

Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Matej Hrinčár

Rozpoznávání dopravních značek v obrázcích

Katedra softwarového inženýrství

Vedoucí bakalářské práce: Mgr. Mikuláš Patočka

Studijní program: informatika, programování

2008

Pod'akovanie

Na tomto mieste by som chcel poďakovať svojmu vedúcemu bakalárskej práce, pánovi Mgr. Mikulášovi Patočkovi za podnetné pripomienky, rady pri návrhu algoritmov, priebežnú kontrolu mojich výsledkov a celkové vedenie práce.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe 29. 5. 2008

Matej Hrinčár

Názov práce: Rozpoznávání dopravních značek v obrázcích

Autor: Matej Hrinčár

Katedra (ústav): Katedra softwarového inženýrství

Vedúci bakalárskej práce: Mgr. Mikuláš Patočka

E-mail vedúceho práce: mikulas@artax.karlin.mff.cuni.cz

Abstrakt

Cieľom práce je vytvoriť program, ktorý bude (pokiaľ možno v reálnom čase) rozpoznávať dopravné značky na obrázkoch. Program by mal rozpoznať značku, ak žiadna jej časť nie je ničím zakrytá a je otočená do rozumného uhlu. Program by nemalo zmiať pozadie alebo objekty podobnej farby. V práci sa riešia dva problémy. Jeden z nich je nájsť značku na vstupnom obrázku, druhý je nájdenú značku rozpoznať. Pre riešenie prvého je využitá farebná segmentácia t.j. extrakcia zaujímavej farby z obrázku. V tomto novom obrázku sa hľadajú tvary podobné dopravným značkám. Na rozpoznanie značky sme použili neurónovú sieť.

Kľúčové slová: rozpoznávanie, dopravné značky, farebná segmentácia, neurónové siete

Title: Program for recognition of traffic signs.

Author: Matej Hrinčár

Department: Department of Software Engineering

Supervisor: Mgr. Mikuláš Patočka

Supervisor's e-mail address: mikulas@artax.karlin.mff.cuni.cz

Abstract

The aim of the following document is to create a program that will be able (preferably in the real time) to recognize the traffic signs on the pictures. The program should recognize the sign on the condition that no part of it is covered by any object and the sign is turned in a reasonable angle. Any background or objects of other colors should not confuse the program. The document solves two problems. The first one is finding the sign on the input image, the second one is recognizing the sign. For solving the first problem the color segmentation is used, which means the extraction of the needed color from the image. In this new image the program finds shapes similar to the traffic signs. Artificial neural network is used to recognize the sign.

Keywords: recognition, traffic signs, color segmentation, artificial neural network, images processing

Obsah

ÚVOD	7
1. DOPRAVNÉ ZNAČKY	8
1.1. ROZDELENIE	8
1.2. VÝSKYT	9
1.3. VYMEDZENIE	10
2. SPÔSOBY RIEŠENIA	11
2.1. ZÁKLADNÉ ROZDELENIE	11
2.2. FAREBNÁ SEGMENTÁCIA	11
2.3. VYHLADANIE MOŽNÝCH DOPRAVNÝCH ZNAČIEK	13
2.4. ROZPOZNANIE DOPRAVNEJ ZNAČKY	13
3. PREHLAD PODOBNÝCH PRÁC	15
4. VLASTNÁ IMPLEMENTÁCIA	17
4.1. FAREBNÁ SEGMENTÁCIA	17
4.1.1. IMPLEMENTÁCIA	18
4.2. NÁJDENIE ZNAČIEK	19
4.3. ROZPOZNÁVANIE	20
4.3.1. PREDSPRACOVANIE	20
4.3.2. ROZPOZNANIE ZNAČIEK ZÁKAZ ZASTAVENIA, STÁTIA A VJAZDU	21
4.3.3. ROZPOZNANIE OSTATNÝCH ZNAČIEK	23
4.4. POPIS TRIED PROGRAMU	24
5. VÝSLEDKY VLASTNEJ IMPLEMENTÁCIE	26
5.1. RÝCHLOSŤ	26
5.2. ÚSPEŠNOSŤ	27
5.2.1. ÚSPEŠNOSŤ VYHLADÁVANIA	27
5.2.2. ÚSPEŠNOSŤ ROZPOZNÁVANIA	28

ZÁVER	30
UŽÍVATEĽSKÁ DOKUMENTÁCIA	31
HĽAVNÝ PROGRAM	31
PROGRAM NA UČENIE NEURÓNOVEJ SIETE	33
LITERATÚRA	34

ÚVOD

Témou tejto bakalárskej práce je rozpoznávanie dopravných značiek na danom vstupnom obrázku alebo videu. Výsledkom práce je program, ktorý v danej fotografii zobrazujúcej dopravnú situáciu označí a pomenuje dopravné značky nachádzajúce sa na vstupnej fotografii. Budeme sa snažiť o čo najmenší počet falošných poplachov (t.j. prípadov, keď program označí objekt, ktorý nie je značkou) a o čo najväčšiu úspešnosť v správnom označení a následnom pomenovaní dopravnej značky. Fotografia by mala byť zhotovená z pohybujúceho sa auta a mala by zachytiť vodičov výhľad na vozovku. Program by mohol slúžiť napr. pre podporu riadenia motorového vozidla a informovať vodiča o dopravnej situácii, poprípade poloautomatizovať riadenie napríklad prispôbovaním maximálnej povolenej rýchlosti. Práca si nekladie za cieľ dosiahnuť výsledky použiteľné pri úplne automatickom riadení vozidla ani úplne porozumieť dopravnej scéne, pretože tento problém je obecné zložitý. Na analyzovanie a vyhodnotenie dopravnej situácie je samotné vyhľadanie a rozpoznanie značky iba malá časť celkového riešenia, napríklad z toho dôvodu, že význam dopravných značiek ukončuje aj križovatka, alebo rozpoznaná značka nemusí byť určená pre nás. Niekedy je dopravná situácia zložitá aj pre skúseného vodiča. Na komplexnú analýzu tohto problému rozsah tejto práce nestačí.

Práca je rozdelená do niekoľkých kapitol. Prvá kapitola popisuje dopravné značky ako také, rozdelí ich do skupín podľa významu, tvaru a farby a vysvetlí zvláštnosti v jednotlivých skupinách. Definuje značky ktoré budeme schopní rozpoznať. Ďalej pojednáva o výskyte značiek, ich pravdepodobným umiestnením, veľkosťou na vstupnej fotografii. Tieto poznatky budú uplatnené pri rozhodovaní, či ide naozaj o dopravnú značku. Druhá popíše rôzne spôsoby možností rozpoznávania dopravných značiek. Tretia kapitola priblíži niektoré podobné projekty zaoberajúce sa rozpoznávaním značiek. Štvrtá kapitola sa zaoberá implementáciou vybraného spôsobu a detailne popíše zvolený spôsob rozpoznávania. Posledná kapitola zhrnie výsledky implementovaného algoritmu. Na záver je ešte priložená užívateľská dokumentácia, v ktorej popisujeme všetky parametre programu.

1. DOPRAVNÉ ZNAČKY

1.1. Rozdelenie

Dopravné značky sa podľa zákona o premávke na pozemných komunikáciách delia do dvoch hlavných kategórií: na zvislé a vodorovné. Tato práca sa zaoberá iba zvislým dopravným značením. Zvislé dopravné značky sa ďalej delia na:

- Výstražné značky
- Zákazové značky
- Príkazové značky
- Informatívne značky
- Dodatočné tabule

Cieľom tejto práce nie je rozpoznávať všetky druhy značiek, ale úspešne rozpoznávať aspoň časť a program napísať tak, aby sa ďalšia skupina, prípadne značka, dala jednoducho pridať. Najväčší dôraz je kladený na zákazové značky. Každá z uvedených skupín je význačná svojím tvarom a farbou, avšak v každej sa nájdú výnimky, ktoré si v nasledujúcich odsekoch popíšeme.

Výstražné značky – až na značenie upozorňujúce na železničné priecestie, sú všetky značky z tejto kategórie trojuholníkového tvaru, s červeným okrajom a piktogramom upresňujúcim ich význam.

Zákazové značky – všetky značky z tejto skupiny majú tvar kruhu. Väčšina z nich je ohraničená červeným okrajom a piktogramom uprostred. Výnimky sú iba vo farebnom stvárnení. Značky zakazujúce státie a zastavenie vozidla sú tvorené aj modrou farbou a značky, ktoré rušia zákazy, nie sú ohraničené červeným okrajom. Nakoniec značky zákazu vjazdu neobsahujú piktogram.

Príkazové značky – túto skupinu tvoria modré kruhové značky s bielym piktogramom, okrem značiek Daj prednosť v jazde a STOP.

Informatívne značky – je to asi najrôznorodjšia skupina značiek, obsahuje značky hlavnej cesty a modré a zelené informatívne tabule, ale aj napr. informácie o názvoch miest.

Dodatkové tabuľky – upresňujú význam dopravných značiek. Patria k nim napríklad značky upresňujúce tvar križovatky, alebo vymedzujúce platnosť značky.

1.2. Výskyt

Jednotlivé dopravné značky sa však na cestách nevyskytujú s rovnakou pravdepodobnosťou, preto sa zameriame iba na značky s vyšším výskytom, najmä pre dostatočné tréningové dáta. Konkrétne značky, ktoré budem schopný rozpoznať, budú uvedené ďalej.

Ako už bolo napísané, predpokladá sa, že vstupná fotografia bude zhotovená z pohybujúceho sa auta a mala by zachytiť vodičov výhľad na vozovku. Predpokladajme ďalej, že horizont je vo vodorovnej polohe, približne v strede fotografie. Potom sa značky malej veľkosti (voči veľkosti fotografie) nebudú vyskytovať v dolnej časti fotografie. Avšak značky, ktoré sú bližšie k pozorovateľovi, sú väčšie a môžu zasahovať aj do vrchnej časti fotografie, podobne sa vo vrchnej časti môžu nachádzať aj menšie značky umiestnené na trámoch. Dokonca by sa dopravné značky, ktoré by mali byť určené pre nás, nemali nachádzať ani v pravej časti fotografie. Tieto vedomosti môžeme uplatniť pri hľadaní značky na fotografii, poprípade odmietnuť podobný objekt, ktorý sa nachádza tam, kde by dopravná značka nemala čo robiť. Alebo môžeme tento poznatok využiť tak, že snímacie zariadenie nasmerujeme správnym smerom (v našom prípade vpravo a nahor) a budeme predpokladať výskyt značiek na celom vstupnom obrázku.



Obrázok 2: Snímacie zariadenie sníma celý výhľad vodiča



Obrázok 2: Snímacie zariadenie je už nasmerované správnym smerom.

1.3. Vymedzenie

Dopravné značky môžu byť čiastočne alebo celkom zakryté, buď inou značkou, alebo iným objektom (stromom, autom...), taktiež byť poškodené vandalizmom alebo prírodným vplyvom. Takto znehodnotené alebo zakryté značky môžu stratiť niektorú zo svojich charakteristík, ako je tvar, farba alebo piktogram. Podobne na fotografii môže byť značka natoľko neostrá, že to znemožní jej vyhľadanie a rozpoznanie. Rozpoznať zdeformované dopravné značenie môže byť nemožné, preto sa v tejto práci obmedzíme na rozpoznanie nezakrytých a úplne nerozostrených značiek (v popísaných prípadoch nedokážeme zaručiť, že značku vôbec nájdeme, popřípade rozpoznáme). Rozmazanie značiek vplyvom otrasov vozidla by sa dalo obmedziť použitím stabilizátora obrazu v snímacom zariadení.

V tejto práci som sa rozhodol zamerať na značky *zákazové*. Tie nám niečo „zakazujú“ a ich prehliadnutie nás môže stať prinajlepšom iba pokutu. Takže výsledný program bude rozpoznať značky (aj to nie všetky) z tejto skupiny.



Obrázok 3: Príklad rozmazanej značky

2. SPÔSOBY RIEŠENIA

2.1. Základné rozdelenie

Problém rozpoznávania dopravných značiek je vo svete stále aktuálny. Svedčí o tom množstvo projektov zaoberajúcich sa podobnou problematikou. Prvý takýto pokus bol už v 80. rokoch minulého storočia. Proces rozpoznávania by sa dal rozdeliť na dve základné časti:

- **Vyhľadanie**
- **Rozpoznanie**

Podľa implementácie prvej časti sa projekty delia (okrem iného) na dve hlavné skupiny. Prvá z nich rozpoznáva (vyhľadáva) značky na *čiernobielych* fotografiách, druhá z nich predpokladá na vstupe aj *farebnú informáciu*.

Z druhej skupiny využívajúcej i farebnú informáciu poukážeme na viac podobných projektov, pretože to tejto skupiny patrí i naša práca. U väčšiny z nich by sa dal postup rozpoznávania rozdeliť do troch krokov:

1. **Farebná segmentácia**
2. **Vyhľadanie možných dopravných značiek**
3. **Rozpoznanie alebo zamietnutie dopravnej značky**

Rozdiel je iba v implementácii jednotlivých krokov. Pozrime sa teraz bližšie na jednotlivé body.

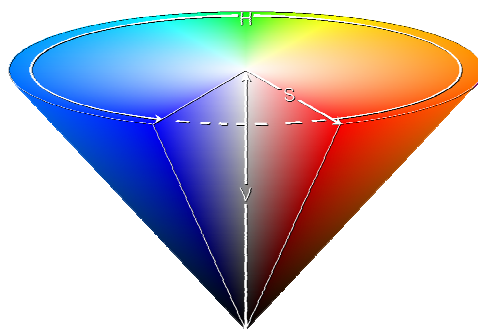
2.2. Farebná segmentácia

Toto je najdôležitejšia časť v procese nájdenia potenciálnych značiek (kandidátov) na vstupnej fotografii, pretože pokiaľ neuspějeme už tu, tak v ďalšom procese už značku nenájdeme. Pod pojmom *farebná segmentácia* v tomto prípade rozumieme získanie iba tých objektov zo vstupného obrázka, ktoré majú danú farbu. Ak nás zaujímajú objekty červenej farby, tak výsledkom takejto segmentácie budú iba tieto objekty. Takto získané objekty potom porovnáme s tvarom, ktorý hľadáme. Pre zjednodušenie nebudú na výstupe segmentácie dáta z pôvodného obrázka, ale iba akási bitová maska, v ktorej jednotlivé body budú indikovať objekt danej farby alebo

pozadie. Výhodou tohto postupu je, že do procesu vyhľadávania nám nevstupujú pre nás nezaujímavé objekty.

Problém je nájsť objekty danej farby. Pre človeka je zistiť, akú farbu má daný objekt, poväčšine jednoduchá úloha. Lež pre program môže byť ťažká a niekedy aj nemožná. Farebnú informáciu znehodnocuje napríklad kompresia fotografie, farba svetla, tieň okolitých objektov, počasie...

Farba sa v „počítačovom svete“ najčastejšie vyjadruje ako kombinácia troch farebných zložiek: *červenej, zelenej a modrej* (tzv. **RGB**-model). Tento model je oproti ostatným modelom lepší v tom, že vstupné dáta už nemusíme nijako prepočítavať a konvertovať (samozrejme iba ak vstupné dáta sú poskytnuté v RGB) a môžeme s nimi priamo pracovať. No ťažšie je už zistiť, či má daná farba potrebný odtieň. Oveľa jednoduchšie je použitie napríklad **HSV** farebného priestoru. Farba sa určuje (kóduje) pomocou troch základných vlastností: **H** (*Hue*) odtieň, **S** (*Saturation*) sýtosť, **V** (*Value*) jas.



Obrázok 4: Grafické znázornenie HSV modelu

zdroj: http://en.wikipedia.org/wiki/HSV_color_space

Odtieň prevažujúcej farby sa väčšinou označuje názvom farby. Podľa neho vieme, že ide o farbu červenú, žltú, modrú a pod. Udáva sa v uhlových stupňoch od 0° do 360°. **Sýtosť** farby určuje, ako je daný odtieň výrazný. Čím je sýtosť menšia, tým je výsledná farba viac nevýrazná, sivá. **Jas** určuje svetlosť farby, z fyzického hľadiska určuje koľko svetla daný objekt odráža, alebo by sa dalo zjednodušene povedať, že je to pomer čiernej farby k danému odtieňu. Jas a odtieň sa väčšinou udáva v percentách od 0 do 100 %.

Výhodou použitia HSV modelu je, že pri získavaní chcenej farby stačí vymedziť zaujímavé úseky z pomyselného kužľa. Nevýhodou tejto metódy je zdržanie spôsobené prevodom z RGB do HSV. Ďalšou nevýhodou použitia HSV je nestabilitosť odtieňu pri veľmi nízkych hodnotách jasú a sýtosti. Napriek týmto nevýhodám je pre jeho jednoduchosť použitia veľmi vhodný a používaný model.

2.3. Vyhľadanie možných dopravných značiek

Keď už máme označené objekty, ktoré sú pre nás zaujímavej farby, musíme zistiť, čo z nich je dopravná značka a čo nie. I toto môžeme robiť viacerými spôsobmi. Často sa využívajú rôzne vlastnosti týchto nájdených objektov. Príliš malé alebo príliš veľké objekty sú zamietnuté, podobne aj tie, ktoré sa nachádzajú tam, kde by sa dopravná značka nevyskytovala. Ďalej môžeme použiť napríklad *template matching*, čiže porovnávanie objektu so šablónou a zamietnuť objekty, ktoré sa s ňou dostatočne nezhodujú alebo využiť informácie o hranách alebo rohových bodoch. Ak zistíme, že by mohlo ísť o dopravnú značku, postúpime ju klasifikátoru, ktorý vyhodnotí, či naozaj ide o značku a rozpozná ju, alebo zamietne s tým, že sa o dopravnú značku nejedná.

2.4. Rozpoznanie dopravnej značky

Do tejto fázy sa zväčša dostanú už iba samotné dopravné značky. Takáto značka môže prejsť ešte predspracovaním, t.j. úpravou niektorých vlastností, ako je napríklad zvýšenie kontrastu, odstránenie nezaujímavých častí a pod. Z tohto výsledku následne „zmeriame“ niektoré pre nás zaujímavé vlastnosti – príznaky, ktoré sú rovnaké pre objekty z rovnakej skupiny a rôzne pre objekty z rôznych skupín. Medzi príznaky môžeme zaradiť rôzne vlastnosti objektu, ako je jeho veľkosť, plocha, svetlosť, kompaktnosť (pomer obvodu k obsahu), priemerná farba, ale môžeme použiť i rafinovanejšie príznaky, ako sú koeficienty rôznych transformácií, použiť niektoré z techník kódovania obrysov. Príznaky by sme mali vyberať tak, aby boli *diskriminabilné*, t.j. aby pre vstupy z rôznych skupín boli príznaky rôzne a *invariantné* (nezávislé) k niektorým vlastnostiam vstupu. V našom prípade ide o zmenu veľkosti, rotácie alebo jasú, teda aby pre podobné vstupy boli aj tieto príznaky podobné. Ako príznak sa v niektorých prípadoch používa rovno aj celý vstupný obrázok.

Ak máme už zvolené príznaky, postúpime ich *klasifikátoru*, ktorý určí do akej triedy objekt patrí. Klasifikovať môžeme niekoľkými spôsobmi. Spomenieme niektoré z nich, ktoré boli použité pri rozpoznávaní značiek

Jednou z metód je porovnávanie s uloženými vzormi. Vezmeme značku a porovnáme ju so všetkými uloženými vzormi (vektormi príznakov). Ako porovnávacie kritérium môže poslúžiť napríklad euklidovská vzdialenosť, iné metriky alebo komplikovanejšie spôsoby. Zo spôsobov, ktoré nám určia do akej triedy nám neznámy vzor patrí, môžeme spomenúť metódu *K najbližších susedov*, kde nájdeme k objektov, ktoré sa na neznámy vzor podobajú a výsledná trieda dostane triedu, ktorá je medzi nimi najviac zastúpená. Špeciálny prípad je, keď $k=1$. Vtedy výslednú triedu určuje vzor, ktorý je nášmu najviac podobný. Druhou metódou je v oblasti rozpoznávania veľmi používaná klasifikácia za pomoci neurónových sietí [8].

3. PREHL'AD PODOBNÝCH PRÁC

V tejto kapitole spomenieme niektoré práce zaoberajúce sa rozpoznávaním značiek.

Do prvej skupiny, definovanej v predchádzajúcej kapitole, patrí napríklad *Systém pro rozpoznávání dopravních značek (The Road Sign Recognition System (RS2))*[1] vyvíjaný na katedre aplikovanej matematiky Fakulty dopravní ČVUT. Je to komplexný systém, ktorý dokáže rozpoznávať značky i z čiernobielej fotografie. Pracuje v niekoľkých krokoch. Najprv podľa polohy vozovky zistí, kde by sa mala nachádzať dopravná značka, potom značku nájde a nakoniec značku klasifikuje do správnej skupiny. Autori uvádzajú úspešnosť správneho rozpoznávania až 98 %.

Z druhej skupiny spomenieme viac prác. Práca [2] kolektívu Jim Torresen, Jorgen W. Bakke a Lukas Sekanina, sa zaoberá rozpoznávaním rýchlostných dopravných značiek. Jej spôsob rozpoznávania sa asi najväčšmi podobá na spôsob, ktorý je využívaný v tejto práci. Rozpoznávanie je, podobne ako to bolo v predchádzajúcom prípade, rozdelené do viacerých krokov. Najprv sa zo vstupného obrázka vytvoria tri nové obrázky. Prvý obsahuje iba objekty červenej farby, druhý objekty, ktorých farba je čierna a posledný objekty biele. Červená farba je definovaná za pomoci RGB modelu a podielov jednotlivých farebných zložiek. V týchto obrázkoch sa za pomoci vopred známych šablón rôznych veľkostí nájde poloha značky. Z nájdenej značky je extrahovaná prvá číslica, ktorá je následne prevedená do binárneho obrázka, zmenšená na štandardnú veľkosť a klasifikovaná za pomoci neurónovej siete. Trénovacia množina bola získaná extrakciou značiek z dopravných situácií. Autori uvádzajú úspešnosť správneho rozpoznávania 91 %. Táto práca bola neskôr [3] rozšírená o sledovanie značky na videu.

Práce [4] a [5] sa od ostatných líšia v tom, ako získavajú trénovacie dáta. To je založené na degradovaní vlastností perfektnej dopravnej značky. Simuluje sa posun, rotácia, pridáva sa šum, posúva sa obdĺžnik pre vyrezanie piktogramu a pod. V [4] je na farebnú segmentáciu použité HSV, v [5] je to RGB a využíva rozdiely farebných zložiek. Práca rozpoznáva trojuholníkové dopravné značky, na ich vyhľadanie skontroluje vlastnosti vysegmentovaného objektu, ako je pomer strán a veľkosť a následne ešte overí existenciu rohov na správnych miestach. Príznamy pre

klasifikátor tvorí samotný obrázok, hodnoty z diskkrétnej cosínovej transformácie a SVD¹ vektor objektu. Na klasifikovanie použili niekoľko metód: neurónovú sieť a princíp K najbližších susedov. Uvedená je úspešnosť 78 %.

Posledná z prác [6], ktorú spomenieme, sa zaoberá rozpoznávaním jedinej značky, a to značky STOP. Na nájdenie červenej farby používa HSV. Značku pripustí ku klasifikovaniu, ak spĺňa danú veľkosť, stranový pomer a je dostatočne symetrická. Na klasifikovanie je použitá neurónová sieť, ktorej vstupom je obrázok o rozmeroch 30x30 a výstupom je informácia o tom, či je na vstupe STOP, alebo nie. Úspešnosť tohto projektu je 95 %.

¹ Singular Value Decomposition

4. VLASTNÁ IMPLEMENTÁCIA

Ako už bolo spomenuté v predchádzajúcej kapitole, programy pre rozpoznávanie dopravných značiek sme rozdelili do dvoch kategórií: na tie, čo na vstupe predpokladajú čiernobiele dáta a tie, čo na vstupe predpokladajú farebné obrázky. My sme si vybrali druhú možnosť a budeme vyhľadávať značky na fotografiách farebných.

Pre zjednodušenie práce s obrázkami sme sa rozhodli využiť knižnicu **OpenCV** [8], ktorá umožňuje spracovávanie obrázkov od úplne nízkej úrovne (prístup k bodom) až po spracovanie na najvyššej úrovni, ako je napr. rozpoznávanie ľudských tvárí. My sme sa rozhodli pre využitie „max. strednej cesty“ a používania funkcie na prácu s obrázkami a funkcie na vyhľadávanie hrán.

Náš program nie je výnimkou a jeho priebeh sa dá rozdeliť do troch základných krokov popísaných v minulej kapitole. V nasledujúcich odsekoch si popíšeme algoritmy a princípy našej implementácie.

4.1. Farebná segmentácia

V druhej kapitole sme uviedli, že budeme rozpoznávať iba značky zákazové. Tie sú kruhové a majú červený okraj. Budeme sa teda sústrediť na objekty červenej farby. Na zistenie, či ide o správnu farbu, sme sa rozhodli použiť HSV model. Aj keď má táto reprezentácia nevýhody, ktoré sme popísali vyššie, použili sme ho pre jeho jednoduchosť a dobré dosiahnuté výsledky.

Vstupný obrázok si teda prevedieme do systému HSV, ďalej si vytvoríme nový obrázok rovnakej veľkosti ako má vstup. Tento nový obrázok bude slúžiť na uloženie výsledkov farebnej segmentácie. Teraz bod po bode prechádzame vstup a podľa podmienky, ktorá bude uvedená neskôr, nastavujeme výstupný bod (ktorého súradnice odpovedajú súradniciam pôvodného obrázka). Ak je podmienka splnená, na výstupe bude farba biela, ináč bude na výstupe čierna farba.

Podmienky, ktoré musia splňovať jednotlivé farebné zložky sme po mnohých pokusoch a meraniach stanovili takto:

- *odtieň H* — musí byť v rozmedzí od 0° — 12° a od 336° — 360° ².
- *sýtosť S* — musí byť v intervale od 39 %—100 %.
- *jas V* — jeho hodnota musí byť aspoň 13 %.

4.1.1. Implementácia

Prechod obrázkom je ale veľmi časovo náročný a keďže budeme hľadať objekty, ktoré musia splňovať mimo iného i veľkostnú podmienku, nemuseli by sme vstupný obrázok prechádzať celý. Postačí nám, ak budeme prechádzať iba každý n -tý riadok (číslo n závisí na minimálnej veľkosti hľadanej značky). Kedykoľvek narazíme na červenú farbu, prejdeme okolie podrobnejšie.

Konkrétne v našom programe je to riešené tak, že si obrázok rozdelíme na štvorce, ktorých veľkosti odpovedajú nášmu stanovenému číslu n . Teraz prechádzame obrázok, každý n -tý riadok. Ak narazíme na červenú farbu, tak prejdeme podrobne štvorček, v ktorom sa aktuálne nachádzame, ale i štvorčeky okolité. Podobne, keď v okolitom štvorčeku nájdeme červenú farbu, prejdeme i jeho susedov. Týmto postupom zabezpečíme, že nájdeme vo vstupnom obrázku všetky objekty, ktoré majú výšku aspoň n .



Obrázok 6: Obrázok určený na spracovanie



Obrázok 6: Ukazuje priebeh farebnej segmentácie. Prechádzali sme po sivých riadkoch. Pokiaľ sme natrafili na červenú farbu, prešli sme podrobne okolité (svetlosivé) štvorce

² Toto rozdelenie do dvoch intervalov je spôsobené polohou „červenej farby“, ktorá sa nachádza v okolí nuly.

Na získaný výsledný obrázok aplikujeme ešte takzvanú dilatáciu, má za úlohu spojiť pri sebe ležiace objekty. Dilatácia je sčítanie každého bodu obrázka s danou maskou. Teda čierny pixel na súradniciach x a y , bude mať po dilatácii hodnotu $\max(x + xm, y + ym)$, kde xm, ym sú body v dilatačnej maske. Príklad dilatácie môžeme vidieť na nasledujúcom obrázku (v našom prípade sme použili masku vyplnenú maticu o rozmeroch 2×2):



Obrázok 7: Naľavo je originál, napravo je výsledok po použití dilatácie

4.2. Nájdenie značiek

Obrázok vygenerovaný predchádzajúcou fázou obsahuje iba objekty, ktoré majú červenú farbu. V tomto obrázku budeme teraz hľadať dopravné značky, v našom prípade sa jedná o značky zákazové, takže budeme hľadať kruh. Na obrázok spustíme funkciu knižnice *OpenCV*, ktorá nám v ňom nájde obrysy. Postupne prejdeme všetky z nich, každú aproximujeme mnohoúhelníkom a z ďalšieho spracovania vylúčime všetky objekty, ktoré nespĺňajú niektorú nami stanovenú podmienku.

Prvá podmienka je na **počet vrcholov**. Vylúčime objekty, ktoré ich majú menej ako 4. Potom testujeme **šírku objektu**, tá musí byť v našom prípade aspoň 20px. Posledný test, ktorý musí daný objekt spĺňať, je **stranový pomer** obkolesujúceho obdĺžnika, ktorý má strany rovnobežné so súradnicovým systémom. Podiel šírky k výške musí byť v rozmedzí 0,7—1,2 čo zodpovedá uhlu natočenia max 45° okolo stojanu značky a max. 33° okolo osi rovnobežnej s horizontom.

Ak objekt splní všetky tri uvedené vlastnosti, pokračujeme v jeho ďalšom spracovaní. Objekt porovnáme bod po bode s dvoma vopred pripravenými šablónami

(*templates matching*) a zistíme, na koľko percent sa s nimi podobajú a vylúčime objekty, ktoré sa zo žiadnou šablónou nezhodujú viac ako na 75 %. Šablóny, s ktorými porovnáваме, môžeme vidieť na nasledujúcom obrázku:



Obrázok 8: Šablóny, s ktorými sa musí objekt zhodovať, aby sme ho označili za dopravnú značku.

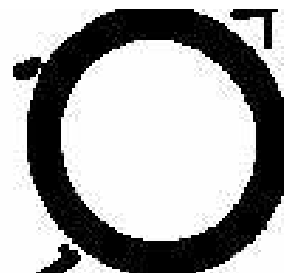
4.3. Rozpoznávanie

V predchádzajúcom bode sme vylúčili objekty, ktoré nespĺňali vlastnosti dopravných značiek, alebo sa na dopravnú značku dostatočne nepodobali. Tie objekty, ktoré nám ostali, prejdú ešte menším predspracovaním a následne budú rozpoznané.

4.3.1. Predspracovanie

Z pôvodného obrázka a z výsledku farebnej segmentácie (ďalej budeme označovať skráteno ako *červený obrázok*) si vyrežeme obdĺžniky odpovedajúce obkolesujúcim obdĺžnikom vyhovujúcich objektov. Výsledok môžeme vidieť na obrázku č. 9.

Prvá fáza predspracovania spočíva v čo najtesnejšom vyrezaní samotnej značky. Pretože do okolia značky môžu zasahovať objekty podobnej farby, nemusela by byť následná extrakcia piktogramu presná. Tesnejšie vyrezanie zaistíme odstránením tých riadkov a stĺpcov z každej strany, ktoré nebudú obsahovať aspoň štvrtinu červených bodov. Výsledok vidíme na obrázku č. 10.



Obrázok 9: Vľavo je vyrezaný obrázok z pôvodného obrázku a vpravo je výrez z červeneho obrázku.



Obrázok 10: Výsledný výrez bez zbytočných okrajov.

Tento výrez následne prevedieme do sivo-tónového a zmeníme jeho rozlíšenie na 32×32px. Nasleduje proces rozpoznávania toho, o akú značku ide.

4.3.2. Rozpoznanie značiek zákaz zastavenia, státia a vjazdu

Tieto značky sa líšia od ostatných v tom, že neobsahujú žiaden piktogram. Značka *zákaz vjazdu*³ obsahuje iba bielu plochu a značky *zákaz státia* a *zákaz zastavenia*, na rozdiel od ostatných zákazových značiek, obsahujú modrú farbu. Preto musí rozpoznanie týchto značiek prebiehať iným spôsobom ako u ostatných. Vo farebnom i čiernobielym obrázku sa sústredíme na štvorec šírky 16px začínajúci na súradniciach (8, 8).

Zákaz vjazdu

Táto značka, ako sme si už povedali, neobsahuje žiaden piktogram. Je potrebné teda overiť, že jej výplň je tvorená iba bielou farbou. To zaistíme dvoma spôsobmi.

V prvom rade zrátame *rozptyl* farby v čiernobielym obrázku. Zrátame ho ako súčet druhých mocnín rozdielu hodnoty v aktuálnom bode a priemernej farby vo

³ Pod značkou zákaz vjazdu si predstavujeme dopravnú značku Zákaz vjazdu všetkých vozidiel v oboch smeroch

vymedzenom štvorci podelený počtom prejdených bodov. Prípustný je iba rozptyl menší ako 50. Druhá podmienka vylúči objekty, ktoré síce sú jednofarebné (ich výplň má malý farebný rozptyl), ale musíme ešte overiť, či ide o bielu farbu. Spočítali sme si ešte počet bielych bodov. *Biely bod* si definujeme tak, že *sýtosť* je maximálne 18 % a *jas* aspoň 11 %, na hodnote *odtieňa* nám teraz nezáleží. Ak je *bielych bodov* viac ako 70 %, tak značku označíme ako *zákaz vjazdu*.

Zákaz zastavenia a zákaz státia

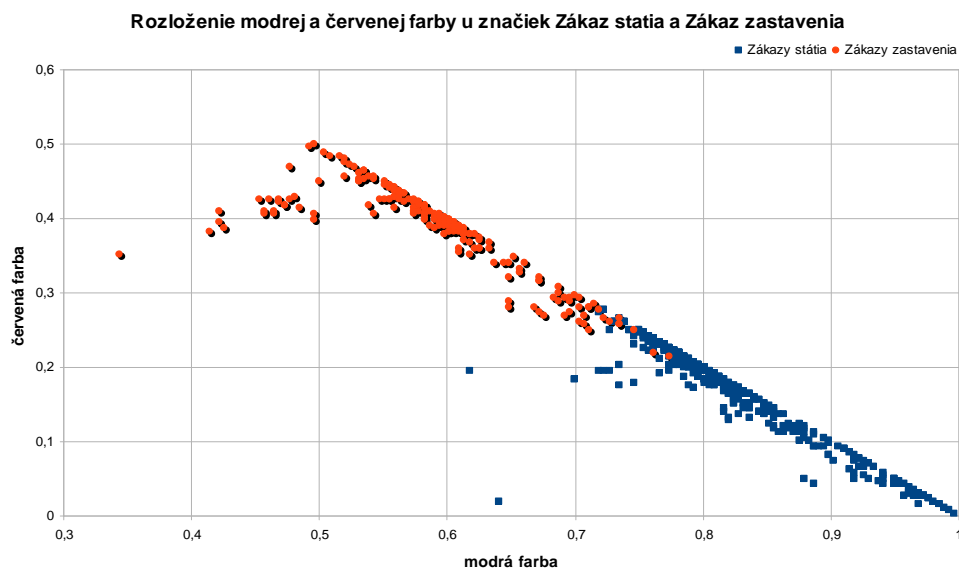
Rozpoznanie, či ide o tieto dve značky, prebieha podobne ako to bolo v predchádzajúcom prípade, teda iba na základe farieb. Počítame si počty *červených* a *modrých bodov*. Modrý bod si definujeme opäť podľa hodnôt HSV tak, že *odtieň* je v rozmedzí od 192° do 300° , pre červený bod musí byť v intervale od 300° do 372° ⁴. *Sýtosť* u oboch je aspoň 40 % a *jas* aspoň 13 %. Pozrime sa na obrázok č. 11, ktorý ukazuje aké percentuálne zastúpenie majú červené a modré body u jednotlivých značiek.

Z grafu je vidno, že červená farba tvorí doplnok k modrej (toto sme vlastne aj očakávali.) Rozhodli sme sa, že tieto dve kategórie značiek budeme oddeľovať porovnávaním pomeru modrej farby.

Značku označíme ako *zákaz zastavenia*, ak je v nej aspoň 34 % a maximálne 72 % modrej farby a vo zvyšných bodoch je aspoň polovica červených. Značku označíme za *zákaz státia*, ak má aspoň 72 % modrých bodov a vo zvyšku je aspoň polovica červených.

Toto rozdelenie nám zaistilo správne klasifikovanie väčšiny týchto značiek. Môžu ale nastať i prípady, kde náš postup zlyhal. Sú to najmä prípady rozmazania značky, kde červený pás splynul s modrým okolím.

⁴ Do 372° myslíme ako do 360° zjednotené s intervalom $0-12^\circ$



Obrázok 11: Ukazuje rozloženie modrej a červenej farby u značiek zákazu státia a zastavenia.

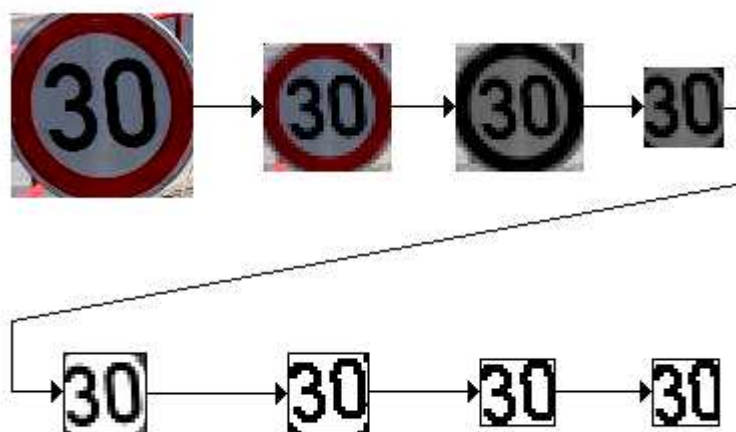
4.3.3. Rozpoznanie ostatných značiek

Teraz už vieme rozpoznávať značky zákazov státí, zastavení a vjazdov. Tieto značky boli od ostatných rozdielne vo farebnom stvárnení, a preto sa museli spracovať samostatne. Ostatné zákazové značky sú vo svojom prevedení viac jednotné. U týchto značiek nás bude zaujímať ich piktogram.

Predspracovanie

Piktogram by sa mal v prípade, že máme značku dobre vyrezanú, nachádzať vo štvorci veľkosti 20px, začínajúci na pozícii (6, 6)⁵. Zamerajme sa iba na tento výrez, ktorý prevedieme do čiernobieleho obrázka, ktorý farebne znormalizujeme. Následne ho prevedieme do dvojfarebného obrázka, kde čierna farba je pôvodné biele pozadie, a bielou bude vykreslený piktogram. Z každej strany odstránime tie riadky a stĺpce, ktoré obsahujú menej ako dva biele body a tie body, ktoré patria červenému orámovaniu značky. Výsledok spracovania normalizujeme na veľkosť 16×16px. Celý proces predspracovania je vidieť na obrázku 12.

⁵ Pripomeňme, že značka má už normalizovanú veľkosť 32×32px.



Obrázok 12 Priebeh predspracovania obrázka a vyrezania piktogramu.

Rozpoznanie

Na rozpoznanie o akú značku ide sme sa rozhodli použiť doprednú neurónovú sieť. Ako algoritmus učenia sme si zvolili backpropagation. Má 256 vstupných, 80 skrytých a 6 výstupných neurónov (pre každú značku jeden). Na vstup naučenej neurónovej siete privedieme celý piktogram, ktorý sme získali v predchádzajúcom procese predspracovania a pozrieme sa na výstup neurónovej siete. Rozhodneme sa takto. Ak výstup jedného z neurónu je aspoň 0,8 a ostatné z nich sú pod 0,4, vstupnú značku zaradíme do skupiny reprezentovanej práve neurónom, ktorého výstup je väčší ako 0,8. V ostatných prípadoch označíme vstupnú dopravnú značku ako neznámu.

4.4. Popis tried programu

V tejto podkapitole si stručne popíšeme jednotlivé triedy použité v programe.

Trieda ImageProcessor

Je to základná trieda, ktorá má za úlohu otvoriť vstupný obrázok, poprípade video a predkladať vstupné obrázky, alebo jednotlivé snímky videa triede *SignRecognizer*.

Trieda SignRecognizer

Táto trieda sa stará o rozpoznanie dopravných značiek na vstupnom obrázku. V konštruktore sú vytvorené šablóny na porovnávanie a taktiež pokiaľ je potreba

načítaný súbor s polohami dopravných značiek. Činnosť triedy zahŕňa celý proces spracovania, čiže obsahuje farebnú segmentáciu, nájdenie značky a rozpoznanie značky. Na rozpoznanie využíva triedu *ImageList*.

Trieda ImageList

Táto trieda slúži ako kontajner na ukladanie nájdených značiek. Pri ukladaní ju rovno rozpoznáme. Inicializuje sa v nej neurónová sieť. Rozpoznanie zahŕňa aj predspracovanie výrezu popísané vyššie.

5. VÝSLEDKY VLASTNEJ IMPLEMENTÁCIE

Zadaním práce bolo vytvoriť program, ktorý bude pokiaľ možno v reálnom čase a úspešne rozpoznávať dopravné značky. Poďme sa pozrieť ako sme túto úlohu splnili.

5.1. Rýchlosť

Program bol vyvíjaný na počítači s procesorom Intel Pentium Dual-Core s taktom 1.6GHz a s pamäťou Ram 1024 MB. Nasledujúce údaje o rýchlosti sa budú vzťahovať k tejto konfigurácii.

Rýchlosť spracovania je závislá na veľkosti vstupnej fotografie. Pre rozhodnutie, akú veľkosť použiť, sme sa rozhodli použiť nasledovné meranie: spočítali sme si časy, ktoré potrebujeme na farebnú segmentáciu⁶. Tento čas je minimálny na spracovanie fotky, t.j. pokiaľ by sa na obrázku ďalej značky nehľadali ani nerozpoznávali.

Veľkosť obrázka	Čas potrebný na farebnú segmentáciu	Max. dosiahnuteľné fps ⁷
640×480	17 ms	58
800×600	26 ms	38
1024×748	41 ms	24

Tabuľka 1: Zobrazuje časy potrebné k farebnej segmentácii pre rôzne rozlíšenia vstupu

Pre spracovanie v reálnom čase je potrebné dosiahnuť aspoň 25fps. Spracovanie obrázku o rozmeroch 1024×748 je tesne pod touto hranicou. Nesmieme však zabudnúť na čas, ktorý ešte strávime vyhľadaním značky, preto tento rozmer obrázku vylúčim. Rozmer 800×600 by síce mohol vyhovovať a obrázky by sme spracovávali dostatočne rýchlo, ale použité snímacie zariadenie neponúkalo výstupy v tomto rozlíšení. Preto sme sa rozhodli zamerať na obrázky s rozlíšením 640×480.

⁶ Použili sme efektívnejšie prechádzanie popísané v kapitole 4.1.1

⁷ Fps: frames per second, počet obrázkov spracovaných za jednu sekundu.

Menšie obrázky by určite urýchlili spracovanie, ale pre ich nižšiu kvalitu by klesla úspešnosť rozpoznávania.

Celkový čas potrebný na celý proces rozpoznávania sa skladá z časov potrebných na:

- Farebnú segmentáciu
- Nájdenie značiek
- Rozpoznanie značiek
- Zobrazenie výsledkov

Čas potrebný na farebnú segmentáciu sme už uviedli. Ako dlho bude trvať nájdenie značky a jej rozpoznanie je závislé od počtu značiek na scéne. Nasledujúca tabuľka zobrazuje priemerné namerané časy

Farebná segmentácia:	17ms
Nájdenie značiek:	2—5ms
Rozpoznanie značiek:	2—5ms
Celkový čas pri zobrazovaní výsledku:	31ms
Celkový čas pri nezobrazovaní výsledku:	25ms

Tabuľka 2: Zobrazuje časy potrebné na jednotlivé fázy spracovania.

Ako môžeme vidieť, cieľ — rozpoznávať dopravné značky v reálnom čase, sa nám podarilo splniť.

5.2. Úspešnosť

Úspešnosť implementácie rozdelíme na úspešnosť správneho označenia značky a úspešnosť následného správneho rozpoznania konkrétnej značky.

5.2.1. Úspešnosť vyhľadávania

Na testovanie tohto faktoru sme použili 382 obrázkov dopravnej situácie. Na nich sme manuálne vyznačili 372 dopravných značiek. Tento zoznam sme poskytli nášmu programu, ktorý nám povedal, ktoré značky zo zoznamu neoznačil a naopak,

koľko označil takých, ktoré sa v zozname nenachádzali. Program nevyznačil 62 z nich. *Úspešnosť vyhľadania je teda asi 83%.*

Nerpoznanie značky bolo väčšinou spôsobené rozmazaním značky červeným objektom, ktorý zasahoval do dopravnej značky, alebo príliš bledou farbou červeného orámovania.

Na úspešnosť vyhľadania značky sa dá pozerat' aj iným spôsobom. Keďže snímky boli zhotovené sekvenčne počas jazdy vozidla, alebo by to boli jednotlivé snímky videa, nebudeme počítat' značky v obrázkoch, ale značky v priestore (teda ak sme značku nenašli na jednom obrázku, ale tá istá značka sa objaví na ďalšom a tú úspešne zaregistrujeme, budeme to považovať za úspešné nájdenie značky). V popísanom prípade bude úspešnosť väčšia, ale presné čísla sme nepočítali.

5.2.2. Úspešnosť rozpoznávania

Úspešnosť rozpoznania sme merali ako úspešnosť správneho rozhodnutia naučenej neurónovej siete na testovacej množine. Testovaciu množinu tvorili značky, ktoré neboli použité pri tréovaní a obsahovala i značky zo skupín, ktoré sme neurónovú sieť neučili. Počítali sme úspešnosť správnej klasifikácie značky, ale i úspešnosť zamietnutia neznámej značky.

Uvedme značky, ktoré sme sieť naučili a ich počty v tréovanej a testovacej množine a úspešnosť ich rozpoznania.

Značka	Počet trén. vzorov	Počet test. vzorov	Úspešne rozpoznané	Percentuálna úspešnosť
Zn. č. 1	124	28	28	100 %
Zn. č. 2	26	6	6	100 %
Zn. č. 3	121	53	49	92 %
Zn. č. 4	141	45	44	98 %
Zn. č. 5	49	8	8	100 %
Zn. č. 6	18	5	4	80 %
Neznáme	-	207	157 ⁸	75 %
Celkom známe	479	145	140	96 %
Celkom všetky	479	352	297	84 %

Tabuľka 3: Zobrazenie úspešnosti rozpoznávania.

⁸ Pod úspešným rozpoznáním myslíme, ak sieť označila značku za neznámu.



Obrázok 13: Obrázok znázorňuje dopravné značky, ktoré sme naučili program rozpoznávať

Značky 7, 8 a 9 boli spracovávané iným spôsobom a úspešnosť ich správneho rozpoznania bola niečo nad 95%. Značiek, ktoré neboli z tejto skupiny, ale náš program ich rozpoznal ako nejakú z nich, bolo 5. Boli to najmä prípady, keď bol piktogram veľmi svetlý a náš program označil značku za zákaz vjazdu.

Ako vidíme, úspešnosť rozpoznania značiek, ktoré sme mali naučené, je pomerne vysoká, avšak značky, ktoré neurónová sieť nevidela, označovala v niektorých prípadoch za známe značky. Tomu by sa dalo možno zabrániť, ak by sme sieť naučili viac druhov značiek.

ZÁVER

Úspešnosť nášho programu nie je síce veľmi vysoká, ale i tak sa približuje k výsledkom podobných projektov. Možnosti zlepšenia vidíme napríklad v získaní lepších príznakov z ktorých je značka rozpoznávaná, alebo vyskúšaní iných klasifikátorov. Zlepšeniu detekcie by mohlo pomôcť i vyššie rozlíšenie vstupov, prípadne by sme mohli použiť dva snímacie zariadenia, jedno s nízkym a jedno s vysokým rozlíšením, kde by to druhé bolo použité až ak by sme na prvom detekovali značku. So zvyšovaním rozlíšenia však stúpajú nároky na výpočtovú kapacitu počítača. Ďalší vývoj vidíme v rozšírení práce s videom a možnosť dáta získavať i z kamery. Vylepšiť by sa dalo i užívateľské rozhranie, napr. pridaním hlasového výstupu.

Na záver si dovoľíme tvrdiť, že práca splnila očakávania, ktoré sme si pri jej zadaní stanovili. Dokáže značky vyhľadávať a rozpoznávať v reálnom čase.

UŽÍVATEĽSKÁ DOKUMENTÁCIA

V tejto kapitole popíšeme použitie nášho programu a detailne popíšeme všetky jeho parametre.

Hlavný program

Hlavný program, ktorý slúži na rozpoznanie značiek v danom obrázku alebo videu, sa spúšťa príkazom:

```
znacky.exe súbor [parametre]
```

Kde *súbor* je vstupný súbor na spracovanie. Súbory, ktoré sme schopní spracovať, môžu byť buď *obrázky* alebo *videá* typu *avi*. Parametre majú nasledujúcu syntax a význam, všetky z nich sú nepovinné:

```
[-type=video | -t=video] [-type=image | t=video]
```

```
[-saveroi | -s] [-showtimes | -T] [-noshowoutput | -n]
```

```
[-showoriginal | -o] [-showred | -r] [-showfps | -f]
```

```
[-norecognize | -N] [-saveoutput | -S] [-nnfile=nazov]
```

```
[-signaturefile=nazov] [-signlist=nazov]
```

Teraz si popíšeme ich význam podrobnejšie (ďalej uvádzame iba dlhé verzie prepínačov).

- **-type=video, -type=image:** Tento prepínač explicitne určuje typ vstupných dát. Pokiaľ nie je použitý, program sa pokúsi typ vstupných zistiť sám.
- **-saveroi:** Tento prepínač sa používa v prípade, že chceme ukladať nájdené a vyrezané značky. Súbory sa budú ukladať v rovnakom priečinku ako vstupný súbor a ich názov je tvorený pôvodným názvom, ku ktorému sa pridá jednoznačné číslo a prípona *.bmp*. Pokiaľ sa budú značky aj rozpoznávať, uloží sa aj dvojfarebný obrázok, ktorý sa prekladá na vstup neurónovej siete. Tento súbor bude mať príponu *.final.bmp*. Teda ak vstupný súbor je *Obrázky\znacka.jpg*, uložia sa súbory *Obrázky\znacka_0.bmp* a *Obrázky\znacka_0.final.bmp*.

- **–showtimes:** Tento prepínač sa uplatní, iba ak na vstup privedieme video. Po jeho použití sa bude po každom dvadsiatom snímku videa na štandardný výstup vypisovať údaj o počte spracovaných snímkach za sekundu.
- **–noshowoutput:** Ak je použitý tento prepínač, tak sa nezobrazí výsledné okno z označenými rozpoznávanými a nájdenými značkami.
- **–showoriginal:** Zobrazí sa aj pôvodný obrázok alebo pôvodné video.
- **–showred:** Použitím tohto prepínača umožníme programu zobraziť výsledok farebnej segmentácie.
- **–norecognize:** Keď je použitý tento prepínač, značka sa nebude rozpoznávať, ale iba sa označí.
- **–saveoutput:** Tento parameter programu určuje, či sa má výsledný spracovaný obrázok uložiť do súboru a uplatní sa iba ak na vstup programu privedieme obrázok. Názov výsledného súboru je tvorený pôvodným názvom a príponou *.out.jpg*.
- **-nnfile=súbor:** Parameter *súbor* tohto prepínača určuje cestu k vopred vyrobenej a uloženej neurónovej sieti. Pokiaľ prepínač neje použitý, neurónová sieť sa hľadá v súbore *ck.xml*.
- **-signnamefile=súbor:** Parameter *súbor* tohto prepínača určuje cestu k súboru, ktorý obsahuje názvy značiek ktoré sme schopný rozpoznáť. Jeho štruktúra je nasledujúca. Každý riadok obsahuje názov jednej značky. Prvé štyri riadky sú rezervované pre názvy nasledujúcich značiek. Prvý riadok je názov *neznámej značky* (momentálne nie je nikde použité), druhý riadok je názov značky *zákaz zastavenia*, tretí je názov značky *zákaz vjazdu* a štvrtý obsahuje názov značky *zákaz státia*. Na ďalších riadkoch sa nachádzajú názvy značiek v tom poradí ako sme ich učili neurónovú sieť. Teda na piatom riadku bude názov tej značky, ktorej sme prideliť 0-tý výstupný neurón, na šiestom tej, ktorej sme prideliť 1. výstupný atd.
- **-signlist=súbor:** Parameter *súbor* tohto prepínača určuje cestu k súboru, ktorý obsahuje zoznam obrázkov a polohy značiek na nich. Ak je prepínač použitý, hľadá sa vstupný obrázok v zozname a pokiaľ je v ňom nájdený,

vypíšu sa informácie u nenájdenej značkách a značkách navyše, teda o takých ktoré program našiel ale v zozname neboli uvedené. Formát súboru je nasledovný, na každom riadku je cesta a názov k obrázku x-ová a y-ová súradnica značky. Ak je na jednom obrázku viac značiek, musia byť uvedené na zvláštnych riadkoch.

Program na učenie neurónovej siete

Na naučenie sa neurónovej siete sme vytvorili vedľajší program. Ten má za úlohu spracovať tréningové a testovacie dáta a naučiť prípadne otestovať úspešnosť neurónovej siete. Využili sme doprednú neurónovú sieť a učili ju za pomoci algoritmu backpropagation. Spustenie vyzerá nasledovne:

```
neurony.exe      subor.xml      veľkosť_vstupov      počet_výstupov  
počet_skrytých_neurónov [testuj]
```

Kde **súbor.xml** je názov výstupného súboru, do ktorého bude zapísaná naučená neurónová sieť, prípadne z ktorého bude načítaná v prípade testovania.

Veľkosť_vstupov určuje veľkosť obrázkov, ktoré privádzame na vstup v pixloch. Počet vstupných neurónov bude druhá mocnina tejto hodnoty

Počet výstupov udáva počet neurónov vo výstupnej vrstve, čiže počet tried, ktoré budeme schopní rozpoznať.

Počet skrytých neurónov určuje, ako už názov napovedá, počtu neurónov v skrytej vrstve. Momentálne program podporuje iba jednu skrytú vrstvu.

Testuj ak je použitý i tento parameter, sieť sa nebude učiť, ale načíta neurónovú sieť zo súboru a otestuje ju na vstupných dátach.

Program k svojej správnej funkcii potrebuje ešte dva súbory, jeden s názvom **train.txt**, druhý s názvom **test.txt**. Prvý z nich sa použije ak sieť učíme, druhý sa použije, ak použijeme parameter **testuj**. Tieto súbory majú takýto formát: Každý riadok obsahuje cestu k obrázku, medzeru a číslo výstupného neurónu, ktorý má byť aktívny, v prípade súboru test.txt, môže byť toto číslo -1, ktoré označuje vzor, ktorý nebol sieť naučený.

LITERATÚRA

- [1] Systém pro rozpoznávání dopravních značek (*The Road Sign Recognition System (RS2)*); <http://euler.fd.cvut.cz/projekty/rs2/>
- [2] J. Torresen, J.W. Bakke, and L. Sekanina; *Efficient recognition of speed limit signs*; In Proc. of 7th IEEE International Conference on Intelligent Transportation Systems; 2004; <http://home.ifi.uio.no/~jimtoer/>
- [3] Jim Torresen, Jorgen W. Bakke and Yizhuang Yang. *A camera based speed limit sign recognition system*; 2006; <http://home.ifi.uio.no/~jimtoer/>
- [4] Hiroyuki Ishida, Tomokazu Takahashi, Ichiro Ide, Yoshito Mekada, and Hiroshi Murase. *Identification of degraded traffic sign symbols by a generative learning method*. Graduate School of Information Science, Nagoya University Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601, Japan; 2006; <http://murase.m.is.nagoya-u.ac.jp/index-en.html>
- [5] Hsiu-Ming Yang, Chao-Lin Liu, Kun-Hao Liu, and Shang-Ming; *Traffic Sign Recognition in Disturbing Environments*; Huang Department of Computer Science, National Chengchi University Taipei 11605, Taiwan; 2003; <http://www.springerlink.com/content/ngvgt3ja2m6w8ccb/>
- [6] Henry X. Liu and Bin Ran; *Vision-Based Stop Sign Detection and Recognition System for Intelligent Vehicle*; Department of Civil and Environmental Engineering University of Wisconsin at Madison; 2001; <http://www.ce.umn.edu/~liu/>
- [7] Open source Computer Vision Library (OpenCV); <http://opencvlibrary.sourceforge.net/>
- [8] Neurónové siete na http://en.wikipedia.org/wiki/Artificial_neural_network