

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

Katedra řídicí techniky

## Vývoj rozpoznávače izolovaných, rukou psaných znaků pro účel rychlé volby kontaktů na dotykovém telefonu pro nevidomé

**Eva Uhliariková**

Vedoucí: Ing. Daniel Novák, Ph.D.  
Obor: Kybernetika a robotika  
Studijní program: Systémy a řízení  
Květen 2018



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Uhliariková** Jméno: **Eva** Osobní číslo: **439584**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Kybernetika a robotika**  
Studijní obor: **Systemy a řízení**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Vývoj rozpoznávače izolovaných, rukou psaných znaků pro účel rychlé volby kontaktu na dotykovém telefonu pro nevidomé**

Název bakalářské práce anglicky:

**Development of a Recognizer for Isolated Hand-Drawn Characters for the Purpose of a Quick-Dial Function in a Phone for the Visually Impaired**

Pokyny pro vypracování:

1. Seznamte se s existujícím mobilním systémem pro nevidomé uživatele vyvíjeném na katedře kybernetiky [1]
2. Seznamte se s problematikou on-line i off-line rozpoznávání ručně psaných izolovaných znaků [2] [3] a s algoritmy použitými v opensource projektu CellWriter [4]
3. Implementujte jednoduchý rozpoznávač číslic a velkých písmen jako aplikaci pro Android a integrujte ho do existujícího systému v roli rozhraní pro rychlou volbu telefonních kontaktů.
4. Kvantitativně otestujte spolehlivost výsledného rozpoznávače na nejméně deseti uživateli, z toho pěti nevidomých nebo těžce slabozrakých.
5. Zhodnoťte vhodnost celkového konceptu rychlého vytáčení pomocí ručně psaných znaků pro cílovou skupinu nevidomých nebo těžce slabozrakých uživatelů.

Seznam doporučené literatury:

- [1] Petr Svobodník: Zpřístupnění mobilních telefonů se systémem Android pro nevidomé uživatele. Diploma thesis. Czech Technical University in Prague, 2013.
- [2] B. K. Iwana, V. Frinken and S. Uchida: A Robust Dissimilarity-Based Neural Network for Temporal Pattern Recognition. 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, 2016, pp. 265-270.
- [3] B. Szöcs, S. Vajda and J. Robu: D.A.C Draw and Calc - the intuitive calculator. 2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, Subotica, 2012, pp. 157-163.
- [4] Levin, Michael. CellWriter: Grid-Entry Handwriting Recognition. 2007.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**doc. Ing. Daniel Novák, Ph.D., Analýza a interpretace biomedicínských dat FEL**

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

Datum zadání bakalářské práce: **15.01.2018**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

doc. Ing. Daniel Novák, Ph.D.  
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.  
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studentky

## Poděkování

Děkuji doc. Ing. Danielovi Novákovi PhD. za vedení a za možnost pracovat na tomto projektu. Dále bych chtěla poděkovat Ing. Janu Hadáčkovi za užitečné rady a pomoc. Velký dík patří příteli a rodině za podporu. V neposlední řadě bych chtěla poděkovat profesoru Giorgio Giacinto za vstřícnou pomoc během vypracovávání práce v zahraničí. Speciální dík patří také všem dobrovolníkům, jež byli ochotní podílet se na sběru dat a testování.

## Prohlášení

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

V Praze, 25. května 2018

## Abstrakt

Tato práce seznamuje čtenáře s kmenovou aplikací, která je primárně určena pro osoby se zrakovým postižením. Dále navrhuje a implementuje novou funkci zrychleného volání za pomoci ručně psaných znaků, jež je speciálně navržena pro použití v mobilním telefonu s kmenovou aplikací. Řeší vhodné metody rozpoznávání a navrhuje vhodnou variantu, jež je následně implementována do Android aplikace. V této práci jsou představeny i nové originální sady dat, jež jsou použity k rozpoznávání znaků a byly pořízeny za spolupráce s dobrovolníky z více než 10 evropských zemí. V závěru je představeno testování této aplikace na cílových uživateliích a taktéž diskuze nad výsledky tohoto testu.

**Klíčová slova:** rozpoznávání znaků; rukopis

**Vedoucí:** Ing. Daniel Novák, Ph.D.  
ČVUT,  
Fakulta Elektrotechnická,  
Katedra Kybernetiky,  
Resslova 307/9,  
Praha

## Abstract

This work familiarises a reader with the root application primarily used for visually impaired users. It proposes and implements a new function of quick-dial with the use of hand-written characters which is specially designed for mobile phone usage with the root application mentioned above. It researches suitable methods of recognition and presents the appropriate approach which is further implemented in the Android application. Moreover, this paper presents new data sets which are used for the character recognition and were gathered with the help of international volunteers from more than ten European countries. Lastly, the user testing results are presented together with a discussion of their conclusions.

**Keywords:** characters recognition, handwriting

**Title translation:** Development of a Recognizer for Isolated Hand-Drawn Characters for the Purpose of a Quick-Dial Function in a Phone for the Visually Impaired

# Obsah

<b>1 Úvod</b>	<b>1</b>	6.3 Realizace rozpoznávače v Matlabu	28
1.1 Motivace	1	6.4 Prototyp rozpoznávací aplikace	28
1.2 Cíl	1	<b>7 Vyhodnocení klasifikátoru</b>	<b>31</b>
<b>2 Mobilní telefony pro osoby se zrakovým postižením</b>	<b>3</b>	7.0.1 Testování úspěšnosti	31
2.1 Osoby se zrakovým postižením	3	7.0.2 Procentuální úspěšnost klasifikace	31
2.2 Mobilní telefony pro nevidomé	3	7.0.3 Matice záměn (Confusion matrix)	31
2.2.1 iPhone	3	7.0.4 TP, TN, FP, FN	32
2.2.2 Dotykové telefony se systémem Android	4	7.0.5 ROC křivka	33
2.2.3 Telefony s velkými tlačítky	4	<b>8 Uživatelské testování</b>	<b>35</b>
2.3 Kmenová aplikace	4	8.1 Dotazník před testováním	35
2.3.1 Ovládání	4	8.2 Scénář testování	37
2.3.2 Uživatelské rozhraní	5	8.3 Dotazník po testování	37
2.3.3 Menu a funkce	5	8.4 Výsledky testování	38
2.4 Funkce zrychleného volání	7	<b>9 Rozšíření o písmena velké abecedy</b>	<b>41</b>
<b>3 Úvod do rozpoznávání znaků</b>	<b>9</b>	<b>10 Závěr</b>	<b>43</b>
3.1 Off-line rozpoznávání	9	10.1 Budoucí práce	43
3.2 On-line rozpoznávání	10	<b>A Použitá cizí slova a zkratky</b>	<b>45</b>
3.2.1 CellWriter	12	<b>B Výsledky testování</b>	<b>47</b>
<b>4 Klasifikace</b>	<b>13</b>	B.1 Dotazník před testováním	47
4.1 Příznak	13	B.2 Testování úspěšnosti rozpoznávače	47
4.2 Techniky rozpoznávání	14	B.3 Dotazník po testování	48
4.2.1 Algoritmus k-nejbližších sousedů	14	<b>C Grafická ukázka sady znaků</b>	<b>49</b>
4.3 Redukce dimenzí u příznakových vektorů	15	<b>D Obsah přiloženého CD</b>	<b>51</b>
4.3.1 Analýza hlavních komponent (PCA)	15	<b>E Literatura</b>	<b>53</b>
4.3.2 Lineární diskriminační analýza (LDA)	16		
<b>5 Návrh řešení</b>	<b>17</b>		
5.1 Preprocessing	18		
5.1.1 Normalizace a binarizace	18		
5.1.2 Gaussovo rozmazání (Gaussian Blur)	19		
5.1.3 Gradient	19		
5.1.4 Kontury objektu	19		
5.2 Výběr příznaků	20		
5.3 Redukce příznaků	20		
5.4 Rozpoznání	22		
5.5 Možnost odmítnutí	22		
<b>6 Implementace</b>	<b>25</b>		
6.1 Použité nástroje	25		
6.2 Aplikace pro sběr dat	26		

## Obrázky

2.1 Ukázka uživatelského rozhraní . . .	5
2.2 Grafické znázornění hierarchie menu . . . . .	6
3.1 Obecný off-line rozpoznávací systém . . . . .	9
3.2 Obecný on-line rozpoznávací systém . . . . .	11
4.1 Obrázek znázorňující princip algoritmu k-NN . . . . .	14
5.1 Návrh postupu pro získání příznakového vektoru . . . . .	18
5.2 Ukázka gradientu na číslici 0 podle osy X a osy Y . . . . .	19
5.3 Preprocessing znaku 0 . . . . .	20
5.4 Ukázka 20 hlavních komponent získaných pomocí PCA z binárních masek trénovacích znaků . . . . .	21
5.5 Uspořádání trénovacích dat v prostoru dle 3 nejvýznamnějších dimenzí po použití LDA (ve skutečnosti jsou data v 10 dimenzovém prostoru) . . . . .	22
6.1 Aplikace pro sběr dat . . . . .	26
6.2 Příklady testovacích znaků, které nepatří do žádné třídy. . . . .	28
7.1 Tabulka záměn . . . . .	32
7.2 ROC křivka . . . . .	34
C.1 Ukázka znaků ze sady . . . . .	50

## Tabulky

6.1 Přehled zastoupení národností mezi dobrovolníky . . . . .	27
6.2 Počet vzorků v datové sadě . . . . .	27
7.1 Matice záměn . . . . .	32
7.2 Hodnoty pro TP, TN, FP a FN při použití našeho klasifikátoru . . . . .	33
7.3 Tabulka měření výkonnosti klasifikátoru . . . . .	33
8.1 Matice záměn . . . . .	38
8.2 Hodnoty TP, TN, FP a FN dosažené během testování . . . . .	39
8.3 Úspěšnost rozpoznání znaků pro jednotlivé účastníky testování . . . . .	39
B.1 Odpovědi testovaných dobrovolníků na dotazník před testováním. . . . .	47
B.2 Přehled rozpoznávaných znaků . . . . .	48
B.3 Tabulka s odpověďmi na dotazník po testování . . . . .	48



# Kapitola 1

## Úvod

### 1.1 Motivace

Mobilní telefony se za posledních několik let staly nedílnou součástí našich životů. Zatímco na přelomu 20. a 21. století si lidé začínali pořizovat své první tlačítkové mobilní telefony, dnes zažíváme přesun k chytrým dotykovým "smartphonům", které nabízí, mimo možnost volání a posílání SMS zpráv, širokou škálu dalších funkcí. Navzdory tomu, že pro většinu lidí je ovládání těchto dotykových mobilů velmi snadné a intuitivní, tak existují také početné skupiny, pro které může být komplikované si z takového telefonu pouze zavolat, natož pak používat ostatní funkce. Příkladem jedné z těchto skupin jsou osoby se zrakovým postižením. V současné době dle Světové zdravotnické organizace můžeme na světě nalézt až 253 milionů zrakově postižených osob [1] a právě mobilní telefon má díky své kompaktnosti a všestrannosti potenciál stát se výborným pomocníkem nejenom při telefonování, ale také při jiných každodenních činnostech. Této příležitosti využili na Fakultě elektrotechnické ČVUT na katedře počítačů, kde byl Petrem Svobodníkem navrhnout launcher<sup>1</sup> pro systém Android, který má za úkol co nejvíce usnadnit používání dotykového telefonu právě zrakově postiženým lidem. V této chvíli má kmenová aplikace již mnoho speciálních funkcí, jako je například čtečka knih, rozpoznávání barev či rozpoznávání bankovek. Tématem této bakalářské práce bude rozšíření tohoto systému o další zajímavou funkci a to o funkci rychlého volání pomocí ručně psaného znaku.[2]

### 1.2 Cíl

Hlavním cílem této práce je vytvoření funkčního rozpoznávače číslic, který bude následně implementován v mobilní aplikaci. Aplikace je určena především pro speciální mobilní telefon s kmenovou aplikací a má za úkol zajistit funkci zrychlené volání, která by usnadnila lidem se zrakovým postižením uskutečnit telefonní hovor až 10 oblíbeným kontaktům. Problém bude řešen ve čtyřech krocích. Během prvního kroku bude implementován jednoduchý program pro systém Android, který následně umožní sběr dat nezbytných

---

<sup>1</sup>aplikace, která upravuje vzhled a funkce systému

pro zhotovení a testování funkčního klasifikátoru. Tyto data je nezbytné posbírat za účasti dostatečného množství dobrovolníků z rozličných sociálních, věkových, ale i národních skupin. Druhý krok bude zaměřen na implementaci vhodného a funkčního rozpoznávače číslic. Během jeho vytváření budou zváženy různé metody a postupy, pro získání co nejuspokojivějších výsledků. V předposledním kroku bude zhotovena funkční aplikace pro mobilní telefon, ve které bude zahrnut patřičný rozpoznávač. A během závěrečného kroku bude vytvořená aplikace otestována na cílové skupině nevidomých osob. Toto testování je velmi důležitým krokem, jelikož jen při něm je možné zjistit, zdali je tato aplikace pro osoby nevidomé či jinak zrakově indisponované vyhovující a je-li tedy možné ji bez problému využívat.

## Kapitola 2

### Mobilní telefony pro osoby se zrakovým postižením

#### 2.1 Osoby se zrakovým postižením

Lidské oko je zrakovým orgánem a zrak je jedním z pěti základních smyslů člověka, který zdravé osobě umožňuje vnímat až 80% všech informací, jež přijímá z okolního světa. Oko je složeno z několika částí a v případě, že se v kterékoliv z těchto částí nachází vada či zde dojde k poškození, tak dochází k vadám zraku. Vady zraku jsou v dnešní době běžné a dioptrické brýle potřebuje velká část dospělé populace. Pouze v případě, že je pro člověka již nedostačující korigovat zrakovou vadu dioptrickými brýlemi a tato vada má dopad na činnosti v běžném životě, říkáme, že má osoba zrakové postižení. Osoby se zrakovým postižením je možné rozřadit do několika skupin, přičemž Světová zdravotnická organizace (WHO) používá následující řazení:

- Slabá nebo žádná slabozrakost
- Střední slabozrakost
- Silná slabozrakost
- Těžce slabý zrak
- Praktická nevidomost
- Úplná nevidomost

[3, par.54] Toto řazení je pouze jedním z mnoha možných. U rozličných organizací a skupin se můžeme setkat s různými klasifikacemi zrakového postižení.

#### 2.2 Mobilní telefony pro nevidomé

##### 2.2.1 iPhone

Přestože mobilní telefon iPhone není primárně určen pro osoby se zrakovým postižením, tak disponuje několika funkcemi, které umožňují těmto lidem jeho

použití. Mezi tyto funkce patří VoiceOver či lupa. VoiceOver je na gestech založená funkce čtení displeje, která je kompatibilní také se všemi aplikacemi pro iPhone. Dále lze využít virtuálního asistenta Siri, jež umožní uživateli ovládat jeho telefon pomocí hlasu. Mimo to je možné iPhone spárovat přes Bluetooth s obnovitelnou Brailovou klávesnicí. [4]

### ■ 2.2.2 Dotykové telefony se systémem Android

Taktéž mobilní telefony s Androidem disponují funkcemi, jež usnadňují použití telefonu nevidomým lidem. TalkBack je funkce, jež stejně jako VoiceOver seznamuje uživatele s obrazovkou pomocí hlasového výstupu. Funkce lupa je dostupná pouze na některých verzích systému Android. Také je možné pomocí změn v nastavení upravit displej tak, aby byl lépe čitelný. [5]

### ■ 2.2.3 Telefony s velkými tlačítky

Tyto telefony bývají zpravidla určeny pro seniory a disponují často pouze základními funkcemi jako volání či posílání textových zpráv. Díky jejich jednoduchému designu a velkým tlačítkům, jsou taktéž vhodnou alternativou pro osoby se zrakovým postižením. Tyto zařízení lze najít v nabídce nejrůznějších výrobců mobilních telefonů, například Aligator, Alcatel či Doro.

## ■ 2.3 Kmenová aplikace

Kmenová aplikace byla představena již v roce 2014. Tato aplikace je typu launcher a dříve byla dostupná ke stažení pro všechny telefony se systémem Android. Později však byla integrována do samotného mobilního telefonu určeného pro zrakově postižené osoby. Telefon je vybaven předním i zadním fotoaparátem, nabízí možnost připojit se k bezdrátovým sítím, obsahuje sloty pro 2 SIM karty a disponuje senzory jako jsou například akcelerometr, senzor přiblížení či senzor světla.[2]

### ■ 2.3.1 Ovládání

Jednoduchost kmenové aplikace je založena především na tom, že je možné vše ovládat za pomoci čtyř následujících gest:

1. Krátký dotek jedním prstem, který slouží k procházení menu.
2. Dlouhý dotek jedním prstem slouží k vybrání aktuální položky, přijetí hovoru, potvrzení dialogu nebo otázky a vypnutí budíku.
3. Krátký dotek dvěma prsty může zprostředkovat přečtení nebo zopakování aktuální položky, přečtení napsaného textu, přečtení jména kontaktu nebo čísla při příchozím hovoru, start nebo pauzu u MP3 přehrávače, rádia nebo u čtečky knih

4. Dlouhý dotek dvěma prsty je určen ke kroku zpět o jednu úroveň, k zavěšení příchozího hovoru, k opuštění probíhající činnosti nebo k odpovědi na dialog či otázku.

Díky tomuto ovládání je velice snadné a rychlé se s telefonem naučit pracovat. [2]

### 2.3.2 Uživatelské rozhraní

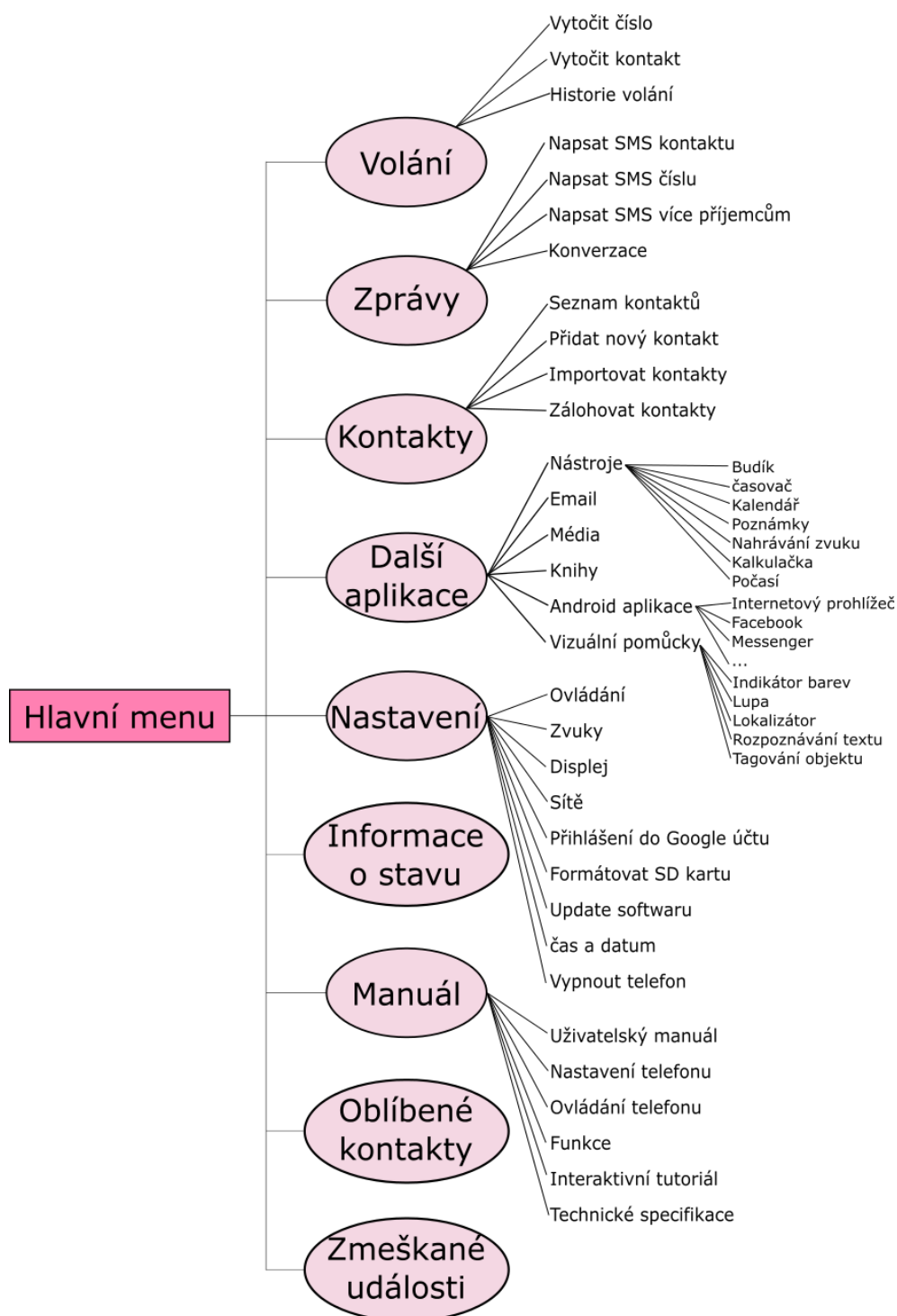
Uživatelské rozhraní kmenové aplikace je tvořeno pouze černými obrazovkami s velkými bílými nápisy. Je tomu tak právě proto, aby bylo pro osoby, které mají sice zrakové postižení, ale nejsou úplně nevidomí, snadné se orientovat a aby byly z obrazovky eliminovány rušivé elementy. Přesto je grafický výstup na displeji pro orientaci až druhotný. Hlavním prostředkem, který umožňuje práci s telefonem, jsou zvukové komentáře, které seznamují uživatele s textem na obrazovce, či s tím na čem se zrovna nachází váš prst. Hlasový výstup doplňují vibrace, které umožňují přisun informací skrz hmat. [2]



Obrázek 2.1: Ukázka uživatelského rozhraní

### 2.3.3 Menu a funkce

Všechny funkce kmenové aplikace jsou uspořádány v hierarchickém menu, které umožňuje jednoduchou orientaci.



**Obrázek 2.2:** Grafické znázornění hierarchie menu

Kmenová aplikace kromě volání a psaní SMS zpráv nabízí další tradiční funkce jako budík, kalendář či poznámky. Mimo to nalezneme v telefonu i mnohem pokročilejší funkce zaměřené přesně na potřeby nevidomých. Smartphone uživateli nabízí čtečku knih, indikátor barev, rozpoznání bankovek, ale

také lze využít běžné Android aplikace s funkcí TalkBack, jako jsou například Facebook, WhatsApp či internetový prohlížeč.[2]

## 2.4 Funkce zrychleného volání

Funkce zrychleného volání je dnes k dispozici ve většině mobilních telefonů. Zrychlené volání můžeme najít na různých telefonech v rozličných variantách. Na klasických tlačítkových je možno zrychleně volat pomocí dlouhého stisku určitého tlačítka na klávesnici. V dnešních moderních dotykových telefonech můžeme zrychleného volání využívat taktéž ve formě dlouhého stisku číslice na digitální klávesnici nebo lze oblíbené kontakty rovnou ukládat na plochu telefonu. Existují i telefony, které mají speciální SOS tlačítko, jež se zpravidla nachází na zadní straně telefonu, které po zmáčknutí okamžitě zavolá přiřazenému kontaktu či na zadané pohotovostní číslo. Přestože je to také forma zrychleného volání, je určena především pro nouzové situace.





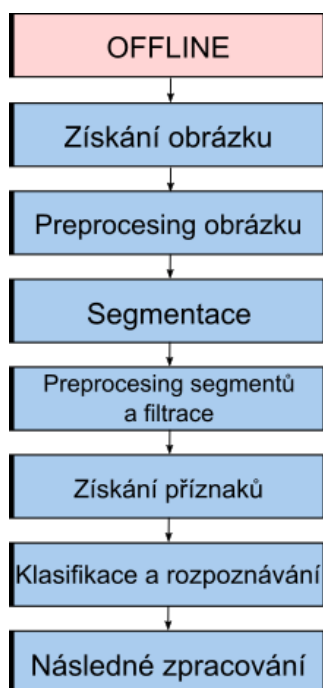
## Kapitola 3

### Úvod do rozpoznávání znaků

Rozpoznávání znaků je problém, při kterém počítač automaticky identifikuje vložený znak, a který lze řešit několika způsoby. Vstupem je naskenovaný dokument, obrázek či data ze zařízení jako jsou mobilní telefon nebo tablet. Máme-li přichystán vstup, nic nám nebrání přistoupit k rozpoznávání. V závislosti na typu vstupních dat lze rozpoznávání dělit do dvou základních skupin a to na off-line a on-line. [6]

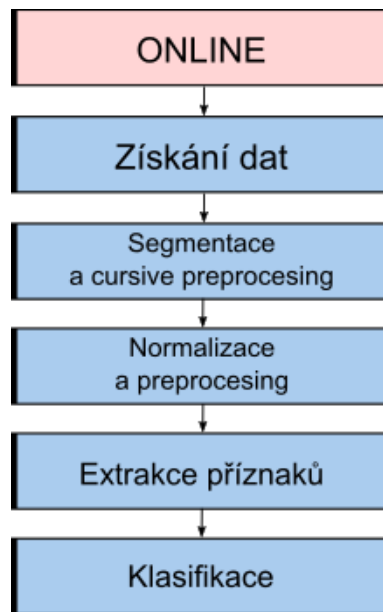
#### 3.1 Off-line rozpoznávání

Off-line rozpoznávání je charakteristické tím, že vstupní data jsou zpravidla získána skenováním dokumentů či pořízením obrázku z fyzických předloh z reálného světa. Do klasifikátoru jsou tedy následně vždy vkládána data ve formě pixelů.



Obrázek 3.1: Obecný off-line rozpoznávací systém





**Obrázek 3.2:** Obecný on-line rozpoznávací systém

#### 1. Získání dat

Data pro on-line rozpoznávání jsou zachycována při přeježdění po digitálním povrchu, během čehož lze zaznamenávat nejenom souřadnice, ale také sílu dotyku, čas nebo pohybovou akci.

#### 2. Segmentace a cursive preprocessing

Samotné znaky či jiné části lze od sebe oddělit například pomocí jednotlivých tahů či v závislosti na čase zápisu. Pomocí 'cursive preprocessing' jsou extrahována jednotlivá slova či písmena z textu, což při klasifikaci rukopisu znamená, že hledá optimální hrany pro rozdělení.

#### 3. Normalizace a preprocessing

Preprocessing u on-line rozpoznávání je možno rozdělit na

- Redukce informací - jelikož počítač přijímá data v určité vzorkovací frekvenci, je potřeba z těchto dat vybrat pouze důležitá a odstraňovat duplicitní body
- Eliminace nedokonalostí - jedná se například o vyhlazení nedokonalostí vzniklých třesem ruky či nedokonalostí hardwaru, redukci falešných bodů (wild point reduction), které se objevují především kvůli nedokonalosti hardwaru či o odstranění háků (Hook removal), jež často vznikají při dotknutí a zvednutí pera z displeje
- Normalizace - je používána k normalizaci náklonu, velikosti slov, znaků či komponent a korekci posunutí dolní linie (baseline drift corection) [8]

Po tomto kroku již lze využít klasifikační metody 'Template matching'.

4. Extrakce příznaků

Z připravených dat lze extrahovat nejrůznější příznaky. Oproti off-line rozpoznávání lze využít příznaků jako jsou například rychlost psaní či počet tahů.

5. Klasifikace

Tento krok probíhá stejně jako u off-line rozpoznávání.

### 3.2.1 CellWriter

Zajímavým příkladem on-line rozpoznávacího systému může být například open source program CellWriter, jež slouží k rozpoznávání jednotlivých znaků. Tento program musí každý uživatel před trénovat vlastními testovacími vzorky. Na získané vzorky se aplikuje vyhlazovací algoritmu, který odstraní nedokonalosti. Během preprocessingu jsou vykonány úkony jako diskvalifikace trénovacích vzorků s pravděpodobně špatným výsledkem při primárním rozpoznávání, optimální mapping tahů mezi trénovacím a testovacím znakem či penalizace trénovacích vzorků s výraznými rozdíly oproti vstupu. Pro výpočet vzdálenosti mezi vzorky se užívá čtvercové Eukleidovské vzdálenosti (squared Euclidean distance). [9]

## Kapitola 4

### Klasifikace

Klasifikace je problémem, jež lze rozřadit do dvou samostatných skupin:

- Binární klasifikace

Při tomto typu klasifikace, dochází k zařazování objektu do jedné ze dvou možných tříd. Příkladem metod, které bývají použity při binární klasifikaci jsou rozhodovací stromy, Bayesovské sítě, neuronové sítě a další.

- Multiclass klasifikace

Během 'multiclass' klasifikace je počet tříd, do kterých může být prvek zařazen, tři a více. Pro tuto klasifikaci jsou používány například naive Bayes, k-NN, neuronové sítě, rozhodovací stromy a mnohé další metody. /cite25

#### 4.1 Příznak

Mluvíme-li o příznaku v souvislosti se strojovým učením a rozpoznáváním vzorů, pak se jedná o měřitelnou či pozorovatelnou vlastnost objektu, který chceme použít při klasifikaci. Čím více příznaků objekt popisuje, tím detailnější bude jeho popis a je větší šance, že bude správně klasifikován. Soubor příznaků jednoho objektu pak tvoří příznakový vektor.[?]cite:1)

Existuje několik typů příznaků:[4]

- a) Kategorické

Příkladem kategorických příznaků mohou být krevní skupiny. Mezi některými příznaky této skupiny nelze počítat vzdálenost.

- b) Binární

Binární příznak je takový, který může nabývat pouze jednu ze dvou možných variant. Například tyto příznaky mohou vyjadřovat, zda zkoumaný objekt má či nemá určitou vlastnost.

- c) Ordinální

Tyto příznaky mají svou posloupnost. Příkladem je věk. Umožňují měřit vzdálenost mezi objekty.

- d) Celočíselné  
Příznak je vyjádřen celočíselnou hodnotou.
- e) Reálné  
Příznak je vyjádřen pomocí reálného čísla.

## 4.2 Techniky rozpoznávání

Pro rozpoznávání ručně psaných izolovaných znaků můžeme najít v literatuře mnohé techniky. Patří mezi ně například

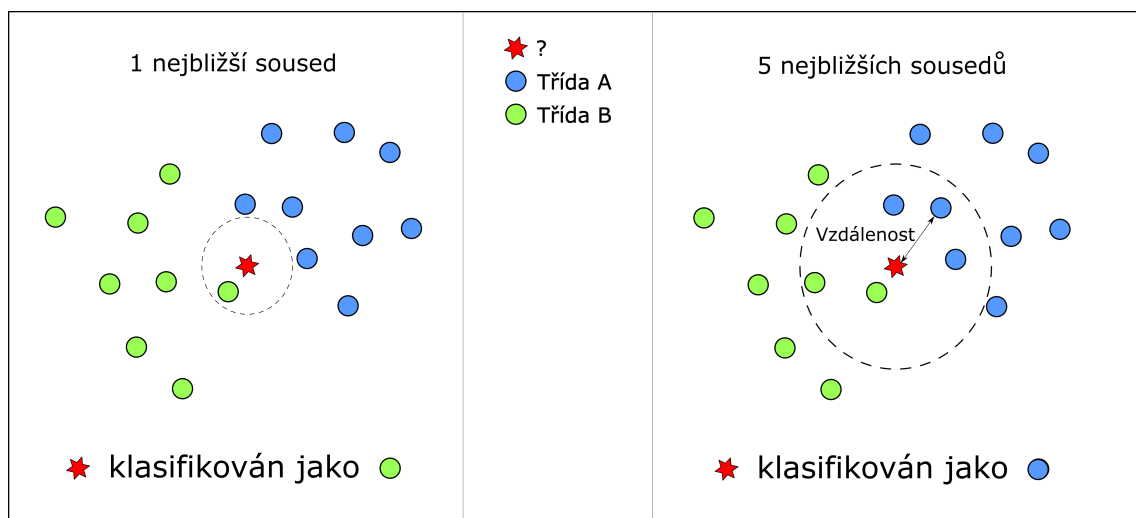
- Template matching (tato metoda nepracuje s příznakovými vektory)
- Rozhodovací stromy, rozhodovací seznamy
- Statistické klasifikátory (Bayesův klasifikátor, skrytý Markovův model (HMM) a další),
- Neuronové sítě a deep learning

[10]

Během procesu rozpoznávání hrají velmi významnou roli ve výsledném výkonu preprocessing, správný výběr příznaků a postprocessing. [10]

### 4.2.1 Algoritmus k-nejbližších sousedů

Algoritmus k-nejbližších sousedů (k-NN či k-nearest neighbors) je jednou z možných metod užívaných pro klasifikaci. Mezi algoritmy strojového učení patří k těm nejjednodušším. Při klasifikaci objektu pomocí algoritmu k-NN je v prostoru vyhledáno právě **k** nejbližších sousedů, přičemž objekt je přiřazen do třídy, která je mezi **k** sousedy zastoupena nejvíce krát.



Obrázek 4.1: Obrázek znázorňující princip algoritmu k-NN

Při klasifikaci pomocí k-NN lze využít nejrůznějších metrik pro výpočet vzdáleností mezi objekty. Příkladem může být

- Euklidovská metrika,

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4.1)$$

, kde  $d$  je vzdálenost dvou stejně velkých vektorů  $\vec{x}$  a  $\vec{y}$  o délce  $n$ .

- Manhattanská metrika

$$d(\vec{x}, \vec{y}) = \sum_{i=1}^n |y_i - x_i| \quad (4.2)$$

, kde  $d$  je vzdálenost dvou stejně velkých vektorů  $\vec{x}$  a  $\vec{y}$  o délce  $n$ .

- Hammingova metrika, která udává, v kolika pozicích se dva stejně dlouhé řetězce liší a při užití s binárními řetězci pro ní platí stejný matematický vztah jako pro metriku Manhattanskou,

- Minkovského metrika,

$$d(\vec{x}, \vec{y}) = \sqrt[m]{\sum_{i=1}^n (|x_i - y_i|)^m} \quad (4.3)$$

, kde  $d$  je vzdálenost dvou stejně velkých vektorů  $\vec{x}$  a  $\vec{y}$  o délce  $n$  a  $m > 1$ . V případě, že se  $m$  limitně blíží  $\infty$ , jedná se již o metriku Čebyševovu.

- Spearmanova a další.[11]

## 4.3 Redukce dimenzí u příznakových vektorů

Redukce dimenzí je mapování dat do nižších dimenzí tak, aby nedůležité informace mohly být odstraněny a zůstali pouze ty nejvýznamnější. Redukci dimenzí u příznakových vektorů je možné provést dvěma způsoby:

- **Selekce příznaků** vybírá podmnožinu důležitých příznaků z originální sady.
- **Extrakce příznaků** transformuje data z vysoko dimenzionálního prostoru do prostoru o méně dimenzích. Patří zde například PCA, LDA a další.[12]

### 4.3.1 Analýza hlavních komponent (PCA)

PCA nebo-li analýza hlavních komponent je jednou z nejpoblárnějších technik pro redukcii dimenzí. Pro realizaci je nutné sestavit z originálního datasetu význačný vektor a kovarianční matici a následně z ní vypočítat vlastní vektory, které jsou pak seřazeny sestupně podle hodnot vlastních čísel. Vlastní vektory s největšími vlastními čísly nesou nejvíce informací. [13] [14]

### ■ 4.3.2 Lineární diskriminační analýza (LDA)

Počítá lineární transformaci, která maximalizuje Euklidovskou vzdálenost mezi prostředky jednotlivých tříd zatímco minimalizuje variance uvnitř třídy. [14, p.12]



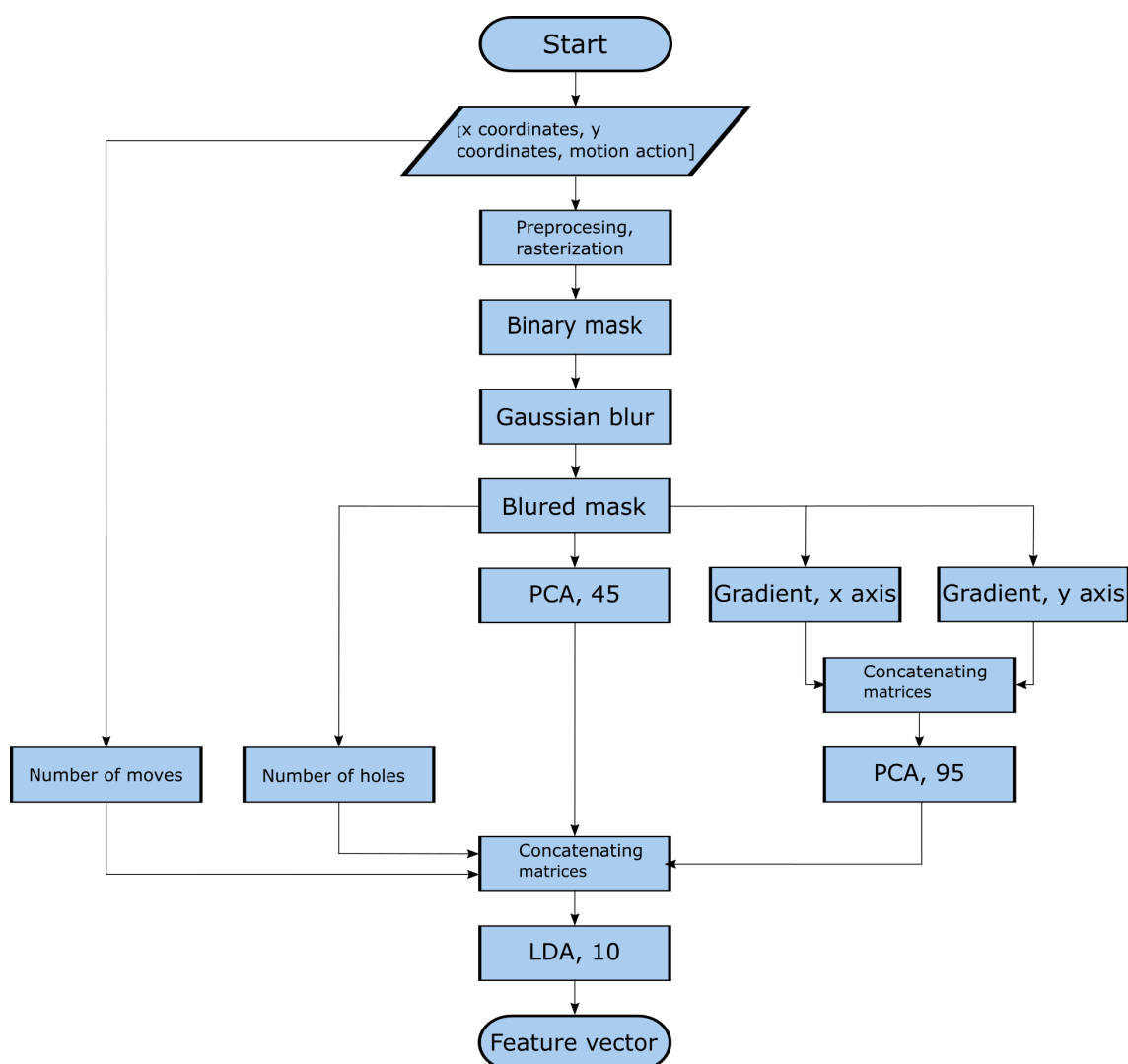


## Kapitola 5

### Návrh řešení

Jelikož v navrhovaném systému jsou vstupní data ve formě vektorů a byla zaznamenána na displeji mobilního telefonu, jedná se on-line rozpoznávání. Tato data zaznamenávají obrysy jednotlivých čísel. Při vytváření návrhu řešení byly zvažovány různé metody a postupy. Podobný problém byl již řešen například v aplikaci CellWriter[13], která avšak vyžaduje před-trénování od uživatele nebo také v projektu D.A.C, který se zaměřuje na rozpoznávání číslic a užívá k tomu metodu neuronových sítí[15], která může při rozpoznávání znaků dosahovat vynikajících výsledků. [16]

Pro vytvoření našeho rozpoznávače byl však vybrán algoritmus k-NN, který byl zvolen díky své jednoduchosti a přesto poměrně dobré úspěšnosti. Návrh postupu pro získání příznakového vektoru je uveden ve vývojovém diagramu 5.1. Během návrhu bylo také zvažováno použití Freemanova Chain kódu jako příznaku, avšak nakonec nebylo přistoupeno k jeho použití.



Obrázek 5.1: Návrh postupu pro získání příznakového vektoru

## 5.1 Preprocessing

Abychom ze získaných dat mohli dostat jednotlivé příznaky, je nutné aby tyto data prošla preprocessingem.

### 5.1.1 Normalizace a binarizace

Jelikož je každý znak na vstupu jiné velikosti, je zapotřebí jej přeškálovat na fixní rozměry: výška = 48 a šířka = 27. Tyto rozměry byly určeny experimentálně během implementace. Vstupní znaky jsou ve vektorové formě a pro získání vybraných příznaků je nutné převedení na binární masku. Tohoto lze

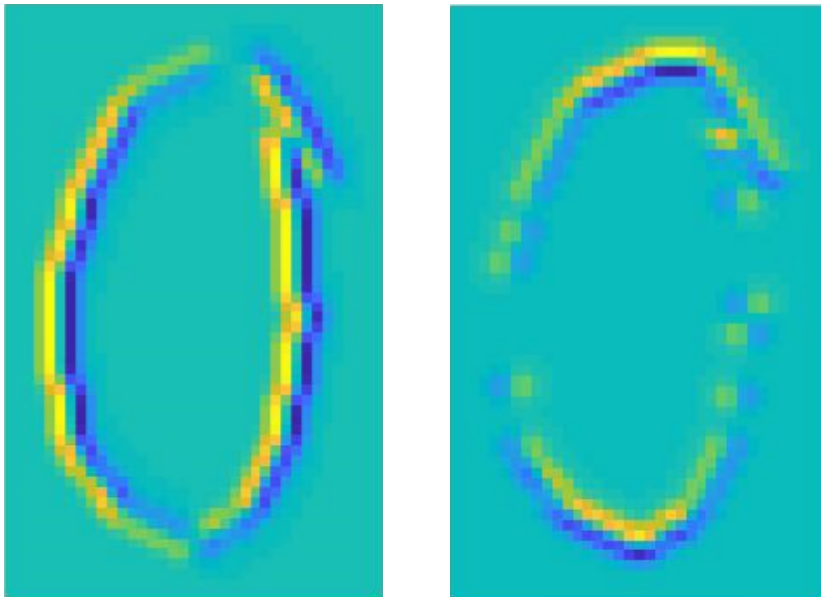
docílit použitím Bresenhamova algoritmu, který zajistí vykreslení linií mezi dvěma body na bitmapu.

### ■ 5.1.2 Gaussovo rozmazání(Gaussian Blur)

Jelikož šířka linie znaku je pouze jeden pixel, je vhodné získanou binární masku rozmazat, aby pravděpodobnost, že se dva znaky překryjí, byla vyšší. Na binární masku je aplikováno Gaussovo rozmazání se směrodatnou odchylkou 0,7, která byla určena experimentálně.

### ■ 5.1.3 Gradient

Z rozmazané masky lze následně následně získat obrazové gradienty vůči ose X a ose Y.



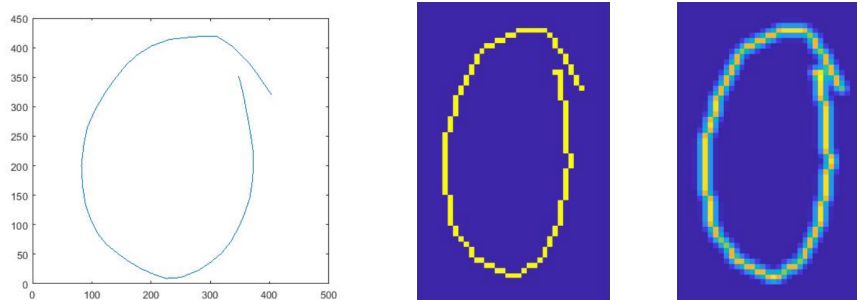
Obrázek 5.2: Ukázka gradientu na číslici 0 podle osy X a osy Y

### ■ 5.1.4 Kontury objektu

Rozmazaná maska je použita k nalezení kontur objektu. Tyto kontury můžeme využít k určení počtu děr ve znaku, přičemž platí

$$\text{počet děr ve znaku} = \text{počet kontur} - 1$$

. Rozmazaná maska je použita, jelikož v případě kdy číslo zůstane nepatrně otevřené, tak se rozmazáním uzavře. Tento příklad lze vidět na obrázku 5.3.



Obrázek 5.3: Preprocessing znaku 0

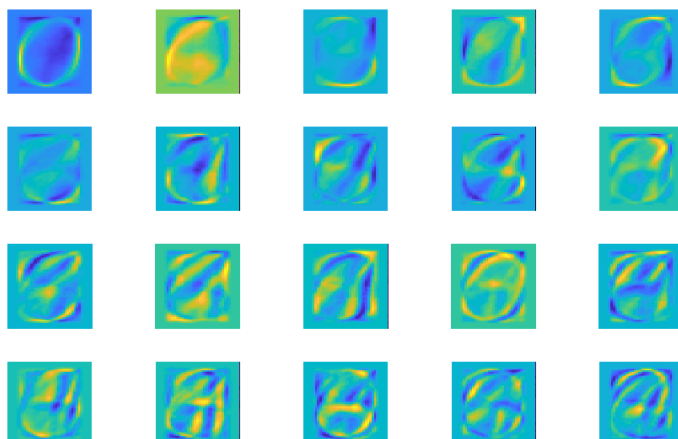
## 5.2 Výběr příznaků

Vybrání správných příznaků je klíčové pro vytvoření dobře fungujícího klasifikátoru. Pro náš klasifikátor byly vybrány následující příznaky:

- Počet tahů  
Tento příznak je získán z originálních dat.
- Počet děr ve znaku
- Gradienty podle osy x a y
- Pixely rozmazané masky

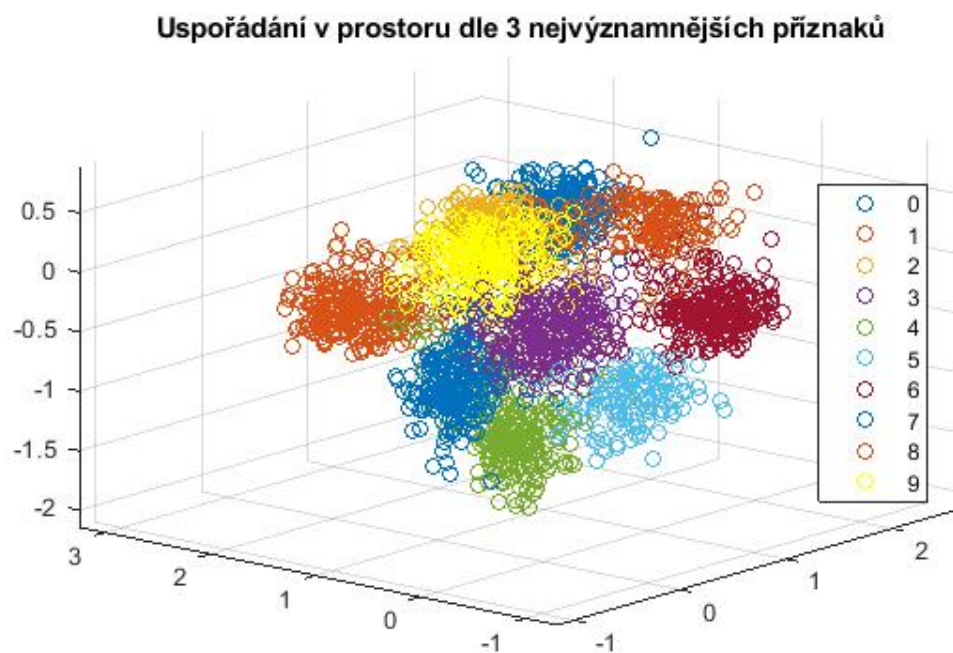
## 5.3 Redukce příznaků

Jelikož dimenze příznaků získaných z rozmazané masky a z masek gradientů jsou ve velkých dimenzích, je nutné tyto příznaky zredukovat. V tomto případě je pro redukcí těchto dat zvolena analýza hlavních komponent. Tato metoda zde bude použita dvakrát, jednou na rozmazanou masku a podruhé na gradienty. PCA je zde použito dvakrát, jelikož příznaky z těchto dvou skupin nejsou stejného charakteru a musely by být přenásobeny konstantou. Dimenze, na kterou budou příznaky redukovány, byla určena experimentálně.



**Obrázek 5.4:** Ukázka 20 hlavních komponent získaných pomocí PCA z binárních masek trénovacích znaků

Po extrahování nejdůležitějších příznaků z rozmazané masky a gradientů je z těchto příznaků vytvořen nový příznakový vektor, do kterého jsou ještě přidány příznaky počet děr a počet tahů. Jelikož je dimenze tohoto vektoru stále velmi vysoká, aplikujeme na něj lineární diskriminační analýzu. Pomocí LDA je počet dimenzí zredukováno na deset a v této podobě je již příznakový vektor vložen do klasifikátoru.



**Obrázek 5.5:** Uspořádání trénovacích dat v prostoru dle 3 nejvýznamnějších dimenzí po použití LDA (ve skutečnosti jsou data v 10 dimenzovém prostoru)

## 5.4 Rozpoznání

Po získání příznakového vektoru, je možné přistoupit k rozpoznání znaku. Do zvolené metody k-NN jsou vkládány vždy příznakové vektory trénovacích dat a příznakový vektor testovaného znaku. Pro výpočet vzdálenosti mezi jednotlivými příznakovými vektory je nejvhodnější užít Eukleidovské metriky. Experimentálně bylo určeno  $k = 5$ .

## 5.5 Možnost odmítnutí

Při implementaci klasifikátoru je velmi důležité zahrnout taktéž možnost odmítnutí nebo-li takzvanou **reject option**. Jelikož vstup zadává uživatel, tak může docházet k chybám. Vzhledem k povaze aplikace, ve které bude rozpoznávač použit, je velmi nežádoucí, aby docházelo k zařazení do nesprávné třídy. Pro uživatele by takovéto zařazení znamenalo značnou nepříjemnost, což by vedlo k nespokojenosti s aplikací.

Pro vytvoření reject option lze využít 3 následující podmínky:

1. Znak má méně než 5 tahů
2. Nejbližší soused se nachází maximálně ve vzdálenosti  $d$

3. Mezi 'k' nejbližšími sousedy jsou maximálně 2 různé klasifikační skupiny.

Všechny tyto podmínky musejí být testovaným znakem splněny, jinak bude odmítnut. První podmínka byla určena za předpokladu, že pro napsání jakékoliv číslice či velkého písmena se neuzívá více než 4 tahů. Pro použití druhé podmínky musí být vykonáno experimentální měření, při kterém bude vybrána nevhodnější vzdálenost tak, aby byl dosažen co nejlepší poměr hodnot precize a senzitivity. Třetí podmínka zohledňuje situaci, kdy se klasifikovaný znak vyskytne na pomezí více tříd a je tedy vhodnější tento znak odmítnout než nesprávně klasifikovat.





# Kapitola 6

## Implementace

V této kapitole budou postupně popsány implementace tří aplikací, které byly vytvořeny v průběhu vývoje rozpoznávače číslic. První z nich je implementována v Android Studiu a slouží ke sběru dat, druhá je vytvořená v Matlabu a představuje klasifikátor společně s vyhodnocením jeho úspěšnosti a poslední aplikací je samotný rozpoznávač znaků pro Android zařízení. Všechny tyto aplikace lze nalézt na příloženém CD. Seznam obsahu CD je uvedeno v příloze D.

### 6.1 Použité nástroje

Během implementace byly použity následující nástroje:

- MATLAB 2017b

MATLAB je software, který poskytuje vhodné prostředí pro vědeckotechnické výpočty, vizualizaci či vytváření algoritmů a skriptů, počítačovou simulaci i vytváření aplikací s uživatelským rozhraním.

- Statistical Pattern Recognition Toolbox

Statistical Pattern Recognition Toolbox je sbírka metod pro statistické rozpoznávání znaků v Matlabu. Tento toolbox byl vyvinut na elektrotechnické fakultě ČVUT. [17]

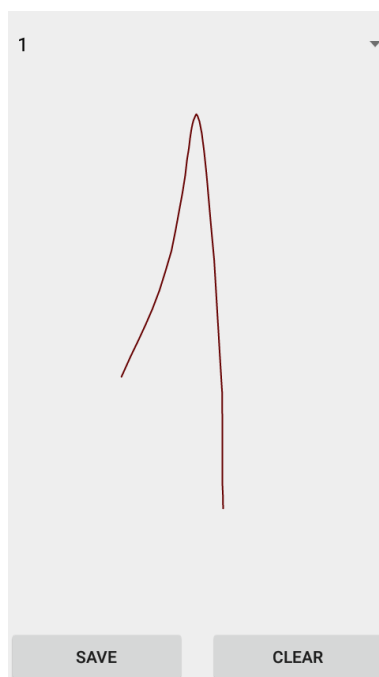
- Android Studio 2.3

Android Studio je oficiální vývojové prostředí pro vytváření aplikací, jež jsou určeny k použití na zařízeních s operačním systémem Android.

- OpenCV 3.4.1 OpenCV je open source knihovna, která obsahuje více než 2500 optimalizovaných klasických i nejmodernějších algoritmů pro počítačové vidění a strojové učení. OpenCV je možné využívat z prostředí jazyka C, C++, Java, Python či Matlab.[10] Tato knihovna byla zvolena, jelikož práce s maticemi z OpenCV je velmi podobná jako v Matlabu, což umožňuje jednodušší přesun kódu do Android studia.

## 6.2 Aplikace pro sběr dat

Ještě před samotnou implementací rozpoznávače znaků je velmi důležité mít nachystána data. Pro sběr dat byla nejprve vytvořena jednoduchá aplikace v Android Studiu, která zajišťuje ukládání právě napsaného znaku do textového souboru.



Obrázek 6.1: Aplikace pro sběr dat

Aplikace je složena pouze ze čtyř komponent a to ze spinneru, dvou tlačítek a Draw view. Selekcí položky ze spinneru dobrovolník vybere jaký znak bude zapisovat. Vybraný znak následně napíše na displej do draw view.

Při přejíždění prstu po displeji jsou do ArrayListu zaznamenávány souřadnice a akce (dotknutí se displeje, pohyb po displeji nebo zvednutí prstu z displeje), zároveň je na displej vykreslena trajektorie ručně zapsaného znaku. Po nakreslení znaku, je možné ho pomocí tlačítka 'SAVE' uložit nebo v případě, že byl znak napsán chybně, je možné ho smazat tlačítkem 'CLEAR'. Aby mohly být znaky znovu načteny a dále zpracovány, je důležité zvolit vhodný způsob zápisu. Pro každou třídu charakterů je připraven jeden textový soubor. Pomocí třídy FileWriter je každý jednotlivý znak do souboru zapsán jako posloupnost souřadnic a akcí, kdy každá dvojice souřadnic s akcí jsou zapsány na samostatný řádek. Po zapsání všech souřadnic a akcí jednoho vzorku je přidán řádek se 'zavináčem' (@), který odděluje od sebe v souboru jednotlivé znaky.

Pro sběr dat bylo osloveno **30** dobrovolníků z **12** zemí světa, přehled zastoupení jednotlivých národností je zaznamenáno v tabulce 6.1.

Země	Počet dobrovolníků
Česká republika	8
Itálie	8
Maďarsko	1
Německo	2
Polsko	2
Rakousko	2
Rumunsko	1
Řecko	1
Slovensko	2
Španělsko	1
Švýcarsko	1
USA	1
<b>celkem:</b>	<b>30</b>

**Tabulka 6.1:** Přehled zastoupení národností mezi dobrovolníky

Počet vzorků pro každé číslo není stejný, jelikož bylo dobrovolníky často vloženo jiné množství vzorků než bylo požadováno, a také protože některé znaky byly nečitelné či s chybou a musely být smazány.

Znak	Počet vzorků v trénovací sadě	Počet vzorků v testovací sadě
<b>0</b>	284	100
<b>1</b>	253	100
<b>2</b>	273	99
<b>3</b>	293	100
<b>4</b>	297	100
<b>5</b>	298	100
<b>6</b>	280	100
<b>7</b>	301	100
<b>8</b>	263	100
<b>9</b>	276	100
<b>Chybný znak</b>	0	107

**Tabulka 6.2:** Počet vzorků v datové sadě

Získané vzorky byly rozřazeny do jednotlivých složek, kdy ve složce 0 můžeme najít všechna testovací data pořízená za pomoci 11 dobrovolníků a ve složkách 1-19, data trénovací, kdy autorem vzorků je pro každou složku vždy jiná osoba.

Jelikož při funkci zrychleného volání zadává vstupní znak uživatel, díky čemuž dochází k chybám, je důležité, aby byl klasifikátor schopen detekovat,

zda byl zadán symbol, jež patří do jedné z klasifikačních tříd, či byl znak na vstupu chybný. Pro otestování této vlastnosti klasifikátoru, bylo potřeba shromáždit také vzorky s chybnými znaky, které nepatří do žádné z klasifikačních tříd a budou přidány k testovací sadě dat.



**Obrázek 6.2:** Příklady testovacích znaků, které nepatří do žádné třídy.

### 6.3 Realizace rozpoznávače v Matlabu

Pro realizaci rozpoznávače bylo zvoleno prostředí MATLAB, jelikož je vhodné pro výpočty s maticemi, vizualizaci dat a také jelikož skriptovací prostředí vytváří prostor pro rychlé a snadné testování algoritmů. Vytvořená aplikace lze spustit skriptem 'start.m'. Prvním krokem při implementaci rozpoznávače bylo načtení získaných dat z textových souborů. Načtení zajišťuje vytvořená funkce 'loadData(path)', která načítá jednotlivé hodnoty ze souboru do matic. Po té co jsou data připravena v maticích, je pro jejich další zpracování použita funkce 'raster(data, height, width)', která vrací počet tahů a binární masku, jež vytvoří z načtených dat za pomoci Bresenhamova algoritmu, o zadané výšce a šířce (šířka  $i$  a výška znaku je přeskálována na rozměry bitmapy).

Následně je implementován Bresenhamův algoritmus, který umožňuje převod znaku z vektorové formy na binární masku. Tato binární maska je následně použitím funkce 'imgaussfilt(M,  $\sigma$ )', kde je směrodatná odchylka  $\sigma = 0,7$  a  $M$  je binární maska, rozmazána. Pro zredukování dimenzí příznakových vektorů byly použity funkce PCA a LDA z knihovny STPRtool. Z této knihovny byla použita také funkce k-NN. V Matlabu byly také implementovány výpočty pro ověření funkčnosti a úspěšnosti klasifikátoru.

### 6.4 Prototyp rozpoznávací aplikace

Poté, co byl vytvořen a otestován rozpoznávač v Matlabu, je potřeba integrovat stejný rozpoznávač do mobilní aplikace. Jelikož kmenová aplikace funguje na operačním systému Android, bylo nejvhodnější uskutečnit implementaci

rozpoznávací aplikace pomocí vývojového prostředí Android Studio. Pro usnadnění práce s maticemi a pro algoritmus k-NN je použita knihovna OpenCV.

Aplikace se skládá pouze z jednoho view, na které se zaznamenává ručně psaný znak. Po té co je znak zapsán, je zavolána metoda `classify(data)`, třídy `Classifier`, která zaznamenaná data zpracovává stejným způsobem jako rozpoznávač v Matlabu, přičemž jsou extrahovány příznakové vektory. Pro redukci dimenzí příznakových vektorů je zapotřebí nahrát do aplikace modelové matice pro PCA a LDA, které byly již vytvořeny v Matlabu. Tyto matice jsou uloženy ve formátu bin, aby jejich velikost byla co možná nejmenší a nahrají se ze souborů do paměti aplikace vždy po jejím spuštění. Mimo to je také potřeba nahrát trénovací sady dat v již zredukované formě a ty jsou použity s algoritmem k-NN. Výsledek klasifikace je přehrán pomocí hlasového komentáře a také zobrazen na displeji.



# Kapitola 7

## Vyhodnocení klasifikátoru

### 7.0.1 Testování úspěšnosti

Po vytvoření klasifikátoru je nutné si ověřit také jeho funkčnost a úspěšnost klasifikace. Klasifikátor je testován na testovací sadě u které jsou třídy jednotlivých příkladů známé. Množina tříd je  $C = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, x$ , kde  $x$  reprezentuje možnost odmítnutí.

Ze získaných dat byla v poměru 1:3 oddělena data testovací od dat trénovacích. Data byla rozdělena tak, aby vzorky znaků v sadě testovací byly získány od jiných osob než znaky v datech trénovacích.

Každý znak z testovací sady je klasifikován klasifikátorem a výstup klasifikátoru je porovnán se známou anotací. Na základě toho můžeme vyhodnotit následující výkonnostní metriky klasifikátoru:

1. Procentuální úspěšnost klasifikace
2. Matice záměn(Confusion matrix)
3. TP, FP, TN, FN vzhledem k reject option
4. ROC křivka

### 7.0.2 Procentuální úspěšnost klasifikace

Procentuální úspěšnost klasifikace je dána vztahem

$$\text{Úspěšnost} = \frac{\text{počet správně klasifikovaných znaků}}{\text{počet všech klasifikovaných znaků}} \cdot 100 = 96,6\%$$

Tato úspěšnost je získána z výsledků klasifikátoru bez použití reject option. Tento způsob určení úspěšnosti je velmi jednoduchý a samotný je nedostačující.

### 7.0.3 Matice záměn (Confusion matrix)

Matice záměn znázorňuje klasifikaci jednotlivých dat ze všech klasifikačních tříd. Řádky představují jednotlivé skupiny testovacích vzorků a sloupce jejich přiřazení do tříd.

-	0	1	2	3	4	5	6	7	8	9	x
0	93	0	0	0	0	0	1	0	0	0	6
1	0	94	0	1	0	0	0	0	0	0	5
2	0	0	92	1	0	0	0	0	0	0	6
3	0	0	0	94	0	0	0	1	1	0	4
4	0	0	0	0	88	0	0	2	0	0	10
5	0	0	0	1	1	86	0	0	0	0	12
6	0	0	0	0	1	0	94	0	0	0	5
7	0	0	0	1	3	0	0	90	0	1	5
8	0	0	0	0	0	0	0	0	91	0	9
9	2	0	0	0	0	0	0	0	0	82	16
x	0	0	0	0	7	0	0	0	2	1	97

Tabulka 7.1: Matice záměn

V tabulce 7.1 jež znázorňuje matici záměn lze najít výsledky pro klasifikátor s reject option 5.5.

#### 7.0.4 TP, TN, FP, FN

Chceme-li detailnější analýzu klasifikátoru, je pouhý výpočet procentuální přesnosti již nedostačující a je zapotřebí zavést další metriky, které upřesní úspěšnost klasifikátoru. V prediktivní analýze je možné se setkat se speciálním typem matice záměn, která má dva řádky a dva sloupce. Uvažujme, že je klasifikace binární, a tedy každý objekt může být zařazen do třídy pozitivní(P) nebo negativní(N). V našem případě, je znak pozitivní tehdy, když patří do třídy 0-9 a negativní jsou znaky třídy 'X'. Po té, co jsou testovací znaky ohodnoceny klasifikátorem, mohou být příklady z testovací sady rozřazeny do jedné ze čtyř skupin.

		Skutečná třída	
		pozitivní	negativní
Klasifikovaná třída	pozitivní	Skutečně pozitivní True positive TP	Falešně pozitivní False positive FP
	negativní	Skutečně negativní True negative TN	Falešně negativní False negative FN

Obrázek 7.1: Tabulka záměn

- TP Zde patří znaky ze tříd 0-9, které byly zařazeny klasifikátorem do třídy, ke které patří.



TP	FP
904	27
TN	FN
97	78

**Tabulka 7.2:** Hodnoty pro TP, TN, FP a FN při použití našeho klasifikátoru

- TN Zde patří znaky ze třídy 'X', jež byly klasifikovány jako 'X'.
- FP Do této skupiny patří všechny znaky, které byly klasifikátorem zařazeny do jedné z pozitivních tříd, ale doopravdy do této třídy nepatří.
- FN V této skupině jsou testovací příklady za skupin 0-9, které byly ohodnoceny jako negativní, ale ve skutečnosti jsou pozitivní.

Díky těmto hodnotám, lze dopočítat další statistické hodnoty. V tabulce 7.3 jsou uvedeny vzorce pro výpočet a výsledky pro náš klasifikátor.

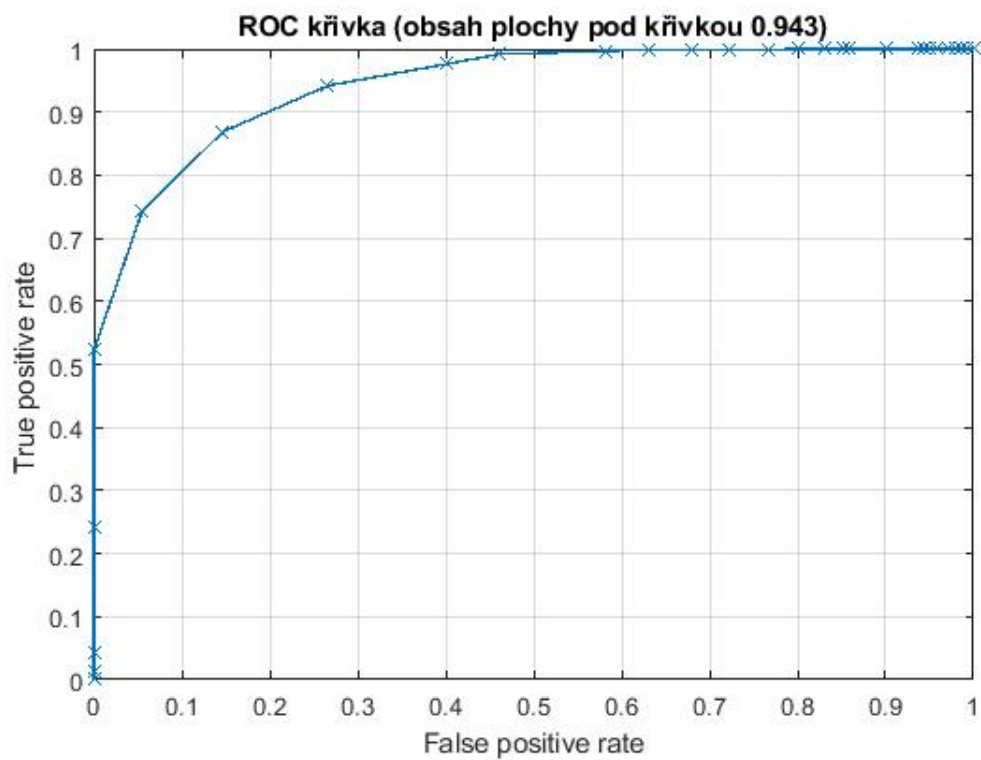
Sensitivita = sensitivity (recall, true positive rate)	$\frac{TP}{TP+FN}$	0,9206
Specificita = specificity (true negative rate)	$\frac{TN}{TN+FP}$	0,7823
Precize = precision (positive predictive value)	$\frac{TP}{TP+FP}$	0,9710
Míra falešně pozitivních = false positive rate (fall-out)	$\frac{FP}{FP+TN}$	0,2177
Přesnost = accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	0,9051

**Tabulka 7.3:** Tabulka měření výkonnosti klasifikátoru

### 7.0.5 ROC křivka

ROC křivka je nástrojem, který slouží k hodnocení a optimalizaci binárního klasifikátoru. Tato křivka znázorňuje vztah mezi 'True positive rate' (míra skutečně pozitivních) a 'False positive rate' (míra falešně negativních).

Jedním z parametrů, který rozhoduje o tom, jaký znak bude odmítnut a tím pádem zařazen mezi 'negativní', je vzdálenost prvního nejbližšího souseda. Tato vzdálenost tedy bude dělicím bodem  $\theta$ . Při změnách jeho hodnoty, se mění také hodnoty specificity a senzitivity, díky čemuž pro každé  $\theta \in \mathfrak{R}$  dostaneme právě jeden bod na křivce ROC. [14] ROC křivka prochází vždy od počátečního bodu [0,0] do bodu [1,1] a je neklesající. Uvažovali bychom náhodný klasifikátor, tak by jeho ROC křivka měla podobu diagonály, oproti tomu pokud by byl klasifikátor ideální a zařadil by všechny objekty správně, ROC křivka by vedla z bodu [0,0] vzhůru do [0,1] a následně do koncového bodu [1,1] a obsah pod ní by byl roven jedné. Lze tedy říci, že čím více se obsah ROC křivky blíží jedné, tím je klasifikátor spolehlivější. [18]



**Obrázek 7.2:** ROC křivka

Z této křivky lze tedy vyčíst, že klasifikátor je kvalitní. S pomocí ROC křivky byla také stanovena vzdálenost  $d$  pro 2. podmínku možnosti odmítnutí 5.5 a to  $d = 0,67$ .

# Kapitola 8

## Uživatelské testování

Po vytvoření mobilní aplikace přichází na řadu uživatelské testování, které je zásadní částí při vývoji jakékoliv aplikace. Právě při testování na cílové skupině je možno nejen odhalit nedostatky či další problémy, ale taktéž dostat velmi důležitou zpětnou vazbu, která je klíčová pro dokončení a doladění aplikace.

Průběh testování je rozdělen do tří elementárních částí:

1. Dotazník před testováním
2. Testovací úkoly
3. Dotazník po testování

Díky spolupráci se Sjednocenou organizací nevidomých a slabozrakých České republiky (SONS) bylo možné se setkat se skupinou zrakově postižených dobrovolníků, kteří následně prošli testováním. Další část dobrovolníků, kteří se taktéž podíleli na testování, byly již osoby bez zrakového postižení z různých sociálních skupin.

### 8.1 Dotazník před testováním

1. Jaký je Váš věk?
  - a. 0-20
  - b. 20-40
  - c. 40-60
  - d. 60 a více
2. Jaké je Vaše pohlaví?
  - a. muž
  - b. žena
3. Viděl jste někdy během svého života?
  - a. ano

- b. ne
4. Specifikujte prosím Váš stav zraku.
- a. zdravý zrak bez optické korekce
  - b. zdravý zrak s optickou korekcí
  - c. středně slabý zrak
  - d. těžce slabý zrak – nemohu číst ani s optickou korekcí
  - e. praktická nevidomost - jsem schopen vnímat pouze světlo
  - f. úplná nevidomost
5. Jaký typ mobilního telefonu používáte?
- a. tlačítkový
  - b. dotykový
  - c. jiný
  - d. mobilní telefon nepoužívám
6. Znáte mobilní speciální mobilní telefon pro nevidomé s kmenovou aplikací?
- a. ano, ale nikdy jsem s ním nepracoval
  - b. ano, pracoval jsem s ním, ale nezaujal mě nebo není určen pro mě
  - c. ano a používám ho
  - d. ne, nikdy jsem o něm neslyšel
7. Znáte funkci zrychleného volání?
- a. ano
  - b. ne
8. Myslíte, že byste tuto službu na svém telefonu využíval?
- a. ano
  - b. ne
  - c. nevím/možná
9. Kolika kontaktům voláte častěji než 3x týdně?
- a. 0
  - b. 1-2
  - c. 3-5
  - d. 6-10
  - e. více než 10

10. Pro kolik kontaktů byste využíval funkci zrychleného volání?

- a. 0
- b. 1 – 2
- c. 3-5
- d. 6 - 10
- e. 11 – 15
- f. více než 15

## 8.2 Scénář testování

Po vyplnění dotazníku bude dobrovolník seznámen se speciálním telefonem s kmenovou aplikací a bude mu vysvětlen princip funkce zrychleného volání. Poté bude mít za úkol splnit následující úkol.

1. Účastník testování je vyzván k psaní číslic 0 – 9 na displej mobilního telefonu, přičemž každou číslici zapisuje právě pětkrát. Během tohoto testu je zaznamenávána úspěšnost rozpoznání jednotlivých číslic.
2. Dobrovolník zapíše 5 krát znak, který nepatří do žádné z možných tříd.

## 8.3 Dotazník po testování

1. Líbí se Vám funkce zrychleného volání?
  - a. líbí
  - b. líbí, ale dala by se ještě vylepšit
  - c. nelíbí
2. Pamatoval byste si pod kterými čísly máte uložené jednotlivé kontakty?
  - a. ano
  - b. ne
3. Myslíte si, že byste tuto funkci na svém telefonu využíval?
  - a. ano
  - b. jen v případě vyšší úspěšnosti rozpoznávaných čísel
  - c. ne
4. V případě, že byste chtěl tuto funkci využívat, stačilo by Vám 10 dostupných znaků?
  - a. ano
  - b. ne

5. Je pro Vás pohodlnější a rychlejší uskutečnit hovor normálně či pomocí ručně napsaného znaku?
- a. normálně
  - b. pomocí ručně psaného znaku
6. Máte nějaké další náměty či připomínky, které by mohly přispět k zlepšení této funkce?

## 8.4 Výsledky testování

Testování se zúčastnilo 10 osob, z čehož tři ženy a dva muži byli nevidomí či těžce slabozrací a zbylí čtyři muži a jedna žena byli bez jakéhokoliv zrakového postižení.

Všem účastníkům se nápad užití rozpoznávače pro funkci zrychleného volání velmi líbil a všechny osoby se zrakovým postižením by tuto funkci využívalo, jelikož by jim umožnila pohodlnější a rychlejší volání. Jeden dobrovolník uvedl, že by tuto funkci využil v situacích, kdy není připojen k internetu a nemá příležitost ovládat svůj telefon hlasem či v případě, že je v okolí velký hluk a špatně slyší hlasové komentáře. Všechny osoby také uvedly, že by jim stačilo pro zrychlené volání maximálně 10 kontaktů.

Ze zrakově postižených osob byl jeden muž nevidomý již od narození a právě tento muž měl zprvu problémy se zapisováním číslic, jelikož tyto znaky nikdy neviděl ani nepoužíval, avšak po názorné "ukázce" jak se píše, byl schopen úspěšně některé z nich zapsat a také uvedl, že by neměl problém se je naučit. Z tohoto poznatku lze usuzovat, že pro osoby, které jsou nevidomé již od narození, by mohl být problém tuto funkci používat.

Jednotlivé odpovědi na otázky jsou v příloze B.

-	0	1	2	3	4	5	6	7	8	9	X
0	46	0	0	1	0	0	2	0	0	0	1
1	0	46	1	2	1	0	0	0	0	0	0
2	1	0	48	0	1	0	0	0	0	0	0
3	2	0	1	43	1	0	0	0	0	2	1
4	0	0	0	0	50	0	0	0	0	0	0
5	0	0	0	0	0	44	3	1	1	1	0
6	1	0	1	0	0	0	48	0	0	0	0
7	1	0	2	0	0	0	0	47	0	0	0
8	0	0	1	0	0	0	1	0	47	1	0
9	1	2	0	3	0	0	0	0	1	40	3
X	1	3	1	1	6	2	0	1	0	0	35

Tabulka 8.1: Matice záměn

TP	FP
459	51
TN	FN
35	5

**Tabulka 8.2:** Hodnoty TP, TN, FP a FN dosažené během testování

V tabulce 8.4 jsou vypočteny hodnoty pro další ukazatele jež hodnotí výsledky testu.

Recall	0,989
Specificity	0,625
Precision	0,9
False positive rate	0,59
Accuracy	0,8981

Dále byla vypočítána úspěšnost rozpoznání (bez měření možnosti odmítnutí) pro jednotlivé osoby.

Dobrovolníci	1	2	3	4	5	6	7	8	9	10
Úspěšnost	0,78	1	1	0,96	0,76	0,94	1	0,84	0,92	0,9

**Tabulka 8.3:** Úspěšnost rozpoznání znaků pro jednotlivé účastníky testování

Z těchto výsledků je patrné, že osoby nevidomé měly nejhorší výsledky, zatímco těžce slabozrací měli lepší výsledky než většina dobrovolníků bez zrakového postižení.





## Kapitola 9

### Rozšíření o písmena velké abecedy

Chceme-li aby klasifikátor rozpoznával také velká písmena, není potřeba měnit výše popsany postup rozpoznávání.

Pro přidání znaku do aplikace, je pouze potřeba nahrát tyto znaky ve vytvořené aplikaci v MATLABU a vygenerovat nové příznakové soubory s vektory a modely pro PCA i LDA, které by byly následně vloženy místo stávajících modelů do odpovídající složky rozpoznávací aplikace. Je zřejmé, že s každým přidaným znakem bude úspěšnost klasifikátoru klesat. Jelikož při testování všechny osoby uvedly, že je pro ně možnosti uložit si až 10 kontaktů pro tuto funkci dostačující, nebylo přistoupeno k přidání dalších znaků do aplikace. Jelikož pro nevidomé osoby je psaní běžných čísel a písmen občas problematické, lze předpokládat, že by docházelo k častým záměnám mezi podobnými znaky.



# Kapitola 10

## Závěr

Cílem této práce bylo vytvořit rozpoznávač znaků, jež bude využíván v aplikaci pro zrychlené volání, která bude určena pro osoby se zrakovým postižením. Během procesu vytváření této aplikace byly implementovány další dvě pomocné aplikace, z čehož jedna byla pro sběr trénovacích dat, jež jsou nezbytné pro rozpoznávání, a druhá byla vytvořena v Matlabu kvůli možnosti vizualizace dat a práce s maticemi. Při implementaci byla pro klasifikaci použita metoda k-NN, která má i přes svou jednoduchost uspokojivé výsledky.

Po zhotovení byla mobilní aplikace testována a všechny osoby z cílové skupiny jí označili za užitečnou, přičemž by jí sami rádi používali. Velká písmena již nebyla přidána do trénovací sady dat, jelikož nebylo dostatek dobrovolníků, kteří by byli ochotni poskytnout takové množství znaků, aby mohly být použity pro trénování a testování klasifikátoru, a proto nejsou zahrnuty v aplikaci. Avšak všichni respondenti uvedli, že 10 znaků pro funkci zrychleného volání je více než dostačující a při tomto počtu by neměli problém se zapamatováním si, které znaky jsou přiřazeny jakému kontaktu.

### 10.1 Budoucí práce

Přestože úspěšnost rozpoznávače je přijatelná, daly by se podniknout další kroky k jeho zdokonalení. Pro lepší výsledky by bylo vhodné přidat nové příznaky do příznakového vektoru a rozšířit trénovací sady dat. Také možnost, která by uživateli umožnila přidat do datových sad své vlastní vzorky, by navýšila úspěšnost a mohla by umožnit, aby i lidé s velmi charakteristickým rukopisem mohli používat tuto aplikaci. V případě, že by bylo potřeba rozšířit datovou sadu o nové vzorky, bylo by vhodné do aplikace pro sběr dat přidat počítadlo, které by pomohlo k přesnějšímu zadávání znaků. Také by mohla být přidána funkce, která by vytvořila zvláštní složku pro vzorky každé jednotlivé osoby, což by přispělo k zpřehlednění dat a s daty by se dalo následně lépe nakládat.

Během uživatelského testování aplikace byly navrženy další zajímavé potenciální způsoby využití tohoto rozpoznávače. Dotazovaní by tento rozpoznávač rádi používali také pro zrychlený přechod mezi položkami v menu speciálního mobilního telefonu s kmenovým systémem nebo namísto číselné klávesnice v

případě volání s hlasovým automatem, který po volajícím vyžaduje vkládání čísel.



## Příloha A

### Použitá cizí slova a zkratky

---

confusion matrix	matice záměn
launcher	aplikace, která upravuje vzhled a funkce systému
LDA	Linear Discriminant Analysis
naive Bayes	naivní Bayesovský klasifikátor
OpenCV	Open Source Computer Vision Library
PCA	Principal Component Analysis
postprocessing	následné zpracování dat
preprocessing	předběžné zpracování dat
reject option	možnost odmítnutí

---



## Příloha B

### Výsledky testování

#### B.1 Dotazník před testováním

-	1	2	3	4	5	6	7	8	9	10
Věk	63	54	38	48	53	23	28	24	27	23
Pohlaví	muž	žena	žena	žena	muž	muž	muž	muž	muž	žena
Viděl jste někdy během svého života?	a	a	a	a	b	a	a	a	a	a
Specifikujte prosím Váš stav zraku.	f	e	e	f	f	a	a	a	a	a
Jaký typ mobilního telefonu používáte?	b	b	b	b	b	b	b	b	b	b
Znáte mobilní speciální mobilní telefon pro nevidomé s kmenovou aplikací?	c	a	b	c	b	d	d	d	b	a
Znáte funkci zrychleného volání?	b	a	a	b	a	b	a	a	a	a
Myslíte, že byste tuto službu na svém telefonu využíval?	c	a	a	a	a	c	a	a	b	b
Kolika kontaktům voláte častěji než 3x týdně?	c	d	c	c	c	b	e	e	a	b
Pro kolik kontaktů byste využíval funkci zrychleného volání?	c	c	c	d	c	b	d	c	a	a

**Tabulka B.1:** Odpovědi testovaných dobrovolníků na dotazník před testováním.

#### B.2 Testování úspěšnosti rozpoznávače

Tabulka rozpoznávaných čísel během testování. Každý sloupec tvoří výsledky jednoho dobrovolníka, který zapisoval čísla 0-9 právě 5 krát a čísla v buňkách tedy vyjadřují výsledek klasifikátoru.

-	1	2	3	4	5	6	7	8	9	10
0	x0000	00000	00000	00000	60000	00000	00000	36000	00000	00000
1	11111	11111	11111	11111	11111	11111	11111	41111	11111	13123
2	22222	22222	22222	22222	20242	22222	22222	22222	22222	22222
3	33303	33333	33333	33333	39033	33393	33333	33433	33x32	33333
4	44444	44444	44444	44444	44444	44444	44444	44444	44444	44444
5	75555	55555	55555	55555	56665	55555	55555	55555	55555	55958
6	66666	66666	66666	66666	66666	66626	66666	66666	66606	66666
7	37777	77777	77777	73777	17777	77777	77777	77777	77777	77777
8	88888	88888	88888	88888	88888	88888	88888	28988	68888	88888
9	09xx9	99999	99999	9999x	99411	49999	99999	98994	99999	99999
x	53xxx	xx4x5	xxx7x	1x2xx	xx4xx	x4xxx-	xxx4x	x4xxx	xx41x	x01xx

Tabulka B.2: Přehled rozpoznávaných znaků

### B.3 Dotazník po testování

-	1	2	3	4	5	6	7	8	9	10
Líbí se Vám funkce zrychleného volání?	a	a	a	a	a	a	b	b	b	a
Zapamatoval byste si ke kterému kontaktu máte uložený jaký kontakt?	a	a	a	a	a	a	a	b	a	a
Myslíte si, že byste tuto funkci na svém telefonu využíval?	a	a	a	a	a	c	a	b	c	c
Je pro Vás pohodlnější a rychlejší uskutečnit hovor normálně či pomocí ručně napsaného znaku?	a	a	a	a	a	a	a	a	a	-
Je pro Vás pohodlnější a rychlejší uskutečnit hovor normálně či pomocí ručně napsaného znaku?	b	b	b	b	b	a	b	b	a	a
Máte nějaké další náměty či připomínky, které by mohly přispět k zlepšení této funkce?	-	-	-	-	-	-	-	-	-	-

Tabulka B.3: Tabulka s odpověďmi na dotazník po testování

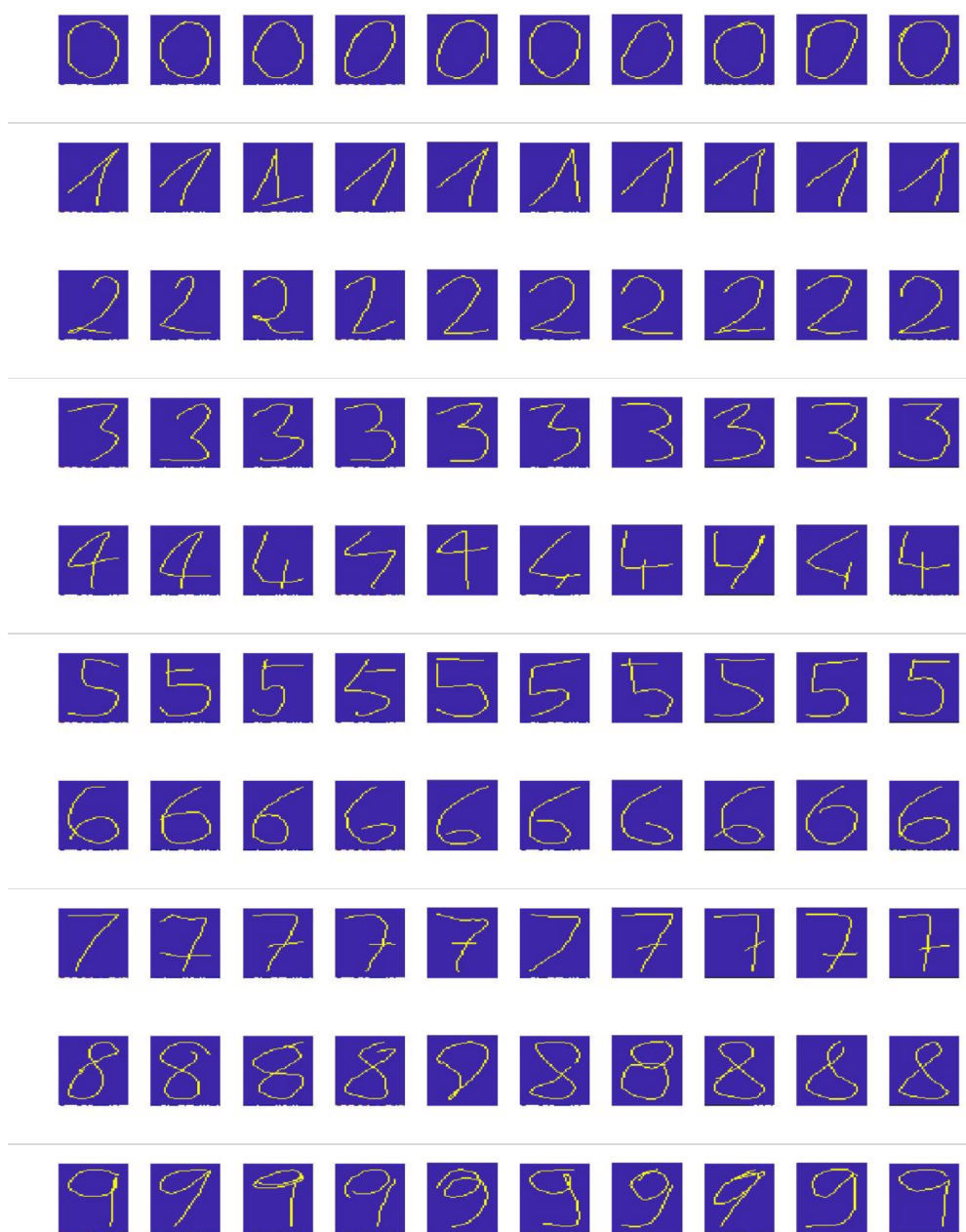




## Příloha C

### Grafická ukázka sady znaků

V této části jsou graficky znázorněny znaky 0-9, jež se nachází v datové sadě, která je použita pro rozpoznávání.



Obrázek C.1: Ukázka znaků ze sady



## Příloha D

### Obsah přiloženého CD

BPuhliarikovaeva.pdf	Tento text
sberdat	Složka s Android aplikací pro sběr dat včetně zdrojových kódů
matlab	Složka s implementovaným rozpoznávačem v Matlabu (spouští se skriptem Start
matlab/testd	Složka ve které se nacházejí složky se soubory, v nichž jsou zaznamenány testovací (složka 0) a trénovací data (složka 1-19)
gestures	Složka s projektem jež implementuje rozpoznávač jako mobilní aplikaci



## Příloha E

### Literatura

- [1] “Blindness, world health organization.” [Online]. Available: <http://www.who.int/blindness/en/>
- [2] P. Svobodník, “Zpřístupnění mobilních telefonů se systémem android pro nevidomé uživatele,” 2013.
- [3] Classifications. [Online]. Available: <http://apps.who.int/classifications/icd10/browse/2016/en#/H53-H54>
- [4] Vision accessibility - iphone. [Online]. Available: <https://www.apple.com/accessibility/iphone/vision/>
- [5] Android accessibility overview - android accessibility help. [Online]. Available: <https://support.google.com/accessibility/android/answer/6006564?hl=en>
- [6] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta, “Online and offline character recognition: A survey,” pp. 0967–0970, April 2016.
- [7] Z. Noubigh and M. Kherallah, “A survey on handwriting recognition based on the trajectory recovery technique,” pp. 69–73, April 2017.
- [8] W. Guerfali and R. Plamondon, “Normalizing and restoring on-line handwriting,” *Pattern Recognition*, vol. 26, no. 3, pp. 419 – 431, 1993. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/003132039390169W>
- [9] M. Levin, “Cellwriter: Grid-entry handwriting recognition,” 2007.
- [10] “statistic.” [Online]. Available: <https://cmp.felk.cvut.cz/cmp/software/stprtool/stprtool.pdf>
- [11] P. M. B. Vitányi, “Information distance in multiples,” *CoRR*, vol. abs/0905.3347, 2009. [Online]. Available: <http://arxiv.org/abs/0905.3347>
- [12] C. J. Burges *et al.*, “Dimension reduction: A guided tour,” *Foundations and Trends® in Machine Learning*, vol. 2, no. 4, pp. 275–365, 2010.
- [13] P. H. Richar Duda and D. Stork, “Pattern classification,” 2001.

- [14] C. H. Chen, *Handbook of pattern recognition and computer vision*, 4th ed. Singapore: Imperial College Press, c2010.
- [15] S. B.Szocs and J.Robu, “D.a.c draw and calc - the intuitive calculator,” pp. 157 – 163, 2012.
- [16] V. B.K.Iwana and S.Uchinda, “A robust dissimilarity-based neural network for temporal pattern recognition,” pp. 265 – 270, 2016.
- [17] “Statistical pattern recognition toolbox for matlab.” [Online]. Available: <https://cmp.felk.cvut.cz/cmp/software/stprtool/stprtool.pdf>
- [18] Z. Bortlíček, “Roc křivky,” 2008. [Online]. Available: [https://is.muni.cz/th/106210/prif\\_m/Diplomova\\_prace.pdf](https://is.muni.cz/th/106210/prif_m/Diplomova_prace.pdf)