Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Návrh vhodné topologie neuronové sítě pro rozpoznání objektů v trávních porostech

Tomáš Polívka

© 2016 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Tomáš Polívka

Informatika

Název práce

Návrh vhodné topologie neuronové sítě pro rozpoznání objektů v trávních porostech

Název anglicky

The suitable neural network topology design for object recognition in the grass.

Cíle práce

Cílem bakalářské práce je prostudovat a navrhnout vhodnou architekturu perceptronové neuronové sítě určené k rozpoznávání základních tvarů (kruhů, čtverců, obdélníků) v trávních porostech. Fotografie travního porostu pořízené mobilním telefonem či robo-optickou jednotkou budou předkládány neuronové síti. Na základě experimentů s transformací obrazu a současně s velikostí vstupní a skryté vrstvy vzhledem k rychlosti učení sítě, za předpokladu dodržení maximální možné schopnosti rozpoznání tvarů, bude navržena optimální vstupní matice neuronů a topologie skryté vrstvy. Jako nástroj pro analýzu bude použit Neuroph (http://neuroph.sourceforge.net).

Metodika

Rešeršní část BP bude založena na analýze odborných a vědeckých dokumentů (zejména monografií) a následně budou získané poznatky synteticky využity k návrhové části. V té se zaměře na možnosti využití neuronových sítí k rozpoznání objektů, zhodnocení jejich možností a oblasti použití. Na základě rešerše navrhněte několik variant topologí neuronových sítí. Experimentujte s množstvím neuronů ve skryté vrstvě a s možnostmi dynamického přeskupování neuronů v síti. Pro experimenty využívejte nástroje Neuroph. Na základě zjištěných znalostí navrhněte vhodnou topologii sítě, doporučte zda je nástroj Neuroph vhodný pro vývoj neuronových sítí požadovaného typu. Výsledky analýzy publikujte.

Doporučený rozsah práce

40 str

Klíčová slova

Percepronová neuronová síť, Rozpoznávání objektů, Neuroph, Roboticko – optická jednotka

---

Doporučené zdroje informací

F. Y. Shih. Image Processing and Pattern Recognition: Fundamentals and Techniques. Wiley-IEEE Press, 2010 ISBN: 978-0-470-40461-4

Předběžný termín obhajoby

2015/16 LS – PEF

Vedoucí práce

Ing. Josef Pavlíček, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 22. 2. 2016

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 22. 2. 2016

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 13. 03. 2016

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Návrh vhodné topologie neuronové sítě pro rozpoznání objektů v travních porostech" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 14.3.2016 _____

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce Ing. Josefu Pavlíčkovi, Ph.D. za odborné konzultace a ochotu vést tuto práci. Také chci poděkovat svým rodičům, kteří mě vytrvale podporují po dobu mých studií. Děkuji.

Návrh vhodné topologie neuronové sítě pro rozpoznání objektů v travních porostech

The siutable neural network topology design for object recognition in the grass

Souhrn

Tato bakalářská práce popisuje testování umělých neuronových sítí v závislosti na učebním souboru dat pro rozpoznání základních tvarů v travním porostu. Experimentováním s různými topologiemi sítí v programu Neuroph Studio byly zaznamenány hodnoty o úspěšnosti rozpoznání. Testování probíhalo na třech učebních množinách základních tvarů. Ověření schopnosti učení sítí bylo zkoušeno pomocí fotografií čtyř základních tvarů pořízených v trávě (čtverec, obdélník, kruh a trojúhelník). Při vyhodnocování nejvhodnější topologie byla definována dvě hlavní kritéria – počet úspěšných rozpoznání tvarů a rozdělení podle rozpoznaných objektů.

Teoretická část práce popisuje funkcionalitu a vlastnosti neuronových sítí, rozpoznání obrazu pomocí detekce hran a existující aplikace pro rozpoznání druhů rostlin.

Klíčová slova

Perceptronová neuronová síť, Rozpoznávání objektů, Neuroph, Roboticko – optická jednotka

Summary

This thesis describes testing of artificial neural network depending on learning set for object recognition in grass. Values of experimenting with different topologies in Neuroph Studio were recorded. Testing was conducted on three sets of learning basic shapes. Verification of the ability of learning networks were tested using four basic shapes captured in the grass (square, rectangle, circle and triangle). When evaluating the most appropriate topology was defined by two main criteria - number of successful recognition of shapes and distribution by recognized shapes.

The theoretical part describes the functionality and features of neural networks, image recognition using edge detection and existing applications for recognition of plant species.

Key words

peceptron neural network, object recognition, Neuroph, robo – optical unit

Obsah

1 Úvod 12
2 Cíl práce a metodika 13
2.1 Cíl práce 13
2.2 Metodika 13
3 Teoretická východiska 14
3.1 Již existující nástroje pro rozpoznávání rostlin 14
LeafSnap 14
FlowerChecker 15
3.2 Neuronové sítě - teorie 16
Model neuronu 16
Aktivační funkce 17
Topologie sítě 18
Možné změny během činnosti sítě 18
Aktivní režim 19
Adaptační režim 19
Organizační režim 19
Vícevrstvá neuronová síť 20
Algoritmus zpětného šíření chyby 21
3.3 Teorie detekce hran – rozpoznání obrazu 21
Hrana 22
Hranový bod 22
Hranové detektory založené na hledání maxim prvních derivací 23
Konvoluční masky 23
Hranové detektory založené na druhé derivaci 25
Využití pro práci 27
3.4 Neuroph 28

4 Vlastní práce 29
4.1 Vytvoření a natrénování neuronové sítě pomocí aplikace Neuroph Studio 29
Vytvoření trénovací a testovací množiny dat základních tvarů 29
4.2 Vytvoření projektu a natrénování sítě 31
4.3 Zhodnocení výsledků 36
4.4 Testování topologie sítě v závislosti na trénovací množině 37
Testovací množina dat 37
Trénovací množiny dat 37
4.4.1.1 Množina A 37
4.4.1.2 Množina B 38
4.4.1.3 Množina C 38
Postup testování 39
Trénovací množina A 40
Trénovací množina B 41
Trénovací množina C 42
5 Zhodnocení výsledků 44
5.1 Výsledky jednotlivých učebních souborů 44
Množina A 44
Množina B 44
Množina C 44
5.2 Celkové výsledky sítí 45
Podle počtu rozpoznaných objektů 45
Seřazení podle rozpoznaných tvarů 45
5.3 Shrnutí výsledků 46
6 Závěr 47
7 Seznam použitých zdrojů 48

Seznam obrázků

Obrázek 1 - Ukázka aplikace LeafSnap 14
Obrázek 2- Ukázka aplikace FlowerChecker 15
Obrázek 3- Model neuronu 16
Obrázek 4 - Aktivační funkce neuronů 17
Obrázek 5 - Acyklická a cyklická architektura (v pravo) 18
Obrázek 6 - Vícevrstvá neuronová síť 20
Obrázek 7 - Profily v okolí hranových bodů 22
Obrázek 8- Ukázka první derivace hrany 23
Obrázek 9 - Zobrazení funkce druhé derivace 25
Obrázek 10 - Struktura aplikace Neuroph 28
Obrázek 11 - Trénovací množina dat 29
Obrázek 12 - Testovací množiny dat 30
Obrázek 13 - Vytvoření projektu 31
Obrázek 14 - Vytvoření neuronové sítě pro rozpoznání obrazu 31
Obrázek 15 – Výběr testovací množiny 32
Obrázek 16 - Výběr množiny pro nerozpoznání 33
Obrázek 17 - Parametry vstupních dat 33
Obrázek 18 - Parametry neuronové sítě 34
Obrázek 19 – Neuronová síť 34
Obrázek 20 – Graf celkové chybovosti učení sítě 35
Obrázek 21 - Foto tvarů v trávě 37
Obrázek 22 - Množina B 38
Obrázek 23 - Množina C 38

Seznam tabulek

Tabulka 1 - Nastavení parametrů 35
Tabulka 2 - Přehled přesností určení 36
Tabulka 3 - Parametry nastavení 39
Tabulka 4 - Výsledky měření A_1 40
Tabulka 5 - Výsledky měření A_2 40
Tabulka 6 - Výsledky měření A_3 41
Tabulka 7 - Výsledky měření A_4 41
Tabulka 8 - Výsledky měření B_1 41
Tabulka 9 - Výsledky měření B_2 42
Tabulka 10 - Výsledky měření B_3 42
Tabulka 11 - Výsledky měření B_4 42
Tabulka 12 - Výsledky měření C_1 42
Tabulka 13 - Výsledky měření C_2 43
Tabulka 14 - Výsledky měření C_3 43
Tabulka 15 - Výsledky měření C_4 43
Tabulka 16 - Souhrn měření množiny A 44
Tabulka 17 - Souhrn měření množiny B 44
Tabulka 18 - Souhrn měření množiny C 44
Tabulka 19 - Celkové zobrazení podle počtu rozpoznání 45
Tabulka 20 - Porovnání sítí B_4 a B_2 45
Tabulka 21- Rozpoznání trojúhelníku 46

1 Úvod

Tématem mé práce je navrhnout vhodnou topologii neuronové sítě pro rozpoznání základních objektů v travním porostu a experimentovat s různými topologiemi sítí. Poznatky získané experimenty by měly být dále využity pro rozpoznání rostlin. V přírodě se vyskytuje okolo 298000 (1) druhů rostlin. My laici rostliny rozpoznáváme podle různých aspektů jako je například stonek, plod, velikost rostliny, list, květ – když zkombinujeme tyto morfologické a jiné vlastnosti dohromady, tak si vytvoříme docela jasný úsudek o tom, co to je za rostlinu. Ovšem za předpokladu, že danou rostlinu známe. Pokud ne, můžeme si vzít k ruce například herbář a podle výrazných znaků rostlinu najít. A však zaznamenat všechny aspekty do jednoho je velmi složité. Pokud se zaměříme na listy, záleží na spoustě dalších vlastnostech. Začněme s tvarem listu. Tvar listu je velmi nepravidelný, takže i podle otočení, nebo naklonění listu si můžeme udělat špatný úsudek. Dalším problémem je, že listy se mění postupem času a nastávají nám tyto jevy – růst, stárnutí, změna barvy, usychání, možnost degenerace. A aby toho nebylo málo, je také problém s odlišnostmi v rámci jednoho druhu v závislosti na klimatických a půdních podmínkách. Můžeme říci, že každá rostlina je svým způsobem originální. Je proto velmi složité některé rostliny rozpoznat. Pro přípravu neuronových sítí na rozpoznávání rostlin je nejprve nutné začít s experimentováním se základními tvary, které zjednodušeně reprezentují tvary listů.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem bakalářské práce je prostudovat a navrhnout vhodnou architekturu perceptronové neuronové sítě určené k rozpoznávání základních tvarů (kruhů, čtverců, obdélníků) v trávních porostech. Tyto tvary zjednodušeně reprezentují tvary listů rostlin. Fotografie tvarů v travním porostu pořízené mobilním telefonem či robo-optickou jednotkou jsou předkládány neuronové síti. Na základě experimentů s transformací obrazu a současně s velikostí vstupní a skryté vrstvy vzhledem k rychlosti učení sítě, za předpokladu dodržení maximální možné schopnosti rozpoznání tvarů, je navržena optimální vstupní matice neuronů a topologie skryté vrstvy. Jako nástroj pro analýzu je použit Neuroph Studio.

2.2 Metodika

Rešeršní část práce je založena na analýze odborných a vědeckých dokumentů a následně získané poznatky jsou využity k návrhové části. Průzkum možností rozpoznání obrazů pomocí detekce hran a prostudování teorie neuronových sítí. Zaměření na možnosti využití neuronových sítí k rozpoznání objektů, zhodnocení jejich možností a oblasti použití. Na základě rešerše je navrhnuto několik variant topologií neuronových sítí. Experimentováním s množstvím neuronů ve skryté vrstvě a počtem skrytých vrstev jsou nalezeny vhodné topologie neuronových sítí pro rozpoznání základních tvarů v travním porostu. Pro experimenty je využit nástroj Neuroph.

3 Teoretická východiska

3.1 Již existující nástroje pro rozpoznávání rostlin

LeafSnap

LeafSnap (2) je jednou z aplikací která se zabývá rozpoznáváním rostlin podle listů. Byla vyvinuta vědci z Kolumbijské univerzity, univerzity v Marylandu a ze Smithonského institutu. Tato volně dostupná bezplatná aplikace využívá vizuální rozpoznávací software, který pomáhá rozpoznat druhy stromů z fotografií jejich listů.

Postup při použití aplikace je takový, že nejdříve utrhneme list stromu a umístíme jej na bílou plochu a pořídíme fotografii přímo z aplikace. Pokud je mobilní zařízení připojeno k internetu, tak aplikace začne vyhledávat shody oproti své databázi a pokusí se zjistit, o který druh stromu se jedná. Jako první dostaneme výsledky, které se nejvíce podobají hledané fotografii. Ve výsledcích jsou fotografie dané rostliny ve vysokém rozlišení, takže není problém si obrázek detailně přiblížit a ujistit se, že se jedná o správnou rostlinu. Při úspěšném rozpoznání zjistíme nejen název rostliny, ale i fotografie listu, plodu, květu, semen, kůry a to vše doplněné ještě o měřítko.

Aplikace obsahuje i sadu mini her, které Vás zábavnou formou vyzkouší z vašich znalostí druhů rostlin.

Nevýhodou je, že aplikace je zatím dostupná pouze ve dvou verzích. První obsahuje databázi rostlin, které se vyskytují v oblasti Severovýchodní Ameriky, a druhá je pro rostliny, které se vyskytují ve Velké Británii. Dalším omezením je, že aplikace je dostupná pouze pro zařízení, která fungují na platformě iOS od společnosti Apple.

Obrázek 1 - Ukázka aplikace LeafSnap[1]

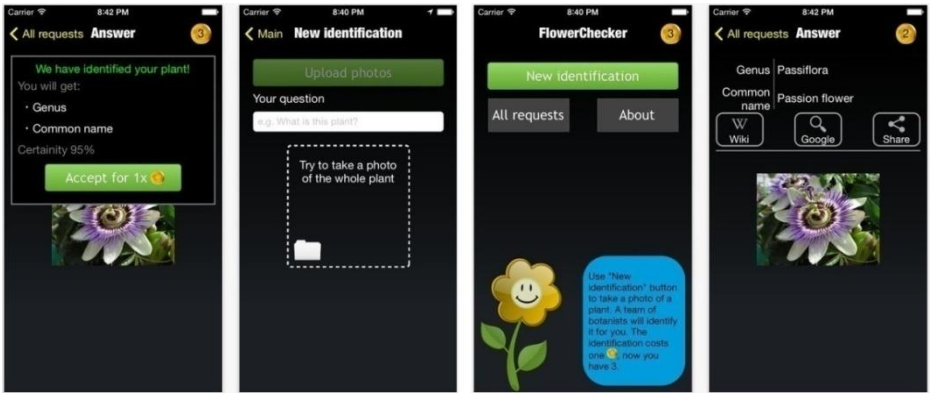
[1] Zdroj: https://itunes.apple.com/us/app/leafsnap/id430649829?mt=8

FlowerChecker

FlowerChecker (3) je další možností rozpoznávání rostlin. Tuto aplikaci vynalezli čeští studenti, kteří se seznámili díky Techsquatu (4). Což je ubytování pro studenty, kteří mají chuť se zdokonalovat ve svých oborech a komunikovat s dalšími ambiciózními studenty.

Použití a funkčnost aplikace je následující. Nejdříve pořídíme snímek rostliny, kterou chceme identifikovat. Aplikace snímek odešle spolu s informacemi o místě pořízení skupině odborníků. Vědci rozpoznají rostlinu na základě fotografie a místa pořízení. Odešlou zpátky odpověď i s mírou jistoty správnosti rozpoznání. Aplikace umožňuje jedno rozpoznání zdarma a další jsou zpoplatněna částkou 20 Kč. Vždy se platí za již vyřešené rozpoznání rostliny, takže se nemůže stát, že bychom zaplatili a nedostali odpověď.

Vývojáři této aplikace se nechali slyšet, že plánují spustit i rozpoznávání strojové, ale pro ten účel potřebují nasbírat spoustu ukázkových příkladů. Díky informacím o místě pořízení fotografie lze také s daty pracovat dále. Například botanici mohou mapovat výskyt méně častých druhů rostlin.



Obrázek 2- Ukázka aplikace FlowerChecker[2]

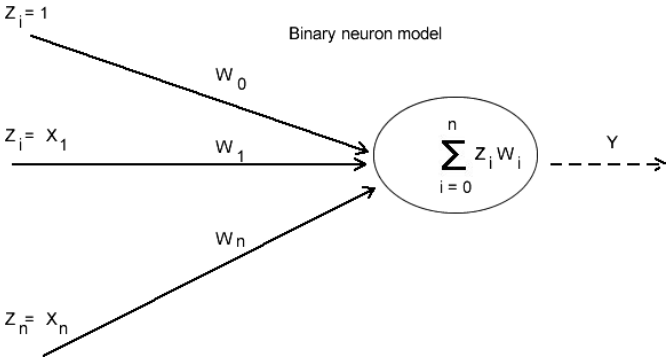
[2] Zdroj: https://itunes.apple.com/cz/app/id916709270

3.2 Neuronové sítě - teorie

Neuronová síť (5) je algoritmus, který je inspirován stavbou a funkčností lidského mozku tj. biologickými neuronovými sítěmi. Jako u biologických sítí jsou zde základní stavební jednotkou neurony. Ty jsou mezi sebou propojeny spoji, kterým u biologických sítí říkáme synapse. Na rozdíl od nich, v umělých neuronových sítích jsou spoje ohodnoceny váhami. Můžeme tedy říci, že neuronová síť vzniká vzájemným propojením neuronů. Počet a způsob propojení neuronů určuje architekturu sítě, kterou nazýváme topologií sítě.

Model neuronu

Model neuronu je základním stavebním prvkem sítě a je složen ze tří částí – vstupní, výstupní a funkční. U vstupu hraje roli váha vstupu a na jejím základě může být vstup potlačen, nebo zvýhodněn. Poté funkční část zpracuje hodnoty vstupů a vygeneruje výstup.



Obrázek 3- Model neuronu[3]

Na obrázku výše je graficky znázorněn model neuronu s jeho vstupy Z a váhami W kde jsou zpracovány tak, že suma součinů vstupů Z a jejich vah W je zpracována výstupní funkcí a poté vyvedena na výstup.

[3] Zdroj: Ing. Josef Pavlíček, Ph.D.

Aktivační funkce

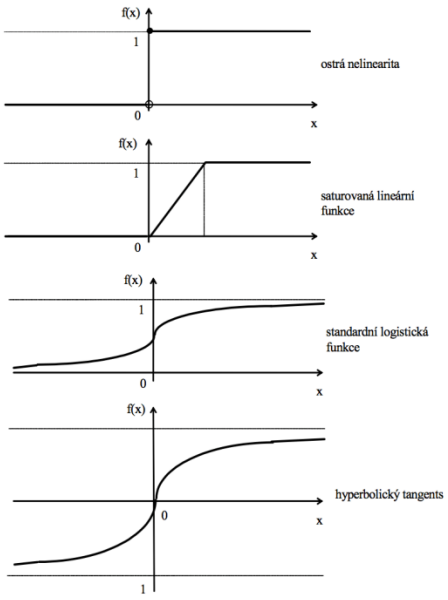
Využívané aktivační funkce na výstupech jednotlivých neuronů

$$f(x) = \begin{cases} 1 \; pokud \; x \geq 0 \\ 0 \; pokud \; x < 0 \end{cases}$$ ostrá nelinearita

$$f(x) = \begin{cases} 1 & x \geq 1 \\ x & 0 \leq x \leq 1 \\ 0 & x < 0 \end{cases}$$ saturovaná lineární funkce

$$f(x) = \frac{1}{1+e^{-x}}$$ standardní (logistická) sinoida

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}}$$ hyperbolický tangens



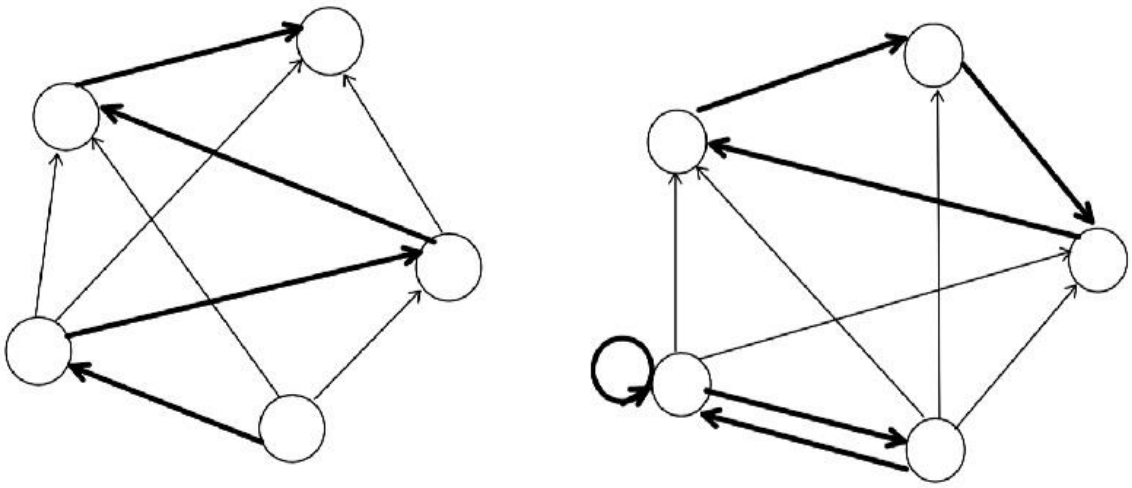
Obrázek 4 - Aktivační funkce neuronů[4]

[4] Zdroj: http://www1.osu.cz/~volna/Neuronove_site_skripta.pdf

Topologie sítě

Topologie sítě neboli návrh sítě můžeme rozdělit na síť s určitou a úplnou topologií. Rozdíl je v tom, že síť s úplnou topologií má propojeny neurony všemi možnými způsoby. Zatímco síť s určitou topologií takové propojení nemá, ale můžeme síť rozšířit o hrany, které chybí k úplnému propojení a doplnit je o váhy s nulovou hodnotou. Tak dostaneme síť úplnou, ale bude se chovat jako síť původní.

Dále můžeme architekturu rozdělit na síť cyklickou a acyklickou. (6) Cyklická síť má ve své struktuře uzavřený okruh mezi libovolným počtem neuronů. Může to být také jeden neuron, u kterého je jeho výstup zároveň jeho vstupem. U acyklické architektury graf neobsahuje žádnou uzavřenou smyčku. Takto uspořádané neurony můžeme rozdělit do vrstev a vrstvy lze uspořádat tak, že spoje mezi neurony povedou z vyšších vrstev do nižších.



Obrázek 5 - Acyklická a cyklická architektura (v pravo)[5]

Možné změny během činnosti sítě

Během činnosti sítě může docházet k následujícím změnám: (7)

1) Aktivní - stav neuronů
2) Adaptační - konfigurační váhy
3) Organizační -Topologie sítě

[5] Zdroj: http://www1.osu.cz/~volna/Neuronove_site_skripta.pdf

Aktivní režim

V aktivním režimu je počáteční stav sítě pevně určen konfigurací a topologií sítě. Jediný prvek, který projde změnou, jsou neurony, tedy jejich hodnota. Na začátku je vstup sítě nastaven na určité hodnoty a ostatní neurony jsou v počátečním stavu. Po inicializaci sítě dojde k výpočtu, což má za následek změny stavů neuronů. Cílem je, aby po konečném počtu operací se stav neuronů již dále neměnil. Výsledek výpočtu sítě je reprezentován hodnotami výstupních neuronů

Adaptační režim

V adaptačním režimu dochází ke změně vah na spojeních mezi jednotlivými neurony, ostatní nastavení sítě zůstává beze změn. Tento proces je označován také jako učení, neboli trénování neuronové sítě. Učení sítě probíhá za pomoci trénovací množiny dat, která definuje danou problematiku adaptace sítě. Postup učení můžeme rozdělit na dva způsoby: učení s učitelem a učení bez učitele.

V případě učení s učitelem jsou v učebním souboru dvojice hodnot. Hodnota vstupní a k té přiřazená hodnota výstupní. Je tedy zadán vstup a požadovaný výstup. Učení probíhá tak, že je zadána vstupní hodnota a po výpočtu sítě je výstup zkontrolován proti požadované hodnotě a následně jsou podle předem stanoveného algoritmu změněny váhy neuronové sítě. Tyto adaptační algoritmy provádějí změny vždy po předložení jednoho vektoru z učebního souboru.

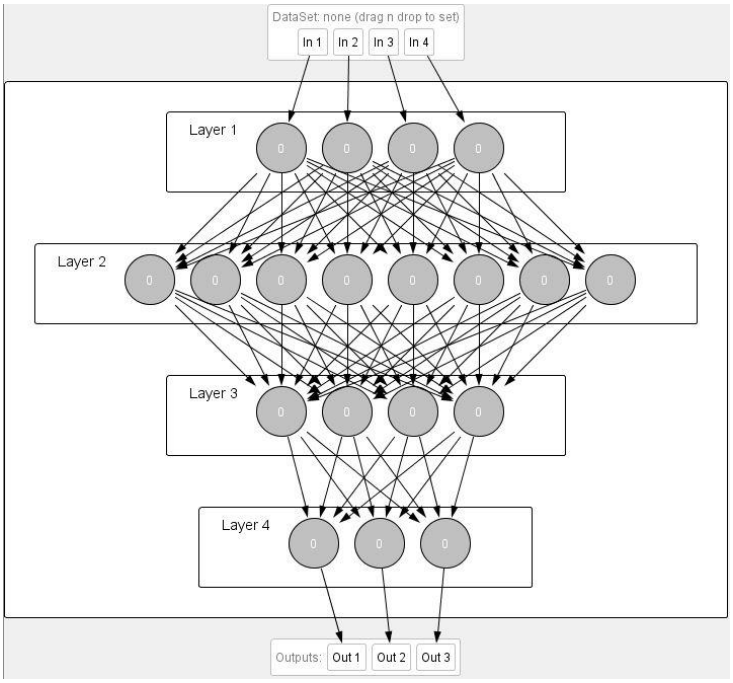
Při učení bez učitele obsahuje učební soubor pouze vstupy sítě. Není zde tedy obsažena požadovaná hodnota vstupu. Změny vah probíhají pouze na základě znalosti vstupu a výsledku výpočtu sítě. Neuronová síť v tomto případě při adaptaci rozřadí vstupy podle podobnosti do kategorií, ve kterých se nacházejí podobné vstupy.

Organizační režim

Tento režim spočívá ve změně topologie neuronové sítě. Nestává se, že by se síť měnila sama v procesu učení. Na začátku se zvolí určitá topologie sítě a provádí se na ní testy pomocí testovací množiny dat. Podle výsledku se přejde do organizačního režimu a provede se zásah do topologie sítě. Například při neuspokojivém výsledku jsou přidány neurony, nebo pokud je topologie moc složitá na daný problém, provede se redukce neuronů.

Vícevrstvá neuronová síť

Vícevrstvá neuronová síť je speciálním případem acyklické struktury. Nejvyšší vrstvou jsou zde vstupní neurony. Ty pouze přeposílají signály na všechny neurony ve vrstvě nižší a neprovádějí žádné výpočty. Další vrstvě se říká vrstva skrytá. Skrytých vrstev mezi vstupem a výstupem může být několik (n) s libovolným počtem neuronů. Výstup z neuronu vyšší vrstvy vede do všech vstupů neuronů z vrstvy nižší. Žádné jiné spojení neuronů v této síti neexistují. Nestane se tedy, že by byl spojen neuron s neuronem z nesousedící vrstvy a ani z vrstvy stejné. Neuronová síť s jednou skrytou vrstvou a jednou výstupní vrstvou se označuje jako dvouvrstvá. Pokud obsahuje dvě skryté vrstvy, tak nese označení třívrstvá. Vstupní vrstva se do označení nezahrnuje.



Obrázek 6 - Vícevrstvá neuronová síť

Třívrstvá neuronová síť, skládající se ze dvou skrytých vrstev s označením 8-4 a výstupní vrstvou, která obsahuje 3 výstupní neurony.

Algoritmus zpětného šíření chyby

Algoritmus zpětného šíření chyby neboli Back Propagation (7) se používá u vícevrstvých neuronových sítí a je to jeden z nejpoužívanějších učících algoritmů. Využívá se u sítí, které nemají zpětné vazby. Chyba se dostane z výstupu zpět k první vrstvě přes strukturu skrytých vrstev. K tomuto učení je potřeba znát vstupní a výstupní hodnotu, pro určení chyby. Učení pomocí zpětného šíření chyb můžeme rozdělit na následující fáze.

Nejdříve předložíme síti vstupní hodnoty. Síť provede výpočet a přivede hodnoty na výstup. Hodnotu z výstupu zkontrolujeme proti hodnotě požadované. A zjistíme chybu výstupu.

Následně je chyba šířena zpět od výstupu ke vstupním neuronům. Chybu jednotlivých neuronů definuje součet chyb neuronů následující vrstvy vynásobených odpovídajícími váhami. Každý neuron tedy obdrží dílčí chybu, kterou přispívá k celkové chybě. U první vrstvy, tedy vrstvy vstupní se nepočítá s chybou, jelikož tato vrstva pouze předává vstupní hodnoty následující skryté vrstvě.

Po zjištění chyby u každého neuronu dojde k přepočítání a následné úpravě jednotlivých vah mezi neurony. Celý tento proces se opakuje, dokud není chybovost sítě minimální.

3.3 Teorie detekce hran – rozpoznání obrazu

Detekce hran (8) patří spolu s redukcí šumu k základním operacím s obrazem. Při detekci hran dostaneme zvýrazněné hrany, které nesou největší informaci o celém obrazu. Detekujeme taková místa, ve kterých se nachází barevný přechod, nebo změna jasu. Aby byla detekce co nejpřesnější, provádí se před detekováním hran redukce šumu. To zabrání detekování falešných hran. Pro redukci šumu můžeme použít například Gaussův filtr.

Po detekování hran výrazně zmenšíme množství dat v obraze a přitom zachováme strukturální informaci obrazu pro další práci s obrazem.

„Neurofyziologický a psychofyzický výzkum ukazuje, že pro zrakové vnímání vyšších organismů jsou důležitá místa v obraze, kde se náhle mění hodnota jasu (významné hrany).“ (8)

Hrana

Hran může být chápána jako hranice objektů ve snímku nebo náhlá změna v jasu či barvě. Díky tomu můžeme detekovat objekty v daném snímku. Jelikož hrana je diskrétní aproximací gradientu, je tedy vektorem o dvou složkách a určuje směr a rychlost růstu obrazové funkce.

$$f(x, y)$$

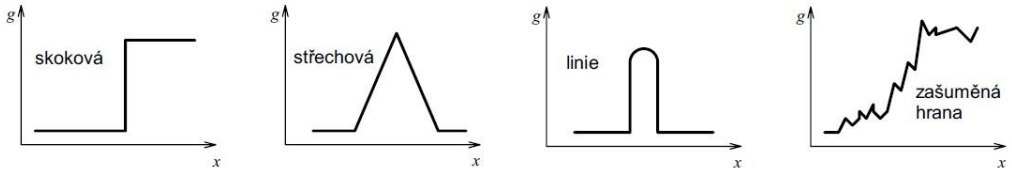
Hrany vznikají nespojitostí textury. Na snímku se nám to může projevit v těchto případech:

1) Nespojitostí v normále k povrchu
2) Nespojitostí hloubky
3) Lesk – odrazivost povrchu
4) Změnou barvy nebo textury povrchu
5) Stínem a nesouvislostí osvětlení

Hranový bod

Hranové body se používají pro hledání hranic objektů. Je to seskupení pixelů s velkým modulem gradientu.

Jasové profily v okolí hranových bodů:



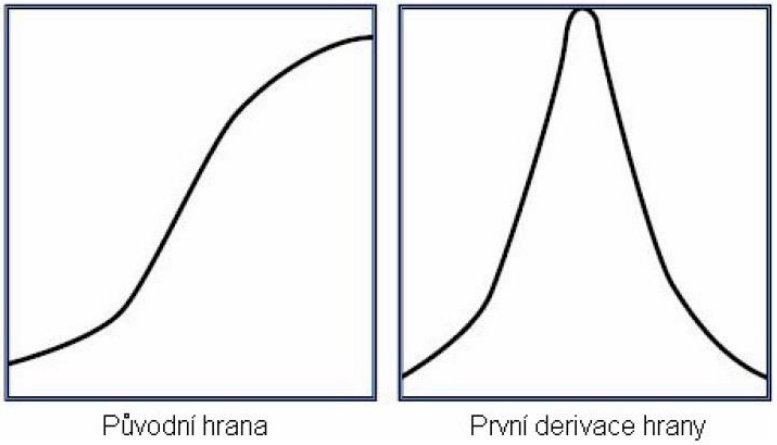
Obrázek 7 - Profily v okolí hranových bodů[6]

Z těchto vyobrazených možností tvaru hran se lze v praxi setkat pouze se zašuměnou hranou. Ostatní příklady jsou idealizované.

[6] Zdroj: http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/22EdgeDetectionCz.pdf

Hranové detektory založené na hledání maxim prvních derivací

Tyto detektory fungují na principu derivace obrazu a detekováním velké hodnoty derivace. Tam kde se nachází hrana, bude velká hodnota derivace, protože v místech, kde hrana není, se derivace rovná nule. Metodám první derivace se také říká gradientní metody, protože pracují s hranovým gradientem.



Obrázek 8- Ukázka první derivace hrany[7]

Konvoluční masky

Jeden ze způsobu výpočtu gradientu pomocí konvolučních masek. (9) Konvoluce se nazývá zpracovávání dvou funkcí, kdy používáme dvě matice. Zpracováváme jednu matici (obraz) pomocí jiné matice zvané „jádro“. Hranový operátor je konvoluční jádro, které se skládá z různých podob. Pokud aplikujeme konvoluci obrazu s tímto jádrem, dostaneme na výstupu hledanou složku gradientu.

Většinou se používají konvoluční masky s lichým počtem řádků a sloupců, protože máme veprostřed reprezentativní pixel a můžeme mu dát větší váhu. Konvoluční masky jsou náchylné vůči šumu, ale se zvětšováním masky se stávají odolnější vůči rušení výsledků šumem. Výsledný obraz je poté více rozostřen a výpočet je časově náročnější, než u menších masek.

[7] Zdroj: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=18131

3.3.1.1 Robertsův hranový detektor

Robertsova konvoluční maska je velmi jednoduchý a rychlý operátor. Používá jen okolí 2x2 aktuálního pixelu. Nejčastěji se používá na obrazy ve stupních šedi. Kvůli malému rozměru matice je velmi náchylný na označení šumu v obraze za hranu.

$$h_1 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}; h_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

3.3.1.2 Prewittové hranový detektor

Gradient tohoto operátoru je odhadován v matici 3x3 pro osm směrů. Tyto masky vznikají pootočením. Nakonec se vybere jen ta maska, která má největší modul gradientu.

$$h_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix};\ h_2 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}; h_3 = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & -1 \end{pmatrix}; \ldots$$

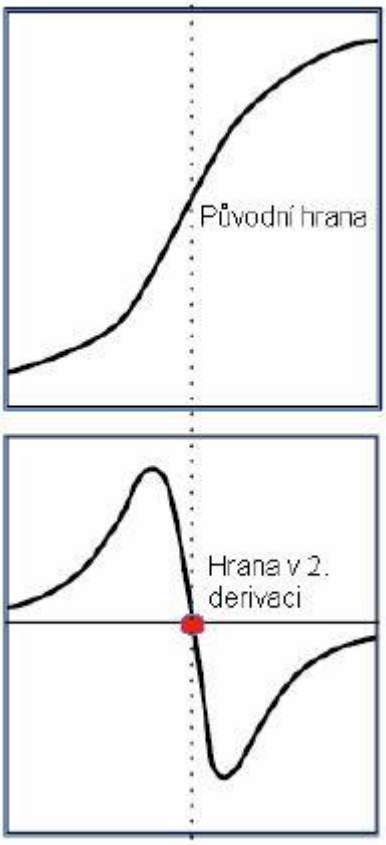
3.3.1.3 Sobelův hranový detektor

Sobelův operátor využívá tři masky v rozměru 3x3. Dává větší váhu středu a tím by mělo docházet k lepší detekci hran. Je často využíván k detekci vodorovných a svislých hran. Takže k detekci postačí masky *h1* (vodorovné hrany) a *h3* (svislé hrany).

$$h_1 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix};\ h_2 = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix}; h_3 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Hranové detektory založené na druhé derivaci

Tyto hranové detektory používají průnik nulou k detekování hrany. Anglicky jsou nazývány zero-crossing. Při druhé derivaci původní hrany, funkce protne osu v nule a tam se nachází hrana. Tento postup detekuje hranu, která je označena pouze jedním pixelem, a výsledkem jsou velmi tenké linie. Využívá skutečnosti, že je jednodušší vyhledávat průchody signálu nulou, než maximem.



Obrázek 9 - Zobrazení funkce druhé derivace[8]

[8] Zdroj: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=18131

3.3.1.4 Cannyho hranový detektor

Cannyho hranový detektor (10) je složen z několika kroků a pro své vlastnosti je jeden z nejčastěji používaných detektorů. Pro správnou funkčnost detektoru musí být splněny tyto požadavky: (9)

- Detekční kritérium – detekují se pouze místa, která jsou důležitými hranami, tím pádem nedojde k detekci falešných hran
- Lokalizační kritérium – vzdálenost mezi skutečnou a nalezenou hranou musí být co nejmenší
- Požadavek jedné odezvy - každá hrana musí být detekována pouze jednou

Abychom dosáhli výše vypsaných kritérií, musíme použít více postupů. Standardní postup se skládá ze čtyř kroků:

Eliminace šumu Gaussovým filtrem

1) Nejprve použijeme odstranění šumu pomocí Gaussova filtru, protože všechny obrázky pořízené fotoaparátem osahují šum. To zajistí, že v obraze zůstanou pouze výrazné hrany. Nepodstatné, menší hrany budou filtrem odstraněny. Je důležité zvolit správnou hodnotu σ, aby došlo ke správné detekci šumu. Pokud bude hodnota σ moc malá, dojde k detekci nevýznamných detailů a vznikne více falešných hran, naopak při moc velké hodnotě σ zaniknou slabé hrany a nebudou detekovány.

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

Takto se vypočítá konvoluční maska, která se poté aplikuje na celý obraz.

Příklad Gaussova filtru při použití σ = 1.4:

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

2) Určení gradientu z první derivace

V tomto kroku určíme velikosti a směry gradientů pomocí Sobbelova operátoru, který je lépe odolný vůči šumu.

$$h_1 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix};\ h_2 = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & -0 \end{pmatrix};\ h_3 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

3) Nalezení lokálních maxim (Thining)

V této části detekujeme hrany v místě největšího gradientu pomocí lokálního maxima a body, které nejsou lokálním maximem, odebíráme. Tím z rozmazaných hran uděláme ostré hrany, které budou tenčí než hrany původní. Tento proces provedeme pomocí velikosti a směru gradientu, které máme z předchozího kroku.

4) Eliminace nevýznamných hran (Thresholding)

Posledním krokem Cannyho detektoru je prahování. Většina hran, které máme teď v obraze, jsou už pravé hrany, ale pořád zde můžeme nalézt hrany vzniklé silným šumem, barevnou variací, nebo nerovným povrchem. Řešení spočívá v určení dvou prahů, které považujeme za hranu a co už hrana není. Jeden práh určuje horní hodnotu a druhý dolní hodnotu, mezi kterými se může gradient pohybovat. Pokud rozhodujeme o bodu, který má hodnotu vyšší než je horní práh, označíme hranu silně. Body, které se nachází pod dolní hranicí, potlačíme a body které se nachází mezi horní a dolní hranicí označíme tenkou hranou.

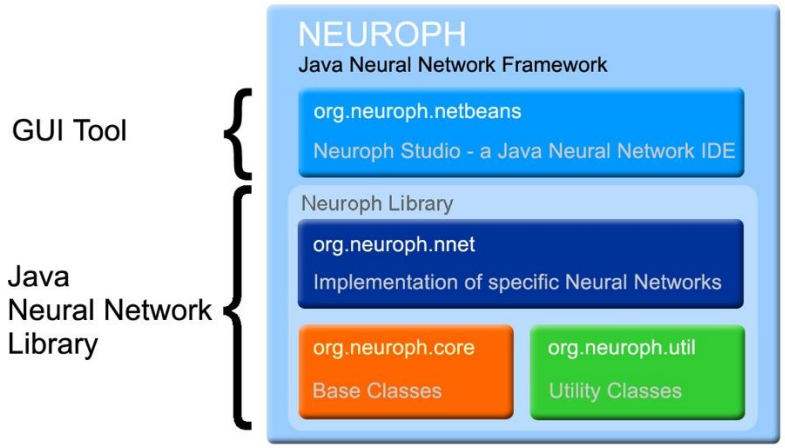
Využití pro práci

Detekce hran je jednou z možných dílčích procesů při rozpoznávání obrazu. Ve své práci jsem prakticky teorii hran nevyužil, ale je to teoretická příprava pro další navazující práci, ve které bude využita úprava obrazu pomocí detekce hran, a neuronové sítě budou podrobeny dalšímu výzkumu.

3.4 Neuroph

Neuroph (11) je Java Framework pro vývoj a stavbu běžných neuronových sítí. Obsahuje dobře navrženou a volně dostupnou Java knihovnu, která obsahuje malé množství základních tříd. Tyto třídy korespondují se základním konceptem neuronových sítí. Celý Framework je vydán jako software s otevřeným zdrojovým kódem. Můžeme ho tedy používat pro soukromé i komerční účely podle licence Apache 2.0.

Tato platforma má i aplikaci Neuroph Studio s propracovaným GUI pro jednoduché vytváření neuronových sítí. Neuroph Studio nám umožňuje vytvořit si neuronovou síť, kterou natrénujeme pomocí vlastní trénovací množiny dat a následně síť můžeme testovat pomocí námi zvolených testovacích dat. Po tomto procesu zjistíme, s jak velkou přesností bylo určení správné. Takto naučenou neuronovou síť můžeme exportovat ve formě souboru a použít jí ve vlastní Java aplikaci.



Obrázek 10 - Struktura aplikace Neuroph[9]

[9] Zdroj: http://neuroph.sourceforge.net/

4 Vlastní práce

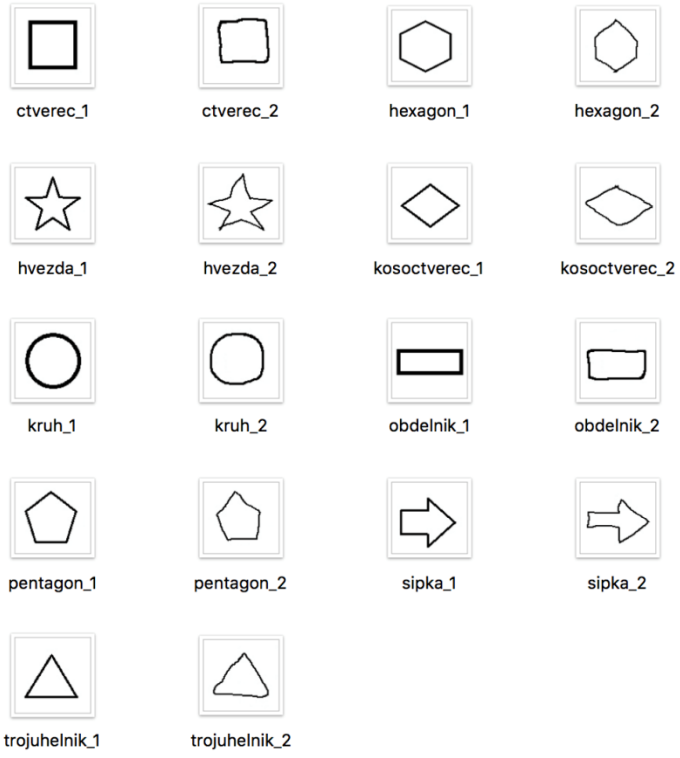
4.1 Vytvoření a natrénování neuronové sítě pomocí aplikace Neuroph Studio

Neuroph Studio poskytuje prostředí pro vytváření a trénování neuronových sítí. Také obsahuje specializovaný nástroj pro trénování neuronových sítí pro rozpoznání obrazu. Vytvoření a naučení sítě se skládá z následujících kroků:

1) Vytvoření Neuroph projektu
2) Vytvoření neuronové sítě určené k rozpoznání obrazu
3) Naučení sítě pomocí trénovacích dat
4) Otestování sítě testovací množinou dat
5) Uložení pro budoucí možné využití sítě ve vlastní Java aplikaci

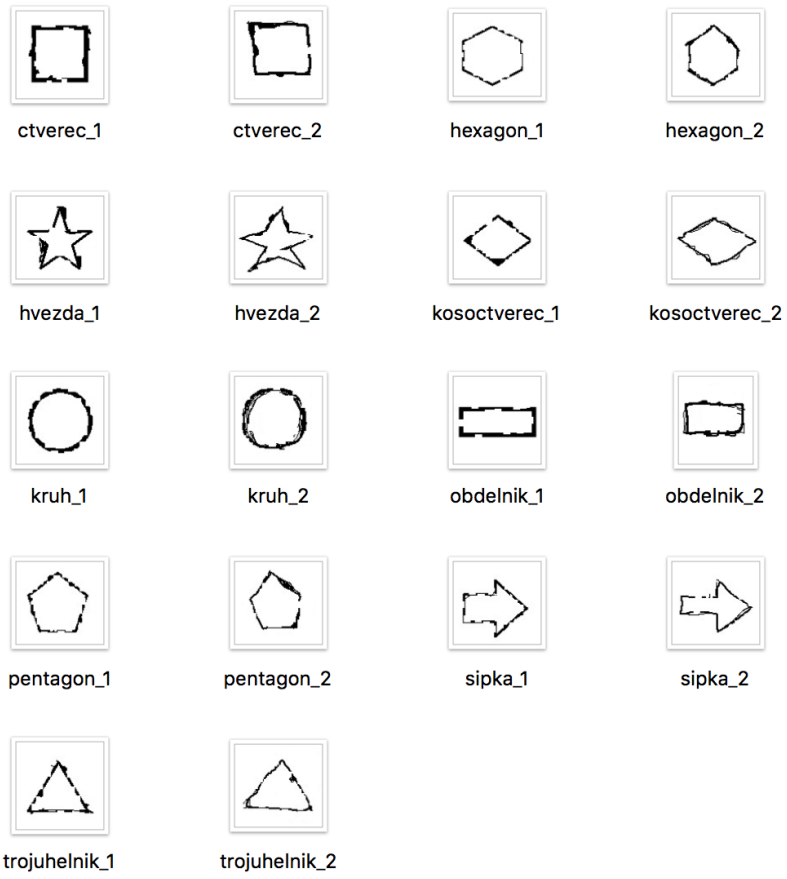
Vytvoření trénovací a testovací množiny dat základních tvarů

Pro ukázku funkcionality Neuroph Studia byla vytvořena základní trénovací množina dat v podobě základních tvarů. Obrazy jsou v rozlišení 100x100 pixelů. Sada tvarů byla vytvořena v jednoduchém grafickém editoru MS Paint.



Obrázek 11 - Trénovací množina dat

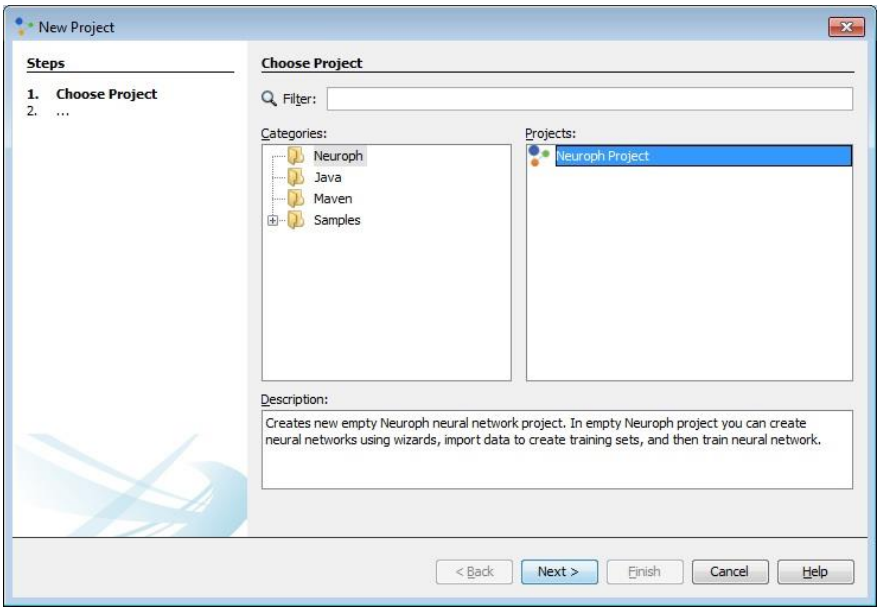
Testovací množina dat byla vytvořena upravením trénovací množiny. Vznikla tedy množina neznámých tvarů, která nebyla síti předložena v rámci trénování. Testovací obrazy jsou také v rozlišení 100x100 a slouží k otestování schopnosti naučené sítě rozpoznat obrazy podobné obrazům z trénovací množiny dat.



Obrázek 12 - Testovací množiny dat

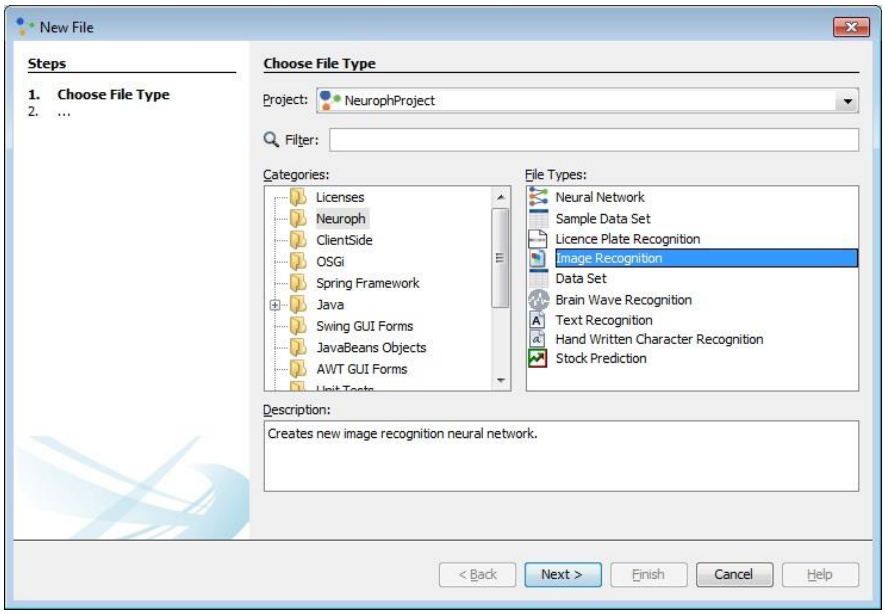
4.2 Vytvoření projektu a natrénování sítě

Prvním krokem je vytvoření nového projektu, ve kterém následně vytvoříme neuronovou síť a trénovací množinu.



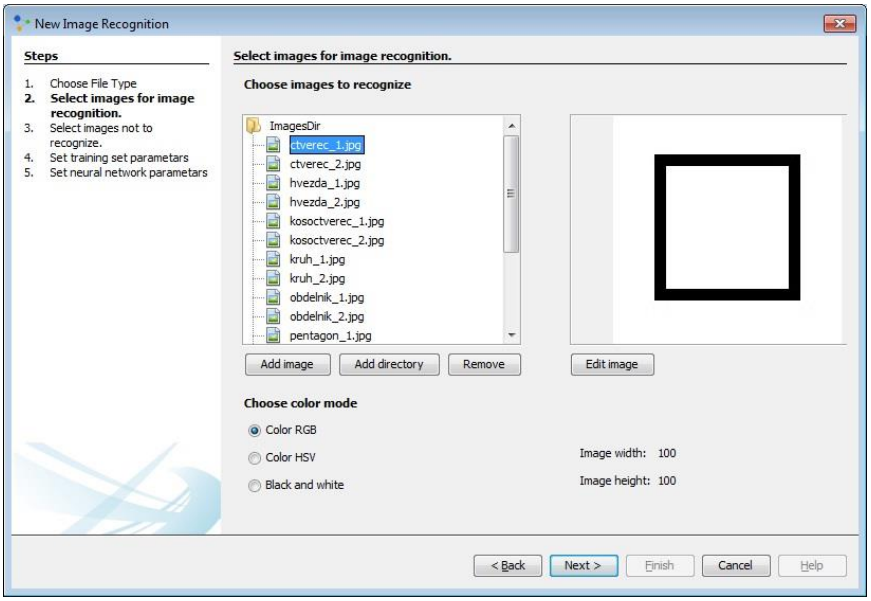
Obrázek 13 - Vytvoření projektu

V novém projektu vytvoříme soubor s neuronovou sítí. Na výběr máme z několika typů průvodců vytvoření neuronové sítě podle námi zvolené problematiky. V tomto kroku zvolíme průvodce pro přidání neuronové sítě pro rozpoznání obrazu.



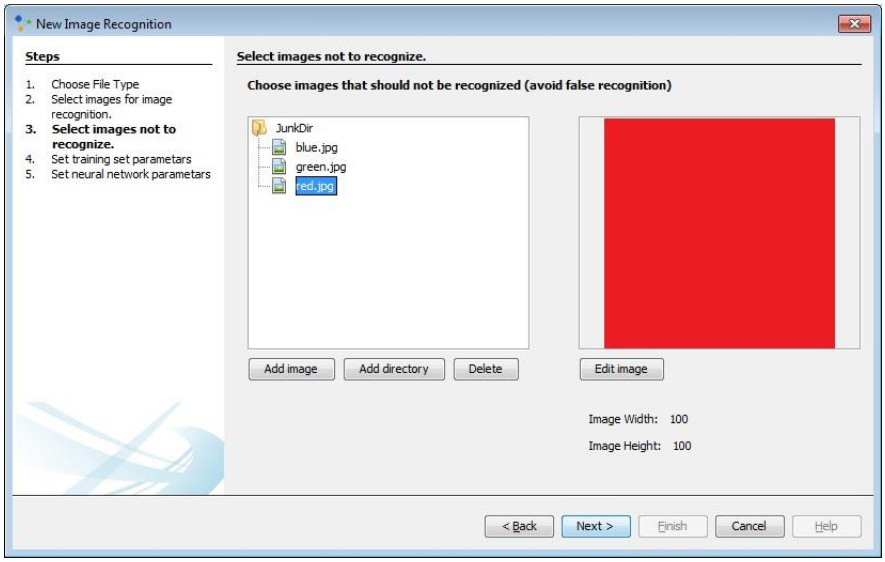
Obrázek 14 - Vytvoření neuronové sítě pro rozpoznání obrazu

V dalším kroku vybereme množinu obrazů, které chceme, aby se síť naučila rozpoznávat. Obrazy může přidávat jednotlivě, nebo po celých adresářích. Máme zde také na výběr, jestli chceme používat barevný či černobílý mód. Při použití černobílého módu jsou pixely v obraze reprezentovány jako [0, 1] a tím se zmenší množství vstupních neuronů. Pro rozpoznávání černobílých obrazů je potom potřeba méně složitá topologie neuronové sítě. V této fázi máme ještě možnost obrazy upravit pomocí jednoduchého editoru. Provádět můžeme úpravy typu – otočení, změna velikosti, nebo převedení do odstínů šedi.



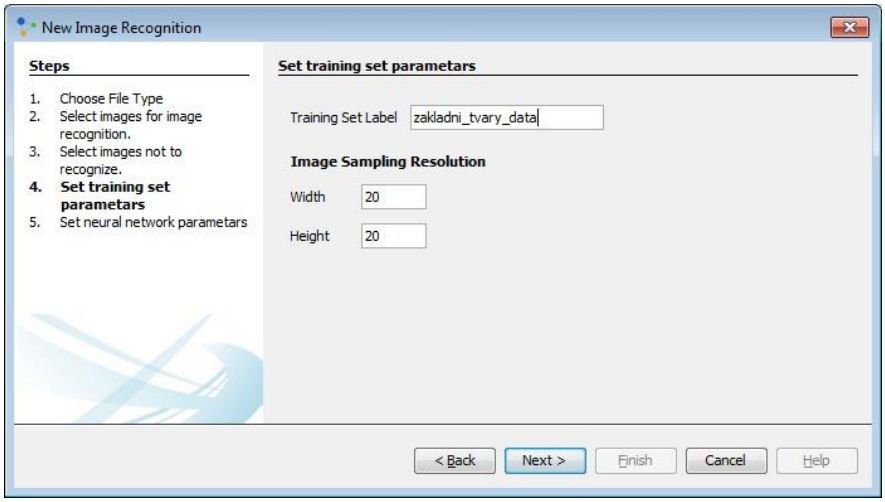
Obrázek 15 – Výběr testovací množiny

Další fází je výběr obrazů, které nemají být rozpoznány. Tyto obrazy pomáhají minimalizovat nesprávná rozpoznání. Obvykle se používají obrazy vyplněné červenou, zelenou a modrou barvou. Musíme si sami zvolit podle kontextu a využití našeho rozpoznání, které obrazy nechceme, aby byly rozpoznány.



Obrázek 16 - Výběr množiny pro nerozpoznání

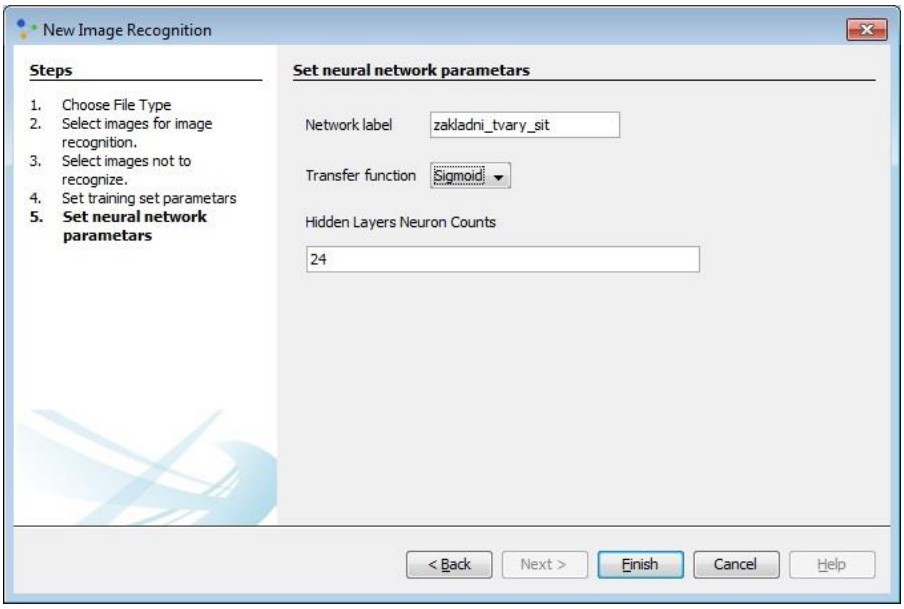
Poté pojmenujeme testovací množinu dat pro případ, že v našem projektu budeme experimentovat i s dalšími trénovacími množinami. Nastavení šířky a výšky změní rozměr u všech vstupních obrazů. Budou tedy zobrazeny v menší škále a učení bude probíhat snadněji a rychleji. Rozměry obrazu udávají také velikost vstupního vektoru a počet neuronů ve vstupní vrstvě.



Obrázek 17 - Parametry vstupních dat

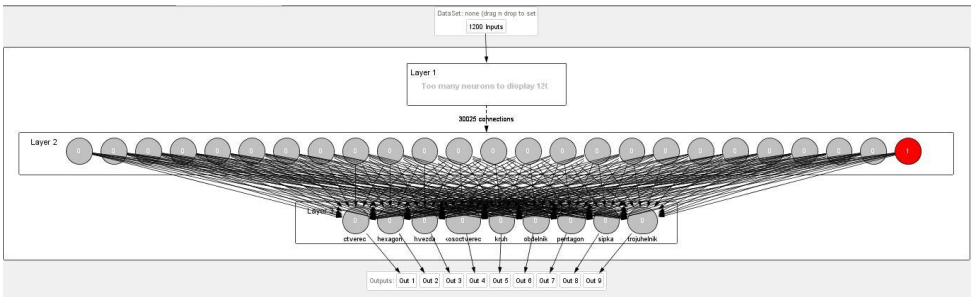
Posledním krokem průvodce je přidání neuronové sítě a nastavení jejich parametrů. Nejprve síť pojmenujeme pro případ, že budeme později vytvářet další sítě. Jako další parametr je přenosová funkce. Zde je na výběr buďto sigmoida, nebo hyperbolický tangens. Nejčastěji se používá sigmoida, ale v některých případech dosáhneme lepších výsledků využitím tangensu. Nakonec zvolíme počet neuronů ve skryté vrstvě a počet

skrytých vrstev. To je nejdůležitější nastavení sítě. Skryté vrstvy se nacházejí mezi vstupní a výstupní vrstvou. Cílem je mít co nejmenší možný počet vrstev a neuronů, které se úspěšně zvládnou naučit na trénovací množinu dat. Menší počet neuronů znamená rychlejší učení, menší počet iterací a lepší generalizaci. Adekvátní počet neuronů a vrstev záleží na počtu vstupních a výstupních neuronů. Nelze obecně určit počet neuronů a skrytých vrstev, protože každé rozpoznávání je speciálním případem. Ideální počet se proto často nalezne až po sérii pokusů a experimentů s topologií neuronové sítě.



Obrázek 18 - Parametry neuronové sítě

Po dokončení průvodce se v projektu vytvoří dva soubory. Jeden soubor s neuronovou sítí, která je připravena na proces učení a druhý s trénovací množinou dat. Graficky zobrazená neuronová síť obsahuje 3 vrstvy – vstupní, skrytou a výstupní. Ve skryté vrstvě je 24 neuronů a ve výstupní je 9 neuronů, každý z nich reprezentuje jeden ze základních tvarů z trénovací množiny.



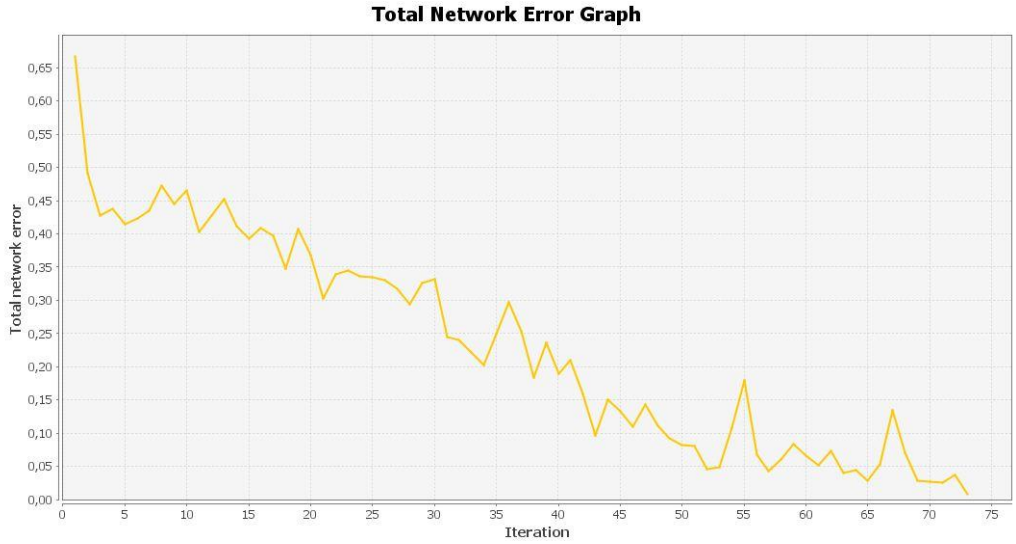
Obrázek 19 – Neuronová síť

Při spuštění trénování se v dialogovém okně nastavují trénovací parametry. Pro rychlost učení a moment se používají hodnoty v intervalu [0,1] a pro maximální chybu sítě je doporučená hodnota 0.1. Při vzorovém učení bylo použito následující nastavení.

Maximální počet iterací	Bez omezení
Maximální chyba sítě	0.01
Moment	0.7
Krok	0.2

Tabulka 1 - Nastavení parametrů

Po natrénování sítě je zobrazen graf celkové chybovosti a průběhu učení neuronové sítě.



Obrázek 20 – Graf celkové chybovosti učení sítě

Neuroph Studio disponuje funkcionalitou pro testování naučené neuronové sítě. Testování se skládá z předkládání neznámých obrazů neuronové síti, která vyhodnocuje, kterému prvku z testovací množiny se nejvíce obraz podobá. Po vyhodnocení má odpovídající výstupní neuron největší hodnotu. Výstupní hodnoty se na neuronech pohybují v intervalu [0,1]. Čím blíže je výstupní hodnota číslu 1, tím je určení přesnější. V následující tabulce je zobrazena přesnost určení rozpoznání objektů z testovací množiny.

Čtverec_1	0.9827
Čtverec_2	0.6997
Hexagon_1	0.6463
Hexagon_2	0.9230
Hvězda_1	0.9862
Hvězda_2	0.9734
Kosočtverec_1	0.9887
Kosočtverec_2	0.9722
Kruh_1	0.9262
Kruh_2	0.8620
Obdélník_1	0.9096
Obdélník_2	0.0676
Pentagon_1	0.9083
Pentagon_2	0.9100
Šipka_1	0.9316
Šipka_2	0.9188
Trojúhelník_1	0.8041
Trojúhelník_2	0.4135

Tabulka 2 - Přehled přesností určení

4.3 Zhodnocení výsledků

Při testování dvouvrstvé sítě s 24 neurony ve skryté vrstvě pomocí testovací množiny bylo dosaženo uspokojivého výsledku. Většina přesností určení rozpoznání se pohybuje kolem hodnoty 0.9, cože je velice vysoká přesnost určení za předpokladu, že stoprocentní rozpoznání je hodnota 1. Nejlepšího výsledku u rozpoznání neznámého tvaru dopadl Kosoctverec_1 s přesností určení 0.9887. Nejhůře dopadlo rozpoznání tvaru Obdelník_2 z testovací množiny dat s přesností určení 0.0676, ale stále to byla největší hodnota neuronu pro daný tvar, takže určení bylo správné.

4.4 Testování topologie sítě v závislosti na trénovací množině

Testovací množina dat

V rámci výběru vhodné topologie sítě pro tři odlišné trénovací množiny byla zvolena jedna testovací množina dat. Testovací data byla pořízena mobilním telefonem Apple iPhone 5 v rozlišení 2448 x 3264 px. Fotografie by mohly být, také pořízeny pomocí robo – optické jednotky. Jedná se o fotografie čtyř základních tvarů v travním porostu – čtverec, obdélník, kruh a trojúhelník. Tyto tvary byly vystřiženy z černého materiálu a následně zachyceny v trávě. Pro účely rozpoznávání byly tyto fotografie zmenšeny na velikost 100x133 pixelů při zachování poměrů stran.

Obrázek 21 - Foto tvarů v trávě

Trénovací množiny dat

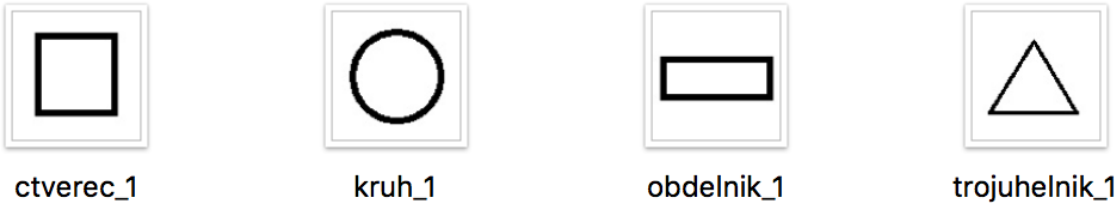
Jako učební materiál pro neuronové sítě byly zvoleny tři množiny trénovacích dat. Tyto množiny simulují různou velikost a způsob pořízení dat.

4.4.1.1 Množina A

Soubor devíti odlišných základních tvarů, které byly vytvořeny ve dvojím provedení. Jednou pomocí rýsovacího nástroje a podruhé jako simulace nakreslení tvarů pomocí ruky. Tato množina obsahuje tedy celkem 18 obrazů a je z trénovacích množin nejobsáhlejší. Grafické znázornění množiny viz obrázek 11 – Trénovací množina dat

4.4.1.2 Množina B

Množina B byla vytvořena zredukováním trénovací množiny A, na tvary obsažené v testovací množině. Tento trénovací soubor obsahuje tedy čtyři základní tvary - čtverec, obdélník, kruh a trojúhelník. Redukce tvarů zmenší velikost učebního souboru a tím zjednoduší proces učení. Neuronová síť nebude vyžadovat takovou složitost, jako u větších trénovacích souborů a proces učení bude rychlejší.



Obrázek 22 - Množina B

4.4.1.3 Množina C

Tato množina obsahuje fotografie čtyř základních tvarů, které byly pořízeny stejným mobilním telefonem, jako množina pro testování naučených neuronových sítí. Opět byla změněna velikost fotografií na 100x133 pixelů. Vystřižené tvary byly zachyceny na bílém podkladu, čímž vzniklo jednolité pozadí.

Obrázek 23 - Množina C

Postup testování

Průběh procesu učení a testování sítě je stejný jako u kapitoly „Vytvoření a natrénování neuronové sítě pomocí aplikace Neuroph Studio“. Experimentováním s počtem skrytých vrstev a počtem neuronů v nich obsažených byl vytvořen následující přehled testů. Všechny učební procesy byly provedeny v jednotném nastavení učebních kritérií. Stejným nastavením a testovacím souborem zaručíme objektivní výsledky u různých topologií.

Maximální počet iterací	Bez omezení
Maximální chyba sítě	0.01
Moment	0.7
Krok	0.2

Tabulka 3 - Parametry nastavení

Testování jednotlivých množin bylo rozděleno na experimenty s jednou a s dvěma skrytými vrstvami. Při výběru počtu neuronů v jednotlivých vrstvách byla vždy nejprve hledána topologie s nejmenším možným počtem neuronů, se kterými byla síť schopna naučit se trénovací množinu dat. Dalším testem bylo zvednutí počtu neuronů na dvojnásobnou hodnotu. U dvouvrstvých sítí byl zdvojnásoben počet neuronů u obou skrytých vrstev.

Po naučení bylo provedeno testování pomocí fotografií objektů v travním porostu a výsledky, tedy hodnoty přesnosti určení byly zaneseny do tabulek.

Trénovací množina A

4.4.1.4 Sítě s jednou skrytou vrstvou

1 skrytá vrstva - 14 neuronů										
Síť A_1		Výstup								
		Obdélník	Šipka	Hvězda	Kosočtverec	Kruh	Čtverec	Hexagon	Trojúhelník	Pentagon
Vstup	Obdélník	0,2236	0,0038	0,0087	0,0022	0,0016	0,0152	0,0005	**0,4164**	0,0046
	Šipka	-	-	-	-	-	-	-	-	-
	Hvězda	-	-	-	-	-	-	-	-	-
	Kosočtverec	-	-	-	-	-	-	-	-	-
	Kruh	0,0003	0,0005	0,0072	0,0207	0,0007	**0,6874**	0,0014	0,0056	0,0004
	Čtverec	0,0019	0,0100	0,0210	0,0017	0,0002	0,0001	0,0053	**0,1750**	0,0037
	Hexagon	-	-	-	-	-	-	-	-	-
	Trojúhelník	0,0061	0,1074	**0,4580**	0,0003	0,0000	0,0002	0,0020	0,3883	0,0413
	Pentagon	-	-	-	-	-	-	-	-	-

Tabulka 4 - Výsledky měření A_1

Počet správně rozpoznaných tvarů: 0

1 skrytá vrstva - 28 neuronů										
Síť A_2		Výstup								
		Obdélník	Šipka	Hvězda	Kosočtverec	Kruh	Čtverec	Hexagon	Trojúhelník	Pentagon
Vstup	Obdélník	0,1680	0,6020	0,2478	0,7864	0,6507	0,5424	0,6065	0,5678	**0,8800**
	Šipka	-	-	-	-	-	-	-	-	-
	Hvězda	-	-	-	-	-	-	-	-	-
	Kosočtverec	-	-	-	-	-	-	-	-	-
	Kruh	0,0647	0,3798	0,3434	0,8476	0,6413	0,6429	**0,9207**	0,8341	0,8654
	Čtverec	0,0427	0,7149	0,5891	0,5336	0,5367	0,5930	0,4812	0,7463	**0,8790**
	Hexagon	-	-	-	-	-	-	-	-	-
	Trojúhelník	0,0877	0,8052	0,7753	0,7847	0,4096	0,5908	0,5039	0,5677	**0,9527**
	Pentagon	-	-	-	-	-	-	-	-	-

Tabulka 5 - Výsledky měření A_2

Počet správně rozpoznaných tvarů: 0

4.4.1.5 *Sítě s dvěma skrytými vrstvami*

2 skryté vrstvy – dvakrát 36 Neuronů										
Síť A_3		Výstup								
		Obdélník	Šipka	Hvězda	Kosočtverec	Kruh	Čtverec	Hexagon	Trojúhelník	Pentagon
Vstup	Obdélník	**0,8137**	0,0031	0,0000	0,0000	0,0000	0,1574	0,0003	0,0161	0,0001
	Šipka	-	-	-	-	-	-	-	-	-
	Hvězda	-	-	-	-	-	-	-	-	-
	Kosočtverec	-	-	-	-	-	-	-	-	-
	Kruh	0,0008	0,0007	0,0004	0,0000	0,0903	**0,8446**	0,0004	0,0008	0,0004
	Čtverec	0,0000	0,0200	0,0000	0,0748	0,0110	0,0001	**0,2657**	0,0007	0,0164
	Hexagon	-	-	-	-	-	-	-	-	-
	Trojúhelník	**0,6647**	0,1206	0,0004	0,0001	0,0000	0,0011	0,0004	0,0038	0,0002
	Pentagon	-	-	-	-	-	-	-	-	-

Tabulka 6 - Výsledky měření A_3

Počet správně rozpoznaných tvarů: 1

2 skryté vrstvy – dvakrát 72 neuronů										
Síť A_4		Výstup								
		Obdélník	Šipka	Hvězda	Kosočtverec	Kruh	Čtverec	Hexagon	Trojúhelník	Pentagon
Vstup	Obdélník	0,0001	**0,2023**	0,0006	0,0336	0,0130	0,0003	0,0044	0,0005	0,0004
	Šipka	-	-	-	-	-	-	-	-	-
	Hvězda	-	-	-	-	-	-	-	-	-
	Kosočtverec	-	-	-	-	-	-	-	-	-
	Kruh	0,0001	**0,2023**	0,0006	0,0336	0,0130	0,0003	0,0044	0,0005	0,0004
	Čtverec	0,0001	**0,2023**	0,0006	0,0336	0,0130	0,0003	0,0044	0,0005	0,0004
	Hexagon	-	-	-	-	-	-	-	-	-
	Trojúhelník	0,0004	0,0068	0,0109	0,0025	0,0002	0,0003	0,0180	**0,0209**	0,0053
	Pentagon	-	-	-	-	-	-	-	-	-

Tabulka 7 - Výsledky měření A_4

Počet správně rozpoznaných tvarů: 1

Trénovací množina B

4.4.1.6 *Sítě s jednou skrytou vrstvou*

1 skrytá vrstva - 13 neuronů					
Síť B_1		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	0,0097	0,2909	0,0008	**0,4978**
	Kruh	0,0021	0,0949	**0,1002**	0,0755
	Čtverec	0,0986	0,0501	**0,6029**	0,0357
	Trojúhelník	**0,7912**	0,1951	0,1459	0,0110

Tabulka 8 - Výsledky měření B_1

Počet správně rozpoznaných tvarů: 1

1 skrytá vrstva - 26 neuronů					
Síť B_2		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	**0,2188**	0,1319	0,0014	0,1942
	Kruh	0,0065	0,0798	**0,8753**	0,0005
	Čtverec	0,0390	0,0252	0,0142	**0,5335**
	Trojúhelník	0,1786	0,0902	0,0008	**0,2246**

Tabulka 9 - Výsledky měření B_2

Počet správně rozpoznaných tvarů: 2

4.4.1.7 Sítě s dvěma skrytými vrstvami

2 skryté vrstvy – dvakrát 13 neuronů					
Síť B_3		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	0,0385	0,0062	0,0078	**0,7873**
	Kruh	0,0123	0,0378	0,0102	**0,5031**
	Čtverec	0,0416	0,0042	0,0214	**0,6656**
	Trojúhelník	0,0350	0,0092	0,0050	**0,8225**

Tabulka 10 - Výsledky měření B_3

Počet správně rozpoznaných tvarů: 1

2 skryté vrstvy - dvakrát 26 neuronů					
Síť B_4		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	**0,0713**	0,0372	0,0320	0,0335
	Kruh	0,0988	0,0059	**0,8843**	0,0094
	Čtverec	**0,0652**	0,0371	0,0304	0,0370
	Trojúhelník	0,0175	0,0086	0,0142	**0,5799**

Tabulka 11 - Výsledky měření B_4

Počet správně rozpoznaných tvarů: 2

Trénovací množina C

4.4.1.8 Sítě s jednou skrytou vrstvou

1 skrytá vrstva - 4 neurony					
Síť C_1		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	0,0685	**0,0875**	0,0348	0,0781
	Kruh	0,0672	**0,0886**	0,0353	0,0790
	Čtverec	0,0115	0,0030	0,0815	**0,8340**
	Trojúhelník	**0,8776**	0,0036	0,0023	0,0056

Tabulka 12 - Výsledky měření C_1

Počet správně rozpoznaných tvarů: 1

1 skrytá vrstva - 8 neuronů					
Síť C_2		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	0,4750	0,0327	**0,7623**	0,0579
	Kruh	0,0009	**0,3683**	0,0111	0,0663
	Čtverec	0,0410	0,0019	**0,2082**	0,0386
	Trojúhelník	**0,4500**	0,0039	0,0230	0,3727

Tabulka 13 - Výsledky měření C_2

Počet správně rozpoznaných tvarů: 2

4.4.1.9 Sítě s dvěma skrytými vrstvami

2 skryté vrstvy – dvakrát 9 neuronů					
Síť C_3		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	0,0303	0,0366	0,0689	**0,5368**
	Kruh	0,0158	0,0009	**0,1421**	0,0231
	Čtverec	0,0636	0,0113	0,2483	**0,6795**
	Trojúhelník	0,0581	0,0047	0,1057	**0,7677**

Tabulka 14 - Výsledky měření C_3

Počet správně rozpoznaných tvarů: 1

2 skryté vrstvy – dvakrát 18 neuronů					
Síť C_4		Výstup			
		Obdélník	Kruh	Čtverec	Trojúhelník
Vstup	Obdélník	0,0314	0,0022	**0,1428**	0,0022
	Kruh	0,0020	0,0121	**0,1522**	0,0298
	Čtverec	0,0050	0,0087	0,0056	**0,1867**
	Trojúhelník	0,0425	0,0198	0,0008	**0,2609**

Tabulka 15 - Výsledky měření C_4

Počet správně rozpoznaných tvarů: 1

5 Zhodnocení výsledků

5.1 Výsledky jednotlivých učebních souborů

Byl vytvořen přehled topologií sítí a jejich úspěšnosti rozpoznání za jednotlivé učební soubory. K jednotlivým sítím byly zaznamenány správně rozpoznané tvary pro pozdější celkové porovnání mezi odlišnými množinami.

Množina A

Název sítě	Počet vrstev	Počet neuronů	Počet správných rozpoznání	Správně rozpoznané tvary
Síť A_1	1	14	0	-
Síť A_2	1	28	0	-
Síť A_3	2	36	1	obdélník
Síť A_4	2	72	1	trojúhelník

Tabulka 16 - Souhrn měření množiny A

Množina B

Název sítě	Počet vrstev	Počet neuronů	Počet správných rozpoznání	Správně rozpoznané tvary
Síť B_1	1	13	1	čtverec
Síť B_2	1	26	2	obdélník, trojúhelník
Síť B_3	2	13	1	trojúhelník
Síť B_4	2	26	2	obdélník, trojúhelník

Tabulka 17 - Souhrn měření množiny B

Množina C

Název sítě	Počet vrstev	Počet neuronů	Počet správných rozpoznání	Správně rozpoznané tvary
Síť C_1	1	4	1	kruh
Síť C_2	1	8	2	kruh, čtverec
Síť C_3	2	9	1	trojúhelník
Síť C_4	2	18	1	trojúhelník

Tabulka 18 - Souhrn měření množiny C

5.2 Celkové výsledky sítí

Srovnání všech topologií sítí napříč učebními soubory byla seřazena podle jednotlivých kritérií.

Podle počtu rozpoznaných objektů

Pří srovnání sítí podle počtu rozpoznání dopadly nejlépe sítě C_2, B_4 a B_2 s celkovým počtem dvou rozpoznání.

Název sítě	Počet vrstev	Počet neuronů	Počet správných rozpoznání	Správně rozpoznané tvary
Síť C_2	1	8	2	kruh, čtverec
Síť B_4	2	26	2	obdélník, trojúhelník
Síť B_2	1	26	2	obdélník, trojúhelník
Síť B_1	1	13	1	čtverec
Síť C_1	1	4	1	kruh
Síť A_3	2	36	1	obdélník
Síť B_3	2	13	1	trojúhelník
Síť C_3	2	9	1	trojúhelník
Síť C_4	2	18	1	trojúhelník
Síť A_4	2	72	1	trojúhelník
Síť A_1	1	14	0	-
Síť A_2	1	28	0	-

Tabulka 19 - Celkové zobrazení podle počtu rozpoznání

Seřazení podle rozpoznaných tvarů

Při rozpoznávání objektů je možné porovnat mezi sebou sítě, které rozpoznaly stejné tvary a to pomocí přesnosti určení.

Název sítě	Počet vrstev	Počet neuronů	Přesnost určení obdélníku	Přesnost určení trojúhelníku
Síť B_4	2	26	0,0713	0,5799
Síť B_2	1	26	0,2188	0,2246

Tabulka 20 - Porovnání sítí B_4 a B_2

Při zhodnocení výsledků u sítí B_2 a B_4 viz tabulka 20 můžeme říci pouze, že síť B_4 je lepšího rozpoznání trojúhelníku a síť B_2 je lepší pro obdélník. Ani jedna z těchto sítí nedosahuje lepších výsledků u obou tvarů zároveň, tudíž nelze určit, která ze sítí je lepší.

Název sítě	Počet vrstev	Počet neuronů	Přesnost určení trojúhelníku
Síť B_3	2	13	0,8225
Síť C_3	2	9	0,7677
Síť C_4	2	18	0,2609
Síť A_4	2	72	0,0209

Tabulka 21- Rozpoznání trojúhelníku

U porovnání sítí, které rozpoznaly pouze trojúhelník, dopadla nejlépe síť z testovací množiny B a s dvěma skrytými vrstvami s třinácti neurony v obou skrytých vrstvách.

5.3 Shrnutí výsledků

Na základě naměřených výsledků nelze určit nejlepší topologii umělé neuronové sítě pro rozpoznání všech základních tvarů. Odlišné topologie v kombinaci s různými učebními soubory jsou vhodné na rozpoznání jednotlivých tvarů. Lze tedy pouze určit vhodnou trénovací množinu a topologii pro jednotlivé tvary.

6 Závěr

Cílem bakalářské práce bylo navrhnutí vhodné topologie umělé neuronové sítě pro rozpoznání základních tvarů vyfocených v travním porostu a experimentování s architekturou neuronových sítí.

V rešeršní části jsem nejprve zjistil momentální situaci v oblasti rozpoznávání. Následně jsem se zaměřil na způsoby úpravy obrazu pomocí detekce hran, architekturu neuronových sítí a jejich funkcionalitu u procesu učení.

Na základě poznatků získaných z rešeršní části práce jsem v praktické části experimentoval s učebními soubory a topologiemi neuronových sítí. Všechny naměřené výsledky byly zaneseny do tabulek pro zhodnocení výsledků měření. Pro další výzkum je potřeba experimentovat s úpravou obrazu pomocí detekce hran v kombinaci s neuronovými sítěmi. Tím se budu zabývat v navazující práci.

Cíle této bakalářské práce byly splněny.

7 Seznam použitých zdrojů

1. **WALL, Tim.** 8,74 Million Species on Earth. *News Discovery.* [Online] 23. Srpen 2011. [Citace: 20. Únor 2016.] http://news.discovery.com/earth/plants/874-million-species-on-earth-110823.htm.

2. Leafsnap. *An electronic field guide.* [Online] 2011. [Citace: 20. Únor 2016.] http://leafsnap.com/.

3. FlowerChecker. *FlowerChecker, plant identification app.* [Online] 9. Únor 2015. [Citace: 20. Únor 2016.] http://www.flowerchecker.com/cs.

4. **MOKRÝ, Vladimír, ARMAN, Adam a KADLEC, Jan.** Techsquat. *Techsquat introduction.* [Online] [Citace: 20. Únor 2016.] http://techsquat.com/.

5. Statsoft. *Úvod do neuronových sítí.* [Online] 5. Únor 2013. [Citace: 20. Únor 2016.] http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_linky.pdf.

6. **VESELÝ, Arnošt.** *Úvod do umělé inteligence.* Praha, 2005. ISBN 80-213-1361-7.

7. **VOLNÁ, Eva.** Neuronové sítě 1. *Studijní materiály.* [Online] 2008. [Citace: 20. Únor 2016.] http://www1.osu.cz/~volna/Neuronove_site_skripta.pdf.

8. **HLAVÁČ, Václav.** Hledání hran. *Fakulta elektrotechnická ČVUT.* [Online] 3. leden 2015. [Citace: 15. únor 2016.] http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/22EdgeDetectionCz.pdf.

9.**HLAVÁČ, Václav.** Zpracování signálu a obrazu. *FEL ČVUT.* [Online] 7. Prosinec 1999. [Citace: 20. Únor 2016.] http://neuron.tuke.sk/pluchta/Pocitacove%20Videnie/Prednasky/NIECO/HLAZSO.PDF.

10. Canny Edge Detection. *Cse.iitd.* [Online] 23. Březen 2009. [Citace: 20. Únor 2016.] http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf.

11. Neuroph. *Java Neural Network Framework Neuroph.* [Online] 2008. [Citace: 20. Únor 2016.] http://neuroph.sourceforge.net/.

12. **BISKUP, Roman.** Možnosti neuronových sítí. [Online] 2009. [Citace: 20. Únor 2016.] http://www.pef.czu.cz/cs/?dl=1&f=13043.

13. CS231n Convolutional Neural Networks for Visual Recognition. *Github.* [Online] [Citace: 20. Únor 2016.] http://cs231n.github.io/.

14. **ŠŤASTNÝ, Petr.** Rozpoznávávání objektů pomocí neuronových sítí. [Online] 2014. [Citace: 20. Únor 2016.] http://is.muni.cz/th/255824/fi_m/Stastny_Petr.pdf.

15. **STRAKA, Stanislav.** Segmentace obrazu. *IS Muni.* [Online] 2009. [Citace: 20. Únor 2016.] http://is.muni.cz/th/72784/fi_m/dp.pdf.

16. Survey on Various Edge Detection Techniques on Noisy Images. *Science Direct.* [Online] 2014. [Citace: 20. Únor 2016.] http://www.sciencedirect.com/science/article/pii/S221201731200312X.

17. **PÁLKA, Zbyněk.** Realizace hranového detektoru s využitím vlnkové transformace. *Vysoké učení technické v Brně.* [Online] 2009. [Citace: 20. Únor 2016.] https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=18131.