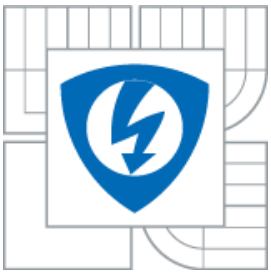
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS

VYHODNOCENÍ TESTOVÝCH FORMULÁŘŮ POMOCÍ OCR

TEST FORM EVALUATION BY OCR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

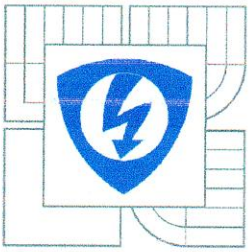
AUTOR PRÁCE — BC. PETR NOGHE

AUTHOR

VEDOUCÍ PRÁCE — ING. ONDŘEJ KALLER

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student:	Bc. Petr Noghe	***ID:***	119551
Ročník:	2	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Vyhodnocení testových formulářů pomocí OCR

POKYNY PRO VYPRACOVÁNÍ:

Seznamte s principy zpracování 2D obrazového signálu a konkrétními metodami rozpoznání psaného textu OCR (Optical Character Recognition).
Ve vhodném prostředí (Matlab, v opodstatněném případě jiné) navrhněte systém pro tvorbu, OCR a statistické vyhodnocení testových formulářů subjektivních testů kvality obrazu. Software by měl zvládat vyhodnotit jak testové otázky zjišťovací, tak i doplňovací.
Realizujte navržený SW. Proveďte jeho testování na statisticky významném vzorku testových formulářů, zjistěte míru chybovosti. Tuto míru případnými úpravami minimalizujte.

DOPORUČENÁ LITERATURA:

[1] MARQUES, O. Practical Image and Video Processing Using MATLAB. New Jersey: Wiley & Sons, 2011.

[2] HANSELMAN, D. LITTLEFIELD, B. Mastering MATLAB 7. New Jersey: Prentice Hall, 2004.

Termín zadání:	11.2.2013	***Termín odevzdání:***	24.5.2013

Vedoucí práce: Ing. Ondřej Kaller
Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Práce se zabývá vyhodnocováním formulářů pomocí optického rozpoznávání znaků v obraze. První část práce je věnována zpracování obrazu a používaným metodám pro OCR analýzu. V praktické části je vytvořena databáze vzorových znaků a z používaných metod je vybrána metoda založená na korelaci vzorů a rozpoznávaných znaků. Program pro návrh formulářů, jejich vyhodnocení a opravu špatně klasifikovaných znaků je navrhnut v grafickém prostředí programu MATLAB. Na závěr je vyhodnoceno několik formulářů a zjištěna míra úspěšnosti navrhnutého programu.

KLÍČOVÁ SLOVA

OCR, optické rozpoznání textu, zpracování obrazu, ručně psaný text, korelace, segmentace

ABSTRACT

This thesis deals with the evaluation forms using optical character recognition. Image processing and methods used for OCR is described in the first part of thesis. In the practical part is created database of sample characters. The chosen method is based on correlation between patterns and recognized characters. The program is designed in a graphical environment MATLAB. Finally, several forms are evaluated and success rate of the proposed program is detected.

KEYWORDS

OCR, optical character recognition, image processing, handwritten text, correlation, segmentation

NOGHE, P. *Vyhodnocení testových formulářů pomocí OCR*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2013. 57 s. Diplomová práce. Vedoucí práce: Ing. Ondřej Kaller.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma *Vyhodnocení testových formulářů pomocí OCR* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 24. května 2013 .....

podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Ondřeji Kallerovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále děkuji respondentům za vyplnění formulářů.

V Brně dne 24. května 2013 .....

podpis autora

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Práce se zabývá optickým rozpoznáním psaného textu, zkráceně OCR (z anglického Optical Character Recognition). Což je metoda umožňující digitalizaci tištěných textů, s nimiž pak lze pracovat jako s normálním počítačovým textem. OCR je používáno v aplikacích, kde se požaduje čitelnost dat lidským okem, při zpracování peněžních transakcí, v obchodních systémech, detekcí registračních značek vozidel, nebo při vyhodnocování dotazníků.

Cílem práce je seznámit se s principy zpracování 2D obrazového signálu a konkrétními metodami rozpoznání psaného textu OCR. V prostředí MATLAB navrhnout systém pro tvorbu a statistické vyhodnocení testových formulářů. Software by měl zvládat vyhodnotit jak testové otázky zjišťovací, tak i doplňovací. Navrhnutý program vyzkoušet na statisticky významném vzorku testových formulářů a zjistit míru chybovosti. Tuto míru případnými úpravami minimalizovat.

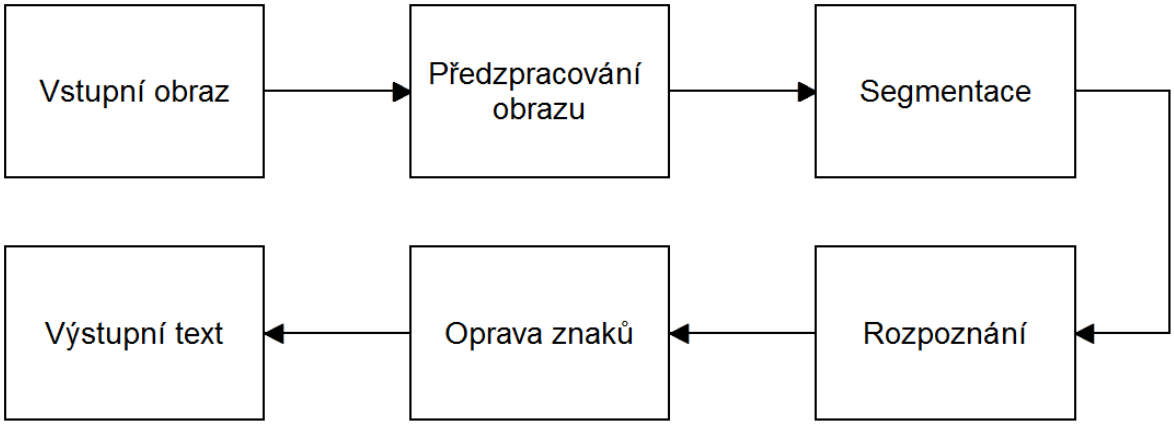
Tato práce je členěna do pěti základních částí. První kapitola je věnována zpracování obrazu před samotným použitím OCR. Používané metody OCR jsou popsány ve druhé části. V kapitole 3 je uveden popis, jak byla vytvořena databáze vzorových znaků. Kapitola 4 je zaměřena na návrh systému v grafickém prostředí MATLAB a popisu jeho funkčnosti. Testování formulářů a vyhodnocení úspěšnosti navrhnutého systému obsahuje kapitola 5. Celkové hodnocení je uvedeno v závěru.

1 ZPRACOVÁNÍ OBRAZU PRO OCR

Zpracování obrazu pro OCR analýzu je započato tím, že je pomocí skeneru sejmuta předloha. V další fázi je nejprve třeba provést rozdělení celistvých oblastí, neboť stránka může obsahovat bloky textu, obrázky nebo tabulky. Některé programy samy oddělují části textů od obrázků a zpracují i text rozdělený do sloupců. V této práci je nutná pomoc zvenčí. Při návrhu předlohy v programu je automaticky zaznamenána poloha bloků, které budou následně vyhodnocovány.

Potom co je vstupní obraz rozdělen, jsou nalezeny oblasti s textem. Předtím než jsou znaky rozpoznány, je nutné nejprve provést segmentaci což je rozdělení na jednotlivé znaky. Aby segmentace proběhla správně, musí být provedeno předzpracování obrazu. V bloku předzpracování jsou obsaženy tři nejdůležitější procesy: odstranění zašumění, prahování a natočení do správné polohy.

Text je rozdělen do řádků a řádky jsou rozděleny na samostatné znaky. Poté se může uplatnit rozpoznávací logika. Pro rozpoznání textu je důležité naučit program podle použité metody charakteristické rysy jednotlivých znaků. Rozpoznání je tedy proces porovnávání charakteristických rysů znaků zjištěných z neznámého textu s charakteristickými rysy vzorů jednotlivých znaků. Rozpoznání znaků nemůže být nikdy dokonalé, a proto je před výstup zařazen blok, který porovnává získané slova se slovy ze slovníku daného jazyka a opravuje chyby. Předpokládá se spoluúčast obsluhy, která musí dodat programu chybějící inteligenci. Například pokud je určité písmeno špatně rozpoznáváno, obsluha naučí program toto písmeno správně identifikovat. Jednotlivé kroky zpracování obrazu a následného rozpoznání jsou zobrazeny v diagramu na Obr. 1.1.

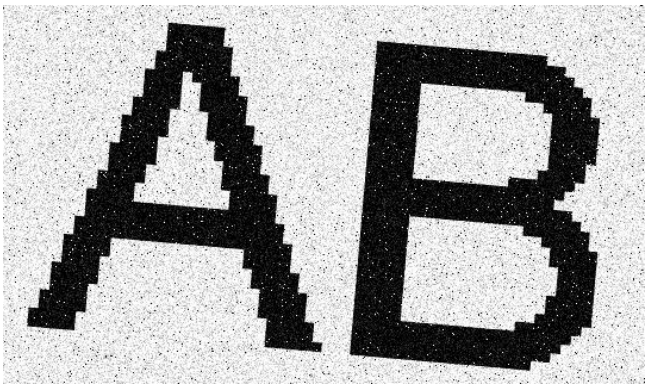


Obr. 1.1: Průběh zpracování obrazu pomocí OCR

1.1 Vstupní obraz

Pomocí skeneru je převedena předloha vstupního obrazu ze spojité analogové formy na papíře na digitální formu v obrazovém formátu. Digitální forma obrazu je reprezentována dvourozměrnou maticí, kde každý pixel má vlastní hodnotu jasu. Při 8 bitovém vyjádření je hodnota jasu v rozmezí od 0 do 255, pokud je obraz černobílý.

Na Obr. 1.2 je vidět příklad vstupního obrazu, který je pootočený a zatížený šumem. Vylepšení kvality obrazu má na starosti modul předzpracování obrazu, jehož účelem je potlačit kazy obrazu a zvýraznit důležité detaily.



Obr. 1.2: Obrázek s příkladem vstupního obrazu

1.2 Předzpracování obrazu

Předzpracování obrazu obsahuje tyto části. Omezení zašumění, prahování, natočení do správné polohy a binární morfologické operace. Hlavním cílem předzpracování je to, aby byl obraz co nejpřijatelnější pro segmentaci a následné rozdělení objektů do tříd.

Aby toto rozdělení bylo možné s prakticky využitelnou úspěšností, musí se od sebe třídy lišit ve znacích, podle kterých jsou rozpoznávány klasifikátorem. Je samozřejmé, že při velmi podobných znacích klasifikátor často nepozná, do jaké třídy objekt zařadit. Pro zvýraznění meziskupinových rozdílů, vyčištění dat od šumu a vypuštění případných nadbytečných informací slouží mimo jiné i předzpracování dat. Předzpracování má na výsledek klasifikace značný vliv. Ten může být pozitivní, jako je výše zmíněné zvýraznění meziskupinových rozdílů nebo i menší výpočetní náročnost, ale i negativní, ve smyslu zmenšení množství informací, která data obsahují [1].

1.2.1 Omezení šumu

Omezování šumu v obrazu spočívá na modelu šumu, který má nulovou střední hodnotu a jehož hodnoty v sousedních prvcích jsou nezávislé. Za předpokladu, že hodnoty jasu obrazu v prvcích pokrytých maskou jsou téměř konstantní, je možno získat lepší poměr signálu k šumu váhovaným průměrováním hodnot těchto sousedících prvků s výsledky odpovídajícími počtu průměrovaných hodnot a tedy velikosti masky [2].

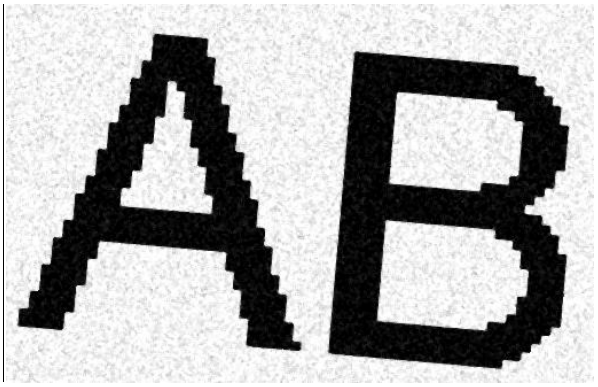
Bohužel předpoklad konstantní hodnoty signálu nebývá splněn, zejména v okolí hran obrazu, což vede ke zkreslení, které se v obrazu projevuje jako ztráta ostrosti. Bývá pak třeba hledat kompromis mezi potlačením šumu a rozostřením. Ovlivnění je možné zvolenou velikostí masky a různým váhováním prvků. Má-li střední prvek a prvky jemu nejbližší vyšší váhy, je rozostření menší, ovšem za cenu omezení funkce potlačení šumu. Následující masky rozměru 3 x 3 ukazují vývoj od rovnoměrné masky po masku omezující vliv vzdálenějších prvků [2]:

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}.$$

Tyto operátory se uplatňují při potlačování šumu, který postihuje všechny elementy obrazu stejnou měrou. V praxi se však vyskytuje i jiný typ rušení, tzv. šum typu „pepř a sůl“, který postihuje jen jednotlivé izolované body obrazu v důsledku například impulsního rušení při přenosu nebo při ztrátě obrazových bodů při pořizování obrazu CCD senzorem.

Obecným přístupem k tomuto typu rušení je detekovat chybný prvek a poté pro něj určit novou aproximativní hodnotu interpolací ze sousedních vzorků. Nejčastější je použití mediánového filtru, který setřídí prvky vstupního obrazu, pokryté maskou zvolené velikosti a za výstupní prvek dosadí prostřední prvek setříděné posloupnosti. Mediánový filtr je příkladem koncepčně jednoduchého lokálního nelineárního operátoru [2].

Nevýhodou je porušování tenkých čar a ostrých rohů. Tuto vlastnost lze redukovat použitím křížového okolí místo obdélníkového [3]. Použitím mediánového filtru dojde k vyhlazení obrazu, ale také může dojít ke ztrátě detailů. Je důležité zvolit optimální velikost matice, pro kterou bude medián počítán. Na příkladu níže (Obr. 1.3) byly rozměry masky použité pro mediánovou filtraci 3 x 3.



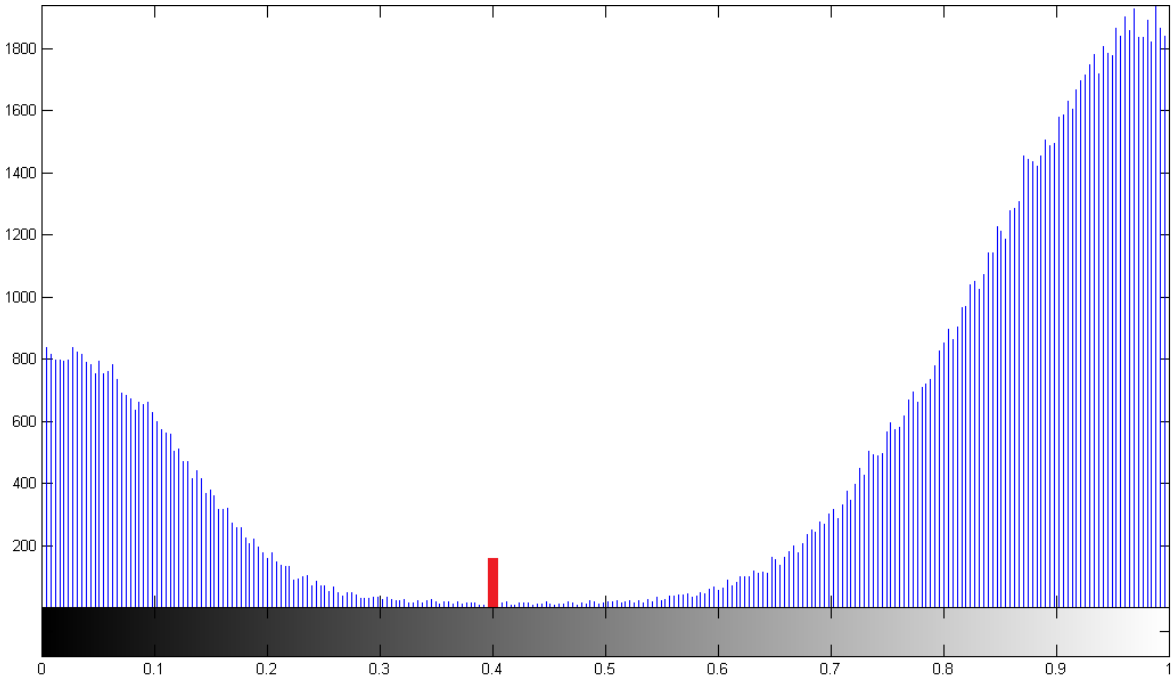
Obr. 1.3: Obrázek s příkladem po mediánové filtraci

1.2.2 **Prahování**

V procesu předzpracování obrazu je nejdůležitější operací prahování. Pomocí této operace je v tomto případě přeměněn vstupní šedotónový obraz s 8 bitovou reprezentací na výstupní obraz binární, který je reprezentován 1 bitem na pixel. Pro úspěšné prahování je nejdůležitější správná volba hodnoty prahu. Tato hodnota může být zvolena subjektivně nebo pomocí některé automatické metody.

Jako příklad automatické metody je zde uvedena metoda, která určuje hodnotu prahu z histogramu. Histogram je funkce, která přiřadí každé úrovni jasu nebo barvy odpovídající četnost příslušného jasu nebo barvy v obraze. Na uvedeném příkladu (Obr. 1.4) je vidět, že největší četnost dosáhly hodnoty od 0 do 0,2 a od 0,7 až do 1.

Jestliže se v obraze objevují objekty podobného jasu a odlišného jasu od pozadí, výsledný histogram bude dvouvrcholový. Jeden z vrcholů náleží množství výskytu jasových elementů objektů a druhý odpovídá výskytu jasových úrovní pozadí. Výskyty hodnot mezi oběma vrcholy nejsou v obraze tolik časté, je tedy zřejmé, že odpovídají hraničním jasům mezi objekty a pozadím. Výsledný práh by měl splňovat požadavek minimální segmentační chyby. Práh segmentace pro oddělení objektu od pozadí se tedy určí jako nejméně četná hodnota jasu mezi dvěma vzdálenými maximy. U tohoto obrazu by to byla hodnota 0,4, což je na Obr. 1.4 vyznačeno červeným obdélníkem. Pro více vrcholový histogram lze určit více prahů, a to vždy mezi dvěma maximy. Určení prahu je efektivní, jen když jsou objekty jasově odlišné od pozadí. Pokud ovšem mají pozadí a objekty podobné jasové úrovně, potom práh určený z histogramu nemusí být vůbec použitelný [4].



Obr. 1.4: Určení prahu z histogramu

U Obr. 1.5 byla určena hodnota prahu podle vzorce 1.1, který je dán podílem součtu všech hodnot jasů I a počtem bodů v obraze. Kde X je počet vodorovných a Y svislých bodů. Hodnota jasu je reprezentována 1, nebo 0. Tím je určena průměrná hodnota jasu na jeden pixel. Hodnota prahu podle vzorce 1.1 je pro příkladový obraz 0,7.

$$práh = \frac{\sum_{x=0}^{X}\sum_{y=0}^{Y} I(x,y)}{X \cdot Y} \tag{1.1}$$

Obr. 1.5: Obrázek s příkladem po prahování obrazu

Místo prahování lze také využít detekce hran a linií v obraze, které slouží k nalezení tzv. hranové reprezentace, tj. odvozeného obrazu, v němž jsou hrany v první etapě zdůrazněny diferenčními operátory a následně pak ty z nich, které se jeví dostatečně výrazné, vymezeny prahováním. Taková prahová reprezentace může v dalším postupu sloužit například pro segmentaci obrazu, založenou na hranách [2].

1.2.3 **Natočení**

Natáčením obrazu do správné pozice je odstraněna chyba, která mohla být způsobena špatně vloženým papírem při skenování. Pro detekci správného natočení musí obraz obsahovat kontrolní značky. U vzorového obrazu byla využita dominantní svislá čára u písmene „B“. Jelikož je známo, kde se tato čára má při správném natočení nacházet, stačí jen otáčet obrazem a pomocí algoritmu najít správné natočení (Obr. 1.6).

Natočení ve 2D prostoru probíhá pro každý pixel jednotlivě podle počátku souřadnicového systému o úhel α. Rotaci reprezentuje násobení $A \cdot R$, kde $A = [x, y]$ jsou souřadnice bodu a $R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$ je transformační matice pro rotaci [5].

(1.2)

Pro vybrané úhly natočení je vstupní obraz procházen zleva doprava a pro každý sloupec je spočítán počet pixelů zobrazeného znaku, které se v něm nacházejí. Histogram $H = (h_1, \ldots, h_n)$, kde h_i je součet sloupce pixelů zobrazeného znaku budeme nazývat vertikální projekce. Hledaný úhel pro korekci natočení je pro úhel, kde je maximální hodnota vertikální projekce největší [6].

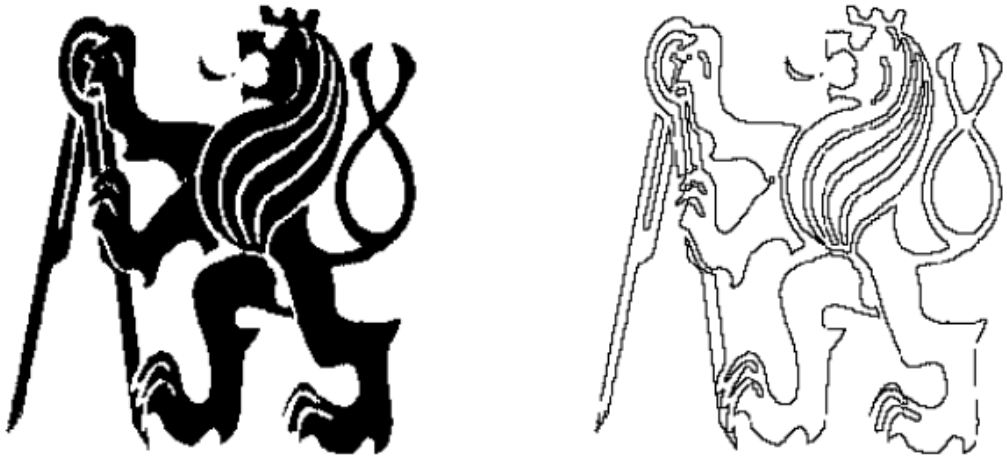
Obr. 1.6: Obrázek s příkladem po natočení.

1.2.4 **Binární morfologické operace**

Posláním těchto transformací, které se uplatňují při zpracování a analýze segmentovaných binárních obrazů, je odstranění šumu, vyrovnání hran objektů, oddělení slitých nebo naopak sloučení rušením rozdělených objektů, nalezení obrysů objektů nebo jejich skeletu [2].

Operace eroze je používána k odstranění tenkých čar a malých elementů. Po aplikaci eroze na obrázek dojde k rozdělení spojených znaků (viz Obr. 1.7). Pomocí této operace lze aplikovat jednoduchý detektor hran, obrys obrazu vznikne odečtením erodované verze obrazu od jeho černobílého originálu (viz Obr. 1.8).

Obr. 1.7: Operace eroze (převzato z [7])



Obr. 1.8: Obrys obrazu (převzato z [7])

Operace dilatace je opačná k operaci eroze. Používá se ke spojení čar nebo k zaplnění malých děr a úzkých zálivů v objektech (viz Obr. 1.9).

Obr. 1.9: Operace dilatace (převzato z [7])

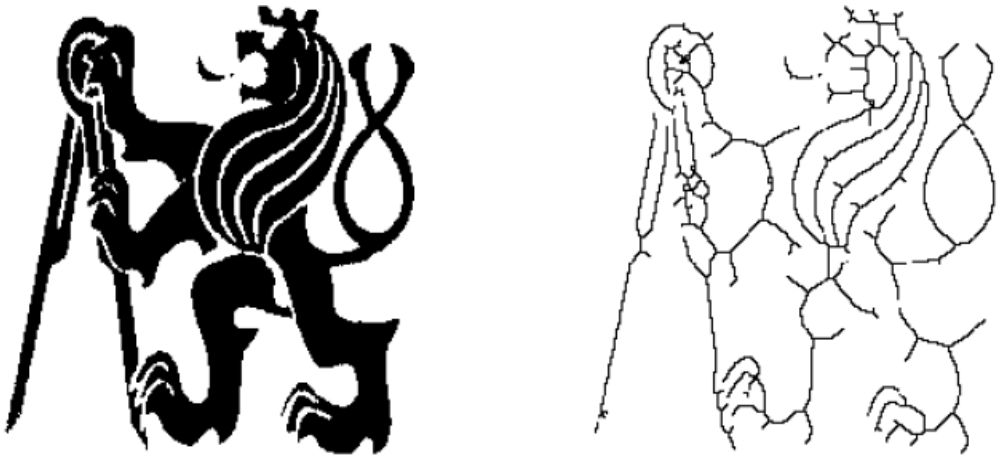
Operace uzavření je používaná ke spojování blízkých objektů, zaplňování malých děr v obraze a vyhlazování tvaru objektů (viz Obr. 1.10).

Obr. 1.10: Operace uzavření (převzato z [7])

Operace otevření je používána pro odstraňování šumu a pro oddělování objektů spojených tenkou čarou (viz Obr. 1.11) [4]. Na rozdíl od eroze a dilatace, opakované použití operace otevření respektive uzavření už nemá další vliv na výsledek.

Obr. 1.11: Operace otevření (převzato z [7])

Obecný postup operace k nalezení skeletu objektu spočívá ve ztenčování zadané množiny (viz Obr. 1.12). Pod tímto pojmem můžeme rozumět útvary čar, které vystihují danou množinu a zachovávají její geometrickou podstatu. Algoritmů na získání skeletu je velké množství, přičemž jejich výstupy se různí, protože neexistuje přesná definice skeletu množiny v rastrové grafice.

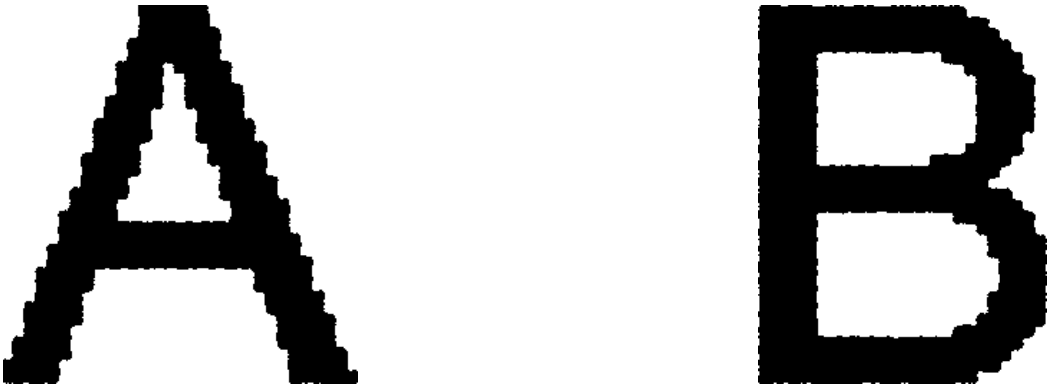


Obr. 1.12: Operace skelet (převzato z [7])

1.3 Segmentace

Vstupní obraz byl zbaven šumu, bylo provedeno prahování a také byl správně natočen. Následuje blok segmentace, jehož úkolem je rozdělení textu na jednotlivé znaky.

Segmentace je řazena mezi obtížné oblasti ve zpracování obrazu. Často je ztěžována řadou negativních vlivů, od nerovnoměrného osvětlení až po nedostatek informací o analyzovaném obraze. Proto také neexistuje univerzální metoda, která by byla aplikovatelná v každé oblasti analýzy obrazu. Algoritmy, které se segmentací obrazu zabývají, je možné rozlišovat podle způsobu provedení. Existuje řada přístupů, zde zmíníme jen tři nejpoužívanější. Mezi klasické techniky patří prahování. Iterativní metody založené na slučování nebo dělení regionů podle stupně podobnosti jejich vlastností. A jako poslední jsou metody, jejíž základ tvoří detekce hran v obraze [8].



Obr. 1.13: Obrázek s příkladem rozděleného obrazu na znaky A a B

Každá metoda má jisté nedostatky, například citlivost na šum nebo výpočetní náročnost, v některé oblasti však svými vlastnostmi může vynikat nad ostatními. Důsledkem toho vznikají kombinací tradičních přístupů různé hybridní metody, které zpravidla věnují svojí pozornost jen úzkému okruhu dané problematiky. Díky tomu ale dosahují často lepších výsledků než metody, které jsou orientovány obecněji [8].

Pro vzorový obraz nebylo nutné použít morfologické operace k rozdělení. Proto je konečná segmentace na jednotlivé znaky velice jednoduchá (Obr. 1.13). Složitější úkol nastává, pokud jde o ručně psaný text, protože některé znaky mohou být spojeny.

1.4 Klasifikace a oprava znaků

Klasifikace rozdělených znaků je závislá na použité metodě pro rozpoznání. V minulosti bylo navrhnuto mnoho metod pro klasifikaci znaků. Přehled běžně používaných metod je v kapitole 2.

Oprava znaků může probíhat dvěma způsoby, buď pomocí obsluhy, nebo automaticky použitím slovníku. Program je obsluhou naučen špatně rozpoznané znaky a dokáže je tak po opětovném spuštění správně rozpoznat.

Pokud je použit slovník, jsou výstupní rozpoznané znaky spojovány do slov a porovnávány s jeho databází. Když není nalezena shoda, je nahlášena chyba a slovo opraveno. Nejjednodušší způsob organizace slovníku je jednoduchý seznam slov. Výhodou tohoto přístupu je jeho jednoduchost. Jeho velkými nevýhodami jsou však velké paměťové nároky a neefektivní prohledávání slovníku [6].

Nejčastěji používaný způsob uspořádání slovníku je trie (Slovo *trie* pochází z anglického „retrieval“ – získávání, vyhledávání). Trie je stromová datová struktura, v jejíchž uzlech jsou uložena písmena slov. Každá cesta v trie představuje jedno slovo slovníku. Slova se stejným prefixem, mají stejný začátek cesty v tomto stromu, jehož délka koresponduje s délkou prefixu. Proto je trie označován jako prefixový strom. Výhody, které trie přináší, jsou komprese slovníku a tím snížení nároků na paměť a urychlení procedury vyhledávání ve slovníku [6].

2 METODY OCR

První patent na přístroj využívající OCR získal již v roce 1929 Gustav Tauschek z Německa. Jednalo se o mechanický přístroj využívající šablony. Když se překryla šablona s obrazem, vyslal fotodetektor signál, že se jedná o totožný znak [9].

První generace komerčních systémů se objevuje v letech 1960–1965. Využívá se pro jednoduché zpracovávání znaků. Z důvodu vyšší jistoty rozpoznání byly pro tyto systémy speciálně vyvinuty znaky, které však vypadaly velmi uměle. Začaly se také objevovat systémy s vícefontovou zásobou, které byly již schopny rozpoznávat znaky napsané různými fonty [9].

V polovině 60. let se začaly objevovat systémy druhé generace, které byly schopny rozpoznat běžné strojově vytisknuté texty a již měly i rozpoznávací schopnosti pro ručně psaný text. Zlom však přinesl teprve rok 1966, kdy se v USA standardizovalo tzv. písmo OCR-A, první písmo umožňující strojové čtení. Tento font byl velmi dobře navržen pro optické rozpoznávání a stále zůstal čitelný i pro lidi. Také byl vytvořen Evropský font (označen jako OCR-B), který byl pro lidi ještě lépe čitelnější [9].

Systémy třetí generace jsou z poloviny 70. let 20. století. Dovedou rozpoznávat dokumenty nižší kvality a ručně psané texty. Dnešní OCR systémy již jsou na velmi vysoké úrovni. Úspěšnost rozpoznání znaků přesahuje u latinky 99% [9].

Existuje několik metod pro optické rozpoznávání znaků, při kterých zjišťujeme různé charakteristické rysy znaků. Nejideálnější stav je ten, kdy každý znak má naprosto jiné rysy neopakovatelné u ostatních znaků. V tomto případě kdy máme znaky abecedy a číslice toto bohužel neplatí. Hodně znaků si je podobných a u textu psaného rukou je někdy složité jednotlivé znaky určit okem, natož pak pomocí počítače [9].

Problematika strojového čtení běžných tiskových dokumentů obsahující znaky latinské abecedy, je dnes považována za úspěšně vyřešenou. Pro dokumenty obsahující znaky jiných abeced, nebo dokumenty psané rukou ještě nebylo dosaženo tak uspokojivých výsledků. A proto jsou tyto úlohy stále součástí výzkumu [10].

Všechny OCR metody mají společný cíl, ale všechny dílčí procesy jsou odlišné. To se samozřejmě týká i vstupních dat, které mohou být různě naskenovaný text, psaný text rukou, kontrolní obrazy generované pro zabezpečení nebo text na digitálních fotografiích. To znamená, že pro použití OCR není jedna univerzální metoda [5].

2.1 Principy klasifikace a rozpoznávání

Rozlišuje se klasifikace, při které jsou objekty zařazeny do předem známého, pevného počtu tříd podle jejich společných vlastností. Do této kategorie patří například klasifikace znaků, kterou se bude tato práce zabývat. Na druhé straně je rozpoznávání, kde počet tříd není předem znám a třídy identifikujeme až během vlastního rozpoznávání, jako například rozpoznání plynulé řeči [11].

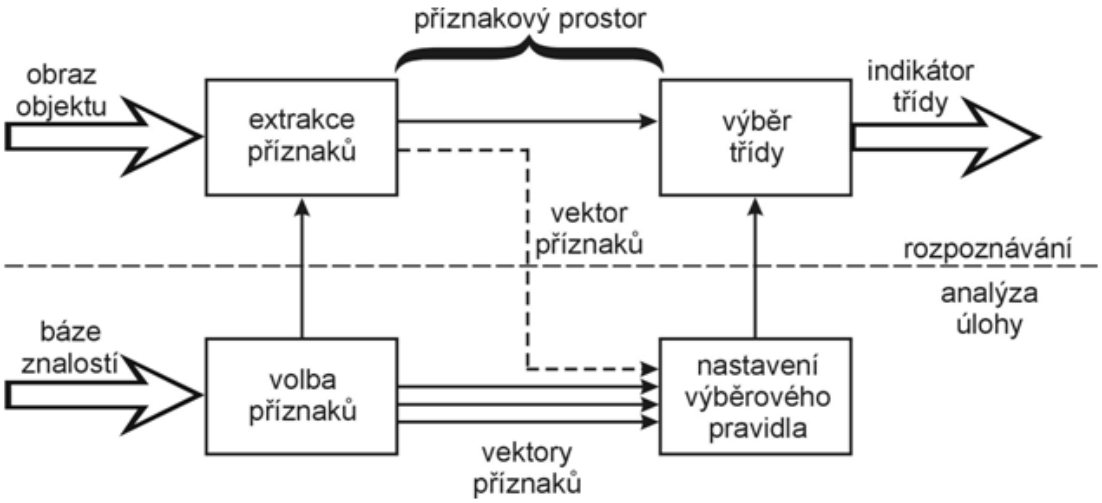
Pro objekt, jehož třídu chceme určit, je nejprve nutné zjistit jeho vlastnosti. Vlastnosti objektu mohou být reprezentovány buď příznaky, nebo strukturálním popisem [11]. Klasifikační metody jsou podle druhu popisu rozděleny na příznakové a strukturální.

Obě metody mají své výhody i nevýhody a spíše se vzájemně doplňují, než si konkurují. Proto se v praxi používá obou, strukturálních pro získání strukturálního popisu celku a příznakových většinou pro rozpoznání jednotlivých primitiv [12].

2.1.1 **Příznakové metody**

Každý znak je popsaný sadou několika příznaků, přičemž tyto příznaky jsou uspořádané do vektoru, řetězce nebo do stromu. Příznaky by měly být voleny tak aby jich bylo co nejméně, protože získávání většího počtu příznaků zpomaluje proces klasifikace a současně bychom zvolenými příznaky měli být schopní dostatečně přesně určit třídu objektu [6].

Jako vektor příznaků lze použít například přímo naměřené veličiny, spektrální koeficienty, histogram jednoho řádku snímku, úhlové projekce průsečíků průvodiče se sejmutým znakem, atd. [11]. Schéma klasifikace pomocí příznakových metod je zobrazeno na Obr. 2.1.



Obr. 2.1: Schéma klasifikace pomocí příznakové metody (převzato z [13])

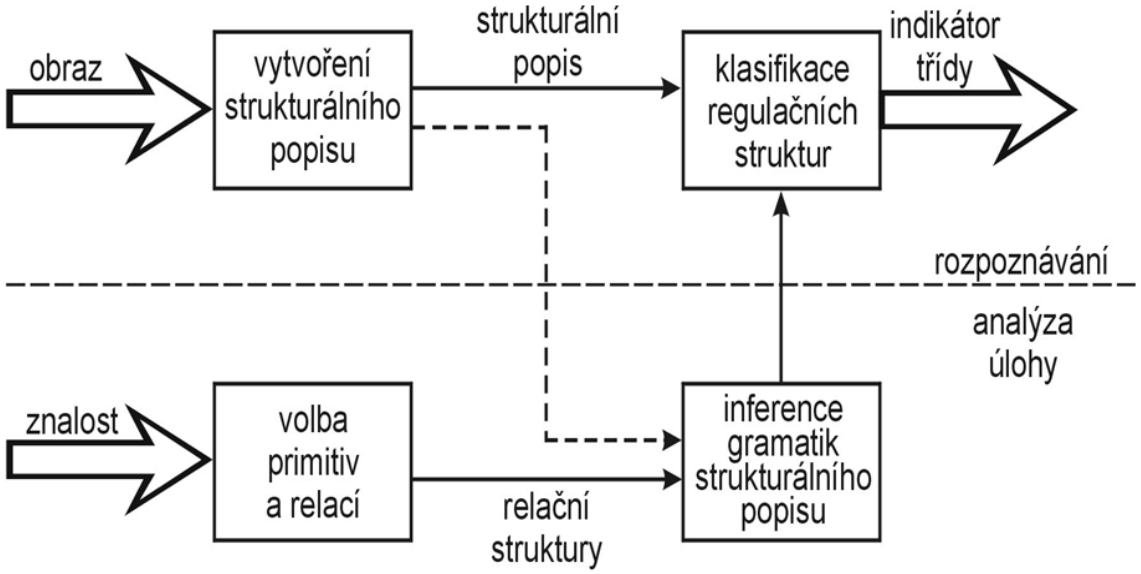
2.1.2 **Strukturální metody**

V případě velmi složitých obrazů může být informace o jejich struktuře natolik podstatná, že bez jejího využití nemůže být dosaženo žádoucích výsledků. Navíc počet příznaků v takových úlohách by byl neúnosný pro příznakové metody [12].

Strukturální metody rozpoznávání vycházejí z kvalitativního popisu objektu, kdy jsou jednotlivé znaky popisovány geometrickou a topologickou strukturou znaků [14].

U těchto metod jsou pro strukturní popis objektu určeny primitiva, vlastnosti primitiv a prostorové, časové nebo funkční relace mezi primitivy. Vytvořený symbolický popis vystihuje strukturální vlastnosti objektu. Postup při extrakci primitiv a vytváření strukturních obrazů je následující. Počet typů primitiv i relací mezi nimi by měl být co nejmenší. Primitiva by měla odpovídat základním strukturním elementům objektu, jimiž lze objekt vyčerpávajícím způsobem popsat. Primitiva musejí být snadno extrahovatelná a klasifikovatelná. Nalezení primitiv a relací mezi nimi by mělo být algoritmicky co nejjednodušší [11]. Schéma klasifikace pomocí strukturálních metod je zobrazeno na

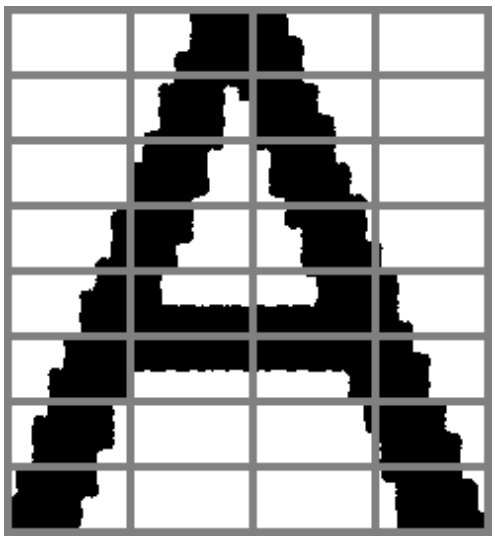
Obr. 2.2.



Obr. 2.2: Schéma klasifikace pomocí strukturální metody (převzato z [13])

2.2 Rozdělení obrazu do pásem

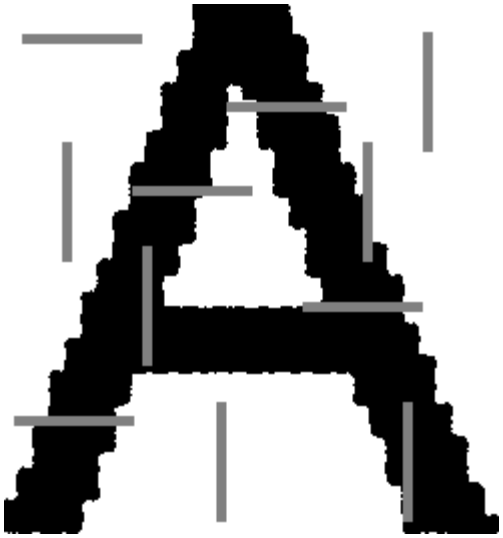
Obraz znaku (Obr. 2.3) je rozdělen do segmentů. V každém segmentu dojde k sečtení černých bodů. Pokud je součet větší než předem zvolená prahová hodnota, je hodnota segmentu nastavena na 1, pokud je součet menší než tento práh, segment má hodnotu 0. Výsledky jednotlivých segmentů jsou potom porovnávány se vzorovými znaky. Jeden segment je tvořen několika body. Čím více segmentů, tím je metoda úspěšnější, ale narůstá výpočetní náročnost. Například pokud velikost jednoho obrazu je 40 x 20 bodů, je rozdělen na 32 segmentů o rozměrech 8 x 4 bodů. Výstupní znak je ten, jehož referenční obraz hlásí nejvíce shod se vstupním obrazem. Tato metoda je nejjednodušší, ale je vhodná pouze pro rozpoznávání strojového textu, kde se hledané znaky téměř neliší od znaků vzorových.



Obr. 2.3: Obraz rozdělený do pásem

2.3 Metoda průsečíků

Na vstupní obraz (Obr. 2.4) jsou přidány předem zvolené vektory a hledá se počet průsečíků se znakem. Metoda je závislá na velikosti znaku a jeho pootočení. Tato metoda je také vhodná pouze pro rozpoznávání strojového textu.

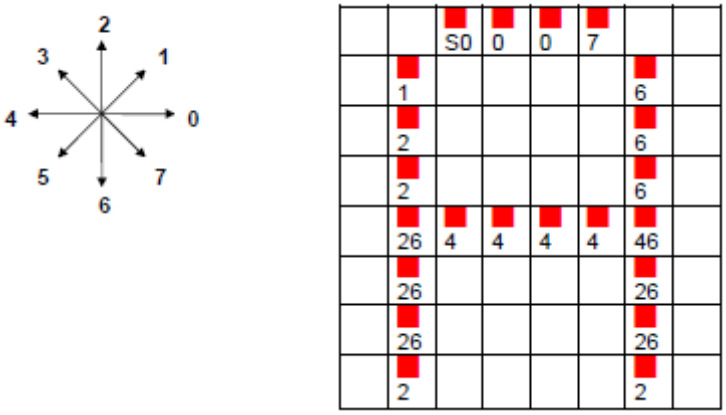


Obr. 2.4: Obraz s vektory průseček

2.4 Řetězové kódování

Řetězové kódování neboli Freemanův kód lze s výhodou použít k popisu tvaru jednotlivých etanolů i neznámého objektu a porovnávat pouze oba popisné řetězce. Takto kódován je skelet znaku. To znamená, že tloušťka čáry znaku je jeden pixel. Výhodou je možnost zpětné rekonstrukce objektu a nezávislost na posunutí. Nevýhodou je nutnost definovat vždy stejně počáteční místo popisu [10].

Na příkladu použití Freemanova kódu zobrazeném na Obr 2.5 lze znak A popsat jako řetězec: „000766666622244444666222221“.



Obr. 2.5: Směrová růžice Freemanova kódu a příklad kódování [10]

2.5 Statistické momenty

Statistické momenty jsou příznaky získané pomocí statistických charakteristik. Na jasovou funkci definující hodnoty pixelů v obraze je možnost pohlížet jako na náhodnou veličinu a využít ke klasifikaci jen hodnoty některých statistických veličin [10].

Momenty lze využít v mnoha aplikacích zpracování obrazu jako je rozpoznávání vzorů, kódování obrazu, odhad polohy atd. Popisují obsah obrazu a jeho rozložení vzhledem k jeho osám. Jsou utvořeny tak, aby obsahovaly jak globální, tak detailní geometrické informace o obrazu. Vzhledem k těmto faktům se dají využít pro charakterizování jednotlivých znaků a tím například získaní vhodných vstupních dat pro neuronovou síť [3].

Výhodou je nezávislost na posunutí a natočení obrazu, malý objem výstupních dat. Nevýhodou je výpočetní náročnost a nutnost určit chování a počet jednotlivých statistických veličin pro všechny etalony [10].

2.6 Umělá neuronová síť

Umělá neuronová síť se snaží vytvořit matematický model neuronových sítí živých organismů. Proces učení je napodobení navazování nových spojení mezi neurony pomocí synapsí [4].

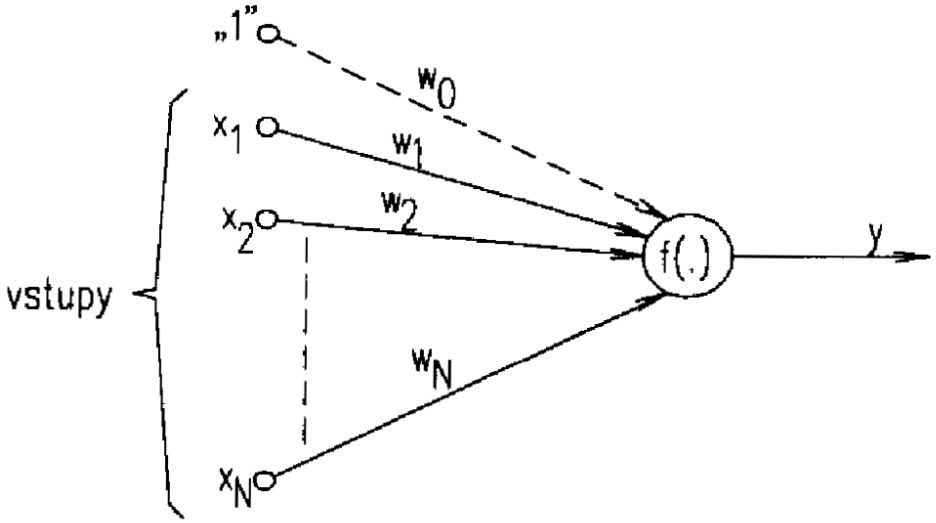
Základní stavební a funkční jednotkou neuronové sítě je neuron (Obr. 2.6), který má N vstupů, jeden výstup a je charakterizován rovnicí (2.1) udávající jeho výstup y pro vstupní vektor $x = [x_1, x_2, x_3, \ldots x_N]^T$, přičemž $w = [w_1, w_2, w_3, \ldots w_N]^T$ je vektor aktuálních vah, w_0 je aktuální práh neuronu a $f(\alpha)$ je zvolená, ale dále už neměnná zpravidla nelineární funkce, která se nazývá charakteristika neuronu. Argument této funkce se nazývá aktivace neuronu [2].

$$y = f\left(\sum_{i=1}^{N} w_i x_i - w_0\right) \tag{2.1}$$

Může se jednat o síť s metodou využívající učení s učitelem (např. perceptron) nebo bez učitele (např. Kohenenova síť) [1]. Perceptron je jedním ze základních stavebních prvků složitějších neuronových sítí. Je tvořen jedním neuronem, který má nastavitelné váhové koeficienty a práh. Perceptron sám o sobě dokáže klasifikovat data nejvýše do dvou tříd. Pro klasifikaci do více tříd se musí použít vícevrstvá neuronová síť [4].

Jak je vidět z rovnice (2.1) neurony jsou jednoduché matematické funkce a jsou v síti uspořádány do vrstev. Síť má jednu vstupní vrstvu, které se předává vektor deskriptorů a jednu výstupní vrstvu, z které se získá vektor představující třídu vzoru. Kromě toho zpravidla síť obsahuje ještě alespoň jednu vnitřní vrstvu neuronů [6].

Princip klasifikace je pak dán chováním neuronů. Jednotlivým vstupům neuronu jsou dány váhy, poté jsou data agregována a dochází k výpočtu aktivační funkce, která se stává výstupem neuronu a pokud je neuron ve výstupní vrstvě, stává se výstupem celé neuronové sítě [1].



Obr. 2.6: Model umělého neuronu (převzato z [2])

Asi největší předností neuronových sítí je schopnost učení, abstrakce a generalizace. Nezávislost na statistickém rozložení a vysoká tolerance šumu v datech je také oproti ostatním metodám výhodou. To vše je ale vykoupeno složitým systémem nastavení sítě a jejích parametrů, včetně prvotních vah synapsí, a dlouhou dobou učení [1].

2.7 2D korelační funkce

Korelační funkce jsou ve velkém využívané ve zpracování signálů, kde se vlastnosti mnohých signálů a jejich možnost vysílání a zpracování, upravují například tak, aby výsledek autokorelační funkce byl maximální (například jedno ostré maximum v čase), naopak se všemi ostatními signály minimální (charakter šumu) [5].

Pokud by byl zpracovávaný obraz kopií vzoru, byla by tato metoda velice jednoduchá. V reálných situacích však není nikdy zpracovávaný obraz přesnou kopií vzoru, neboť bývá zatížen šumem nebo geometrickým zkreslením. Výstupem je znak, jehož vzájemná korelace vzorového a vstupního obrazu dosáhla největšího výsledku, tzv. míru souhlasu. Metoda je časově a výpočetně náročná, a to i když nedochází ke geometrickým zkreslením nebo natočením [4]. Bohužel je tato technika citlivá na šum a variaci stylů. Je však lehce implementovatelná [9].

Dvojrozměrná korelační funkce udává vztah mezi veličinami, funkcemi $x,y = f(m,n)$, které jsou závislé na dvou proměnných m,n *v závislosti* na vzájemném posunutí τ,σ. Funkce dvou proměnných x,y reprezentuje dvojrozměrný obraz,kde proměnné x a y mají stejné rozměry M a N. Výsledek 2D korelační funkce je popsán rovnicí (2.2) [5].

$$ {}_{2D}R_{x,y}(\tau,\sigma) = \sum_{m=0}^{M}\sum_{n=0}^{N} x(m,n).\,y(m-\tau,n-\sigma) \tag{2.2}$$

2.7.1 **Funkce xcorr2**

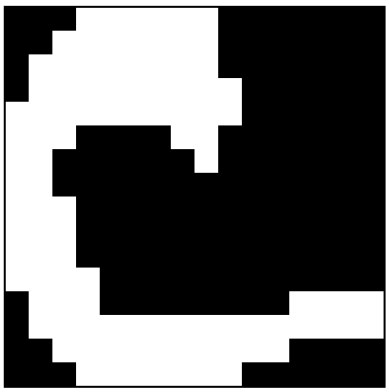
V programovém jazyku MATLAB je možné využít funkci *xcorr2*. Tato funkce vrací matici křížové korelace dvou matic *x* a *y*, které nemusí mít stejné rozměry *M* a *N* v závislosti na vzájemném posunutí τ,σ. Výsledek korelační funkce nezávisí jen na „tvaru“ signálu, ale i na velikosti hodnot funkce [15]. Vzorec pro funkci *xcorr2* je (2.3)

$$xcorr2 = {}_{2D}R_{x,y}(\tau,\sigma) = \sum_{m=0}^{Mx-1}\sum_{n=0}^{Nx-1} x(m,n).\bar{y}(m+\tau,n+\sigma) \tag{2.3}$$

kde $\bar{y}$ je komplexně sdružená matice a rozměr výstupní matice je součtem příslušných rozměrů, od kterých je odečtena hodnota jedna, jak je vidět ze vzorců (2.4)

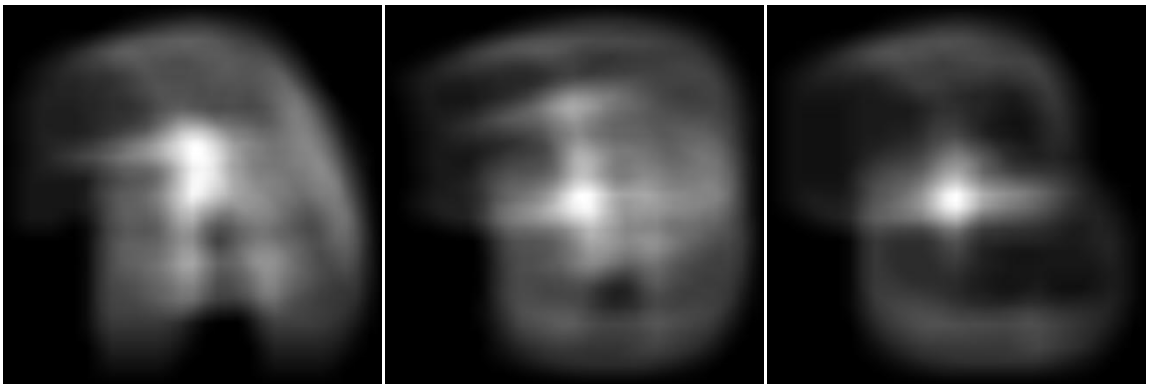
$$0 \le \tau < Mx + My - 1 \qquad \text{a} \qquad 0 \le \delta < Nx + Ny - 1\,. \tag{2.4}$$

Na obrázcích Obr. 2.7 až Obr. 2.9 je uveden příklad pro rozpoznávání pomocí funkce *xcorr2*.Pro tento příklad je jako vstupní obraz použit binární obraz se znakem C (Obr. 2.7). Tento obraz je postupně korelován s databází vzorových matic (Obr. 2.8). Rozměry obrazů jsou 20x20 pixelů. Příklady 2D korelací jsou zobrazeny na Obr. 2.9. Pro úplnost je třeba dodat, že úroveň jasu každé korelační funkce byl upravený tak, aby zobrazil výsledek v plném jasovém rozsahu. Pro každou korelaci je zjištěna maximální hodnota jasu v obraze. Z Obr. 2.9 je vidět, že největší hodnoty jasu jsou v blízkém okolí středu funkce, které odpovídá nulovému posuvu.



Obr. 2.7: Vstupní obraz se znakem C

Obr. 2.8: Vzorové matice pro znaky A, B a C



Obr. 2.9: Křížová korelace vstupního obrazu se vzorovými znaky

V tomto příkladu jsou pro názornost uvedeny pouze tři cykly korelace. V prvním cyklu je korelace vstupního obrazu se vzorem písmena A. Maximální hodnota této korelace je 81. V druhém cyklu je maximální hodnota korelace vstupního obrazu se vzorem písmena B rovna 95 a je větší než v předchozím cyklu. Jako prozatímní výstupní znak je zapsán znak "B". Ve třetím cyklu je maximální hodnota korelace vstupního obrazu se vzorem písmena C rovna 101. Tato hodnota i po proběhnutí všech cyklů zůstala největší a jako výstupní znak je tedy správně určen znak "C". Maximální hodnoty pro všechny tři cykly jsou zapsány v Tab. 2.1.

2.7.2 **Funkce corr2**

Dále existuje v jazyku MATLAB funkce *corr2*, ve které je počítán korelační koeficient podle rovnice (2.5)

$$r = \frac{\sum_{m}^{M}\sum_{n}^{N}(x_{mn} - \bar{x})(y_{mn} - \bar{y})}{\sqrt{\left(\sum_{m}^{M}\sum_{n}^{N}(x_{mn} - \bar{x})^2\right)\left(\sum_{m}^{M}\sum_{n}^{N}(y_{mn} - \bar{y})^2\right)}}, \tag{2.5}$$

kde x a y musí být matice o stejných rozměrech M a N, $\bar{x}$ respektive $\bar{y}$ jsou střední hodnoty matice. Výsledek r je skalární hodnota korelačního koeficientu, která určuje relativní míru lineární závislosti dvou náhodných veličin [15].

Korelační koeficient může nabývat hodnot od -1 do +1

r = 0, pak x a y jsou lineárně nezávislé

r > 0, pak x a y jsou lineárně závislé

r < 0, pak x a y jsou nepřímo lineárně závislé

Pokud se hodnota korelačního koeficientu pohybuje kolem nuly, značí to lineární nezávislost. Pokud se hodnota blíží k 1 respektive k -1, značí to přímou respektive nepřímou lineární závislost. Pro některé hodnoty korelačního koeficientu však nelze na první pohled určit, zda je či není významná a potom je nutné významnost korelačního koeficientu testovat.

Výsledky výpočtu korelačního koeficientu pro příklad uvedený v kapitole 2.7.1 jsou v tabulce Tab. 2.1, kde jsou porovnány s výsledky z výpočtu maximální hodnoty křížové korelace. Na rozdíl od funkce *xcorr2* u *corr2* byly relativně větší rozdíly mezi největší hodnotou a ostatními hodnotami. Tím se snižuje riziko špatné klasifikace. Z těchto tří výsledků lze usoudit, že výpočet korelačního koeficientu je pro klasifikaci znaků použitelný. Použití korelační metody je popsáno v kapitole 3.

Tab. 2.1: Srovnání korelačních metod

Vzorová matice	Hodnota maxima křížové korelace (xcorr2)	Hodnota korelačního koeficientu (corr2)
A	81	0.0599
B	95	0.0689
C	101	0.4845

3 DATABÁZE VZOROVÝCH ZNAKŮ

Pro tuto práci byla vybrána testovací databáze navrhnutá profesorem Simonem M. Lucasem (dostupná z [16]). V databázi jsou obsaženy ručně psané znaky. Příklad číslic je zobrazen na Obr. 3.1. Miniatury všech neupravených znaků jsou ukázány v příloze (1. Příloha: Databáze vzorů).

Obr. 3.1: Příklad databáze čísel (převzato z [1])

3.1 Vytvoření databáze

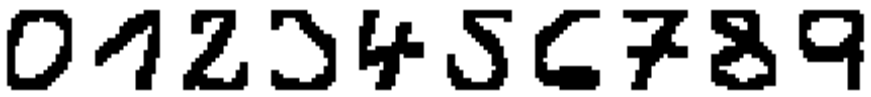
Znaky z této databáze byly upraveny do binárního obrazu, normalizovány a velikostně upraveny na rozměr matice 20 x 16 pixelů. Velikost matice je kompromisem mezi rychlostí výpočtu a úspěšností klasifikace. V databázi je obsaženo 39 znaků ve 36 třídách (10 číslovek a 26 písmen) a jsou proto uloženy ve struktuře o rozměrech 36 x 39. Struktura je navrhnuta tak, aby do ní bylo možno přidávat další znaky a tím rozšiřovat databázi.

Původ databáze je v USA. V této zemi jsou lehké odlišnosti v pravopisu některých znaků, proto musely být některé znaky upraveny tak, aby se shodovaly se zvyky ručně psaných čísel a tiskacích písmen u nás. Příklady takto upravených čísel jsou na Obr 3.2 a písmen na Obr. 3.3.

Dále byla pořízena tréninková databáze o počtu 10 znaků pro každou třídu, jejíž příklady čísel jsou na Obr. 3.4 a příklady písmen na Obr. 3.5. Celkem tedy 10 x 36 = 360 tréninkových znaků. Tyto znaky byly napsány deseti různými lidmi. Pro tréninkovou databázi byla provedena zkouška, jejíž výsledky jsou uvedeny v kapitole 5.1. Výsledky byly velmi dobré a tak byla tréninková databáze přidána do testovací, tím došlo k rozšíření databáze na velikost 36 x 49 = 1764 znaků.

Obr. 3.2: Upravená čísla testovací databáze

Obr. 3.3: Upravená písmena testovací databáze



Obr. 3.4: Upravená čísla tréninkové databáze

Obr. 3.5: Upravená písmena tréninkové databáze

3.2 Příklad klasifikace

Extrakce příznaků v korelační metodě je založená na výpočtu korelačního koeficientu. Pro každou třídu znaků existuje 49 vzorů. U rozpoznávání čísel proběhne pro každý neznámý znak 10 x 49 = 490 výpočtů korelačního koeficientu. Pokud jde o rozpoznávání písmen, proběhne pro každý neznámý znak 26 x 49 = 1274 výpočtů korelačního koeficientu.

Rozpoznávání probíhá pouze buď pro čísla, nebo pro písmena. Podle toho je přizpůsobena velikost matice na *i* sloupců a *j* řádků, do které jsou ihned ukládány všechny vypočtené korelační koeficienty. Pro každé číslo tedy vznikne matice $C_{i,j}$ o velikosti 10 x 49 bodů. Pro každé písmeno vznikne matice $P_{i,j}$ o velikosti 26 x 49 bodů.

Následným zpracováním této matice dojde k zařazení znaku do třídy. Mohly by být použity operace jako například výpočet průměru pro každou třídu, výpočet mediánu atd. V tomto případě je použito jednoduché nalezení souřadnice *i*, ve které je největší hodnota z této matice. Znak je tak klasifikován a zařazen do třídy podle toho, ve kterém sloupci je nalezena největší hodnota korelačního koeficientu.

Protože konečné rozpoznávání probíhá v hledání maximálního korelačního koeficientu, měly by být vzorové znaky od sebe co nejodlišnější. To znamená, že je vhodné, aby třída obsahovala co největší počet znaků, a případně i jejich chyb, které se mohou vyskytnout. S rostoucím počtem znaků se však zvětšuje doba výpočtu a proto je počet 49 vzorů v každé třídě rozumným kompromisem.

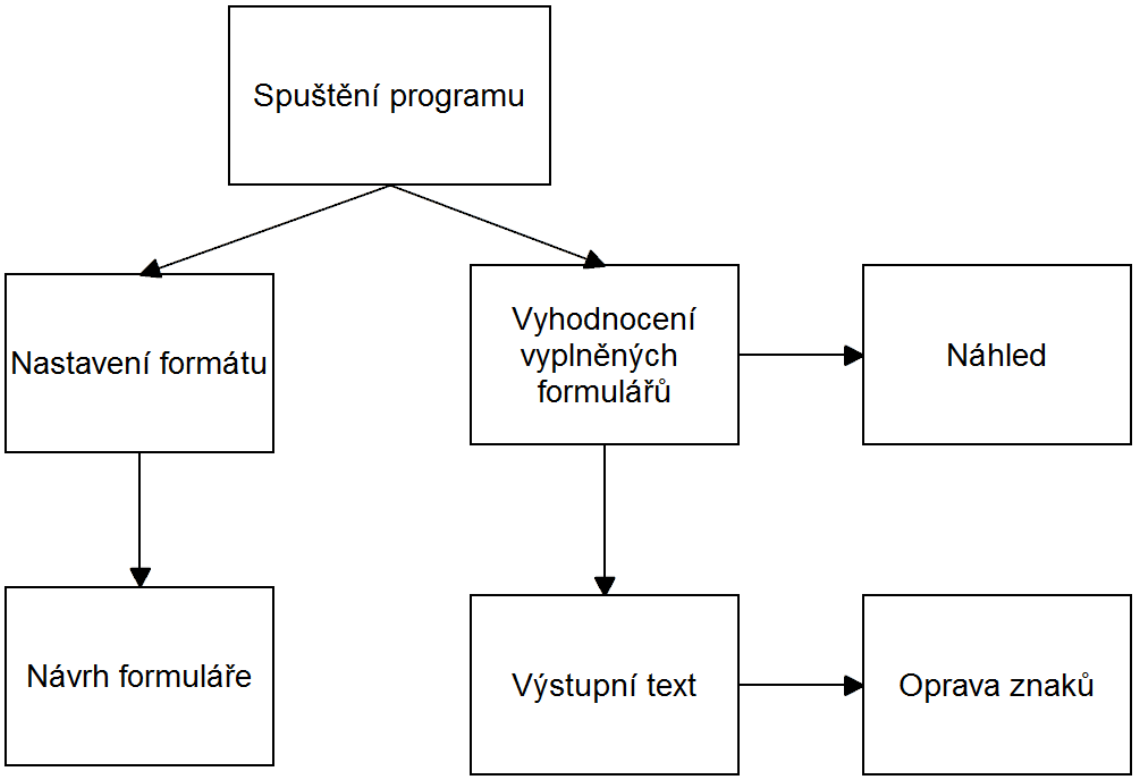
4 OCR PROGRAM

Praktická část diplomové práce se zabývá klasifikací ručně psaných písmen a číslic. Úkolem je navrhnout program, který rozpozná zapsaný text v kolonkách a číslice napsané na řádku. To znamená, že rozpoznávání textu a číslic bude probíhat zvlášť. Diakritika v textu nebude vyhodnocována, protože by databáze znaků musela být rozšířena o znaky s diakritikou. Pro klasifikaci byla při návrhu programu vybrána jako klasifikační nástroj korelační metoda. Konkrétně výpočet korelačního koeficientu, jehož vzorec je v kapitole 2.7.2.

Její výhoda spočívá v jednoduchosti a v rychlosti výpočtu. Rychlost výpočtu je však závislá na velikosti vstupních obrazů a na počtu vzorových matic. Velkou výhodou je také jednoduché rozšiřování databáze vzorových znaků. Tato metoda koreluje vstupní obraz hledaného znaku s databází všech znaků. Vzor, s kterým má vstupní obraz hledaného znaku největší hodnotu korelačního koeficientu, je určen jako výstupní znak.

4.1 Návrh programu

Jako prostředek k vytvoření programu bude sloužit software MATLAB od vývojářů společnosti The MathWorks, což je speciální programové prostředí s vlastním programovacím jazykem. Tento software je používán pro modelování, matematické výpočty, simulace, analýzu a k dalším vědeckotechnickým potřebám. Základním výpočetním prvkem programu MATLAB jsou matice. Program byl navrhnut v grafickém prostředí programu MATLAB verze 7.12 (R2011a).



Obr. 4.1: Diagram programu

Navrhnutý program je rozdělen do několika částí podle diagramu na Obr. 4.1. Všechny části jsou dále v textu detailně popsány. Spustitelný je v programu MATLAB souborem „uvodni_obrazovka1.m“.

Po spuštění je zobrazeno okno úvodní obrazovka (Obr. 4.2), které je tvořeno třemi tlačítky. Prvním tlačítkem je spuštěn návrh formuláře. Druhým tlačítkem vyhodnocení formulářů. Posledním tlačítkem je program ukončen.

Obr. 4.2: Úvodní obrazovka

4.2 Popis programu pro návrh formuláře

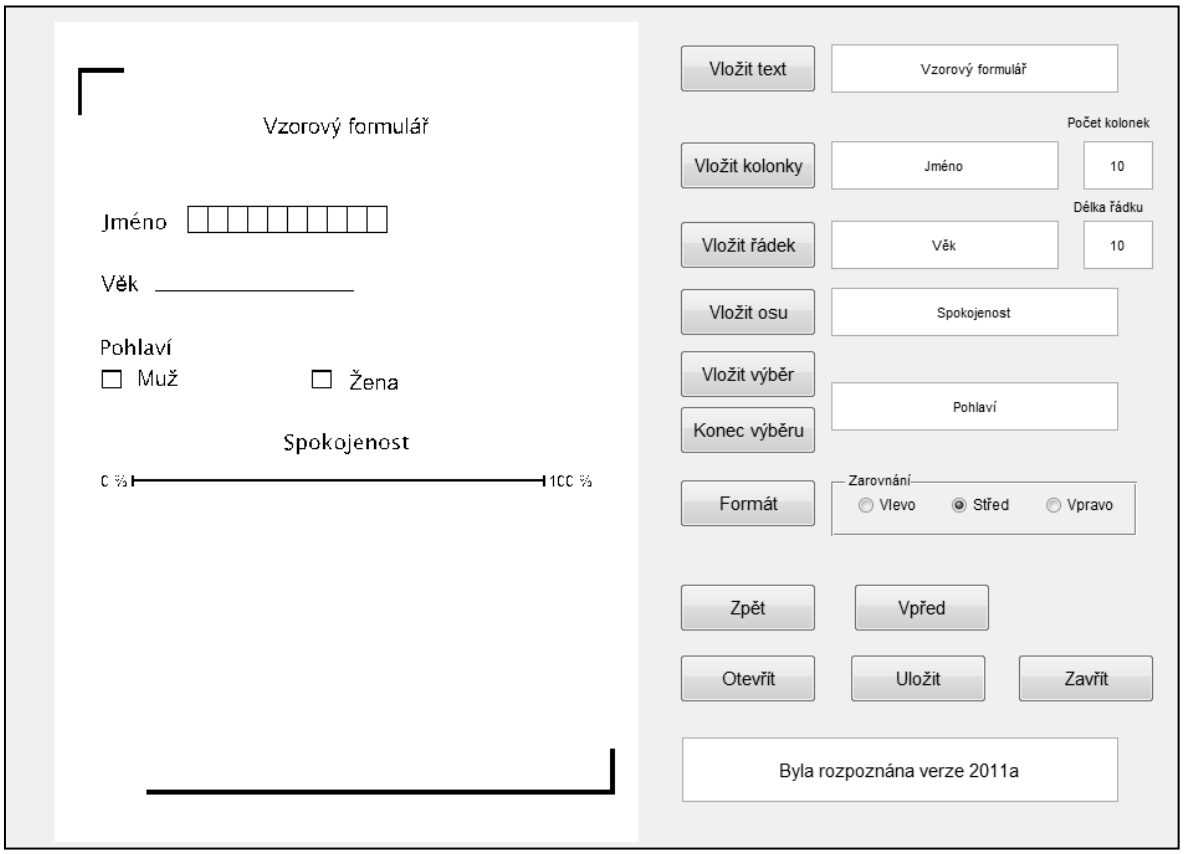
Stiskem tlačítka „Návrh“ je otevřeno okno programu (Obr. 4.3) pro návrh testových formulářů. Ihned po spuštění je automaticky zjištěna verze programu MATLAB, která je rozhodující pro výběr algoritmu, který převádí text na obraz. Verze je vypsána do spodního okna, které slouží pro vypisování stavů při návrhu formuláře. Od verze programu MATLAB 2011 a vyšší obsahuje MATLAB funkci *TextInserter*, která umožňuje vykreslení textu v obraze nebo ve videu. Pro starší verze je vkládání textu zdlouhavější. Je totiž nutné “probliknout“ text do samostatného okna, z něj pořízený snímek zkopírovat do schránky a až potom je možné vykreslení textu v obraze. Starší verze vkládání má také omezení v délce vkládaného textu, která je závislá na velikosti monitoru.

Funkce *TextInserter* však neumí vkládat české znaky a proto pokud je ve vkládaném slově obsažen znak s diakritikou, je v těchto případech použita starší metoda.

Dále jsou při spuštění ze složky data načteny soubory „okna.mat“, ze kterého jsou načítány naposledy vyplněné hodnoty v textových polích a „ulozeny_format.mat“, ze kterého je načten poslední uložený formát papíru (okraje, velikost písma, mezera mezi řádky, rozdělení stránky).

Okno programu je rozděleno na dvě hlavní části. Vlevo je zobrazena bílá plocha, se stejným poměrem stran jako formát papíru A4 (210 × 297 mm), sloužící pro vkládání textu

a objektů. Na Obr. 4.3 je na tomto místě vidět příklad navrhnutého formuláře, který obsahuje všechny možnosti vložených textů a objektů. Vpravo jsou umístěna tlačítka a pole pro zapisování textu.



Obr. 4.3: Návrh formulářů

Nejprve je nutné nastavit formát papíru, potom je teprve možné přistoupit k vytváření testového formuláře podle jednoduchého algoritmu:

1. Je proveden výběr zarovnání textu.

2. Požadovaný text je zapsán do editačního pole příslušející danému tlačítku.

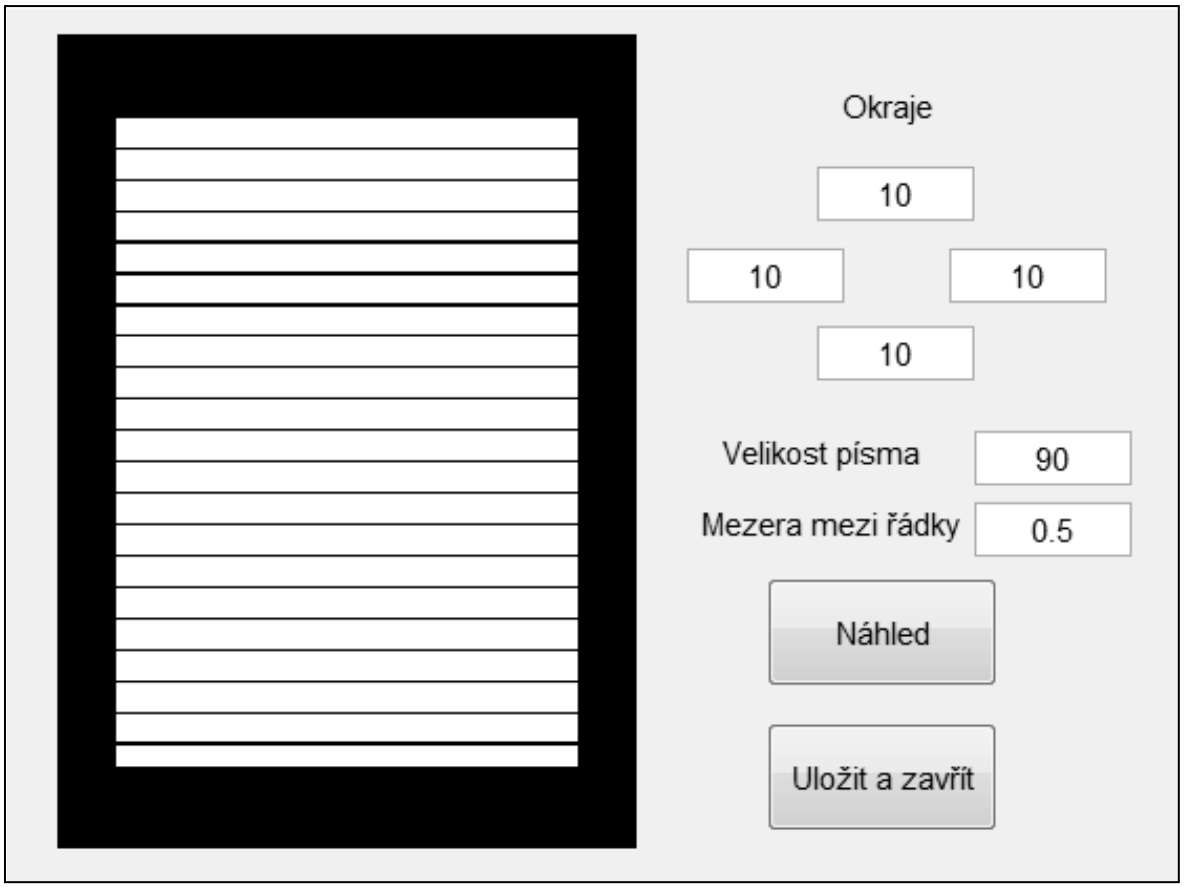
3. Stiskem tlačítka je vybrán objekt, který bude vložen.

4. Kliknutím myší do bílé plochy je vybrána pozice, kde bude umístěn zvolený objekt.

Tlačítkem „Formát“ je otevřeno nové okno (Obr. 4.4), kde se nastavuje formát stránky. Okraje jsou nastavovány v procentech, velikost písma je uvedena v pixelech a mezera mezi řádky je násobkem výšky písmena. Tlačítko „Náhled“ zobrazí vlevo náhled na rozvržení stránky. Jsou zde zobrazeny okraje a jednotlivé řádky. Nastavené hodnoty jsou uloženy po stisknutí tlačítka „Uložit a zavřít“ do souboru „ulozeny_format.mat“.

Před umístěním jakéhokoliv objektu je možnost zvolit volbu zarovnání, která umožňuje zarovnat vkládaný text „Vlevo“, na „Střed“ nebo „Vpravo“.

Stiskem tlačítka „Vložit text“ mohou být například vkládány nadpisy nebo jiné doplňkové texty, které nejsou ukládány spolu s formulářem a pro vyhodnocování formulářů nejsou důležité. Na příkladu (Obr. 4.3) je takto vložen text „Vzorový formulář“.



Obr. 4.4: Okno pro nastavení formátu

Při stisknutí tlačítka „Vložit kolonky“ je z editovacího okna „Počet kolonek“ nastaven počet kolonek. Stejným způsobem je při stisknutí tlačítka „Vložit řádek“ z editovacího okna „Délka řádku“ nastavena délka řádku. Potom jsou zadané hodnoty vykresleny.

Po stisku tlačítka „Vložit osu“ je nejprve vložen název osy a teprve potom je pod tento text vykreslena samotná osa. Obraz osy je vygenerován jen, pokud dojde ke změně velikosti písma a potom je uložen do souboru „osa.mat“, ze kterého je po dalším stisku tlačítka „Vložit osu“ načítán.

Vkládání výběru je složitější než předešlé postupy. Po prvním stisku tlačítka „Vložit výběr“ je vložen název skupiny například „Pohlaví“. Dalšími stisky jsou vkládány výběry v našem případě „Muž“ nebo „Žena“. Pro ukončení vkládání výběru je nutno stisknout tlačítko „Konec výběru“, jinak nelze pokračovat v dalším návrhu.

Při vkládání všech objektů jsou vždy do proměnné ukládány souřadnice těchto objektů a současně do jiné proměnné jejich názvy, kromě nadpisů bez objektů.

Stiskem tlačítka „Uložit“ jsou před uložením na papír vykresleny značky pro správné natočení a určení velikosti obrazu. Potom je uživateli zobrazena klasická nabídka, kde zadá název, formát, ve kterém si přeje navrhnutý formulář uložit a místo, kam se má formulář uložit. Jako výchozí je nastaven obrázkový formát BMP a složka „Výstup“. Při ukládání je automaticky uložen pod stejným jménem i soubor s příponou „.mat“, který je popisem formuláře. V tomto souboru jsou obsaženy hodnoty důležité pro pozdější vyhodnocování navrhnutých formulářů (souřadnice objektů a jejich názvy).

Tlačítkem „Otevřít“ lze otevřít a upravovat předtím uložené formuláře. Musí k nim však být soubor s popisem formuláře.

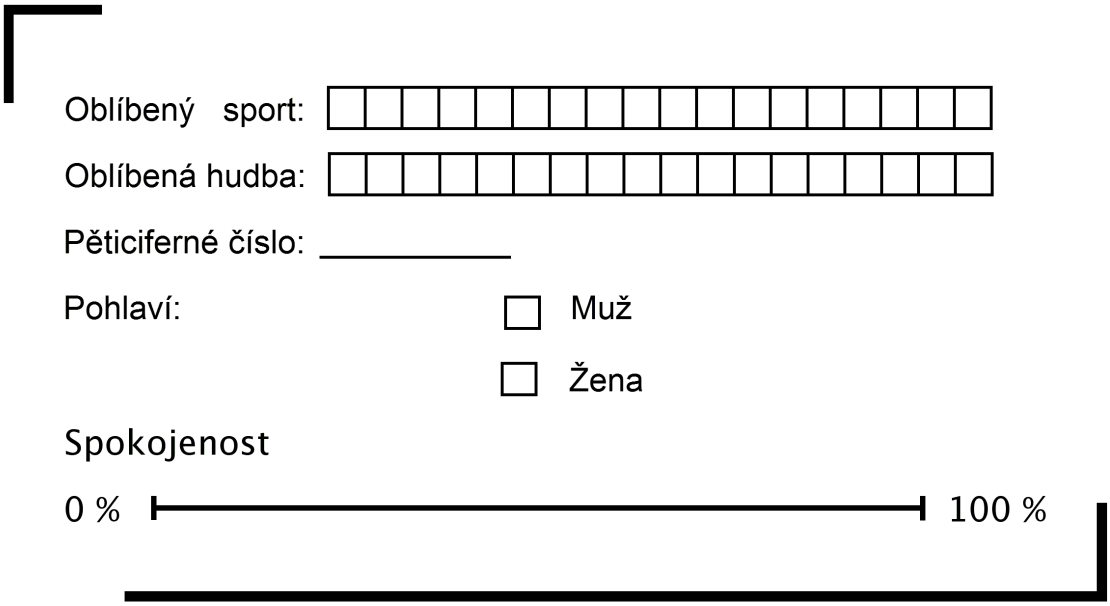
Pokud při vkládání objektů dojde k nějaké chybě, například překlepem nebo špatným umístěním, lze pro nápravu škody použít tlačítka „Zpět“ případně „Vpřed“.

Tlačítkem „Zavřít“ jsou uloženy do souboru „okna.mat“ naposledy vyplněné hodnoty textových polí a návrh formuláře je ukončen.

4.2.1 **Příklad návrhu formuláře**

Formulář, který byl rozdán mezi respondenty, byl navrhnut a vyplněn dříve než byl vytvořen program pro návrh formuláře. Aby mohly být vyplněné formuláře správně vyhodnoceny, byl v programu pro návrh znovu navržen totožný vzorový formulář (Obr. 4.5). Tím vznikl popisný soubor formuláře důležitý pro jeho správné vyhodnocení. Jediný rozdíl mezi těmito dvěma formuláři je ten, že starší verze neobsahuje spodní vodorovnou čáru, která je důležitá pro detekci správného natočení. Avšak ve starším formuláři lze pro detekci správného natočení použít spodní osy. Každý navrhnutý formulář v programu pro návrh formuláře obsahuje značky pro zajištění rozměrů a správného natočení.

Pro všechny formuláře stačí jeden soubor s popisem formuláře. V tomto souboru jsou zapsány souřadnice pozic testovaných polí a celková velikost formuláře mezi značkami ohraničení, která slouží k odstranění geometrického zkreslení. Dále soubor obsahuje názvy pro vypisování do tabulky v programu MS Office Excel, které usnadňují vyhodnocení.



Obr. 4.5: Navrhnutý formulář

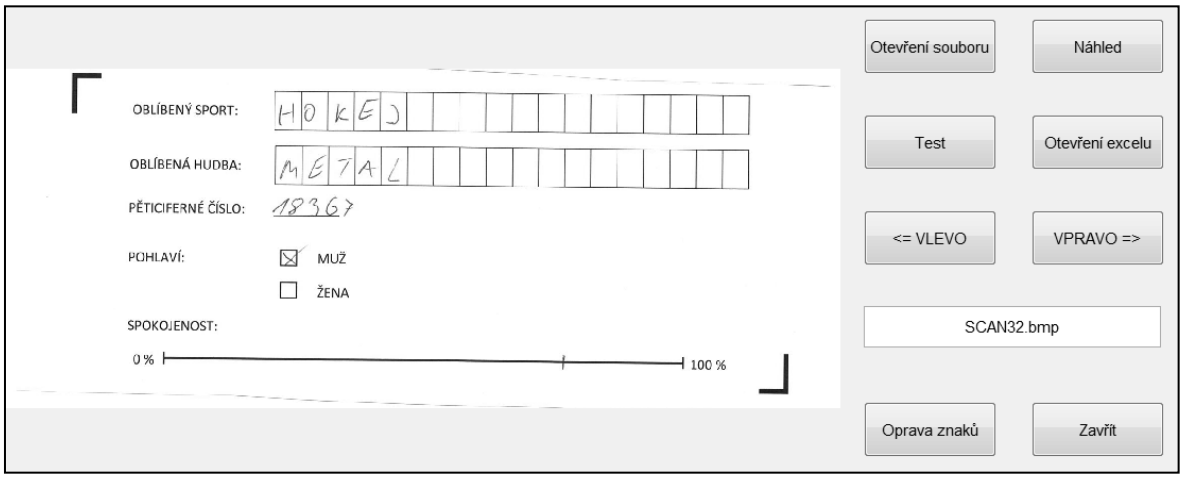
Navrhnutý formulář obsahuje pět testovaných polí. Dvě pole pro zapisování písmen do kolonek a po jednom poli pro zapisování čísel na řádek, volbu zaškrtnutím políčka a zjištění hodnoty z osy.

Formulář byl vytisknut v několika kopiích a rozdán mezi respondenty. Po vyplnění byly formuláře černobíle naskenovány. Naskenované formuláře jsou přiloženy v příloze (2. Příloha: Vyplněné formuláře).

4.3 Popis programu pro vyhodnocení

Stiskem tlačítka „Vyhodnocení“ v okně úvodní obrazovky je otevřeno okno programu pro vyhodnocení testových formulářů. Okno programu (Obr. 4.6) je tvořeno vlevo místem pro zobrazení testovaného formuláře a vpravo místem pro funkční tlačítka.

Po stisku tlačítka „Otevření souboru“ je otevřeno okno s klasickou nabídkou pro nahrání souboru. Uživatel může otevřít jeden nebo několik souborů. Vstupní soubory mohou být v jakémkoliv obrazovém formátu (např.: JPG, BMP, GIF, PNG, atd.). Současně s obrazovými soubory je automaticky načten soubor s popisem formuláře. Tento soubor se nemusí jmenovat stejně, ale ve složce s formuláři, které mají být vyhodnoceny, musí být jen jeden takovýto soubor.



Obr. 4.6: Vyhodnocení formulářů

Pokud je nahráno více vstupních obrazů, je možno mezi nimi pomocí tlačítek „VLEVO“ a „VPRAVO“ listovat. Název daného souboru je vždy uveden v textovém editoru pod šipkami.

Pomocí tlačítka „Náhled“ je možné zobrazit obraz formuláře ve větší velikosti.

Tlačítko „Oprava znaků“ slouží ke spuštění další části programu, která je popsána v kapitole 4.6. Toto tlačítko lze stisknou až potom co proběhne test alespoň jednoho formuláře. Tlačítkem „Zavřít“ je program ukončen.

Nejdůležitější je tlačítko „Test“, kterým je spuštěna OCR analýza všech otevřených formulářů. Průběh testování je vypisován do *Command Window* programu MATLAB. Po dokončení výpočtu se v okně pro vypisování stavů při testování formulářů objeví nápis „Konec výpočtu“.

Zmáčknutím tlačítka „Test“ je započat průběh několika po sobě následujících procedur. Nejprve je provedeno hrubé prahování s konstantním prahem, kdy dojde k rozdělení dokumentu na text a pozadí papíru. Dále je na binárním obrazu nalezena spodní vodorovná značka. Hledání probíhá tak, že maska jedniček (rozměry: 3 pixely na výšku a 15 na délku) je postupně posunována po celém obraze. Pokud jsou pod touto maskou také samé jedničky, jsou do výstupního obrazu na pozici masky uloženy samé jedničky, pokud ne, jsou do proměnné uloženy nuly. Ve výstupním obraze je jen tato spodní vodorovná značka. Tato značka slouží k určení správného natočení obrazu. Jelikož v tomto případě u testovaných formulářů takto upravená spodní značka chybí je místo ní použita osa.

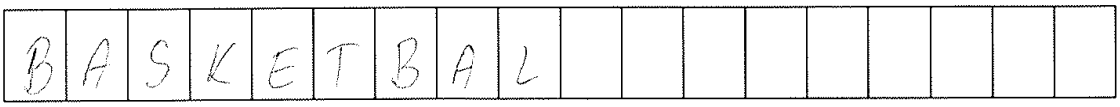
Správně natočený obraz je v prvním kroku stejným způsobem porovnáván s maskou jedniček s rozměry 4 x 12 a ve druhém kroku s maskou jedniček s rozměry 12 x 4, výsledkem součtu těchto dvou kroků je obraz obsahující jenom značky ohraničení formuláře.

Podle tohoto obrazu je potom oříznut vstupní obraz. Následuje důležitá věc a to kontrola velikosti obrazu. Pokud je zjištěn rozdílný rozměr od originálu, vypočte se koeficient podílem úhlopříčky vstupního obrazu a originálního formuláře. Koeficientem jsou násobeny souřadnice jednotlivých vyhodnocovaných polí. Velikost originálu a souřadnice polí jsou obsaženy v popisném souboru formuláře, který byl vytvořen společně s originálním formulářem.

Do souboru „obrazy.mat“ jsou uloženy všechny obrazy testovaných formulářů pro další zobrazení v okně oprava špatně klasifikovaných znaků (viz kapitola 4.4). Následuje zpracování jednotlivých polí.

4.3.1 **Vyhodnocení písmen**

Jako první jsou vyhodnocovány písmena zapsaná v kolonkách (Obr. 4.7).

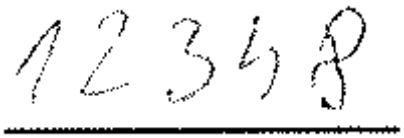


Obr. 4.7: Text zapsaný v kolonkách

Nejprve je provedeno prahování a mediánová filtrace s maskou o rozměrech 3 x 3 bodů. Potom jsou pomocí histogramu zjištěny polohy svislých čar a písmena jsou načítána z kolonek. Před samotnou funkcí rozpoznávání znaků je provedeno zúžení čáry znaku až na skelet a potom je provedena dilatace, která zabezpečí, že čára všech znaků bude stejně silná. Písmeno je oříznuto na minimální velikost a zmenšeno na stejný rozměr jako vzorové znaky. Následuje funkce rozpoznání, která porovnává jednotlivé znaky s databází písmen.

4.3.2 **Vyhodnocení čísel**

U čísel zapsaných na řádku (Obr. 4.8) je nejprve provedeno prahování, potom následuje procedura, která odstraňuje vodorovnou čáru. Navazuje nejtěžší úkol a to segmentace na jednotlivé znaky, která je provedena ve třech krocích. Skeletizace, dilatace a opět skeletizace. Tyto tři kroky dokonale rozdělí spojené znaky. Rozdělené znaky jsou po jednom klasifikovány.



Obr. 4.8: Čísla napsaná na řádku

Klasifikace u rozdělených číslic probíhá stejně jako v předchozím případě u písmen. Je provedena skeletizace, dilatace, písmeno je oříznuto na minimální velikost a zmenšeno

na stejné rozměry jako mají vzorové znaky. Následuje funkce rozpoznání, která porovnává jednotlivé znaky s databází čísel.

Pro pozdější možnost opravy špatně rozpoznaných znaků a rozšíření databáze jsou po klasifikaci pro každý rozpoznávaný znak (čísla a písmena zvlášť) uloženy všechny návrhy rozpoznání a obrazy navrhovaných znaků do souboru „navrh.mat“.

4.3.3 Vyhodnocení zaškrtávacích políček

U zaškrtávacích políček (Obr. 4.9) je kontrolováno jedno políčko po druhém. Nejprve je provedeno prahování, potom je políčko ořízнuto na rozměry čtverce. Dále je proveden součet černých bodů v tomto čtverci. Pokud je součet větší než třetina obsahu čtverce, je usouzeno, že políčko bylo zaškrtnuto. Mechanizmus rozpoznání obsahuje práh, takže je rozpoznáno, pokud nebylo vybráno ani jedno políčko.

Obr. 4.9: Zaškrtávací políčka

4.3.4 Určení hodnoty z osy

U posledního pole pro určení procent z osy je u vstupního obrazu (Obr. 4.10) nejprve provedeno prahování a mediánová filtrace s maskou o velikosti 5 x 5 bodů. Obraz je ořízнut tak, aby na něm byla jenom osa. Podle rozměrů osy je vypočítán rozměr masky nul, která je postupně porovnávaná s celým obrazem a na výstupu zanechá v obraze pouze vodorovnou čáru. Po odečtení vstupního obrazu od obrazu s vodorovnou čárou je získán obraz s hledanou značkou. Pokud jsou u osy napsány ještě nějaké znaky jako u našeho příkladu, je jako zaškrtnutí v ose brána nejdelší svislá čára. Poloha značky pro určení procent z osy je vyhodnocována pomocí lineární stupnice.

Obr. 4.10: Osa pro určení procent

Výsledky vyhodnocení všech polí jsou po skončení výpočtu uloženy do tabulky a zapsány do souboru ve formátu Microsoft Office Excel, který je možné otevřít přímo z programu stiskem tlačítka „Otevření excelu“.

4.4 Oprava špatně klasifikovaných znaků

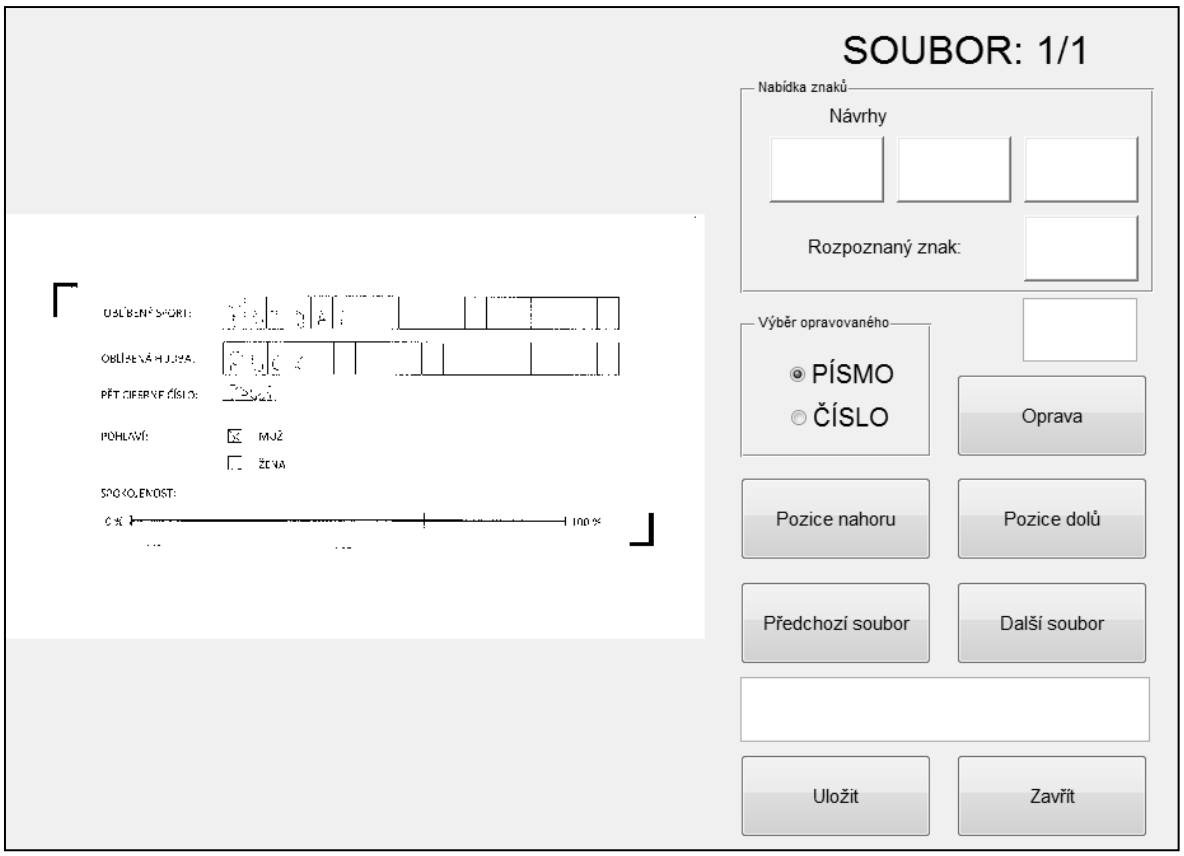
Úspěšnost klasifikace není nikdy stoprocentní a některé znaky nejsou rozpoznány správně. Proto byla navržena další část programu, pomocí které lze špatně rozpoznané znaky ručně opravit. Při této opravě je obraz opraveného znaku přidán do databáze vzorových znaků.

Tuto část programu lze spustit po testování formulářů z okna programu pro testování (Obr. 4.6) stiskem tlačítka „Oprava znaků“.

Po otevření okna pro opravu znaků (Obr. 4.11) je vlevo zobrazen právě opravovaný formulář. Pokud bylo vyhodnocováno více formulářů, je zobrazen první a vpravo nahoře je napsáno, který soubor z kolika je právě opravován. Náhledy opravovaných formulářů jsou načítány ze souboru „obrazy.mat“.

V rámečku „Nabídka znaků“ je zobrazen v zamačkávacích oknech znak, který byl rozpoznán a nad ním, pokud byla rozpoznána chyba, jeden nebo tři návrhy.

Pod tím se nachází „Výběr opravovaného“, kde lze vybrat, jestli se budou opravovat písmena nebo číslice.



Obr. 4.11: Oprava špatně rozpoznaných znaků

Napravo je tlačítko „Oprava“ a nad ním kolonka, která slouží k ručnímu zapsání znaku, pokud ani jeden z nabízených znaků není správným znakem.

Následují čtyři navigační tlačítka. Dvě spodní „Předchozí soubor“ a „Další soubor“ pro přepínání mezi formuláři a nad nimi dvě tlačítka „Pozice dolů“ a „Pozice nahoru“ pro pohyb po stránce formuláře.

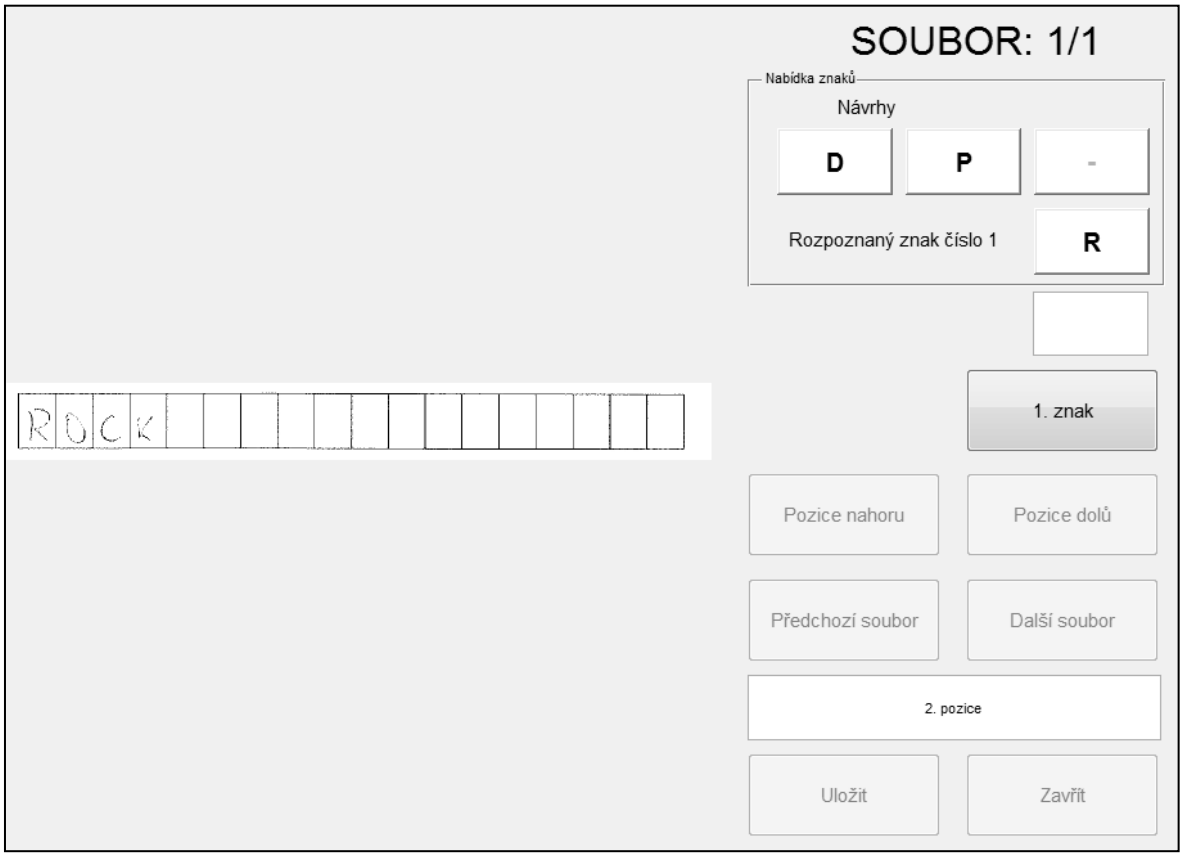
Pod tím je okno, kde se vypisují důležité informace o stavu programu. Poslední dvě tlačítka slouží k uložení opravených dat a uzavření okna programu.

4.4.1 Proces opravování

Vzhled okna v průběhu opravy znaku je vidět na Obr. 4.12. Proces opravování probíhá takto. Nejprve je obsluhou vybráno, zda se budou opravovat písmena nebo čísla. Potom je

tlačítky „Předchozí soubor“ a „Další soubor“ vybrán soubor s formulářem, kde je předpokládaná chyba. Na stránce se tlačítky „Pozice dolů“ a „Pozice nahoru“ vybere místo, které má být opraveno.

Stiskem tlačítka „Oprava“ je v okně zobrazen opravovaný objekt. Současně se provede výpočet, při kterém jsou zjištěny případné návrhy znaků. Ze souboru „navrh.mat“ jsou načteny návrhy rozpoznání pro čísla nebo písmena, které byly automaticky uloženy při klasifikaci. Program seřadí jednotlivé znaky sestupně podle korelačního koeficientu, a pokud je rozdíl mezi prvním a následujícím menší než 0,2, zařadí ho mezi návrhy. Pokud je velikost korelačního koeficientu u prvního znaku větší než u druhého o 0,2, předpokládá se, že znak byl rozpoznán správně a není ho potřeba opravovat. Hodnota 0,2 byla zvolena na základě testování. Je zvolena schválně větší hodnota, aby docházelo k planým poplachům, než aby se stalo, že některý znak nebude opraven.



Obr. 4.12: Průběh opravy špatně rozpoznaných znaků

Název tlačítka „Oprava“ se změní podle toho, na kterém místě byla nalezena potenciální chyba. V zamačkávacích tlačítcích se objeví návrhy nejpravděpodobnějších znaků. Následuje výběr uživatele tím, že zamáčkne některé ze tří navrhovaných znaků. Pokud žádný znak nevyhovuje lze napsat znak ručně do okna nad tlačítkem s pořadím znaku „1. znak“. Potom je nutné volbu potvrdit stiskem tlačítka „1. znak“. Tento postup se opakuje, dokud není opraven celý řetězec.

Stiskem tlačítka „Uložit a zavřít“ dojde k přepsání špatně rozpoznaných znaků v MS Office Excel a uložení normalizovaného obrazu znaku do databáze vzorových znaků v souboru „databaze.mat“. Dojde tím k rozšíření databáze a k vylepšení klasifikace.

5 ÚSPĚŠNOST KLASIFIKACE

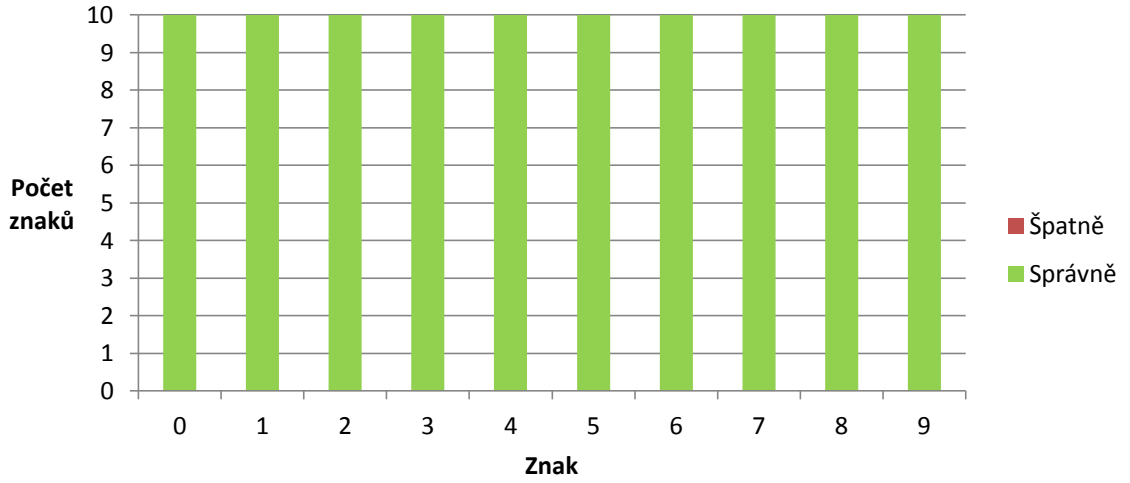
5.1 Výsledky testování tréninkové sekvence

Dříve než je přistoupeno k vyhodnocování formulářů je nutno navrhnutý algoritmus klasifikace vyzkoušet na tréninkové sekvenci. Ta obsahuje od každého znaku 10 vzorů. Celkem 10 x 36 = 360 tréninkových znaků. Tyto znaky byly napsány deseti různými lidmi. Klasifikace písmen a číslic probíhá zvlášť.

Výsledky pro testování čísel jsou v tabulce Tab. 5.1. Na Obr. 5.1 jsou tyto hodnoty vyneseny do grafu. Úspěšnost klasifikace u testovací množiny čísel byla 100 %.

Tab. 5.1: Úspěšnost klasifikace čísel tréninkové sekvence

Znak	Správně	Špatně	Úspěšnost [%]
0	10	0	100%
1	10	0	100%
2	10	0	100%
3	10	0	100%
4	10	0	100%
5	10	0	100%
6	10	0	100%
7	10	0	100%
8	10	0	100%
9	10	0	100%



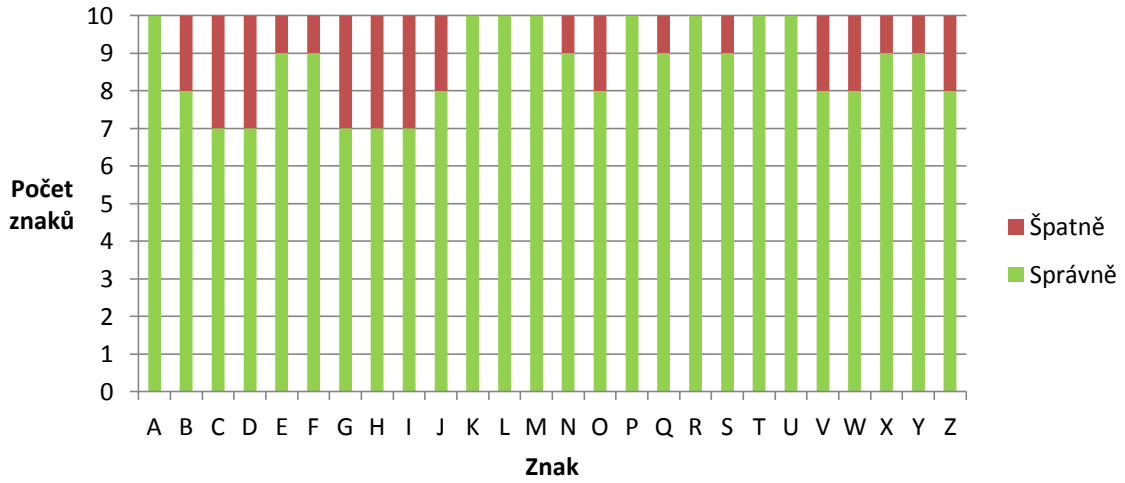
Obr. 5.1: Úspěšnost klasifikace čísel tréninkové sekvence

Výsledky pro testování písmen jsou v tabulce Tab. 5.2. Na Obr. 5.2 jsou tyto hodnoty vyneseny do grafu. Průměrná úspěšnost klasifikace u testovací množiny písmen byla 87 %. Menší úspěšnost než v předchozím případě je dána větším počtem tříd.

Po proběhnutí testu klasifikace na tréninkové sekvenci byla původní databáze 36 x 39 znaků rozšířena o tréninkové znaky na konečnou velikost 36 x 49 znaků.

Tab. 5.2: Úspěšnost klasifikace písmen tréninkové sekvence

Znak	Správně	Špatně	Úspěšnost [%]
A	10	0	100%
B	8	2	80%
C	7	3	70%
D	7	3	70%
E	9	1	90%
F	9	1	90%
G	7	3	70%
H	7	3	70%
I	7	3	70%
J	8	2	80%
K	10	0	100%
L	10	0	100%
M	10	0	100%
N	9	1	90%
O	8	2	80%
P	10	0	100%
Q	9	1	90%
R	10	0	100%
S	9	1	90%
T	10	0	100%
U	10	0	100%
V	8	2	80%
W	8	2	80%
X	9	1	90%
Y	9	1	90%
Z	8	2	80%



Obr. 5.2: Úspěšnost klasifikace písmen tréninkové sekvence

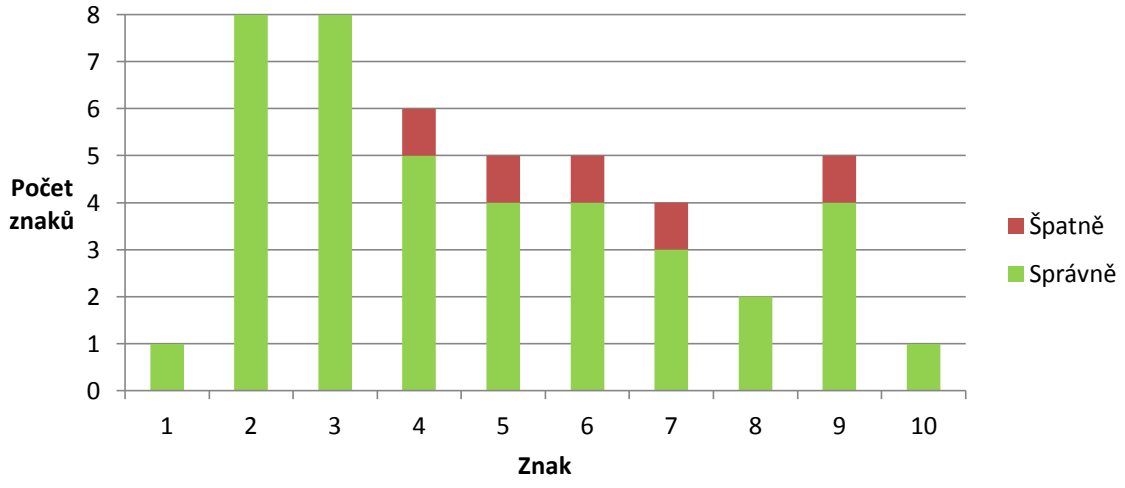
5.2 Výsledky testování formulářů po znacích

Otestováno bylo celkem 9 vyplněných formulářů. Výpočet vyhodnocení všech devíti formulářů trval přesně 2 minuty. Formuláře obsahovali celkem 45 čísel. Všechny znaky byly správně segmentovány. Správně rozpoznáno bylo 40 znaků, chybně 5 znaků. U pěti tříd čísel bylo naprosto bezchybné rozpoznání. U zbylých pěti tříd znaků se vyskytlo po jedné chybě. Ve dvou případech byl znak chybně klasifikován do třídy 9.

Údaje pro úspěšnost klasifikace čísel jsou uvedeny v tabulce Tab. 5.3, kde je vždy uveden znak, jeho četnost, počet správně a špatně rozpoznaných znaků, procentuální úspěšnost a také s jakým znakem byl znak zaměněn. V grafu na Obr. 5.3 je vynesen počet správně a špatně klasifikovaných čísel.

Tab. 5.3: Úspěšnost klasifikace čísel

Znak	Četnost	Správně	Špatně	Úspěšnost [%]	Chyby
0	1	1	0	100	-
1	8	8	0	100	-
2	8	8	0	100	-
3	6	5	1	83	1
4	5	4	1	80	9
5	5	4	1	80	9
6	4	3	1	75	5
7	2	2	0	100	-
8	5	4	1	80	7
9	1	1	0	100	-



Obr. 5.3: Úspěšnost klasifikace čísel

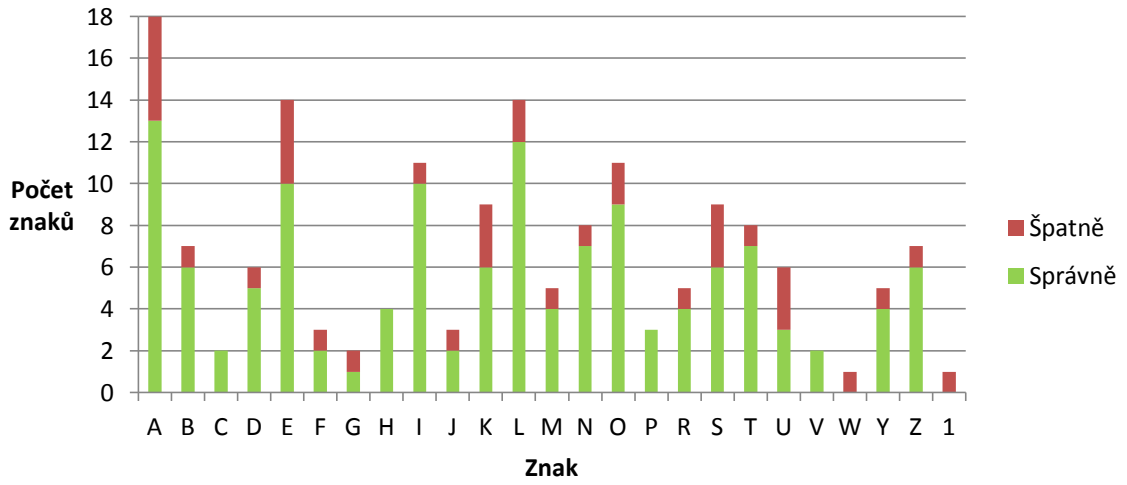
Formuláře obsahovali celkem 164 písmen. Správně rozpoznáno bylo 128 a špatně 36 znaků. Nejlépe si vedly znaky C, H, P, a V, které neměly žádnou chybu, avšak tyto znaky měly malou četnost. Nejvíce se vyskytovaly znaky A, E, I, O a L. Znaky Q a X se nevyskytovaly vůbec. V jednom případě se objevilo číslo 1, ale protože šlo o rozpoznávání písmen, číslo nebylo rozpoznáno. Nejvíce špatně klasifikovaných bylo do třídy F, kam bylo chybně zařazeno 6 písmen. Malý rozdíl byl shledán u písmen U a V. Tyto písmena způsobily dohromady 4 chyby klasifikace.

Ačkoliv diakritika nebyla při návrhu uvažována, některé znaky s diakritikou byly rozpoznány. Ve třech případech z osmi byl znak rozpoznán správně i s diakritikou. Ve všech třech případech šlo o dlouhé I.

Údaje pro úspěšnost klasifikace písmen jsou uvedeny v tabulce Tab. 5.4, kde je jako v předchozím případě vždy uveden znak, jeho četnost, počet správně a špatně rozpoznaných znaků, procentuální úspěšnost a také s jakým znakem byl znak zaměněn. Úspěšnost klasifikace pro písmena je vynesena v grafu na Obr. 5.4.

Tab. 5.4: Úspěšnost klasifikace písmen

Znak	Četnost	Správně	Špatně	Úspěšnost [%]	Chyby
A	18	13	5	78	F,F,I,I,X
B	7	6	1	17	A
C	2	2	0	100	-
D	6	5	1	50	Z
E	14	10	4	93	F,F,I,T
F	3	2	1	67	T
G	2	1	1	50	M
H	4	4	0	100	-
I	11	10	1	91	F
J	3	2	1	67	S
K	9	6	3	78	I,R,R
L	14	12	2	79	Z,Z
M	5	4	1	40	A
N	8	7	1	75	V
O	11	9	2	64	D,D
P	3	3	0	100	-
R	5	4	1	50	P
S	9	6	3	78	F,G,J
T	8	7	1	75	I
U	6	3	3	67	V,V,V
V	2	2	0	100	-
W	1	0	1	0	U
Y	5	4	1	60	J
Z	7	6	1	100	T
1	1	0	1	0	A



Obr. 5.4: Úspěšnost klasifikace písmen

5.3 Vyhodnocení formulářů

Pro rozpoznávání textu byla celková úspěšnost 78 %. Jen pět slovních spojení dosahovalo 100 % úspěšnosti rozpoznání. V dalších třech slovních spojeních se vyskytla jedna chyba. Největší úspěšnost byla u formuláře číslo 9. Nejhorší u formuláře číslo 3. V tabulce Tab. 5.5 a Tab. 5.6 jsou vypsány výsledky pro jednotlivé formuláře. V tabulkách není vyznačena diakritika, neboť při návrhu programu nebyla schválně uvažována.

U rozpoznávání čísel byly výsledky celkově poněkud lepší. Všechna čísla byla správně rozdělena na jednotlivé číslice. U šesti případů byla úspěšnost rozpoznání rovna 100 %. Pro rozpoznávání číslic byla celková úspěšnost 89 %. V tabulce Tab. 5.7 jsou výsledky pro jednotlivé formuláře.

Tab. 5.5: Rozpoznání textu (1.část)

Formulář	Originál	Rozpoznáno	[%]
1	FOTBAL	TDTAAL	50
2	CYKLISTIKA	CYKLIJTIRA	80
3	HOKEJ	HOKEJ	100
4	FORMULA 1	FOPMVLA A	63
5	LYZOVANI	LYTDVINF	50
6	POBIHANIPOLESE	POBIHXNIPOLESE	93
7	BASKETBAL	BFSIETBAL	78
8	JIZDA NA KOLE	JIZDA NA KOLT	91
9	KRYGLISTIKA	RRYGLISTIKA	91

Tab. 5.6: Rozpoznání textu (2.část)

Formulář	Originál	Rozpoznáno	[%]
1	ROCK	ROCK	100
2	METAL ZELENA	METAZ ZELEVI	73
3	METAL	AFIAZ	20
4	PINK FLOYD	PINK FLOYD	100
5	HOUSE	HOUSE	100
6	NUJAZZELETROSWING	NVSAZZILETROSUINM	71
7	HEAVY METAL	HFFVJ METAL	70
8	DRUM AND BASS	ZRVM AND BAGF	64
9	ZDUB SI ZDUB	ZDUB SI ZDUB	100

Tab. 5.7: Rozpoznávání čísel

Formulář	Originál	Rozpoznáno	[%]
1	25625	25625	100
2	92416	92416	100
3	18367	17157	40
4	84315	84315	100
5	62183	62183	100
6	13721	13721	100
7	12348	12348	100
8	28451	28491	80
9	02354	02359	80

U rozpoznání zaškrtnutých kolonek a čtení procent z osy se nevyskytovaly v žádném případě chyby. Výsledky těchto testů jsou v tabulce Tab. 5.8.

Tab. 5.8: Rozpoznání zaškrtnutých kolonek a čtení procent z osy

Formulář	Pohlaví	spokojenost
1	Muž	67 %
2	Muž	68 %
3	Muž	77 %
4	Muž	64 %
5	Muž	99 %
6	Muž	51 %
7	Muž	8 %
8	Muž	75 %
9	Žena	2 %

V tabulce Tab. 5.9 jsou vypsány celkové výsledky jednotlivých úloh rozpoznání. Pokud by se jednotlivé úlohy rozpoznání daly zprůměrovat, tak celková úspěšnost by byla 92 %.

Vyhodnocované formuláře jsou přiloženy v příloze.

Tab. 5.9: Celkové shrnutí

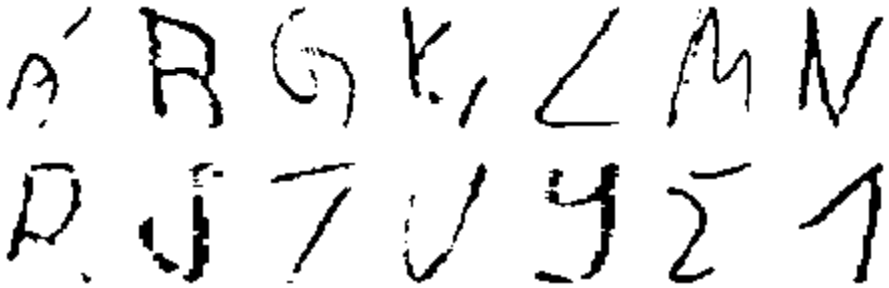
TEXT	ČÍSLA	POLÍČKA	OSA
78 %	89 %	100%	100%

5.4 Příklady klasifikace znaků

V této kapitole jsou uvedeny příklady správné a špatné klasifikace znaků. Znaky jsou zobrazeny, tak jak byly získány z formulářů. Předtím než tyto znaky vcházejí do funkce rozpoznání, je u nich provedena inverze barev. Na Obr. 5.5 je vidět příklad správně klasifikovaných písmen. Jde o znaky B, D, H, Í, K, N, S, U, V a Z.

Obr. 5.5: Příklady správně klasifikovaných písmen

Příklady špatně rozpoznaných znaků jsou na Obr. 5.6. Tyto znaky byly chybně klasifikovány jako I, A, M, R, Z, A, V místo správné klasifikace jako Á, B, G, K, L, M a N. V druhém řádku byly znaky klasifikovány jako P, J, I, V, J, T a A místo správné klasifikace jako R, S, T, U, Y, Ž a 1.



Obr. 5.6: Příklady špatně klasifikovaných písmen

Na Obr. 5.7 je vidět příklad správně klasifikovaných čísel od 0 do 9.

Obr. 5.7: Příklady správně klasifikovaných čísel

Znaky na Obr. 5.8 jsou všechny chybně klasifikované číslice. Byly chybně klasifikovány jako 1, 9, 9, 5 a 7 místo správné klasifikace 3, 4, 5, 6 a 8.

Obr. 5.8: Špatně klasifikovaná čísla

5.5 Oprava špatně rozpoznaných znaků

Klasifikace znaků nemá úspěšnost 100%, a proto je možné pokusit se zvětšit úspěšnost rozšířením databáze o znaky, které byly špatně rozpoznány. Databáze znaků bude rozšířena v navrhnutém modulu pro opravu špatně klasifikovaných znaků popsaném v kapitole 4.4.

Po rozšíření databáze znaků úspěšnost klasifikace vzrostla na 99 %. Jediná chyba vznikla tím, že do kolonek pro písmena bylo zapsáno číslo 1, které při opravě bylo přidáno ke třídě I a po opakovaném testování bylo jako I také rozpoznáno. Takže v tomto případě nejde o chybnou klasifikaci programu. Výsledky jsou zapsány v tabulkách Tab. 5.10, Tab. 5.11 a Tab. 5.12. V tabulkách není opět vyznačena diakritika, neboť při návrhu nebyla schválně uvažována.

Tab. 5.10: Rozpoznání textu po opravě (1.část)

Formulář	Originál	Rozpoznáno	[%]
1	FOTBAL	FOTBAL	100
2	CYKLISTIKA	CYKLISTIKA	100
3	HOKEJ	HOKEJ	100
4	FORMULA 1	FORMULA I	88
5	LYZOVANI	LYZOVANI	100
6	POBIHANIPOLESE	POBIHANIPOLESE	100
7	BASKETBAL	BASKETBAL	100
8	JIZDA NA KOLE	JIZDA NA KOLE	100
9	KRYGLISTIKA	KRYGLISTIKA	100

Tab. 5.11: Rozpoznání textu po opravě (2.část)

Formulář	Originál	Rozpoznáno	[%]
1	ROCK	ROCK	100
2	METAL ZELENA	METAL ZELENA	100
3	METAL	METAL	100
4	PINK FLOYD	PINK FLOYD	100
5	HOUSE	HOUSE	100
6	NUJAZZELETROSWING	NUJAZZELETROSWING	100
7	HEAVY METAL	HEAVY METAL	100
8	DRUM AND BASS	DRUM AND BASS	100
9	ZDUB SI ZDUB	ZDUB SI ZDUB	100

Tab. 5.12: Rozpoznávání čísel po opravě

Formulář	Originál	Rozpoznáno	[%]
1	25625	25625	100
2	92416	92416	100
3	18367	18367	100
4	84315	84315	100
5	62183	62183	100
6	13721	13721	100
7	12348	12348	100
8	28451	28451	100
9	02354	02354	100

6 ZÁVĚR

Tématem této diplomové práce bylo vyhodnocení testových formulářů pomocí OCR.

V první části práce bylo popsáno zpracování obrazového signálu a seznámeno s některými používanými metodami pro rozpoznání psaného textu. Pro klasifikaci znaků byla vybrána korelační metoda, konkrétně výpočet korelačního koeficientu. Dále byla vytvořena databáze vzorových znaků, kterou je možné dále rozšiřovat. V praktické části práce byl v grafickém prostředí MATLAB navrhnut systém pro tvorbu a statistické vyhodnocení testových formulářů. Navrhnutý program obsahuje tři hlavní části: návrh formuláře, vyhodnocení formulářů a opravu špatně klasifikovaných znaků.

V části pro návrh formuláře byl navrhnut vzorový formulář a devět kopií bylo rozdáno mezi respondenty. Formulář obsahoval pole pro zapisování písmen do kolonek, pole pro zapisování čísel na řádek, pole pro volbu zaškrtnutím políčka a pole pro zjištění hodnoty z osy.

Vyplněné formuláře byly v části programu pro vyhodnocení formulářů otestovány. Při návrhu programu bylo rozhodnuto, že program nebude vyhodnocovat diakritiku. Bylo to z důvodu, aby se nezvětšoval počet tříd pro klasifikaci písmen. Úspěšnost pro klasifikaci textu byla 78 %. Pro klasifikaci číslic byla úspěšnost větší a to 89 %. To bylo způsobeno hlavně tím, že u klasifikace písmen je více než dvojnásobek tříd než u klasifikace čísel. Rozpoznání zaškrtnutých políček a odečítání procent z osy bylo stoprocentní. Chyby v rozpoznání byly většinou u znaků těžko okem čitelných, avšak u rukou vyplněných formulářů se tomuto nedá vyhnout.

V části oprava špatně klasifikovaných znaků byly špatně klasifikované znaky uloženy do databáze vzorových znaků. S rozšířenou databází bylo znovu provedeno vyhodnocení formulářů. Úspěšnost pro rozpoznávání textu narostla na 99 %. Objevila se jedna chyba, která byla způsobena tím, že rozpoznání textu a číslic probíhalo zvlášť. Pro rozpoznávání číslic narostla úspěšnost na 100 %. Míra chybovosti navrhnutého systému byla minimalizována na nulu.

Výhodou navrhnutého systému je, že obsluha může rozšiřováním databáze učit program novým rukopisům. Nevýhodou je to, že spolu s rozšiřováním databáze roste doba výpočtu. Úspěšnost rozpoznání textu by se také zvětšila, pokud by byla přítomna databáze slov pro opravu chyb.

LITERATURA

[1] ZÁLESKÁ, K. *Klasifikace obrazů pomocí umělých neuronových sítí*. Brno: Masarykova univerzita v Brně, Přírodovědecká fakulta, Centrum pro výzkum toxických látek v životním prostředí, 2011. 53 s. Bakalářská práce. Vedoucí bakalářské práce: Ing. Milan Blaha, Ph. D.

[2] JÁN, J. *Číslicová filtrace, analýza a restaurace signálů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2002. ISBN 80-214-1558-4.

[3] MOŽNÝ, K. *Rozpoznávání pomocí umělé inteligence*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřící techniky, 2010. 64 s. Bakalářská práce. Vedoucí bakalářské práce: Ing. Luděk Červinka.

[4] KRAJÍČEK, P. *Rozpoznání SPZ/RZ*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřící techniky, 2010. 70 s. Diplomová práce. Vedoucí diplomové práce: Ing. Peter Honec.

[5] KAPUSTA, J. *OCR modul pro rozpoznání písmen a číslic.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2010. 70 s. Diplomová práce. Vedoucí diplomové práce: prof. Ing. Milan Sigmund, CSc.

[6] VESELÝ, P. *OCR Analýza*. Brno: Masarykova univerzita v Brně, Fakulta informatiky, 2005. 72 s. Diplomová práce. Vedoucí diplomové práce: Mgr. Lukáš Svoboda.

[7] HLAVÁČ, V. *Přednáška z předmětu Centrum strojového vnímání: Matematická morfologie*. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra kybernetiky. URL: <cmp.felk.cvut.cz/˜hlavac> [cit. 14. 5. 2013].

[8] FEICHTINGER, F. *Studium vlivu parametrů vybraných segmentačních metod na analýzu obrazů buněk*. Brno: Masarykova univerzita v Brně, Fakulta informatiky, Aplikovaná informatika, 2008. 48 s. Diplomová práce. Vedoucí diplomové práce: Ing. Pavel Matula, Ph. D.

[9] ŠVEC, Z. ŽLÁBEK, J. *Systémy OCR*. Praha: České vysoké učení technické v Praze, Fakulta stavební, 2010. Semestrální práce z předmětu Kartografická polygrafie a reprografie. URL: <http://verkapet.wz.cz/kapr.html> [cit. 14. 5. 2013].

[10] PETYOVSKÝ, P. *Přednáška kurzu Aplikace počítačového vidění: Strojové čtení obrazových kódů a znaků.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřící techniky. [cit. 14. 5. 2013].

[11] MATOUŠEK, V. *Přednáška kurzu Umělá inteligence a rozpoznávání: Rozpoznávání a klasifikace*. Plzeň: Západočeská univerzita v Plzni, Fakulta aplikovaných věd. Katedra informatiky a výpočetní techniky, 2013.

[12] ŽELEZNÝ, M. *Strukturální metody rozpoznávání*. Plzeň: Západočeská univerzita v Plzni, Fakulta aplikovaných věd. Katedra kybernetiky. [cit. 14. 5. 2013].

[13] ŠOTEK, K. *Informačné a riadiace systémy v doprave*, Žilina: Žilinská univerzita, Fakulta riadenia a informatiky, 2010.

[14] VIKTORA, J, VYMETÁLEK, P. *Optical Character Rocognition*. Praha: České vysoké učení technické v Praze, Fakulta stavební, 2008. Semestrální práce z předmětu Kartografická polygrafie a reprografie. URL: <http://geo3.fsv.cvut.cz/vyuka/kapr/SP/2008_2009/vymetalek_viktora/index.html> [cit. 14. 5. 2013].

[15] MATLAB, The MathWorks. Help. URL: <http://www.mathworks.com/help/matlab> (součást programového prostředí MATLAB)

[16] LUCAS, S. Database of handwritten alphadigits. URL: <http://www.cs.nyu.edu/~roweis/data.html> [cit. 14. 5. 2013].

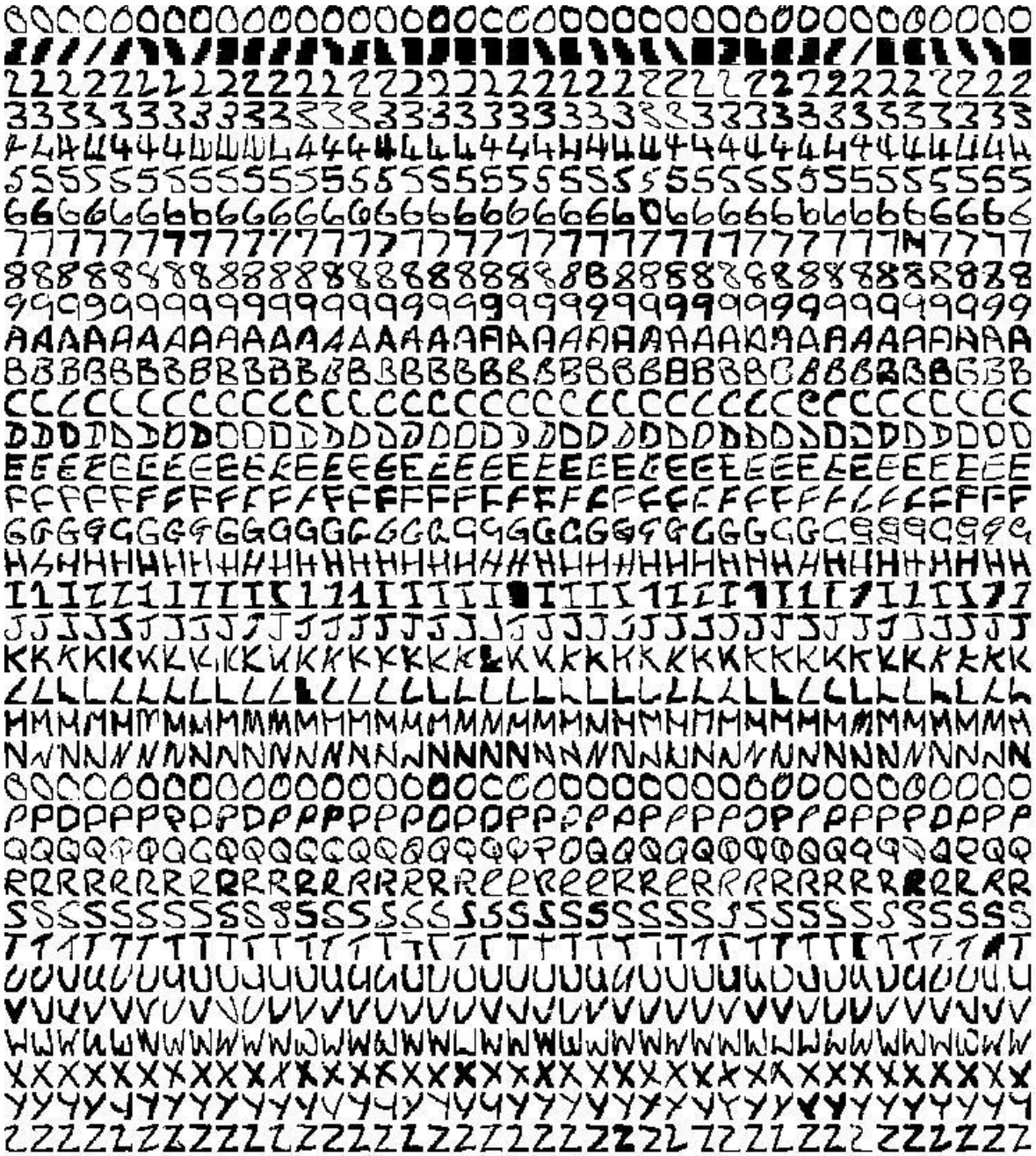
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BMP	Formát pro ukládání rastrové grafiky
corr2	Funkce pro výpočet korelačního koeficientu
f(α)	Charakteristika neuronu
H	Histogram
h_i	Vertikální projekce
I(x,y)	Hodnota jasu pro pixel se souřadnicemi x a y
MATLAB	Skriptovací programovací jazyk
OCR	Optické rozpoznávání znaků (Optical *Character Recognition*)
OCR - A	První písmo umožňující strojové čtení
OCR - B	Evropský font umožňující strojové čtení
xcorr2	Funkce pro výpočet křížové korelace

SEZNAM PŘÍLOH

1. Databáze vzorů
2. Vyplněné formuláře

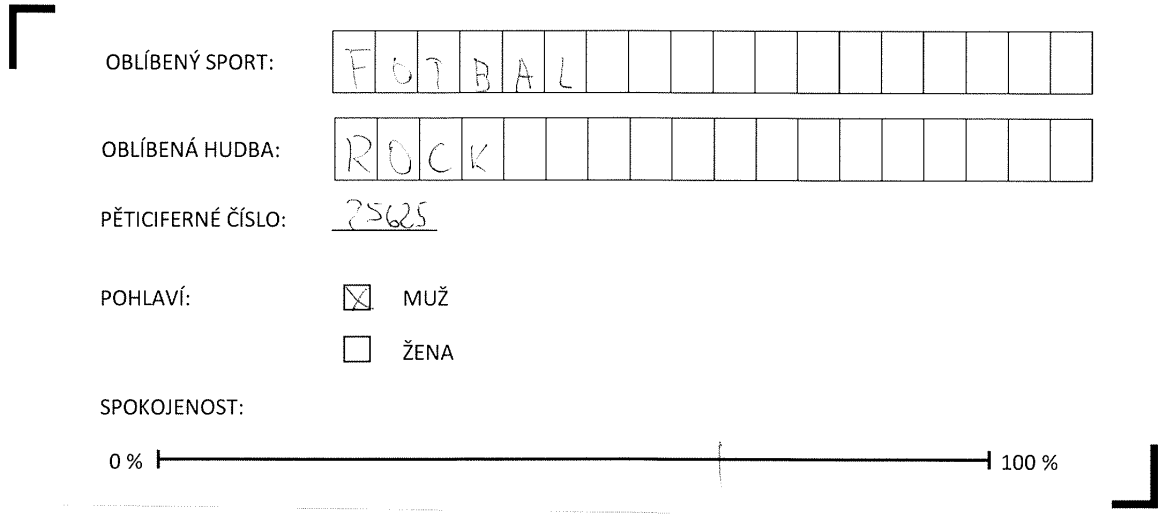
1. Databáze vzorů



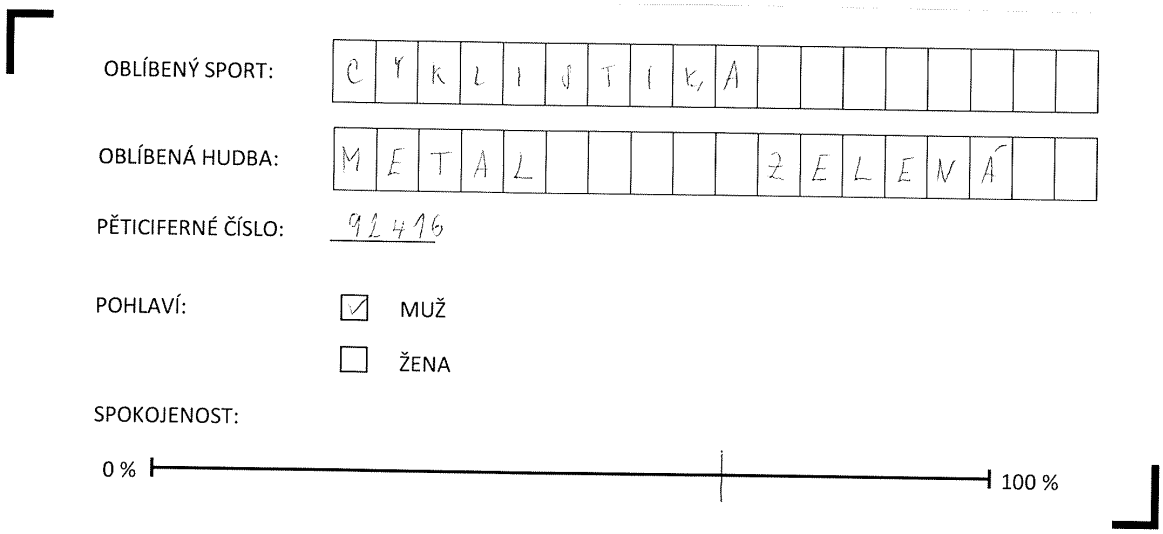
(dostupné z [16])

2. Vyplněné formuláře

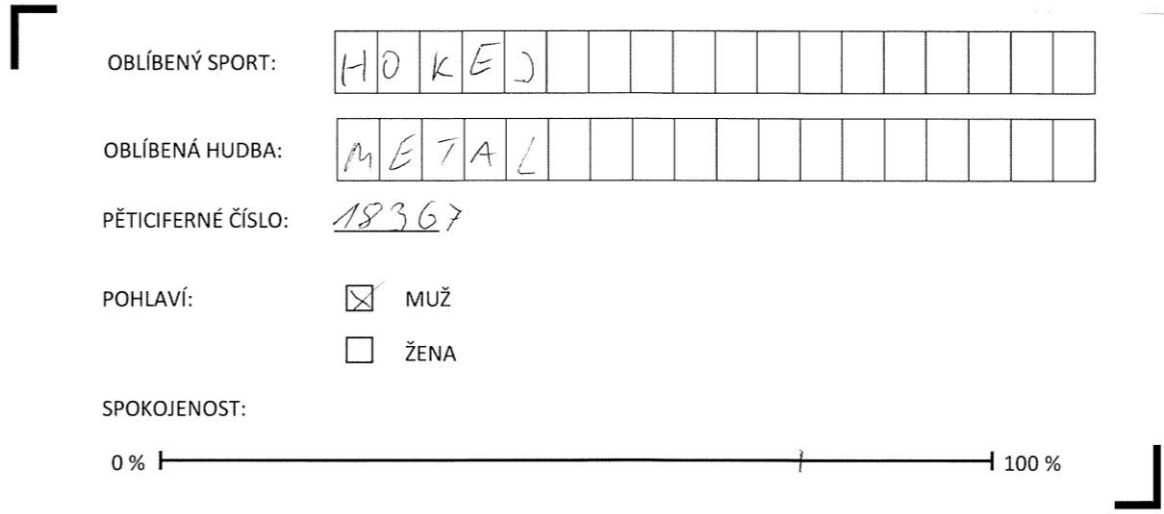
Formulář číslo 1:



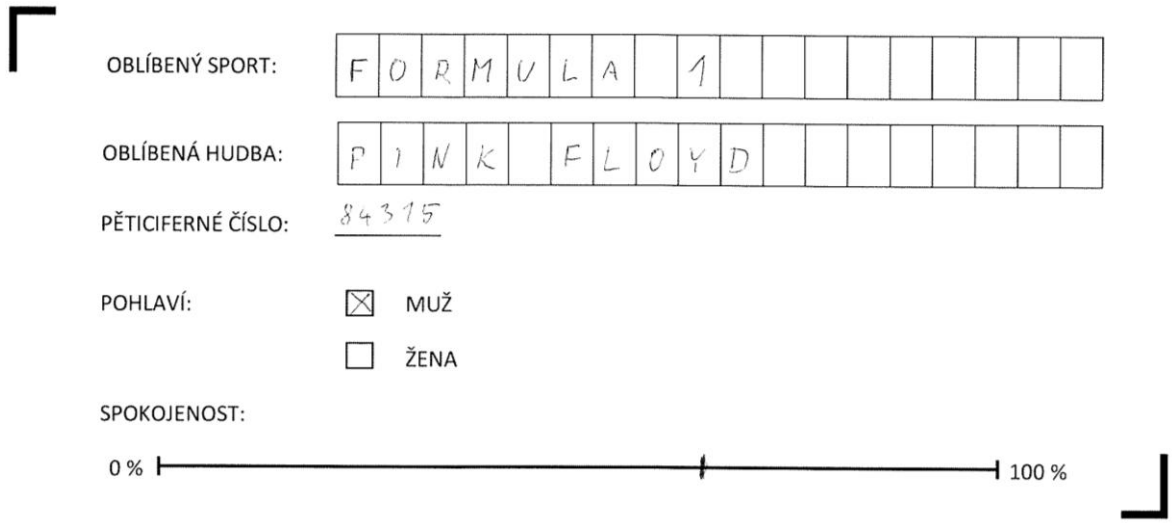
Formulář číslo 2:



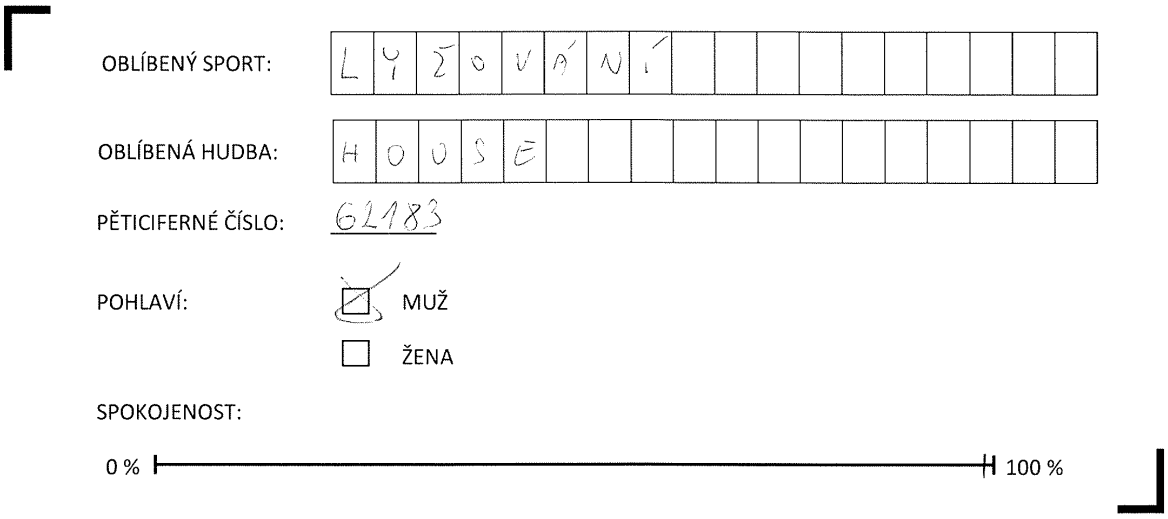
Formulář číslo 3:



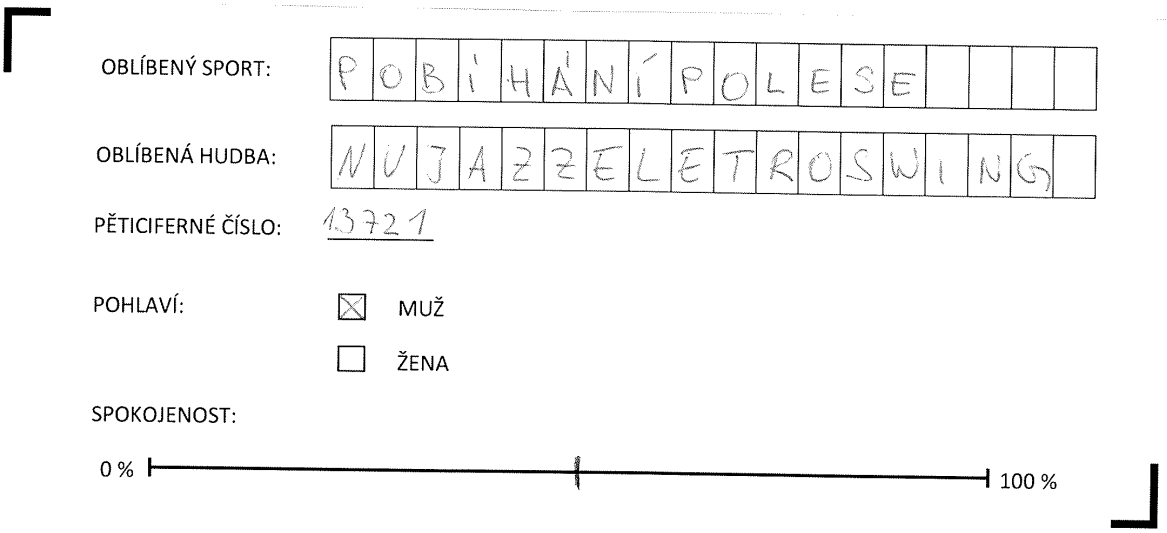
Formulář číslo 4:



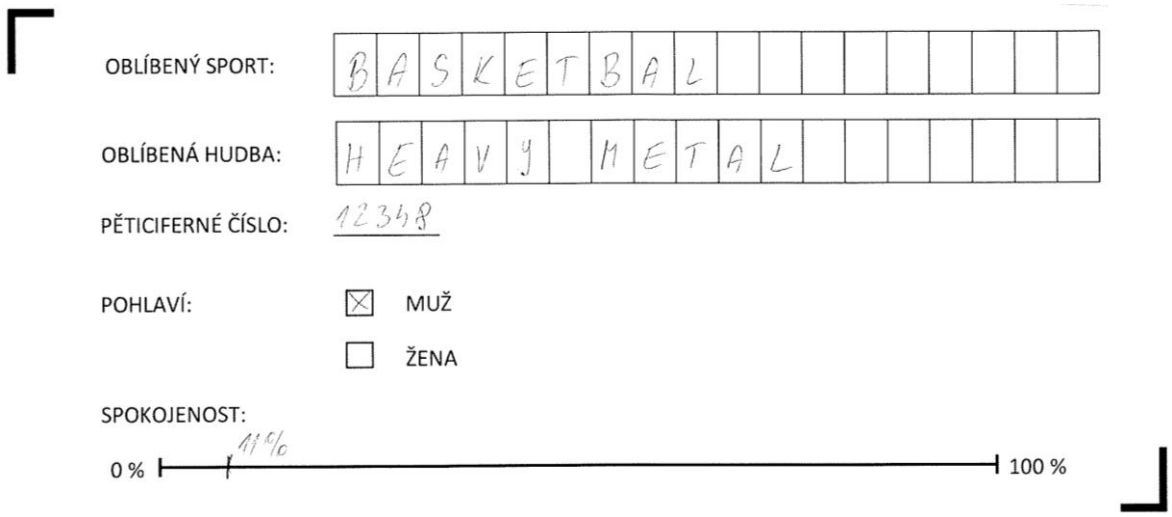
Formulář číslo 5:



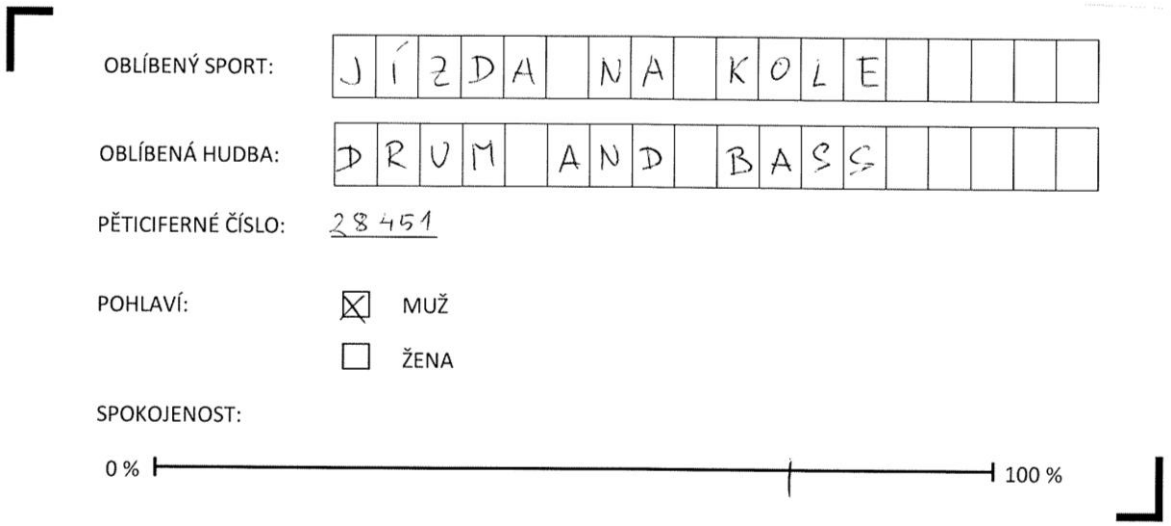
Formulář číslo 6:



Formulář číslo 7:



Formulář číslo 8:



Formulář číslo 9: