VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

VÝVOJ METOD ROZPOZNÁNÍ ČESKÉHO ZNAKOVÉHO JAZYKA METODAMI STROJOVÉHO UČENÍ

DEVELOPMENT OF CZECH SIGN LANGUAGE RECOGNITION METHODS USING MACHINE LEARNING APPROACH

ZKRÁCENÁ VERZE DISERTAČNÍ PRÁCE

SHORT VERSION OF DOCTORAL THESIS

AUTOR PRÁCE — **Ing. JAN ŠNAJDER**

AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. JIŘÍ KREJSA, Ph.D.**

SUPERVISOR

BRNO 2025

Klíčová slova
Rozpoznání znakového jazyka; Strojový překlad; Neuronová síť; Transformer; TensorFlow; MediaPipe

Keywords
Sign language recognition; SLR; Machine translation; Neural network; Transformer; TensorFlow; MediaPipe

Místo uložení rukopisu:
Knihovna VUT-FSI v Brně, Technická 2896/2, 61669 Brno.
Jan Šnajder, 2025
ISBN 80-214-
ISSN 1213-4198

Obsah

1 Úvod **4**

2 Formulace cílů **6**
2.1 Návrh metody pro rozpoznání a klasifikaci české znakové abecedy 6
2.2 Návrh metody pro překlad vymezené komunikační oblasti českého znakového jazyka 7

3 Rozpoznání a klasifikace české znakové abecedy **8**
3.1 Rozpoznání a klasifikace statických písmen jednoruční české znakové abecedy 8
3.1.1 Extrakce příznaků 8
3.1.2 Klasifikace 10
3.1.3 Výsledky 11
3.2 Mobilní aplikace pro překlad jednoruční české znakové abecedy 12
3.2.1 Uživatelské rozhraní 12
3.2.2 Integrace komponent 13
3.2.3 Klasifikace v živém přenosu 13
3.3 Rozpoznání a klasifikace diakritiky české znakové abecedy 13
3.3.1 Datová sada 14
3.3.2 Extrakce příznaků 15
3.3.3 Klasifikace 16
3.3.4 Výsledky 17

4 Rozpoznání a překlad českého znakového jazyka **19**
4.1 Datová sada českého znakového jazyka 19
4.1.1 Sběr dat 19
4.1.2 Automatizovaná příprava dat 20
4.1.3 Výsledky 22
4.2 Překlad českého znakového jazyka 24
4.2.1 Extrakce příznaků 24
4.2.2 Překlad 25
4.2.3 Trénink 26
4.2.4 Výsledky 27

5 Závěr **32**
5.1 Přínosy disertační práce 33

Seznam zdrojů a použité literatury **35**

1 Úvod

Znakový jazyk je velice komplexní způsob komunikace; propojuje nejen pohyb rukou, ale také řeč těla a v neposlední řadě i mimiku. Je to přirozený jazyk pro neslyšící a kromě toho, že je to hlavní způsob jejich dorozumívání, je to i jazyk, ve kterém přemýšlí. Podobně jako u mluveného jazyka, neexistuje univerzální znakový jazyk. Každý znakový jazyk má unikátní gramatiku a vlastní slovní zásobu, která se vyvíjela nezávisle na ostatních jazycích v různých regionech nebo kulturních skupinách. V průběhu let si mnoho zemí vytvořilo vlastní znakový systém, jako například americký znakový jazyk (ASL), britský znakový jazyk (BSL), nebo v případě České republiky, český znakový jazyk (ČZJ). Na rozdíl od mluvených jazyků, které se opírají o sluchové a verbální prvky, znakové jazyky využívají vizuální podněty a prostorové uspořádání k vyjádření významu, často zahrnující simultánní gesta a neverbální signály k interpretaci složitých myšlenek.

Z pohledu rozpoznání lze znakovou řeč rozdělit do dvou částí:

- Prstová abeceda a prstové hláskování (angl. fingerspelling) nahrazující mluvené hláskování je způsob, jak vyjádřit jednotlivá písmena. Jeho hlavní využitím je hláskování vlastních jmen osob nebo míst, případně slov, které ve znakovém jazyce nemají svůj ekvivalent[1]. Prstová abeceda je také užitečná při výuce znakového jazyka, neboť poskytuje pomoc v případě, pokud student nezná nebo si nemůže vzpomenout na konkrétní gesto.
- Znaková gesta jsou základním stavebním kamenem a nezbytnou součástí každého znakového jazyka. Každé gesto lze zhruba přeložit na slovo mluveného jazyka nebo na koncept známý jako glosa – zjednodušená textová reprezentace znaku. Právě znaková gesta se využívají v drtivé většině znakové komunikace. Jejich skládáním vznikají věty, které mají vlastní gramatiku zcela odlišnou od mluvených jazyků.

Obě části jsou nedílnou součástí většiny znakových systémů. Jejich rozpoznávání a překlad je častým předmětem nejrůznějších výzkumů již téměř třicet let [1]. Přístupy k této problematice lze rozdělit do dvou skupin.

První skupinou jsou řešení, která využívají specializovanou instrumentaci. Zde se řadí například použití rukavic vybavených senzorikou, jako jsou akcelerometry a gyroskopy, k detekci pozice jednotlivých prstů [2, 3], nebo metody s využitím 3D technologie Kinect™ [4, 5]. Ačkoliv rozpoznání znakové řeči za pomoci specializované instrumentace obvykle zvyšuje kvalitu rozpoznání, zároveň snižuje použitelnost dané metody v praxi.

Druhou skupinou jsou přístupy, které využívají pouze klasická dvoudimenzionální obrazová data z monokulární kamery. Tyto přístupy získaly pozornost především po úspěších v oblasti rozpoznání mluvené řeči. Prvotní pokusy byly založeny na extrakci příznaků pomocí tradičních metod zpracování obrazu a jejich následné klasifikaci použitím klasifikátorů, jako jsou skrytý Markovův model (angl. Hidden Markov Model – HMM) nebo metoda podpůrných vektorů (angl. Support Vector Machine – SVM) [6, 7]. Ačkoliv tyto metody vykazovaly slibné výsledky, jsou velice výpočetně náročné a schází jim robustnost, převážně při použití na sekvencích obrazů. To se změnilo s nástupem strojového učení, především pak umělých neuronových sítí. Důležitou roli ve vývoji

[1] Znakový jazyk často nemá gesta pro velice úzce zaměřené termíny.

metod rozpoznání znakového jazyka hrají konvoluční neuronové sítě (angl. Convolutional Neural Network – CNN), které, jak se ukázalo, jsou velice efektivní pro extrakci příznaků. V případě statických obrazových dat jsou extrahované příznaky klasifikovány nejčastěji obyčejnými dopřednými neuronovými sítěmi. U sekvencí se využívají speciální architektury rekurentních sítí, jako jsou Gated Recurrent Unit (GRU) [8] nebo Long Short-Term Memory (LSTM) sítě [9], které, díky mechanismům jako vnitřní stav a paměť, dokážou udržet kontext napříč delšími sekvencemi.

V poslední dekádě se k překladu znakového jazyka na mluvený začalo přistupovat stejně jako k jakémukoliv jinému překladu, a to využitím metod zpracování přirozeného jazyka. Zde se začaly uplatňovat především mechanismy založené na pozornosti (angl. attention mechanism), které mají původ v neurovědě a kognitivní psychologii [10]. U strojového překladu tyto mechanismy vylepšily často používané seq2seq modely – enkodér-dekodér architektury mapující vstupní sekvenci na sekvenci výstupní [11]. Na mechanismu pozornosti je založena i architektura Transformer [12], která spustila revoluci na poli strojového překladu. K zachycení vztahů napříč celou vstupní sekvencí simultánně využívá mechanismus sebe-pozornosti (angl. self-attention mechanism), a díky tomu je schopná pokrýt dlouhodobé závislosti.

Tato disertační práce se zabývá vývojem metod strojového učení pro rozpoznání a překlad českého znakového jazyka. Hlavní motivací pro tuto úlohu je propojení světa neslyšících s jeho okolím. K jejímu naplnění je potřeba metod, které budou dostupné a jednoduše použitelné ve většině situací, což vylučuje využití specializované instrumentace. Z tohoto důvodu budou jako vstupy využity pouze dvoudimenzionální obrazová data. Práce se věnuje jak znakové abecedě, tak samotným gestům, neboť obě oblasti jsou nedílnou součástí znakového jazyka. Kromě samotných metod jsou v rámci práce vytvořeny i datové sady.

Práce je uspořádána následovně: na začátku jsou formulovány jednotlivé cíle a jejich dílčí části. Následně je představena vlastní práce, od rozpoznání české znakové abecedy, přes diakritiku až po samotnou metodu pro překlad českého znakového jazyka. V závěru jsou sumarizovány výsledky a přínosy této disertační práce.

2 Formulace cílů

Na základě motivace práce a analýzy současného stavu byly formulovány cíle disertační práce. Bylo rozhodnuto, že práce se bude věnovat jak znakové abecedě, tak gestům znakového jazyka. Formulace spočívá především v upřesnění toho, jakým způsobem budou cíle naplněny a specifikaci dílčích cílů.

2.1 Návrh metody pro rozpoznání a klasifikaci české znakové abecedy

První část disertační práce se věnuje vývoji metody pro rozpoznání české znakové abecedy. Vzhledem k dostupnosti bude využita datová sada české znakové abecedy představená v publikaci [16]. Jedná se o velice rozsáhlou a vhodnou datovou sadu. Díky jejímu použití bude také možné porovnání s metodou představenou v původní publikaci.

Součástí české znakové abecedy je i diakritika. Ta zanáší do znakového hláskování pohyb a v rámci prvního cíle by měla být řešena její klasifikace. V současnosti neexistuje vhodná datová sada pro tuto úlohu, tudíž prvním krokem bude její vytvoření. Metoda pro klasifikaci diakritiky bude založena na některém z pokročilých typů rekurentní sítě, jakými jsou LSTM nebo GRU. Klasifikace diakritiky by mohla sloužit zároveň jako mezikrok a příprava na rozpoznání znakových gest a bylo by vhodné využít metody, které lze aplikovat i na tuto úlohu.

K extrakci příznaků bude využit přístup klíčových bodů pózy znakujícího. Jedná se o inovativní přístup, při kterém je vstupní obraz nahrazen body, které definují klíčové části lidského těla. Tímto krokem jsou nahrazeny konvoluční neuronové sítě trénované na míru na dané úloze. Zároveň se jedná o přístup, který je vhodnější pro případné využití na mobilních platformách. Vzhledem k hlavní motivaci práce, kterou je propojení světa neslyšících s jeho okolím, by vytvořená metoda měla být testována na mobilní platformě. Cílem tohoto testování je dokázat, že všechny součásti dané metody lze na těchto platformách nasadit.

Na základě těchto informací lze vytyčit následující dílčí cíle:

1. Metoda pro rozpoznání české znakové abecedy založená na extrakci příznaku klíčových bodů pózy znakujícího. Pro trénování bude využit datová sada z publikace [16].
2. Ověření funkčnosti metody na mobilní platformě v reálném čase.
3. Vytvoření datové sady, kterou lze využít pro rozpoznání a klasifikaci znakové diakritiky.
4. Metoda pro rozpoznání diakritiky znakové abecedy. Metoda by měla být využitelná společně s metodou vytvořenou v bodě 1.

2.2 Návrh metody pro překlad vymezené komunikační oblasti českého znakového jazyka

Existují dva přístupy pro rozpoznání znakové řeči: rozpoznání izolovaných gest a kontinuální rozpoznání. První zmiňovaný se snaží mapovat znakový jazyk na mluvený jedna ku jedné, a neřeší rozdíly v gramatice a slovosledu. Z hlediska motivace této práce nedává takový přístup smysl, neboť, ač se jedná o komplexní úlohu, její využití v praxi je minimální. Kontinuální rozpoznání se snaží tyto rozdíly řešit a generovat gramaticky správné překlady. Zároveň se jedná o velice aktuální téma, kterému se v posledních pěti letech dostává spoustu pozornosti. Z těchto důvodů se práce zabývá kontinuálním rozpoznáním českého znakového jazyka.

Ačkoliv existují kontinuální datové sady pro německý [18], korejský [19] či čínský [20] znakový jazyk, český znakový jazyk takový zdroj nemá. Prvním krokem k překladu českého znakového jazyka by mělo být vytvoření datové sady, kterou půjde pro tento účel využít. Datová sada by měla mít omezenou slovní zásobu a obsahovat pouze slabě anotovaná data[1]. Publikace [18, 21, 22] ukazují, že při použití sofistikovanějších metod pro překlad lze k učení využít pouze takto anotovaná data. Kromě toho by se cílová datová sada měla skládat z delších vzorků v řádech několika vět či souvětí. Tím by se mohla odlišit od již existujících a pomoci k posunu překladu znakové řeči opět o pár krůčků dál. Celý postup přípravy datové sady by měl být v co nejvyšší míře automatizovaný tak, aby umožnil případná budoucí rozšíření.

Samotný překlad českého znakového jazyka se obdobně jako v případě znakové abecedy bude skládat ze dvou částí: extrakce příznaků a jejich klasifikace. První krok, extrakce příznaků, bude opět využívat klíčové body pózy znakujícího. Díky tomu, že tento přístup bude již použit u znakové abecedy, bude možné využít nabyté zkušenosti a tím urychlit vývoj. Druhý krok spočívá v klasifikaci příznaků. Vzhledem k vytyčeným vlastnostem datové sady českého znakového jazyka bude nutné využít sofistikované modely. Takovým modelem je aktuálně například Transformer, který vykazuje zajímavé výsledky jak v překladu znakového jazyka [21], tak v oblastech strojového překladu, zpracování přirozeného jazyka či generativních modelech.

Uvedené upřesnění tohoto cíle lze v bodech shrnout jako:

1. Kontinuální datová sada českého znakového jazyka, která bude mít následující vlastnosti:
    - omezenou komunikační oblast a tím i slovní zásobu,
    - slabě anotovaná, tudíž bez anotací glos,
    - delší sekvence než existující datové sady,
    - v co největší míře automatizovaný proces tvorby.
2. Metoda pro kontinuální překlad českého znakového jazyka založená na extrakci příznaků v podobě klíčových bodů pózy znakujícího a klasifikaci pomocí modelu Transformer.

[1] Slabě anotovaná data vynechávají anotace glos. Glosa je zjednodušená textová reprezentace jednotlivých gest znakového jazyka.

3 Rozpoznání a klasifikace české znakové abecedy

3.1 Rozpoznání a klasifikace statických písmen jednoruční české znakové abecedy

Česká jednoruční znaková abeceda se skládá z 27 znaků – každé písmeno má jeden znak. Většina těchto znaků je statických, výjimkou je písmeno „J“ a v některých případech i písmeno „K“. Obě tyto výjimky obsahují pohyb ruky během znaku, nicméně u obou lze považovat tvar ruky za unikátní během celého pohybu. Z tohoto důvodu lze celou úlohu zjednodušit na klasifikaci statických obrazových dat.

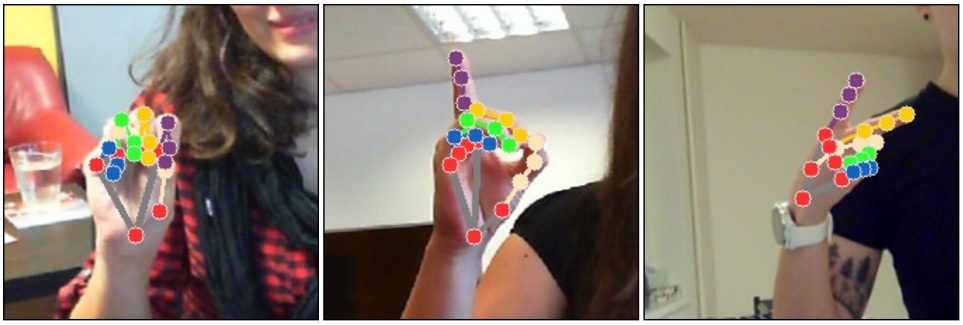
Tato část disertační práce využívá datovou sadu české znakové abecedy představenou v publikaci [16], ve které lze také nalézt shrnutí relevantních parametrů. Její rozsáhlost a zpracování jsou na velice dobré úrovni, díky čemuž je vhodná pro účely této práce.

Navržený přístup se stará jak o rozpoznání, tak o klasifikaci. Jádrem metody je extrakce příznaků v podobě klíčových bodů pózy za pomocí knihovny MediaPipe Hands (MPH) [14]. Výstupem tohoto kroku jsou souřadnice klíčových bodů prstů a dlaně. Tyto souřadnice jsou následně klasifikovány pomocí dopředné neuronové sítě. Výstupem celého procesu je predikované písmeno pro daný obrazový vstup.

3.1.1 Extrakce příznaků

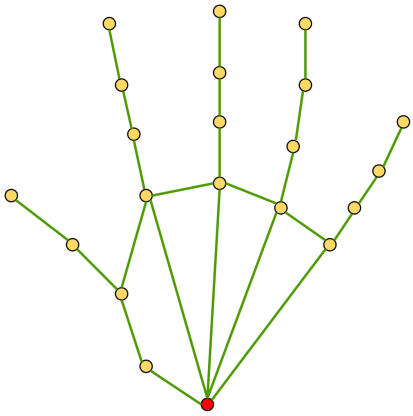
Páteří nejen extrakce příznaků, ale celého algoritmu je knihovna MediaPipe Hands. Ta slouží k lokalizaci a extrakci klíčových bodů rukou a je součástí nástroje MediaPipe [13], což je skupina řešení pro nejrůznější úlohy v oblasti počítačového vidění. Součástí je i implementace v jazyce Python, která byla v této práci využita.

MPH zajišťuje extrakci klíčových bodů rukou pomocí dvou konvolučních neuronových sítí. První má za úkol najít oblast ve vstupním obraze, ve které se nachází dlaň a prsty. Druhá síť je trénována tak, aby v této oblasti detekovala 21 klíčových bodů. Obě sítě byly, dle publikace [14], trénovány na datasetech jak ve venkovním, tak vnitřním prostředí. Součástí tréninkových dat byla i syntetická data. Ukázku vizualizace klíčových bodů na datech z datové sady české jednoruční znakové abecedy lze vidět na obrázku 3.1.



Obrázek 3.1: Vizualizace extrahovaných klíčových bodů, zleva: „S“, „D“, „K“.

Predikce souřadnic probíhá ve všech třech osách. Horizontální a vertikální souřadnice jsou normalizované do intervalu od nuly do jedné. Hloubková souřadnice je vztažena relativně k souřadnici zápěstí. Všech 21 klíčových bodů lze vidět na obrázku 3.2.



Obrázek 3.2: Klíčové body extrahované pomocí MPH.

V rámci extrakce příznaků jsou všechny horizontální a vertikální souřadnice transformovány tak, aby byly relativní k souřadnici zápěstí. Ta je na obrázku 3.2 vyobrazena červeně. Tento krok nejenom unifikuje souřadnicový systém všech tří os, ale také zajišťuje indiferentnost extrahovaných příznaků vůči pozici v obraze. Tudíž nezáleží na tom, kde v obraze se ruka nachází, ale pouze na relativní pozici jednotlivých klíčových bodů vůči zápěstí.

Kromě zmiňovaných 21 souřadnic bodů MPH vrací pravděpodobnost, s jakou je ruka v obraze přítomna a zda se jedná o levou, či pravou ruku. To, o kterou ruku se jedná, je velice užitečná informace a umožňuje klasifikaci levorukých znakujících. V případě, že je detekován levoruký znakující, stačí pouze změnit znaménka u horizontálních souřadnic.

Vzhledem k závislosti navrženého algoritmu na MPH, byla během prvních pokusů zkoumána konfigurace knihovny a její vliv na výslednou klasifikaci. Prvním parametrem, který byl pozorován, byl počet dimenzí. Jak bylo již zmíněno, MPH vrací souřadnice ve třech dimenzích, nicméně souřadnice hloubky je, z velké části, pouze odhadem. Autoři uvádějí, že byla trénována především na syntetických datech, a tudíž v rámci prvotních testů bylo cílem ověřit, jaký (a jestli) má zahrnutí hloubkové informace vliv na výslednou klasifikaci.

Druhým zkoumaným parametrem byla tzv. důvěra detekce (z angl. detection confidence). Jedná se o vstupní parametr MPH, který se zadává v intervalu od nuly do jedné a indikuje, jaká je minimální důvěra, aby mohla být detekce považována za úspěšnou. Výchozí hodnotou důvěry detekce je 0,50. Zároveň však při prvotních manuálních experimentech s knihovnou MPH bylo pozorováno, že vyšší hodnota vede k přesnější estimaci klíčových bodů rukou. V rámci tohoto pokusu byly extrahovány příznaky z celého datasetu pro čtyři různé konfigurace tohoto parametru. Výsledné počty úspěšně extrahovaných příznaků lze vidět v tabulce 3.1.

Výsledné hodnoty skrývají několik kritických informací. Průměrný počet trénovacích obrazů pro jednotlivá písmena se pohyboval okolo 21 tisíc, a toto číslo rapidně klesalo se zvyšující se hodnotou parametru důvěry detekce. Pro nejvyšší pozorovanou hodnotu důvěry 0,8 klesl průměrný počet obrazů, ze kterých byly úspěšně extrahovány příznaky, na 12 536. Navíc při vyšších hodnotách důvěry detekce ztrácela datová sada své

Důvěra detekce	Průměr	Minimum
-	21 381,7	18 994
0,5	20 046,3	16 974
0,6	18 979,3	15 445
0,7	16 765,8	12 326
0,8	12 536,6	6 343

Tabulka 3.1: Počet vzorků, které byly úspěšně extrahovány pro různé konfigurace parametru důvěry detekce.

rovnoměrné rozložení, neboť MPH měl často problémy s detekcí u stejných písmen, kterými byly především „M“, „N“ a „Q“. Všechna tato tři písmena se znakují s dlaní kolmo k pozorovateli, a tudíž je pro MPH často obtížné vůbec ruku detekovat. Na druhou stranu pro nižší hodnoty důvěry detekce 0,5 a 0,6 zůstával počet snímků s úspěšnou extrakcí relativně vysoký (20 046 a 18 979). Nicméně po manuální kontrole bylo zjištěno, že část extrahovaných příznaků byla kvůli nepřesné lokalizaci dlaně nesprávně detekována.

3.1.2 Klasifikace

Pro klasifikaci byla využita klasická dopředná plně propojená neuronová síť. Parametry finální topologie lze nalézt v tabulce 3.2. Síť se skládá z plně propojených vrstev a dropoutu – regularizační vrstvy zabraňující přetrénování. Hodnota dropoutu byla nastavena na 10 %, což znamená, že náhodně vybraných 10 % neuronů předešlé vrstvy je vypnuto – jejich výstup se nastaví na nulu. Díky tomu je síť schopna lépe generalizovat a učit se netriviální závislosti. Na výstupu sítě je vrstva mající 27 neuronů, kde každý představuje jedno písmeno a jejichž hodnoty reprezentují pravděpodobnost, že vstupní data spadají do dané třídy.

Velikost vstupu	Vrstva	Velikost	Akt. funkce	Velikost výstupu
42 / 63	Plně propojená	512	*tanh*	512
512	Dropout	10 %	-	512
512	Plně propojená	256	*tanh*	256
256	Dropout	10 %	-	256
256	Plně propojená	128	*tanh*	128
128	Dropout	10 %	-	128
128	Plně propojená	27	*softmax*	27

Tabulka 3.2: Finální topologie pro klasifikaci české znakově abecedy.

Výše popsaná neuronová síť byla implementována pomocí knihovny TensorFlow [23] v jazyce Python. Trénink využívá stochastický optimalizační algoritmus ADAM (Adaptive Moment Estimation), což je standardně používaný algoritmus zajišťující rychlou a stabilní konvergenci. Ztrátová funkce je realizována pomocí křížové entropie, respektive její varianty sparse categorical crossentropy, která je běžně používaná pro klasifikační úlohy. Velikost jednoho batche byla nastavena na 4096 vzorků – jedná se o počet vzorků, které neuronová síť zpracuje během jedné aktualizace vah. Menší velikost vede k větší variabilitě gradientu, a tím zamezuje uvíznutí v lokálním minimu.

Větší velikosti poskytují stabilnější průběh, nicméně činí optimalizaci výpočetně náročnější.

Průběh tréninku se často určuje po jednotlivých epochách, kde jedna epocha znamená průchod celé trénovací množiny neuronovou sítí. Průběh tréninku vykazoval velice rychlý růst přesnosti v prvních epochách. Po dvaceti epochách bylo trénování zastaveno, neboť začalo docházet k přetrénování.

3.1.3 Výsledky

Výsledná neuronová síť byla testována na datasetu denotovaném jako „PG“. Jedná se o část datové sady, která je v relativně dobré kvalitě a je od znakujícího, který nebyl součástí trénovací množiny. Výsledné přesnosti jsou shrnuty v tabulce 3.3.

Důvěra detekce	**0,5**	**0,6**	**0,7**	**0,8**
2D klasifikace	96,35 %	96,51 %	97,47 %	**97,83 %**
2D klasifikace + extrakce	**93,98 %**	91,44 %	84,92 %	64,94 %
3D klasifikace	96,79 %	97,41 %	97,38 %	**97,83 %**
3D klasifikace + extrakce	**94,41 %**	92,29 %	84,85 %	64,94 %

Tabulka 3.3: Výsledky klasifikace české znakové abecedy pro různé konfigurace extrakce příznaků.

Druhý a čtvrtý řádek tabulky 3.3 popisují úspěšnost klasifikace s užitím souřadnic dvou a tří dimenzí. Výsledky ukazují, že vyšší hodnota důvěry detekce znamená vyšší přesnost klasifikace a pro konfiguraci 0,8 dosahovala přesnost klasifikací až 97,83 %. Nicméně v případě, že vezmeme v potaz i snímky, ze kterých nebyly extrahovány příznaky a označíme je za neúspěšné, pak přesnost klasifikace rychle klesá se zvyšující se hodnotou parametru důvěry detekce.

Konfigurace 0,8 extrahovala souřadnice pouze cca 66 % z testovacích vzorků, a úspěšnost její klasifikace klesla pro tři dimenze z 97,83 % na 64,94 %.

Z výsledků není patrný nějaký zásadní rozdíl mezi využitím dvou nebo tří dimenzí. Ačkoliv klasifikace s třemi dimenzemi vykazuje o něco málo lepší přesnost, skutečnost, že je souřadnice hloubky pouhým odhadem, ji činí nepředvídatelnou a její využití musí být ověřeno v reálné aplikaci.

Z pohledu jednotlivých písmen, přesnost klasifikace se ve většině případů blížila 100 %. Výjimkou je trojice písmen „M“, „N“ a „Q“. Jak již bylo poznamenáno v předchozí kapitole, tyto tři písmena jsou nejen velice podobná, ale zároveň jejich tvar a orientace způsobují problémy MPH při extrakci příznaků. Druhou výjimkou jsou písmena „K“ a „Z“, která se, pro některé znakující, liší pouze v poloze palce. Ten je navíc z velké části schovaný za celou rukou, a tím pádem se může stát, že je špatně lokalizován při extrakci příznaků. To, že se většina chyb pohybuje okolo malého počtu písmen, znamená, že tento neduh lze eliminovat na vyšší aplikační vrstvě.

Dosažené výsledky dokazují, že natrénovaná síť dokáže generalizovat a je schopná klasifikovat znaky znakujícího, kterého do té doby neviděla. Celkově se dá říct, že v případě extrakce klíčových bodů je velice důležitá kvalita vstupních dat. Ze znalostí nabytých při extrakci příznaků lze konstatovat, že nižší hodnoty parametru důvěry detekce jsou více vhodné pro využití v aplikacích s živým záznamem. Ačkoliv konfigurace s vyšší hodnotou

tohoto parametru dosahovaly lepších kvalifikačních výsledků, jejich využití je omezené, neboť z velké části vstupních dat nejsou schopné klíčové body extrahovat.

3.2 Mobilní aplikace pro překlad jednoruční české znakové abecedy

Vzhledem k obecným cílům práce, které směřují k metodám využitelným na mobilních platformách, bylo v dalším kroku přistoupeno k otestování všech částí algoritmů právě na těchto platformách. Jako cílový operační systém byl vybrán operační systém Android, neboť se jedná o nejrozšířenější mobilní operační systém[1]. Hlavním přínosem mobilní aplikace je možnost otestovat nasazení jednotlivých částí algoritmu, především MPH a neuronové sítě vytvořené v knihovně TensorFlow. Dále testování odezvy, funkčnosti a celkově chování představeného řešení na datech v živém přenosu. V rámci testování byl využit mobilní telefon Google Pixel 7a, který byl představen v roce 2023 a jedná se o telefon střední třídy. Pro vývoj aplikace bylo použito vývojové prostředí Android Studio.

3.2.1 Uživatelské rozhraní

Uživatelské rozhraní se navrhuje přímo v prostředí Android Studio, k dispozici je grafický designer, ve kterém lze vytvářet, upravovat a propojovat jednotlivé komponenty. Na pozadí je grafický návrh reprezentován souborem ve formátu XML (Extensible Markup Language), což je textový soubor, strukturovaný podle specifického schématu. Tyto soubory popisují jednotlivé grafické prvky a jejich atributy. V rámci tohoto prostředí bylo vytvořeno jednoduché uživatelské rozhraní. Jako předloha byly použity příklady aplikací z repozitáře MediaPipe[2].

Uživatelské rozhraní poskytuje následující funkcionalitu:

- Lze zvolit mezi vstupem z kamery mobilního telefonu nebo vybrat záznam z galerie telefonu. V prvotní verzi aplikace byl implementován pouze vstup z kamery, neboť se jedná o komplexnější přístup a v případě jeho funkčnosti je případné rozšíření o vstup z galerie velice přímočaré.
- Výstup klasifikace české znakové abecedy. Jedná se o textové pole, které nelze uživatelem editovat. Jsou zde vypisována jednotlivá písmena, která byla predikována klasifikátorem. Tlačítkem RESET se maže současný výstup.
- Vizuální vstup, který zobrazuje aktuální vstup dle zvoleného typu, buďto z kamery nebo z galerie. V příkladu od MediaPipe byly ve vstupu vizualizovány klíčové body ruky. Tato funkcionalita byla odebrána, nicméně mohla by být zapnutá uživatelem v případě potřeby.
- Tlačítko, kterým lze měnit mezi přední a zadní kamerou telefonu.

[1] Podle webu *https://gs.statcounter.com* byl na konci roku 2024 Android nejpoužívanější mobilní operační systém s podílem na trhu 72,24 %.

[2] Dostupné on-line na *https://github.com/google-ai-edge/mediapipe.*

3.2.2 Integrace komponent

Jak bylo zmíněno, celá aplikace vychází z příkladu aplikace pro detekci a zobrazení klíčových bodů rukou od MediaPipe. To znamená, že knihovna MPH je v této aplikaci již zabudována. Jedná se o knihovnu v Javě, která poskytuje poměrně jednoduché rozhraní. Jakmile je schválen přístup ke kameře, je nainicializována třída *HandLandmarksHelper*, která se stará o extrakci klíčových bodů. Samotná exekuce probíhá na pozadí víceméně bez jakéhokoliv zásahu, MPH je napojeno přímo na zdroj obrazu (v tomto případě kameru), ze kterého si po každé extrakci vyžádá nový snímek. Po úspěšné extrakci jsou klíčové body předány aplikační vrstvě pomocí zpětného volání.

Pro prvotní testy byla zvolena neuronová síť trénovaná s hodnotou parametru důvěry detekce 0,6 a zahrnutou hloubkovou informací. Tato síť byla před nasazením převedena do TFLite verze. TFLite (TensorFlow Lite) je odlehčená verze TensorFlow knihovny určená pro nasazení na mobilních či vestavěných zařízeních. TFLite neuronové sítě jsou optimalizované pro nízkou latenci a běh na zařízeních s omezeným výkonem. Obrovskou výhodou je, že Android Studio podporuje externí zdroje ve formě TFLite modelů a díky tomu je celková integrace velice jednoduchá.

3.2.3 Klasifikace v živém přenosu

Extrahované klíčové body jsou zpracovány tak, jak je popsáno v kapitole 3.1.1 a jsou předány neuronové síti pro klasifikaci. Jejím výstupem je vektor, který obsahuje pravděpodobnosti toho, že vstupy spadají do dané klasifikační třídy. Vzhledem k tomu, že výstupem je 20-30 písmen za sekundu, bylo nutné výstupy nejprve zpracovat před tím, než budou prezentovány uživateli.

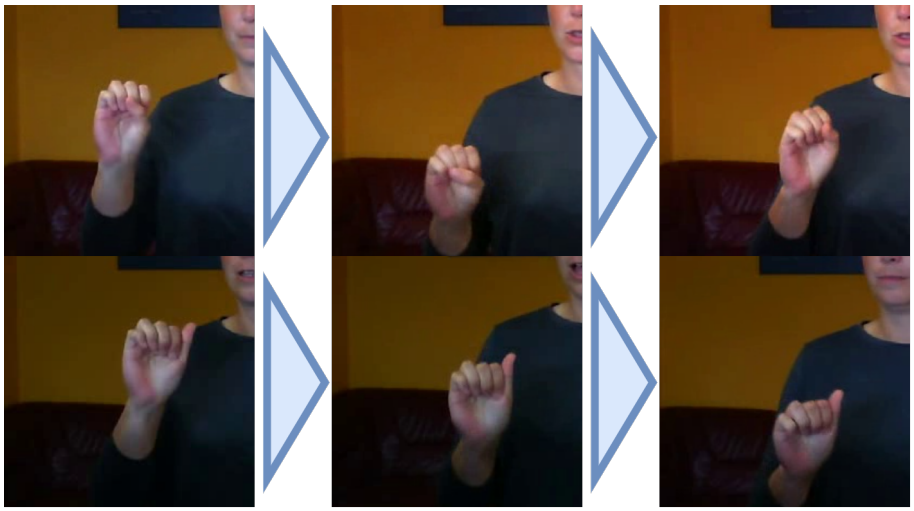
Pro zpracování byl v prvotní fázi použit klouzavý průměr. Výsledné pravděpodobnosti za posledních 10 snímků byly zprůměrovány. Jestliže maximum z těchto pravděpodobností dosáhlo hodnoty 0,75 či vyšší, bylo písmeno, jemuž náležela daná pravděpodobnost, prohlášeno za predikované. Tento relativně jednoduchý přístup se ukázal v praxi jako velice účinný. Ačkoliv algoritmus má své nedostatky, například není možná detekce stejného písmena dvakrát po sobě, většinu z nich lze minimalizovat na vyšší aplikační vrstvě.

Výše uvedený algoritmus byl testován na živém přenosu dat a byla testována odezva celého systému. Při využití procesoru pro extrakci příznaků dosahovaly inferenční časy 60-110 milisekund, což odpovídá 9-17 snímkům za sekundu. Při použití grafické jednotky se inferenční časy snížily na 30-50 milisekund, což je 20-30 snímků za sekundu. V obou případech se jedná o očekávané hodnoty. Čas inference nemá vliv na uživatelskou zkušenost, neboť MPH běží na pozadí a není poznat, že extrakce klíčových bodů je použita jen na některé snímky.

3.3 Rozpoznání a klasifikace diakritiky české znakové abecedy

Kromě statických znaků pro jednotlivá písmena existuje i způsob, jak česká znaková abeceda reprezentuje diakritiku – pohybem ruky ve tvaru daného znaménka, zatímco prsty jsou drženy v rozložení pro dané písmeno. To znamená, že písmeno „á“ je

znakováno jako běžné „a“ s tím rozdílem, že znakující provede pohyb ruky shora dolů, čímž naznačuje přítomnost čárky. Pohyb pro háček i čárku se snaží zachytit obrázek 3.3.



Obrázek 3.3: Ukázka diakritiky, horní řada: „š“, spodní řada: „á“.

Tato kapitola představuje metodu pro klasifikaci diakritiky české znakové abecedy. Jedná se o rozšíření klasifikace české znakové abecedy představené v kapitolách 3.1 a 3.2. Metoda je založená na stejných nástrojích a knihovnách – MediaPipe a TensorFlow. Zároveň díky tomu, že se jedná o úlohu klasifikace obrazových sekvencí, by tato část mohla sloužit jako důkaz konceptu a základní stavební kámen pro klasifikaci gest znakového jazyka.

Navržený přístup funguje velice obdobně jako u klasifikace písmen. Pomocí knihovny MPH jsou extrahovány příznaky v podobě klíčových bodů. Sekvence těchto příznaků jsou poté předloženy neuronové síti skládající se z LSTM neuronů. Na jejím výstupu je hodnota pro každou třídu popisující pravděpodobnost, že vstupní sekvence spadá do dané třídy.

3.3.1 Datová sada

Na rozdíl od znaků české znakové abecedy v současné době neexistuje datová sada pro diakritiku české znakové abecedy. Z tohoto důvodu je logickým prvním krokem tvorba takové datové sady. Ta by se měla skládat z videosekvencí zachycujících znakující během prstového hláskování slov, které diakritiku obsahují. Zároveň, podobně jako u datové sady z publikace [16], by měla být vytvořená metoda validována na datech od znakujícího, který nebyl součástí trénovací množiny.

Pro účely této datové sady a následného testování byla vytvořena menší databáze vět, které byly navrženy tak, aby obsahovaly co nejvíc diakritiky. Tyto věty byly následně prstově hláskovány profesionálními českými překladateli znakového jazyka. Celý tento proces byl nahráván několika kamerami z různých úhlů, díky čemuž byl zvýšen celkový počet nahrávek. Všechny záznamy byly nahrávány s 30 snímky za sekundu a s fixním rozlišením 640x480 pixelů, kde každý pixel obsahoval tři kanály (RGB).

Z každé nahrávky byly vystříhány části s jednotlivými písmeny. Stříhání bylo prováděno manuálně. Cílem bylo, aby začátek každé sekvence byl v momentě, kdy je znak pro dané písmeno poprvé rozpoznatelný. Moment, kdy znak zmizel, byl konec sekvence. Všechny takto vystříhané nahrávky byly rozděleny do třech skupin: háčky,

čárky a bez diakritiky. Pohyb ruky během znakování kroužku na „u“ byl nerozpoznatelný od háčku. Navíc významově je „ů“ stejné jako „ú“ a v drtivé většině případů je to pro koncového čtenáře irelevantní.

Data byla rozdělena do množin pro jednotlivé fáze učení modelu. Validační část dat byla vytvořena tak, aby se skládala z dat od znakujícího, který není součástí ani trénovací, ani testovací části. Celkově se datová sada skládala z 1291 trénovacích a 156 validačních a testovacích sekvencí. Trénovací sada obsahovala 408 vzorků pro čárku, 436 pro háček a 447 pro třídu bez diakritiky.

3.3.2 Extrakce příznaků

Podobně jako u klasifikace písmen je i u diakritiky použit nástroj MediaPipe Hands. Díky tomu, že klasifikace diakritiky musí být použita v kombinaci s klasifikací písmen, nezpůsobuje využití MPH v tomto případě žádnou výpočetní zátěž navíc. V prvním kroku byly z veškerých nahrávek vyextrahovány souřadnice klíčových bodů a přešlo se k jejich zpracování.

Tím, že prsty nemění během znakování diakritiky svou relativní pozici vůči sobě, je použití všech 21 klíčových bodů redundantní. Z tohoto důvodu byla využita pouze souřadnice zápěstí, která je zobrazena na obrázku 3.2 červeně, a která byla při klasifikaci písmen použita jako počátek souřadnicového systému. Z pohledu dimenzí dává smysl použít jen horizontální a vertikální, které MPH vrací ve formě absolutních hodnot s počátkem v levém horním rohu obrazu v intervalu od nuly do jedné. Souřadnice hloubky je pro klíčový bod zápěstí vždy nula, a tudíž její použití nedává smysl.

Souřadnice zápěstí byly transformovány tak, aby každý krok vyjadřoval změnu pozice. Díky tomuto kroku nezáleží na pozici zápěstí v obraze. Změna byla vypočítána jako rozdíl mezi souřadnicí zápěstí v současném kroku a souřadnicí zápěstí v kroku předešlém. První pár souřadnic byl vždy nastaven jako nulový. Následující rovnice popisují výpočet změny pozice zápěstí.

$$w_{xi} = \begin{cases} (x_{i+1} - x_i), & \text{pro } i > 0 \\ 0, & \text{pro } i = 0 \end{cases} \tag{3.1}$$

$$w_{yi} = \begin{cases} (y_{i+1} - y_i), & \text{pro } i > 0 \\ 0, & \text{pro } i = 0 \end{cases} \tag{3.2}$$

kde w_{xi} a w_{yi} popisují změnu souřadnice zápěstí pro horizontální x a vertikální y osu. Index snímků je označen jako i a n značí celkový počet snímků v dané sekvenci.

Během sběru dat bylo zjištěno, že směr, ve kterém znakující znakuje háček (zprava do leva, nebo zleva do prava), se může pro různé znakující lišit. Kvůli tomu byly všechny sekvence zahrnuty v trénování dvakrát – jednou se svým původním směrem znakování a podruhé s opačným směrem znakování. Opačný směr byl uměle vytvořen vynásobením horizontální souřadnice mínus jednou. Tímto způsobem byla datová sada uměle augmentována.

Všechny sekvence byly uměle prodlouženy maskovací hodnotou tak, aby měly stejnou délku. Tato hodnota je poté nastavena v maskovací vrstvě, která je vložena na začátek topologie. Vrstva se stará o to, aby veškeré kroky sekvence, kde všechny vstupy mají maskovací hodnotu, byly přeskočeny. Jedná se o standardní krok u knihovny

TensorFlow, který pomáhá optimalizovat tréninkový proces. Díky tomuto kroku lze zpracovávat různě dlouhé sekvence. Maskovací hodnota byla nastavena na -10 – mimo rozsah běžných vstupních hodnot. Výstupy extrakce příznaků nebyly nijak dále normalizovány ani standardizovány, neboť možný rozsah příznaků se pohyboval mezi mínus jednou a jednou, což je efektivní rozsah aktivační funkce hyperbolický tangens.

Výstupem extrakce příznaků byly stejně dlouhé sekvence dvou čísel vyjadřujících změnu pozice zápěstí od předchozího snímku.

3.3.3 Klasifikace

Pro klasifikaci byla využita rekurentní neuronová síť koncipovaná z LSTM neuronů. Architektura LSTM neuronu byla navržena tak, aby minimalizovala vliv mizejícího nebo explodujícího gradientu, a díky tomu je síť složená z těchto neuronů schopná si zapamatovat dlouhodobé závislosti. Až donedávna se jednalo o primární metodu využívanou pro strojový překlad, a tudíž její využití pro klasifikaci diakritiky bylo nasnadě.

Navržený přístup uvažoval s klasifikací, která má na vstupu sekvenci dvou hodnot vyjadřujících změnu pozice zápěstí. Na výstupu klasifikace jsou tři klasifikační třídy: háček, čárka a bez diakritiky. Z pohledu rozměrů vstupů a výstupů se jedná o poměrně přímočarou úlohu, jedinou komplikací je to, že se jedná o sekvence mající variabilní délku.

V úvodní fázi bylo vyzkoušeno mnoho různých konfigurací a topologií, které se lišily především v počtu a velikosti skrytých LSTM vrstev. Detailní parametry topologie, která vykazovala nejstabilnější chování, lze vidět v tabulce 3.4. Součástí tabulky je i počet trénovatelných parametrů jednotlivých vrstev. Z této informace lze jasně vyčíst, že LSTM vrstvy jsou velice komplexní.

Vrstva	**Velikost**	**Akt. funkce**	**Výstup**	**Počet parametrů**
Maskovací	-	-	sekvence	-
LSTM	32	*tanh*	sekvence	4480
Dropout	10 %	-	sekvence	-
LSTM	16	*tanh*	sekvence	3136
Dropout	10 %	-	sekvence	-
LSTM	8	*tanh*	číslo	800
Plně propojená	3	*softmax*	číslo	27

Tabulka 3.4: Topologie a vlastnosti zvolené neuronové sítě pro klasifikaci diakritiky.

Pro implementaci sítě byla opět využita knihovna TensorFlow a programovací jazyk Python. Trénink byl realizován pomocí stochastického algoritmu ADAM a ztrátovou funkci představovala opět křížová entropie, respektive její varianta sparse categorical crossentropy. Velikost jednoho batche byla nastavena na 64 sekvencí.

Samotný trénink vykazoval rychlý růst přesnosti v prvních 50 epochách. Hodnota ztrátové funkce klesala i po této epoše, což signalizuje, že síti se zvyšovala „jistota“ s jakou klasifikovala tréninkové vstupy. Zhruba po 100 epochách se obě metriky zasaturovaly a nedocházelo k dalšímu zlepšení. Celý proces tréninku trval okolo 15 minut, nicméně je nutno podotknout, že se jedná pouze o čas klasifikace, nikoliv extrakce příznaků pomocí MPH.

3.3.4 Výsledky

Vyhodnocení výsledků klasifikace diakritiky české znakové abecedy probíhalo ve dvou částech. V první části byla vyhodnocena přesnost klasifikace na validační části datové sady. Druhá část testovala využití společně s klasifikací písmen české znakové abecedy na nahrávkách vět znakovaných hláskovaných pomocí prstové abecedy.

Výsledky validační části

Natrénovaná neuronová síť byla validována na validační části datové sady. Jak již bylo zmíněno, ta se skládala pouze z nahrávek znakujícího, který nebyl zahrnut v trénovací množině. Tímto byla validována schopnost sítě generalizovat. Celková přesnost na validačních datech dosahovala 87,82 % a lze ji dále zkoumat na matici zmatení zobrazené v tabulce 3.5. Řádky reprezentují třídu, do které daný vzorek skutečně spadá. Naproti tomu sloupce představují třídu, která byla predikovaná modelem. Tímto způsobem lze ve výsledcích hledat nejrůznější vzorce chování a současně lze tyto informace využít i v případném dalším zpracování.

	Třída	háček	čárka	bez diakritiky
Ref. hodnota	háček	52	1	0
	čárka	2	43	6
	bez diakritiky	3	7	42
		Predikce		

Tabulka 3.5: Matice zmatení pro validační část datové sady.

Výsledky nahraných vět

Druhá část vyhodnocení výsledků spočívala v otestování na nahrávkách hláskovaných vět. K detekci sekvencí, ve kterých natrénovaná síť měla hledat danou diakritiku, byla využita neuronová síť představená v kapitole 3.1. Na výstupy této klasifikace byl aplikován klouzavý průměr, podobně jako v kapitole 3.2. Jakmile skončila sekvence, kdy bylo detekováno stále stejné písmeno, byla tato sekvence vložena do klasifikátoru diakritiky.

Výsledný typ diakritiky byl určen dle následujících pravidel:

- Klasifikace byla považována za úspěšnou v případě, že výstup neuronové sítě pro některou klasifikační třídu překročil práh 0,75. V případě, že tento práh překročen nebyl, třída „bez diakritiky“ byla použita jako výstup.
- Byla vytvořena vyhledávací tabulka mapující jednotlivá písmena na možný typ diakritiky. Jestliže výstup klasifikace pro diakritiku predikovaného písmena neexistoval ve vyhledávací tabulce, třída „bez diakritiky“ byla použita jako výstup.

Tímto způsobem byly predikovány celé věty. Opět byla využita data, která nebyla součástí trénovací množiny. Výsledky zobrazené v tabulce 3.6 neobsahovaly mezery, ty byly manuálně přidány, aby byla zlepšena čitelnost. Každá ze čtyř vět obsahuje

1	Reference	KŮŇ A HŘÍBĚ CVÁLAJÍ
	Predikce 1	KŮŇ A HŘÍBĚ CVALAJÍ
	Predikce 2	KŮŇ ASA HŘÍBĚ CVALAJÍ
2	Reference	ŠERMÍŘ ŤAL SOUPEŘŮV ÚD
	Predikce 1	ŠERMIŘ ŤAL SOUPEŘŮV ŮD[3]
	Predikce 2	ŠERMIŘ ŤAL SOÚPEŘŮV ŮD
3	Reference	ŘÍČKA MĚLA MĚLKÝ BŘEH
	Predikce 1	ŘIČKA MĚLA MĚLKY BŘEH
	Predikce 2	ŘIČKA MĚLA MĚLKY BŘEH
4	Reference	DŘEVĚNÁ KÁĎ PLNÁ HŘÍBKŮ
	Predikce 1	DŘEVĚNÁ KÁD PLNÁ HŘIBKŮ
	Predikce 2	DŘEVĚNA KAĎ PLNA HŘIBKŮ

Tabulka 3.6: Testování klasifikace diakritiky na nahrávkách vět.

referenční text – to, co mělo být znakováno, a dvě predikce. Každá predikce byla separátně znakovaná a nejedná se o stejnou instanci znakování z více kamer. Červeně vyznačená písmena signalizují nesprávnou predikci a oranžově vyznačená písmena ukazují predikci, při které byl predikován háček místo čárky.

Právě jev, že klasifikátor predikoval třídu čárka jako třídu háček, nastal v 62,5 % chyb a byl dále zkoumán. Bylo odhaleno, že znakující má často tendenci držet ruku v daném znaku i poté, co je dokončen pohyb pro diakritiku typu čárka. Tento efekt byl navíc amplifikován použitím klouzavého průměru. Nicméně jediné písmeno, které může mít jak čárku, tak háček, je písmeno „E“. Tato skutečnost činí tento neduh řešitelný při dalším zpracování.

Správnost diakritiky výrazně závisí na přesnosti a časování vstupní sekvence a tím pádem na přesnosti klasifikace abecedy. Přes 85 % chyb zobrazených v tabulce 3.6 lze klasifikovat jako chybu druhého druhu, to znamená, že diakritika byla přítomna, ale klasifikátor ji neodhalil. Obecně lze tento druh chyby považovat za přijatelnější, neboť výsledný text je více čitelný než v případě, že by klasifikátor predikoval diakritiku tam, kde by neměla být.

Toto vyhodnocení validovalo použitelnost a funkčnost klasifikace diakritiky v reálném čase. Oba cíle byly verifikovány a považovány za splněné. Výsledné řešení je schopno generalizovat a klasifikovat diakritiky znakované v obou směrech. Zároveň by tento přístup šel poměrně jednoduše rozšířit i pro levoruké znakující. Důležitým poznatkem z této části práce je to, že nástroj MediaPipe může být ve spojení s LSTM sítěmi využit pro klasifikaci sekvencí obrazů. Tento přístup má potenciál pro utilizaci na složitějších úlohách, jako je klasifikace gest znakového jazyka.

[3]Písmeno „Ů“ ve slově ŮD je správně, neboť znakující chyboval a znakoval kroužek místo čárky.

4 Rozpoznání a překlad českého znakového jazyka

4.1 Datová sada českého znakového jazyka

Výstupy prvního cíle ukázaly, že extrakce příznaků v podobě klíčových bodů vykazuje potenciál. Nicméně ještě předtím, než mohl být tento přístup aplikován na kontinuální překlad, bylo potřeba vytvořit vhodnou datovou sadu pro český znakový jazyk. Tato kapitola se věnuje tvorbě takové datové sady. Ta je založena na poznatcích z aktuálních publikací v této oblasti a výsledná data by měla splňovat určité požadavky:

- Omezená slovní zásoba, nejlépe zaměřená na jedno téma. Inspirací je datová sada RWTH-PHOENIX-Weather2014T [18] zabývající se předpovědí počasí.
- Vynechání mezi-překladu na glosy. Výstupy publikací posledních let dokazují, že sofistikované klasifikátory zvládají úlohu i bez tohoto kroku [21, 22].

Tyto požadavky dovolují, aby byla velká část procesu automatizovaná. Zároveň vynechání reprezentace glos umožňuje tvorbu datové sady znakového jazyka bez jeho znalosti. Výstupem tohoto kroku není jen samotná datová sada, ale i proces její přípravy. Ten je díky automatizaci relativně rychlý a umožňuje v budoucnu poměrně jednoduchým způsobem rozšířit datovou sadu představenou v této kapitole.

4.1.1 Sběr dat

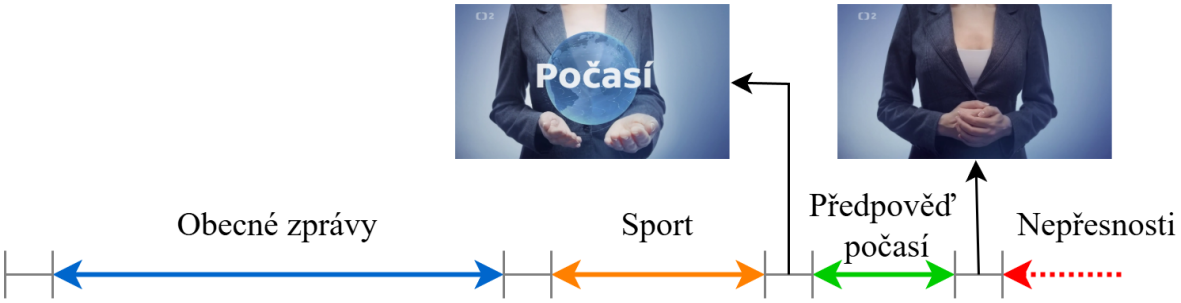
Součástí každodenního vysílání České televize je krátký segment s názvem Zprávy v českém znakovém jazyce. Díky archivu, který je dostupný na *www.ceskatelevize.cz/ivysilani*, je veřejně dostupných posledních cca 15 let tohoto pořadu. S pomocí softwaru třetí strany bylo sesbíráno 4 713 nahrávek datovaných od ledna 2009 až po prosinec 2023.

Obrázek 4.1: Příklady grafických přechodů na a ze segmentu o počasí.

Stažené nahrávky neobsahují transkript ani titulky, a tudíž není dostupný žádný psaný zdroj toho, co bylo v daném dni znakováno. Kvůli této skutečnosti bylo nutné pro dodržení určité úrovně automatizace využít automatické rozpoznání řeči. Další výzvou

pro automatizaci byl fakt, že grafické přechody a obecně grafika pořadu se během let několikrát změnily. Příklady některých grafických přechodů mezi jednotlivými segmenty jsou vyobrazeny na obrázku 4.1.

Standardně byla délka jednotlivých nahrávek okolo 10 minut, nicméně objevují se i výjimky mající 3 minuty. Každá nahrávka se skládá z tří segmentů – obecné zprávy, sport a předpověď počasí. Typická časová osa těchto segmentů je zobrazena na obrázku 4.2. Mnoho nahrávek obsahuje nepřesný střih, kvůli čemuž jsou delší a obsahují upoutávky na jiné pořady, či dokonce začátek pořadu po zprávách. Ačkoliv nás v kontextu této práce zajímal především segment předpovědi počasí, metoda automatizace přípravy datové sady by měla být použitelná obecně i na ostatních segmentech.



Obrázek 4.2: Typický časový průběh pořadu Zprávy ve znakovém jazyce.

Kvalita jednotlivých nahrávek se liší převážně dle data vysílání. Zatímco dřívější verze pořadu jsou dostupné v rozlišení 480p (640x480), od roku 2016 jsou všechny nahrávky ve 1080p rozlišení (1920x1080). Pro účely této práce bylo určeno maximální rozlišení 720p (1280x720). Nahrávkám s vyšším rozlišením byla snížena kvalita, zatímco nahrávky s nižším rozlišením byly ponechány beze změny. Jednalo se o kompromis mezi velikostí nahrávek a jejich kvalitou. Cílem tohoto kroku nebylo unifikovat rozlišení napříč všemi nahrávkami. Takový krok by drasticky snížil výslednou kvalitu, která je pro některé extraktory příznaků potřeba. Případné další úpravy kvality jsou ponechány na dané aplikaci.

4.1.2 Automatizovaná příprava dat

Hlavním cílem přípravy dat je extrahovat kýžený segment (v tomto případě předpověď počasí) z každé nahrávky a vygenerovat psaný transkript. Vedlejším produktem tohoto procesu je eliminace vadných vzorků. Součástí stažených dat jsou nahrávky, které vůbec neobsahují zprávy ve znakovém jazyce nebo obsahují špatnou audio stopu. Manuální kontrola tisíců nahrávek by byla extrémně časově náročná. Proces přípravy dat by měl mít na výstupu seznam neúspěšných vzorků, čímž se drasticky sníží počet nahrávek, které je potřeba manuálně kontrolovat.

Extrakce segmentů

Vzhledem k tomu, že segment předpovědí počasí je jen částí celých zpráv ve znakovém jazyce, bylo jej potřeba z nahrávky extrahovat. Pro všechny nahrávky platí, že předpověď počasí je posledním segmentem zpráv. Jednotlivé segmenty zpráv jsou odděleny grafickým přechodem. Nahrávka zpravidla končí společně s koncem předpovědi počasí, nicméně bylo

zjištěno, že u mnoha nahrávek tento střih není tak přesný a nedá se na něj spolehnout. Bylo potřeba navrhnout extrakci segmentu tak, ať je cílový segment oříznut z obou stran, na začátku i na konci. To navíc zajišťuje i možné použití pro libovolné jiné segmenty, například sport.

Za jádro celého procesu extrakce byl zvolen algoritmus měření strukturního indexu podobnosti (SSIM z angl. Structural Similarity Index Measurement) [24]. Pomocí něj je možné měřit podobnost mezi dvěma obrazy. Podobnost není v tomto případě měřena pixel po pixelu, ale porovnávají se strukturní informace celého obrazu. Tato informace se skládá ze tří aspektů: jas, kontrast a složení. Algoritmus SSIM se tímto snaží simulovat schopnost lidského zraku vnímat podobnosti mezi obrazy a je často používán v oblastech, kde se kvalita obrazů výrazně liší. Obvykle se počítá za pomocí pohyblivého okna, kde se výše zmíněné aspekty spočtou pro jednotlivé lokální regiony a poté jsou zkombinovány do jednoho čísla, tzv. skóre podobnosti, dle následující rovnice:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)\cdot(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\cdot(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.1}$$

kde μ_x a μ_y představují průměrné hodnoty jasových složek obrazů x a y, σ_x^2 a σ_y^2 označují rozptyly jasových složek, respektive kontrasty. Kovariance mezi jasovými složkami, respektive podobnost struktury je označena σ_{xy}. Konstanty C_1 a C_2 slouží k zajištění stability výpočtu v případě, že jsou hodnoty jasu nízké.

Výsledné skóre podobnosti je v rozmezí od mínus jedné do jedné, kde spodní mez indikuje přesně opačné obrazy, nula znamená, že si obrazy nejsou vůbec podobné a jedna značí shodu mezi dvěma obrazy.

Výhodou tohoto přístupu je možnost identifikovat snímky bez textu s relativně nízkými výpočtovými nároky. Navíc výstup lze transformovat na procentuální podobnost, kterou lze využít v dalším zpracování a optimalizacích. Nevýhodou je potřeba startovacího a koncového snímku kýženého segmentu, které představují nutný vstup uživatele. Tyto snímky musí mít stejné rozlišení jako snímky nahrávky, ve které je segment hledán.

Právě možnost transformovat výsledky SSIM na procentuální podobnost byla využita pro optimalizaci extrakce segmentů předpovědi počasí. Místo toho, aby byla kontrolována podobnost každého snímku vůči snímku referenčnímu, je využit fakt, že přechody mezi snímky trvají několik sekund, a že předpověď je posledním segmentem zpráv.

Algoritmus optimalizace začíná hledáním počátečního snímku v poslední minutě celé nahrávky. Snímek je považován za nalezený v případě, že výsledná podobnost překročí 85 %. Jakmile je počáteční snímek nalezen, začne hledání koncového snímku a je přeskočeno 10 sekund nahrávky. Tento skok zajišťuje, že se dostaneme z přechodu a zároveň by nás kratší segmenty než 10 sekund ani nezajímaly. Pakliže je výsledek podobnosti nižší než 70 %, algoritmus skočí o půl sekundy dopředu. Ve zbylých případech, tedy pro podobnost mezi 70 a 85 %, se s největší pravděpodobností blížíme cílovému obrazu, a tudíž algoritmus skočí pouze o čtvrt sekundy.

Pokud nejsou nalezeny oba snímky, algoritmus se restartuje s tím, že oblast hledání je rozšířena o jednu minutu. Tento proces pokračuje, dokud nejsou nalezeny oba snímky nebo nebyla prohledána celá nahrávka.

Anotace

Cílem druhé části automatizované přípravy dat bylo anotovat extrahované segmenty. Manuální anotace téměř 5 000 nahrávek, kde každá nahrávka má zhruba 40-sekundovou stopáž, by byla velice časově náročná. Navíc u takto repetitivního úkolu je velký prostor pro lidské chyby. Z těchto důvodů byla snaha tuto část automatizovat.

Kvůli tomu, že extrahované nahrávky neobsahovaly titulky, jedinou cestou, jak generovat text, bylo automatické rozpoznání řeči. Pro tento účel byl využit UWebASR [25], webový engine pro automatické rozpoznání české a slovenské mluvené řeči. Kromě webového rozhraní poskytuje nástroj i HTTP programovatelné rozhraní s několika příklady použití. Vstupem do nástroje je audio soubor, výstupem je mnoho formátů od čistého textu až po WebTT (Web Video Text Tracks) formát – transkript s časovými značkami. Textové výstupy zahrnují všechny náležitosti psaného textu, jakými jsou interpunkce, diakritika atd.

Celý proces anotace se skládal z několika kroků. V prvním kroku byla ze segmentů extrahována audio stopa za pomoci nástroje FFmpeg. Tyto audio stopy byly poté vstupem do nástroje UWebASR, který generoval jejich transkript. Na závěr byly generované texty zpracovány následujícím způsobem:

- Byla odstraněna veškerá interpunkce, kromě teček na koncích vět. Ty byly nahrazeny speciálním znakem.
- Věty byly rozděleny dle mezer na jednotlivá slova.
- Na začátek a konec sekvencí byly přidány speciální znaky značící start a konec sekvence.

Tímto zpracováním byly z textů vytvořeny sekvence tokenů. Token představuje jednotku textu, v našem případě slovo[1]. Sekvence tokenů byly exportovány ve formátu CSV.

4.1.3 Výsledky

Vyhodnocení automatizované přípravy dat

V rámci výsledku byla v první části vyhodnocena automatizovaná příprava datové sady. Část extrakce segmentu byla úspěšně otestována. Pro očekávaný vstup – zprávy ve znakovém jazyce se segmentem předpovědí počasí v poslední minutě nahrávky – trvala extrakce okolo pěti sekund. Algoritmus našel cílový segment v 99 % případů, v opačném případě byl uložen název souboru pro pozdější manuální kontrolu. Ta ukázala, že část neúspěšných extrakcí byla způsobena příliš vysokým prahem pro podobnost mezi obrazy a jeho snížení stačilo pro úspěšnou extrakci. Zbylé neúspěšné nahrávky byly skutečně vadné, jednalo se většinou o úplně jiné pořady, případně poškozené video. Z pozdějšího dalšího zpracování datové sady je zřejmé, že všechny vadné nahrávky byly eliminovány během této fáze. Výstupem této části bylo 4 706 segmentů předpovědi počasí z původních 4 713.

Druhá část, která měla za úkol anotovat nahrávky, byla uskutečněna za pomoci nástroje UWebASR. Proces anotace jedné nahrávky trval přibližně 30 sekund. Tato část

[1] Specializované jazykové modely využívají i tokeny v podobě části slov (kořen, předpona, přípona).

byla závislá na softwaru třetí strany, a tudíž nebylo možné provést další optimalizace. Výsledný transkript je velice přesný a jeho celková kvalita je nad očekávání. Všechny transkripty byly manuálně zkontrolovány a opraveny. Audio stopa 7 nahrávek byla vadná a tyto nahrávky byly odstraněny. Na výstupu této části zbývalo v datové sadě 4 699 nahrávek. Kromě vadných nahrávek byly v textech přítomny i přeřeknutí, či zaseknutí mluvčích a občasné chyby automatického rozpoznání řeči. Nicméně tyto chyby byly koncentrované především ve specifických frázích a byly manuálně opraveny.

Vyhodnocení datové sady

Kromě automatizovaného procesu byla výstupem i datová sada čítající 4 699 anotovaných nahrávek předpovědi počasí. Tabulka 4.1 shrnuje podrobnější parametry datové sady.

Ačkoliv datová sada představená v publikaci [18] obsahuje okolo 8 000 nahrávek, jednotlivé nahrávky jsou mnohem kratší a obsahují pouze krátkou větu či frázi. Průměrná délka nahrávky datové sady představené v této práci je okolo 34 sekund, s nejdelší nahrávkou 91 sekund a nejkratší 23 sekund. Podrobnější analýza ukázala, že se v datové sadě nachází několik nahrávek, které jsou výrazně delší než zbytek. Nicméně tyto nahrávky byly zahrnuty do výsledné datové sady a jejich odstranění je přenecháno na případné aplikaci.

Počet vzorků	4699
Celková délka vzorků (vstup)	44 h 32 min
Průměrná délka vzorku (vstup)	34,1 s
Maximální / minimální délka vzorku (vstup)	91,2 / 23 s
Počet znakujících (muži / ženy)	8 (2 / 6)
Průměrná délka vzorku (výstup)	57,4 slov
Maximální / minimální délka vzorku (výstup)	138 / 23 slov
Velikost slovníku	1679 slov

Tabulka 4.1: Parametry datové sady českého znakového jazyka.

Nahrávky byly rozděleny na trénovací, validační a testovací sety. Validační a testovací sety byly vytvořeny tak, aby bylo zajištěno rovnoměrné rozdělení napříč roky jejich původu. Jak již bylo zmíněno, rozlišení a celková kvalita nahrávek se s průběhem let zvyšovala.

Průměrná délka výstupních sekvencí byla 57,4 slova, s nejdelší sekvencí 138 a nejkratší 23 slovy. Právě délka výstupních sekvencí činí datovou sadu unikátní, neboť sekvence jsou výrazně delší než v existujících datových sadách. To umožňuje potenciální posun současného stavu překladu znakového jazyka.

Díky tomu, že původ nahrávek je z každodenního vysílání předpovědi počasí, jsou si výstupní sekvence velice podobné. Obecně se každá nahrávka skládá z následujících částí:

1. Krátké shrnutí počasí následující den (kromě pátečních zpráv, které shrnují celý víkend), například *„zítra bude jasno až polojasno, odpoledne místy přeháňky“*.
2. Následují podrobnější informaci o rozdílech v počasí v průběhu dne nebo napříč republikou. Tento segment je nejvíce variabilní částí celé datové sady. Může se skládat z několika vět v případě, že se jedná o výjimečnou meteorologickou situaci, nebo může být úplně vynechán.

3. V další části se předpověď počasí věnuje informováním o teplotě následující den. To se typicky se skládá ze dvou teplotních rozsahů, jeden obecný a druhý pro hory. Ojediněle jsou přidány další teplotní rozsahy při výskytu abnormálních lokálních teplotních výkyvů.
4. Rozloučení, které je velice podobné napříč datovou sadou.

Výsledná slovní zásoba datové sady je 1679 slov, což je v porovnání s publikací [18] méně (2887 slov). Skoro polovina slovní zásoby je přítomna dvakrát nebo jednou. Nicméně 227 slov má více než 100 opakování. Celkově výsledná slovní zásoba splňuje vytyčené cíle, co se týče její velikosti a počtu opakování. Jedná se o menší slovní zásobu s často opakujícími se slovy v různých kontextech.

4.2 Překlad českého znakového jazyka

Posledním cílem této disertační práce je vytvoření metody, pomocí které lze český znakový jazyk překládat. Tato metoda by měla agregovat a využívat všechny doposud nabyté zkušenosti a poznatky. Ty se týkaly především extrakce příznaků pomocí klíčových bodů pózy. K tréninku a prvotnímu testování je využita datová sada představená v kapitole 4.1. Součástí této úlohy je i augmentace dat. Ta má za úkol uměle rozšířit trénovací množinu a zvětšit tak prostor, na kterém je model trénován.

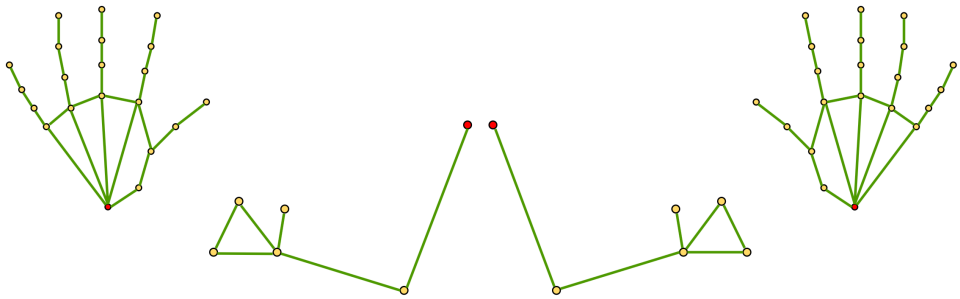
Originální i augmentované příznaky v podobě klíčových bodů jsou použity jako vstup do enkodéru modelu Transformer. Jedná se o autoregresní enkodér-dekodér model, který na základě celého vstupu postupně generuje výstup. V našem případě generuje jedno slovo za druhým, dokud není vygenerován token značící konec sekvence, nebo není dosažena maximální délka výstupní sekvence. Kontext dosavadního výstupu je udržován zpětnou vazbou, kdy předchozí výstup je současným vstupem do dekodéru.

4.2.1 Extrakce příznaků

Extrakce příznaků v této aplikaci využívala, oproti klasifikaci abecedy a diakritiky, navíc i knihovnu MediaPipe Pose (MPP) [15], která je stejně jako MPH založená na konvolučních neuronových sítích. MPP generuje 32 klíčových bodů po celém těle, nicméně znaková řeč využívá převážně vrchní část těla, tudíž byly použity pouze klíčové body z této oblasti.

Podobně jako u MPH, i MPP produkuje souřadnice klíčových bodů v absolutním souřadnicovém systému v rozmezí od nuly do jedné. Bylo nutné transformovat všechny souřadnice tak, aby jejich hodnoty byly indiferentní vůči poloze v obraze. Pro každou část těla byla zvolen její počátek souřadnicového systému. Pro prsty to byl klíčový bod zápěstí a pro paže to byla ramena. Vizualizaci souřadnic lze vidět na obrázku 4.3, kde jsou červenou barvou označeny body, které slouží jako počátek pro danou část.

Při zpracování byly vynechány souřadnice počátků jednotlivých souřadnicových systémů. Vzhledem k tomu, že se jedná o počátky, byly hodnoty souřadnic vždy rovny nule, a tudíž neměly žádnou přidanou hodnotu pro překlad. Finální podoba klíčových bodů čítala celkem 50 párů souřadnic, 20 pro každou ruku a 5 pro každou paži. Všechny snímky, u kterých MPP nebylo schopno nalézt pózu znakujícího, byly vynechány. Kromě oříznutí nepřesnosti na začátku a na konci byly tímto krokem zredukovány rozměry vstupu a tím zkrácena celková délka potřebného kontextu.



Obrázek 4.3: Vizualizace klíčových bodů a počátků jejich souřadného systému (červeně).

V rámci přípravy dat byla prozkoumána i délka jednotlivých vzorků. Ta hraje hlavní roli, neboť model Transformer očekává celou sekvenci, a tudíž jeho rozměry rostou s maximální délkou vzorku. Tento průzkum odhalil 9 nahrávek, které byly výrazně delší než zbytek, a jejich vynechání by zkrátilo potřebný kontext o 40 %. Z tohoto důvodu byly tyto nahrávky z datové sady pro účely této úlohy odstraněny.

Dalším krokem byla normalizace dat. Ta měla za úkol normalizovat hodnoty jednotlivých příznaků tak, aby se maximalizovalo využití funkčního rozsahu aktivačních funkcí, a tím se ulehčilo trénování. Pro normalizaci byla zvolena normalizace rozpětím neboli min-max normalizace. Vzhledem k úspěšnému využití aktivační funkce hyperbolický tangens v předešlých úlohách, byl uvažován rozsah oboru hodnot této funkce, tedy všechny příznaky byly normalizovány do rozsahu $(-1, 1)$. Tímto způsobem byly normalizovány všechny sekvence.

Následovala augmentace dat, která měla za úkol vygenerovat z existujících vzorků vzorky nové a tím zajistit dostatek dat pro trénink. K tomuto účelu byl vyvinut proces, který na základě pravděpodobnosti postupně aplikuje nejrůznější operace na originálních datech. Mezi tyto operace bylo zařazeno přidání šumu, škálování, rotace a zkosení. Důležitým faktorem augmentace je náhodnost, která v navrženém procesu byla přítomna na všech úrovních.

4.2.2 Překlad

Výstupem výše uvedené extrakce příznaků byly sekvence 100 souřadnic klíčových bodů. Každá sekvence těchto bodů měla svou slovní reprezentaci, na kterou měl model sekvencí klíčových bodů namapovat. Z charakteru úlohy lze hovořit o mapování sekvencí s délkou n na sekvence o délce m, kde n je mnohonásobně vyšší než m. Zatímco nejdelší vstupní sekvence měla délku 2 162 „snímků“, nejdelší výstupní sekvence pouze 139 slov. To poukazuje na komplexnost, kterou musí překladový model pokrýt.

Veškerá implementace proběhla v knihovně TensorFlow v jazyce Python. Ten v současné době nepodporuje model Transformer, neboť se jedná o poměrně novou architekturu. Proto bylo přistoupeno k vlastní implementaci tohoto modelu. Architektura modelu byla implementována po vzoru publikace [12], která model poprvé publikovala.

Rozdílem oproti standardnímu modelu Transformer byla embedding vrstva enkodéru. V případě klasického překladu je prvním krokem embedding vrstva, která má za úkol vytvořit z jednotlivých slov (tokenů) jeho vektorovou reprezentaci. Kvůli tomu, že v našem případě už vstup ve vektorové podobě je, byla místo toho použita plně propojená vrstva,

která má za úkol rozšířit vstupní vektor a najít mezi jednotlivými souřadnicemi nejrůznější vztahy.

Výstupem Transformeru je pro každé slovo ze slovní zásoby jeho pravděpodobnost, že je dalším výstupem. Jako výstup bylo vždy vybráno slovo s nejvyšší pravděpodobností (angl. greedy decoding). Jedná se o způsob vyplývající z charakteru úlohy – každý vstup má právě jeden správný výstup. Nicméně u jiných úloh, například generativních, se často využívají sofistikovanější algoritmy.

Parametry

Jednotlivé parametry topologie jsou shrnuty v tabulce 4.2. Většina parametrů sítě byla určena experimentálně a dle výpočetních limitů použité instrumentace.

Dropout	20 %
Počet enkodérů/dekodérů	4
Počet hlav	4
Velikost plně-propojených vrstev	1024
Rozměr embeddingu	1024

Tabulka 4.2: Parametry navrhnutého modelu Transformer.

Jako ztrátová funkce byla využita varianta křížové entropie, sparse categorical crossentropy. Pro trénink síť využívá stochastický optimalizační algoritmus ADAM, navíc s upravenou rychlostí učení (angl. learning rate). Ta byla nastavena tak, aby v prvních krocích rychleji měnila váhy.

Kvůli komplexnosti a velikosti modelu rostly i výpočetní nároky, které se ukázaly jako limitující. Pro trénink byl využit počítač s procesorem AMD Ryzen s 32 GB paměti RAM na 64bitovém operačním systému Windows 10. Výpočty pak obstarávaly dvě grafické karty GeForce RTX 3060, každá s 12 GB GDDR6 RAM paměti. Jak bylo zjištěno během samotného tréninkového procesu, velikost batche mohla být nastavena maximálně na 4, neboť při vyšší velikosti docházely stroji zdroje.

4.2.3 Trénink

Prvotní testy ukázaly, že model má na jakýkoliv vstup jednotnou predikci. V průběhu prvních několika epoch se počet různých predikcí zvýšil, nicméně jednalo se o vyšší jednotky variací. Z tohoto chování vyplynulo, že model není schopný nalézt potřebné informace ve vstupní sekvenci klíčových bodů pózy. Důvodů tohoto chování mohlo být několik, z nichž nejpravděpodobnější je příliš velká délka vstupních sekvencí. Z tohoto důvodu bylo přistoupeno k jejich zkrácení.

Originální vstupní nahrávky měly 25 snímků za sekundu, což způsobilo, že nejdelší nahrávka obsahovala 2 162 snímků. Překladový model musí být schopný pokrýt takto dlouhý kontext. Avšak z pohledu znakového jazyka a jeho rozpoznání by nemělo být potřeba tak vysoký počet snímků, a proto prvním způsobem, jak zkrátit vstupní data, bylo podvzorkování nahrávek. To bylo aplikováno na všechna vstupní data, včetně augmentovaných, a byly využity různé vzorkovací frekvence. Cílem je nalézt optimum mezi zkrácením vstupního kontextu a množstvím ztracených informací. Použité počty snímků za sekundu společně s označením modelu, které je použito dále v práci, lze vidět v tabulce 4.3.

Označení modelu	D1	D2	D3	D5	D10
Snímků za sekundu	25	12,5	8,3	5	2,5
Maximální délka sekvence	2162	1081	721	433	217

Tabulka 4.3: Testované konfigurace podvzorkování vstupních nahrávek.

4.2.4 Výsledky

Trénink s originálními nahrávkami vykazoval vesměs vždy stejný průběh. Model se během prvních několika epoch naučil na jakýkoliv vstup odpovídat konečným počtem variací. Po těchto epochách většinou nastal moment, kdy model nebyl schopný vylepšovat svou přesnost generalizováním vstupů a začalo docházet k přetrénování. To se projevovalo rostoucí přesností na tréninkových datech, zatímco na validačních docházelo k opakování slov a výstupy přestaly pozbývat jakýkoliv smysl.

Pro podvzorkovaná data byl průběh prvních pár epoch podobný, avšak s větší variací výstupů. Nicméně pro všechna podvzorkování nastala fáze, kdy se model skutečně snažil o překlad a na každý vstup byl vygenerován originální výstup. Trvání této fáze bylo s rostoucím podvzorkováním delší. Čím menší počet snímků za sekundu, tím déle model vydržel v této stabilní fázi. Následovalo přetrénování, které se projevovalo obdobně jako v případě originálních nahrávek.

Kvantitativní výsledky

V rámci vyhodnocení výsledků byla provedena kvantitativní evaluace pomocí metrik BLEU (Bilingual Evaluation Understudy) [26] a ROUGE (Recall-Oriented Understudy of Gisting Evaluation) [27]. Volba metrik vycházela z použitých metrik v publikacích [18, 21, 22]. Kvantitativní výsledky slouží především k prvotnímu porovnání jednotlivých modelů. Obě metriky byly spočteny jak pro validační, tak pro testovací datovou sadu. Validační set je část dat, která sice nebyla součástí trénování, nicméně během trénování sloužila k ladění hyperparametrů modelů a jejich vyhodnocení. Naproti tomu testovací set je použit až pro natrénované modely, a tudíž jeho výsledky jsou zcela nezávislé na tréninku. Pro ideálně vyváženou datovou sadu by výsledné hodnoty pro obě tyto části měly být velice podobné.

Podrobné vyhodnocení metrik pro validační a testovací set lze vidět v tabulce 4.4. Z výsledků je patrné, že kvantitativně není mezi těmito sety příliš velký rozdíl. Hodnoty a jednotlivé trendy napříč daty jsou velice obdobné. Metrika BLEU se soustředí na přesnost n-gramů. Její hodnoty jsou pro modely D2 a D3 (12,5 a 8,3 snímků za sekundu) velice podobné, pro vyšší podvzorkování začínají pozvolna klesat. Kvůli délce překládaných sekvencí by hlavním zaměřením mělo být BLEU-4, které reflektuje plynulost a přesnost syntaxe překladu a je nejblíže kvalitativnímu lidskému hodnocení. Zároveň je velmi citlivé na chyby, kde i malá odchylka překladu může vést k nízkému skóre. Nejvyšší BLEU-4 skóre 18,4 dosáhla konfigurace D3 s 8,3 snímky za sekundu.

Druhou metrikou byla ROUGE, která se více než na přesnost zaměřuje na zpětné pokrytí, což znamená, kolik relevantního obsahu bylo zahrnuto v generovaném textu oproti referenci. Varianta ROUGE-L se soustředí na nejdelší společné posloupnosti, což umožňuje měřit zachování větné struktury. Pro tuto metriku je trend výsledných hodnot relativně stejný jako v případě BLEU-4. Nejvyšší hodnota ROUGE-L je 48,2 pro model D3, ostatní konfigurace dosahovaly nižších hodnot. Výsledné hodnoty metrik naznačují, že model D3

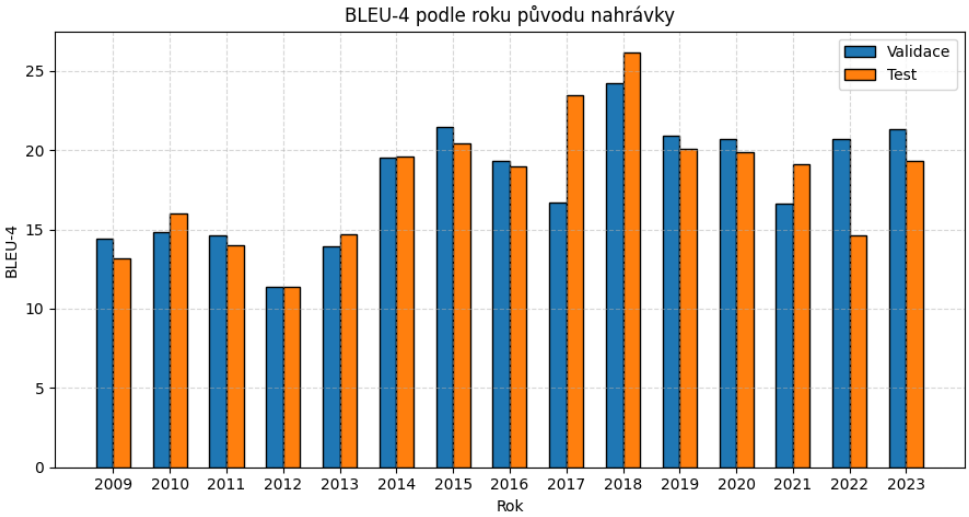
s podvzorkováním na 8,3 snímků za sekundu je nejvhodnější kompromis mezi zkrácením vstupního kontextu a množstvím ztracených informací.

		Snímků za sekundu			
		12,5 (D2)	8,3 (D3)	5 (D5)	2,5 (D10)
Validační set	BLEU-1	**42,3**	41,1	38,4	39,4
	BLEU-2	**31,6**	31,2	28,3	29,4
	BLEU-3	**24,1**	**24,1**	21,4	23,9
	BLEU-4	18,1	**18,4**	15,6	16,6
	ROUGE-L	47,0	**48,4**	45,2	47,6
Testovací set	BLEU-1	**41,5**	41,3	38,5	38,7
	BLEU-2	30,7	**31,3**	28,3	28,4
	BLEU-3	23,3	**23,4**	20,3	21,5
	BLEU-4	17,4	**18,4**	16,0	15,6
	ROUGE-L	46,3	**48,2**	45,6	46,2

Tabulka 4.4: Vyhodnocení metrik BLEU a ROUGE pro různá podvzorkování.

Vzhledem k délkám jednotlivých sekvencí byla z pohledu kvantitativních výsledků za nejlepší prohlášena varianta s trojnásobným podvzorkováním, resp. s 8,3 snímky za sekundu. Ta vykazovala nejlepší výsledky pro metriku BLEU-4 i ROUGE-L jak pro validační, tak pro testovací datovou sadu a byla využita dále v práci.

Pro nabytí představy o významu hodnot jednotlivých metrik lze použít výsledky z publikace [22]. Ta dosahovala na německé datové sadě předpovědí počasí PHOENIX14T BLEU-4 skóre 22,52 a ROUGE-L skóre 48,90. Na čínské datové sadě CSLDaily to byly hodnoty 15,40 pro BLEU-4 a 42,36 pro ROUGE-L. Jedná se o podobné hodnoty jako v případě této práce, která se věnuje delším sekvencím.



Obrázek 4.4: Skóre BLEU-4 podle roku původu nahrávky.

V rámci kvantitativního vyhodnocení byly zkoumány i hodnoty metrik v závislosti na roku původu nahrávky. Tvorba datové sady ukázala, že nahrávky z dřívějších let mají horší kvalitu. Během přípravy datové sady byly všechny nahrávky škálovány na rozlišení 1280x720, nicméně dřívější nahrávky byly v rozlišení 1024x576. Přelom v rozlišení byl v průběhu roku 2014. Zároveň můžou být přítomny i jiné rozdíly v kvalitě, které nejsou tak jednoduše kvantifikovatelné. Byla vyhodnocena pouze metrika BLEU-4, neboť se jednalo o jednu z nejrelevantnějších metrik. Měření bylo provedeno s modelem D3.

Výsledky jsou zobrazeny v grafu na obrázku 4.4 a lze vidět, že jak validační, tak testovací část datové sady kopíruje poměrně podobný trend. Do roku 2013 včetně bylo průměrné skóre BLEU-4 okolo 13, od roku 2014 okolo 20. Najdou se i výjimky, jakou je například rok 2022, kdy testovací část dosáhla pouze skóre 14,6. Naproti tomu v roce 2018 obě části dosahovaly nejvyšších hodnot, validační část 24,2 a testovací část 26,2.

Kvalitativní výsledky

Ačkoliv kvantitativní výsledky poskytují jednoduché srovnání jednotlivých modelů, úlohy strojového překladu a obecně přirozeného zpracování jazyka doposud pořád spoléhají na manuální zhodnocení člověkem. Při reprezentaci každé predikce pouze jedním referenčním textem není žádná metrika definitivně schopná zhodnotit kvalitu překladu, neboť každou referenci lze přeložit nepřeberným množstvím kombinací. V kontextu předpovědi počasí takovými příklady může být použití „šestadvacet“ místo „dvacet šest“ nebo „občasný déšť“ místo „přeháňky“. Zatímco kvalitativně se jedná o přesné překlady, kvantitativní zhodnocení by bylo nulové. Z tohoto důvodu kvantitativní vyhodnocení slouží jen jako jeden z ukazatelů a je nutné celkový překlad zhodnotit i kvalitativně.

#		**Text**
1	R	Zítra bude jasno nebo skoro jasno, ráno a dopoledne hlavně v Čechách místy mlhy i mrznoucí nebo nízká oblačnost. Během dne od jihozápadu pozvolné přibývání frontální oblačnosti. Nejvyšší teploty deset až čtrnáct stupňů Celsia v tisíci metrech na horách kolem šesti stupňů Celsia na Šumavě až deset stupňů. Dnešní zprávy jsou u konce. Příjemný večer.
	P	Zítra bude polojasno až skoro jasno, ráno a dopoledne místy mlhy nebo nízká oblačnost. Nejvyšší teploty deset až čtrnáct stupňů v tisíci metrech na horách kolem osmi stupňů. Zprávy jsou u konce. Hezký večer.
2	R	Zítra bude oblačno až zataženo jen zpočátku na severu a severovýchodě místy sněžení. Postupně polojasno až skoro jasno zejména v Čechách. Nejvyšší teploty minus dva až plus dva stupně na horách kolem minus čtyř. Zprávy ve znakovém jazyce jsou u konce. Pěkný večer.
	P	Zítra bude oblačno až zataženo a zejména na severu a severovýchodě místy slabé sněžení. Nejvyšší teploty minus čtyři až nula v tisíci metrech na horách kolem minus dvou stupňů Celsia. Dnešní zprávy končí. Hezký večer.

3	R	Zítra bude oblačno až zataženo zpočátku místy postupně jen ojediněle slabý déšť nebo mrholení, v polohách nad sedm set metrů srážky smíšené nebo sněhové. Nejvyšší teploty pět až devět stupňů v tisíci metrech na horách bude kolem nuly. Upozornění hydrometeorologického ústavu, v noci a ráno se může místy tvořit náledí. Zprávy jsou u konce. Hezký večer.
	P	Zítra bude zataženo až oblačno místy slabý déšť nebo mrholení, nad pět set metrů srážky sněhové. Nejvyšší teploty dva až šest stupňů v tisíci metrech na horách kolem nuly. Zprávy jsou u konce. Hezký večer.
4	R	V sobotu bude zataženo až oblačno na jihu Moravy místy jinde jen ojediněle mrholení ojediněle mlhy. V neděli bude zataženo až oblačno ojediněle mlhy místy mrholení zejména na jihovýchodě, v Čechách postupně místy déšť nebo přeháňky. Nejvyšší denní teploty pět až devět stupňů Celsia. To je pro dnešek všechno. Hezký zbytek dne.
	P	V sobotu bude zataženo s nízkou oblačností nebo mlhy ojediněle mrholení. Nejvyšší denní teploty pět až osm stupňů na horách kolem tří stupňů Celsia. To byly zprávy ve znakovém jazyce. Hezký večer.
5	R	Zítra bude jasno až polojasno, při přechodně zvětšené oblačnosti se ojediněle objeví přeháňky nebo bouřky. Nejvyšší teploty sedmadvacet až jednatřicet stupňů Celsia na horách bude kolem třiadvaceti. Zprávy v českém znakovém jazyce jsou u konce. Přeji hezký večer.
	P	Zítra bude jasno nebo skoro jasno. Nejvyšší teploty dvacet sedm až třicet jedna stupňů Celsia. To je ze zpráv v českém znakovém jazyce vše. Hezký večer.

Tabulka 4.5: Příklady generovaných překladů, R – Reference, P – Překlad.

Příklady jednotlivých překladů jsou zobrazeny v tabulce 4.5. Jedná se o predikce testovací části datové sady modelem D3. Překlad se potýkal s několika fenomény, z nichž většina je v tabulce. Téměř ve všech příkladech lze vidět, že model správně sumarizuje hlavní informace, které přináší reference. Detailnější informace jsou modelem často vynechány, například v překladu #1 chybí geografická místa zmiňovaná v referenci (Čechy, Šumava) a také byla vynechána prostřední věta o přibývání oblačnosti. V některých případech model správně překládá i geografické termíny, například překlad #2, kde správně předpovídá „sněžení na severu a severovýchodě“.

Zajímavou částí jsou teploty. Ty model vždy předpovídá v jednom rozsahu pro nížiny a jednou teplotou pro hory, což je obecný tvar předpovědí teplot, který následuje většina dat. Dekodér se zpravidla snaží najít číselnou informaci o teplotě ve vstupních datech z enkodéru. Když je teplota v datech nalezena, dekodér z ní vytvoří rozsah a z toho následně složí větu. Rozmezí teplot se tedy často blíží referenčnímu, což dokazují všechny uvedené příklady. V opačném případě, kdy teplota není ve vstupních datech nalezena, ji dekodér určuje z dosavadního kontextu. Například když je jasno, či polojasno, je generickou predikcí 20 až 24 stupňů Celsia. Z toho vyplývá, že model

nepotřebuje informace z enkodéru k tomu, aby predikoval smysluplný překlad a začne využívat pro generování pouze dekodér – výstupy tak dávají gramaticky smysl, ale vůbec neberou v potaz vstupy znakové řeči.

V datech se často vyskytuje i jev, kdy je překlad správný, ale při kvantitativní evaluaci by tak vyhodnocen nebyl. Překlad #5 toto ukazuje, kdy jsou v referenci uvedeny teploty „sedmadvacet až jednatřicet“, nicméně predikováno je „dvacet sedm až třicet jedna“. Kromě teplot se tento jev ukazuje i u obecné předpovědi, viz #5 reference: „jasno až polojasno“, překlad: „jasno nebo skoro jasno“, nebo #3 reference: „oblačno až zataženo“, překlad: „zataženo až oblačno“.

Ačkoliv většina vstupních dat začíná souslovím „zítra bude“, existují případy, kdy tomu tak není. To se týká většinou pátečních zpráv, které předpovídají počasí na celý víkend. Jedná se skutečně pouze o zlomek vstupních dat a bylo očekáváno, že model nebude schopný se toto naučit. Opak byl pravdou a v některých případech, jako je překlad #4, byl model schopný přeložit začátek sekvence jinak.

Kapitola 4.1.3 vyhodnocovala datovou sadu a rozdělovala jednotlivé vzorky do čtyř částí. Z hlediska jejich překladu je lze ohodnotit následovně:

1. Krátké shrnutí počasí následujícího dne – to je přeloženo vesměs vždy správně, s drobnými odchylkami. Překladový model je schopný přeložit i geografické detaily.
2. Podrobnější informace – ty jsou zpravidla vynechány a málo kdy se jim model věnuje. Pro jejich predikci by bylo nejspíš potřeba více dat, neboť tahle část je velice variabilní.
3. Předpověď teplot – predikce generuje jeden rozsah pro nížiny a jednu teplotu pro hory. Hodnotami jsou teploty zpravidla poměrně přesné, nicméně stává se, že dekodér informaci nenajde a teplotu domýšlí z kontextu.
4. Rozloučení – to je přítomno vždy, jeho přesnost se těžko posuzuje, neboť není známo, zda se znakovaná rozloučení odlišují.

Celkově lze prohlásit, že model kvalitativně překonal očekávání. Generuje smysluplné texty, které sumarizují referenci. Obecné informace jsou zpravidla přeloženy přesně, detailnější informace jsou modelem vynechány. V případě, že model ztratí kontext vstupů enkodéru, začne generovat výstup pouze na základě doposud vygenerovaného textu. Vylepšení by mohlo přinést rozšíření datové sady, případně rozdělení vstupů na jednotlivé věty, k čemuž je nicméně potřeba znalost znakového jazyka.

5 Závěr

Tato disertační práce se věnovala vývoji metod strojového učení pro rozpoznání českého znakového jazyka. V rámci pojednání byly stanoveny obecné cíle, které práci dělí na dvě úlohy: rozpoznání české znakové abecedy a překlad gest českého znakového jazyka.

Společnou úlohou napříč všemi cíli je extrakce příznaků, která je realizována v podobě klíčových bodů pózy. Pro tyto účely byl využit nástroj MediaPipe, pomocí kterého jsou z obrazu extrahovány klíčové body prstů, rukou a paží. Díky tomuto přístupu nejsou příznaky ovlivněny prostředím a zároveň jsou nezávislé na znakovém jazyce, na rozdíl od extraktorů trénovaných přímo na dané úloze. Jedná se o inovativní přístup, kterému se dostává stále větší pozornosti.

První část řešení se věnuje rozpoznání české znakové abecedy. V rámci tohoto cíle bylo mířeno na řešení použitelné v reálných aplikacích na mobilních platformách. Pro úlohu byla využita datová sada představená v publikaci [16]. Extrahované příznaky v podobě klíčových bodů byly klasifikovány dopřednou neuronovou sítí a bylo dosaženo přesnosti 94,4 %, což bylo zlepšení oproti původní publikaci. Výsledky byly verifikovány v reálném čase na mobilní platformě s operačním systémem Android. Pro tyto účely byla vytvořena mobilní aplikace a algoritmus, který predikuje znakovaná písmena z videonahrávky. Tímto byla ověřena funkčnost řešení a její případná použitelnost v reálné aplikaci.

V rámci první části byla také řešena diakritika znakové abecedy, což je specifická věc pro českou znakovou abecedu. Ta je znakována pohybem ruky ve tvaru diakritiky. Byla vytvořena datová sada diakritiky, která čítala okolo 1 600 videí. Pro její klasifikaci byly využity rekurentní neuronové sítě LSTM. Kromě kvantitativního vyhodnocení přesnosti klasifikace, která dosahovala až 88 %, bylo uskutečněno i kvalitativní vyhodnocení na nahrávkách hláskovaných vět prstovou abecedou. Tím se validovala použitelnost a funkčnost klasifikace diakritiky v reálném čase.

Druhá část disertační práce se věnovala překladu českého znakového jazyka. Kvůli absenci vhodné datové sady pro český znakový jazyk bylo přistoupeno k jejímu vytvoření. V rámci práce byla vytvořena kontinuální datová sada, kterou tvoří nahrávky předpovědí počasí z on-line archivu České televize. Výsledná sada obsahuje přes 4 500 nahrávek o celkové stopáži přes 44 hodin se slovní zásobou 1 679 slov. Oproti existujícím datovým sadám jsou jednotlivé vzorky mnohem delší a díky tomu představují výzvu pro překladové modely. Kromě datové sady byl v tomto kroku vyvinut algoritmus pro její přípravu, který se snaží o maximální míru automatizace a usnadňuje tak její případná budoucí rozšíření.

Poslední část se věnovala vývoji modelu pro překlad českého znakového jazyka. Přístup se skládal z extrakce příznaků, která byla ověřena během předchozích kroků, a modelu Transformer, jež je aktuální architekturou používanou pro strojový překlad. Po prvotních neúspěších, které byly způsobeny příliš dlouhými vstupy ve spojení s nedostatkem výpočetních zdrojů, byly představeny modely využívající podvzorkované vstupy. V rámci kvantitativního vyhodnocení byl za nejlepší prohlášen model s trojnásobným podvzorkováním, tedy s 8,3 snímky za sekundu. Ten dosahoval BLEU-4 skóre 18,4 a ROUGE-L skóre 48,2, což jsou hodnoty srovnatelné s výsledky v publikaci [22], avšak pro mnohem delší sekvence. Současně bylo odhaleno, že kvalita vstupních nahrávek se výrazně odráží na výsledném skóre. Z kvalitativního pohledu bylo zjištěno, že výsledný model generuje smysluplné texty, které zpravidla sumarizují

reference. Ačkoliv jsou podrobnější detaily často vynechány, z výstupů je patrné, že model je schopný překládat český znakový jazyk.

Výsledkem práce jsou metody strojového učení pro rozpoznání a klasifikaci českého znakového jazyka. Jejich funkčnost a aplikovatelnost byla v rámci práce ověřena. Všechny cíle disertační práce byly splněny.

5.1 Přínosy disertační práce

Využití klíčových bodů pózy jako příznaků pro klasifikaci

Příznaky v podobě klíčových bodů pózy jsou využity napříč všemi cíli disertační práce. Výstupy této práce mohou sloužit jako důkaz, že takto extrahované příznaky jsou více než dostačující náhradou za příznaky extrahované konvolučními sítěmi trénovanými pro danou úlohu, což bývá standardní postup. Výsledky dosažené v rámci klasifikace české znakové abecedy v kapitole 3.1 překonávají v přesnosti klasicky extrahované příznaky [16]. Stejný jev lze pozorovat i při klasifikaci izolovaných gest znakového jazyka na datové sadě AUTSL [28]. Představený přístup výrazně vylepšuje přesnost oproti původní referenční metodě prezentované v publikaci [17], která se liší pouze v extrakci příznaků.

Velkou výhodou tohoto přístupu je nezávislost na prostředí. Takto extrahované příznaky skutečně reprezentují důležitý kontext pro znakový jazyk, tedy pozici rukou a prstů. Nevýhodou je závislost na nástrojích třetích stran. Nicméně díky rostoucí popularitě tohoto přístupu se zlepšuje nejen kvalita těchto nástrojů, ale i jejich optimalizace pro nejrůznější platformy.

Mobilní aplikace pro překlad české znakové abecedy

V kapitole 3.2 byla představena mobilní aplikace pro překlad české znakové abecedy. Ta ověřuje funkčnost a aplikovatelnost metody pro překlad české znakové abecedy v reálném čase. Ačkoliv se v současném stavu jedná pouze o důkaz konceptu, může sloužit jako základní stavební kámen pro složitější uplatnění, například výuku znakové abecedy či překladač.

Datová sada diakritiky české znakové abecedy

Pro účely klasifikace diakritiky české znakové abecedy byla vytvořena obsáhlá datová sada, viz kapitola 3.3. Ta sestává z nahrávek profesionálních překladatelů znakujících jednotlivá písmena, ať s diakritikou, nebo bez. Kromě toho obsahuje datová sada i nahrávky celých vět, obsahujících dostatek variability pro testování a verifikaci algoritmů pro překlad české znakové abecedy. Důkazem využitelnosti datové sady je publikace [29], která používá datovou sadu ke klasifikaci diakritiky pomocí metody optického toku.

Metoda automatizované přípravy datové sady českého znakového jazyka

Kapitola 4.1 se věnuje datové sadě českého znakového jazyka. Manuální příprava takové datové sady sestává ze střihu jednotlivých segmentů a jejich anotace, která v tomto případě znamená transkripci mluveného projevu. Jedná se o velice časově náročnou činnost, která je navíc náchylná na lidské chyby. Z těchto důvodů byla představena metoda, která přípravu datové sady automatizuje. Segmenty jsou extrahovány na

základě porovnání s okrajovými referenčními snímky, a anotace probíhá pomocí automatického rozpoznání mluvené řeči. Díky tomu je minimalizována časová náročnost.

Důkazem funkčnosti této metody je datová sada českého znakového jazyka představená v rámci práce. Ta sestává ze segmentů předpovědi počasí o celkové délce přes 44 hodin. Díky metodě je datová sada poměrně jednoduše rozšiřitelná.

Metoda překladu českého znakového jazyka

Kapitola 4.2 představuje model pro překlad českého znakového jazyka. Jedná se o model Transformer, jenž je moderní model nalézající stále větší uplatnění v široké škále úloh. Model je trénován na datové sadě představené v kapitole 4.1. Oproti jiným publikacím věnujícím se překladu znakového jazyka se představený model soustředí na mnohem delší sekvence, které je schopný překládat. Výsledné hodnoty metrik jsou srovnatelné s aktuálními publikacemi. Model obstál i při kvalitativním vyhodnocení, kde i přes své nedostatky je schopný ve vstupech rozpoznat a přeložit znakový jazyk.

Seznam zdrojů a použité literatury

[1] Starner, T., Weaver, J. & Pentland, A. Real-time American Sign Language Recognition using Desk and Wearable Computer Based Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, roč. 20 č. 12, s. 1371-1375, ISSN 1939-3539. Dostupné z DOI: 10.1109/34.735811.

[2] Lu, C., Amino, S. & Jing, L. Data Glove with Bending Sensor and Inertial Sensor Based on Weighted DTW Fusion for Sign Language Recognition. *Electronics*, 2023, roč. 12, č. 3, s. 613. Dostupné z DOI: 10.3390/electronics12030613.

[3] Simoes Dias, T., Junior, J.J.A.M. & Pichorim, S.F. An Instrumented Glove for Recognition of Brazilian Sign Language Alphabet. *IEEE Sensors Journal*, 2022, roč. 22 č. 3, s. 2518-2529, ISSN 1558-1748. Dostupné z DOI: 10.1109/JSEN.2021.3136790.

[4] Yeh, W.Y., et al. Sign Language Recognition System via Kinect: Number and English Alphabet. *2016 International Conference on Machine Learning and Cybernetics*, 2016, s. 660-665, ISSN 2160-1348. Dostupné z DOI: 10.1109/ICMLC.2016.7872966.

[5] Pugeault, N. & Bowden, R. Spelling it out: Real-time ASL fingerspelling recognition. *2011 IEEE International Conference on Computer Vision Workshops*, 2011, s. 1114-1119, ISBN 978-1-4673-0063-6. Dostupné z DOI: 10.1109/ICCVW.2011.6130290.

[6] Truong, A. & Zaharia, T. Dynamic Gesture Recognition with Laban Movement Analysis and Hidden Markov Models. *Proceedings of the 33rd Computer Graphics International*, 2016, Dostupné z DOI: 10.1145/2949035.2949041.

[7] Li, J., et al. Robust Hand Gesture Recognition Using HOG-9ULBP Features and SVM Model. *Electronics*, 2022, roč. 11, č. 3, s. 988. Dostupné z DOI: 10.3390/electronics11070988.

[8] Kyunghyun, C., et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv*, 2011. Dostupné z DOI: 10.48550/arXiv.1406.1078.

[9] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 1997, roč. 9, č. 8, s. 1735-1780. Dostupné z DOI: 10.1162/neco.1997.9.8.1735.

[10] Kramer, S.F., Wiegmann, D.A. & Kirlik, A. Attention: From History to Application. *Attention: From Theory to Practice*, 2006, s. 3-15. Dostupné z DOI: 10.1093/acprof:oso/9780195305722.003.0001.

[11] Sutskever, I., Vinyals, O., Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv*, 2014. Dostupné z DOI: 10.48550/arXiv.1409.3215.

[12] Vaswani, A. et al. Attention is All you Need. *Advances in Neural Information Processing Systems*, 2017. Dostupné z DOI: 10.48550/arXiv.1706.03762.

[13] Lugaresi, C. et al. MediaPipe: A framework for Building Perception Pipelines. *arXiv*, 2019. Dostupné z DOI: 10.48550/arXiv.1906.08172.

[14] Zhang, F. et al. MediaPipe Hands: On-Device Real-Time Hand Tracking. *arXiv*, 2020. Dostupné z DOI: 10.48550/arXiv.2006.10214.

[15] Bazarevsky, V. et al. Blazepose: On-Device Real-Time Body Pose Tracking. *arXiv*, 2020. Dostupné z DOI: 10.48550/arXiv.2006.10204.

[16] Krejsa, J. & Věchet, S. Czech Sign Language Single Hand Alphabet Letters Classification. *2020 19th International Conference on Mechatronics -*

Mechatronika (ME), 2020, s. 1-5, ISBN 978-1-7281-5602-6. Dostupné z DOI: 10.1109/ME49197.2020.9286667.

[17] Sincan, O. M., & Keles, H. Y. AUTSL: A large scale multi-modal Turkish sign language dataset and baseline methods. *IEEE Access*, 2020, roč. 8, s. 181340-181355. Dostupné z DOI: 10.1109/ACCESS.2020.3028072.

[18] Camgoz, N. C. et al. Neural Sign Language Translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, s. 7784-7793. Dostupné z DOI: 10.1109/CVPR.2018.00812.

[19] Ko, S. K. et al. Neural Sign Language Translation Based on Human Keypoint Estimation. *Applied Sciences*, 2019, roč. 9, č. 13, s. 2683. Dostupné z DOI: 10.3390/app9132683.

[20] Zhou, H. et al. Improving Sign Language Translation with Monolingual Data by Sign Back-Translation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, s. 1316-1325. Dostupné z DOI: 10.1109/CVPR46437.2021.00137.

[21] Camgoz, N. C. et al. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, s. 10023-10033. Dostupné z DOI: 10.1109/CVPR42600.2020.01004.

[22] Wong, R., Camgoz, N. C. & Bowden, R. Sign2GPT: Leveraging Large Language Models for Gloss-Free Sign Language Translation. *The Twelfth International Conference on Learning Representations*, 2024. Dostupné on-line z: openreview.net/forum?id=LqaEEs3UxU.

[23] Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv*, 2016. Dostupné z DOI: 10.48550/arXiv.1603.04467, software dostupný z: tensorflow.org.

[24] Wang, Z. et al. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004, roč. 13, č. 4, s. 600-612. Dostupné z DOI: 10.1109/TIP.2003.819861.

[25] Švec, J. et al. UWebASR – Web-based ASR engine for Czech and Slovak. *CLARIN Annual Conference 2018 Proceedings*, 2018, s. 190-193. Dostupné on-line z: lindat.mff.cuni.cz/services/uwebasr/.

[26] Papineni, K. et al. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, s. 311-318. Dostupné z DOI: 10.3115/1073083.1073135.

[27] Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 2004, s. 74-81. Dostupné on-line z: aclanthology.org/W04-1013.pdf.

[28] Šnajder, J. & Krejsa, J. MediaPipe and its Suitability for Sign Language Recognition. *29th International Conference Engineering Mechanics 2023*, 2023, s. 251-254, ISSN 1805-8256. Dostupné z DOI: 10.21495/em2023-251.

[29] Krejsa, J., Věchet, S. & Šnajder, J. Motion Analysis of Czech Sign Language Alphabet CNN based Classifier via Optical Flow. *30th International Conference Engineering Mechanics 2024*, 2024, s. 170-173, ISSN 1805-8256. Dostupné z DOI: 10.21495/em2024-170.

Strukturovaný životopis

Osobní informace

Jméno:	Ing. Jan Šnajder
Datum narození:	2.8.1995
Státní příslušnost:	CZ
Email:	Jan.Snajder@vutbr.cz

Vzdělání

2008 - 2014	Gymnázium Petra Bezruče, Frýdek-Místek
2014 - 2017	Bc., Základy strojního inženýrství, FSI, VUT v Brně
2017 - 2020	Ing., Mechatronika, FSI, VUT v Brně
2020 - doposud	Ph.D., Inženýrská Mechanika, FSI, VUT v Brně

Studijní pobyty

únor 2019 - červen 2019	Erasmus+	elektrotechnika, TEI of Crete, Řecko
únor 2024 - březen 2024	Freemovers	asistent, NCKU, Taiwan

Pracovní zkušenosti

2018 - 2020	Bender Robotics, Development Intern
2019 - 2020	Garrett Motion, Student Engineer
2020 - doposud	Bender Robotics, Embedded Systems Engineer

Publikace autora

Typ	**Počet**	**Citace**
Konferenční příspěvek	5	1

Jazyková znalost

Český jazyk	rodilý mluvčí
Anglický jazyk	plynulá znalost (C1/C2)

Dovednosti

Programování - C/C++ (embedded), Python, Matlab
mikrokontroléry, zpracování signálu, umělá inteligence

Abstrakt
Tato disertační práce se zabývá vývojem metod strojového učení pro rozpoznání českého znakového jazyka, a to jak v oblasti znakové abecedy, tak gest. Pro obě úlohy rešeršní část shrnuje relevantní metody, datové sady a současný stav poznání. Na základě této analýzy jsou definovány dva hlavní cíle. Všechny představené metody využívají extrakci příznaků v podobě klíčových bodů pózy získaných pomocí nástroje MediaPipe. První cíl využívá tyto klíčové body pro klasifikaci znakové abecedy a diakritiky. Ta je realizována kombinací různých typů neuronových sítí. Výsledná metoda je kvantitativně vyhodnocena a demonstrována v mobilní aplikaci. Druhý cíl se zaměřuje na kontinuální překlad vymezené komunikační oblasti českého znakového jazyka. Pro tyto účely práce představuje rozsáhlou datovou sadu nahrávek předpovědí počasí, pro jejichž překlad je využita architektura Transformer. Výsledný model dosahuje srovnatelných výsledků s podobnými studiemi, avšak pro delší sekvence. Kvalitativní analýza potvrdila, že model úspěšně rozpozná a překládá český znakový jazyk.

Summary
This dissertation focuses on the development of machine learning methods for recognition of Czech Sign Language, for both the sign alhpabet and gestures. For each area, the review section summarizes the relevant methods, available datasets, and current state-of-the-art. Based on this analysis, two main objectives are set. All presented methods utilize feature extraction in the form of pose-based key points obtained via MediaPipe tool. The first objective employs these key points for the classification of sign language alphabet and diacritics, using a combination of different types of neural networks. The resulting method is quantitatively evaluated and demonstrated in a mobile application. The second objective focuses on the continuous translation of a defined communication domain of Czech Sign Language. For this purpose, the dissertation introduces a large dataset of weather forecast recordings which are translated using a Transformer architecture. The resulting model achieves performance comparable to that of similar studies, but for longer sequences. A qualitative analysis confirmed that the model successfully recognizes and translates Czech Sign Language.