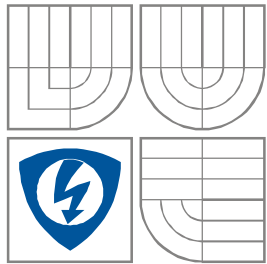


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

OCR MODUL PRO ROZPOZNÁNÍ PÍSMEN A ČÍSLIC

OCR MODULE FOR RECOGNITION OF LETTERS AND NUMBERS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Ján Kapusta

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Ing. Milan Sigmund, CSc.

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Ján Kapusta
Ročník: 2

ID: 78530
Akademický rok: 2009/10

NÁZEV TÉMATU:

OCR modul pro rozpoznání písmen a číslic

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami nejčastěji používanými pro rozpoznávání textu. Porovnejte tyto metody z hlediska výpočetní náročnosti a úspěšnosti rozpoznávání. Navrhněte a naprogramujte metodu pro rozpoznávání písmen a číslic v obrázcích malých rozměrů. Navrženou metodu otestujte na reálných obrazových datech. Porovnejte úspěšnost rozpoznávání u různých druhů obrazových dat.

DOPORUČENÁ LITERATURA:


- [1] BARTOŇEK, D. Počítačová grafika I. Studijní opora. Brno: FAST VUT v Brně, 2005.
- [2] HORÁK, K., KALOVÁ, I., PETYOVSÝ, P., RICHTER, M. Počítačové vidění. Elektronické skriptum. Brno: FIT VUT v Brně, 2008.
- [3] JAN, J. Číslicová filtrace, analýza a restaurace signálů. Brno: FEKT VUT v Brně, 1997.

Termín zadání: 8.2.2010

Termín odevzdání: 21.5.2010

Vedoucí práce: prof. Ing. Milan Sigmund, CSc.

Konzultanti diplomové práce:


prof. Dr. Ing. Zbyněk Raida
předseda oborové rady



UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Ján Kapusta
Bytem: Nová 1, Medzibrod, 97696, Slovensko
Narozen/a (datum a místo): 5. srpna 1986 v Banské Bystrici

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: OCR modul pro rozpoznání písmen a číslic

Vedoucí/ školitel VŠKP: prof. Ing. Milan Sigmund, CSc.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 21. května 2010

.....
Nabyvatel

.....
Autor

ABSTRAKT

Práca rozoberá základné metódy pri rozpoznávaní znakov v obraze. Popisuje celý proces rozpoznávania od prípravy obrazu, spracovania, extrakcie rysov až po priradovacie algoritmy. Slúži ako zhodnotenie súčasne používaných metód a algoritmov používaných pri rozpoznávaní znakov graficky účelovo skresleného alebo upraveného obrazu tzv. „captcha“. Porovnáva metódu založenú na invariantných momentoch a neurónovej sieti ako výslednom klasifikátore a metódu založenú na korelácií vzorov a rozpoznávaných znakov.

KLÍČOVÁ SLOVA

rozpoznávanie textu z obrazu, OCR, segmentácia, extrakcia rysov, klasifikácia, NN, Hu momenty, korelácia, captcha

ABSTRACT

This paper describes basic methods used for optical character recognition. It explains all procedures of recognition from adjustment of picture, processing, feature extracting to matching algorithms. It compares methods and algorithms for character recognition obtained graphically distorted or else modified image so-called „captcha“, used in present. Further it compares method based on invariant moments and neural network as final classifier and method based on correlation between normals and recognized characters.

KEYWORDS

optical character recognition, OCR, segmentation, feature extracting, matching algorithms, NN, Hu moments, correlation, captcha

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

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma OCR modul pro rozpoznání písmen a číslic 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 jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 21.5.2010

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práci prof. ing. Milanu Sigmundovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 21.5.2010

.....

(podpis autora)

OBSAH

Zoznam obrázkov	x
Zoznam tabuliek	xii
Úvod	1
1 História a súčasný stav	2
2 OCR systémy	3
2.1 Spracovanie obrazu.....	4
2.1.1 Prahovanie	5
2.1.2 Odstránenie šumu a vyhladenie hrán.....	6
2.1.3 Segmentácia.....	7
2.1.4 Rotácia.....	7
2.2 Metódy používané pri extrakcii rysov	8
2.2.1 Rozdelenie do pásem	8
2.2.2 Metóda priesečníkov.....	9
2.2.3 Štruktúrálna analýza	9
2.2.4 Metóda invariantných momentov	10
2.2.5 Metóda 2D korelačnej funkcie.....	11
2.3 Metódy pri konečnom rozpoznávaní znakov.....	14
2.3.1 Neurónové siete	14
2.3.2 Metóda zhody	15
2.3.3 Adaboost.....	16
2.3.4 Metóda podporných vektorov.....	17
2.4 Zhodnotenie výpočtovej náročnosti uvedených metód	17
3 Vlastný návrh OCR pre „t-zones“	18
3.1 Analýza obrazových dát.....	18
3.2 Predspracovanie.....	19
3.2.1 Prahovanie, odstránenie „šedého“ prechodu	20
3.2.2 Odstránenie šumu, „čiernych“ kriviek, rekonštrukcia obrazu	20
3.3 Segmentácia.....	22

3.4	Momentová metóda	25
3.4.1	Extrakcia charakteristických rysov v momentovej metóde	25
3.4.2	Konečné rozpoznávanie v momentovej metóde	26
3.4.3	Popis blokov programu momentovej metódy, ovládanie	27
3.4.4	Výsledky a zhodnotenie momentovej metódy	28
3.5	Korelačná metóda	33
3.5.1	Extrakcia charakteristických rysov v korelačnej metóde	33
3.5.2	Konečné rozpoznávanie v korelačnej metóde	33
3.5.3	Popis blokov programu korelačnej metódy, ovládanie.....	34
3.5.4	Výsledky a zhodnotenie korelačnej metódy	37
3.6	Porovnanie výsledkov oboch metód	42
4	Záver	43
	Literatúra	44
	Zoznam symbolov, veličín a skratiek	45
	Zoznam príloh	48

ZOZNAM OBRÁZKOV

Obr. 2.1:	Typické časti spracovania pri OCR	3
Obr. 2.2:	Vstupné obrazy	3
Obr. 2.3:	Príklad postupu pri rozpoznávaní znaku.....	4
Obr. 2.4:	Šedotónový obraz a grafické rozloženie úrovni jasú	4
Obr. 2.5:	Prahovacie funkcie.....	5
Obr. 2.6:	Prahovaný obraz.....	5
Obr. 2.7:	Histogram príslušiaci k obr. 2.6.....	5
Obr. 2.8:	Metódy používané pri extrakcii rysov	9
Obr. 2.9:	Štruktúrna analýza obrazu.....	9
Obr. 2.10:	Základné korelačné vlastnosti 1D funkcií.....	12
Obr. 2.11:	Základné korelačné vlastnosti 2D funkcií.....	13
Obr. 2.12:	Neurón a sieť.....	15
Obr. 2.13:	Príklad algoritmu adaboost	16
Obr. 2.14:	Oddelenie tried.....	17
Obr. 3.1:	Originály generované na „t-zones“	18
Obr. 3.2:	Jednotlivé zložky obrazov.....	19
Obr. 3.3:	Prahované obrazy	20
Obr. 3.4:	Kroky predspracovania	21
Obr. 3.5:	Histogramy.....	23
Obr. 3.6:	Príklady segmentácie	24
Obr. 3.7:	Štruktúra neurónovej siete	26
Obr. 3.8:	Dosiahnuté testovacie výsledky pri tréovaní siete	26
Obr. 3.9:	Príklad klasifikácie.....	27
Obr. 3.10:	Blokový diagram programu momentovej metódy	28
Obr. 3.11:	Graf výskytu rozpoznání v momentovej metóde	30
Obr. 3.12:	Graf pravdepodobností rozpoznania v momentovej metóde	31
Obr. 3.13:	Graf pravdepodobností zobrazenia v momentovej metóde	32
Obr. 3.14:	Blokový diagram programu korelačnej metódy	34
Obr. 3.15:	Hlavné okno spustiteľného programu „OCR_T_zones“	35
Obr. 3.16:	Okno programu zobrazujúce rozpoznávané slovo.....	35

Obr. 3.17: Podiel chyby zobrazených číslic pri neúspešnom rozpoznaní.....	38
Obr. 3.18: Graf výskytu rozpoznaní v korelačnej metóde	39
Obr. 3.19: Graf pravdepodobností rozpoznaní v korelačnej metóde.....	40
Obr. 3.20: Graf pravdepodobností zobrazenia v korelačnej metóde.....	41
Obr. 3.21: Úspešnosť rozpoznaní znakov v momentovej a korelačnej metóde	42
Obr. 3.22: Úspešnosť rozpoznaní slov v momentovej a korelačnej metóde	42

ZOZNAM TABULIEK

Tab. 3.1	Úspešnosť segmentácie pre znaky	24
Tab. 3.2	Úspešnosť segmentácie pre slová	24
Tab. 3.3	Počet výskytu javov v momentovej metóde	29
Tab. 3.4	Úspešnosť klasifikácie slov v momentovej metóde.....	29
Tab. 3.5	Pravdepodobnosť výskytu javov v momentovej metóde.....	30
Tab. 3.6	Pravdepodobnosť rozpoznania v momentovej metóde.....	31
Tab. 3.7	Pravdepodobnosť zobrazenia v momentovej metóde	32
Tab. 3.8	Počet výskytu javov v korelačnej metóde.....	37
Tab. 3.9	Úspešnosť klasifikácie slov v korelačnej metóde	38
Tab. 3.10	Pravdepodobnosť výskytu javov v korelačnej metóde	39
Tab. 3.11	Pravdepodobnosť rozpoznania v korelačnej metóde	40
Tab. 3.12	Pravdepodobnosť zobrazenia v momentovej metóde	41

ÚVOD

OCR je anglickou skratkou *Optical Character Recognition* čo znamená optické rozpoznávanie znakov. Jedná sa teda o automatické rozpoznanie (identifikáciu) grafických znakov (písmen, číslíc, ...) snímaných opticky. V reálnom svete sa počiatky metódy začali vyvíjať už v 30. rokoch minulého storočia. Samozrejme ako sa zvyšoval výpočtový výkon počítačov tak aj samotné metódy naberali na sofistikovanosti. Rozpoznávanie tak môže prebiehať na úrovni znakov, celých slov a dokonca aj spojení slov.

V úplných začiatkoch pre rozpoznávanie číslíc bol požitý mechanický stroj, ktorý fungoval na princípe zdroja svetla, šablóny, rozpoznávaného znaku a fotodetektora. Keď šablóna a skúmaný znak boli prekryté, žiadne svetlo nedopadlo na fotodetektor a stroj vyhodnotil znak za zistený. Avšak v prítomnosti sa na tento spôsob rozpoznávania veľmi nehodí skratka OCR pretože to predstavuje digitálne spracovanie (prahovanie, segmentácia, vyhladenie šumu) a následná klasifikácia znakov obrazu pomocou počítača. Vývoj tohto smeru sa nezastavil ani po tom čo hlavný cieľ bolo napríklad digitalizácia skenovaného alebo odfoteného strojového textu, alebo rozvoj rozpoznávania písmen z rukou písaného písma. Prechod medzi mechanickým a počítačovým je možné vidieť vo vývoji tohto systému v triediacich procesoch v poštových službách, kde bolo nemožné s postupným exponenciálnym nárastom zásielok využívať ľudskú prácu.

V posledných rokoch ako nastal rozvoj internetu a tým aj podnikateľského prostredia ktoré je zamerané hlavne na tento kybernetický priestor, sa „hitom“ stalo využívanie „botov“ na registráciu na rôzne komerčné stránky a „spamovanie“ ostatných užívateľov pomocou elektronickej pošty a podobne, bolo nutné vložiť do procesu registrácie krok kde len človek bude schopný ho prekonať. A preto vznikla autentifikácia rozpoznaním slov, písmen, číslíc z deformovaného, zašumeného, a inak znehodnoteného obrazu, kde „boti“ by nemali mať žiadnu šancu, ale aj to je už minulosťou. Otázka je, či sa môže takýto systém rozpoznávania dostať na vyššiu úroveň, čo sa týka počtu rozpoznaných znakov, ako človek. Budúcnosť ukáže, avšak v takom prípade, že by to bolo možné, taká ochrana pred „botmi“ by stratila zmysel.

Cieľ diplomovej práce je navrhnúť funkčnú metódu pre rozpoznávanie kontrolného obrazca používaného ako autentifikácia pri odosielaní SMS z portálu „t-zones“ s použitím známych aj vlastných metód používaných pri rozpoznávaní podobných obrazcov. Funkčnosť popisuje možnosť využitia metódy v reálnej aplikácii.

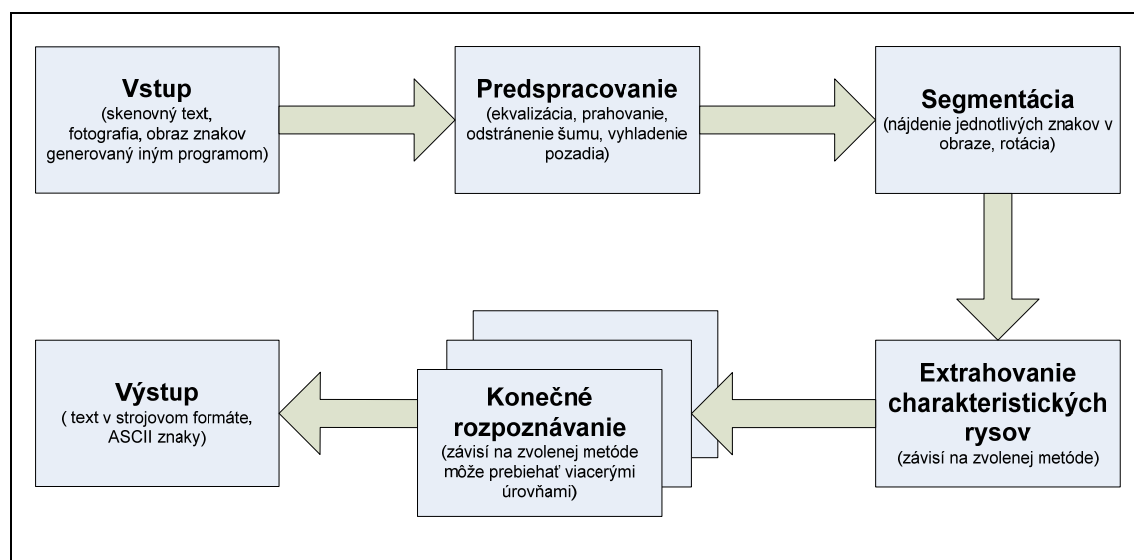
1 HISTÓRIA A SÚČASNÝ STAV

- 1929 Gustav Taushek získal patent na svoj mechanický stroj, ktorý využíval šablóny, fotodetektor, a fungoval tak, že keď správna šablóna a znak boli v zákryte, žiadne svetlo zo zdroja sa nedostalo na fotodetektor
- 1950 David H. Shepard vyvinul systém založený na obrazovej analýze a bol schopný rozoznať niekoľko fontov
- 1955 Prvý komerčne využívaný systém pre čítanie kreditných kariet pre platobné účely
- 1965 Používanie OCR strojov pre triedenie pošty v USA a Veľkej Británii
- 1978 Ray Kurzweil začal predávať komerčný počítačový program, neskôr odkúpi jeho firmu Xerox

V súčasnej dobe je rozpoznávanie strojového textu rôznych fontov takmer vyriešené, pozornosť sa upriamuje na ručne písaný text a graficky znehodnotenú obrázky znakov slúžiace na zabezpečenie proti internetovému pirátstvu.

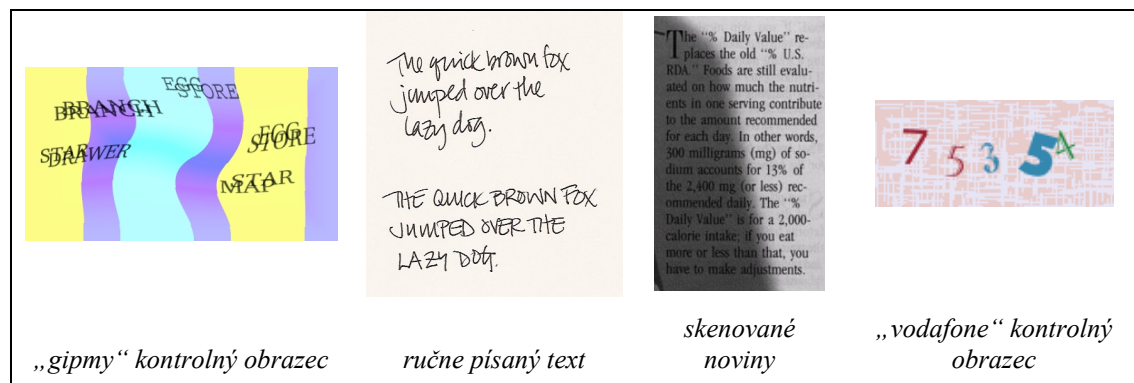
2 OCR SYSTÉMY

Základným atribútom pre automatické rozpoznávanie textu je ukázať (naučiť) program vzory znakov a ich charakteristické rysy aby bol k nim schopný priradiť znaky nachádzajúce sa v obraze. Závisí to hlavne na zvolenej metóde, kde je možné podľa potreby použiť klasifikáciu podľa obsiahnutého slovníka daného jazyka, kde rozpoznávanie prebieha na úrovni slov. Rozpoznávanie teda prebieha „porovnávaním“ charakteristických rysov neznámych znakov zistených v obraze s charakteristickými rysmi vzorov jednotlivých znakov.



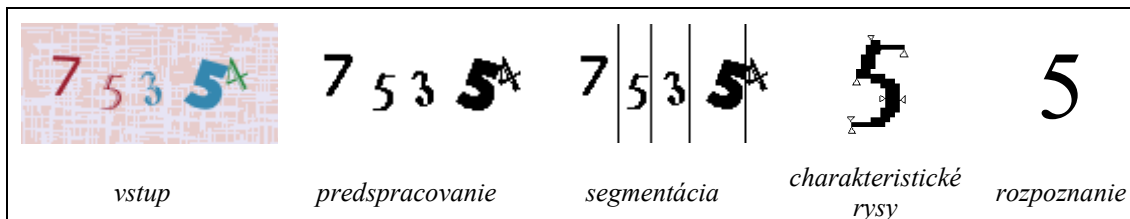
Obr. 2.1: Typické časti spracovania pri OCR

Každá OCR metóda má spoločný cieľ no takmer všetky čiastkové procesy sa líšia, to sa samozrejme týka aj vstupných dát. Kde to môže byť rôzne naskenovaný text, písaný text rukou, kontrolné obrazy generované pre zabezpečenie, a už v pomerne veľkej odlišnosti vstupných dát je možné zhodnotiť že jedna univerzálna metóda nie je. Jednotlivé bloky charakterizujú základné kroky, ktoré vo väčšine súčasných programov prebiehajú v zobrazenom poradí podľa obr. 2.1.



Obr. 2.2: Vstupné obrazy

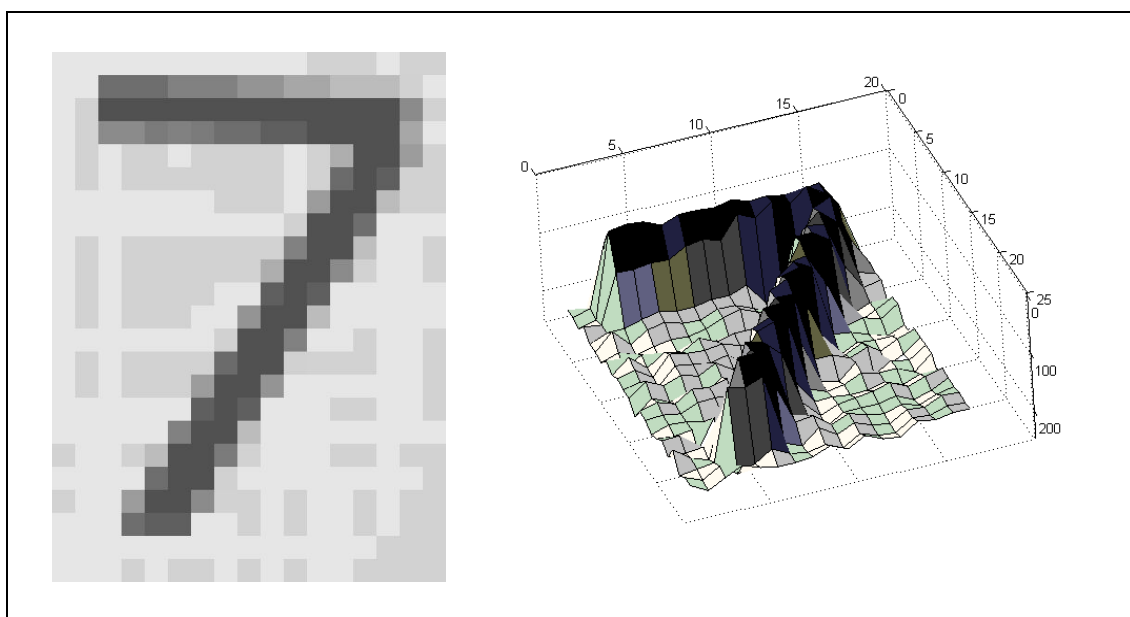
Vstup môže byť už z predtým spomenutých zdrojov, avšak väčšinou je to obrazový formát ďalej spracovávaný podľa kritérií, ktoré sú zvolené ako najvhodnejšie pre danú metódu. Musí sa však počítať s tým, že pri každom kroku spracovania (obr. 2.3) sa informácia obsiahnutá v pôvodnom obraze znižuje. Je to cena za priblíženie obrazu vnímaného ľudskými zmyslami počítačovému videniu.



Obr. 2.3: Príklad postupu pri rozpoznávaní znaku

2.1 Spracovanie obrazu

Hlavný cieľ spracovania je úprava obrazu v jeho digitálnej podobe tak, aby bol čo najpriateľnejší pre ďalšie operácie postupu rozpoznávania znakov v ňom obsiahnutých. Obraz je tvorený obrazovými elementmi nazývanými pixely. Podľa typu obrazu, a farebnej alebo šedotónovej hĺbky jednotlivého pixelu, je možné pozeráť sa na obraz ako na maticu o rozmeroch obrazu kde každý pixel predstavuje napríklad diskretnú hodnotu jasů v rozmedzí 0 – 255 pri „8bitovej“ bitmape.

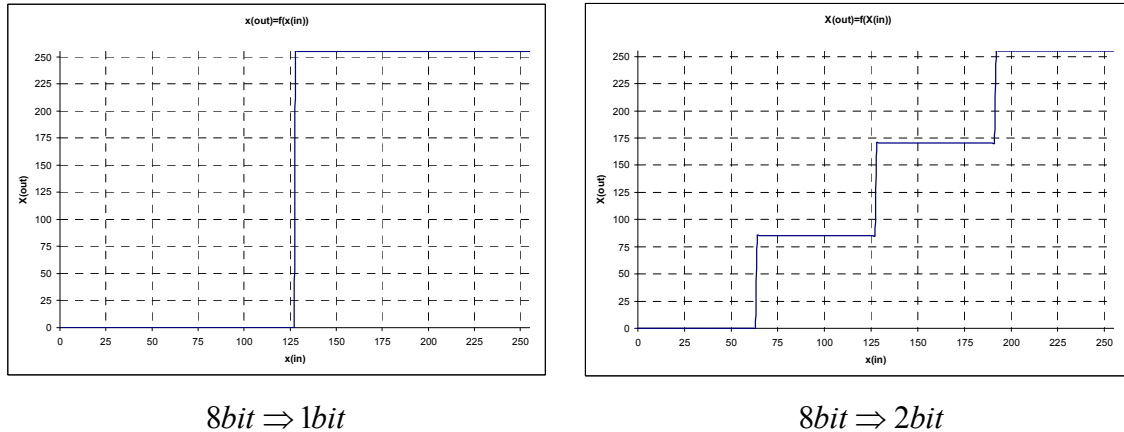


Obr. 2.4: Šedotónový obraz a grafické rozloženie úrovni jasů

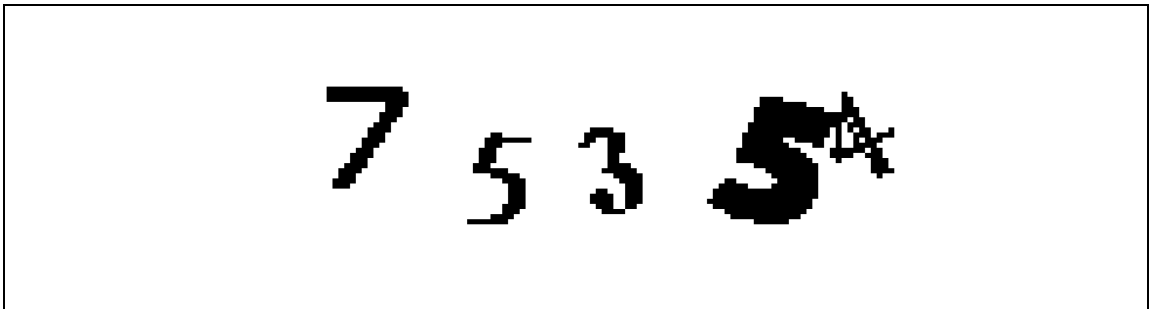
Pri farebnom obraze, uloženom pomocou troch farebných zložiek, a to červenej, zelenej a modrej zložky, je každá farba vyjadrená pomocou intenzity danej farebnej zložky. To znamená napríklad, že fialová farba má súradnice $R = 255, G = 0, B = 255$, čierna farba $R = 0, G = 0, B = 0$, a biela $R = 255, G = 255, B = 255$.

2.1.1 Prahovanie

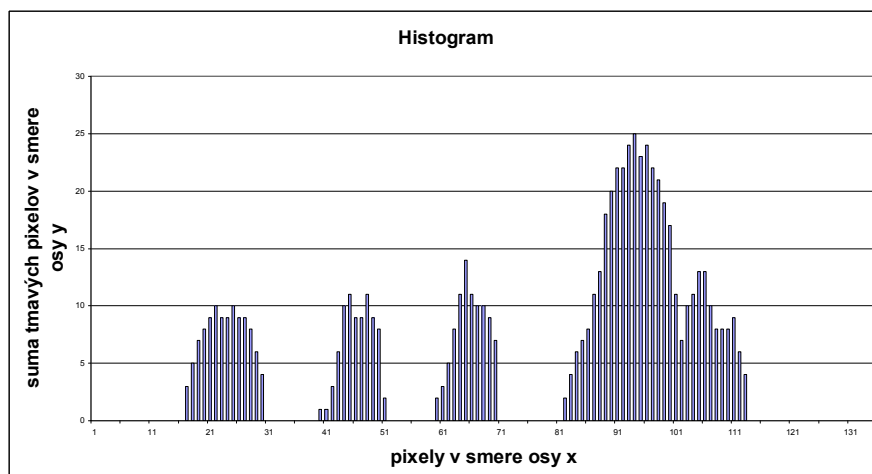
Prahovanie je proces, kde je sa určí hranica (úroveň), podľa ktorej sa pixelom priradí maximálna alebo minimálna úroveň jasu. Je to vlastne vymedzenie pixelov, ktoré majú informačnú hodnotu. Predpokladom je, že obraz už nie je zašumený a pozadie odstránené (obr. 2.6). Histogram na obr. 2.7 zobrazuje sumu čiernych pixelov vo vertikálnej ose. Príklady prahovacích funkcií sú na obr. 2.5.



Obr. 2.5: Prahovacie funkcie



Obr. 2.6: Prahovaný obraz



Obr. 2.7: Histogram príslušiaci k obr. 2.6

2.1.2 Odstránenie šumu a vyhladenie hrán

Tento krok je dôležitý pre prípravu obrazu na ďalšie spracovanie. Odstránenie šumu, vyhladenie hrán a eliminovanie vplyvu pozadia na obsiahnuté znaky. Je možné sem zahrnúť aj proces tzv. normalizácie, kde veľkosť znaku upravíme tak, aby jeho veľkosť bola upravená podľa požiadaviek na extrakciu rysov. Tento krok je veľmi dôležitý, pretože je na začiatku reťazca a môže ovplyvniť v konečnom dôsledku veľmi podstatné vlastnosti obrazu, potrebné napríklad pri následnej segmentácii.

V prípadoch, ktorými sa zaoberá táto práca, prirodzený šum neexistuje. Prirodzeným šumom je myslený šum, ktorý vzniká na nedokonalostiach záznamovej techniky, kódovaním alebo v prenosovom kanáli. Tu je ale zámerne generovaný pre znemožnenie rozpoznania, alebo vznikne pri postupnom spracovaní (osamotené biele pixely, malé fragmenty pri prahovaní, segmentácii) V oblasti spracovania obrazových dát poznáme hlavné druhy šumu:

- Gaussov šum
- Výstrelový šum

Gaussov šum vzniká tepelným pôsobením materiálov a elektrických obvodov v zariadeniach spracúvajúcich analógovú (nečíslicovú) obrazovú informáciu.

Výstrelový šum môže vznikať tiež v prenosovom reťazci, no skôr v zariadeniach digitálneho charakteru napr. CCD snímač. Je často umelo pridaný a používaný zamedzeniu rozpoznania kontrolných obrazov. Pre jeho odstránenie sa používajú filtre:

- Lineárne
 - Dolná priepust
 - Gaussov filter s konvulučnou masku
 - Priemerovanie
- Nelineárne
 - Mediánový filter
 - Prahovanie vlnkových koeficientov

Vyhladenie hrán je veľmi podobná operácia ako odstránenie šumu. Ostrá štruktúra hraníc je vyhladená (rozmazaná). V tejto práci je v podstate použitý mediánový filter pre „Ibitovú“ farebnú hĺbku obrazu (biela, čierna). Použitá maska filtra má rozmer 3x3 pixely a vyhodnotí sumu bielych pixelov v maske. Následne podľa nastaveného prahu nahradí novou hodnotou, hodnotu centrálného pixelu masky. Rovnako je aplikovaná jeho funkcia aj pre vyhladenie ostrých hrán.

2.1.3 Segmentácia

Segmentácia je dôležitou a náročnou časťou, kde sa už špecifikujú jednotlivé znaky. Hlavná úloha segmentácie spočíva v oddelení objektov obsiahnutých v celom obraze na základe kritérií, ktoré špecifikujú daný objekt.

Ako je možné vidieť na obr. 2.3, prvé 3 znaky je možné bez problémov lokalizovať, pri posledných dvoch je nutné uskutočniť ďalší proces, alebo jednoducho lokalizovať tieto dva znaky ako celok. Následne vykonať rotáciu, vypočítať náležiaci histogram a lokalizovať minimá. Pomerne jednoduchá segmentácia prebieha pri rozoznávaní znakov v strojovom písanom texte, kde sú presne dané riadky a medzery medzi písmenami. Metódy využívané pri segmentácii:

- Segmentácia prahovaním

»závisí na určení prahu z histogramu obrazu. Histogram môže zobrazovať ako, sumu pixelov v závislosti na rovinnej súradnice obrazu obr. 2.7, tak aj sumu v závislosti na hodnote ich jasu. Nedokonalosť metódy je možné nájsť hneď pri segmentácii obrazov z „t-zones“, kde sa v oboch prípadoch histogramu hranice znakov prekrývajú.

- Detekcia hrán

»využíva rôzne gradientné operátory (Laplaceov, Sobelov). Gradientný operátor je závislý na skokových zmenách rozloženia jasu, prípadne farby v obraze. Problémy nastávajú pri výskyte hrán, kde by nemali byť a naopak, prípadne využitie gradientného operátora pre zmeny farieb tam, kde je výrazná zmena jasu a naopak.

- Nárastom homogénnej oblasti

»znamená členenie obrazu do súvislých homogénnych oblastí. Proces môže prebiehať smerom od rozčlenenia obrazu do veľkého množstva malých oblastí, nasledovaný spojovaním (podľa konkrétneho kritéria) do menšieho množstva veľkých oblastí. Rovnako môže prebiehať smerom opačným.

- Korelácia so vzorom

»je metóda poskytujúca výsledky nielen pri segmentácii. Základ je nájdenie korelačnej vlastnosti medzi vzorom a obrazom kde je vzor hľadaný. Nevýhodou je vysoká závislosť na otočení, veľkosti objektov, úroveň šumu.

2.1.4 Rotácia

Rotácia, v tomto prípade v 2D priestore prebieha pre každý pixel jednotlivo podľa počiatku súradnicového systému o uhol α . Rotáciu reprezentuje násobenie $A \cdot R$, kde $A = [x, y]$ sú súradnice bodu a

$$R = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}. \quad (2.1)$$

Podobne je možné uskutočniť aj zmenu veľkosti („scaling“) avšak posun nie. Preto sa súradnice bodu $A = [x, y]$ prevedú do homogénneho tvaru na $A' = [xw, yw, w]$, kde

w je váha bodu (obvykle volíme váhu $w = 1$) a touto váhou sa musia vynásobiť zvyšné súradnice x a y . Transformačná matica pre rotáciu je

$$R' = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.2)$$

a nové súradnice bodu $A'' = [x'', y'', w'']$ sa spočítajú ako $A'' = A' \cdot R'$. Pre použitie rotácie okolo ľubovoľného streda, musíme použiť postup posun, rotácia, posun. Matica pre posun je

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \delta_x & \delta_y & 1 \end{bmatrix}, \quad (2.3)$$

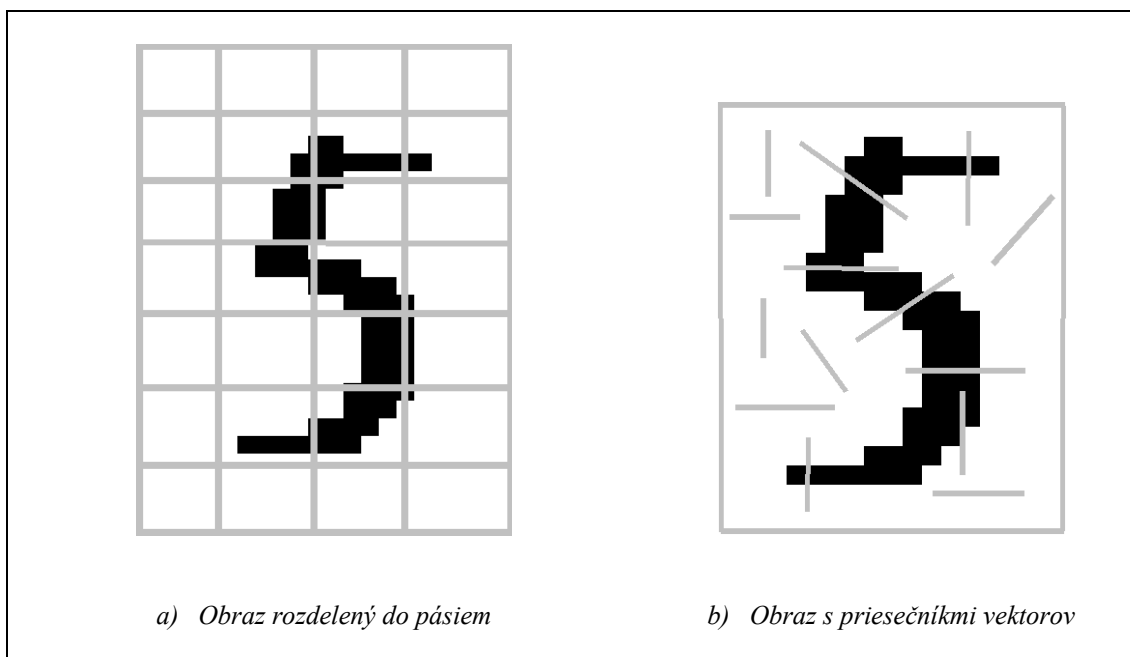
kde koeficienty δ_x a δ_y sú dané vektorom posunutia $\vec{p} = [\delta_x, \delta_y]$, ktorý udáva ako ďaleko a ktorým smerom bude bod posunutý. [6]

2.2 Metódy používané pri extrakcii rysov

Extrakcia rysov je proces zistenia charakteristických rysov znakov obsiahnutých v obraze. Najlepší prípad je vtedy, keď rozdielnosť rysov pre jednotlivé triedy znakov je čo najodlišnejšia. Metódy je potrebné voliť s ohľadom na daný prípad. Samozrejme nie je možné vytvoriť jeden silný, obecný a univerzálny extraktor rysov. V prípade že by taký existoval výsledný klasifikátor by bol triviálny. V tomto prípade je snaženie vytvoriť extraktor istou simuláciou myslenia ľudského mozgu. Je to až neuveriteľné v akom množstve hladín dokáže myslieť, v prípade potreby meniť celý obecný popísaný reťazec rozpoznávania. Práca ľudského mozgu je vo svojej podstate geniálna, pretože jeho hlavná vlastnosť spočíva v maximálnej efektívite myslenia. Inteligencia znamená maximálny výkon pri čo najnižšom vynaložení energie. Pri triedení jabĺk a hrušiek si nikto nebude všimáť tvar keď rozhodujúcim faktorom je farba (ak nepredpokladáme zelenú odrodu jabĺk). Naopak pri triedení rýb, bude hlavným faktorom ich veľkosť (samozrejme podľa prípadu aj farba šupín, tvar, ...) Najpodstatnejšia je znalosť domény a proces získania rysov.

2.2.1 Rozdelenie do pásem

Obrázok je rozdelený do pásiem (zón) obr. 2.8a v každom pásme sa spočíta hustota tmavých miest a to poslúži ako rys. Táto metóda je vhodná pre použitie v strojovom texte, kde rozdielnosť jednotlivých znakov od normálov je minimálna. Problémy môžu nastať pri rozpoznávaní šikmého textu (kurzíva). Čím je jemnejšia sieť mriežky, tým sú rysy presnejšie, avšak pri rozpoznávaní znaku môže nastať ďalší problém a to keď rozpoznávajúci znak bude zdeformovaný alebo inak odlišujúci sa od originálu. Pri tejto metóde závisí presnosť na veľkosti znaku, pootočení.



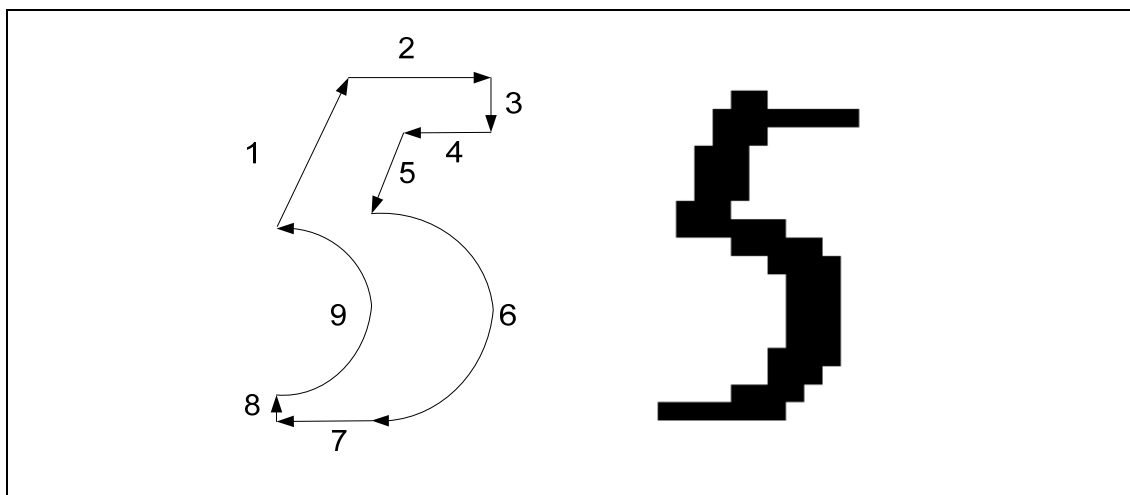
Obr. 2.8: Metódy používané pri extrakcii rysov

2.2.2 Metóda priesečníkov

Metóda priesečníkov hľadá počet priesečníkov s vopred zvolenými vektormi, je pomerne často využívaná komerčnými programami pre rozpoznávanie textu. Aj tu je možné vidieť závislosť na rotácii a veľkosti znaku obr. 2.8b.

2.2.3 Štruktúrálna analýza

V štruktúrálnej analýze je popisovaná hlavne geometrická a topologická štruktúra znaku. Hľadajú sa geometrické prvky ako sú úsečky, oblúky, koncové body (obr. 2.8.) Voči metódam priesečníkov a metóde rozdelení do pásiem, je viac tolerantná ako voči šumu tak aj voči skresleniu a hrúbke znaku.



Obr. 2.9: Štruktúrálna analýza obrazu

2.2.4 Metóda invariantných momentov

Základné a centrálné geometrické momenty sú deskriptory popisujúce jasové rozloženie hodnôt objektov, samy o sebe však nie sú nezávislé na obecné afinnej transformácii. Z tohto dôvodu sa používajú pre kompozíciu zložitejších štruktúr nazvaných momentové invarianty, ktorých je celá rada pre široký rozsah úloh. Najstaršia a najznámejšia je sada siedmich momentových invariantov (tzv. Hu momenty) I_1 až I_7 zostavených z hľadiska nezávislosti na rotácii, translácii (danou presnou segmentáciou objektu) a zmene merítka (daná normalizáciou geometrických momentov): Momentové invarianty si môžeme predstaviť ako funkcie momentov obrazu. Momenty

$$\eta_{pq} = \iint P_{pq}(x, y) f(x, y) dx dy, \quad (2.4)$$

kde $(p+q)$ je rád momentu, sú priemtom obrazovej funkcie $f(x, y)$, ohraničenej $R \times R$ do množiny polynómov. $\{P_{pq}(x, y)\}$. Z ôsmich momentových invariantov (sedem Hu momentov a jeden ďalší) je známe, že momenty sú nanajvyšš 3. rádu.

$$I_1 = \eta_{20} + \eta_{02} \quad (2.5)$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \quad (2.6)$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.7)$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (2.8)$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.9)$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\ + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) - \eta_{03}^2 \quad (2.10)$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \\ - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.11)$$

$$I_8 = (\eta_{20}\eta_{02}) - (\eta_{11})^2 \quad (2.12)$$

Pre rozpoznávanie objektov sa využíva skupina vybraných momentových invariantov s vysokou diskriminabilitou vzhľadom k spracovanej úlohe, často v kombinácii s niektorými z jednoduchších radiometrických alebo fotometrických deskriptorov obrazu. [8]

2.2.5 Metóda 2D korelačnej funkcie

Korelácia znamená vzájomný vzťah medzi dvoma veličinami, funkciami. Korelačné funkcie sú vo veľkom využívané v spracovaní signálov, kde sa vlastnosti mnohých signálov a ich možnosť vysielania a spracovania, upravujú napríklad tak, aby výsledok autokorelačnej funkcie bol maximálny (napríklad jedno ostré maximum v čase), naopak zo všetkými ostatnými signálmi minimálny (charakter šumu). V programovom jazyku Matlab[®] je možné využiť funkciu `xcorr`, ktorá je popísaná nižšie.

Jednorozmerná korelačná funkcia $r_{x,y}(\tau) = f(x,y)$ tzv. 1D udáva vzťah medzi veličinami, funkciami $x, y = f(t)$, ktoré sú závislé na jednej premennej t v závislosti na vzájomnom posunutí τ . Keďže je jedno či jedna funkcia predbieha a druhá spomaľuje je možné ju definovať

$${}_{1D}r_{x,y}(\tau) = \int_{-\infty}^{+\infty} x(t) \cdot y(t-\tau) = \int_{-\infty}^{+\infty} x(t+\tau) \cdot y(t), \quad (2.13)$$

kde v prípade diskrétného času $T = \{0,1,2,\dots,M-1\}$ a podmienky $D(x,y) \in \{0,1,2,\dots,M-1\}$, prejde integrál na sumu

$${}_{1D}R_{x,y}(\tau) = \sum_{t=0}^M x(t) \cdot y(t-\tau) = \sum_{t=0}^M x(t+\tau) \cdot y(t), \quad (2.14)$$

Pri zámene poradia x, y platí rovnako

$${}_{1D}R_{y,x}(\tau) = \sum_{t=0}^M y(t) \cdot x(t-\tau) = \sum_{t=0}^M y(t+\tau) \cdot x(t), \quad (2.15)$$

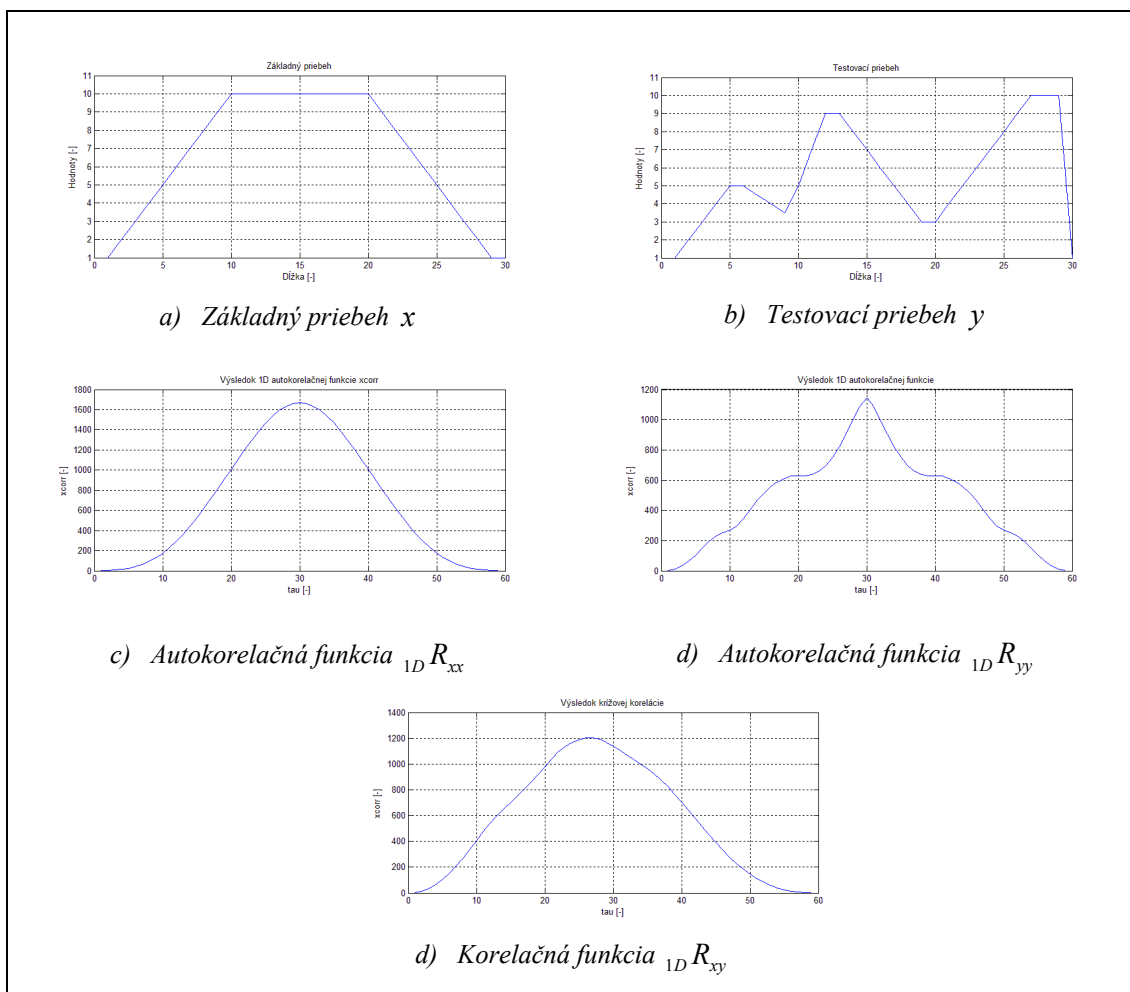
a pri autokorelačnej funkcii platí

$${}_{1D}R_{x,x}(\tau) = \sum_{t=0}^M x(t) \cdot x(t-\tau) = \sum_{t=0}^M x(t+\tau) \cdot x(t), \quad (2.16)$$

Výsledok korelačnej funkcie nezávisí len na „tvare“ signálu, ale aj na veľkosti hodnôt funkcie (obr. 2.9), kde je vidieť že maximum autokorelačnej funkcie pre základný a testovací priebeh závisí na sume (pri diskrétnych hodnotách dĺžky) alebo integrále (pri spojitých hodnotách dĺžky) oboch korelačných priebehov. Je dôležité si všimnúť osovú symetriu v osi maxima pre výsledky autokorelačných funkcií, pretože z (2.16) vyplýva

$${}_{1D}R_{x,x}(\tau) = R_{x,x}(-\tau). \quad (2.17)$$

Výraznejšie maximum pri priebehoch s väčšou diferenciou od strednej hodnoty. Samozrejme je na prvý pohľad z (2.16) jasné, že maximum autokorelácie bude pre posuv $\tau = 0$. Výsledky funkcie `xcorr` hovoria niečo iné. Problém je v tom, že funkcia automaticky posunie koniec jednej prvej funkcie na začiatok druhej, matematicky



Obr. 2.10: Základné korelačné vlastnosti 1D funkcií

$${}_{1D}R_{x,y}(\tau) = \sum_{t=0}^M x(t-M+\tau) \cdot y(t) = \sum_{t=0}^M x(t-M) \cdot y(t-\tau), \quad (2.18)$$

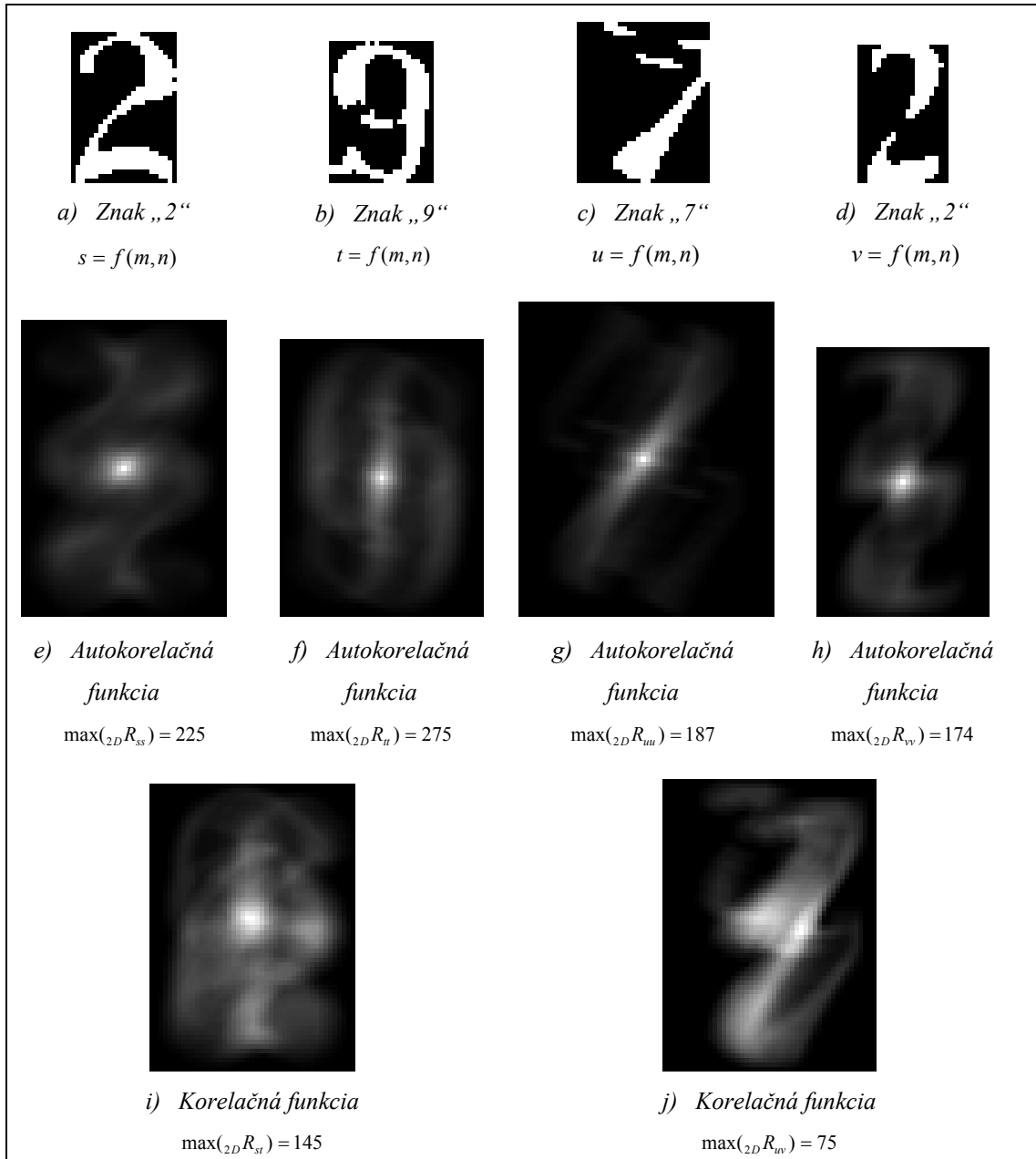
Dvojrozmerná korelačná funkcia $r_{x,y}(\tau) = f(x,y)$ tzv. 2D udáva vzťah medzi veličinami, funkciami $x,y = f(m,n)$, ktoré sú závislé na dvoch premenných m,n v závislosti na vzájomnom posunutí τ,σ . Funkcie dvoch premenných x,y reprezentuje napríklad dvojrozmerný obraz. Pre čisto reálny obor hodnôt $H(x,y) \in R$, v prípade diskretných hodnôt $m,n = \{0,1,2,\dots,M-1\}$ a podmienky $D(m,n) \in \{0,1,2,\dots,M-1\}$, je 2D korelačná funkcia popísaná

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

$${}_{2D}R_{x,y}(\tau,\sigma) = \sum_{m=0}^M \sum_{n=0}^N x(m+\tau,n+\sigma) \cdot y(m,n) \quad (2.14)$$

Korekcia pre funkciu `xcorr2`

$$\begin{aligned}
{}_{2D}R_{x,y}(\tau, \sigma) &= \sum_{m=0}^M \sum_{n=0}^N x(m-M, n-N) \cdot y(m-\tau, n-\sigma) \\
{}_{2D}R_{x,y}(\tau, \sigma) &= \sum_{m=0}^M \sum_{n=0}^N x(m-M+\tau, n-N+\sigma) \cdot y(m, n)
\end{aligned}
\tag{2.14}$$



Obr. 2.11: Základné korelačné vlastnosti 2D funkcií

Príklady 2D korelácie je možné vidieť na obr. 2.10. Pre úplnosť treba dodať, že úroveň jasovej každej korelačnej funkcie bol upravený tak, aby zobrazil výsledok v plnom jasovom rozsahu. Maximá autokorelačných funkcií sú výrazné. Rovnako výkon jasovej ohraničeného blízkeho okolím k stredu funkcie (nulový posun) bude odlišný ako v prípadoch ostatných korelácií.

2.3 Metódy pri konečnom rozpoznávaní znakov

Základným cieľom je navrhnuť systém, ktorý pomerne spoľahlivo priradí vzor a znak (pri zachovaní istej pravdepodobnosti) a to aj pri možných odlišnostiach charakteristických rysov normálu a znaku. Ďalšia dôležitá vlastnosť je schopnosť učenia, ktorej cieľom je zvýšiť počet správne klasifikovaných znakov.

2.3.1 Neurónové siete

Neurónová sieť je vzájomné prepojenie jednotiek neurónov. Neurón je elementárna jednotka pre spracovanie informácie, analógia s hradlom v integrovaných obvodoch. Neurónová sieť, tak ako to už vyplýva z názvu, sa prirodzene nachádza v prírode. Sieť je zložená z jednotlivých neurónov, z ktorých každý má niekoľko vstupov (*dendridov*) a jeden výstup (*axon*). Neurón funguje tak, že reaguje na veľkosť podráždenia na svojich dendridoch, presnejšie povedané na sume podráždenia od všetkých dendridov dohromady. Prirodzene každý neurón má svoju prahovú hodnotu, podľa ktorej je na axone výstup logická 1 alebo 0. Je to len prirovnanie, pretože nejde čisto len o binárny obvod ale o komparátor s prevodnou charakteristikou a sumátorom na vstupe. Ak je potrebné signál upraviť (zoslabiť, invertovať) deje sa to na prechode medzi axonom a dendritom, toto spojenie má názov *synapsia*.

Keďže veľkosť signálov sa charakterizuje reálnymi číslami, je možné neurónovú sieť popísať matematicky. Matematická schéma je na obr. 2.12a. Výstup neurónu je

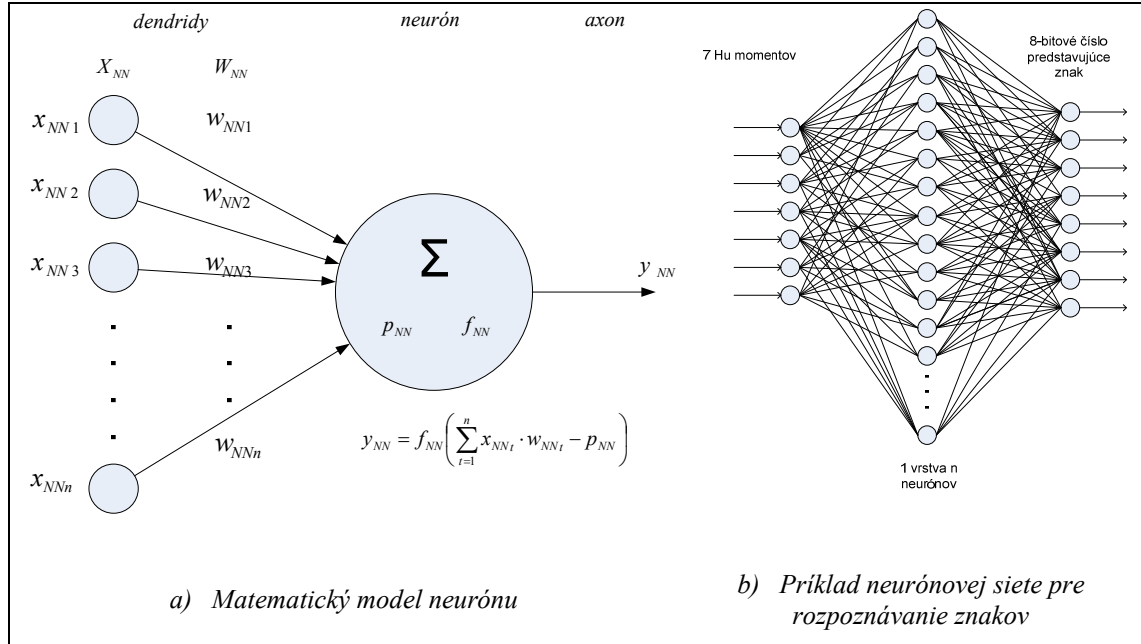
$$y_{NN} = f_{NN} \left(\sum_{t=1}^n x_{NNt} \cdot w_{NNt} - p_{NN} \right), \quad (2.13)$$

kde vektor $X_{NN} = (x_{NN1}, x_{NN2}, x_{NN3}, \dots, x_{NNn})$ popisuje vstupy neurónu z výstupov iných neurónov, vektor $W_{NN} = (w_{NN1}, w_{NN2}, w_{NN3}, \dots, w_{NNn})$ popisuje zoslabenie, f_{NN} je prenosová funkcia (sigmoida), p_{NN} je prah neurónu. Invertovanie signálu sa deje na úrovni synapsií. Sieť neurónov, kde každá vrstva neurónov používa pre vstup výstup z predchádzajúcej vrstvy sa nazýva sieť s dopredným šírením feed-forward. Takáto sieť má konečnú dĺžku odozvy a nemá spätnú väzbu, preto sa nedokáže učiť. U sietí so spätnou väzbou sa môže stať, že vzruch sa môže šíriť stále dookola, prípadne zväčšovať amplitúdu, avšak spätná väzba prináša pamäť.

Učenie siete môže byť „s učiteľom“ a „bez učiteľa“. Učenie siete „s učiteľom“ prebieha tak, že sieť porovnáva svoj výstup s výstupom učiteľa a následným nastavovaním váh synapsií znižuje rozdiel medzi skutočným a učiteľovým výstupom. Priestor definovaný koeficientmi v matici je obrovský, preto je nutné použiť algoritmus efektívne hľadajúci správne hodnoty koeficientov. Učenie siete „bez učiteľa“ nemá žiadny vzor alebo výstup učiteľa, ktorý by mohol považovať za hodnotu pravú, alebo správnu. Algoritmus pracuje tak, že vo výstupe hľadá vzorky s určitými vlastnosťami podľa korelácie.

Po nastavení „optimálnych“ koeficientov sa sieť nachádza vo vybavovacej fázi, kde už koeficienty zostanú nezmenené. Na vstup sa jednoducho privedú dáta a počíta sa odozva siete na výstupe. Samozrejme ak je v sieti spätná väzba musí sieť počítať tak

dlho, až kým výstup skonverguje. Deteguje sa rozdiel vo výsledku s predchádzajúcim výpočtom, a keď je tento rozdiel malý výpočet sa ukončí a výsledok sa prehlási za správny. Môže sa však stať že výstup bude konvergovať k nekonečne vysokým hodnotám, a tým pádom sa odozva nedá spočítať. [9]



Obr. 2.12: Neurón a sieť

2.3.2 Metóda zhody

Metóda alebo skupina metód ktorá je založená na výpočte euklidovskej vzdialenosti medzi vektorom rysov skúmaného znaku a vektorom popisujúcim triedu. Pre n -dimenzionálny priestor a dva vektory

$$\vec{P}_m = (p_{m1}, p_{m2}, p_{m3}, \dots, p_{mn}), \vec{Q}_m = (q_{m1}, q_{m2}, q_{m3}, \dots, q_{mn}) \quad (2.14)$$

tohto priestoru platí, že euklidovská vzdialenosť je

$$E_m(\vec{P}_m, \vec{Q}_m) = \sqrt{(p_{m1} - q_{m1})^2 + (p_{m2} - q_{m2})^2 + \dots + (p_{mn} - q_{mn})^2}$$

$$E_m(\vec{P}_m, \vec{Q}_m) = \sqrt{\sum_{i=1}^n (p_{mi} - q_{mi})^2} \quad (2.14)$$

Metóda zhody je použiteľná pri klasifikácii, pri ktorej sú jednotlivé extrahované rysy dobre separované v n -dimenzionálnom priestore, a vlastná klasifikácia spočíva v porovnaní euklidovskej vzdialenosti klasifikovaného znaku a euklidovskej vzdialenosti a vopred determinovanými triedami.

2.3.3 Adaboost

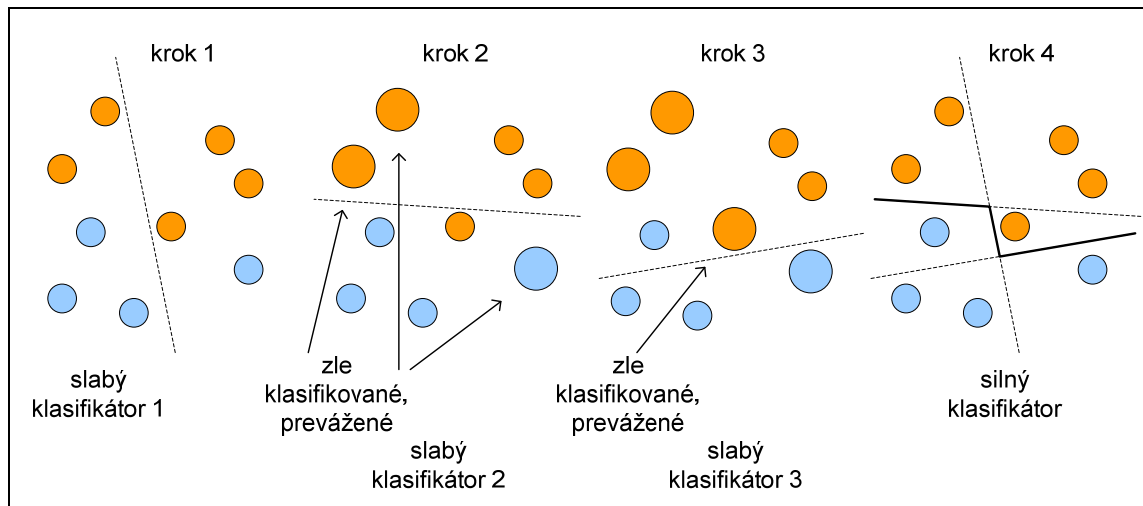
Adaboost je učiaci sa algoritmus, ktorý kombinuje rozhodnutie niekoľkých jednoduchých klasifikátorov a dosahuje tak lepších výsledkov, ako by bolo možné dosiahnuť pri použití klasifikátorov samotných. Názov vznikol z „*adaptive boosting*“ umožňuje teda pridávanie slabých klasifikátorov až do dosiahnutia požadovanej chyby klasifikátoru.

Jeho vstup je tréningová množina, a výstup klasifikátorov do dvoch tried. Výsledný klasifikátor $H(x)$ je lineárna kombinácia slabých klasifikátorov $h_t(x)$

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right). \quad (2.15)$$

Algoritmus učenia adaboost-u prebieha v cykle. V každom cykle je potrebné prejsť nasledujúce kroky:

- 1.» nájsť najlepší (jeho vážená chyba je na tréningovej množine čo najmenešia) slabý klasifikátor pri danom odvážení D_t tréningových dát, kde D_t je váha.
- 2.» overiť, že chyba tohto klasifikátoru je menšia ako 0,5
- 3.» spočítať koeficient slabého klasifikátora v lineárnej kombinácii $H(x)$
- 4.» aktualizovať váhy D_t , to spôsobí to, že váha zle klasifikovaných meraní sa zväčší a váha dobre klasifikovaných meraní sa zmenší, to má za následok, že v nasledujúcom kroku bude hľadaný slabý klasifikátor, ktorý musí lepšie klasifikovať zlé merania. [5]

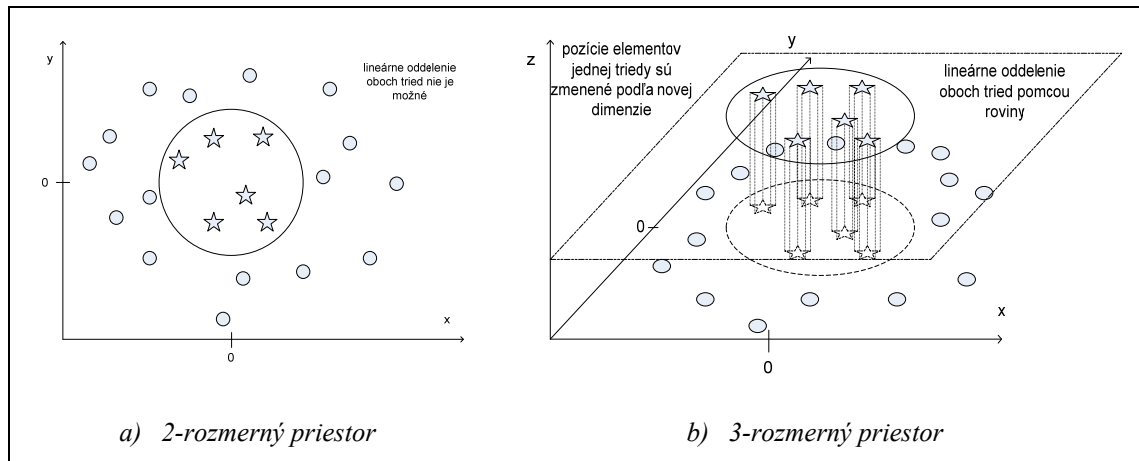


Obr. 2.13: Príklad algoritmu adaboost

Adaboost je vhodné použiť v prípadoch, kde diskriminabilita charakteristických rysov nie je zaručená, môže sa meniť, a aj samotná váha jednotlivých rysov nie je rovnaká, v podstate je hľadaný silný klasifikátor z množstva slabých. Vo veľkej miere sa algoritmus adaboost využíva v reálnych aplikáciách.

2.3.4 Metóda podporných vektorov

K alternatíve neurónovým sieťam patrí relatívne nová metóda podporných vektorov (support vector machine), ktorá tvorí kategóriu tzv. jadrových algoritmov (kernel machine). Neurónové siete sa vyznačujú jednoduchými a efektívnymi algoritmami. Avšak z hľadiska riešenia obcej úlohy, nájsť hranice, ktoré by oddeľovali určité triedy vo vstupnom priestore, sú obmedzené schopnosťou naučiť sa len lineárne oddeľovače (priamky, roviny).



Obr. 2.14: Oddelenie tried

Tieto metódy využívajú efektívne algoritmy pre nájdenie lineárnej hranice, ale zároveň sú schopné reprezentovať nelineárne funkcie. Základná myšlienka je prevod pôvodného vstupného priestoru do iného, viacrozmerného priestoru, kde sa triedy oddelia lineárne. Ako je možné vidieť na obr. 2.14a, oddeliť triedy lineárne (priamkou, rovinou) nie je možné, avšak podľa princípu pridáme 3. dimenziu a prvkom, ktoré sú uprostred kružnice, pridáme ďalšiu súradnicu a tak ich posunieme nahor. Triedy potom môžeme jednoducho oddeliť lineárne, a to rovinou rovnobežnou s rovinou danou osami x a y , nachádzajúcou sa medzi týmito dvoma triedami (obr. 2.11b). [7]

Pri využití tejto metódy je potrebná vysoká znalosť charakteru extrahovaných rysov. Pri riešení klasifikácii viac ako 3-rozmerného priestoru je ďalší priestor oddelený nadrovinami.

2.4 Zhodnotenie výpočtovej náročnosti uvedených metód

Náročnosť uvedených metód závisí na počte matematických operácií, ktoré je nutné s obrazom vykonať. Preto najväčšiu záťaž vykazujú metódy pri extrakcii rysov. Spôsoby klasifikácii vykazujú menšiu záťaž (s vektorom rysov už neprebiehajú zložité operácie). Z toho vyplýva, že nenáročné metódy extrakcii rysov pre OCR sú metódy: Rozdelenie do pásem, Metóda priesečníkov, Metóda invariantných momentov. Z klasifikácii: Metóda zhody. Náročné metódy extrakcii rysov: Štruktúrna analýza, Metóda 2D korelačnej funkcie; z klasifikácii: Neurónové siete, Adaboost, Metóda podporných vektorov.

3 VLASTNÝ NÁVRH OCR PRE „T-ZONES“

V tejto časti je popísaný konkrétny návrh rozpoznania číslíc z kontrolného obrázku používaného pri odosielaní SMS z internetových stránok. Analýzou dostatočného počtu obrázkov sa jednotlivé metódy upravujú pre čo najlepšie dosiahnuteľné výsledky.

3.1 Analýza obrazových dát

Príklady generovaných obrázkov sú zobrazené na obr. 3.1. Dôležité je zhrnúť ich podstatné vlastnosti a charakteristiky. Originál je uložený v obrazovom formáte *JPG*. Rozmer obrázka je 50x200 pixelov. Je možné si všimnúť na prvý pohľad 3 časti obrazu:

- » „purpurové“ číslice
- » „čierne“ krivky
- » „šedý“ prechod



Obr. 3.1: Originály generované na „t-zones“

- » jednotlivé číslice sú purpurovej farby, ktorá má vo farebnom priestore *RGB* a 8 bitovej hĺbke, súradnice približne

$$[R_p \ G_p \ B_p] = [216 \ 6 \ 113] \quad (3.1)$$

- » čierne krivky použité ako jedna z možností znehodnotenia obrazu strojovo rozpoznávať znaky (číslice) majú pri rovnakých podmienkach súradnice približne

$$[R_c \ G_c \ B_c] = [27 \ 26 \ 27] \quad (3.2)$$

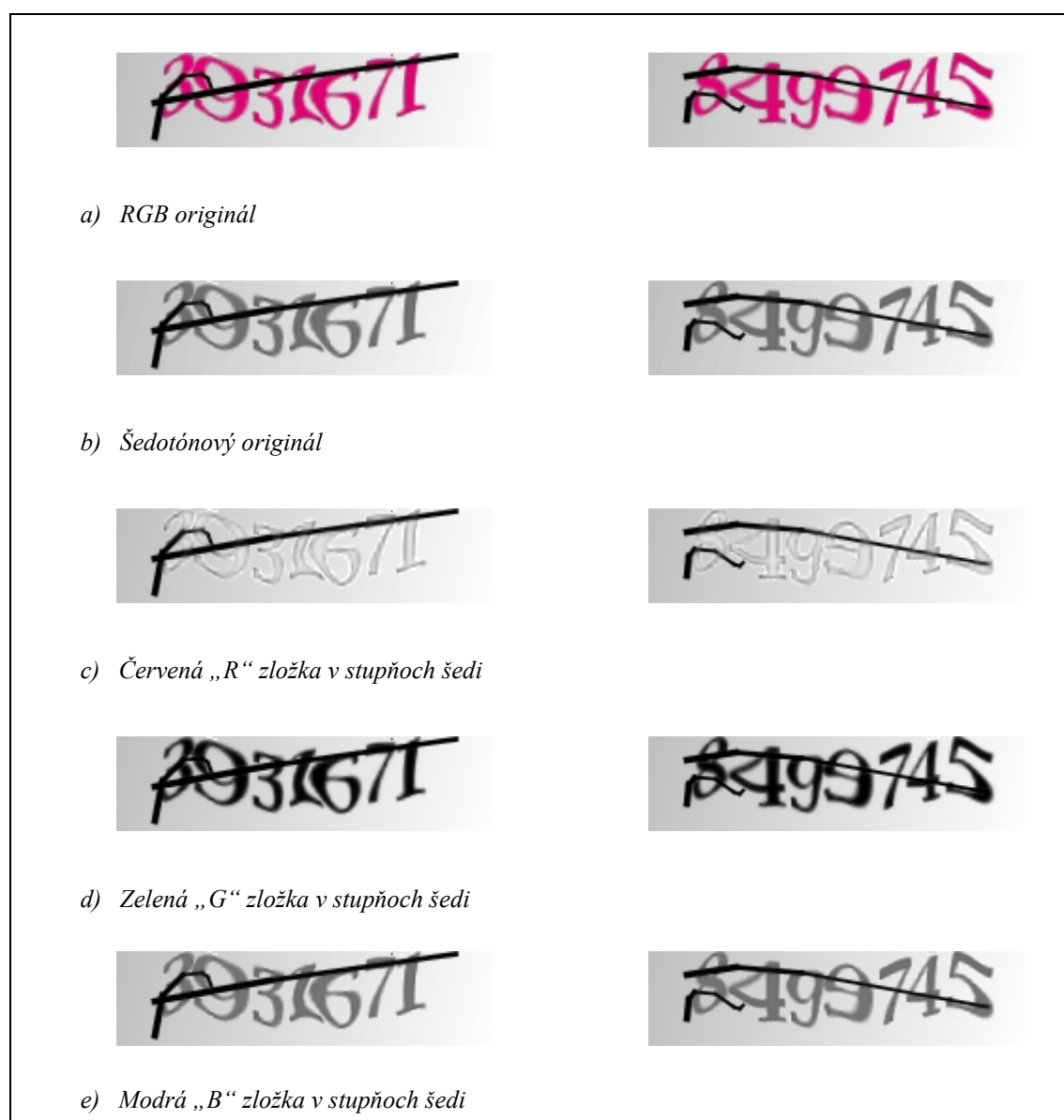
- » šedý prechod je približne lineárny (zľava doprava) v rozmedzí hodnôt

$$R_s = G_s = B_s = 193 \rightarrow 251 \quad (3.3)$$

Takto opísaný farebný priestor bezkompresného a bezstratového formátu odpovedá obrazovému formátu *BMP*. V priestore R^3 má obraz rozmer $Z = (50 \times 200 \times 3)$, to znamená $50 \times 200 = 10000$ obrazových bodov (pixelov), kde každý pixel reprezentujú 3 farebné súradnice farebného priestoru *RGB*. Z čoho vychádza farebná hĺbka $3 \times 8 \text{ bit} = 24 \text{ bit}$ na jeden pixel a celkovú farebnú škálu $2^{24 \text{ bit}} \approx 16,8 \text{ mil}$ farieb.

3.2 Predspracovanie

Po predchádzajúcej analýze je zrejмый ďalší zámer, a to odstrániť nadbytočné a skresľujúce informácie v obraze a ich možnú nápravu. To znamená hlavne odstránenie „čiernych“ kriviek, „šedého“ prechodu a maximalizovať kontrast medzi číslicou a pozadím.



Obr. 3.2: Jednotlivé zložky obrazov

Podľa jednotlivých kritérií a charakteristík nežiaducich informácií v obraze je potrebné vybrať alebo pripraviť obraz s čo najlepšimi vlastnosťami pre konkrétnu operáciu.

3.2.1 Prahovanie, odstránenie „šedého“ prechodu

Podľa obr. 3.2d je vidieť, že najlepší obraz pre prahovanie je ten s najvyšším kontrastom a teda obraz obsahujúci zelenú zložku. Šedý prechod má len veľmi malý kontrast, takže nebude nutné samostatne riešiť jeho odstránenie. Kontrast medzi ľavou a pravou stranou je

$$K_{\xi} = G_{\xi}(0, y) - G_{\xi}(200, y) = 251 - 193 = 58 \quad (3.4)$$

úrovni, ktoré však nezasahujú do kontrastu medzi číslicami a pozadím. To znamená že pri nastavení prahu na hodnotu $P < 193$ sa odstráni aj „šedý“ prechod. Kontrast mediánu zelenej zložky „purpurových“ číslic je

$$K_p = G_{\xi} - G_p = 193 - 6 = 187 \quad (3.5)$$

úrovni, čo je dostačujúce. Strata potrebnej informácie je minimálna. Pre ďalšie matematické operácie je vhodné nastaviť hodnotu informácie na 1 a pozadie na 0.



Obr. 3.3: Prahované obrazy

Proces prahovania je nasledujúci :

$$\begin{aligned} P_p &= 100 \\ G(x, y) < P_p &\rightarrow AA(x, y) = 1, \\ G(x, y) \geq P_p &\rightarrow AA(x, y) = 0 \end{aligned} \quad (3.6)$$

kde P_p je prah, $G(x, y)$ je vstupný obraz a $AA(x, y)$ je výstupný obraz. Výsledné obrazy po prahovaní sú na obr. 3.3.

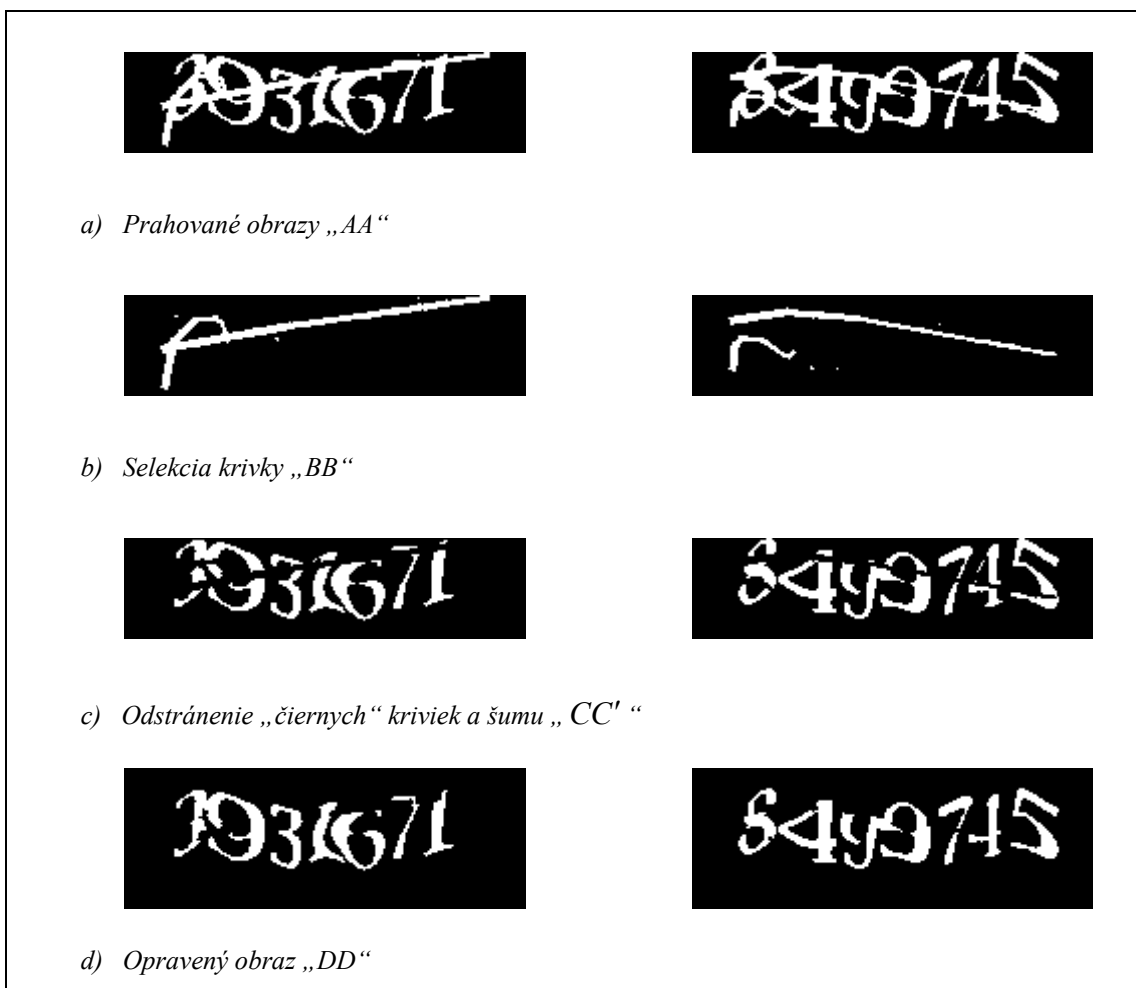
3.2.2 Odstránenie šumu, „čiernych“ kriviek, rekonštrukcia obrazu

Nežiaduce skreslenie prinášajú „čierny“ krivky. V kapitole 3.1 bolo zistené, že krivky majú všetky tri farebné súradnice rovnaké (blížiace sa k nule). Najväčší rozdiel je v obraze červenej zložky. Preto sa z neho vyselektujú a následne odčítajú od „prahovaného“ obrazu

$$\begin{aligned}
P_p &= 140 \\
R(x, y) < P_p &\rightarrow BB(x, y) = 1 \\
R(x, y) \geq P_p &\rightarrow BB(x, y) = 0 \\
CC &= AA - BB
\end{aligned}
\tag{3.7}$$

kde P_p je prah, $R(x, y)$ je vstupný obraz červenej zložky a $BB(x, y)$ je výstupný obraz. Po danej operácii sú výsledky na obr. 3.4b.

Následne je možné použiť operátor pre odfiltrovanie šumu, ktorý má veľkosť okna 3×3 pixely a nerobí nič iné ako sumu bielych pixelov v okne a keď je ich menej ako 3, nahradí hodnotu centrálného pixelu $CC(x, y)$ nulou. Filtrovanie prebieha nasledovne



Obr. 3.4: Kroky predspracovania

$$\begin{aligned}
count &= \sum_{a=1}^3 \sum_{b=1}^3 CC(x-2+a, y-2+b) \\
count < 3 &\rightarrow CC'(x, y) = 0 \\
count \geq 3 &\rightarrow CC'(x, y) = CC(x, y)
\end{aligned}
\tag{3.8}$$

kde $count$ je pomocná premenná, $CC(x, y)$ je obraz so šumom, $CC'(x, y)$ je výsledný obraz bez šumu a a, b sú súradnice v okne. Oprava obrazu znamená rekonštrukciu obrazu po odstránení „čiernych kriviek“ kde sa okrem samotnej krivky odstráni aj časť potrebnej obrazovej informácie. Táto časť je pomerne dôležitá a úspešnosť závisí od vlastností použitého algoritmu. V tejto časti práce je použitý operátor s oknom rovnakej veľkosti 3×3 pixely ako pre odfiltrovanie šumu a počíta sumu bielych pixelov v okne v obraze vyselektovanej „čiernej“ krivky (obr. 3.4b) a v obraze po jej odstránení (obr. 3.4c). V každom okne musí byť zároveň aspoň jeden biely pixel, a v obraze po odstránení musí byť 3 alebo 5 pixel v smere osi y biely. Z tohto popisu vyplýva že je možné opraviť miesta v ktorých je operátor kolmý na krivky, čiže hlavne časti kriviek v smere osi x . Výsledok je na obr. 3.4d. Oprava prebieha nasledovne

$$count(a, b) = \sum_{a=1}^3 \sum_{b=1}^3 CC'(x-2+a, y-2+b) \quad , \quad (3.9)$$

$$count(a) > 1 \wedge count(b) > 1 \wedge [G(x+3, y) \vee G(x+5, y)] \rightarrow DD(x, y) = 1$$

kde $count$ je pomocná premenná, $CC'(x, y)$ je vstupný obraz, $DD(x, y)$ je opravený obraz, a a, b sú súradnice v okne.

3.3 Segmentácia

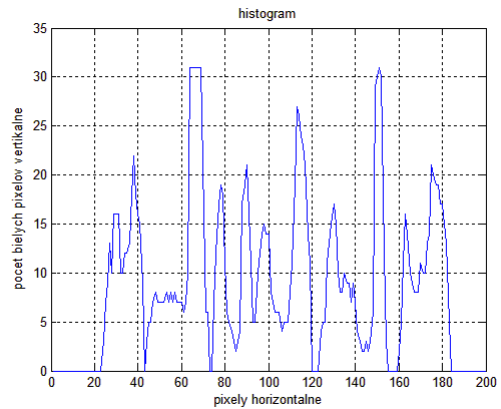
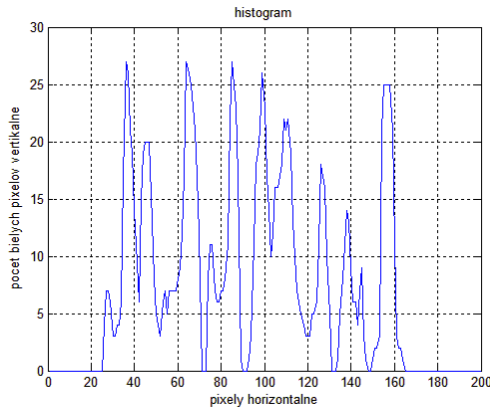
Pre segmentáciu jednotlivých číslic je obecné možné použiť histogram. Histogram je pri dvojúrovňovom obraze suma (počet) bielych pixelov v riadku (v smere osi x) alebo stĺpci (v smere osi y). V tomto prípade bolo potrebné nájsť hranicu rovnobežnú s osou y takže pre histogram predstavuje sumu v smere osi y . Je jasné, že oblasť kde je suma bielych pixelov rovná nule bude predstavovať hranicu. V tomto konkrétnom prípade nie je možné použiť túto techniku pretože jednotlivé čísla sú navzájom vo veľkom počte prípadov spojené. V histograme (obr. 3.5b) sú spojené znaky v prvom príklade „3“ a „9“, „1“ a „6“ nie sú spojené priamo, ale v histograme sa prekrývajú. V druhom príklade sú spojené znaky „9“ a „9“, znaky „7“ a „4“ do seba zasahujú.

Z tohto dôvodu bol použitý iný systém, ktorý je pôvodnému však podobný. Z dostatočného počtu obrazov boli zistené priemerné hodnoty hraníc pre každé z čísel. V ďalšom postupe sú použité histogramy (obr. 3.5c). Súčet červenej a modrej krivky (modrá je vynesená invertovane pre vizuálnu súvislosť s obrazom) určí hranice tak, že lokálne maximum v intervale zvolenom pre každú hranicu individuálne predstavuje novú hranicu. Program spätne zistí, o koľko pixelov je nová hranica odlišná od priemernej hodnoty hranice, a podľa toho upraví nasledujúce priemerné hranice. Dôvod je jednoduchý, ak bude nájdená číslica so šírkou menšou alebo väčšou ako je priemerná hodnota, aj nasledujúce číslice budú posunuté o daný počet pixelov.

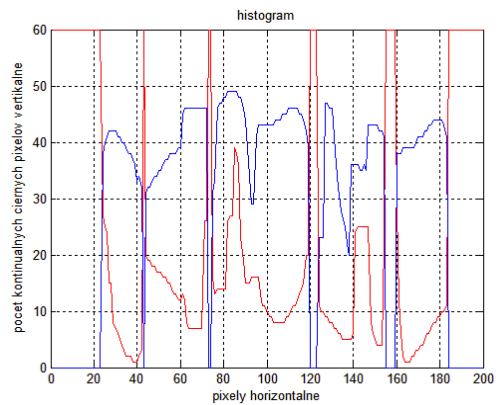
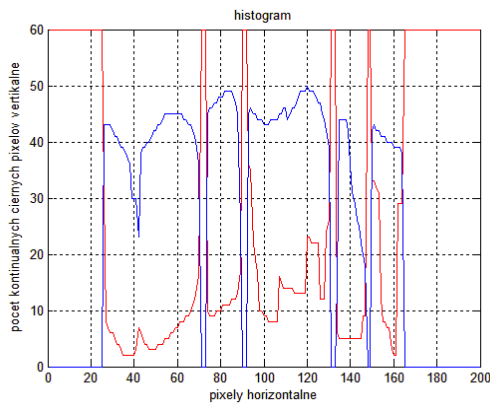
V horizontálnom smere sa číslica oreže podľa prvého bieleho pixelu vo vertikálnom smere. Konečná segmentácia oboch príkladov je na obr. 3.6.



a) „DD“



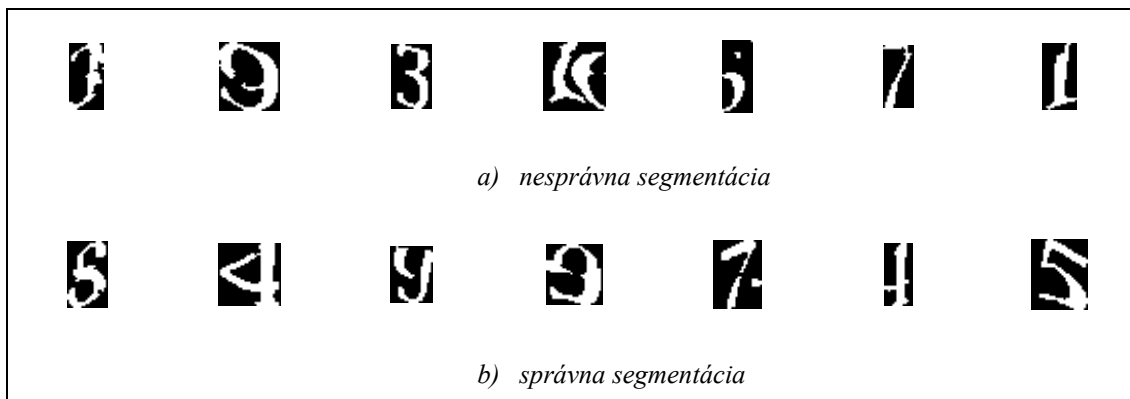
b) *Histogramy obrazov „DD“*



c) *Histogramy obrazov „DD“ pre súvislý počet čiernych pixelov „zhora“ (červená) a „zdola“ (modrá) invertovane*

Obr. 3.5: Histogramy

Prípady nesprávnej segmentácie môžu nastať obr. 3.6a. Pri vzorke 50 vstupných obrazov (slov) je úspešnosť segmentácie pre znaky je uvedená v tab. 3.1, pre slová v tab. 3.2. Keďže jediná nesprávna segmentácia v obraze spôsobí chybu v rozpoznávaní celého slova, úspešnosť bola rozdelená do dvoch tried. φ_z je znaková úspešnosť a počíta sa podľa vzťahu



Obr. 3.6: Príklady segmentácie

$$\varphi_z = \left(1 - \frac{a_z}{x_z}\right) \cdot 100\%, \quad (3.10)$$

kde a_z je počet nesprávne rozsegmentovaných znakov, x_z je počet všetkých znakov. η_o je obrazová úspešnosť a počíta sa podľa vzťahu

$$\varphi_o = \left(1 - \frac{a_o}{x_o}\right) \cdot 100\%, \quad (3.11)$$

kde a_o je počet nesprávne rozsegmentovaných obrazov, x_o je počet všetkých obrazov.

Tabuľka úspešnosti segmentácie pre znaky		
☺	☹	Σ
343	7	350
98,3%	1,7%	100,0%

Tab. 3.1 Úspešnosť segmentácie pre znaky

Tabuľka úspešnosti segmentácie pre slová		
☺	☹	Σ
47	3	50
94,0%	6,0%	100,0%

Tab. 3.2 Úspešnosť segmentácie pre slová

3.4 Momentová metóda

Názov momentová metóda značí využitie momentových invariantov popísaných v kap. 2.2.4. Využitých je 7Hu momentov a jeden ďalší. Momentové invarianty vytvorili špecifický vektor. Tento vektor by mal byť klasifikovaný ďalej natrénovanou neurónovou sieťou, pre ktorú existuje v programovom prostredí Matlab[®] toolbox „*Neural Network Pattern Recognition Tool*“. To vychádza na 8 momentov pre každý rozsegmentovaný znak, a 7 klasifikácií pre celé slovo. Celkovo 56 momentov a 7klasifikácií.

3.4.1 Extrakcia charakteristických rysov v momentovej metóde

Pre extrakciu rysov bola použitá metóda momentových invariantov. (kap. 2.2.4). Podľa (2.4) je možné napísať pre všetky nenulové diskkrétne hodnoty obrazu $P(x, y)$

$$m_{pq} = \sum_{i=1}^x \sum_{j=1}^y i^p j^q, \quad i, j \in P(x, y), \quad (3.10)$$

kde p a q určujú rád momentu, i a j sú súradnice nenulovej hodnoty obrazu $P(x, y)$, x a y udávajú rozmer obrazu $P(x, y)$. Pre centralizované momenty platí zmena (3.10) na

$$M_{pq} = \sum_{i=1}^x \sum_{j=1}^y (i-a)^p (j-b)^q, \quad i, j \in P(x, y), \quad (3.11)$$

kde a a b sú súradnice stredu (ťažiska) obrazu $P(x, y)$. Pre normalizované centralizované momenty platí zmena (3.11)

$$\tilde{M}_{pq} = \frac{M_{pq}}{(m_{pq})^w}, \quad w = \frac{p+q}{2} + 1. \quad (3.12)$$

Pre I_1 až I_8 vo vzťahoch (2.5) až (2.12) platí záměna

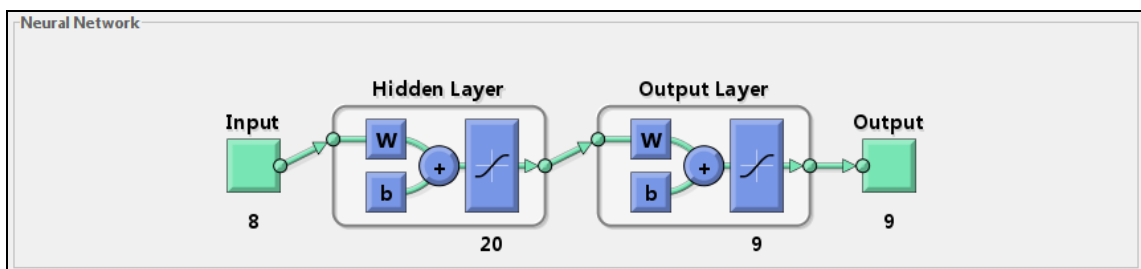
$$\eta_{pq} = \tilde{M}_{pq}, \quad I_{(1,8)} \Rightarrow \tilde{I}_{(1,8)}. \quad (3.13)$$

Tým je dosiahnuté, že výsledné \tilde{I} momenty sú nezávislé na posunutí, veľkosti a rotácii.

3.4.2 Konečné rozpoznávanie v momentovej metóde

Pre výslednú klasifikáciu, bola použitá neurónová sieť (NN) typu feed-forward (sieť s dopredným šírením), to znamená že výstup neurónu jednej hladiny je vstupom neurónu len nasledujúcej hladiny, informácia sa nešíri rekurentne. NN má dve hladiny (obr. 3.7), skrytú hladinu (Hidden layer) a výstupnú hladinu (Output layer). Hladiny majú prenosovú funkciu typu sigmoida. Počet neurónov v skrytej hladine je variabilný. Výstup (Output) z výstupnej hladiny predstavuje výsledok (vektor), počet neurónov je daný podľa počtu tried.

Keďže je potrebná klasifikácia do deviatich tried, výstupný vektor má veľkosť deväť, s jednou jednotkou, označujúcou danú triedu a nulami na zvyšných ôsmich miestach. Príklad klasifikácie je na obr. 3.9.

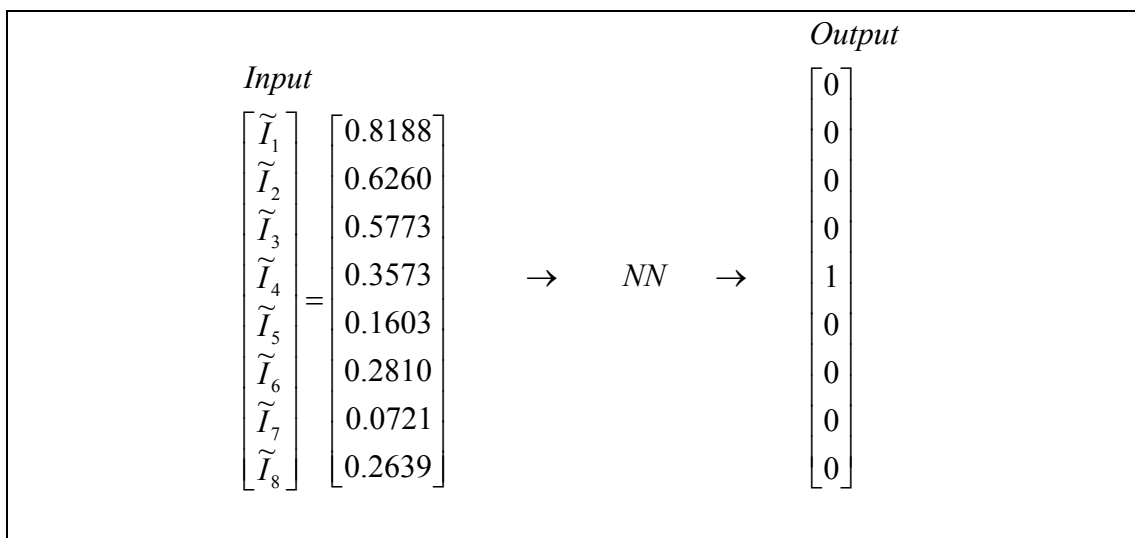


Obr. 3.7: Štruktúra neurónovej siete

K dispozícii bolo 280 vstupných vektorov a rovnaký počet výstupných vektorov rozdelených do troch skupín. Trénovacia skupina $70\% \cdot 280 = 196$ vektorov bola

Output Class	1	2	3	4	5	6	7	8	9	
1	0	0	0	0	0	0	0	0	0	NaN%
2	0	6	3	0	2	0	1	0	2	42.9%
3	0	0	0	0	0	0	0	0	0	NaN%
4	1	0	0	1	0	0	0	0	0	50.0%
5	0	0	0	0	0	0	0	0	0	NaN%
6	0	0	0	0	0	0	0	0	0	NaN%
7	0	1	0	0	0	0	5	0	0	33.3%
8	1	0	0	4	2	0	1	4	0	33.3%
9	0	0	0	0	3	3	0	0	2	25.0%
	0.0%	85.7%	0.0%	20.0%	0.0%	0.0%	71.4%	100%	50.0%	42.9%
	100%	14.3%	100%	80.0%	100%	100%	28.6%	0.0%	50.0%	57.1%
	1	2	3	4	5	6	7	8	9	
	Target Class									

Obr. 3.8: Dosaiahnuté testovacie výsledky pri trénovaní siete



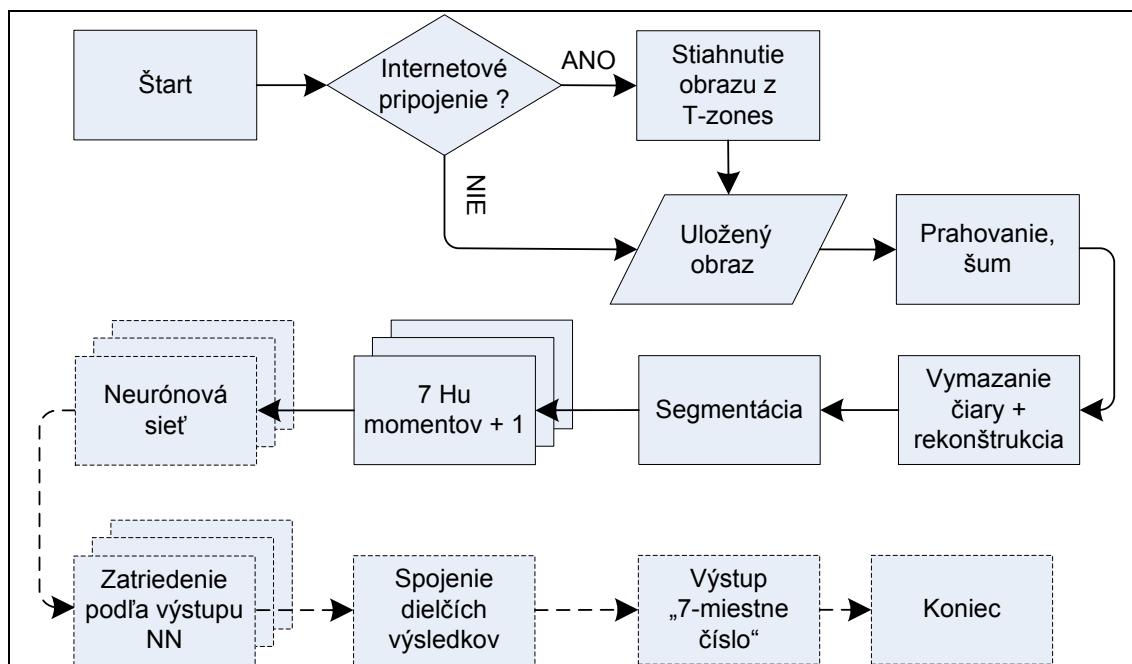
Obr. 3.9: Príklad klasifikácie

použitá na nastavenie jednotlivých váh perceptronov. Validačná skupina $15\% \cdot 280 = 42$ bola použitá pri tréovaní a to tak, že tréovanie sa zastaví, keď MSE (mean squared error) dosiahne minimum. MSE je stredná kvadratická odchýlka medzi výstupným vektorom z NN a cieľovým vektorom (vektor definovaný programátorom ako pravdivý výstup). Tu je ale obecné možné, že minimum je lokálne. Testovacia skupina $15\% \cdot 280 = 42$ má za úlohu otestovať už konečnú verziu NN. Zhodnotenie výsledku testovacej množiny je na obr. 3.8. Konečná úspešnosť klasifikácie a MSE sú dva rozdielne parametre a nesúvisia so sebou priamo.

3.4.3 Popis blokov programu momentovej metódy, ovládanie

Program pre rozpoznávanie pomocou momentovej metódy nie je úplne ukončený, a teda nepoužiteľný užívateľom. Problém spočíval vo výsledkoch, kde pri ideálnom prípade by pravdepodobne bola vhodná kombinácia s inou metódou pre extrahovanie rysov. Z výslednej klasifikácie nie je možné obviňovať neurónovú sieť generovanú v toolboxe „*Neural Network Pattern Recognition Tool*“. Výsledky momentov v 8-dimenzionálnom priestore vykazovali vysoký rozptyl. Samotná metóda bola mnohými vyskúšaná, takže problém v metóde len ťažko hľadať. Úspešnosť klasifikácie nie je zanedbateľná, no v tomto prípade nepoužiteľná.

Štruktúra programu je uvedená na obr. 3.10. Prvá časť, po segmentáciu, je rovnaká pre obidve metódy popísané v tejto práci. Bloky vyznačené prerušovanou čiarou znamenajú, že nie sú fyzicky implementované do konečného programu z dôvodu malej, alebo žiadnej využiteľnosti daného programu, otestovaný však bol. V prílohe B v zložke [TMOB1] je dátový súbor [data.mat], kde v programovom prostredí Matlab[®] stačí načítať vektory [in] (značí 8 momentov) a [out] (vektor veľkosti 9 s hodnotou 1 pri klasifikovanej triede a 0 inde) do „Workspace“ pomocou príkazu [load_data.mat]. Následne spustiť toolbox „*Neural Network Pattern Recognition Tool*“ príkazom [nprtool] a postupovať podľa návodu. Vzorky sú orientované v stĺpcoch. V skrytej vrstve bolo použitých 20 neurónov. Množina vstupných a výstupných vektorov bola rozdelená v pomere podľa kap.3.4.2.



Obr. 3.10: Blokový diagram programu momentovej metódy

Kompletné popísanie spoločných blokov v diagrame programu momentovej metódy je v kap.3.5.3. Zdrojový kód je k dispozícii v prílohe. Pre Stiahnutie a uloženie obrázku z T-zones príloha A.1, Prahovanie p. A.2, Odstránenie šumu p. A.3, Vyhladenie čiary + rekonštrukcia p. A.4, Segmentácia p. A.5, 7 Hu momentov + 1 p. A.7,9,10,11.

3.4.4 Výsledky a zhodnotenie momentovej metódy

Výsledky sú spracované obvyklým spôsobom. V tabuľkách sú zhodnotené jednotlivé javy vyskytujúce sa pri rozpoznávaní. Polia s výskytom úspešných javov sú v tabuľkách vyplnené zelenou farbou a symbolom „☺“, naopak polia s výskytom neúspešných javov sú vyplnené oranžovou farbou a symbolom „☹“, symbol „Σ“ značí súčet úspešných a neúspešných javov v riadku alebo stĺpci, a je vyplnený modrou farbou. Hodnota „NaN“ značí špeciálny prípad, keď v danom stĺpci nebol výskyt úspešnej, ani neúspešnej klasifikácie (tab. 3.7), alebo irelevantnosť výsledku (tab. 3.3).

V tab. 3.3. je zobrazený výskyt jednotlivých javov pri testovaní momentovej metódy vzorkou 40 slov a teda 280 znakov. Rozdiel v sumách stĺpcov a riadkov je v odlišnom pohľade na klasifikáciu. Suma v stĺpci znamená, akým podielom boli zastúpené zobrazené znaky pri výsledku klasifikácie danej číslice v stĺpci. Naopak suma v riadku je významnejšia, pretože znamená podiel zastúpenia klasifikácie číslice pri zobrazení znaku v danom riadku. Samozrejme celková suma úspešných a neúspešných klasifikácií, pri oboch pohľadoch, je rovnaká.

V tab. 3.5, a príslušnom grafe na obr. 3.11, je zobrazený percentuálny výskyt jednotlivých javov pri rozpoznávaní. Symbol „Σ“ v tabuľke určuje úspešnosť rozpoznania znakov, pri momentovej metóde $\varphi_{ZMM} = 37,5\%$.

Tabuľka počtu výskytov jednotlivých javov pri rozpoznávaní													
		Počet rozpoznání pre jednotlivé číslice [-]								Celkom [-]			
		1	2	3	4	5	6	7	8	9	☺	☹	Σ
Zobrazený znak	"1"	0	1	0	14	0	0	0	10	1	0	26	26
	"2"	0	29	0	0	0	0	1	0	2	29	3	32
	"3"	0	7	0	0	0	0	0	16	6	0	29	29
	"4"	0	1	0	3	0	0	0	24	4	3	29	32
	"5"	0	15	0	0	0	0	2	10	12	0	39	39
	"6"	0	4	0	1	0	0	0	8	12	0	25	25
	"7"	0	3	0	0	0	0	25	1	3	25	7	32
	"8"	0	0	0	0	0	0	1	34	1	34	2	36
	"9"	0	6	0	0	0	0	0	9	14	14	15	29
Celkom [-]	☺	0	29	0	3	0	0	25	34	14	105	NaN	NaN
	☹	0	37	0	15	0	0	4	78	41	NaN	175	NaN
	Σ	0	66	0	18	0	0	29	112	55	NaN	NaN	280

Tab. 3.3 Počet výskytu javov v momentovej metóde

Úspešnosť klasifikácie jednotlivých zobrazených znakov je v tab. 3.6. Aj keď momentová metóda v tomto prípade nepodala kvalitné výsledky, výrazne úspešne sú klasifikované znaky „2“, „7“, „8“. Výsledky zvyšných znakov sú blízke „tipovaniu“.

Tab. 3.7 a obr. 3.13 zobrazuje percentuálnu pravdepodobnosť zobrazenia znakov pri výsledku klasifikácie danej číslice v stĺpci. Najlepší výsledok má číslica 7.

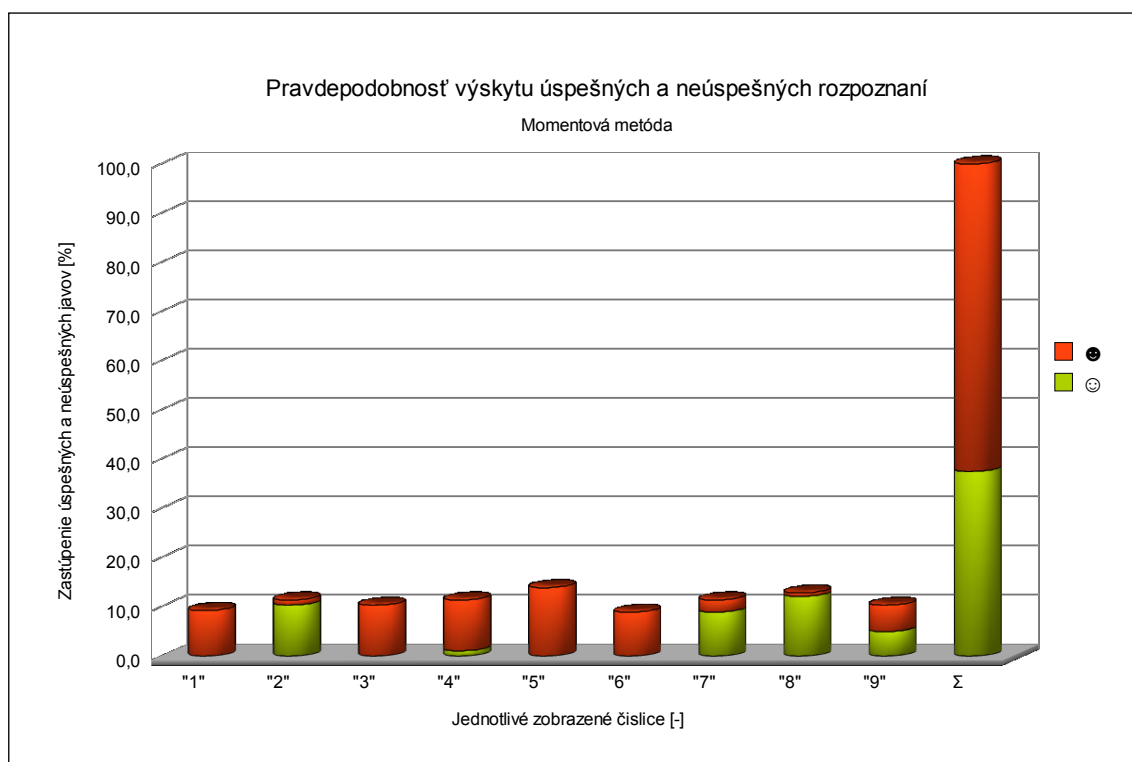
Rozpoznanie celých slov je pre úplnosť v tab. 3.4, aj keď je jasné, že pri normálnom rozložení pravdepodobnosti $\varphi_{Z_{MM}}$ v slovách bude pravdepodobnosť rozpoznania slov $\varphi_{S_{MM}} \cong 0,0\%$. Teoretickú hranicu preklenutia 0,0% je možné určiť tak, že v každom slove môže byť maximálne jedna chyba a teda $\varphi_{Z_{MM}} > 85,71\%$.

Tabuľka počtu celých rozpoznávaných slov		
☺	☹	Σ
0	40	40
0,0%	100,0%	100,0%

Tab. 3.4 Úspešnosť klasifikácie slov v momentovej metóde

Tabuľka pravdepodobností výskytu je dnotlivých javov pri rozpoznávaní											
Zobrazený znak	Počet rozpoznaní pre jednotlivé číslice [%]									Celkom [%]	
	1	2	3	4	5	6	7	8	9	☺	☹
"1"	0,0	0,4	0,0	5,0	0,0	0,0	0,0	3,6	0,4	0,0	9,3
"2"	0,0	10,4	0,0	0,0	0,0	0,0	0,4	0,0	0,7	10,4	1,1
"3"	0,0	2,5	0,0	0,0	0,0	0,0	0,0	5,7	2,1	0,0	10,4
"4"	0,0	0,4	0,0	1,1	0,0	0,0	0,0	8,6	1,4	1,1	10,4
"5"	0,0	5,4	0,0	0,0	0,0	0,0	0,7	3,6	4,3	0,0	13,9
"6"	0,0	1,4	0,0	0,4	0,0	0,0	0,0	2,9	4,3	0,0	8,9
"7"	0,0	1,1	0,0	0,0	0,0	0,0	8,9	0,4	1,1	8,9	2,5
"8"	0,0	0,0	0,0	0,0	0,0	0,0	0,4	12,1	0,4	12,1	0,7
"9"	0,0	2,1	0,0	0,0	0,0	0,0	0,0	3,2	5,0	5,0	5,4
Σ [%]										37,5	62,5

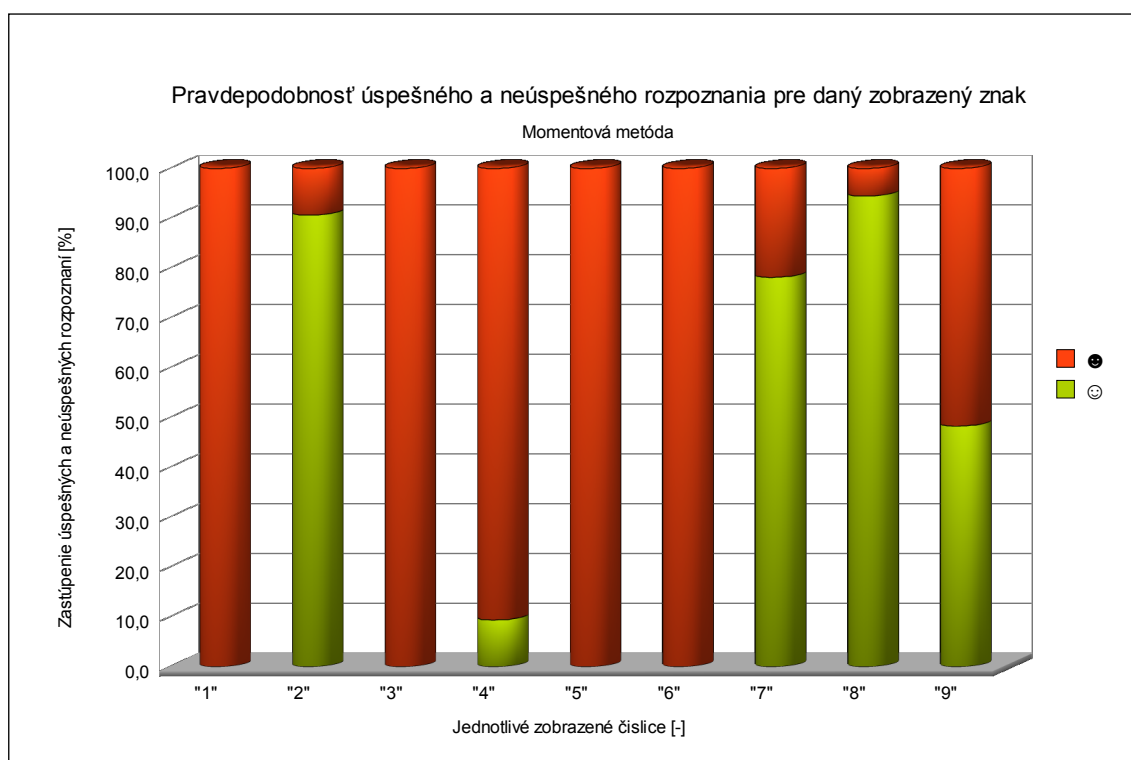
Tab. 3.5 Pravdepodobnosť výskytu javov v momentovej metóde



Obr. 3.11: Graf výskytu rozpoznaní v momentovej metóde

Tabuľka pravdepodobností počtu rozpoznania pre daný zobrazený znak											
Zobrazený znak	Počet rozpoznaní pre jednotlivé číslice [%]									Celkom [%]	
	1	2	3	4	5	6	7	8	9	☺	☹
"1"	0,0	3,8	0,0	53,8	0,0	0,0	0,0	38,5	3,8	0,0	100,0
"2"	0,0	90,6	0,0	0,0	0,0	0,0	3,1	0,0	6,3	90,6	9,4
"3"	0,0	24,1	0,0	0,0	0,0	0,0	0,0	55,2	20,7	0,0	100,0
"4"	0,0	3,1	0,0	9,4	0,0	0,0	0,0	75,0	12,5	9,4	90,6
"5"	0,0	38,5	0,0	0,0	0,0	0,0	5,1	25,6	30,8	0,0	100,0
"6"	0,0	16,0	0,0	4,0	0,0	0,0	0,0	32,0	48,0	0,0	100,0
"7"	0,0	9,4	0,0	0,0	0,0	0,0	78,1	3,1	9,4	78,1	21,9
"8"	0,0	0,0	0,0	0,0	0,0	2,8	2,8	94,4	2,8	94,4	5,6
"9"	0,0	20,7	0,0	0,0	0,0	0,0	0,0	31,0	48,3	48,3	51,7

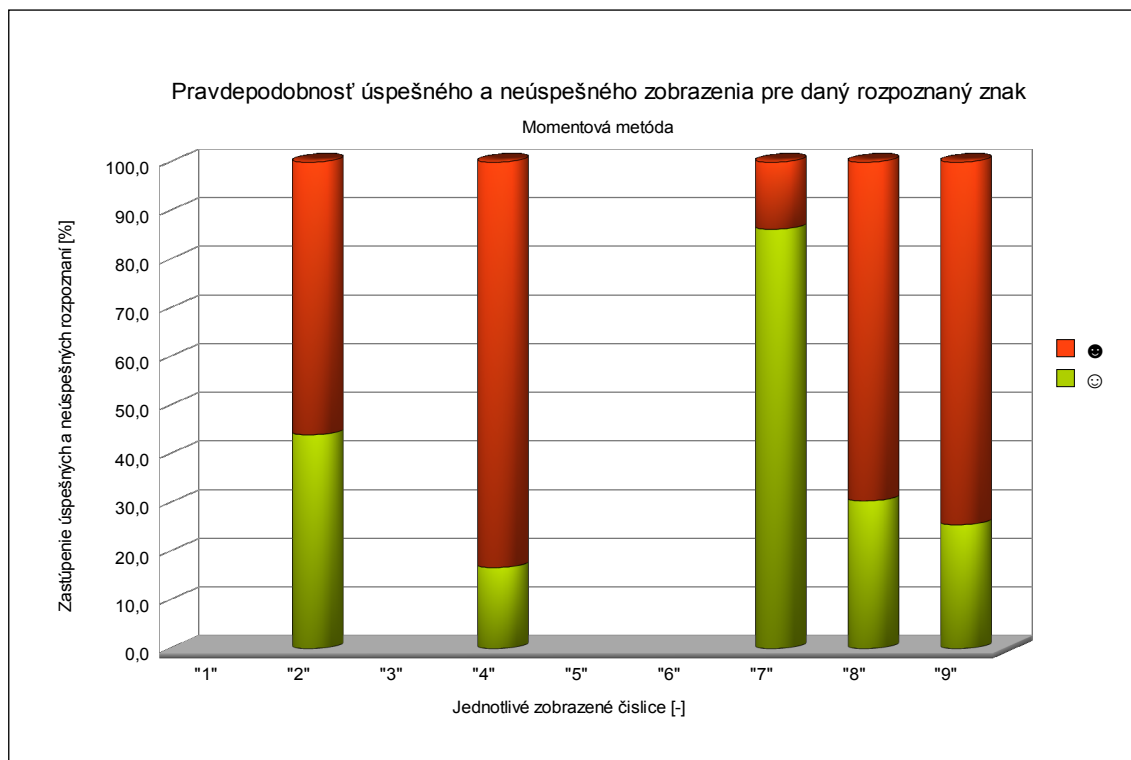
Tab. 3.6 Pravdepodobnosť rozpoznania v momentovej metóde



Obr. 3.12: Graf pravdepodobností rozpoznania v momentovej metóde

Tabuľka pravdepodobností počtu zobrazenia pre daný rozpoznávaný znak										
Počet rozpoznání pre jednotlivé číslice [%]										
Zobrazený znak		1	2	3	4	5	6	7	8	9
	"1"	NaN	1,5	NaN	77,8	NaN	NaN	0,0	8,9	1,8
	"2"	NaN	43,9	NaN	0,0	NaN	NaN	3,4	0,0	3,6
	"3"	NaN	10,6	NaN	0,0	NaN	NaN	0,0	14,3	10,9
	"4"	NaN	1,5	NaN	16,7	NaN	NaN	0,0	21,4	7,3
	"5"	NaN	22,7	NaN	0,0	NaN	NaN	6,9	8,9	21,8
	"6"	NaN	6,1	NaN	5,6	NaN	NaN	0,0	7,1	21,8
	"7"	NaN	4,5	NaN	0,0	NaN	NaN	86,2	0,9	5,5
	"8"	NaN	0,0	NaN	0,0	NaN	NaN	3,4	30,4	1,8
	"9"	NaN	9,1	NaN	0,0	NaN	NaN	0,0	8,0	25,5
Celkom [%]	☺	NaN	43,9	NaN	16,7	NaN	NaN	86,2	30,4	25,5
	☹	NaN	56,1	NaN	83,3	NaN	NaN	13,8	69,6	74,5

Tab. 3.7 Pravdepodobnosť zobrazenia v momentovej metóde



Obr. 3.13: Graf pravdepodobností zobrazenia v momentovej metóde

3.5 Korelačná metóda

Korelačná metóda je metóda využívajúca korelačné vlastnosti zobrazených znakov. Podrobne je popísaná v kap. 2.2.5.

3.5.1 Extrakcia charakteristických rysov v korelačnej metóde

Extrakcia rysov v korelačnej metóde je založená na výpočte 2D korelácie nenormalizovanej funkcie `[xcorr2]`.

Pre každú triedu znakov, kde $T_{i,j}$ je normálový uložený znak a $i = \{1,2,\dots,9\}$, je zvolený istý počet správne rozsegmentovaných znakov s ktorými prebehne 2D korelácia s každým neznámym rozsegmentovaným znakom. Pre každú triedu bol zvolený počet normálových znakov 25 a teda $j = \{1,2,\dots,25\}$. Normálové znaky by mohli byť od seba čo najodlišnejšie, pretože konečné rozpoznávanie prebieha v hľadaní maxima korelačnej funkcie. To znamená, že je vhodné aby trieda obsahovala čo najväčší počet znakov, a prípadne aj ich chýb, ktoré sa môžu vyskytnúť (napríklad chyby spôsobené nedokonalou opravou čiernej čiary).

Z predchádzajúceho popisu vyplýva, že pre každý neznámy rozsegmentovaný znak U prebehne $25 \cdot 9 = 225$ 2D korelácií, z každej je vytiahnuté maximum.

$$H_{i,j} = \max[\text{xcorr2}(U, T_{i,j})] \quad (3.14)$$

Pre každý znak vznikne matica H veľkosti $(i \times j)$. Príklady výsledkov 2D korelácií je možné vidieť na obr. 2.11. Maximá autokorelačných funkcií sú výrazné. Tento proces je výpočtovo pomerne náročný, keď pre extrakciu rysov je potrebné spočítať $25 \cdot 9 \cdot 7 = 1575$ korelácií.

3.5.2 Konečné rozpoznávanie v korelačnej metóde

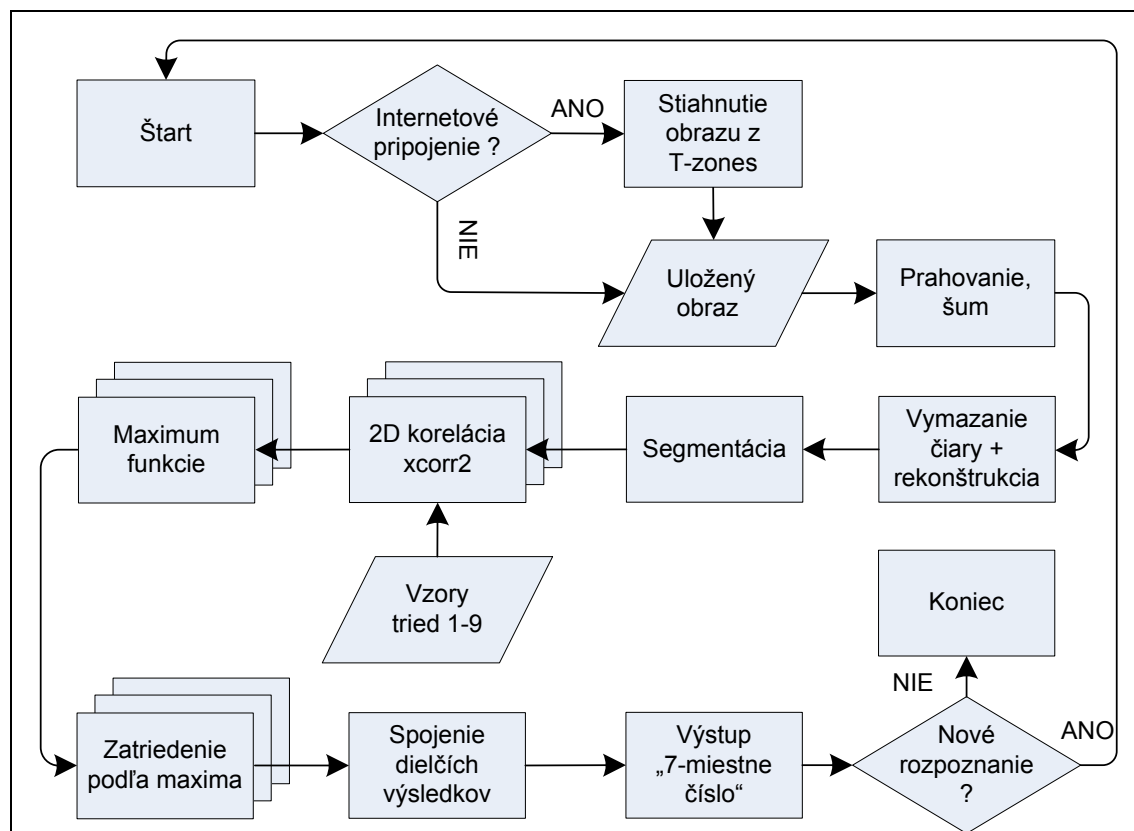
So sadou 225 hodnôt je možné ďalšie spracovanie, ako výpočet priemeru pre každú triedu, výpočet mediánu, a iné. V tejto práci je použité jednoduché nájdenie súradnice i pre maximum hodnoty prvku v matici H

$$\text{find}(i) = \max(H_{i,j}) \quad (3.15)$$

i teda predstavuje stĺpec matice H a zároveň danú triedu znakov. Sú možné rôzne kombinácie pre opravu a testovanie výsledku. Výsledky rozdielnosti maxim 2D korelačnej funkcie je vidieť na obr. 2.11.

3.5.3 Popis blokov programu korelačnej metódy, ovládanie

Program korelačnej metódy je so svojimi popísanými vlastnosťami funkčný a prístupný v zdrojovom tvare v spustiteľnej aplikácii. Obidve verzie sa nachádzajú v prílohe B. Pre zdrojový tvar (prípona „.m“) je nutné spustenie pod programovým prostredím Matlab®. Spustiteľný aplikačný súbor (prípona „.exe“) je možné spustiť podľa pokynov uvedených na konci kapitoly. Vlastnosti programu obidvoch tvarov sú identické. Metóda je založená na extrakcii rysov pomocou 2D korelačnej funkcie a klasifikácia následným nájdením maxima.

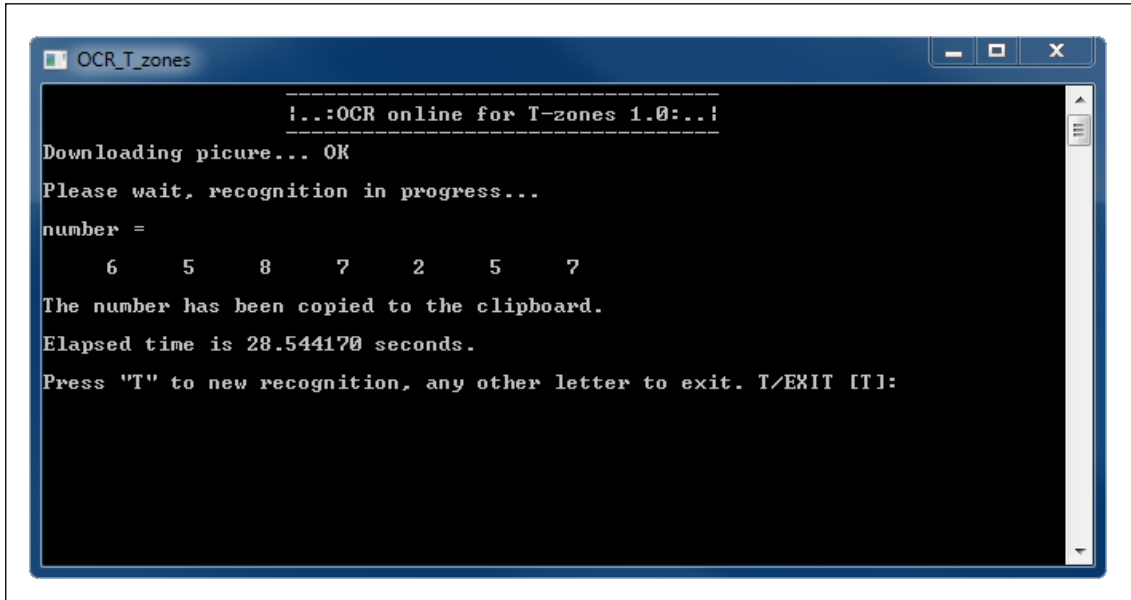


Obr. 3.14: Blokový diagram programu korelačnej metódy

Štruktúru programu je možné opísať pomocou obr. 3.14. Jednotlivé časti sú spúšťané sekvenčne, presne tak ako je to vyznačené. Ako prvé sa program snaží stiahnuť nový obraz z <http://sms.t-zones.cz/open/captcha.jpg> a následne ho uložiť do pracovnej zložky. V prípade neúspechu je využitý posledný uložený obraz. Náhľad na zdrojový kód tohto procesu je možný v príloha A.1 Ďalej nasleduje spracovanie obrazu spôsobom popísaným v kapitole 3.2. Zdrojové kódy prahovania, odstránenia šumu, vymazania čiernej krivky, rekonštrukcie obrazu sú prístupné v prílohe A.2, A.3, A.4. Nasleduje segmentácia podľa kapitoly 3.3, kód v prílohe A.5.

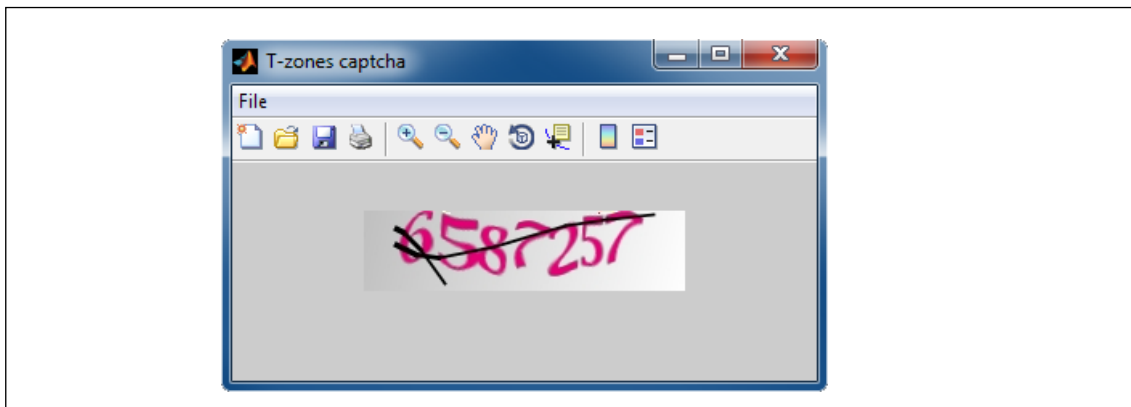
Hlavným „mozgom“ celého programu je 2D korelácia, ktorá daný neznámy rozsegmentovaný znak skoreluje s vzormi všetkých tried, a výsledky uloží do matice. Takto prebehne za sebou korelácia všetkých 7 neznámych znakov obsiahnutých v jednom obraze. Kód v prílohe A.6, A.7. V prílohe A.7 je nájdenie ťažiska znaku ,

potom nasleduje centrovanie znaku. Tento proces nie je nevyhnutne potrebný, no pri testovaní kombinácií s inými metódami klasifikácie bol potrebný. Výsledná klasifikácia prebieha pomocou nájdenia maxima, kód je v prílohe A.8. Je tam taktiež pokusný výpočet priemeru korelácií v triedach. Po skončení klasifikácie je zobrazený výsledok, a v prípade záujmu je možné nové rozpoznávanie.



Obr. 3.15: Hlavné okno spustiteľného programu „OCR_T_zones“

K dispozícii sú dve verzie s názvom hlavnej zložky [TMOB2_ALL] a [TMOB2]. Rozdiel je v tom, že v pracovnom adresári verzie [TMOB2_ALL] je možné prezrieť čiastkové výsledky spracovania obrazu uložené vo formáte „.bmp“. Spustenie programu pod programovým prostredím Matlab® prebieha zadaním názvu hlavného súboru príkazu [mainf] do príkazového okna. Samozrejme musí byť nastavená pracovná zložka s názvom [MATLAB]. Musí byť sprístupnený zápis na disk a to pre prípad stiahnutia rozpoznávaného obrazu priamo do pracovnej zložky (je vhodné skopírovať celú zložku). Výsledok sa zobrazí v príkazovom okne.



Obr. 3.16: Okno programu zobrazujúce rozpoznávané slovo

Spustiteľnú aplikáciu tzv. „Stand Alone Application“ (SAP) je možné spustiť odkazom [OCR_t_zones.lnk] v zložke [TMOB2_EXE] alebo priamo [mainf.exe] v zložke

[TMOB2_EXE\MATLAB]. Nevýhoda spustenia SAP spočíva v tom, že skompilované aplikačné súbory musia mať prístup do všetkých knižníc (nie je možné, alebo len veľmi obtiažne, selektovať len potrebné knižnice). Pre spustenie aplikácie je nutné mať nainštalovanú rovnakú verziu programového prostredia Matlab[®], v tomto prípade „R2009b“, alebo nainštalovať súbor knižníc „*Matlab Component Runtime*“ (MCR) verzie v ktorej bol skompilovaný program ako spustiteľná aplikácia. MCR je prístupný v prílohe B v zložke [MCR] a inštaluje sa spustením súboru [MCRInstaller.exe]*. Prípadnú kompiláciu v iných verziách je možné previesť príkazom [mcc_-m_-T_link:exe_mainf] napísaním do príkazového okna. Hlavné okno po spustení SAP je na obr. 3.15 a okno zobrazujúce rozpoznávací obraz je na obr. 3.16 .

V prípade vlastníctva kompletnej verzie Matlab[®] je výhodné spúšťať program pod týmto prostredím, kvôli jednoduchosti. Samotný čas výpočtu uvádzaný v práci je vzťahovaný k spúšťaniu programu pod prostredím Matlab[®] alebo spustením SAP pod kompletnou verziou Matlab[®]. Spustenie SAP s MCR je náročnejšie na výkon PC. Čas potrebný pre výpočet sa v niektorých prípadoch zdvojnásobil. Spustenie pod MCR nie je najvhodnejším riešením, no umožňuje chod aplikácie bez kompletného prostredia Matlab[®].

* pri kompilácii vo verzii „R2008b“ a nainštalovaným MCR aplikácia nefunguje, je nutné v operačnom systéme pridať cestu nainštalovaného MCR [;...~\bin\win32] do „path“ v [System Properties>Advanced>Environment Variables>User Variables].

3.5.4 Výsledky a zhodnotenie korelačnej metódy

Tento odsek je vhodné preskočiť v prípade prečítania kap.3.4.4 a pokračovať v nasledujúcom odseku. Výsledky sú spracované ako v predchádzajúcej metóde. V tabuľkách sú zhodnotené jednotlivé javy vyskytujúce sa pri rozpoznávaní. Polia s výskytom úspešných javov sú v tabuľkách vyplnené zelenou farbou a symbolom „☺“, naopak polia s výskytom neúspešných javov sú vyplnené oranžovou farbou a symbolom „☹“, symbol „Σ“ značí súčet úspešných a neúspešných javov v riadku alebo stĺpci a je vyplnený modrou farbou. Hodnota „NaN“ značí irelevantnosť výsledku tab. 3.9.

V tab. 3.9. je zobrazený výskyt jednotlivých javov pri testovaní korelačnej metódy vzorkou 50 slov a teda 350 znakov. Rozdiel, v sumách stĺpcov a riadkov, je v odlišnom pohľade na klasifikáciu. Suma v stĺpci znamená, akým podielom boli zastúpené zobrazené znaky pri výsledku klasifikácie danej číslice v stĺpci. Naopak suma v riadku je významnejšia, pretože znamená podiel zastúpenia klasifikácie danej číslice pri zobrazení znaku v danom riadku. Samozrejme celková suma úspešných a neúspešných klasifikácií pri oboch pohľadoch je rovnaká. Významný je recipročný jav pri číslici, alebo znaku 1 a 4, kde je výrazná korelačná vlastnosť v časti ich obrazu. Najviac absolútnych klasifikačných chýb dosiahla číslica 1.

Tabuľka počtu výskytov jednotlivých javov pri rozpoznávaní													
		Počet rozpoznání pre jednotlivé číslice [-]								Celkom [-]			
		1	2	3	4	5	6	7	8	9	☺	☹	Σ
Zobrazený znak	"1"	27	0	0	3	0	0	0	0	0	27	3	30
	"2"	0	40	0	0	0	0	0	0	0	40	0	40
	"3"	4	0	44	0	1	0	0	1	0	44	6	50
	"4"	2	0	0	35	0	0	0	1	0	35	3	38
	"5"	0	0	0	0	39	0	0	1	0	39	1	40
	"6"	0	0	2	0	3	35	0	0	0	35	5	40
	"7"	0	0	0	0	0	0	33	0	0	33	0	33
	"8"	0	0	0	0	0	0	0	36	0	36	0	36
	"9"	1	0	0	0	0	0	0	1	41	41	2	43
Celkom [-]	☺	27	40	44	35	39	35	33	36	41	330	NaN	NaN
	☹	7	0	2	3	4	0	0	4	0	NaN	20	NaN
	Σ	34	40	46	38	43	35	33	40	41	NaN	NaN	350

Tab. 3.8 Počet výskytu javov v korelačnej metóde

Celkovo vzniklo ,pri vzorke 350 znakov, 20 neúspešných klasifikácií. Vysoký podiel pre neúspešnú klasifikáciu má neúspešná segmentácia.

V tab. 3.10 a príslušnom grafe na obr. 3.18 je zobrazený percentuálny výskyt jednotlivých javov pri rozpoznávaní . Σ v tabuľke určuje úspešnosť rozpoznania znakov pri korelačnej metóde $\varphi_{z_{KM}} = 94,3\%$.

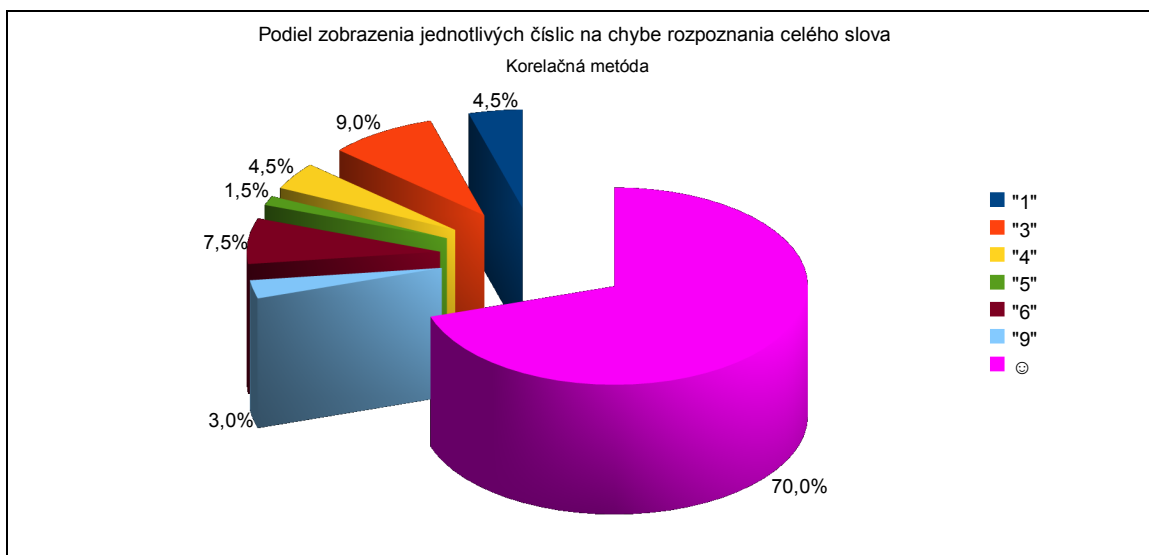
Tab. 3.11 a graf na obr. 3.19 zobrazuje percentuálnu pravdepodobnosť rozpoznania jednotlivých znakov pri zobrazení danej číslice v stĺpci. Najlepší výsledok 100% mali znaky „2“, „7“, „8“. Najhorší relatívny výsledok dosiahol znak „2“ s pravdepodobnosťou rozpoznania 87,5%.

Tab. 3.12 a graf na obr. 3.20 zobrazuje percentuálnu pravdepodobnosť zobrazenia jednotlivých znakov pri výsledku klasifikácie danej číslice v stĺpci. Najlepší výsledok 100% mali číslice 2,6,7 a 9. Najhorší relatívny výsledok dosiahla číslica 1 s pravdepodobnosťou rozpoznania 79,4%.

Tabuľka počtu celých rozpoznávaných slov		
☺	☹	Σ
35	15	50
70,0%	30,0%	100,0%

Tab. 3.9 Úspešnosť klasifikácie slov v korelačnej metóde

Tabuľka rozpoznania celých slov je v tab.3.9 a obr. 3.20, pravdepodobnosť rozpoznania slov $\varphi_{s_{KM}} = 70,0\%$. Na obr. 3.17 je znázornený podiel zobrazených znakov pri chybe rozpoznania slova. Najvyššie zastúpenie 9,0% má znak „3“.

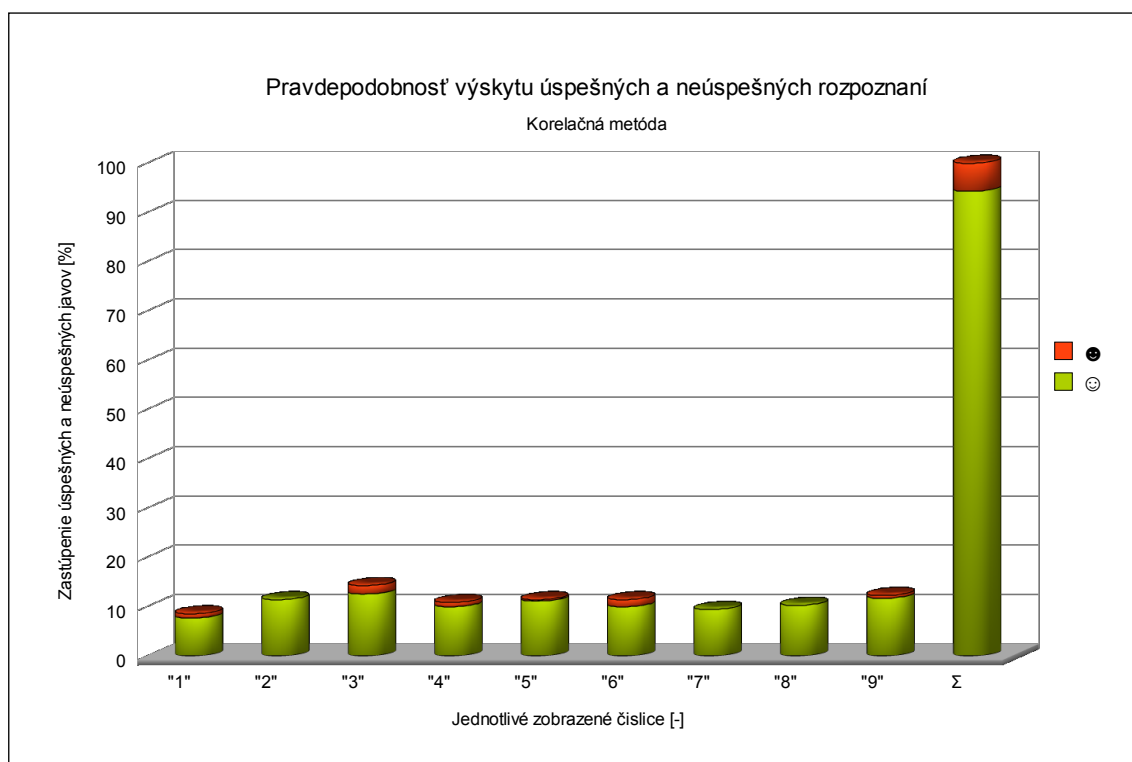


Obr. 3.17: Podiel chyby zobrazených číslic pri neúspešnom rozpoznaní

Korelačná metóda poskytla výsledky, ktoré nie sú zanedbateľné, no na druhej strane je to metóda pomerne selektívna a náročná na výpočtový výkon a čas.

Tabuľka pravdepodobností výskytu jednotlivých javov pri rozpoznávaní											
Zobrazený znak	Počet rozpoznaní pre jednotlivé číslice [%]									Celkom [%]	
	1	2	3	4	5	6	7	8	9	☺	☹
"1"	7,7	0,0	0,0	0,9	0,0	0,0	0,0	0,0	0,0	7,7	0,9
"2"	0,0	11,4	0,0	0,0	0,0	0,0	0,0	0,0	0,0	11,4	0,0
"3"	1,1	0,0	12,6	0,0	0,3	0,0	0,0	0,3	0,0	12,6	1,7
"4"	0,6	0,0	0,0	10,0	0,0	0,0	0,0	0,3	0,0	10,0	0,9
"5"	0,0	0,0	0,0	0,0	11,1	0,0	0,0	0,3	0,0	11,1	0,3
"6"	0,0	0,0	0,6	0,0	0,9	10,0	0,0	0,0	0,0	10,0	1,4
"7"	0,0	0,0	0,0	0,0	0,0	0,0	9,4	0,0	0,0	9,4	0,0
"8"	0,0	0,0	0,0	0,0	0,0	0,0	0,0	10,3	0,0	10,3	0,0
"9"	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,3	11,7	11,7	0,6
Σ [%]										94,3	5,7

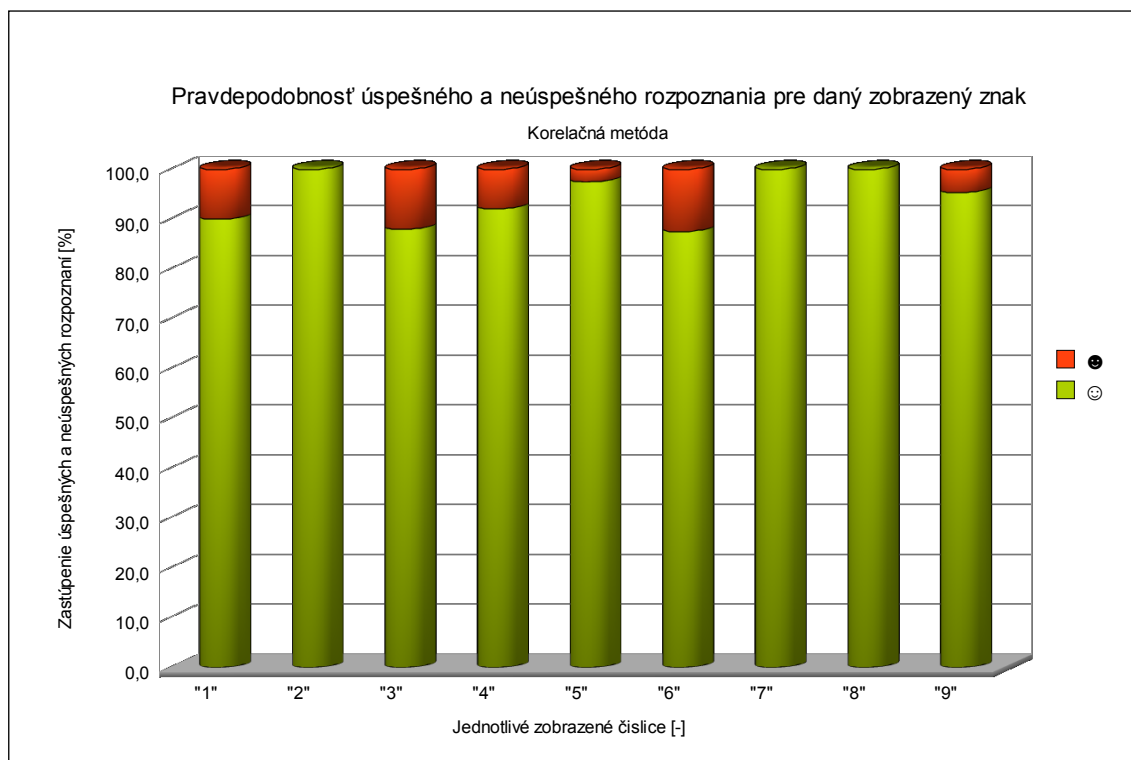
Tab. 3.10 Pravdepodobnosť výskytu javov v korelačnej metóde



Obr. 3.18: Graf výskytu rozpoznaní v korelačnej metóde

Tabuľka pravdepodobností počtu rozpoznania pre daný zobrazený znak											
Zobrazený znak	Počet rozpoznání pre jednotlivé číslice [%]									Celkom [%]	
	1	2	3	4	5	6	7	8	9	☺	☹
"1"	90,0	0,0	0,0	10,0	0,0	0,0	0,0	0,0	0,0	90,0	10,0
"2"	0,0	100,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	100,0	0,0
"3"	8,0	0,0	88,0	0,0	2,0	0,0	0,0	2,0	0,0	88,0	12,0
"4"	5,3	0,0	0,0	92,1	0,0	0,0	0,0	2,6	0,0	92,1	7,9
"5"	0,0	0,0	0,0	0,0	97,5	0,0	0,0	2,5	0,0	97,5	2,5
"6"	0,0	0,0	5,0	0,0	7,5	87,5	0,0	0,0	0,0	87,5	12,5
"7"	0,0	0,0	0,0	0,0	0,0	0,0	100,0	0,0	0,0	100,0	0,0
"8"	0,0	0,0	0,0	0,0	0,0	0,0	0,0	100,0	0,0	100,0	0,0
"9"	2,3	0,0	0,0	0,0	0,0	0,0	0,0	2,3	95,3	95,3	4,7

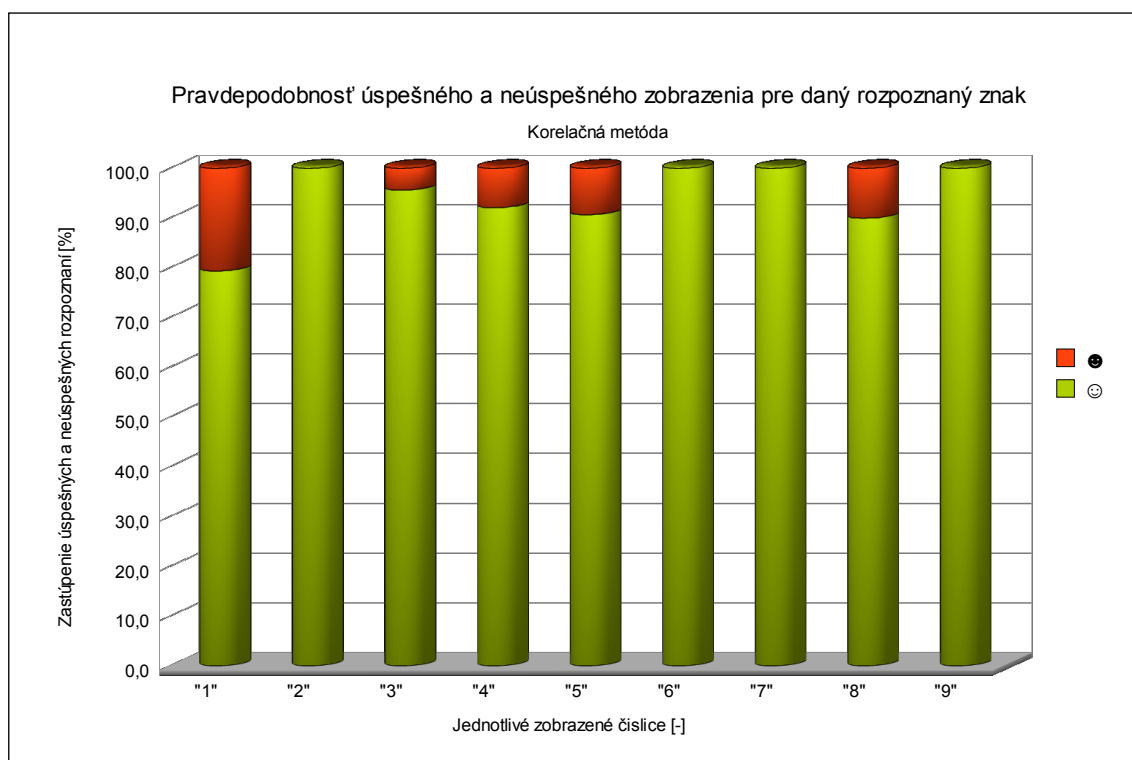
Tab. 3.11 Pravdepodobnosť rozpoznania v korelačnej metóde



Obr. 3.19: Graf pravdepodobností rozpoznania v korelačnej metóde

Tabuľka pravdepodobností počtu zobrazenia pre daný rozpoznávaný znak										
Počet rozpoznání pre jednotlivé číslice [%]										
Zobrazený znak		1	2	3	4	5	6	7	8	9
	"1"	79,4	0,0	0,0	7,9	0,0	0,0	0,0	0,0	0,0
	"2"	0,0	100,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	"3"	11,8	0,0	95,7	0,0	2,3	0,0	0,0	2,5	0,0
	"4"	5,9	0,0	0,0	92,1	0,0	0,0	0,0	2,5	0,0
	"5"	0,0	0,0	0,0	0,0	90,7	0,0	0,0	2,5	0,0
	"6"	0,0	0,0	4,3	0,0	7,0	100,0	0,0	0,0	0,0
	"7"	0,0	0,0	0,0	0,0	0,0	0,0	100,0	0,0	0,0
	"8"	0,0	0,0	0,0	0,0	0,0	0,0	0,0	90,0	0,0
	"9"	2,9	0,0	0,0	0,0	0,0	0,0	0,0	2,5	100,0
Celkom [%]	☺	79,4	100,0	95,7	92,1	90,7	100,0	100,0	90,0	100,0
	☹	20,6	0,0	4,3	7,9	9,3	0,0	0,0	10,0	0,0

Tab. 3.12 Pravdepodobnosť zobrazenia v momentovej metóde

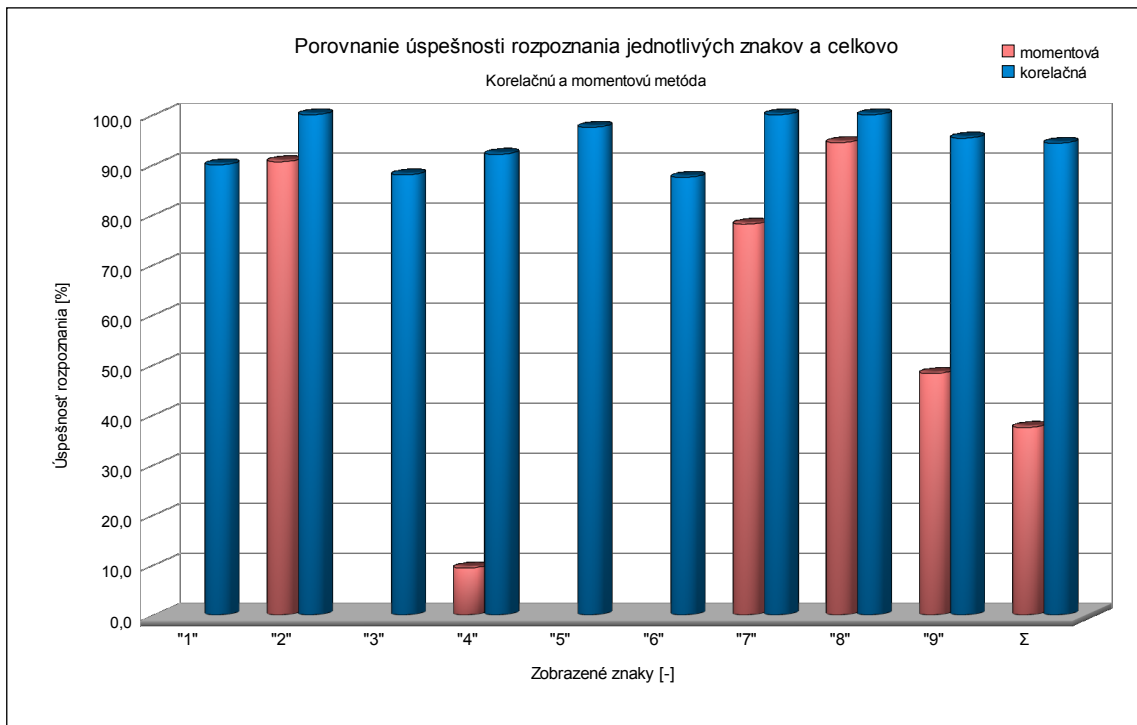


Obr. 3.20: Graf pravdepodobností zobrazenia v korelačnej metóde

3.6 Porovnanie výsledkov oboch metód

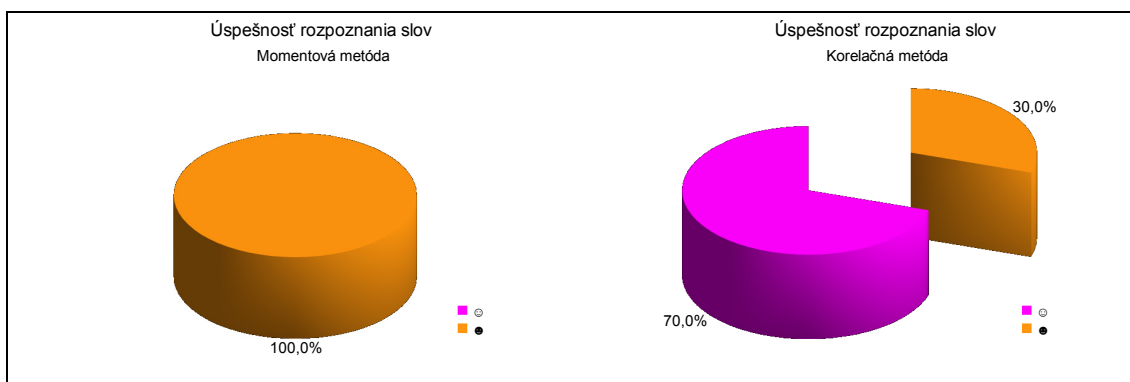
Výsledné porovnanie výsledkov oboch metód v dvoch grafoch.

Prvý na obr. 3.21 porovnáva úspešné rozpoznania znakov „1“ až „9“ symbol „Σ“ všetky úspešné rozpoznania momentovej a korelačnej metódy. Celková úspešnosť znakov pri momentovej metóde bola $\varphi_{Z_{MM}} = 37,5\%$, pri korelačnej metóde $\varphi_{Z_{MM}} = 94,3\%$.



Obr. 3.21: Úspešnosť rozpoznania znakov v momentovej a korelačnej metóde

Druhý graf na obr. 3.22 zobrazuje úspešnosť rozpoznania celých slov pri momentovej metóde $\varphi_{S_{MM}} = 0,0\%$ a pri korelačnej metóde $\varphi_{S_{KM}} = 70,0\%$



Obr. 3.22: Úspešnosť rozpoznania slov v momentovej a korelačnej metóde

4 ZÁVER

V diplomovej práci na tému OCR modul pre rozpoznanie písmen a čísiel boli spracované metódy použité v procese rozpoznávania. Nie každá z uvedených metód je vhodná pre všetky typy znakov obsiahnutých v obraze. Hlavným cieľom bolo navrhnúť a naprogramovať metódu pre rozpoznávanie čísiel v kontrolných obrazoch od „t-zones“. Ako programové prostredie bol použitý Matlab[®], v prípade klasifikácie, v momentovej metóde, toolbox „*Neural Network Pattern Recognition Tool*“

Pri segmentácii bola dosiahnutá úspešnosť rozdelenia znakov (znaková úspešnosť) $\varphi_Z = 98,3\%$ a úspešnosť rozdelenia celých obrazov $\varphi_O = 94,0\%$. Výsledok je akceptovateľný, pretože nízka úspešnosť segmentácie pri rozpoznávaní má za následok zlyhanie nasledujúcich častí, a tým pádom aj rozpoznania.

Pri výslednej klasifikácii v momentovej metóde bola úspešnosť rozpoznania znakov len $\varphi_{Z_{MM}} = 37,5\%$, ktorá je nedostačujúca a teda úspešnosť rozpoznania celých slov je $\varphi_{S_{MM}} \cong 0,0\%$. Teoretická hranica znakovej úspešnosti pre nenulovú úspešnosť rozpoznania celých slov je $\varphi_Z = 85,71\%$, čo odpovedá jednému chybnému rozpoznaniu znaku zo siedmich znakov, obsiahnutých v slove. Možná príčina spočíva v tom, že obrazové momenty pri jednotlivých znakoch nie sú natoľko výrazne odlišené v osemdimensionálnom priestore, a preto výsledný klasifikátor (NN) nie je schopný dosiahnuť vyššiu úspešnosť.

Korelačná metóda bola navrhnutá čisto z poznatkov vlastností jej funkcie. Tento spôsob nie je veľmi sofistikovaný pre jeho vysoké nároky výpočtu, no v danom prípade dosiahol výsledky ktoré je možné považovať za úspešné. Úspešnosť rozpoznania znakov pohybovala na hranici $\varphi_{Z_{KM}} = 94,3\%$. Pri predchádzajúcej úvahe rozloženia jedného chybného znaku na slovo by vychádzala úspešnosť rozpoznania slov $\varphi_Z = 39,9\%$. Vo väčšine prípadov chyba nastala neúspešnou segmentáciou, a vznikli v jednom slove viacnásobné chyby. Úspešnosť rozpoznania slov bola $\varphi_{S_{KM}} = 70,0\%$. Doba trvania rozpoznania jedného slova sa pohybovala na úrovni do 30 sekúnd (závisí od výkonu PC a spôsobu spustenia programu).

Výsledný program by bolo možné využiť pre odosielanie SMS z brány „t-zones“ bez nutnosti prepisovania znakov obsiahnutých v kontrolnom obraze. Samozrejme je možné aj zneužitie pre automatické odosielanie SMS, akéhokoľvek obsahu a v akomkoľvek množstve. Tým by mohla byť, až do ošetrenia typu autentifikácie na stránke, zahľtená pamäť telefónu osoby ktorej by program neustále posielal SMS. Samotné časové obmedzenie posielania správ cez bránu „t-zones“ je 30 sekúnd. Rýchlosť rozpoznania by v tomto prípade bola postačujúca.

Budúcnosť rozpoznávania podobných typov obrazov spočíva v kombinácii jednotlivých metód, v možnosti preskakovať jednotlivé segmenty použité pri rozpoznávaní, ako aj rekurentne sa vracieť k segmentom a ovplyvniť ich výsledky s využitím množstva už získaných informácií o obraze

V prílohe B sa v zložke [OBRAZY] nachádza množina 99 obrázkov stiahnutých z „t-zones“, pre možnosť práce a testovania metód v prípade zmeny autentifikácie alebo zrušenia brány.

LITERATÚRA

- [1] BARTOŇEK, D. Počítačová grafika I. Studijní opora. Brno: FAST VUT v Brně, 2005.
- [2] HORÁK, K., KALOVÁ, I., PETYOVSÝ, P., RICHTER, M. Počítačové vidění. Elektronické skriptum. Brno: FIT VUT v Brně, 2008.
- [3] JAN, J. Číslíková filtrace, analýza a restaurace signálů. Brno: FEKT VUT v Brně, 1997.
- [4] MORI G., MALIK J. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. University of California, Berkeley, 2003. Dostupné na [www:<http://www.cs.sfu.ca/~mori/research/gimpy>](http://www.cs.sfu.ca/~mori/research/gimpy)
- [5] AdaBoost- Wikipedia ,the free encyclopedia. [online], This page was last modified 9. April 2009, Dostupné na [www:<http://en.wikipedia.org/wiki/AdaBoost >](http://en.wikipedia.org/wiki/AdaBoost)
- [6] Grafické transformace- Wikipedia ,the free encyclopedia. [online], This page was last modified 9. April 2009, Dostupné na [www:<http://cs.wikipedia.org/wiki/Grafické_transformace>](http://cs.wikipedia.org/wiki/Grafické_transformace)
- [7] Support vector machine-Wikipedia, the free encyclopedia. [online], This page was last modified 13:48, 1 May 2007, Dostupné na [www:<http://en.wikipedia.org/wiki/Support_vector_machine >](http://en.wikipedia.org/wiki/Support_vector_machine)
- [8] HU,M.-K.:Visual pattern recognition by moment invariants. Information Theory, IEEE Transactionson, ročník 8,1962:s.179–187,ISSN0018-9448.
- [9] Neural network - Wikipedia ,the free encyclopedia. [online], This page was last modified on 24 December 2009 at 20:47., Dostupné na [www:<http://en.wikipedia.org/wiki/Neural_network>](http://en.wikipedia.org/wiki/Neural_network)
- [10] DUDA, Richard O.; HART, Peter E. ; STORK, David G. . *Pattern Classification*. 2nd Edition. New York : Wiley, November 2000. 680 s. ISBN 978-0-471-05669-0.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

$1D$	jednorozmerný, funkcia jednej premennej
$2D$	dvojrozmerný, funkcia dvoch premenných
MCR	Matlab Component Runtime, knižnica potrebná pre spustenie aplikácie
MSE	Mean Squared Error, stredná kvadratická odchýlka
$Matlab^{\circledR}$	programové prostredie vedecké výpočty
NaN	Not a Number, irelevantnosť výsledku
NN	Neural Network, neurónová sieť
OCR	Optical Character Recognition, rozpoznávanie znakov v obraze
PC	Personal Computer, osobný počítač
$pixel$	obrazový bod
SMS	Short Message Service, textová správa
SAP	Stand Alone Application
${}_{1D}R$	jednorozmerná korelačná funkcia, v Matlabe <code>xcorr</code>
${}_{2D}R$	dvojrozmerná korelačná funkcia, v Matlabe <code>xcorr2</code>
R	rotácia
R'	transformačná matica pre rotáciu
S	transformačná matica pre rotáciu
η_{pq}	momenty
I	momentové invarianty
P_{pq}	polynóm
f_{NN}	prenosová funkcia neurónu
X_{NN}	vstup neurónov
W_{NN}	váhy neurónu
p_{NN}	prah neurónu
P_m, Q_m	body priestoru pre výpočet euklidovskej vzdialenosti
$H(x)$	klasifikátor

R_p	intenzita červenej farby v čísliciach
$R_{\check{c}}$	intenzita červenej farby v čiernych krivkách
$R_{\check{s}}$	intenzita červenej farby v šedom prechode
G_p	intenzita zelenej farby v čísliciach
$G_{\check{c}}$	intenzita zelenej farby v čiernych krivkách
$G_{\check{s}}$	intenzita zelenej farby v šedom prechode
$B_{\check{s}}$	intenzita modrej farby v čísliciach
$B_{\check{s}}$	intenzita modrej farby v čiernych krivkách
$B_{\check{s}}$	intenzita modrej farby v šedom prechode
Z	rozmer obrazu
$K_{\check{s}}$	kontrast medzi ľavou a pravou stranou šedého prechodu
K_p	kontrast mediánu zelenej zložky číslic
P_p	prah
$AA(x, y)$	obraz po prahovaní
$BB(x, y)$	čierna krivka v obraze
$CC(x, y)$	obraz bez čiernej krivky
$CC'(x, y)$	obraz bez čiernej krivky a šumu
$DD(x, y)$	opravený obraz
$G(x, y)$	vstupný obraz (zelená zložka)
$P(x, y)$	jednotlivé obrazové segmenty
τ, σ	posun
φ_Z	znaková úspešnosť segmentácie
a_Z	počet nesprávne rozsegmentovaných znakov
x_Z	počet všetkých znakov
φ_O	obrazová úspešnosť
a_O	počet nesprávne rozsegmentovaných obrazov
x_O	je počet všetkých obrazov
m_{pq}	momenty pre diskkrétne hodnoty obrazovej funkcie
M_{pq}	centralizované momenty pre diskkrétne hodnoty obrazovej funkcie

\tilde{M}_{pq}	normalizované centralizované momenty pre diskkrétne hodnoty obrazovej funkcie
\tilde{I}	momentové invarianty pre diskkrétne hodnoty obrazovej funkcie
$\varphi_{Z_{MM}}$	úspešnosť rozpoznania znakov pri momentovej metóde
$\varphi_{Z_{KM}}$	úspešnosť rozpoznania znakov pri korelačnej metóde
$\varphi_{S_{MM}}$	úspešnosť rozpoznania slov pri momentovej metóde
$\varphi_{S_{KM}}$	úspešnosť rozpoznania slov pri korelačnej metóde
$H_{i,j}$	maximum 2D korelačnej funkcie
U	neznámy rozsegmentovaný znak
$T_{i,j}$	normálový znak

ZOZNAM PRÍLOH

A	Zdrojové kódy	49
A.1	Stiahnutie obrazu z T-zones.....	50
A.2	Prahovanie	50
A.3	Odstránenie šumu	51
A.4	Vyhľadzovanie	51
A.5	Segmentácia	52
A.6	Korelácia	52
A.7	Ťažisko.....	53
A.8	Maximum a priemer.....	53
A.9	Základný momentový invariant	54
A.10	Centralizovaný momentový invariant.....	54
A.11	7 Hu momentov + 1	54
B	DVD nosič	56
B.1	Obsah DVD.....	57

A ZDROJOVÉ KÓDY

A.1 Stiahnutie obrazu z T-zones

```
a=1;down=0; % pomocné premenné
try % v prípade erroru preskočí na catch
    [b,map]=imread('http://sms.t-zones.cz/open/captcha.jpg'); % načítanie obrázku pomocou http
    down=1;
    aa=num2str(a);
    d=strcat('tpic',aa,'.bmp'); % príprava názvu uloženého obrazu
    imwrite(b,d,'bmp'); % uloženie obrazu
catch
end

if down==1 % informácia pre užívateľa
    disp('Downloading picture... OK');
else
    disp('Downloading picture... FAILED');
end
```

A.2 Prahovanie

```
e=size(b); %rozmer obrazu b (green zložka)

%prahovanie obrazu s čiarou z obrazu b(green zložka)

for riadok=1:e(1)
    for stlpec=1:e(2)
        if b(riadok,stlpec)<100 %prah nastavený na 100
            b(riadok,stlpec)=255;
        else b(riadok,stlpec)=0;
        end %výsledok binárny obraz
    end
end

imwrite(b,'thres_all.bmp','bmp'); %uloženie obrazu s čiarou

%prahovanie ciary z obrazu a(red zložka)

for riadok=1:e(1)
    for stlpec=1:e(2)
        if a(riadok,stlpec)<140 %prah nastavený na 100
            a(riadok,stlpec)=255;
        else a(riadok,stlpec)=0;
        end %výsledok binárny obraz
    end
end

err=a; %čiara vlozena do premennej err
imwrite(err,'curve.bmp','bmp'); %uloženie obrazu čary

%odstránenie čiary

aaa=b-a; %odstránenie čary odčítaním
imwrite(aaa,'without_curve.bmp','bmp'); % uloženie obrazu bez čiary
```

A.3 Odstránenie šumu

```
%odstranenie sumu okno 3x3

[m,n]=size(aaa);m=m-1;n=n-1;
for a=2:m
    for b=2:n
        count=0;
        for aa=1:3
            for bb=1:3
                if aaa(a-2+aa,b-2+bb)==255
                    count=count+1;
                end
            end
        end
        if count<3
            aaa(a,b)=0;
        end
    end
end
```

A.4 Vyhľadovanie

```
%vyhladzovanie po čare, funkčnosť podľa popisu v kap.3.2.2
%vstupný obraz aaa, výstupný obraz aaaa

[m,n]=size(aaa);m=m-1;n=n-1;
aaaa=zeros(m+21,n+1);aaaa(11:60,1:n+1)=aaa;
errr=zeros(m+21,n+1);errr(11:60,1:n+1)=err;
for a=11:(m+10)
    for b=2:n
        count=0;
        count2=0;
        for aa=1:3
            for bb=1:3
                if aaaa(a-2+aa,b-2+bb)==255
                    count=count+1;
                end
                if errr(a-2+aa,b-2+bb)==255
                    count2=count2+1;
                end
            end
        end
        if count>1 && count2>1 && aaaa(a-1,b)==255 &&...
... (aaaa(a+3,b)==255 || aaaa(a+5,b)==255 )
            aaaa(a,b)=255;
        end
    end
end
```

A.5 Segmentácia

```
%segmentácia, funkčnosť podľa popisu v kap.3.2.3
%temptr,bound - dočasné hranice
%bound2 - výsledné hranice

for a=1:7
    N1=hist((aaaa(:,temptr(a):temptr(a+1))))';

    b=1;g=size(N1);
    while N1(g(1),b)==0
        b=b+1;
    end
    bound(1,a)=b;

    b=70;
    while N1(g(1),b)==0
        b=b-1;
    end
    bound(2,a)=b;

a=a+1;
end
bound;
%konecne naplnenie hranicami
for a=1:7
    bound2(1,a)=temptr(a);
    bound2(2,a)=temptr(a+1);
    bound2(3:4,a)=bound(:,a);
end
```

A.6 Korelácia

```
for g=1:9                                     %cyklus pre 9 adresárov s triedami
    cd(strcat(dir1(1:end-6),'cisla_triedy\',num2str(g)));% adresár
    files = dir;                               %štruktúra súborov v adresári
    nofiles = size(files);                     %počet súborov v adresári
    q=1;ctemp=0;                               %pomocné premenné
    for a=1:nofiles(1)                         %cyklus pre všetky súbory v adr.
        d=files(a).name;
        if d(end)=='p';
            ctemp=ctemp+1;
            x=double(imread(d));%načítanie obrázka z danej triedy
            cd(dir1);
            [centre_x,centre_y] = c_o_m(x);      %ťažisko
            cd(strcat(dir1(1:end-6),'cisla_triedy\',num2str(g)));
            f=size(x);
            b=double(zeros(51,41)); %okno rovnakej veľkosti
            Y=(26-centre_y+1);X=(21-centre_x+1); %súradnice
            b(Y:f(1)+Y-1,X:f(2)+X-1)=x./255;   %posun do okna
            c=xcorr2(aa,b);b1=sum(sum(b));      %2D korelácia
            c1(ctemp,g)=max(max(c));           %vysledky v c1
        end
    end
end
end
```

A.7 Ťažisko

```
function [centre_x,centre_y] = c_o_m(image)
a = 0; %súradnica v smere x (horizontálne)
b = 0; %súradnica v smere y (vertikálne)
count = 0; %pomocná premenná
[sizey,sizex] = size(image);%veľkosť obrazu

    for i=1:sizex %cyklus pre všetky stĺpce
    for j=1:sázet %cyklus pre všetky prvky stĺpca
        if (image(j,i) > 0) %podmienka pre pixel s nenulovou h.
            a = a + i; %pričítanie x-ovej súradnice pixelu k a
            b = b + j; %pričítanie x-ovej súradnice pixelu k b
            count = count + 1;
        end
    end
end

a = floor(a/count); %priemerná hodnota súradnice x nenulového px
b = floor(b/count); %priemerná hodnota súradnice y nenulového px
centre_x=a; %súradnica x ťažiska
centre_y=b; %súradnica y ťažiska
end
```

A.8 Maximum a priemer

```
a=size(c1); %pomocná premenná

for b=1:a(2) %cyklus pre 9 stĺpcov výsledku c1
    lett(:,b)=sort(c1(:,b),1); %zoradenie
    priem(b)=mean(lett(end-19:end,b)); %stredná hodnota z 20 prvkov
end

[r0,t0]=max(priem); %maximum pre priemerné hodnoty,...
t0-poradie stĺpca, klasifikovaná výsledná trieda podľa priemeru
xcorr2

[r1,t1]=max(max(c1)); %maximum pre priemerné hodnoty,...
t1-poradie stĺpca, klasifikovaná výsledná trieda podľa maxima
xcorr2
```

A.9 Základný momentový invariant

```
function m = moment(image,p,q) %image je vstupný obraz, p,q
definujú rád momentu, ak p,q=0 výsledok je vlastne suma bielych
pixelov v obraze
    m = 0;
    [sizey,sizeX] = size(image);

    for i=1:sizeX
        for j=1:sizey
            if (image(j,i) > 0)
                m = m + (i^p * j^q);
            end
        end
    end
end
```

A.10 Centralizovaný momentový invariant

```
function Mom = c_moment(image,p,q,centre_x,centre_y) %image je
vstupný obraz, p,q definujú rád momentu, centre_x,centre_y definujú
ťažisko

    Mom = 0;
    [sizey,sizeX] = size(image);

    for i=1:sizeX
        for j=1:sizey
            if (image(j,i) > 0)
                Mom = Mom + ((i-centre_x)^p * (j-centre_y)^q);
            end
        end
    end
end
```

A.11 7 Hu momentov + 1

```
function [I_all]=t_mom(bound2,aaaa)

for a=1:7 %7 číslic
    x1=bound2(1,a);x2=bound2(2,a);y1=bound2(3,a);y2=bound2(4,a);
    image=aaaa(y1:y2,x1:x2)./255; %selekcia číslice
    [centre_x,centre_y]=c_o_m(image); %ťažisko
    uu00=(moment(image,0,0)); %základný momentový invariant
    %centralizované, normalizované momentové invarianty
    nn11=(c_moment(image,1,1,centre_x,centre_y))/(uu00^2);
    nn20=(c_moment(image,2,0,centre_x,centre_y))/(uu00^2);
    nn02=(c_moment(image,0,2,centre_x,centre_y))/(uu00^2);
    nn12=(c_moment(image,1,2,centre_x,centre_y))/(uu00^2.5);
    nn21=(c_moment(image,2,1,centre_x,centre_y))/(uu00^2.5);
    nn30=(c_moment(image,3,0,centre_x,centre_y))/(uu00^2.5);
    nn03=(c_moment(image,0,3,centre_x,centre_y))/(uu00^2.5);
end
```

```

I_one = nn20 + nn02;

I_two = (nn20 - nn02)^2 + (2*nn11)^2;

I_thr = (nn30 - 3*nn12)^2 + (3*nn21 - nn03)^2;

I_fou = (nn30 + nn12)^2 + (nn21 + nn03)^2;

I_fiv = (nn30 - 3*nn12)*(nn30 + nn12)*((nn30 + nn12)^2 - ...
3*(nn21 + nn03)^2) + (3*nn21 - nn03)*(nn21 + nn03)*(3*...
(nn30 + nn12)^2 - (nn21 + nn03)^2);

I_six = (nn20 - nn02)*((nn30 + nn12)^2 - 3*(nn21 + nn03)^2) + ...
4*nn11*(nn30 + nn12)*(nn21 + nn03);

I_sev = (3*nn21 - nn03)*(nn30 + nn12)*((nn30 + nn12)^2 - 3*...
(nn21 + ... nn03)^2) - (nn30 - 3*nn12)*(nn21 + nn03)*(3*...
(nn30 + nn12)^2 - (nn21 + nn03)^2);

I_eig = (nn20*nn02) - nn11^2;

I_all(a,:) = [I_one      I_two^(1/2) I_thr^(1/3) I_fou^(1/3)
              I_fiv^(1/3) I_six^(1/2) I_sev^(1/3) I_eig^(1/2)];

end

end

```

B DVD NOSIČ

B.1 Obsah DVD

DP_KAPUSTA

» [DIPLOMOVA_PRACA]

» diplomova_praca.pdf

» [MCR]

» MCRInstaller.exe

» [OBRAZY]

» ~.bmp (1)

» ⋮ ⋮

» ~.bmp (99)

» [TMOB1]

» c_moment.m

» c_o_m.m

» data.mat

» main.m

» moment.m

» t_img.m

» t_mom.m

» t_neur.m

» t_pic.m

» [TMOB2]

» [ciska_triedy]*

» [MATLAB]

» c_o_m.m

» final1.m

» mainf.m

» t_img.m

» t_pic.m

» tpic1.bmp

- » **[TMOB2_ALL]**
 - » [cisla_triedy]*
 - » [MATLAB]
 - » c_o_m.m
 - » final1.m
 - » mainf.m
 - » t_img.m
 - » t_pic.m
 - » tpic1.bmp
- » **[TMOB2_EXE]**
 - » [cisla_triedy]*
 - » [MATLAB]
 - » mainf.exe
 - » mainf.prj
 - » mainf_delay_load.c
 - » mainf_main.c
 - » mainf_mcc_component_data.c
 - » readme.txt
 - » mccExcludedFiles.log
 - » OCR t_zones.lnk
- » **DP_KAPUSTA.zip**

```

*» [cisla_triedy]
    » [1]          (1)
    » ⋮           ⋮
    » [9]          (9)
        » ~.bmp    (1)
        » ⋮       ⋮
        » ~.bmp    (25)

```