



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ROZPOZNÁVÁNÍ ZNAKŮ POMOCÍ UMĚLÉ INTELIGENCE

CHARACTERS RECOGNIZING BY ARTIFICIAL INTELLIGENCE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

KAREL MOŽNÝ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. LUDĚK ČERVINKA

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Karel Možný

ID: 109700

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Rozpoznávání znaků pomocí umělé inteligence

POKYNY PRO VYPRACOVÁNÍ:

Úkolem studenta je provést rozpoznávání alfa-numerických znaků na pořízeném snímku. V první části práce je úkolem studenta se seznámit s metodami zpracování obrazu, zvláště pak lokalizace jednotlivých znaků na snímku, a principy klasifikace použitím umělé neuronové sítě. V další části bude úkolem studenta realizace klasifikace jednotlivých znaků. Ověření vlivu různých druhů příznaků a volba jejich optimální množiny, vliv topologie sítě na rychlost učení a přesnost klasifikace. Celý tento projekt bude realizován v jazyce C/C++ s použitím knihovny OpenCV.

DOPORUČENÁ LITERATURA:

[1] HLAVÁČ, Václav, SEDLÁČEK, Miloš. Zpracování signálů a obrazů. Praha : ČVUT, 2005. 255 s. ISBN 80-01-03110-1.

[2] HORÁK, Karel, KALOVÁ, Ilona, PETYOVSKÝ, Petr, RICHTER, Miloslav. Počítačové vidění. Brno : VUT, 2008

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Luděk Červinka

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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.

Abstrakt

Tato práce se zabývá problematikou rozpoznávání znaků v obraze pomocí neuronových sítí s využitím počítačového vidění a statistických momentů. Dále se zabývá návrhem této sítě a implementací této problematiky v programovacím jazyce C++.

Klíčová slova

rozpoznávání textu, OCR, umělé neuronové sítě, segmentace textu, předzpracování obrazu, počítačové vidění

Abstract

This thesis describes problems of character recognition in digital picture and how to solve those problems using artificial neural networks, computer vision and statistical moments. Further it describes design of this network and implementation of solutions in C++ programming language.

Keywords

character recognition, OCR, artificial neural network, text segmentation, picture preprocessing, computer vision

Bibliografická citace

MOŽNÝ, K. *Rozpoznávání znaků pomocí umělé inteligence*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 60 s. Vedoucí bakalářské práce Ing. Luděk Červinka.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma rozpoznávání znaků pomocí umělé inteligence jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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: **31. května 2010**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Ludku Červinkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **31. května 2010**

.....
podpis autora

Obsah

1. ÚVOD.....	4
2. ZÍSKÁNÍ DIGITÁLNÍHO OBRAZU.....	5
2.1 Digitalizace dvourozměrné obrazové funkce.....	5
2.1.1 Interval vzorkování.....	5
2.1.2 Vzorkovací mřížka.....	6
2.1.3 Kvantování hodnot.....	6
3. CCD SNÍMAČE.....	7
3.1 Princip CCD snímačů	7
3.2 Získání a reprezentace dat z CCD snímačů.....	7
3.3 Teplotní šum, čtecí šum.....	8
4. BODOVÉ JASOVÉ TRANSFORMACE.....	9
4.1 Jasové korekce.....	9
4.2 Změna jasové stupnice.....	10
5. GEOMETRICKÉ TRANSFORMACE.....	11
6. HOUGHOVA TRANSFORMACE	12
7. LOKÁLNÍ PŘEDZPRACOVÁNÍ	14
7.1 Filtrace-vyhlazování obrazu.....	14
7.1.1 Filtr s Gaussovým rozložením.....	15
7.2 Další filtry.....	17
7.3 Detekce hran, ostření.....	18
7.3.1 Typy operátorů.....	19
7.3.2 Cannyho hranový detektor.....	20
8. BARVY V DIGITÁLNÍM OBRAZE.....	23
8.1 Barevné modely.....	23
8.1.1 Model RGB, RGBA.....	23
8.1.2 Model CMY, CMYK.....	23
8.1.3 Model HSV.....	23
8.1.4 Model HSL.....	24
8.2 Transformace barev.....	24

8.2.1 Otsuho prahovací metoda (Otsu thresholding).....	25
9. STATISTICKÉ MOMENTY.....	29
9.1 Funkce generující moment.....	30
9.2 Neortogonální momenty.....	32
9.3 Kartézské momenty.....	32
9.4 Centralizované momenty.....	33
9.5 Symetrické vlastnosti.....	33
9.6 Množina invariantních (hu) momentů.....	35
10. NEURONOVÉ SÍTĚ.....	37
10.1 Procesní jednotka-Neuron.....	37
10.2 Spojení neuronů.....	39
10.3 Aktivace a výstupy.....	39
10.4 Topologie Sítí.....	40
10.4.1 Feed-forward.....	40
10.4.2 Rekurentní síť.....	40
10.5 Učení Umělých neuronových sítí.....	41
10.5.1 Možnosti učení.....	41
10.5.2 Back-Propagation.....	42
10.5.3 Algoritmus RPROP.....	44
10.5.4 Kaskádové učení.....	45
11. ALGORITMUS PROGRAMU.....	46
11.1 Barevný převod obrazu.....	47
11.2 Filtrování šumu.....	47
11.3 Prahování.....	47
11.4 Kompenzace pootočení obrazu.....	48
11.5 Segmentace stránky na řádky.....	49
11.6 Segmentace řádků na znaky.....	49
11.7 Výpočet momentů.....	50
11.8 Trénovací a klasifikační data.....	50
11.9 Použité neuronové síť.....	51
11.9.1 Síť učená algoritmem Rprop.....	51

11.9.2	Sít' učená kaskádním učením.....	52
11.9.3	Sít' vytvořená v Matlabu.....	53
11.9.4	Zhodnocení funkce sítí.....	54
12.	FUNKCE VYTVOŘENÝCH PROGRAMŮ.....	55
12.1	Použité knihovny.....	55
12.1.1	OpenCV.....	55
12.1.2	FANN.....	55
12.2	Hlavní program.....	55
12.3	Program pro učení metodou Rprop.....	56
12.4	Program pro kaskádové učení.....	56
12.5	Soubor s daty pro učení.....	56
13.	ZÁVĚR.....	57
14.	SEZNAM LITERATURY.....	58

1. ÚVOD

Optické rozpoznávání znaků (OCR z angl. Optical character recognition) je mechanický nebo elektronický překlad obrazů psaného nebo tištěného textu do formy zpracovatelné na počítači případně jiných strojích. OCR je často používáno k převodu knih a dokumentů do elektronických souborů například kvůli digitalizaci, archivování nebo publikování na internetu. Umožňuje editovat tento text vyhledávat v něm pomocí fulltextového hledání, snáze jej skládat zobrazovat a tisknout, aniž by docházelo k poškození originálu díla nebo vzniku artefaktů při kopírování. Problematika OCR spadá do kategorií umělé inteligence, počítačového vidění a rozpoznávání vzorů.

OCR systémy obvykle potřebují kalibraci pro specifické fonty, dřívější verze vyžadovaly předprogramování pomocí obrazů každého ze znaků a pracovaly pouze na jeden typ fontu. Dnes jsou zcela běžné inteligentní systémy s vysokým stupněm úspěšnosti rozpoznání pro většinu fontů. Některé systémy jsou schopny svůj výstup naformátovat velmi věrohodně včetně obrázků, odstavců a dalších netextových komponentů předlohy.

2. ZÍSKÁNÍ DIGITÁLNÍHO OBRAZU

Matematicky lze obraz vyjádřit pomocí spojité funkce jasu $f(x,y)$ dvou argumentů v rovině. Obraz je definován jako funkce jasu zejména proto, protože jas je veličina, která odpovídá vnímání obrazu člověkem a měří jej většina snímacích zařízení (TV kamery, CCD prvky). Vstupní informací ovšem nemusí být při snímání jen jas, ale mohou jí být i jiné veličiny, jako jsou intenzita rentgenového záření, ultrazvuk nebo tepelné záření. Dále však budeme uvažovat, že vstupním signálem je jasová funkce.

Obrazové funkce lze rozdělit na spojité a diskrétní. Spojitou funkcí je rozuměna funkce, jejíž definiční obor i obor hodnot je spojitý. Pokud definiční obor i obor hodnot tvoří množina diskrétních bodů, funkci nazýváme diskrétní (digitální, číslicovou), která je vhodná pro zpracování pomocí výpočetní techniky.

Pokud je obraz monochromatický (jednobarevný v odstínech šedi). Je reprezentován jedinou obrazovou funkcí $f(x,y)$. Barevný obraz lze definovat jako soubor několika obrazových funkcí, nebo vektorem hodnot – např. jasů pro jednotlivé barevné složky obrazu.

2.1 DIGITALIZACE DVOUROZMĚRNÉ OBRAZOVÉ FUNKCE

Digitální obraz lze získat převodem analogového signálu pomocí digitalizace. Ta spočívá ve vzorkování obrazu do matice o rozměrech $M \times N$ obrazových bodů a v kvantování původní spojité jasové úrovně do K intervalů pro každý vzorek, čímž jasové funkce digitalizovaného obrazu bude nabývat pouze celočíselných hodnot. Čím vyšší počet úrovní K a rozlišení $M \times N$, tím lépe bude obraz aproximován, zvýší se ovšem výpočetní náročnost při dalších operacích.

2.1.1 Interval vzorkování

Interval vzorkování je definován Shannonovým teorémem. Ten říká, že vzorkovací frekvence musí být alespoň dvakrát větší, než nejvyšší frekvence obsažená ve vzorkovaném signálu. Z toho lze vyvodit fyzikální interpretaci pro digitalizovaný obraz: Interval vzorkování je nutné volit tak, aby byl menší, nebo roven polovině rozměru nejmenších detailů v obraze.

2.1.2 Vzorkovací mřížka

Dále je třeba volit vhodné uspořádání bodů pro vzorkování – vybrat vzorkovací mřížku. Výhodné je použití pravidelné vzorkovací mřížky. Existují pouze tři pravidelné n -úhelníky, jejichž síť pokrývá úplně pokrývá rovinu: trojúhelník, čtverec a šestiúhelník. Nejčastěji se využívá čtvercová mřížka. Snadno se technicky realizuje a odpovídá reprezentaci pomocí pravoúhlé matice.

2.1.3 Kvantování hodnot

Další součástí digitalizace je kvantování obrazové funkce. Pro číslicové zpracování je třeba, aby hodnota jasu byla vyjádřena jako číselný údaj. Nejčastěji se používá kvantování do K stejných intervalů. V případě číslicového zpracování je počet úrovní určen počtem použitých bitů. Platí:

$$K = 2^b$$

kde b je počet bitů do kterých je obraz kvantován (obvykle 8).

Počet těchto intervalů musí být dostatečně velký, aby byly přesně zachyceny detaily obrazu a nevznikaly falešné obrysy. Tento jev se stává patrným, pokud počet úrovní kvantizace poklesne pod hodnotu 50, což je přibližný počet úrovní rozlišitelný pro člověka v monochromatickém obraze. Z toho důvodu se někdy využívá nelineárního kvantování s proměnnou délkou intervalu, které zvyšuje rozsah nejnepravděpodobněji zastoupených úrovní jasu.

Lze se také setkat s výrazem binární obraz. Takový obraz uchovává údaje o obrazovém bodě pouze jedním bitem. Kde hodnota 1 označuje objekt a pozadí je označeno hodnotou 0.

3. CCD SNÍMAČE

Nábojově vázané prvky (Charge Coupled Devices — CCDs) jsou velmi citlivé detektory a jejich odezva na osvětlení je lineární. Jsou mechanicky stabilní a především CCD kamery poskytují snímky přímo v digitální podobě, bezprostředně zpracovatelné pomocí počítačů.

3.1 PRINCIP CCD SNÍMAČŮ

Základním principem funkce CCD prvků je vytvoření náboje v polovodiči v důsledku dopadajícího světla, tedy využití fotoelektrického jevu. Vzniklé elektrony se nemohou po čipu volně pohybovat v důsledku svislých a vodorovných negativních potenciálových valů, které tyto elektrony odpuzují. Toto uspořádání vytváří na čipu mřížku tzv. „potenciálových studní“, z nichž nemohou elektrony uniknout. Každá taková studna reprezentuje jeden obrazový bod (pixel), nejmenší část obrazu. Rozměr pixelu a počet pixelů ve svislém a vodorovném směru jsou základními charakteristikami CCD čipu. Jedna ze základních výhod CCD čipů oproti lidskému oku je schopnost akumulace náboje po dlouhou dobu (pixely vystavené většímu množství světla naakumulují více elektronů). Proto mohou CCD postupně nashromáždit dostatek světla i z velmi slabých zdrojů.

3.2 ZÍSKÁNÍ A REPREZENTACE DAT Z CCD SNÍMAČŮ

Jak již bylo řečeno, CCD čip je pokryt sítí elektrod, která udržuje světlem uvolněné elektrony v pixelech. Pokud se na elektrody přivede různé napětí, elektrony mohou být přesouvány mezi sousedními nábojovými studnami. Tak je možné náboj posouvat po ploše čipu. Tento proces je používán pro čtení informace z CCD čipu. Náboj reprezentující úroveň dopadeného světla na jednotlivé pixely je přesunut do výstupního zesilovače, kde je převeden na napětí. Toto napětí se objeví na výstupním pinu CCD čipu. Elektronika pak musí toto napětí změřit pomocí analogově-digitálního převodníku pro každý pixel. Informace o náboji akumulovaném v každém pixelu (a tedy o množství světla, která do každého pixelu dopadlo) tvoří datový soubor

reprezentující obraz. Obsahem obrazového souboru je matice čísel, z nich každé reprezentuje jeden pixel.

Jedním ze základních parametrů zařízení využívající CCD prvky je převodový poměr vyjádřený v elektronech na jednotku převodu z analogového na digitální výstup - tzv. ADU (z angl. Analog to digital unit). Obvyklou hodnotou tohoto parametru je $5 - 1,5e^-/ADU$. Tento převodový poměr je důležitý ve vztahu ke kapacitě pixelů. Každá potenciálová studna je schopna uchovat 50000 až 300000 elektronů v závislosti na jejich rozměrech. Zařízení využívající CCD prvků zpravidla používají 16 bitový A/D převodník s rozlišením 65536 úrovní. Je zřejmé že převádět 50000 e^- na 65536 úrovní nedává smysl a 15 nebo 14 bitový převodník vyhoví k digitalizaci dat z takového čipu. Naopak převedení 300000 e^- na 65536 úrovní vyžaduje zesílení 4 až 5 e^-/ADU , což je docela vhodné.

3.3 TEPLOTNÍ ŠUM, ČTECÍ ŠUM

Jednou z nevýhod technologie CCD je, vznik elektronů v pixelech nejen v důsledku dopadajícího světla, ale i náhodně v závislosti na teplotě okolí, velikosti pixelu, architektuře čipu a výrobní technologii. Tento náboj generovaný tepelně se nazývá temný proud (z důvodu vzniku náboje i za podmínky, že je čip zcela zatemněn), nebo také tepelný šum. Tepelný šum bývá vyjádřen v elektronech za sekundu na pixel při definované teplotě.

Dalším zdrojem šumu způsob jakým je vytvořený náboj z CCD čipu vyčítán – je posouván po čipu a poté převeden na napětí. Elektronika obstarávající tento převod ovšem nepracuje bez šumu, proto vzniká další rušivý faktor - čtecí šum. Tento šum je charakteristický a pokud je použita dostatečně kvalitní elektronika vygenerovaný čtecí šum je natolik malý, že ho lze zanedbat a celkový šum zařízení odpovídá tepelnému šumu samotného čipu.

4. BODOVÉ JASOVÉ TRANSFORMACE

Bodové jasové transformace lze rozdělit do dvou skupin: jasové korekce, kde jas v bodě výstupního obrazu závisí pouze na hodnotě jasu ve vstupním obraze a modifikace jasové stupnice kde je určitá hodnota jasu transformována na jinou hodnotu bez ohledu na pozici v obraze.

4.1 JASOVÉ KOREKCE

Jasové korekce se využívají zejména z důvodu nestejně citlivosti snímacího zařízení vzhledem k geometrickému umístění bodu v obraze, nerovnoměrného osvětlení snímaného objektu a ovlivňování procházejícího světla optickými soustavami.

Jedná-li se o poruchy systematické mohou být potlačeny pomocí jasových korekcí, pokud je známa odchylka každého z obrazových bodů od ideální převodní charakteristiky. Jednou z nejčastějších poruch obrazu je porušení multiplikativním koeficientem $e(i,j)$. Výstupní bod $f(i,j)$ získáme tak, že pro každý původní bod obrazu provedeme operaci

$$f(i, j) = e(i, j)g(i, j) \quad (1)$$

Pro určení degradační transformace e stačí získat obraz o známém průběhu jasové funkce $g(i,j)$ při stálých snímacích podmínkách. Tento obraz $f_c(i,j)$ je vhodné volit s tak, aby jeho kontrast c byl konstantní. Pak lze systematické chyby snímání odstranit dle vztahu

$$g(i, j) = \frac{f(i, j)}{e(i, j)} = \frac{c \cdot f(i, j)}{f_c(i, j)} \quad (2)$$

Tato korekce platí ovšem pouze pro stálé snímací podmínky, jinak se třeba stanovit korekční matici po každé kalibraci snímacího řetězce.

Výpočet podle vztahu (2) může dát i výsledky mimo omezený rozsah jasu digitálního obrazu. V tomto případě je řešením nahrazení hodnot mimo interval

odpovídající krajní hodnotou nebo zhuštění případně posun jasové stupnice tak, aby spadala do tohoto intervalu.

Druhou možností je pořídit obraz objektu f_o , poté obraz za stejných světelných podmínek bez objektu pro korekci nelinearity snímače f_b a obraz se zakrytým objektivem f_f . Pak bude platit

$$f_c(i, j) = M \frac{f_o(i, j) - f_b(i, j)}{f_f(i, j) - f_b(i, j)} \quad (3)$$

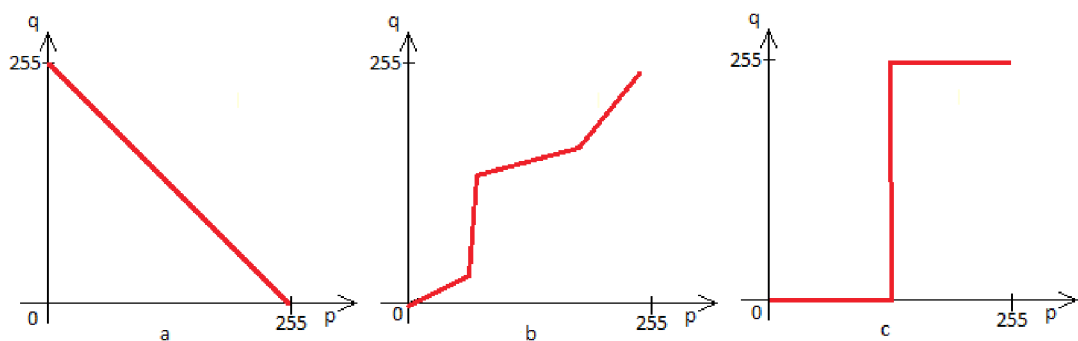
kde M je konstanta určující výsledný kontrast obrazu. A f_c je výsledný kompenzovaný obraz.

4.2 ZMĚNA JASOVÉ STUPNICE

Tato jasová transformace je nezávislá na poloze bodu v obraze. Transformaci T původní stupnice jasu p na novou stupnici q určuje vztah

$$q = T(p) \quad (4)$$

Obvyklými transformacemi jasu jsou: negativ (obr.1a), po částech lineární průběh (obr.1b), prahování (obr.1c).

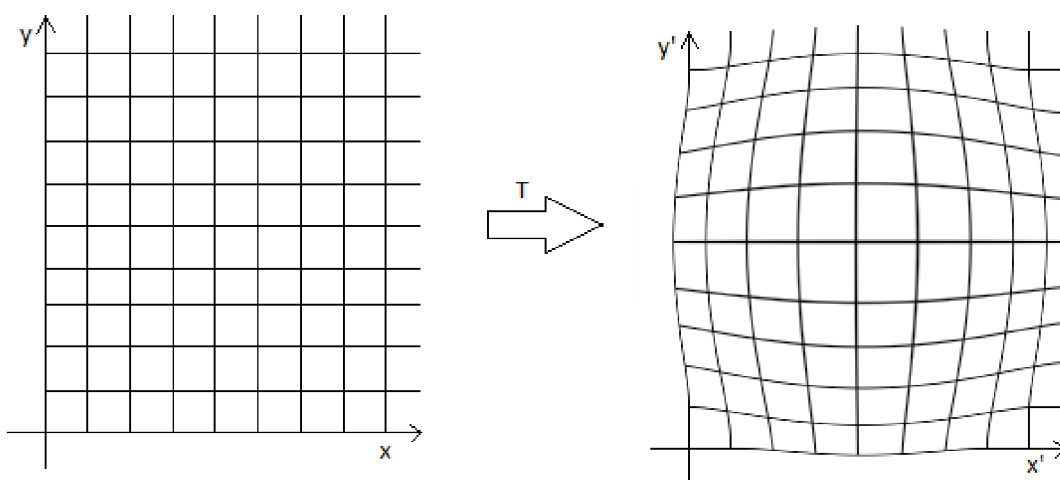


Obrázek 1: transformace jasové stupnice

Transformace jasové stupnice je důležitá především jako operace zvyšující kontrast a tím zjednodušující interpretaci obrazu.

5. GEOMETRICKÉ TRANSFORMACE

V případě, kdy optická osa snímače a snímané plochy svírá jiný, než pravý úhel nebo je snímán jiný než plošný objekt dochází u získaného obrazu ke geometrickým zkreslením. Ke korekci geometrického zkreslení se používají stejné algoritmy jako k geometrickým transformacím. Jedná se o v principu stejné operace.



Obrázek 2: Příklad geometrické transformace

Geometrická transformace T transformuje obrazový bod v rovině (x,y) do bodu (x',y') a je definována vztahy

$$x' = T_x(x, y) \quad , \quad y' = T_y(x, y) \quad (5, 6)$$

V případě rotace, posunu nebo zvětšení obrazu jsou transformační vztahy známy předem nebo je možné hledat na základě znalosti původního a transformovaného obrazu pomocí známých tzv. lícovacích bodů aproximací polynomem:

$$x' = \sum_{r=0}^m \sum_{k=0}^{m-r} a_{rk} x^r y^k \quad , \quad y' = \sum_{r=0}^m \sum_{k=0}^{m-r} b_{rk} x^r y^k \quad (7, 8)$$

Metodou nejmenších čtverců lze určit koeficienty a_{rk} a b_{rk} na základě množiny dvojic bodů ve vstupním a výstupním obraze (x,y) a (x',y') . Po provedení transformace získáme vzorky výstupní obrazové funkce s neceločíselnými souřadnicemi. Poté pomocí některé z interpolací (např. Kubická) známých vzorků v malém okolí bodu získáme hodnoty jasů v potřebných diskrétních bodech.

6. HOUGHOVA TRANSFORMACE

Nejjednodušším případem Houghovy transformace je transformace lineární pro detekci přímek, která přesně odpovídá našim požadavkům detekce řádků textu. V rovině lze přímku popsat rovnicí $y=kx+q$, a zobrazit ji lze jednoduše pomocí souřadnic (x,y) . V Houghově transformaci je hlavní myšlenkou charakterizovat přímku nikoli jako body o souřadnicích x a y , ale pomocí jejích parametrů, směrnice k a posunutí q .

Na základě tohoto faktu může být reprezentována přímka $y=kx+q$ bodem o souřadnicích (q,k) v rovině parametrů. Avšak probléme nastává pokud se jedná o vertikální přímku, kdy dochází k nárůstu parametru k směrem k nekonečnu. Proto je z důvodu snazšího výpočtu lepší parametrizovat přímku pomocí dvou jiných parametrů běžně označovaných jako r a θ . Kde parametr r reprezentuje vzdálenost přímky od počátku souřadnicového systému a θ je úhlem vektoru od počátku k nejbližšímu bodu přímky (viz obr.3). Za předpokladu použití této parametrizace lze rovnici přímky zapsat jako:

$$y = \left(\frac{-\cos\theta}{\sin\theta} \right) x + \left(\frac{r}{\sin\theta} \right) \quad (9)$$

nebo také:

$$r = x\cos\theta + y\sin\theta \quad (10)$$

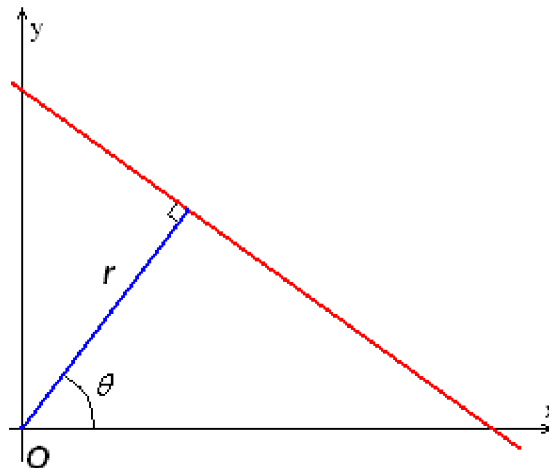
Je tedy možné každé přímce obrazu přiřadit dvojici (r,θ) , která je jedinečná pokud platí $\theta \in \langle 0, \pi \rangle$ a $r \in \mathbb{R}$ nebo pokud $\theta \in \langle 0, 2\pi \rangle$ a $r \geq 0$. Rovina (r,θ) je také někdy označována jako dvourozměrný Houghův prostor.

Jedním bodem roviny může procházet nekonečně mnoho přímek. Pokud má tento bod v rovině souřadnice (x_0, y_0) , pak pro všechny jím procházející přímky platí:

$$r(\theta) = x_0 \cdot \cos\theta + y_0 \cdot \sin\theta \quad (11)$$

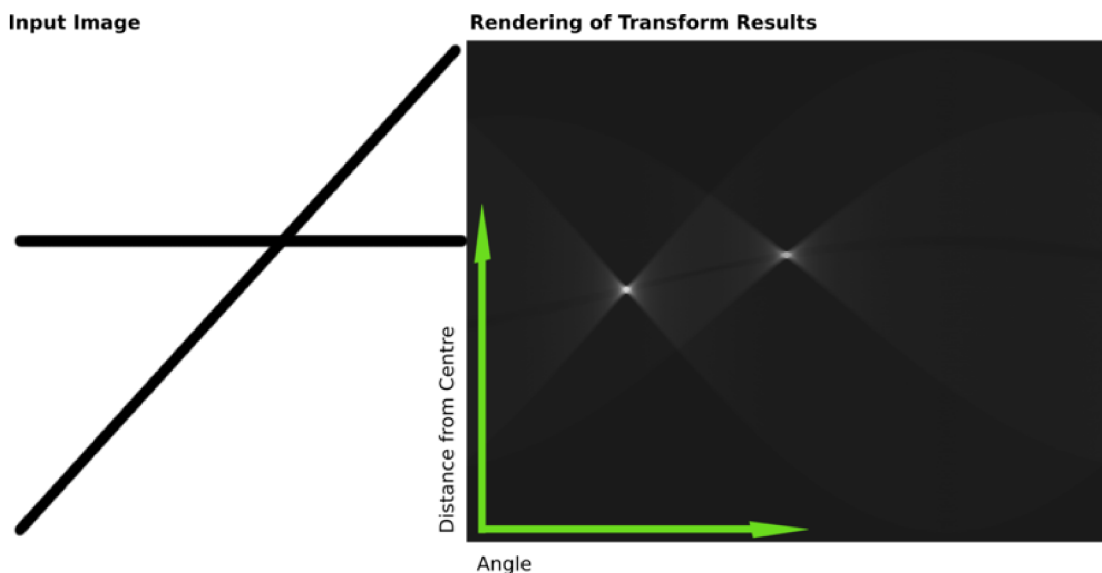
Což odpovídá sinusoidě v rovině (r,θ) , která prochází tímto bodem a je pro něj specifická. Pokud se sinusoidy pro dva body protínají, pak souřadnice bodu

protnutí v Houghově prostoru odpovídají parametrům přímky v obraze, která protíná oba tyto body.



Obrázek 3: znázornění parametrizace

Obecně tedy množina bodů které náleží přímce v obraze vygeneruje v Houghově prostoru sinusoidy na jejichž průsečíku leží právě parametry této přímky. Následující obrázek prezentuje výsledky Houghovy transformace na obraze obsahujícím dvě přímky:



Obrázek 4: výsledek Houghovy transformace[7]

Jak je vidět z pravé části obrázku, dva výrazně světlé body jsou maximy transformace a jejich souřadnice odpovídají parametrům přímek.

7. LOKÁLNÍ PŘEDZPRACOVÁNÍ

Pro výpočet jasů v bodě výstupního obrazu se využívá v tomto případě pouze lokální okolí bodu v obraze vstupním. Lokální předzpracování lze rozdělit do dvou skupin: vyhlazování a gradientní operace.

Vyhlazování obrazu slouží k potlačení šumu (vyšších frekvencí v obraze), čímž ale také dochází k rozostření hran obrazu a ostrých čar, které jsou velmi důležité. Gradientní operace naopak vyšší frekvence zdůrazňuje. Dochází k zostření obrazu a zvýraznění hran. To ale také vede k většímu zašumění.

Podle vlastností funkčního vztahu pro výpočet jasů je také možné rozdělit metody předzpracování na lineární a nelineární. Lineární operace počítají výsledný jas v bodě pomocí jasů okolí O o velikosti $M \times N$ (kvůli symetrii nejčastěji rozměry náleží do množiny lichých přirozených čísel) z vstupního obrazu g a s váhovými koeficienty h . Pro systémy nezávislé na poloze platí diskrétní konvoluce

$$f(i, j) = \sum_{m=j-\frac{M}{2}}^{j+\frac{M}{2}} \sum_{n=j-\frac{N}{2}}^{j+\frac{N}{2}} h(m-i, n-j) g(m, n) \quad (12)$$

Volba transformace, velikosti a tvaru okolí bodu záleží na rozměrech zpracovávaných objektů v obraze. Vlastnosti obrazu lze u velkých objektů zlepšit vyhlazením malých nespojitostí funkce a naopak.

7.1 FILTRACE-VYHLAZOVÁNÍ OBRAZU

Jedním z nejčastějších úkolů řešených pomocí filtrace je vyhlazení šumu v obraze. Pro vysvětlení principu filtrování lze využít jeho nejjednodušší formu – průměrování. Tento filtr funguje na principu přiřazení průměru jasů v obdélníkovém okolí výslednému bodu. Jeho nevýhodou je rozmazávání hran v obraze.

Rozložení hodnot šumu bude mít nulovou střední hodnotu μ , směrodatnou odchylku σ a náhodné veličiny popisující šum v jednotlivých bodech budou nezávislé. Body okolí O mají hodnotu jasů bez šumu f_1, f_2, \dots, f_n a šum je popsán

náhodnými veličinami v_1, v_2, \dots, v_n . Hodnotu v bodě ovlivněném šumem pak vyjádříme pomocí aritmetického průměru

$$\frac{g_1 + \dots + g_n}{n} = \frac{v_1 + \dots + v_n}{n} \quad (13)$$

Kde $v_1 + \dots + v_n$ představuje náhodnou veličinu s nulovou střední hodnotou a směrodatnou odchylkou σ/\sqrt{n} . Amplituda šumu se zmenšila činitelem \sqrt{n} . Pokud máme m stejných obrazů, pouze s rozdílným šumem, pak lze využít k filtraci následující vztah který nerozmazává hrany

$$f(i, j) = \frac{1}{m} \sum_{k=1}^m g_k(i, j) \quad (14)$$

Pokud je k dispozici pouze jeden zašuměný obraz, využije se lokálního průměrování. Jedná se případ diskrétní konvoluce (12). Pro okolí o rozměru 3x3 je pak konvoluční maska

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (15)$$

Nebo pro zvětšení váhy středu resp. 4 okolí (pro lepší aproximaci šumu s Gaussovským rozložením)

$$h = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (16), (17)$$

Pro omezení nežádoucího efektu rozostřování hran je možné využít metodu průměrování s omezením změn jasu, která brání rozmazávání hran požitím intervalu diference mezi jasem původním a jasem filtrovaného bodu.

7.1.1 Filtr s Gaussovým rozložením

Využívají se dva typy tohoto filtru – jednorozměrný (v poli) a dvourozměrný (v ploše). Jednorozměrný filtr využívá vztahu

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (18)$$

Zatímco ve dvou rozměrech je výsledkem součin dvou těchto funkcí po sobě následujících

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (19)$$

kde x udává vzdálenost od počátku ve vodorovné ose, y udává vzdálenost ve svislé ose a σ je směrodatná odchylka udávající velikost okolí ve kterém filtr pracuje.

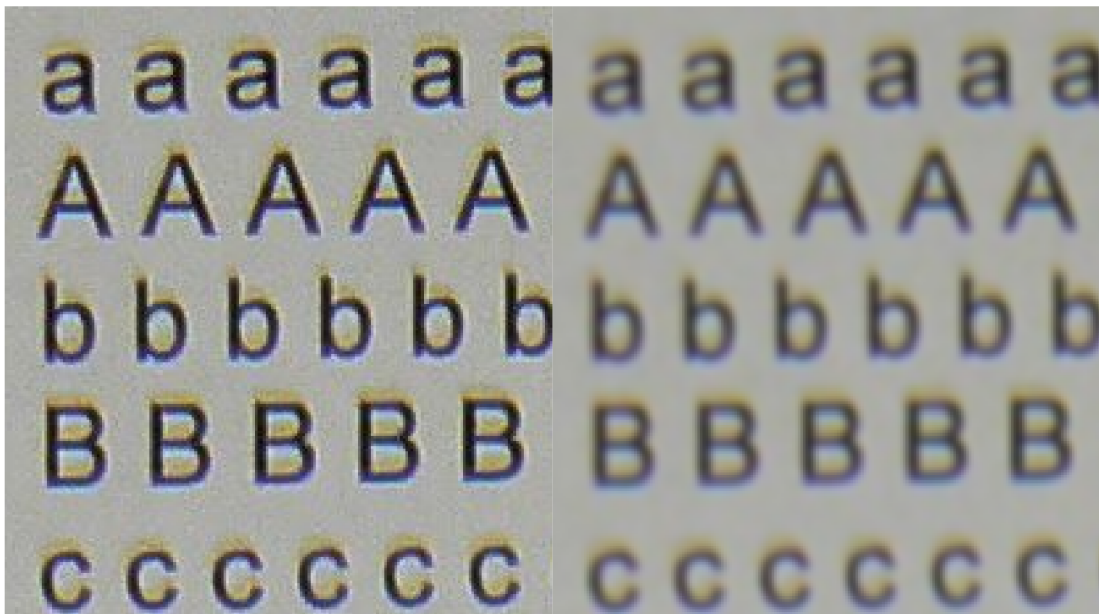
Pokud je použita funkce pro dva rozměry výsledkem je plocha s Gaussovským rozložením hodnot směrem ze středu. Tyto hodnoty jsou potom využity pro vytvoření konvoluční matice, která je aplikována na originální obraz. Hodnota jasu každého pixelu je pak určena jako vážený průměr bodů v okolí O . Nejvyšší hodnota váhy je přidělena bodu na původní pozici (vrchol Gaussovy plochy) a ostatní body mají postupně nižší váhy s jejich rostoucí vzdáleností od středu.

Příklad masky o rozměru 5x5 vypočítané pro $\sigma=1.4$

$$h = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Teoreticky má Gaussova funkce nenulové hodnoty ve všech bodech obrazu, což by znamenalo zahrnout do výpočtů celý obraz. V praxi se však při výpočtu diskrétní aproximace Gaussovy funkce považují hodnoty vzdálenější než 3σ za nulové a tedy lze vzdálenější body z výpočtu vyloučit. Typický rozměr matice potřebný pro výpočet je $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$ (kde $\lceil \cdot \rceil$ je operátorem zaokrouhlení směrem nahoru - angl. „ceiling”) aby byl výsledek dostatečně přesný.

Vzhledem k tomu, že vypočítaná matice je kruhově symetrická, lze dosáhnout stejného výsledku použitím dvourozměrného filtru jako použitím filtru jednorozměrného aplikovaného dvakrát nejprve v horizontálním a pak vertikálním směru (nebo opačně). Z této vlastnosti plyne možnost výpočet urychlit aplikací dvou jednorozměrných filtrů místo jednoho dvourozměrného.

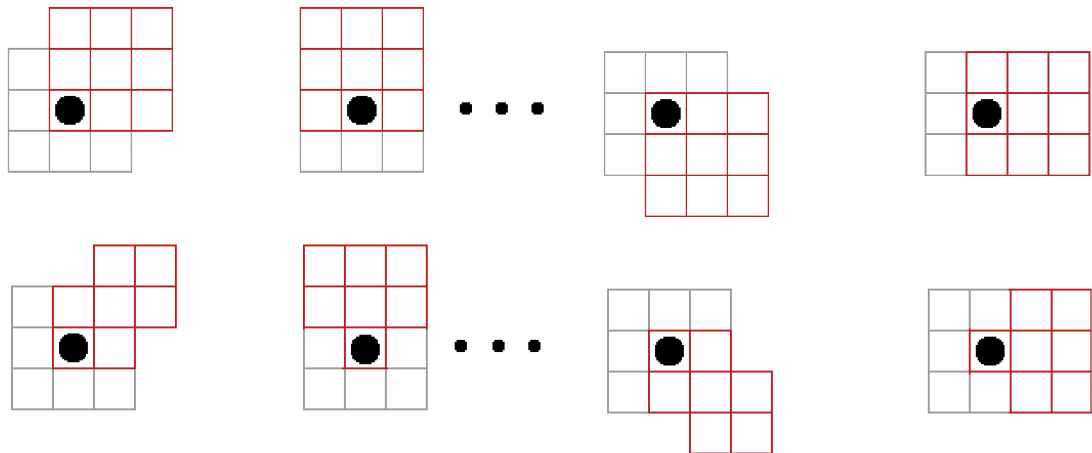


Obrázek 5: obraz před aplikováním filtru Obrázek 6: obraz po aplikaci Gaussova filtru s rozměrem 5x5

7.2 DALŠÍ FILTRY

Filtrace pomocí mediánu je dalším z využívaných filtrů. Je výpočetně poměrně jednoduchý. Stačí vzestupně uspořádat hodnoty jasu v okolí bodu a určit výsledek jako prvek ležící uprostřed této posloupnosti. Tato metoda redukuje stupeň rozmazání hran a potlačuje impulzní šum a lze ji použít iterativně. Její nevýhodou je porušování tenkých čar a ostrých rohů. Tuto vlastnost lze redukovat požitím jiného než obdélníkového okolí (např. křížového).

Vyhlazování metodou rotující masky se snaží najít nejpravděpodobnější část okolí bodu ke které patří podle homogenity jasu. K výpočtu výsledné hodnoty jasu jsou použity pouze body z tohoto homogenního okolí neobsahující hrany. Díky tomu tento filtr nerozmazává hrany a má mírně ostřící charakter. Lze jej také požit iterativně, rychle konverguje do stabilního stavu. Rychlost konvergence ovlivňuje především tvar a velikost použité masky. Čím menší je použitá maska, tím více je potřeba iterací.



Obrázek 7: Filtry rotující masky

7.3 DETEKCE HRAN, OSTŘENÍ

Hranou jsou body obrazové funkce ve kterých dochází k výrazným změnám jasu. Je to vektorová veličina určena velikostí a směrem, která vychází z gradientu ∇ obrazové funkce. Velikost gradientu ∇ určuje vztah (20). Získání hran je operace takřka nezávislá na vlastnostech objektů kdežto spojování hran do hranic je operace na vlastnostech závislá.

$$|\nabla g(x, y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \quad (20)$$

A jeho směr

$$\Psi = \tan^{-1} \left(\frac{\frac{\partial g}{\partial y}}{\frac{\partial g}{\partial x}} \right) \quad (21)$$

Při zpracování digitalizovaného obrazu jsou parciální derivace nahrazeny diferencemi $\Delta_x g(i, j)$ a $\Delta_y g(i, j)$

$$\Delta_x g(i, j) = g(i, j) - g(i-1, j) \quad (22)$$

$$\Delta_y g(i, j) = g(i, j) - g(i, j-1) \quad (23)$$

Ostření obrazu odpovídá zdůraznění vyšších frekvencí tím dojde k zvětšení strmosti hran, ale také zdůraznění šumu. Vztah mezi obrazem g a výsledkem zostření je

$$f(i, j) = g(i, j) - C S(i, j) \quad (24)$$

kde C je koeficient udávající sílu ostření a $S(i, j)$ je operátor udávající strmost změny obrazové funkce v bodě (např. gradient).

7.3.1 Typy operátorů

První skupinou operátorů jsou operátory aproximující derivace funkce pomocí diferencí. Operátory odhadující první derivaci používají několik masek. Směr gradientu se odhadne nalezením masky, která odpovídá největší velikosti gradientu. Tyto operátory se realizují se jednou konvoluční maskou. Do této skupiny patří operátor Robertsův jehož velikost se počítá

$$|\nabla g(i, j)| = |g(i, j) - g(i+1, j+1)| + |g(i, j+1) - g(i+1, j)| \quad (25)$$

a jeho konvoluční masky pak

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (26, 27)$$

dále Sobelův operátor, který je směrově závislý a jeho konvoluční masky pro rozměr 3x3 jsou

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad (28, 29)$$

Masky pro další směry se získají pootočením.

Laplaceův operátor ∇^2 který je určen vztahem (30) je necitlivý vůči otočení, aproximuje druhou derivaci a udává pouze velikost hrany.

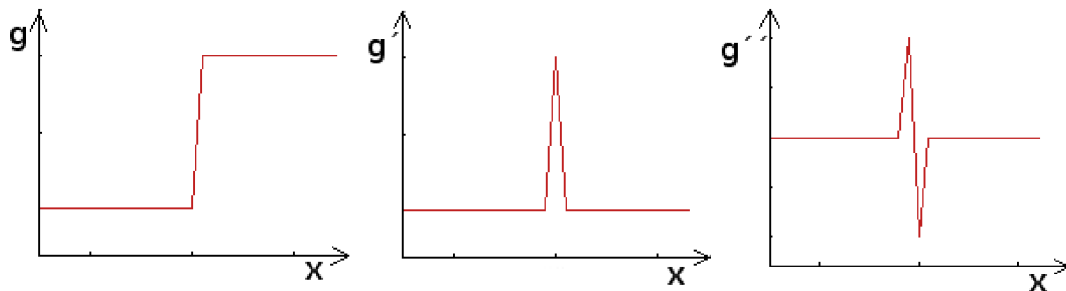
$$\nabla^2(x, y) = \frac{\partial^2 g(x, y)}{\partial x^2} + \frac{\partial^2 g(x, y)}{\partial y^2} \quad (30)$$

Jeho konvoluční masky pro 4-sousedství, pro 8-sousedství o rozměru 3x3

$$h_{4-soused} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h_{8-soused} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (31, 32)$$

Dalšími operátory jsou Robinsonův, Kirschův a Prewittové. Bez statistických znalostí obrazu nelze předem určit nejvhodnější operátor a velmi často se proto

vybírám pokusem. Hlavní nevýhodou této skupiny operátorů je velká závislost na šumu a velikosti objektů v obraze.



Obrázek 8: ilustrace průchodu nulou

Druhou skupinou operátorů jsou operátory detekující průchod druhé derivace nulou. Ty využívají toho, že první derivace v místě hrany dosahuje svého maxima a druhá derivace v tomto místě protíná nulovou hodnotu. Do této skupiny filtrů patří Cannyho hranový detektor a operátor Marra a Hildretové.

Druhou derivaci obrazové funkce lze snadno najít pomocí pomoci filtru s Gaussovým rozložením a pomocí Laplaceova operátoru podle vztahu (33). Ten využívá linearitu použitých funkcí k zaměnění pořadí vyhlazování pomocí Gaussova filtru G pomocí konvoluce a druhé derivace.

$$\nabla^2(G * g) = (\nabla^2 G) * g = 0 \quad (33)$$

7.3.2 Cannyho hranový detektor

Tento hranový detektor je využívá detekční algoritmus optimální vzhledem ke třem kritériím:

- správná detekce – algoritmus by měl v obraze označit tolik hran kolik jich skutečně existuje v předloze
- správná lokalizace – vyznačené hrany by měly být co nejbližší skutečným hranám
- minimální odezva – každá hrana by měla být označena pouze jednou a tam, kde je to možné by šum neměl vytvářet falešné hrany

K dosažení těchto požadavků se využívá variační počet, který nalezne funkci optimalizující určený funkcionál (funkcionál je speciálním operátorem zobrazující z prostoru funkcí do množiny obecně komplexních čísel). Optimální funkce je v Cannyho detektoru popsána součtem čtyř exponenciálních výrazů, které mohou být aproximovány první derivací Gaussovy funkce. Detekce probíhá v následujících krocích:

- Eliminace šumu
- Určení gradientu
- Nalezení lokálním maxim
- Eliminace nevýznamných hran

Eliminace šumu

Eliminace šumu se provádí pomocí konvoluce obrazu s konvoluční maskou vypočítanou Gaussovým filtrem (19). Výsledkem je mírně rozostřený obraz, který není nijak významně ovlivněn šumem (viz obr. 6).

Určení gradientu

Hrana v obraze může být orientována libovolným směrem. Cannyho algoritmus využívá čtyř filtrů pro detekci horizontálních, vertikálních a diagonálních hran. K výpočtu gradientu je vhodné použít například Sobelův operátor, který je necitlivý vůči šumu a jehož výsledkem je velikost gradientu a jeho úhel důležitý v dalších krocích. Tento úhel je zaokrouhlen na jeden ze čtyř úhlů z nichž každý reprezentuje jeden ze směrů (0° , 45° , 90° a 135°).

Nalezení lokálních maxim

Úkolem této operace je vybrat z hodnot gradientů stanovených v předchozím kroku pouze lokální maxima, respektive odebrat body, které maximem nejsou. Tím zajistíme, že hrana bude detekována v místě největšího gradientu. Prakticky to znamená najít obrazové body jejichž hodnota gradientu okolí je ve směru a proti směru gradientu vybraného bodu nižší. Má-li být například bod součástí hrany ve směru 45° musí být jeho gradient vyšší, než gradienty bodů ve směru 135° . Tento

výpočet se provádí tak, že je vybrána vždy matice 3x3 obrazových bodů a v každém z těchto bodů je provedeno porovnání. Po provedení tohoto kroku je výsledkem binární obraz obsahující body hran označené hodnotou 1 a pozadí označené hodnotou 0.

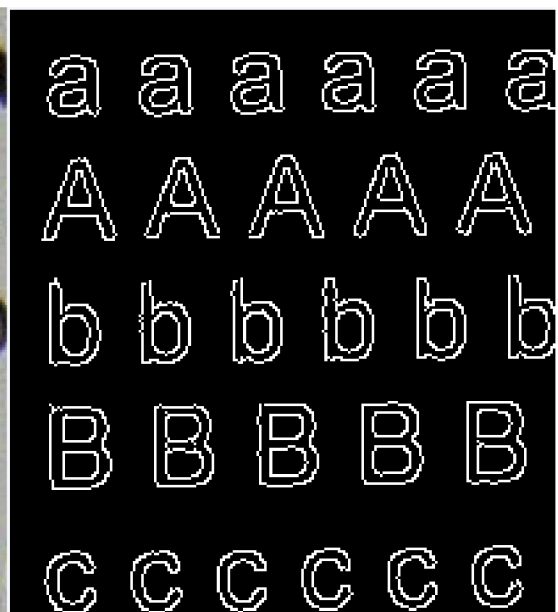
Eliminace nevýznamných hran

V předchozím kroku byla určena poloha hran. Označeny ale byly i ty se zanedbatelnou velikostí. Velké hodnoty gradientů odpovídají hranám častěji než hodnoty malé. Většinou je ale nemožné určit přesně hodnotu, která by odpovídala prahu, který by vždy správně určil zda daný bod je pro hranu relevantní, či nikoliv například kvůli tomu, že velikost gradientu může být ovlivněna šumem. Proto se v tomto kroku se aplikuje prahování s hysterezí.

Zvolíme tedy dolní práh P_d a horní práh P_h mezi kterými se gradient může pohybovat. Pokud hodnota gradientu v bodě je vyšší než P_h je tento bod označen jako hranový. Jestliže testovaný bod leží v rozmezí $P_d < |\nabla g(x, y)| < P_h$ pak je označen pouze v případě, že sousedí s bodem který byl jako hrana označen dříve. Výhodné je začít u bodů s vysokými hodnotami gradientů, tím na počátku získáme hlavní hrany v obraze a dojde k odfiltrování šumu (bodů s velkými gradienty, které nejsou součástí hran).



Obrázek 9: původní obraz



Obrázek 10: výsledek po aplikaci
Cannyho hranového detektoru

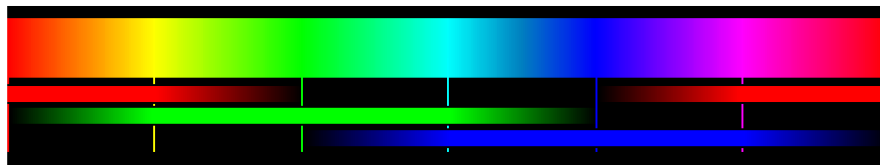
8. BARVY V DIGITÁLNÍM OBRAZE

Barvy v digitálním obraze jsou reprezentovány pomocí několika barevných modelů, které nahrazují spektrum světla viditelné pro lidské oko (vlnové délky 380nm-780nm).

8.1 BAREVNÉ MODELY

8.1.1 Model RGB, RGBA

Jednotlivé složky nahrazují barvy červená-Red (630nm), zelená-Green (530nm) a modrá-Blue (450nm). Čtvrtou složkou u modelu RGBA je tzv. alfa kanál udávající průhlednost obrazového bodu a tím přidávající informaci o délce jednoho bajtu. Alfa kanál může být interpretován dvěma způsoby: Buďto udává že obraz je pokryt průhlednou barvou RGB s průhledností A, nebo je z A procent pokryt neprůhlednou barvou RGB.



Obrázek 11: Spojité barevné spektrum a jeho reprezentace pomocí modelu RGB [8]

8.1.2 Model CMY, CMYK

Použité barvy jsou Cyan-azurová, Magenta-purpurová, Yellow-žlutá a black-černá. Černá barva byla přidána z důvodu neúplného krytí barev CMY. Model je založen na odečítání barev a omezování barevného spektra (subtraktivní míchání barev). Využívá se např. u inkoustových tiskáren.

8.1.3 Model HSV

Tento barevný model nejvíce odpovídá lidskému vnímání barev. Jeho tři složky jsou: Hue – barevný tón neboli odstín, Saturation - sytost barvy představující množství šedi v poměru k odstínu a Value – hodnota jasu, množství bílého světla,

relativní světlost nebo tmavost barvy. Model HSV se využívá v grafických programech, kde bývá zobrazován jako kruh a trojúhelník, neúplný kužel nebo šestistanný jehlan.

8.1.4 Model HSL

Barevný model HSL odpovídá skutečnosti, že nejvíce barev člověk vnímá při poloviční světlosti. Jeho tři složky jsou Hue, Saturation a Lightness – světlost. Pro některé lidi je tento model přirozenější reprezentací než model HSV. Výhodou tohoto modelu je jeho symetričnost vzhledem k světlosti a tmavosti barvy.

8.2 TRANSFORMACE BAREV

Transformace barev se využívá z důvodu rozdílné interpretace barev různých zařízení (např. při tisku na inkoustové tiskárně dochází k barevnému převodu formátu RGB na formát CMYK) nebo kvůli nadbytečnosti barevné informace při zpracování obrazu (odstraněním barevné informace dojde k značnému urychlení operace).

Z hlediska povahy zadání (segmentace fotografie textu na jednotlivé znaky) je barevná informace zcela nadbytečná a získaný obraz je proto vhodné převést nejprve na šedotónový a poté na binární pro co nejjednodušší zpracování.

Převod z obrazu barevného na šedotónový je možné provést několika způsoby:

- přiřadit každé složce barevného obrazu stejnou váhu. Pak pro výslednou intenzitu bodu bude platit

$$I_v = 0,33 I_R + 0,33 I_G + 0,33 I_B \quad (34)$$

- pracovat pouze s převládající složkou obrazu
- upravit váhy podle vnímání barevných složek obrazu člověkem

$$I_v = 0,299 I_R + 0,587 I_G + 0,114 I_B \quad (35)$$

Dalším krokem je převod šedotónového obrazu na binární. Tento převod se obvykle provádí pomocí prahování. Prahování s prahem T je transformace vstupního obrazu f na výstupní binární obraz podle vztahu

$$g(i, j) = \begin{cases} 1 & \text{pro } f(i, j) \geq T \\ 0 & \text{pro } f(i, j) < T \end{cases} \quad (36)$$

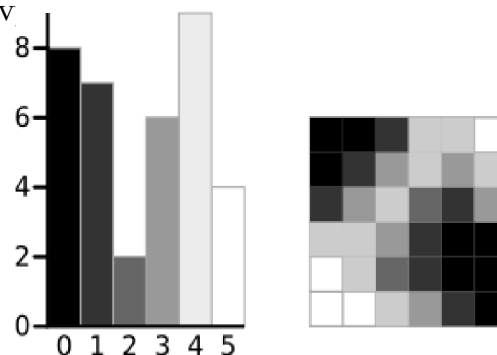
Správná volba prahu T je klíčová pro úspěšný výsledek prahování. Hodnotu prahu lze určovat pomocí některé z metod automatického určování prahu. Nejjednodušší je určení prahu v případě, kdy je známa vlastnost binárního obrazu. Práh je pak určen tak aby tato vlastnost byla splněna. Příkladem je prahování obrazu s textem, který pokrývá $1/p$ plochy stránky. S využitím znalosti poměru ploch objektů a pozadí je pak možné za pomoci histogramu hodnotu prahu T takovou aby právě $1/p$ plochy mělo úroveň jasu menší než T . Tuto informaci však nemáme k dispozici a proto je třeba využít některou z dalších složitějších metod analýzy histogramu.

8.2.1 Otsuho prahovací metoda (Otsu thresholding)

Otsuho prahovací metoda spočívá v iteraci přes všechny možné hodnoty prahu T a výpočtu míry rozptylu pro všechny jasové úrovně na obou stranách prahovací úrovně. Cílem je najít takovou hodnotu prahu T pro které bude součet rozptylů pro pozadí a pro objekty minimální. Funkce algoritmu předvedeme na následujícím obraze o rozměru 6×6 bodů s šesti úrovněmi jasu (pro zjednodušení).

Nyní provedeme v

objektů pro jednu



Obrázek 12: napravo obraz v odstínech šedi a nalevo jeho histogram [9]

Pozadí:

$$\text{váha: } W_b = \frac{8+7+2}{36} = 0,4722$$

$$\text{vážený průměr: } \mu_b = \frac{(0 \cdot 8) + (1 \cdot 7) + (2 \cdot 2)}{17} = 0,6471$$

$$\text{rozptyl: } \sigma_b^2 = \frac{((0-0,6471)^2 \cdot 8) + ((1-0,6471)^2 \cdot 7) + ((2-0,6471)^2 \cdot 2)}{17} = 0,4637$$

Objekty:

$$\text{váha: } W_f = \frac{6+9+4}{36} = 0,5278$$

$$\text{vážený průměr: } \mu_f = \frac{(3 \cdot 6) + (4 \cdot 9) + (5 \cdot 4)}{19} = 3,8947$$

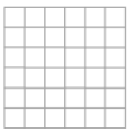
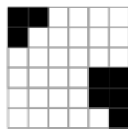
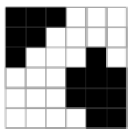
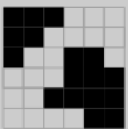
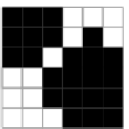
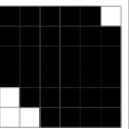
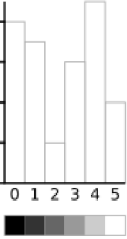
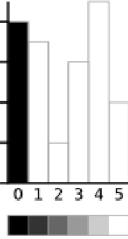
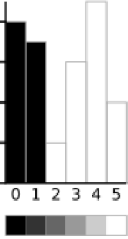
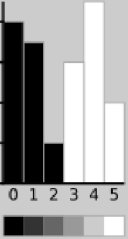
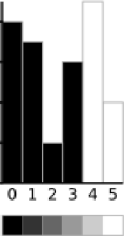
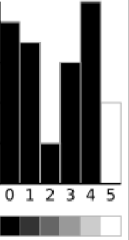
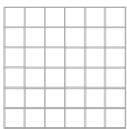
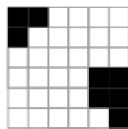
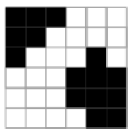
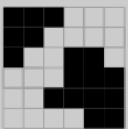
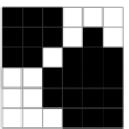
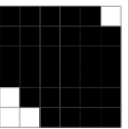
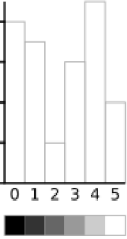
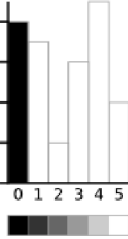
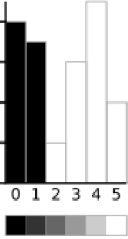
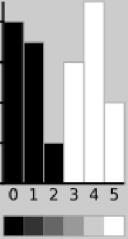
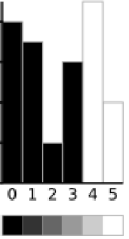
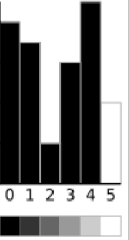
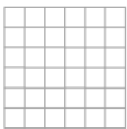
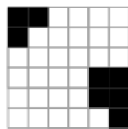
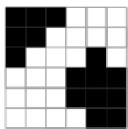
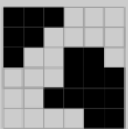
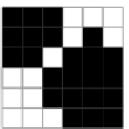
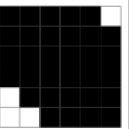
rozptyl:

$$\sigma_f^2 = \frac{((3-3,8947)^2 \cdot 6) + ((4-3,8947)^2 \cdot 9) + ((5-3,8947)^2 \cdot 4)}{19} = 0,5152$$

Dalším krokem je výpočet tzv. vnitřního rozptylu což je součet obou rozptylů vynásobených příslušnými vahami:

$$\sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2 = 0,4722 \cdot 0,4637 + 0,5278 \cdot 0,5152 = 0,4909$$

Tento výpočet je třeba provést pro všechny další hodnoty prahu 0 až 5. Následující tabulka ukazuje vypočítané výsledky a výsledný obraz po vyprahování.

T	0	1	2	3	4	5
W_b	0,000	0,000	0,417	0,472	0,639	0,889
μ_b	0,000	0,000	0,467	0,647	1,261	2,031
σ_b^2	0,000	0,000	0,249	0,464	1,410	2,530
W_f	1,000	0,778	0,583	0,528	0,361	0,111
μ_f	2,361	3,036	2,714	3,895	4,308	5,000
σ_f^2	3,120	1,964	0,776	0,515	0,213	0,000
σ_w^2	3,120	1,527	0,556	0,491	0,978	2,249
[10] [11]						
[12]						
[13]						
[14]						
[15]						

Tabulka 1: Výsledky výpočtů pro Otsuho prahovací metodu

Z tabulky je zřejmé, že použitý příklad výpočtu je také výsledkem, neboť má nejnižší součet vážených rozptylů. Tento postup výpočtu je sice vhodný k vysvětlení teorie avšak je výpočetně dosti náročný obzvláště pak pokud pracujeme s šedotónovým obrazem v rozsahu 8 bitů.

Pomocí jednoduché úpravy je možné vypočítat tzv. vnější rozptyl, který je výpočetně méně náročný. Práh jehož hodnota vnějšího rozptylu je maximální má také minimální hodnotu vnitřního rozptylu, což lze s výhodou využít k značnému zjednodušení výpočtu.

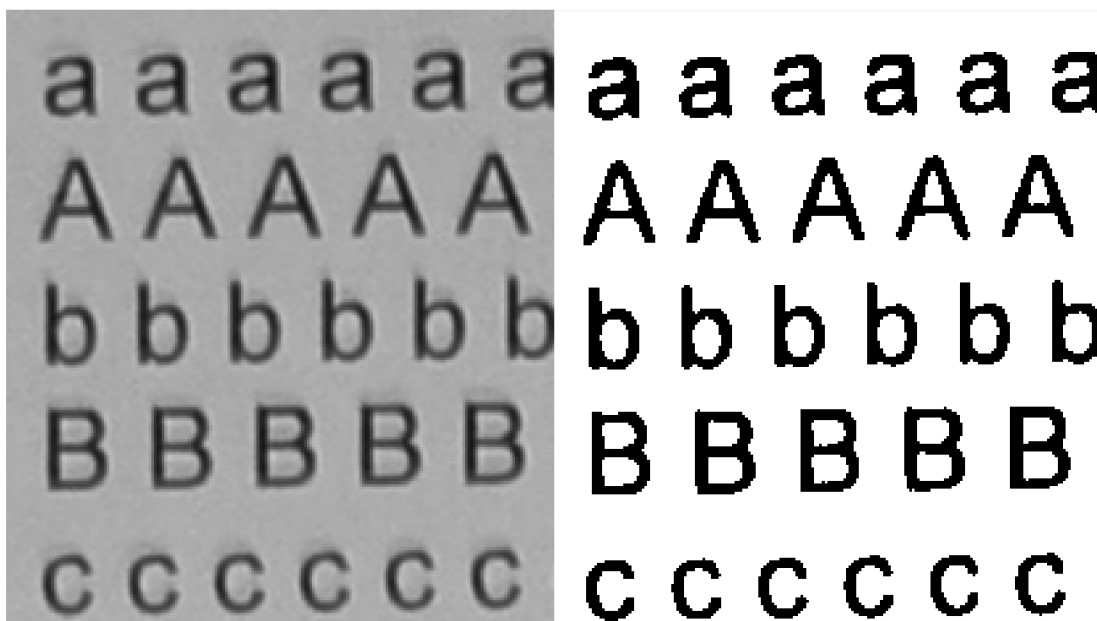
$$\sigma_B^2 = \sigma^2 - \sigma_w^2 = W_b(\mu_b - \mu)^2 + W_f(\mu_f - \mu)^2 = W_b W_f (\mu_b - \mu_f)^2 \quad (37)$$

$$\text{Kde } \mu = W_b \mu_b + W_f \mu_f$$

Následující tabulka srovnává hodnoty vnitřních a vnějších rozptylů pro jednotlivé hodnoty prahu.

T	0	1	2	3	4	5
σ^2_W	3,1196	1,5268	0,5561	0,4909	0,9779	2,2491
σ^2_B	0,0000	1,5928	2,5635	2,6287	2,1417	0,8705

Tabulka 2: srovnání výsledků výpočtu vnitřního a vnějšího rozptylu



Obrázek 14: šedotónový obraz před aplikací prahování

Obrázek 15: obraz po aplikaci prahování

9. STATISTICKÉ MOMENTY

V této práci jsou využity statistické momenty ke klasifikaci jednotlivých vysegmentovaných znaků. Pro každý znak je vypočteno sedm momentů (Hu invariantní množina), které slouží jako vstupní data pro klasifikátor v podobě neuronové sítě.

Momenty lze využít v mnoha aplikacích zpracování obrazu--rozpoznávání vzorů, kódování obrazu, odhad polohy atd. Popisují obsah obrazu (jeho rozložení) vzhledem k jeho osám. Jsou utvořeny tak, aby obsahovaly jak globální tak detailní geometrické informace o obrazu. V této práci jsou využity vzhledem k těmto faktům využity pro charakterizování jednotlivých znaků a tím získání vhodných vstupních dat pro neuronovou síť. Jak bylo řečeno v předchozích kapitolách obraz lze popsat spojitou funkcí dvou proměnných $f(x,y)$. Za tohoto předpokladu lze základní vztah pro moment o řádu $p+q$, uvažovaný přes celou rovinu obrazu ξ vyjádřit jako:

$$M_{pq} = \iint_{\xi} \psi_{pq}(x, y) f(x, y) dx dy \quad p, q = 0, 1, 2, \dots, \infty \quad (38)$$

Kde ψ_{pq} je základní funkce. Základní funkce může mít mnoho potřebných vlastností, které lze přenést do momentu a tím poskytnout popis invariantní vůči posunutí, rotaci či změně měřítka. Aby bylo možné uvedený vztah aplikovat na digitalizovaný obraz je třeba, aby byl tento vztah (38) převeden do diskrétní formy. Funkce hustoty pravděpodobnosti spojitého rozložení je odlišná od funkce diskrétního rozložení. Dále budeme předpokládat že obraz je rozložen do čtvercových bodů o rozměru 1×1 s konstantní intenzitou I pro všechny tyto body. Pak je hodnota jasu P_{xy} v tomto bodě:

$$P_{xy} = I(x, y) \Delta A \quad (39)$$

kde ΔA je plocha jednoho pixelu, která je rovna jedné. Pak tedy pokud uvažujeme moment přes celou rovinu diskretizovaného obrazu lze psát:

$$M_{pq} = \sum_x \sum_y \psi_{pq}(x, y) P_{xy} \quad p, q = 0, 1, 2, \dots, \infty \quad (40)$$

Volba základní funkce závisí na potřebách aplikace a potřebných neměnných vlastnostech. Je možno využít omezení souřadnic x a y nebo transformaci do polárních souřadnic (např. mapování na jednotkovou kružnici).

9.1 FUNKCE GENERUJÍCÍ MOMENT

Pro výpočet statistických momentů je nejprve třeba určit vhodnou funkci pro jejich generování a tu následně převést do diskrétní formy vhodné pro algoritmicizaci a využití v programovacím jazyce. Prvním krokem je tedy nalezení této funkce. K popisu rozložení pravděpodobnosti náhodné veličiny lze využít charakteristickou funkci:

$$X(\omega) = \int_{-\infty}^{\infty} f(x) e^{j\omega x} dx = E[e^{j\omega x}] \quad (41)$$

kde $f(x)$ je hustota pravděpodobnosti, E je střední hodnota, $j = \sqrt{-1}$ a ω je úhlová frekvence. Tento vztah je v podstatě Fourierovou transformací signálu a dosahuje maxima pro $\omega = 0$ pro všechna $f(x) \geq 0$. Pokud $f(x)$ je spojitá hustota pravděpodobnosti reálného kladného čísla x ($x \in R$) je možné definovat exponenciální rozložení. Nahrazením $j\omega$ ve vztahu (41) operátorem s vznikne integrální vztah:

$$M^x(s) = \int_{-\infty}^{\infty} f(x) e^{xs} dx = E[e^{xs}] \quad (42)$$

$M^x(s)$ se nazývá momentovou tvořící funkcí, zde pro jednorozměrné rozložení veličiny x . Tato funkce se využívá pro charakterizaci rozložení ergodického signálu (signál, jehož střední hodnota a rozptyl mohou být určeny z jednoho dostatečně dlouhého pozorování). Rozvojem exponenciálu do Taylorovy řady dostaneme:

$$e^{xs} = \sum_{n=0}^{\infty} \frac{x^n s^n}{n!} = 1 + xs + \frac{1}{2!} x^2 s^2 + \dots + R_n(x) \quad (43)$$

kde $R_n(x)$ udává chybu. Z uvedeného vztahu je vidět, že tato řada bude reprezentovat $x(s)$ v plné míře pouze pokud $R_n(x)=0$. Tedy pokud je rozložení hustoty pravděpodobnosti konečné (omezené velikostí) pak všechny hodnoty ležící mimo jeho rozměr musí být nulové. Z toho pro problematiku zpracování obrazu plyne, že všechny hodnoty jasu ležící mimo rovinu obrazu musí nabývat nulové hodnoty. Shrnutí tohoto faktu a dosazením vztahu (43) do vztahu (42) dostáváme:

$$M^x = \int_{-\infty}^{\infty} f(x) e^{xs} dx = \int_{-\infty}^{\infty} \left(1 + xs + \frac{1}{2!} x^2 s^2 + \dots \right) f(x) dx = 1 + sm_1 + \frac{1}{2!} s^2 m_2 + \dots \quad (44)$$

kde m_n je n -tý moment. Zderivováním vztahu (42) n -krát podle s dostaneme:

$$M_n^x(s) = E[x^n e^{xs}] \quad (45)$$

Pokud je $M^x(s)$ derivovatelný v nule pak moment n -tého řádu je určen vztahem:

$$M_n^x(0) = E[x^n] = m_n \quad (46)$$

Pokud se jedná o Gaussovo rozložení, pak jej lze kompletně popsat pomocí dvou momentů. Jsou jimi střední hodnota ($M_1^x(0)$) a rozptyl ($M_2^x(0) - (M_1^x(0))^2$) a celková plocha ($M_0^x(0)$) je 1. Pokud je potřeba spojený moment $M^{xy}(s)$ pro dva signály (obraz--dvourozměrná jasová funkce) pak ho lze určit jako:

$$M^{xy}(s) = E[e^{(x+y)s}] = E[e^{xs} e^{ys}] \quad (47)$$

A v případě, že x a y jsou nezávislé pak:

$$M^{xy}(s) = E[e^{xs}] E[e^{ys}] = M^x(s) M^y(s) \quad (48)$$

9.2 NEORTOGONÁLNÍ MOMENTY

Podle [16] je možno spojitý dvourozměrný Kartézský moment řádu $p+q$ vyjádřit pomocí Riemannova integrálů jako:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (49)$$

Funkci $f(x,y)$ uvažujeme jako po částech spojitou ohraničenou funkci, jejíž hodnoty nabývají nenulových velikostí pouze v omezeném prostoru ohraničeném rovinou $x-y$ (tedy všechny hodnoty ležící mimo rovinu obrazu jsou nulové, viz předchozí podkapitola). Pokud je toto tvrzení pravdivé, pak existují momenty všech řádů a platí následující tvrzení:

Teorém jedinečnosti: řada po sobě jsoucích momentů m_{pq} (49) je jedinečně definována funkcí $f(x,y)$ a opačně funkce $f(x,y)$ je jedinečně definována pomocí momentů m_{pq} .

Což znamená, že původní obraz lze popsat a rekonstruovat pomocí momentů dostatečně vysokého řádu. Upravením vztahu (47) pro dvě veličiny může být Kartézský moment (49) vyjádřen jako momentová tvořící funkce:

$$M^{xy}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{ux+vy} f(x, y) dx dy \quad (50)$$

Rozvojem do Taylorovy řady pak vznikne vztah:

$$M^{xy}(u, v) = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \frac{u^p v^q}{p! q!} m_{pq} \quad (51)$$

9.3 KARTÉZSKÉ MOMENTY

Diskrétní verze Kartézského momentu (49) pro obraz skládající se z obrazových bodů P_{xy} nahrazením integrálů sumacemi je:

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p y^q P_{xy} \quad (52)$$

kde m_{pq} je dvourozměrný Kartézský moment, M a N jsou rozměry obrazu a výraz

$x^p y^q$ je základní funkce (viz vztah 38).

Moment nultého řádu m_{00} je definován jako celková plocha obrazu. Pokud je vypočten z binárního obrazu o rozměrech $M \times N$ pak se stává počtem obrazových bodů, ze kterých je složen objekt na obraze.

$$m_{00} = \sum_{x=1}^M \sum_{y=1}^N P_{xy} \quad (53)$$

Dva momenty prvního řádu lze použít k nalezení těžiště obrazu. Pokud je výpočet proveden z binárního obrazu a výsledky jsou poté normalizovány vzhledem k ploše (m_{00}) výsledkem jsou souřadnice středu objektu:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (54) \quad (55)$$

Tyto souřadnice popisují jedinečnou pozici v obraze a mohou být využity k výpočtu centralizovaných momentů.

9.4 CENTRALIZOVANÉ MOMENTY

Definice diskretních centralizovaných momentů μ_{pq} popsanych v [16] je:

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^p (y - \bar{y})^q P_{xy} \quad (56)$$

Jedná se tedy v podstatě o posunutý Kartézský moment, což znamená, že centralizované momenty jsou invariantní vůči posunutí. Pro zavedení invariance vůči změně měřítka jsou využity momenty s normalizovaným měřítkem:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}} \quad (57)$$

kde:

$$\gamma = \frac{p+q}{2} + 1 \quad \forall (p+q) \geq 2 \quad (58)$$

9.5 SYMETRICKÉ VLASTNOSTI

Mírou asymetrie v obraze je tzv. koeficient šikmosti, který udává stupeň odchýlení rozdělení pravděpodobnosti náhodné veličiny od střední hodnoty. Koeficient šikmosti je momentem třetího řádu a je nulový v případě symetrického

rozdělení (např. Gaussova). Lze jej vypočítat pomocí momentů μ_{30} a μ_{03} . Směr asymetrie lze určit ze znaménka koeficientu. V [17] je popsána základní funkce $x^p y^q$ jako váhová funkce, která vhodným způsobem vybírá vlastnosti obrazu vzhledem k symetrii rozložení jasu. Tuto vlastnost lze využít k tomu aby centralizované a normalizované momenty nízkého řádu $p+q$ utvořily takový popis obrazu, který bude porovnatelný s existencí symetrie v obrazu. Symetrie v obraze je určována vzhledem k těžišti obrazu a proto je tedy vhodné využít centralizovaných momentů.

Prvních sedm centralizovaných momentů ($\eta_{11}, \eta_{20}, \eta_{02}, \eta_{21}, \eta_{12}, \eta_{30}, \eta_{03}$) s normalizovaným měřítkem je využito k analýze jednotlivých vysegmentovaných znaků jako binárních obrazů pomocí Hu momentů. Znaky symetrické podle osy x nebo y budou mít $\eta_{11}=0$. Pro znaky symetrické podle osy y pak $\eta_{12}=0$ a $\eta_{30}=0$. Pro znaky symetrické podle osy x bude nulový koeficient η_{03} a η_{12} bude kladný. Dále bude platit:

$$\eta_{pq}=0 \quad \forall p=0,2,4\dots ; q=1,3,5\dots \quad (59)$$

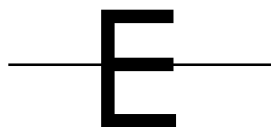
pro znaky symetrické podle osy x . Pro znaky nesymetrické bude platit:

$$\eta_{pq}<0 \quad \forall p=0,2,4\dots ; q=1,3,5\dots \quad (60)$$

a:

$$\eta_{p0}>0, \eta_{0p}>0 \quad \forall p=0,2,4\dots \quad f(x,y)>0 \quad (61)$$

Pro symetrii podle osy y platí stejná pravidla, s tím rozdílem, že jsou vzájemně zaměněny hodnoty parametrů p a q .



Obrázek 16: znak
symetrický podle
osy x



Obrázek 17: znak
symetrický podle
osy y

Z uvedených pravidel je zřejmé, že znaménka momentů mohou být uspořádána tak, aby udávala kvalitativní informaci o popisovaném znaku (např. existence symetrie), zatímco jejich velikost bude udávat kvantitativní popis (např. velikost a hustotu bodů).

znak	η_{11}	η_{20}	η_{02}	η_{21}	η_{12}	η_{30}	η_{03}
A	0	+	+	-	0	0	-
E	0	+	+	0	+	+	0

Tabulka 3: vybrané znaky a momenty indikující symetrii

9.6 MNOŽINA INVARIANTNÍCH (HU) MOMENTŮ

Neortogonální centralizované momenty jsou invariantní vůči posunutí a mohou být normalizovány vzhledem k měřítku. Avšak pokud je zapotřebí aby byly momenty invariantní i vůči otočení je třeba je přeformulovat. K tomu je možné využít jednu z metod, které popisuje M-K. Hu v [17]. Tato metoda se také nazývá metoda absolutně invariantních momentů. Je zapotřebí derivovat výrazy pro momenty invariantní v měřítku a posunutí aplikovaných na momentovou tvořící funkci s využitím rotační transformace. Výsledkem je množina ortogonálních absolutně invariantních momentů, které jsou posléze využity k rozpoznávání znaků. Tyto momenty jsou tedy vypočteny z normalizovaných centralizovaných momentů do třetího řádu podle následujících vztahů:

$$I_1 = \eta_{20} + \eta_{02} \quad (62)$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (63)$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (64)$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (65)$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (66)$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})] \quad (67)$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (68)$$

Jak je vidět ze vztahů, tyto momenty jsou konečného řádu a oproti centralizovaným momentům tedy neposkytují kompletní popis obrazu. Avšak je možné odvodit momenty řádu vyššího pokud je zapotřebí docílit zvýšené přesnosti popisu.

10. NEURONOVÉ SÍTĚ

Jedním z efektivních způsobů pro řešení komplexních problémů je tzv. přístup „rozděl a panuj“. Složitý systém může být rozdělen na jednodušší části proto aby mu bylo porozuměno. A naopak jednoduché prvky mohou být spojovány a tak tvořit složitý systém. Sítě jsou jedním z přístupů, jak tohoto cíle dosáhnout. Existuje velké množství různých typů sítí, avšak všechny jsou charakterizovány dvěma klíčovými komponenty: uzly a jejich spojeními.

Na tyto uzly se můžeme dívat jako na výpočetní jednotky. Mohou přijímat vstupní data, provádět s nimi nejrůznější operace (například jednoduché sčítání) nebo poskytovat výstupy či provádět operace komplexní (uzel může obsahovat další síť).

Spoje určují tok informací mezi uzly. Mohou být jednosměrné kde informace tečou pouze jedním směrem nebo dvoucestné pro tok informací tam i zpět.

Vzájemné ovlivňování uzlů skrze spojení vede ke globálnímu chování sítě, které nelze pozorovat v jednotlivých elementech sítě. To znamená, že schopnosti celé sítě předčí schopnosti samostatných uzlů, což činí ze sítí velmi mocný nástroj k řešení některých problémů ve fyzice, informačních technologiích, telekomunikacích apod.

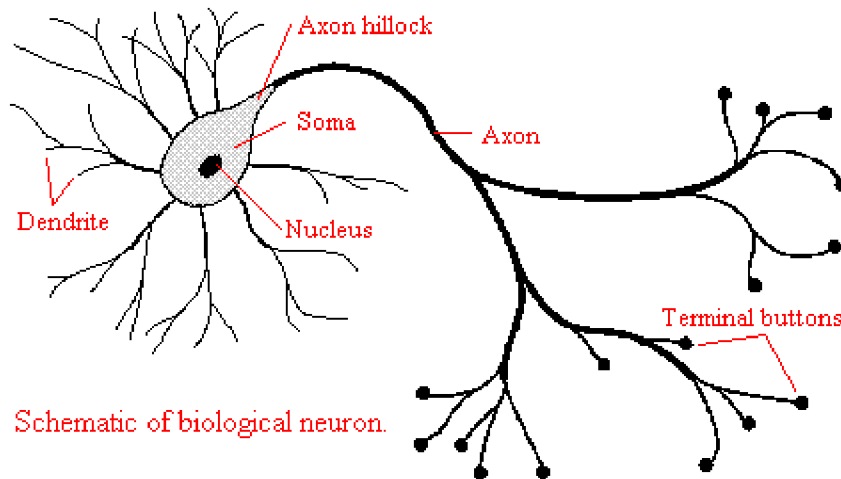
Jeden z typů sítí zaujímá pohled na uzly jako na umělé neurony. Takové sítě se nazývají umělé neuronové sítě (ANN z angl. Artificial neural networks). Umělý neuron je matematickým modelem vytvořeným podle funkce neuronu skutečného biologického.

10.1 PROCESNÍ JEDNOTKA-NEURON

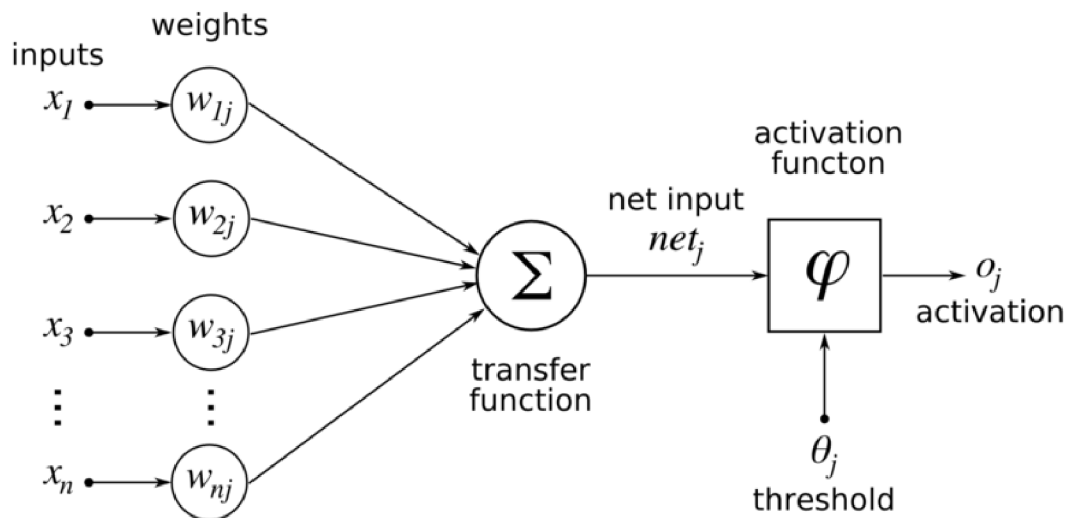
Biologický neuron přijímá signály skrze synapse umístěné na dendridech nebo membráně neuronu. Pokud jsou přijaté signály dostatečně silné (překročí určitý práh), je neuron aktivován a vyšle signál skrze axon. Tento signál může být poslán do další synapse a tím aktivovat další neurony.

Složitost skutečných neuronů je při modelování pomocí neuronů umělých značně zjednodušena. Umělé neurony se skládají ze vstupů (synapse), které jsou násobeny váhami (určují sílu signálu) a výstupu jehož velikost je obvykle dána

matematickou funkcí která určuje aktivaci neuronu. Umělé neuronové sítě tedy kombinují tyto umělé neurony pomocí kterých zpracovávají informace.



Obrázek 18: Biologický neuron [18]



Obrázek 19: Umělý neuron [19]

Každý neuron tedy vykonává relativně jednoduchou práci: přijme vstupy od svých sousedů nebo vnějších zdrojů a ty použije k výpočtu výstupního signálu, který je dále předáván. Druhým úkolem neuronu je upravování vstupních vah. Tyto operace lze pro více neuronů provádět paralelně.

V neuronových sítích je také vhodné rozlišovat celkem tři typy neuronů: vstupní neurony které slouží k příjmu dat z vnějšku sítě, výstupní neurony které

posílají data ven ze sítě a skryté neurony jejichž signály nejsou přístupné a zůstávají uvnitř sítě.

10.2 SPOJENÍ NEURONŮ

Ve většině případů lze předpokládat že každý neuron poskytuje výstup přičítaný na vstupech dalších neuronů ke kterým je připojen. Celkový vstup do neuronu je tedy vážený součet výstupů všech připojených neuronů s přičtením offsetu θ_j .

$$s_j(t) = \sum_i w_{ij}(t) o_i(t) + \theta_j(t) \quad (69)$$

Čím vyšší je hodnota váhy w_{ij} , tím silnější pak je pak vliv vstupu x_i na chování neuronu. V případě kladné váhy w_{ij} se jedná o excitaci (přičtení) v případě záporné váhy o inhibici (odečtení).

10.3 AKTIVACE A VÝSTUPY

Dalším potřebným parametrem je pravidlo podle kterého je určován celkový vliv vstupu neuronu na jeho aktivaci. Tedy funkci ϕ_j která zpracuje celkový vstup $s_j(t)$ a současnou hodnotu aktivace $o_j(t)$ a vytvoří hodnotu novou:

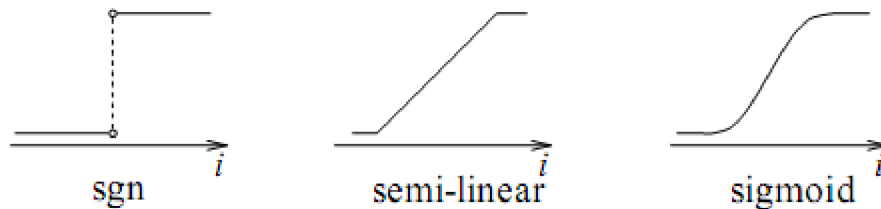
$$o_j(t+1) = \phi_j(o_j(t), s_j(t)) \quad (70)$$

Často se jedná o neklesající funkci celkového vstupu neuronu:

$$o_j(t+1) = \phi_j(o_j(t)) = \phi_j\left(\sum_i w_{(ij)} y_j(t) + \theta_j(t)\right) \quad (71)$$

Avšak aktivační funkce nemusí být neklesající. V podstatě je možno použít libovolnou prahovací funkci: signum, lineární, po částech lineární nebo hladce omezující funkci (viz obr.20). Jako aktivační funkce je často používána sigmoida jejíž průběh je určen jako:

$$o_j = \phi(s_j) = \frac{1}{1 + e^{-s_j}} \quad (72)$$



Obrázek 20: příklady aktivačních funkcí [20]

10.4 TOPOLOGIE SÍTÍ

Neurony mezi sebou mohou být propojeny několika způsoby. Hlavním rozdělením je rozdělení podle distribuce dat. Z tohoto hlediska lze topologii sítí rozčlenit do dvou skupin:

10.4.1 Feed-forward

U tohoto typu sítě data tečou striktně jedním směrem od vstupů sítě k výstupům. Tyto data mohou procházet několika vrstvami neuronů, neexistuje zde však žádná zpětná vazba. To znamená, že spojení neuronů probíhá pouze do následující vrstvy. Do této kategorie spadají sítě typu Perceptron a Adaline. Sítě typu feed-forward mají vrstvenou strukturu. Každá vrstva sestává z neuronů které získávají svůj vstup z předchozí vrstvy a svůj výstup předávají vrstvě následující. Vstupy N_i jsou připojeny k vrstvě $N_{h,1}$ skrytých neuronů. V těchto vstupních jednotkách nedochází k žádnému zpracování dat proto se tato vrstva ve většině případů nepočítá do samotných vrstev sítě (tzn. pokud má síť jednu vrstvu vstupní, jednu skrytou a jednu výstupní bavíme se o ní jako o síti dvouvrstvé). Aktivační funkcí skrytých neuronů je funkce vážených vstupů a offsetu podle rovnice (71) výstup je pak distribuován do další skryté vrstvy dokud není dosaženo vrstvy poslední jejíž výstupy jsou připojeny do neuronů vrstvy výstupní.

10.4.2 Rekurentní síť

Tyto sítě mohou obsahovat zpětnovazební spoje. Oproti feed-forward sítím jsou u tohoto typu důležité dynamické vlastnosti sítě. V některých případech

aktivační funkce prochází procesem relaxace a tím se dostane do stabilního stavu, ve kterém se již tato aktivační funkce nemění. Příkladem toho typu sítě jsou síť Hopfieldovy, Kohonenovy a Andersonovy.

10.5 UČENÍ UMĚLÝCH NEURONOVÝCH SÍTÍ

Neuronové síť musí být nastaveny tak, aby nastavením množiny vstupních dat byl (ať už přímo, nebo skrze proces relaxace) dosaženo požadovaného výstupu. Existuje mnoho různých metod jak ovlivnit sílu spojů mezi jednotlivými neurony. Jedním z nich je nastavení vah explicitně s využitím apriorní znalosti. Dalším způsobem je vytrénování neuronové sítě tak, že jsou jí poskytovány vzory a váhy spojů jsou poté upravovány dle učicího pravidla.

10.5.1 Možnosti učení

Metody učení lze rozdělit do dvou rozdílných skupin. Jsou to:

- Učení s dohledem nebo také Asociativní učení při kterém je síť učena poskytováním vstupních vzorů a jim odpovídajícím vzorům výstupním. Tyto vstupně-výstupní páry mohou být poskytovány vnějším učitelem, nebo systémem obsahujícím síť.
- Učení bez dohledu nebo Sebeorganizační učení při kterém je výstupní neuron trénován tak aby rozpoznal vstupní vzor. Při použití této metody se předpokládá, že síť najde statistické charakteristiky množiny vstupních dat. Na rozdíl od učení s dohledem však síť předem nezná počet kategorií do kterých mají být tyto vstupní data rozděleny; systém si tedy musí vytvořit svoji vlastní reprezentaci reakcí na vstupy.

Výsledkem obou zmíněných metod je upravení vah spojů mezi neurony pomocí modifikačního pravidla. Prakticky všechna pravidla učení mohou být považována za variantu Hebbova pravidla učení navrženého Hebbem v [21]. Základní myšlenkou je to, že pokud jsou dva neurony j a k aktivovány současně, pak jejich vzájemné spojení musí být posíleno. Pokud do j vstupuje výstup k , pak je použita nejjednodušší varianta pravidla, podle které je váha w_{jk} upravena podle následujícího pravidla:

$$\Delta w_{jk} = \gamma o_j o_k \quad (73)$$

kde γ je kladná konstanta reprezentující míru učení. Další běžně užívané pravidlo nepoužívá současnou velikost výstupu neuronu k ale rozdíl mezi současnou a požadovanou velikostí výstupu:

$$\Delta w_{jk} = \gamma o_j (d_k - o_k) \quad (74)$$

kde d_k je požadovaná velikost výstupu neuronu poskytnutá učitelem. Vztahu (74) se také často říká Widrow-Hoffovo pravidlo nebo také tzv. delta pravidlo.

10.5.2 Back-Propagation

Hlavní myšlenkou tohoto řešení je určení chyb jednotlivých neuronů pomocí zpětného šíření chyby výstupní vrstvy neuronů zpět směrem k vstupům. Z tohoto důvodu je se tato metoda také nazývá metoda zpětného šíření chyb. Back-propagation lze také považovat za zobecnění delta pravidla pro nelineární aktivační funkce a vícevrstvé sítě.

Vzhledem k použití nelineárních aktivačních funkcí je třeba zobecnit delta pravidlo. Aktivační funkce je derivovatelná funkce celkového vstupu dána

$$o_j^p = \phi(s_j^p) \quad (75)$$

kde

$$s_j^p = \sum_i w_{ij} o_i^p + \theta_j \quad (76)$$

Pro správné určení zobecněného delta pravidla je třeba zvolit

$$\Delta_p w_{ij} = -\gamma \frac{\delta E^p}{\delta w_{ij}} \quad (77)$$

Míra chyby E^p je definována jako celková kvadratická odchylka pro vzor p na výstupních neuronech

$$E^p = \frac{1}{2} \sum_{o=1}^{N_o} (d_o^p - o_o^p)^2 \quad (78)$$

kde d_o^p je požadovaný výstup pro neuron o na jehož vstupu je vzor p . Dále označíme $E = \sum_p E^p$ jako sečtenou kvadratickou chybu. Pak je možno napsat

$$\frac{\delta E^p}{\delta w_{ij}} = \frac{\delta E^p}{\delta s_k^p} \frac{\delta s_k^p}{\delta w_{ij}} \quad (79)$$

Z rovnice (76) můžeme vidět že druhý člen je

$$\frac{\delta s_j^p}{\delta w_{ij}} = o_i^p \quad (80)$$

a pokud definujeme

$$\delta_j^p = \frac{-\delta E^p}{\delta s_j^p} \quad (81)$$

čímž dostaneme ekvivalent delta pravidla jehož výsledkem je gradient klesání na ploše chyby pokud se budou váhy měnit podle

$$\Delta_p w_{ij} = \gamma \delta_j^p o_i^p \quad (82)$$

Nyní je klíčové zjistit hodnotu δ_j^p pro každý neuron v síti. To lze odvodit podle [20] s následujícím výsledkem:

pro výstupní neurony:

$$\delta_o^p = (d_o^p - o_o^p) \phi'_o(s_o^p) \quad (83)$$

pro skryté neurony:

$$\delta_h^p = \phi'_h(s_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho} \quad (84)$$

Tyto dva vztahy (83,84) určují postup při rekurzivním určování δ pro všechny neurony v síti. Tyto hodnoty jsou posléze využity pro změny vah podle rovnice (82). Tím je tedy získáno obecné delta pravidlo pro síť typu feed-forward používající nelineární neurony.

10.5.3 Algoritmus RPROP

Zkratka RPROP znamená pružné zpětné šíření chyby (z angl. **Resilient backpropagation**). Jedná se adaptivní učící schéma používané při dávkovém učení s dohledem ve vícevrstvých perceptronových sítích. Základním principem Rprop je eliminovat nežádoucí vliv velikosti parciální derivace na změnu vah spojení neuronů. Z tohoto důvodu je využito pouze znaménko derivace k určení směru změny váhy. Velikost této změny je určena tzv. hodnotou změny $\Delta_{jk}^{(t)}$:

$$\begin{aligned} \Delta w_{jk}^{(t)} = & -\Delta_{jk}^{(t)}, \text{ pokud } \frac{\delta E}{\delta w_{jk}}^{(t)} > 0 \\ & +\Delta_{jk}^{(t)}, \text{ pokud } \frac{\delta E}{\delta w_{jk}}^{(t)} < 0 \\ & \text{jinak, } 0 \end{aligned} \quad (85)$$

Kde $\frac{\delta E}{\delta w_{jk}^{(t)}}$ znamená celkovou gradientní informaci všech vzorů z množiny vzorů (dávkové učení). Nahrazením $\Delta_{jk}^{(t)}$ konstantní hodnotou změny Δ se rovnice (85) stává tzv. Manhattan pravidlem.

Druhým krokem učení pomocí Rprop je určení nových hodnot $\Delta_{jk}^{(t)}$. Tento krok je založen na znaménkově závislém adaptačním procesu podobném úpravě míry učení.

$$\begin{aligned} \Delta_{jk}^{(t)} = & \eta^+ * \Delta_{jk}^{(t-1)}, \text{ pokud } \frac{\delta E}{\delta w_{jk}^{(t-1)}} * \frac{\delta E}{\delta w_{jk}^{(t)}} > 0 \\ & \eta^- * \Delta_{jk}^{(t-1)}, \text{ pokud } \frac{\delta E}{\delta w_{jk}^{(t-1)}} * \frac{\delta E}{\delta w_{jk}^{(t)}} < 0 \\ & \text{jinak, } \Delta_{jk}^{(t-1)} \end{aligned} \quad (86)$$

kde $0 < \eta^- < 1 < \eta^+$

Slovy lze tedy toto adaptační pravidlo popsat takto: Pokaždé, když parciální derivace odpovídající váhy w_{jk} změni znaménko, znamená to že poslední změna byla příliš velká a algoritmus přeskočil lokální minimum, tak se hodnota změny $\Delta_{jk}^{(t)}$ zmenší o η^- . Pokud derivace znaménko nezmění, hodnota změny se mírně zvýší aby se zrychlila konvergence v mělkých oblastech. Navíc v případě změny znaménka by

nemělo dojít k adaptaci v případě úspěšného kroku učení. Toho lze v praxi dosáhnout

tak, že se nastaví $\frac{\delta E}{\delta w_{jk}}^{(t-1)} = 0$ ve výše uvedeném vztahu.

Kvůli snížení množství volně měnitelných parametrů, které často vede ke zdoluhavému hledání jsou parametry určující zvýšení a snížení hodnoty nastaveny napevno. Volba η^- je ovlivněna následujícími faktory: Pokud došlo k přeskočení minima, minulá hodnota byla příliš velká. Z důvodu nemožnosti určení hodnoty tohoto překročení z gradientu, je třeba odhadnout správnou hodnotu. Obvykle bývá dobrým odhadem zvolení právě poloviny hodnoty změny, proto $\eta^- = 0.5$. Oproti tomu hodnota η^+ musí být dost velká na to, aby bylo dosaženo rychlého růstu hodnoty změny v mělkých oblastech chybové funkce, na druhé straně proces učení může být citelně narušen pokud příliš velká hodnota tohoto parametru vede k trvalým změnám směru kroku změny váhy. Podle [22] bylo četným experimentováním dosaženo volby $\eta^+ = 1.2$. S touhou hodnotou je dosaženo velmi dobrých výsledků nezávisle na typu zkoumaného problému.

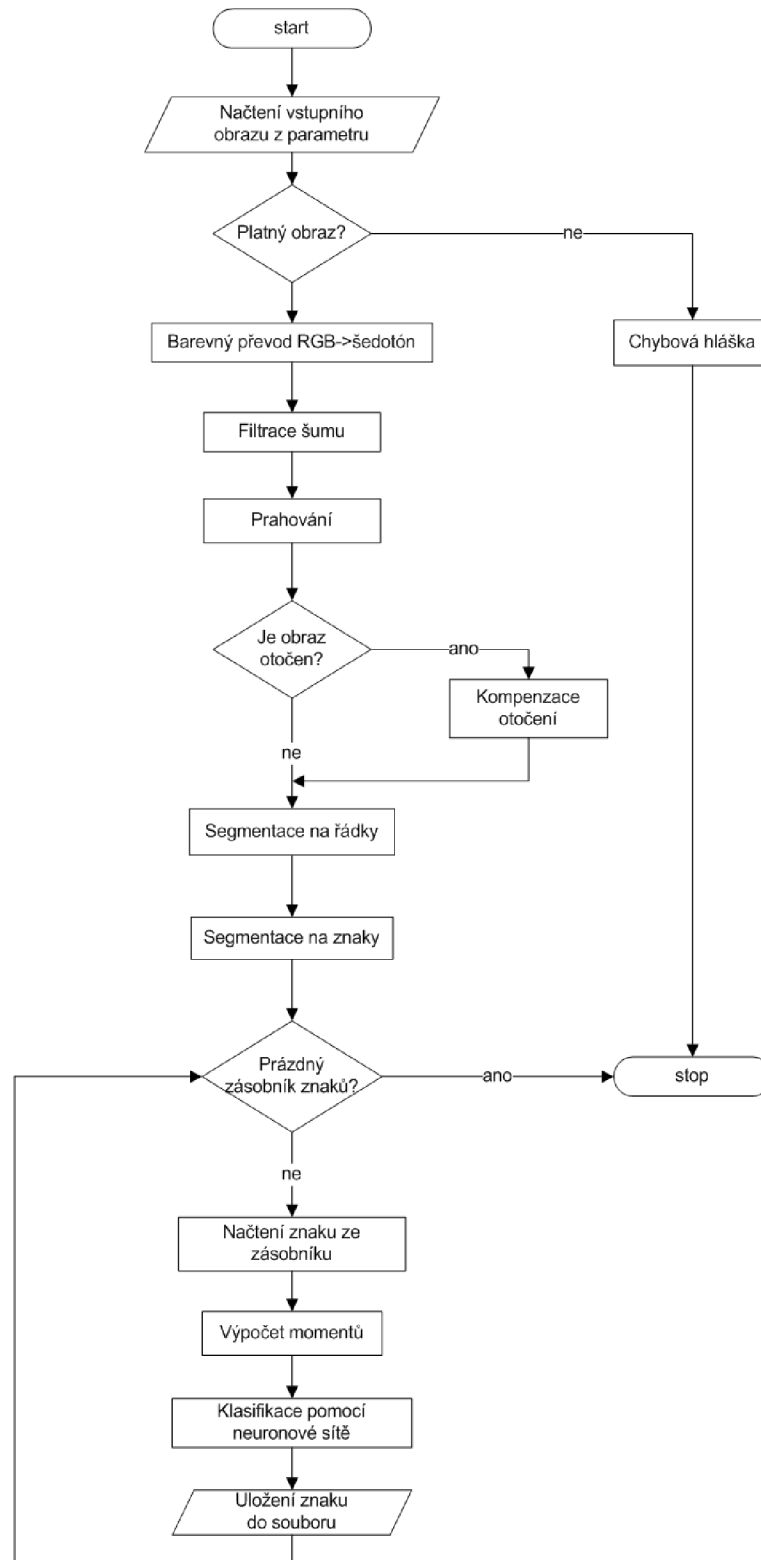
Rprop se tedy pokouší adaptovat proces učení na topologii chybové funkce a sleduje principy dávkového učení nebo také učení po epochách. To znamená že změny vah a adaptace jsou prováděny po získání gradientní informace z celé množiny vzorů.

10.5.4 Kaskádové učení

Kaskádové učení se liší od klasických metod učení tím, že učení začíná s prázdnou sítí (bez skrytých vrstev) a poté přidává neurony jeden za druhým zatímco se síť učí. Hlavní výhodou tohoto přístupu je zejména to, že není třeba předem určovat počet neuronů ve skrytých vrstvách před trénováním sítě. Tento postup také dosahuje lepších výsledků při řešení určitých problémů [23].

Principem kaskádového učení je vytvoření tzv. "kandidátů" tzn. Neuronů učených mimo skutečnou síť z kterých jsou poté vybrány ty jejichž výsledky jsou nejlepší a poté vloženy do sítě. Poté jsou trénovány výstupní spoje a připraveny další kandidáti.

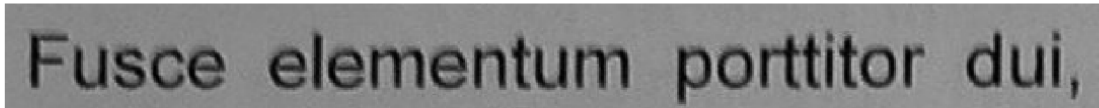
11. ALGORITMUS PROGRAMU



Obrázek 21: Vývojový diagram programu

11.1 BAREVNÝ PŘEVOD OBRAZU

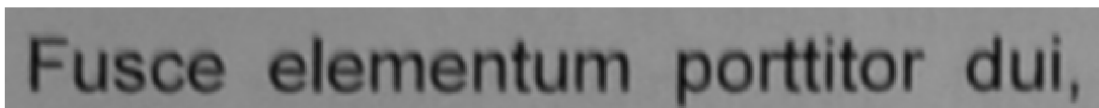
Po načtení vstupního obrazu programem je prvním krokem odstranění nadbytečné barevné informace a tedy převod z barevné oblasti RGB do oblasti šedotónové. Tento převod probíhá podle vztahu (35).



Obrázek 22: výřez použitého obrazu po odstranění nadbytečné barevné informace

11.2 FILTROVÁNÍ ŠUMU

Pro odfiltrování šumu je použit Gaussův filtr o rozměru masky 3x3 obrazové body. Gaussův filtr je vybrán z důvodu dobré aproximace rozložení šumu, velikost byla určena po experimentování jako vhodný kompromis mezi rozostřením obrazu a úrovní filtrace šumu.



Obrázek 23: výřez použitého obrazu po aplikaci Gaussova filtru

11.3 PRAHOVÁNÍ

Dalším krokem je převedení obrazu do formy binárního obrazu a tím odstranění i poslední zbytečné informace v obraze. V tomto kroku je využita Otsuho prahovací metoda popsána v kapitole 8.2.1, tato metoda je velmi výhodná z důvodu svojí rychlosti a efektivnosti.



Obrázek 24: výřez použitého obrazu po vyprahování pomocí Otsuho metody

Jak je vidět z obrázku s vyprahovaným obrazem, dochází ke spojení některých znaků, což později způsobuje problémy při klasifikaci (spojené znaky jsou považovány za jeden znak).

11.4 KOMPENZACE POOTOČENÍ OBRAZU

Nejprve je na vyprahovaný obraz aplikován Cannyho hranový filtr z důvodu značného zlepšení výsledků Houghovy transformace. Kompenzace pootočení obrazu je prováděna podle toho v jakém formátu je vstupní obraz. Pokud je vstupní obraz otočen na šířku Houghova transformace nalezne vyšší počet čar ve vertikálním směru a vstupní obraze je nejprve transponován, aby došlo k jeho přetočení na výšku. Poté jsou odfiltrovány všechny úhly přímek nalezených Houghovou transformací, které neodpovídají nastavené podmínce $-5^\circ < \varphi < 5^\circ$ (tato podmínka počítá s maximálním natočením fotografie o 5° a lze ji v programu snadno pozměnit). Dalším krokem je zprůměrování vyhovujících úhlů a předání této hodnoty jako parametru funkce pro otočení obrazu. Tím dojde narovnění obrazu. Nakonec je třeba znovu provést prahování, protože došlo k interpolaci obrazových bodů a odstranit černé rohy pomocí semínkového vyplňování (vznikly v důsledku přidání chybějících bodů při rotaci).

Sed elit justo, commodo non pellentesque eget, tincidunt ac ipsum. Pellentesque tincidunt tortor fermentum vehicula at ac ipsum. Donec magna turpis, volutpat id, ultrices vel leo. Etiam suscipit quam quis mauris ultricies at posuere. Phasellus id sapien sem, quis condimentum libero. Morbi non Vestibulum euismod fringilla magna, ut elementum purus pretium ac. Sed non tincidunt aliquet, metus libero vehicula lorem, non commodo augue elit. Donec nec urna justo, a egestas ligula. Maecenas condimentum mi ac ipsum congue. In nisl turpis, laoreet quis elementum at, placerat sit amet nulla. Du nunc. Aliquam rutrum ultricies urna, vitae tristique felis hendrerit eget. Morbi urna. Phasellus nec diam sapien, eget semper erat. Integer pulvinar magna sollicitudin at semper purus mollis. Morbi et nisi erat. Etiam a rhoncus leo. dui vitae massa rutrum eleifend. Suspendisse rutrum sollicitudin ipsum, volu orci iaculis quis.

Obrázek 25: výřez obrazu před vyrovnáním otočení

Sed elit justo, commodo non pellentesque eget, tincidunt ac ipsum. Pellente tincidunt tortor fermentum vehicula at ac ipsum. Donec magna turpis, volutpat id, ultrices vel leo. Etiam suscipit quam quis mauris ultricies at posuere. Phasellus id sapien sem, quis condimentum libero. Morbi non Vestibulum euismod fringilla magna, ut elementum purus pretium ac. Sed non tincidunt aliquet, metus libero vehicula lorem, non commodo augue eli Donec nec urna justo, a egestas ligula. Maecenas condimentum mi ac ij congue. In nisi turpis, laoreet quis elementum at, placerat sit amet nulla. Du nunc. Aliquam rutrum ultricies urna, vitae tristique felis hendrerit eget. Morbi urna. Phasellus nec diam sapien, eget semper erat. Integer pulvinar ma sollicitudin at semper purus mollis. Morbi et nisi erat. Etiam a rhoncus leo. dui vitae massa rutrum eleifend. Suspendisse rutrum sollicitudin ipsum, volu orci iaculis quis.

Obrázek 26: výřez obrazu po vyrovnáním otočení

Z obrázku je vidět, že kompenzace pootočení není ideální, bylo ale experimentálně ověřeno že na další zpracování textu toto malé natočení již nemá žádný zásadní vliv a to také z důvodu volby příznaků, které jsou vůči otočení invariantní.

11.5 SEGMENTACE STRÁNKY NA ŘÁDKY

K segmentaci obrazu je využito techniky projekce do jednotlivých os x a y obrazu. Segmentace obrazu na řádky textu probíhá v binárním obraze s využitím pole v němž jsou uloženy počty tmavých bodů na jednotlivých řádcích obrazu. Po vypočítání tohoto pole jsou v něm nalezeny prvky spadající do tolerančního pole nulové hodnoty. Hranice pro určení začátku řádku byla experimentálně určena na 15 obrazových bodů, pro určení konce řádku 5 bodů. Indexy nalezených prvků jsou pak označeny střídavě jako začátky a konce řádků. Obrazové body mezi začátky a konci jsou pak uloženy do datových struktur, kde v každé z nich je tedy uložen jeden řádek textu.

11.6 SEGMENTACE ŘÁDKŮ NA ZNAKY

Segmentace řádků textu na jednotlivé znaky funguje obdobně jako segmentace na řádky, s tím rozdílem, že se do pole ukládají počty bodů v

jednotlivých sloupcích a poté jsou hledány prvky pole s nulovou hodnotou (začátky a konce znaků).

Pokud je zjištěno, že šířka nalezeného znaku je větší než limit, je toto detekováno jako dva překrývající se znaky (např. dvojice „AV”) a dojde k dosegmentování s využitím semínkového vyplňování. Tato metoda funguje tak, že postupně prochází jednotlivé sloupce obrazových bodů, dokud nenarazí na černý bod. Poté je tento bod označen jako semínko (angl. seed) a to je použito pro vyplnění veškerých sousedících tmavých bodů, čímž dojde k vymazání znaku, tento postup se opakuje dokud nejsou odstraněny všechny znaky. Takto je zjištěn počet překrývajících se znaků. Následně dojde k postupnému ukládání jednotlivých znaků a vymaskování zbylých, na základě jejich zjištěného počtu v předchozím kroku. Například u dvojice „AV” se provede dvakrát vyplňování bílou barvou (první „A”, druhé „V”), poté je písmeno „V” odstraněno a uloží se „A”. Následně je odstraněno „A” a uloží se „V”.

11.7 VÝPOČET MOMENTŮ

Jako vstupy pro neuronové sítě, podle nichž jsou klasifikovány jednotlivé znaky slouží H_u momenty do sedmého řádu vypočtené podle vztahů (62) až (68). Tyto momenty jsou vybrány kvůli výhodným vlastnostem: invariance vůči otočení, měřítku a posuvu popsanych v [16] a [17]. Tyto vlastnosti jsou důležité kvůli tomu, že operace předzpracování nejsou dokonalé a výsledné jednotlivé znaky mohou být tedy natočeny nebo posunuty. Postup je tedy takový, že jsou nejdříve spočteny momenty kartézské podle vztahu (52), poté je nalezeno těžiště znaku dle vztahů (54) a (55), toto těžiště je dosazeno do vztahu (56) pro centralizované momenty z nichž jsou spočteny momenty normalizované podle vztahů (57) a (58) a tyto jsou dosazeny do výsledných vztahů pro množinu invariantních momentů. Výsledkem je tedy vždy pole o sedmi prvcích.

11.8 TRÉNOVACÍ A KLASIFIKAČNÍ DATA

Vstupní data jsou klasifikována do jedné z 57 tříd, každá z těchto tříd představuje jeden ze znaků anglické abecedy, ať už malých nebo velkých ve fontu

Arial velikosti 12 bodů. K abecedě jsou dále přidány znaky tečka, čárka, vykřičník, otazník a mezera.

Vstupní data jsou vytvořena ze tří stránek s vytištěnými klasifikovanými znaky. Tyto vytištěné stránky jsou poté vyfotografovány digitálním fotoaparátem o rozlišení 5Mpix. za různých světelných podmínek: za normálního osvětlení v místnosti, při ztlumeném běžném osvětlení a pouze s umělým osvětlením pomocí lampy. Pro každý z klasifikovaných znaků je takto zajištěno 128 vzorků. To znamená pro 56 klasifikovaných tříd celkem 7168 vzorků.

Požadované výstupní data jsou vytvořeny uměle jednoduchým programem a sestávají z matice o rozměrech 57x7168 hodnot. Na každém řádku matice je vždy pouze jedna hodnota rovna jedné, ostatní jsou nulové, což odpovídá ideálnímu rozložení pravděpodobnosti kdy by měl mít pouze jeden znak pravděpodobnost rovnu jedné a ostatní nule.

Data pro klasifikaci jsou pořízena stejným způsobem jako data pro učení sítí. Text pro klasifikaci je vygenerovaná tzv. tiskařská výplň (Lorem Ipsum).

11.9 POUŽITÉ NEURONOVÉ SÍTĚ

Ke klasifikaci bylo použito několik druhů sítí s různou topologií kvůli ověření vlivu změny parametrů na kvalitu naučení, společnou vlastností všech použitých sítí je základní struktura vstupů a výstupů. Vstupů je vždy 7 z důvodu sedmi použitých příznaků-Hu momentů a výstupů je vždy 57 z důvodu klasifikace do 57 tříd (viz kapitola 11.8) . Všechny sítě jsou typu feed-forward. Výstupem neuronové sítě je tedy vždy vektor 57 hodnot s pravděpodobnostmi určujícími příslušnost k dané třídě. Výsledkem je pak ta třída, jejíž pravděpodobnost je nejvyšší.

11.9.1 Síť učená algoritmem Rprop

Tato síť sestává ze dvou vrstev, ve skryté vrstvě je umístěno 80 neuronů s aktivační funkcí ve tvaru sigmoidy (viz obr.20) , učení sítě probíhalo podle algoritmu Rprop s počtem epoch učení 3000 aby se zabránilo přeučení sítě. U této sítě byl počet neuronů ve skryté vrstvě určen tzv. „zlatého pravidla“ uvedeného v [25 kap.5.1.2] podle kterého lze počet neuronů ve skryté vrstvě určit jako

$$q_h \leq \frac{\frac{N}{10} - q_{out}}{q_{in} + q_{out} + 1} \quad (87)$$

kde q_h je počet hledaných neuronů, q_{out} je počet výstupních neuronů, q_{in} je počet vstupních neuronů a N je počet vstupních hodnot (velikost trénovací množiny krát počet vstupních neuronů). Další zvyšování počtu neuronů nad tuto úroveň má již pouze malý vliv efektivitu naučení sítě a učení tím spíše zbytečně zpomaluje. Učením bylo dosaženo $MSE = 6.73 \cdot 10^{-2}$

Příklad výstupního textu:

Fuxce eIemeutum dNhitor du!, !uterduE Iac!uia lacns egextas nt. NuIIam uuuc jnsto, rutruc ut coubiceutum vitao, Eag!x v!tae idsum.

A text původní:

Fusce elementum porttitor dui, interdum lacinia lacus egestas ut. Nullam nunc justo, rutrum ut condimentum vitae, mattis vitae ipsum.

11.9.2 Síť učená kaskádním učením

Tato síť je na počátku učení jednovrstvá. Při učení jsou do skrytých vrstev postupně vkládány neurony. Požadavkem na počátku byl maximální počet neuronů 100 a požadované $MSE = 1.00 \cdot 10^{-2}$.

Učení této sítě trvalo několikanásobně delší dobu, než u algoritmu Rprop. Pro porovnání asi 3,5 hodiny oproti 30 minutám na PC s dvoujádrovým procesorem o taktu 3.5GHz. Síť nakonec využila všech 100 neuronů s automaticky zvolenou aktivační funkcí ve tvaru sigmoidy a dosáhla srovnatelného výsledku jako síť předchozí: $MSE = 7.26 \cdot 10^{-2}$

Příklad výstupního textu:

```
Fusce eIemeutum qNEitNr du!, !uterduE lac!uia lacnx  
egextax nt. NuIIam uunc !nxtto, rutruc ut cNubiceutuE  
vitae, EaR!x v!tae iqxum.
```

11.9.3 Síť vytvořená v Matlabu

Vzhledem k tomu že použitá knihovna pro neuronové sítě (FANN viz kapitola 12.1.2) umožňuje pouze učení pomocí algoritmu Rprop nebo kaskádní učení byla poslední realizovaná síť vytvořena, naučena a ozkoušena v programu Matlab pomocí nástroje *nprtool*. Tento nástroj slouží pro práci s neuronovými sítěmi určenými pro rozpoznávání a klasifikaci.

Síť má dvě vrstvy a je typu feed-forward, ve skryté vrstvě je stejně jako u první sítě 80 neuronů. Všechny neurony mají aktivační funkci sigmoidy. K učení je použita metoda konjugovaných gradientů, jedná se o iterační metodu, jejíž podrobný popis lze nalézt v [26].

Vstupní data jsou náhodně rozdělena do tří skupin: na data pro učení (70% celkových dat), data pro ověření (15%) a data pro testování (15%). Data pro ověření slouží k měření schopnosti zobecnění sítě, data pro testování nemají na proces učení žádný vliv a slouží k měření výkonu sítě během a po ukončení učení.

Této síti se učením podařilo dosáhnout $MSE = 7.22 \cdot 10^{-3}$ při době učení 30 minut jak je ale vidět z následujícího výstupního textu, výsledky sítě jsou spíše horší než v případě sítí vytvořených s pomocí knihovny FANN.

Příklad výstupního textu:

```
FMsMe eIezeMtMz wNbitNr dui, iMtepom laS!uia lacox  
egextax Ht. NHllam nnnQ jnxtto, rMtrNS ut XNudiXeutum  
vitae, max!x v!tae ihsum .
```

11.9.4 Zhodnocení funkce sítí

Lze si všimnout, že u všech použitých sítí dochází k záměnám písmen „l” a „I”, tato písmena jsou si totiž v tištěné podobě velmi podobná a síť tedy není schopna rozpoznat rozdíl. Dalšími záměnami jsou „i” a „!”, tyto znaky jsou klasifikovány špatně z důvodu použitých příznaků, které jsou invariantní vůči rotaci a tedy i otočení, vygenerované momenty jsou tedy velmi podobné nebo stejné a síť se tyto znaky rozpoznávat vůbec nenaučí. Ze stejného důvodu dochází i k záměnám znaků „p” „b” „d” a „q” stejně jako „u” a „n” a dvojice „i” a „j” které jsou si také podobné.

K dalším chybám dochází slítním některých znaků dohromady. Tohoto si lze všimnout na obr. 20, který obsahuje první větu z výše zobrazeného textu.

Některé z těchto problémů jako je neschopnost rozpoznat některé otočené znaky mezi sebou by nejspíše bylo možné odstranit pomocí rozdělení znaku například do kvadrantů a podle hustoty tmavých bodů v každém z kvadrantu pak určit rozdíly (např. pro „p” „b” „d” a „q”). Další možností je využití jiného typu příznaků např. místo Hu-momentů použít komplexní Zernikeho momenty a text profiltrovat pomocí kontextového rozpoznávání (odstranění vykřičníků a velkých písmen uprostřed slov).

Síť	Počet neuronů ve skrytých vrstvách	MSE	Procentní úspěšnost sítě při klasifikaci	Doba učení
Matlab	80	$7,22 \cdot 10^{-3}$	41%	30min.
Rprop	80	$6,73 \cdot 10^{-2}$	62%	45min.
Kaskádní	100	$7,26 \cdot 10^{-2}$	52%	3,5h

Tabulka 4: Srovnání parametrů jednotlivých sítí

12. FUNKCE VYTVOŘENÝCH PROGRAMŮ

Výsledný program sestává z několika komponent. Jsou to tři spustitelné EXE soubory, dva soubory datové a několik dynamických knihoven. Celý projekt využívá knihovny OpenCV verze 2.0 a FANN ve verzi 2.0.0.

12.1 POUŽITÉ KNIHOVNY

12.1.1 OpenCV

Knihovna OpenCV (**Open Source Computer Vision**) je napsána v jazyce C/C++ pro počítačové vidění v reálném čase distribuována pod licencí BSD [24]. Sestává z dynamických knihoven *libcv200.dll*, *libcvaux200.dll*, *libcxcore200.dll*, *libcxts200.dll*, *libhighgui200.dll* a *libml200.dll*.

12.1.2 FANN

Jedná se o knihovnu s otevřeným zdrojovým kódem (open source), která implementuje vícevrstvé neuronové sítě v jazyce C. V knihovně FANN (zkratka z **Fast Artificial Neural Network** – rychlé umělé neuronové sítě) jsou funkce použité při vytváření sítí, jejich učení a klasifikaci příznaků. Tyto funkce jsou shromážděny v souboru *fann.dll*.

12.2 HLAVNÍ PROGRAM

Jedná se o soubor *ocr.exe*. V tomto souboru probíhají nejprve operace předzpracování obrazu (barevné převody, filtrace...), poté je načtena ze souboru *network.net* potřebná neuronová síť a s jejím využitím jsou zde klasifikovány příznaky získané z vysegmentovaných znaků. Po klasifikaci dojde k uložení výsledků do souboru o stejném názvu jako soubor vstupní pouze se změněnou koncovkou na *.txt*. Spuštění této aplikace se provádí přetažením zdrojového obrazu do této aplikace, případně v příkazové řádce jako připsáním cesty k zdroji jako parametru spustitelného souboru.

12.3 PROGRAM PRO UČENÍ METODOU RPROP

Tato aplikace umístěná v souboru *train.exe* slouží pro vytvoření a učení neuronové sítě na základě parametrů zadaných do rozhraní. Spouští se přetažením dat pro učení (*train.data*) do spustitelného souboru. Aplikace se nejprve zeptá na počet neuronů které mají být umístěny ve skryté vrstvě, poté na maximální počet epoch pro učení a požadovanou chybu učení (MSE – střední kvadratická odchylka, určuje rozdíl mezi aktuálním stavem výstupu a požadovaným stavem). V průběhu učení je postupně po každých sto epochách vytištěn výpis zobrazující dosaženou velikost MSE. Po dokončení učení je výsledná neuronová síť uložena do souboru *network.net*.

12.4 PROGRAM PRO KASKÁDOVÉ UČENÍ

Aplikace *cascade_train.exe*. Funkce je velmi podobná jako u aplikace *train.exe*, tzn. ke spuštění dochází opět přetažením datového souboru do aplikace, poté je třeba zadat maximální počet přidávaných neuronů a požadovanou odchylku MSE.

V průběhu učení je po každých pěti přidávaných neuronech zobrazen výpis s dosaženou odchylkou. Po dokončení učení je vytvořená síť uložena do souboru *network.net*.

12.5 SOUBOR S DATY PRO UČENÍ

Soubor *train.data*. V tomto souboru jsou uložena data ze kterých se učí požadovaná neuronová síť, data jsou uložena jako text ve formátu mantisa a exponent, kromě prvního řádku.

Na prvním řádku jsou postupně uložena tato data: počet párů vstupní-výstupní data pro učení, počet neuronů ve vstupní vrstvě a počet neuronů ve výstupní vrstvě. Zbylé řádky jsou již samotná data, hodnoty jsou odděleny alespoň jednou mezerou, řádky končí enterem.

13. ZÁVĚR

V průběhu práce se úspěšně podařilo vytvořit výslednou aplikaci v programovacím jazyce C++ která demonstuje techniku optického rozpoznávání znaků pomocí neuronových sítí na vstupních obrazech pořízených digitálním fotoaparátem.

Práci lze rozdělit do dvou hlavních oblastí: předzpracování a extrakce příznaků s klasifikací. Předzpracování využívá nejprve Houghovy transformace jako prostředku pro vyrovnání pootočení obrazu a následně poměrně velmi efektivní Otsuho prahovací metodu, díky níž je usnadněno získávání příznaků. K segmentaci stránek textu je využito metody průmětů do jednotlivých os, kdy nejprve dojde k promítnutí stránky do osy y čímž je umožněna segmentace na řádky, dalším krokem je promítnutí řádků do osy x a tím rozsegmentování na jednotlivé znaky. Pro ošetření překrývajících se znaků je využito semínkového vyplňování.

Příznaky jsou získávány v podobě sedmi statistických momentů invariantních vůči translaci otočení i měřítku. Získané momenty jsou poté předány do vybrané neuronové sítě, která funguje jako klasifikátor do padesáti sedmi tříd.

Jako nejlepší ze tří testovaných sítí se jevila dvouvrstvá síť architektury feed-forward s osmdesáti neurony ve skryté vrstvě, naučená pomocí algoritmu Rprop. Tato síť dosahovala nejlepšího procentního výsledku rozpoznání znaků a její učení bylo relativně krátké (viz tabulka 4).

Dalšího zlepšení procentní úspěšnosti rozpoznání znaků by nejspíše bylo možné dosáhnout změnou příznaků za jiné (např. vytvořit parametry pomocí skeletonizace znaků, použití Zernikeho komplexních momentů) nebo kombinací s dalšími pokročilejšími metodami jako jsou korekce znaků v závislosti na kontextu a tím odstranění nežádoucích vlastností Hu momentů – nemožnost rozpoznat znaky, které jsou převrácené ať už horizontálně nebo vertikálně.

14. SEZNAM LITERATURY

- [1] HLAVÁČ, Václav, SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. Praha: ČVUT, 2005. 255 s. ISBN 80-01-03110-1.
- [2] HORÁK, Karel, KALOVÁ, Ilona, PETYOVSKÝ, Petr, RICHTER, Miloslav. *Počítačové vidění*. Brno: VUT, 2008
- [3] BRADSKY, Gary, KAEHLER, Adrian. *Learning OpenCV*. First Edition. Sebastopol (California): O'Reilly Media, Inc., 2008. ISBN 978-0-596-51613-0.
- [4] opencv.pdf. Dokumentace knihovny OpenCV, distribuovaná společně s knihovnou, 2009. Dostupné z:
<<http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.0/OpenCV-2.0.0a-win32.exe/download>> [cit. 10.11.2009].
- [5] Dr. Andrew Greensted. *Otsu Thresholding – The Lab Book Pages*. [online]. 11.9.2009, [cit. 20.11.2009].
<<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>>
- [6] *Hough Transform – Wikipedia, the free encyclopedia* [online]. 10.12.2009, [cit. 5.10.2009]. <http://en.wikipedia.org/wiki/Hough_transform>
- [7] Daf-de. *Hough Transform example results* [online]. 31.8.2006, [cit. 28.11.2009].
<<http://en.wikipedia.org/wiki/File:Hough-example-result-en.png>>
- [8] *barva-3.png* [online]. [cit. 30.11.2009]. <<http://barva.navajo.cz/barva-3.png>>
- [9] Dr. Andrew Greensted. *OtsuOrig.png* [online]. [cit. 28.11.2009].
<<http://www.labbookpages.co.uk/software/imgProc/files/otsuOrig.png>>
- [10] Dr. Andrew Greensted. *OtsuT0.png* [online]. [cit. 28.11.2009].
<<http://www.labbookpages.co.uk/software/imgProc/files/otsuT0.png>>
- [11] Dr. Andrew Greensted. *OtsuT1.png* [online]. [cit. 28.11.2009].
<<http://www.labbookpages.co.uk/software/imgProc/files/otsuT1.png>>
- [12] Dr. Andrew Greensted. *OtsuT2.png* [online]. [cit. 28.11.2009].
<<http://www.labbookpages.co.uk/software/imgProc/files/otsuT2.png>>
- [13] Dr. Andrew Greensted. *OtsuT3.png* [online]. [cit. 28.11.2009].
<<http://www.labbookpages.co.uk/software/imgProc/files/otsuT3.png>>
- [14] Dr. Andrew Greensted. *OtsuT4.png* [online]. [cit. 28.11.2009].

<<http://www.labbookpages.co.uk/software/imgProc/files/otsuT4.png>>

[15] Dr. Andrew Greensted. *OtsuT5.png* [online]. [cit. 28.11.2009].

<<http://www.labbookpages.co.uk/software/imgProc/files/otsuT5.png>>

[16] SHUTLER, Jamie . *Informatics Homepages Server* [online]. 2002-08-15 [cit. 2010-05-17]. CVonline_moments. Dostupné z:

<http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/CVonline_moments.html>.

[17] HU, Ming-Kuei. *Visual pattern recognition by moment invariants*. Volume: 8 Issue: 2., 1962. IRE transactions on Information theory, s. 179-187.

10.1109/TIT.1962.1057692 .

[18] *Serendip's Exchange* [online]. 2003, 2003-05-07 [cit. 2010-05-19]. Axon Definition. Dostupné z:

<<http://serendip.brynmawr.edu/bb/kinser/definitions/neuron1.gif>>.

[19] File:ArtificialNeuronModel_english.png. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 14. July 2005, last modified on 1. April 2006 [cit. 2010-05-19]. Dostupné z:

<http://en.labs.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png>

[20] KRÖSE, Ben; VAN DER SMAGT, Patrick. *An Introduction to Neural Networks*. eight edition. Kruislaan 403, NL-1098 SJ Amsterdam : The University of Amsterdam, November 1996. 135 s.

[21] HEBB, Donald Olding. *The organization of behavior : a neuropsychological theory* . New York : Wiley, 1949. 321 s. Dostupné z:

<[http://www.google.com/books?](http://www.google.com/books?hl=cs&lr=&id=VNetYrB8EBoC&oi=fnd&pg=PR7&dq=hebb+organization+of+behavior&ots=y4aWOtZXW3&sig=DOeHIN98nWUT3Ghj1aZCaAd2kes#v=onepage&q&f=false)

[hl=cs&lr=&id=VNetYrB8EBoC&oi=fnd&pg=PR7&dq=hebb+organization+of+behavior&ots=y4aWOtZXW3&sig=DOeHIN98nWUT3Ghj1aZCaAd2kes#v=onepage&q&f=false](http://www.google.com/books?hl=cs&lr=&id=VNetYrB8EBoC&oi=fnd&pg=PR7&dq=hebb+organization+of+behavior&ots=y4aWOtZXW3&sig=DOeHIN98nWUT3Ghj1aZCaAd2kes#v=onepage&q&f=false)>. ISBN 0-8058-4300-0.

[22] RIEDMILLER, Martin Rprop - Description and Implementation Details.

In Technical Report [online]. W-76128 Karlsruhe : Institut für Logik, Komplexität und Deduktionsysteme, University of Karlsruhe, 1994 [cit. 2010-05-22]. Dostupné z:

<<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.3428>>

[23]NISSEN, Steffen. Fast Artificial Neural Network Library [online]. 2010 [cit. 2010-05-22]. FANN Cascade Training. Dostupné z:

<http://leenissen.dk/fann/html/files/fann_cascade-h.html>.

[24] *Licenční úmluva BSD*. Open Source Initiative (OSI)[online]. 2010 [cit. 2010-05-22]. Dostupné z <<http://opensource.org/licenses/bsd-license.php>>

[25]VALA, Tomáš. *Rozpoznávání SPZ z jednoho snímku*. Uherské Hradiště, 2006. 83 s. Diplomová práce. Univerzita Karlova v Praze.

[26]Martin Fodslette Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525-533, 1993.