, weight decay $1{\cdot}10^{-3}$ a rýchlosťou učenia $5{\cdot}10^{-4}$, kde taktiež možno pozorovať silne nestabilnú validáciu.

Už v prvých niekoľkých epochách bolo možné pozorovať, že model sa nevie „trafiť“ do vhodného smeru optimalizácie, čo naznačuje problémy s výpočtom alebo aplikáciou gradientu pri príliš vysokej rýchlosti učenia. Tento jav sa prejavil najmä prudkým poklesom aj vzostupom straty medzi susednými epochami, čo viedlo k nepredvídateľnému správaniu modelu počas učenia a znemožnilo dosiahnutie stabilného minima validačnej straty.

Výsledky počas testovania

Pre finálne zhodnotenie výkonnosti modelov bolo testovanie realizované na testovacej množine datasetu. Vyhodnocovali sa rovnaké metriky ako počas trénovania – WER, CER a Exact Match. Cieľom bolo overiť, ako dobre sú modely schopné generalizovať na nevidené dáta. V tabuľke 5.3 sú uvedené finálne výsledky testovania pre jednotlivé verzie modelov:

Tabuľka 5.3: Finálne hodnoty metrík na testovacej množine pre jednotlivé verzie modelov.

Verzia	WER	CER	Exact Match
A	100,9 %	59,2 %	0,01 %
B	99,7 %	57,6 %	0,01 %

5.4 Analýza výsledkov

Výsledky dosiahnuté počas trénovania aj testovania poukazujú na výrazné limity v schopnosti modelov generalizovať. Hoci priebeh trénovacej straty v oboch verziách A a B (obr. 5.1) ukazuje plynulé znižovanie, validačná strata sa po určitom bode ustálila a ďalej sa nezlepšovala. Tento jav nenaznačuje klasické preučenie, ale skôr stagnáciu učenia – modely nedokázali ďalej optimalizovať na validačných dátach, čo sa potvrdilo aj slabými metrikami na všetkých množinách.

Z hľadiska hodnotenia výkonnosti sú výsledky oboch verzií veľmi slabé. Hodnoty WER presahujú 99 %, čo znamená, že takmer každé slovo bolo rozpoznané nesprávne. Rovnako vysoké sú aj hodnoty CER (nad 57 %), čo indikuje, že ani rozpoznávanie jednotlivých znakov nebolo dostatočne spoľahlivé. Extrémne nízke skóre Exact Match (0,01 %) napokon ukazuje, že takmer žiadny výstup modelu sa úplne nezhodoval s cieľovou transkripciou.

Pri porovnaní verzií A a B (tabuľka 5.2) možno pozorovať, že aj keď verzia B dosiahla o niečo vyššiu validačnú stratu ako verzia A, vykazuje nižšiu trénovaciu stratu a predovšetkým lepšie výsledky vo všetkých sledovaných metrikách. Zároveň bola verzia B trénovaná výrazne kratší čas (iba 55 epôch oproti 87), čo naznačuje efektívnejšie učenie. Tieto rozdiely sú síce malé, no konzistentné naprieč všetkými metrikami, a preto možno konštatovať, že verzia B bola o niečo úspešnejšia než verzia A.

Ďalšie experimenty s vyššou hodnotou learning rate ($5 \cdot 10^{-4}$) nepriniesli lepšie výsledky. Naopak, grafy 5.3 ukazujú, že strata pri týchto verziách oscilovala bez zreteľného zlepšenia – modely boli nestabilné už v počiatočných epochách. Prudké výkyvy v strate medzi susednými epochami naznačujú, že gradienty boli príliš silné, čo viedlo k chaotickému správaniu optimalizácie. Takýto priebeh jednoznačne znemožňuje efektívne učenie.

Celkovo možno konštatovať, že žiadna z testovaných konfigurácií nedosiahla akceptovateľné výsledky a všetky modely zlyhali pri generalizácii na testovaciu množinu.

Možnosti zlepšenia a budúca práca

Na základe analýzy výsledkov je zrejmé, že výkonnosť testovaných modelov bola nedostatočná a vyžaduje si ďalšie vylepšenia. Do budúcnosti by bolo vhodné zamerať sa na nasledujúce oblasti:

- Zlepšenie architektúry modelu – aktuálne použitá architektúra pravdepodobne nedisponuje dostatočnou kapacitou na zachytenie komplexnosti úlohy vizuálneho rozpoznávania reči. Zvážiť možno nasadenie hlbších modelov, využitie predtrénovaných CNN alebo 3D-CNN architektúr (napr. ResNet, R(2+1)D), prípadne ich doplnenie o rekurentné vrstvy (LSTM/GRU) alebo transformerové dekódovanie.

- Dlhšie a stabilnejšie trénovanie – všetky modely boli trénované iba počas niekoľkých desiatok epôch. Pri modeloch A aj B bolo pozorované postupné znižovanie trénovacej

straty, zatiaľ čo validačná strata sa po určitom čase ustálila, no výrazne sa nezhoršovala. To naznačuje, že modely sa stále učili a ďalšie trénovanie – spolu s pokročilou inicializáciou a efektívnymi plánovačmi učenia (napr. *CosineAnnealing*, *OneCycle*) – by mohlo prispieť k zlepšeniu validačnej výkonnosti.

- Rozšírenie augmentačných techník – v testovaných konfiguráciách bola použitá iba základná forma augmentácie. Skúmanie vplyvu pokročilých metód, ako sú *random erasing*, *frame dropping*, *frame jittering* (spomalenie alebo zrýchlenie videa) či časovo konzistentná augmentácia vo videosekvenciách, by mohlo zlepšiť robustnosť a generalizáciu modelov.

- Širšia hyperparametrická variácia – testované boli len dve hodnoty veľkosti batch size a learning rate. Pre precíznejšie doladenie modelov by bolo vhodné skúmať aj iné hodnoty parametrov, ako napríklad *weight decay*, otestovať alternatívne plánovače učenia a zvážiť aj techniky ako *learning rate warm-up*.

- Použitie iného datasetu – dataset GRID predstavuje veľmi kontrolované prostredie, ktoré síce umožňuje presnú predikciu, no nereflektuje realistické scenáre bežnej reči. Skúmanie modelov na rôznorodejších a menej štruktúrovaných datasetoch by mohlo lepšie overiť ich praktickú použiteľnosť.

5.5 Zhrnutie

V tejto kapitole bol podrobne analyzovaný proces testovania modelu LipDesciphNet, ktorého cieľom je rozpoznávanie hovorených viet na základe pohybov pier. Testovanie nadväzovalo na predchádzajúce fázy návrhu a implementácie a slúžilo ako overenie schopnosti modelu generalizovať na nové, doposiaľ nevidené dáta.

Najskôr bola predstavená konfigurácia testovacích experimentov vrátane použitých výpočtových zdrojov, množín dát a konkrétnych nastavení hyperparametrov batch size a learning rate. Zároveň bola zdôraznená potreba dátovej augmentácie, bez ktorej dochádzalo k výraznému pretrénovaniu modelu. Na hodnotenie výkonnosti boli použité štandardné metriky pre úlohy rozpoznávania reči – WER, CER a Exact Match, ktoré poskytli objektívne a kvantitatívne porovnanie rôznych konfigurácií.

V rámci testovania boli identifikované dve verzie modelu (A a B), ktoré dosahovali najstabilnejšie výsledky. Hoci obe verzie vykazovali pokles trénovacej straty a známky učenia, validačná strata sa po určitom čase ustálila, čo naznačuje stagnáciu optimalizácie. Finálne metriky testovania ukázali veľmi nízku schopnosť generalizácie – hodnoty WER sa pohybovali na úrovni 99 % a CER na úrovni 60 % a viac, pričom presné zhody boli takmer nulové.

Na základe tejto analýzy možno konštatovať, že aktuálna architektúra modelu nedosahuje dostatočnú úroveň spoľahlivosti. V závere kapitoly boli preto načrtnuté konkrétne návrhy na zlepšenie – vrátane prepracovania architektúry, dlhšieho a stabilnejšieho trénovania, rozšírenia augmentačných techník, experimentovania s ďalšími hyperparametrami a overenia výkonu modelu na iných datasetoch.

Celkovo testovanie ukázalo, že vytvorený systém síce dokáže učiť sa z dát a stabilne trénovať, no jeho schopnosť preniesť naučené poznatky na nové vstupy je zatiaľ obmedzená. Výsledky tak predstavujú dôležitý východiskový bod pre budúcu prácu a ďalšie vylepšovanie modelu LipDesciphNet.

Kapitola 6

Záver

Cieľom tejto bakalárskej práce bolo vytvoriť systém schopný rozpoznať hovorené slová výlučne na základe pohybov pier, teda bez použitia zvukového vstupu. Práca sa sústredila na spracovanie videozáznamu, presnú extrakciu oblasti úst, úpravu dát do vhodného formátu a návrh modelu, ktorý dokáže vizuálny vstup previesť na textový výstup.

Z technického hľadiska sa podarilo úspešne naplniť všetky stanovené ciele. Bol vyvinutý plne funkčný systém na spracovanie dát z verejne dostupnej množiny LRS2 – od detekcie tváre a extrakcie oblasti pier až po transformáciu údajov do podoby vhodnej na trénovanie neurónovej siete. Navrhnutý model kombinuje 3D konvolučnú sieť, obojsmerné GRU vrstvy a dekódovanie využívajúce CTC a mechanizmus pozornosti. Model sa podarilo úspešne natrénovať a bol schopný generovať výstupy na základe vizuálnych podnetov.

Hoci systém po technickej stránke funguje správne, dosiahnutá presnosť nepostačuje na praktické nasadenie. Výstupy modelu boli nepresné a nestabilné, čo poukazuje na nedostatočnú schopnosť generalizácie. Napriek tomu predstavuje vytvorený systém solídny základ pre ďalší vývoj a výskum v oblasti vizuálneho rozpoznávania reči.

Na tejto práci si cením možnosť dôkladne sa oboznámiť s touto problematikou a získať cenné skúsenosti – od spracovania videozáznamov, cez návrh architektúr v prostredí PyTorch, až po analýzu správania modelov v rôznych podmienkach.

Do budúcna by som chcel pokračovať vo vylepšovaní modelu – skombinovať ho napríklad s predtrénovaným jazykovým modelom, rozšíriť trénovaciu množinu o väčšiu rozmanitosť hovorcov a scén, a tiež sa viac zamerať na dôvody, prečo predikcie nefungovali tak, ako by mali. Možností na zlepšenie je veľa – od iných techník augmentácie, cez úpravu vstupov, až po pokročilejšie stratégie učenia. V dlhodobom horizonte by bolo zaujímavé model rozšíriť aj o zvukový vstup a zamerať sa na reálne aplikácie, napríklad ako pomocný nástroj pre ľudí so sluchovým postihnutím.

Literatúra

[1] Abrar, M. A.; Islam, A. N. M. N.; Hassan, M. M.; Islam, M. T.; Shahnaz, C. et al. Deep Lip Reading-A Deep Learning Based Lip-Reading Software for the Hearing Impaired. In: *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)* online. IEEE, November 2019, s. 40–44. ISSN 2572-7621. Dostupné z: https://doi.org/10.1109/R10-HTC47129.2019.9042439. [cit. 2024-12-20].

[2] Afouras, T.; Chung, J. S.; Senior, A. W.; Vinyals, O. a Zisserman, A. Deep Audio-Visual Speech Recognition. *CoRR* online. 1. vyd., December 2018, abs/1809.02108. Dostupné z: http://arxiv.org/abs/1809.02108. [cit. 2025-04-13].

[3] Aggarwal, C. C. *Neural Networks and Deep Learning: A Textbook* online. 2. vyd. Springer Cham, jún 2023. XXIV, 529 s. ISBN 978-3-031-29642-0. Dostupné z: https://doi.org/10.1007/978-3-031-29642-0.

[4] Agrawal, S.; Omprakash, V. R. a Ranvijay. Lip reading techniques: A survey. In: *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* online. IEEE, Júl 2016, s. 753–757. ISBN 978-1-5090-2399-8. Dostupné z: https://doi.org/10.1109/ICATCCT.2016.7912100. [cit. 2024-12-07].

[5] Amidi, A. a Amidi, S. *Recurrent Neural Networks cheatsheet* online. Stanford University, 16. novembra 2018. Dostupné z: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. Prístupné 2025-01-05.

[6] Assael, Y. M.; Shillingford, B.; Whiteson, S. a Freitas, N. de. LipNet: Sentence-level Lipreading. *CoRR* online. 2, 2016, abs/1611.01599. revidované 2016-12-16. Dostupné z: http://arxiv.org/abs/1611.01599. [cit. 2024-12-19].

[7] Bergmann, D. a Stryker, C. *What is an attention mechanism?* online. IBM, 4. decembra 2024. Dostupné z: https://www.ibm.com/think/topics/attention-mechanism. Prístupné 2025-01-20.

[8] Burkov, A. *Hundred-Page Machine Learning Book* online. 1. vyd. Január 2019. 160 s. ISBN 9781999579500. Dostupné z: https://ia802807.us.archive.org/29/items/pdfcoffee.com_the-hundred-page-machine-learning-bookpdf-pdf-free/pdfcoffee.com_the-hundred-page-machine-learning-bookpdf-pdf-free.pdf. [cit. 2024-12-26].

[9] CESNET. *MetaCentrum (MetaVO)* online. 8. apríla 2025. Dostupné z: https://metavo.metacentrum.cz/. Prístupné 2025-05-06.

[10] Chung, J. S. a Zisserman, A. Lip Reading in the Wild. In: Lai, S.-H.; Lepetit, V.; Nishino, K. a Sato, Y., ed. *Computer Vision – ACCV 2016* online. Springer International Publishing, Marec 2017, sv. 10112, s. 87–103. ISBN 978-3-319-54184-6. Dostupné z: https://doi.org/10.1007/978-3-319-54184-6_6. [cit. 2024-12-01].

[11] Easton, R. D. a Basala, M. Perceptual dominance during lipreading. *Perception & Psychophysics* online, November 1982, zv. 32, č. 6, s. 562–570. ISSN 0031-5117. Dostupné z: https://doi.org/10.3758/BF03204211. [cit. 2024-11-27].

[12] Fenghour, S. *Viseme-based Lip-Reading using Deep Learning* online. 2021. 33–34 s. Disertačná práca. London South Bank University. Vedúci práce Chen, D. D. Dostupné z: https://web.archive.org/web/20221117045544id_/https://openresearch.lsbu.ac.uk/download/aa4b981ca00891bf5a3dcfc1f3d5fdb083b399e7aee82a338f8a041b19ded0ea/10103017/SF_Revised_Thesis.pdf. [cit. 2024-11-28].

[13] GeeksforGeeks. *Connectionist Temporal Classification* online. GeeksforGeeks, 27. decembra 2023. Dostupné z: https://www.geeksforgeeks.org/connectionist-temporal-classification/. Prístupné 2025-05-01.

[14] GeeksforGeeks. *Introduction to Recurrent Neural Networks* online. GeeksforGeeks, 15. novembra 2024. Dostupné z: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/. Prístupné 2025-01-04.

[15] Gupta, S. *7 Common Loss Functions in Machine Learning* online. Whitfield, Brennan. Built In, 13. decembra 2024. Dostupné z: https://builtin.com/machine-learning/common-loss-functions. Prístupné 2024-12-30.

[16] H, R. S. *3D Convolutional Neural Networks (3D CNNs) to Transform Data Analysis* online. Intuitive Tutorials, 1. júla 2024. Dostupné z: https://intuitivetutorial.com/2024/07/01/3d-convolutional-neural-networks-3d-cnns-to-transform-data-analysis/. Prístupné 2025-01-03.

[17] Hao, M.; Mamut, M.; Yadikar, N.; Aysa, A. a Ubul, K. A Survey of Research on Lipreading Technology. *IEEE Access* online. IEEE, November 2020, zv. 8, s. 204518–204544. ISSN 2169-3536. Dostupné z: https://doi.org/10.1109/ACCESS.2020.3036865. [cit. 2024-12-22].

[18] Jarraya, I.; Werda, S. a Mahdi, W. Lip tracking using particle filter and geometric model for visual speech recognition. In: *2014 International Conference on Signal Processing and Multimedia Applications (SIGMAP)* online. IEEE, August 2014, s. 172–179. ISBN 978-9-8985-6596-9. Dostupné z: https://ieeexplore.ieee.org/document/7514498. [cit. 2024-11-28].

[19] Kolena. *WER, CER, and MER* online. 24. septembra 2024. Dostupné z: https://docs.kolena.com/metrics/wer-cer-mer/. Path: Metrics Glossary; Natural Language Processing. Prístupné 2025-05-06.

[20] Kumar, A. *Greedy Search vs Beam Search Decoding: Concepts, Examples* online. Analytics Yogi, 7. augusta 2023. Dostupné z: https://vitalflux.com/greedy-search-vs-beam-search-decoding-concepts-examples/. Prístupné 2025-05-01.

[21] Lu, Y. a Li, H. Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory. *Applied Sciences* online, 2019, zv. 9, č. 8. revidované 2019-04-09. ISSN 2076-3417. Dostupné z: https://doi.org/10.3390/app9081599. [cit. 2024-11-28].

[22] Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E. et al. *CoRR* online, Jún 2019, abs/1906.08172. Dostupné z: https://doi.org/10.48550/arXiv.1906.08172. [cit. 2025-04-14].

[23] Ma, P.; Wang, Y.; Petridis, S.; Shen, J. a Pantic, M. Training Strategies for Improved Lip-Reading. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* online. IEEE, 2022, s. 8472–8476. ISBN 978-1-6654-0540-9. Dostupné z: https://doi.org/10.1109/ICASSP43922.2022.9746706. [cit. 2025-01-15].

[24] Masarykovy univerzity, I. biostatistiky a analýz Lékařské fakulty. *Matematický model a aktivní dynamika neuronu* online. Matematická biologie, 2024. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickych-dat--umela-inteligence--neuronove-site-jednotlivy-neuron--jednotlivy-neuron--matematicky-model-a-aktivni-dynamika-neuronu. Path: Analýza a hodnocení biologických dat; Umělá inteligence; Neuronové sítě - jednotlivý neuron; Jednotlivý neuron; Matematický model a aktivní dynamika neuronu. Prístupné 2024-12-28.

[25] Mathulaprangsan, S.; Wang, C.-Y.; Kusum, A. Z.; Tai, T.-C. a Wang, J.-C. A survey of visual lip reading and lip-password verification. In: *2015 International Conference on Orange Technologies (ICOT)* online. IEEE, December 2015, s. 22–25. ISBN 978-1-4673-8237-3. Dostupné z: https://doi.org/10.1109/ICOT.2015.7498485. [cit. 2024-12-20].

[26] NVIDIA. *Text (Inverse) Normalization* online. NVIDIA, 4. novembra 2025. Dostupné z: https://docs.nvidia.com/nemo-framework/user-guide/latest/nemotoolkit/nlp/text_normalization/wfst/wfst_text_normalization.html. Path: Home; NVIDIA NeMo Framework Developer Docs; Speech AI Tools; (Inverse) Text Normalization; WFST-based (Inverse) Text Normalization; Text (Inverse) Normalization. Prístupné 2025-04-14.

[27] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR* online, December 2019, s. 8024–8035. Dostupné z: https://doi.org/10.48550/arXiv.1912.01703. [cit. 2025-04-11].

[28] Paul, S.; Lakhani, D.; Aryan, D.; Das, S. a Varshney, R. Lip Reading System for Speech-Impaired Individuals. *International Journal for Multidisciplinary Research (IJFMR)* online, Apríl 2024, zv. 6, č. 2. ISSN 2582-2160. Dostupné z: https://doi.org/10.36948/ijfmr.2024.v06i02.18643. [cit. 2024-12-20].

[29] Phung, V. H. a Rhee, E. J. A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets. *Journal of information and communication convergence engineering* online. The Korea Institute of Information and Communication Engineering, September 2018, zv. 16, č. 3, s. 173–178. revidované 2018-08-31. ISSN 2234-8883. Dostupné z: https://doi.org/https://doi.org/10.6109/jicce.2018.16.3.173. [cit. 2025-01-03].

[30] Radhakrishnan, P. *What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network?* online. TDS Archive, 9. augusta 2017. Dostupné z: https://medium.com/data-science/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a. Prístupné 2024-12-30.

[31] Shah, T. *About Train, Validation and Test Sets in Machine Learning* online. TDS Archive, 6. decembra 2017. Dostupné z: https://medium.com/data-science/train-validation-and-test-sets-72cb40cba9e7. Prístupné 2025-01-20.

[32] Singh, R. *Mastering Next Word Prediction with Recurrent Neural Networks (RNNs)* online. Medium, 11. júla 2021. Dostupné z: https://ravjot03.medium.com/mastering-next-word-prediction-with-recurrent-neural-networks-rnns-107eb914f54d. Prístupné 2025-01-17.

[33] Thein, T. a San, K. M. Lip movements recognition towards an automatic lip reading system for Myanmar consonants. In: *2018 12th International Conference on Research Challenges in Information Science (RCIS)* online. IEEE, Máj 2018, s. 1–6. ISSN 2151-1357. Dostupné z: https://doi.org/10.1109/RCIS.2018.8406660. [cit. 2024-11-28].

[34] Theobald, O. *Machine Learning For Absolute Beginners: A Plain English Introduction (Third Edition)* online. 3. vyd. Január 2021. 180 s. ISBN 978-1913666521. Dostupné z: https://mrce.in/ebooks/Machine%20Learning%20for%20Absolute%20Beginners.pdf. [cit. 2024-12-26].

[35] Thorir Mar, I. Finding the total number of multiply and accumulate operations in a LSTM layer. *Insights into LSTM architecture* online. 10. novembra 2021. Dostupné z: https://thorirmar.com/post/insight_into_lstm/. Prístupné 2025-01-07.

[36] Toolify. *Unlocking Security Secrets: The Power of Automated Lip Reading Technology* online. Toolify, 24. februára 2024. Dostupné z: https://www.toolify.ai/ai-news/unlocking-security-secrets-the-power-of-automated-lip-reading-technology-1824034. Path: Home; AI News; Unlocking Security Secrets: The Power of Automated Lip Reading Technology. Prístupné 2025-12-20.

[37] Torchaudio. *CTC_DECODER* online. PyTorch, 2024. Dostupné z: https://pytorch.org/audio/main/generated/torchaudio.models.decoder.ctc_decoder.html#torchaudio.models.decoder.ctc_decoder. Path: Docs; torchaudio.models.decoder; ctc_decoder; Nightly (unstable). Prístupné 2025-05-01.

[38] Xu, K.; Li, D.; Cassimatis, N. a Wang, X. LCANet: End-to-End Lipreading with Cascaded Attention-CTC. In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)* online. IEEE, Máj 2018, s. 548–555. ISBN 978-1-5386-2335-0. Dostupné z: https://doi.org/10.1109/FG.2018.00088. [cit. 2024-12-20].

[39] Zhang, A.; Lipton, Z. C.; Li, M. a Smola, A. J. *Beam Search* online. Dive into Deep Learning, 2023. Dostupné z: https://d2l.ai/chapter_recurrent-modern/beam-search.html. Path: Home; 10. Modern Recurrent Neural Networks; 10.8 Beam Search. Prístupné 2025-05-01.

[40] Zoric, G. a Pandzic, I. S. Automatic lip sync and its use in the new multimedia services for mobile devices. In: *Proceedings of the 8th International Conference on Telecommunications, 2005. ConTEL 2005.* online. IEEE, Jún 2005, sv. 2, s. 353–358. ISBN 953-184-084-9. Dostupné z: https://doi.org/10.1109/CONTEL.2005.185904. [cit. 2024-12-04].

Príloha A

Obsah priloženého pamäťového média

Priložené pamäťové médium je nasledujúcej štruktúry:

```
BP_matus_pestun/
|-- xpestu00-BP.pdf
|-- xpestu00-BP.zip
|-- LipDesciphNet/
|       |-- extract_mouth_regions.py
|       |-- lrs2_before_implementation.ipynb
|       |-- lrs2_model_implementation.ipynb
|       |-- requirements.txt
|       |-- README.md
|       |-- data/
|       |-- models/
|       |-- src/
|           |-- data_loaders.py
|           |-- frames_processing.py
|           |-- mouth_region_extraction.py
|           |-- visualizations.py
|           |-- torch/
|               |-- dataset.py
|               |-- helpful_functions.py
|               |-- models.py
|               |-- predict.py
|               |-- train.py
|               |-- validate.py
|               |-- visualization.py
|-- web_app/
        |-- README.md
        |-- backend/
        |   |-- helpful_functions.py
        |   |-- main.py
        |   |-- model.pth
        |   |-- model.py
        |   |-- requirements.txt
        |-- frontend/
            |-- *
            |-- src/
                |-- *
                |-- App.tsx
                |-- Constants.ts
                |-- api.ts
                |-- index.css
                |-- components/
                    |-- PredictedSpeechTextArea.tsx
                    |-- WebcamCapture.tsx
```

Popis jednotlivých adresárov a súborov:

- `xpestu00-BP.pdf`: Finálna verzia bakalárskej práce vo formáte PDF.
- `xpestu00-BP.zip`: Zdrojové súbory systému LaTeX na vytvorenie PDF bakalárskej práce.
- `LipDesciphNet/`: Adresár obsahujúci časti kódu, v ktorých sa analyzujú dáta a trénujú modely.
  - `extract_mouth_regions.py`: Jeden z hlavných skriptov, ktorý slúži na extrakciu oblasti úst a normalizáciu transkripcií pre každé video. Výstupom je LMDB databáza.
  - `lrs2_before_implementation.ipynb`: Prvý Jupyter notebook, zameraný na analýzu celkového problému čítania z pier a samotného datasetu. Odporúča sa ho spustiť pred druhým notebookom.
  - `lrs2_model_implementation.ipynb`: Druhý notebook, ktorý obsahuje kód pre trénovanie modelu LipDesciphNet. Odporúča sa ho spustiť až po analýze dát z predchádzajúceho notebooku.
  - `requirements.txt`: Zoznam Python balíčkov potrebných na spustenie oboch notebookov.
  - `README.md`: Technická dokumentácia popisujúca prácu s jednotlivými notebookmi, spúšťanie kódu a prehľad použitých funkcií a tried. Podrobnejší návod na prípravu dát a spustenie skriptov sa nachádza v prílohe B.
  - `data/`: Prázdny adresár, do ktorého je potrebné vložiť dataset LRS2 a extrahované oblasti úst. Podrobnosti viď B.
  - `models/`: Prázdny adresár, kde sa ukladajú checkpointy a verzie modelu LipDesciphNet. Je nevyhnutný pre správne fungovanie trénovania.
  - `src/`: Obsahuje všetky pomocné Python skripty používané v notebookoch.
    - `data_loaders.py`: Funkcie a triedy na načítanie a spracovanie dát z datasetu LRS2.
    - `frames_processing.py`: Predspracovanie a analýza jednotlivých snímok videí.
    - `mouth_region_extraction.py`: Pomocné funkcie na extrakciu oblasti úst zo snímok.
    - `visualizations.py`: Vizualizácia snímok, porovnania a podobne.
    - `torch/`: Adresár, ktorý obsahuje všetky skripty a triedy súvisiace s trénovacou fázou (používané najmä v `lrs2_model_implementation.ipynb`).
      - `dataset.py`: Trieda pre uchovávanie oblastí úst a ich transkripcií ako PyTorch Dataset.
      - `helpful_functions.py`: Pomocné funkcie počas trénovania, ako výpočet metrík, early stopping a pod.
      - `models.py`: Implementácia finálnej architektúry modelu LipDesciphNet.
      - `train.py`, `validate.py`, `predict.py`: Skripty obsahujúce hlavné funkcie pre trénovanie, validáciu a testovanie modelu.
      - `visualization.py`: Vizualizácia strát a metrík jednotlivých verzií modelov alebo počas trénovania.

- `web_app`: Adresár obsahujúci výslednú webovú aplikáciu, teda jej backendovú a frontendovú časť.
  - `README.md`: Technická dokumentácia s popisom fungovania webovej aplikácie a jej jednotlivých častí. Podrobnejší návod na spustenie sa nachádza v prílohe B.
  - `backend/`: Adresár obsahujúci celú backendovú časť webovej aplikácie, implementovanú pomocou FastAPI v jazyku Python.
    * `helpful_functions.py`: Obsahuje pomocné funkcie použité na spracovanie oblastí úst prijatých z frontendovej časti.
    * `main.py`: Hlavný skript, ktorý inicializuje FastAPI a zabezpečuje spracovanie sekvencií oblastí úst z frontendu. Tieto sekvencie sa odovzdajú modelu ako vstup a výsledkom je predikcia vyslovenej vety.
    * `model.pth`: Uložené váhy najlepšej verzie modelu LipDesciphNet, ktoré sa používajú na predikciu toho, čo človek hovorí v reálnom čase.
    * `model.py`: Obsahuje triedu modelu LipDesciphNet a jeho načítanie.
    * `requirements.txt`: Zoznam Python balíčkov potrebných na spustenie backendu webovej aplikácie.
  - `frontend/`: Adresár obsahujúci celú frontendovú časť webovej aplikácie, implementovanú pomocou technológie React.
    * `*`: Označuje ostatné adresáre a súbory automaticky vygenerované Reactom.
    * `src/`: Adresár obsahujúci všetky potrebné TypeScript súbory na vytvorenie responzívnej webovej aplikácie, ktorá extrahuje oblasti úst z webkamery.
      · `*`: Označuje ostatné automaticky vygenerované adresáre a súbory.
      · `App.tsx`: Hlavná rodičovská komponenta, z ktorej sa skladá vizuál celej aplikácie.
      · `Constants.ts`: Súbor obsahujúci všetky konštanty používané v aplikácii, napríklad fixnú dĺžku sekvencie snímok oblastí úst.
      · `api.ts`: Obsahuje funkciu `makePrediction()`, ktorá komunikuje s backendovou časťou aplikácie cez API a získava predikciu z modelu.
      · `index.css`: CSS súbor určený na jednoduché štýlovanie HTML prvkov.
      · `components/`: Adresár obsahujúci dve hlavné komponenty aplikácie.
        - `PredictedSpeechTextArea.tsx`: Komponent slúžiaci na zobrazenie výstupu modelu – teda toho, čo človek pravdepodobne povedal.
        - `WebcamCapture.tsx`: Komponent zodpovedný za inicializáciu webkamery, detekciu oblasti úst a volanie API pre predikciu.

Príloha B

Inštalačný postup a príprava dát pre systém

Táto príloha obsahuje návod na spustenie projektu vrátane nastavenia prostredia pomocou Minicondy na operačnom systéme Linux. Popisuje postup stiahnutia a prípravy datasetu LRS2, spustenie skriptov na spracovanie dát a trénovanie modelu LipDeciphNet. Taktiež opisuje spustenie výslednej webovej aplikácie, ktorá pozostáva z backendu vytvoreného vo frameworku FastAPI a frontendu vytvoreného pomocou Reactu a Vite.

Inštalácia Minicondy a príprava prostredia

Táto časť popisuje kompletný postup od stiahnutia Minicondy až po pripravenie Python prostredia na operačnom systéme Linux. Je to odporúčaný spôsob, ako sprevádzkovať všetky skripty projektu.

V termináli stiahnite a spustite inštalátor Minicondy pre Python 3.12. Pre inú architektúru ako `x86_64` (napr. `aarch64`) je potrebné zvoliť vhodnú verziu inštalačného skriptu dostupnú na oficiálnej stránke[1].

```
$ wget https://repo.anaconda.com/miniconda/Miniconda3-py312_24.1.2-0-Linux-
    x86_64.sh
$ bash Miniconda3-py312_24.1.2-0-Linux-x86_64.sh
```

Po dokončení inštalácie reštartujte terminál alebo načítajte zmeny:

```
$ source ~/.bashrc
```

Overte úspešnosť inštalácie:

```
$ conda --version
```

Vytvorte nové virtuálne prostredie s názvom `lipdesciphnet`:

```
$ conda create -n lipdesciphnet python=3.12
```

Overte úspešné vytvorenie prostredia. Tento príkaz zobrazí všetky dostupné prostredia:

[1] https://www.anaconda.com/docs/getting-started/miniconda/main

```
$ conda env list
```

Aktivujte prostredie:

```
$ conda activate lipdesciphnet
```

Pred inštaláciou závislostí je potrebné nainštalovať balíček `pynini`, ktorý je zásadný pre normalizáciu transkripcií. Tento balík však nie je dostupný pre všetky platformy (napr. `aarch64`) cez `conda-forge`. Ak inštalácia zlyhá, odporúča sa nájsť iný spôsob inštalácie alebo ručne skompilovať balík zo zdrojového kódu.

```
$ conda install -c conda-forge pynini
```

Presuňte sa do koreňového adresára projektu, ktorý slúži pre analýzu dát a trénovanie modelu `./LipDesciphNet/` a nainštalujte všetky potrebné balíčky:

```
$ pip install -r requirements.txt
```

Potom sa presuňte do adresára projektu, ktorý obsahuje backendovú časť výslednej webovej aplikácie `./web_app/backend/` a opäť nainštalujte závislosti:

```
$ pip install -r requirements.txt
```

Pre používanie Jupyter Notebookov nainštalujte rozšírenie. Potom môžete toto prostredie používať ako jadro v Jupyteri:

```
$ conda install ipykernel
$ python -m ipykernel install --user --name=lipdesciphnet
```

Teraz je prostredie pripravené na spustenie skriptov, notebookov aj webovej aplikácie.

Analýza dát a trénovanie modelu

Táto časť sa venuje stiahnutiu a príprave datasetu na čítanie z pier, ktorý bol použitý v tejto práci, spusteniu hlavných skriptov na analýzu dát a trénovanie modelu, ako aj spusteniu výslednej webovej aplikácie.

Stiahnutie a príprava datasetu LRS2

Na analýzu a trénovanie bol použitý dataset LRS2 (angl. *Lip Reading Sentences 2*). Tento dataset nie je súčasťou odovzdaného pamäťového média, pretože jeho šírenie je podmienené licenčnými podmienkami.

Pre získanie tohto datasetu je potrebné navštíviť oficiálnu stránku[2] projektu LRS2 a požiadať o prístup k jeho stiahnutiu.

Po úspešnom získaní a dekompresii datasetu je potrebné skopírovať jeho obsah do adresára `./LipDesciphNet/data/LRS2/`. Ak tento adresár neexistuje, je potrebné ho manuálne vytvoriť. Následne sa presuňte do tohto adresára. Je dôležité, aby adresárová štruktúra vyzerala nasledovne:

[2] https://www.robots.ox.ac.uk/~vgg/data/lip_reading/lrs2.html

```
LRS2/
|--- main/
|--- pretrain/
|--- pretrain.txt
|--- test.txt
|--- train.txt
|--- val.txt
```

Adresár `main/` obsahuje množstvo podadresárov, v ktorých sa nachádzajú trénovacie, validačné a testovacie videá vo formáte `.mp4`, spolu s ich textovými transkripciami vo formáte `.txt`. Adresár `pretrain/` obsahuje rovnakú štruktúru, ale pre predtrénovaciu množinu.

Textové súbory `pretrain.txt`, `test.txt`, `train.txt` a `val.txt` obsahujú relatívne cesty k adresárom, ktoré reprezentujú jednotlivé množiny (predtrénovacia, testovacia, trénovacia a validačná).

Treba mať na pamäti, že tento dataset je pomerne veľký – jeho celková veľkosť je približne 50 GB.

Spustenie skriptov na analýzu dát a trénovanie modelov

Po úspešnej inštalácii datasetu LRS2 a jeho presunutí do adresára `./LipDesciphNet/data/LRS2/` je možné začať spúšťať hlavné skripty, konkrétne notebooky `lrs2_before_implementation.ipynb` a `lrs2_model_implementation.ipynb`.

Odporúča sa najprv spustiť `lrs2_before_implementation.ipynb`, keďže poskytuje dodatočné informácie o datasete, spôsobe manipulácie s dátami, extrakcii oblasti úst, normalizácii transkripcií a podobne. Notebook `lrs2_model_implementation.ipynb` slúži na finálny tréning modelu LipDesciphNet.

Ešte pred ich spustením je však potrebné extrahovať oblasti úst z jednotlivých videí a normalizovať k nim prislúchajúce transkripcie. Na tento účel slúži skript `extract_mouth_regions.py`, ktorý prechádza každé video, extrahuje z každého snímku oblasť úst a ukladá ich ako zoznam obrázkov vo formáte PNG.

Tieto oblasti úst, spolu s informáciou o tom, do akej množiny patria, a s normalizovanými transkripciami, sa ukladajú do LMDB databázy.

Najprv je potrebné prejsť do adresára `./LipDesciphNet/` a spustiť skript:

```
$ python extract_mouth_regions.py ./data/LRS2/ ./data/lrs2_mouth_regions.lmdb
```

Kde `./data/LRS2/` je cesta k LRS2 datasetu a `./data/lrs2_mouth_regions.lmdb` je cesta, kam sa má uložiť LMDB databáza s oblasťami úst a ich transkripciami.

Treba mať na pamäti, že tento proces je výpočtovo náročný a môže trvať niekoľko hodín. V prípade, že sa proces preruší alebo zlyhá, nie je potrebné začínať odznova – stačí opäť spustiť vyššie uvedený príkaz. Všetko, čo už bolo spracované, je uložené v databáze a nebude spracované znova.

Taktiež treba počítať s tým, že výsledná LMDB databáza bude mať približne 60 GB, aj keď oblasti úst sú čo najviac skomprimované. V kombinácii s pôvodným LRS2 datasetom, ktorý má približne 50 GB, to znamená značné nároky na úložný priestor.

Po úspešnej extrakcii oblastí úst a normalizácii ich prislúchajúcich transkripcií môžete bez problémov spustiť hlavné notebooky.

Spustenie výslednej webovej aplikácie

Výsledná webová aplikácia bola testovaná v prehliadači Google Chrome a pozostáva z týchto dvoch častí:

1. Backend (FastAPI) – obsluhuje predikciu pomocou trénovaného modelu `LipDesciphNet`
2. Frontend (React/Vite) – zabezpečuje snímanie oblastí úst pomocou webkamery, odosielanie týchto oblastí do backendu a zobrazenie predikcie na základe pohybov pier

Celý systém je možné spustiť lokálne podľa nasledujúcich krokov.

Spustenie backendu (FastAPI server)

Ak ešte nemáte vytvorené Python prostredie, pozrite si kapitolu B, ktorá popisuje inštaláciu Minicondy a prípravu prostredia. Po nainštalovaní všetkých závislostí sa uistite, že v adresári `./web_app/backend/` sa nachádza aj súbor `model.pth`, ktorý obsahuje váhy najlepšej natrénovanéj verzie modelu LipDesciphNet.

Následne sa presuňte do tohto adresára a spustite backend pomocou príkazu:

```
$ uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

Po úspešnom spustení bude backend dostupný na adrese `http://localhost:8000`.

Spustenie frontendu (React/Vite aplikácia)

Presuňte sa do adresára `./web_app/frontend/`, kde sa nachádza React aplikácia.

Nainštalujte závislosti (vyžaduje sa `npm`):

```
$ npm install
```

Následne spustite vývojový server príkazom:

```
$ npm run dev
```

Po spustení sa v termináli zobrazí adresa, na ktorej beží frontend. Typicky to býva `http://localhost:5173`, čo je predvolený port Vite vývojového servera.

Frontend komunikuje s backendom cez endpoint `/predict` a odosiela mu sekvencie oblastí úst zakódované v base64 formáte. Výsledok predikcie sa následne zobrazí v prehliadači.