

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Rozpoznávání druhů plodnic vysokých hub z digitálních
obrazů

Vypracoval: Bc. Lukáš Pícek
Vedoucí práce: Ing. Marek Hruží Ph.D.

Květen 2015

Čestné prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

Datum: 12.5.2016

.....
podpis

Poděkování

Rád bych poděkoval vedoucímu diplomové práce Ing. Markovi Hrúzovi PhD. za jeho dlouhodobé odborné vedení, podnětné konzultace, cenné připomínky a rady poskytnuté v průběhu studia k vypracování této práce.

Také děkuji panu Rolandu Baranovičovi, spravujícímu web www.nahuby.sk, za poskytnutí databáze fotografií hub.

Dále děkuji MetaCentru za umožnění přístupu k výpočetní zdrojům, na kterých probíhala většina výpočtů. MetaCentrum je spravováno pod programem "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042).

Na závěr bych rád poděkoval mé nejdražší nastávající za poskytnutou morální podporu a ochotu kdykoliv a jakkoliv mi pomoci.

Anotace

Hlavním cílem této diplomové práce je návrh a implementace systému, jehož úkolem je automatické rozpoznávání plodnic vysokých hub z digitálních obrazů s cílem o co nejmenší odchylku za použití minima informace od uživatele. Práce prezentuje teoretické informace o vybraných metodách a následně se zabývá i jejich praktickým využitím. Konkrétněji jde o analýzu vstupních dat a jejich předzpracování, které zahrnuje automatickou segmentaci plodnice a následně i klasifikaci upravených digitálních snímků. Klasifikace se zabývá výhradně konvolučními neuronovými sítěmi jakožto state-of-the-art metodou pro rozpoznávání obrazových dat. Je prezentována stručná historie a několik již známých postupů užívaných pro optimalizaci klasifikace. Následuje výhradně praktická část, ve které je na základě experimentálně získaných výsledků, představen a aplikován postup pro rozpoznávání plodnic vysokých hub. Závěr práce se věnuje návrhu konkrétního systému sloužícího k využití v reálném světě.

Klíčová slova: OpenCV, konvoluční neuronové sítě, houby, plodnice vysokých hub, segmentace, GrabCut, Deep learning, klasifikace, SVM

Abstract

The main aim of this thesis is to design and implement a computer system which would manage automatic mushroom species recognition from digital images. High accuracy with only minimal user interaction is a must. This thesis is divided into several parts where theoretical information about selected methods and their practical usage are presented. Especially with input data analysis, data preprocessing including automatic segmentation of a mushroom and finally classification of preprocessed data. Classification is mainly focused on deep convolutional neural networks as state-of-the-art method for image data classification. History is briefly introduced together with common neural network training and classification approaches related to previous research. Followed by the practical part which introduces the main ideas for better image classification. Basically, the theoretical background is converted to practical use in the form of experiments. These experiments prove and disprove deep convolutional neural networks' theoretical knowledge on mushroom recognition topics. In conclusion, the thesis is focused on a possible and specific system which is supposed to help with recognition of mushrooms in the real world.

Keywords: OpenCV, convolutional neural networks, mushrooms, segmentation, GrabCut, classification, Deep learning, SVM

Obsah

1	Úvod	1
1.1	Motivace	2
1.2	Popis problému	3
1.2.1	Určovací znaky plodnic a obecné způsoby jejich rozpoznávání	3
1.2.2	Předzpracování dat	5
1.2.3	Klasifikace	6
2	Klasifikace obrazových dat	7
2.1	Tradiční přístup	7
2.2	Deep learning	8
2.3	Hodnocení kvality klasifikace	9
2.3.1	Precision a recall	9
2.3.2	Matice záměn	10
2.4	Rozdělení dat	11
2.5	Cross-validation	12
2.6	Support Vector Machine	12
2.6.1	Kernel Trick	13
3	Konvoluční neuronové sítě	15
3.1	Historie	15
3.2	Charakteristika	16
3.3	Architektura	17
3.3.1	Konvoluční vrstva	18
3.3.2	Poolingová vrstva	18
3.3.3	Fully Connected vrstva	18
3.3.4	Loss vrstva	19
3.4	Trénování	19
3.4.1	Pre-training	20

3.4.2	Fine-tuning	20
3.4.3	Backpropagation	20
3.5	Hledání optimálních parametrů	21
3.5.1	Stochastic Gradient Descent	21
3.5.2	Adaptivní metody	23
3.6	Best practice	24
3.6.1	Aktivační funkce	24
3.6.2	Overlapping Pooling	25
3.6.3	Local Response Normalization	25
3.6.4	Data Augmentation	25
3.6.5	DropOut	26
4	Segmentace	27
4.1	Metody segmentace	28
4.1.1	Prahování	28
4.1.2	Segmentace na základě detekce hran	29
4.1.3	Hledání oblastí	31
4.1.4	Srovnávání se vzorem	32
4.1.5	Mean-Shift	33
4.1.6	GrabCut	34
5	Návrh řešení a implementace	35
5.1	Dataset	36
5.1.1	Vzorová data	36
5.1.2	Získání dat	37
5.2	Předzpracování dat	37
5.2.1	Transformace dat	38
5.2.2	Normalizace	39
5.3	Segmentace	39
5.3.1	Výběr metody	40
5.3.2	Testování metody	41
5.4	Augmentace	42
5.5	Velikost dat	43
5.6	RGB vs Gray	44
5.7	CNN vs SVM	45
5.8	Klasifikace pomocí CNN	46

5.8.1	Caffe framework	46
5.8.2	Výběr parametrů sítě	46
5.8.3	Trénování	48
5.9	Návrh a zpracování mobilní aplikace	50
5.9.1	Navržené řešení	50
5.9.2	Výsledná aplikace	51
5.9.3	Použité technologie	51
6	Možnosti rozšíření a vylepšení	53
7	Závěr a zhodnocení	55
8	Přílohy	60

Seznam obrázků

1.1	Obecné znaky plodnic vysokých hub. Převzato z [2].	3
1.2	Různé tvary třeně (a) a klobouků (b) u plodnic vysokých hub. Převzato z [2].	4
2.1	Schéma klasifikace využívající Deep learning a vizualizace některých úrovní příznaků. Převzato z [3].	8
2.2	Vyjádření hodnot precision a recall formou grafického zobrazení. Převzato z [4].	10
2.3	Ilustrace možných rozdělení dat do trénovací a testovací sady. . .	12
2.4	Grafické znázornění Kernel Tricku, respektive transformace dvou- rozměrných nelineárně separovatelných dat do vyšší dimenze, která nám umožňuje lineární rozdělení. Převzato z [5].	13
3.1	Příklad využití neuronové sítě k tématickému přemalování ob- rázku. Převzato z [10].	16
3.2	Ilustrace možných rozdělení dat do trénovací a testovací sady. . .	17
3.3	Ilustrace max-poolingu. Obraz je rozdělen do 4 stejných regionů. Nalezená maxima jsou zvýrazněna červenou barvou.	19
3.4	Grafické znázornění získávání hodnoty Momenta a Nesterova Mo- menta. Převzato z [13].	22
3.5	Levý obrázek reprezentuje běžnou neuronovou síť se dvěma skry- tými vrstvami. Na pravém obrázku je zobrazena stejná architektu- ra s využitím Dropoutu. Převzato z [17].	26
4.1	Výsledek prahování obrázku (a), prezentovaný jako binární obraz (b), kde černá barva reprezentuje pozadí a bílá popředí.	29
4.2	Ukázka různých tvarů hran, kde g reprezentuje úroveň jasu a x jednotlivé pixely v řadě.	30

4.3	Příklad použití quad-tree struktury.	32
4.4	Příklad konvergence k lokálnímu maximu při použití metody Mean-Shift.	33
4.5	Příklad funkce metody GrabCut, kde (a) reprezentuje označení pozadí a popředí na obrazových dat a (d) výsledný binární obraz, kde černá barva reprezentuje pozadí a bílá popředí. Převzato z [22].	34
5.1	Ilustrace ideálního formátu trénovacích a testovacích dat pro různé druhy hub.	36
5.2	Různé druhy transformace obrazu do čtvercové podoby. Obrázek (a) reprezentuje roztažení, obrázek (b) zrcadlení a v poslední řadě (c) reprezentuje nastavení snímku o černé okraje.	38
5.3	Výsledky segmentace při použití metody GrabCut (a) a prahování (b).	40
5.4	Zobrazení augmentovaných dat vytvořených pomocí rotace a posunutí reálného segmentu plodnice.	42
5.5	Normalizovaná matice záměn znázorňující mezitřídní chybovost klasifikace. Maximální hodnota 1 reprezentuje 100% úspěšnost klasifikace.	49
5.6	Grafické znázornění hodnot Accuracy a Loss získané v průběhu klasifikace.	50
5.7	Uživatelské rozhraní vytvořené iOS aplikace. Levý obrázek ilustruje označení popředí a pozadí pro segmentační metodu GrabCut. Pravý obrázek zobrazuje vysegmentovanou plodnici a výsledek klasifikace.	51

Seznam tabulek

2.1	Čtyři způsoby zaklasifikování umožňující směrodatnější vyhodnocení kvality klasifikátoru.	9
2.2	Příklad normalizované matice záměn v binární úloze. Hodnoty v tabulce reprezentují procentuální zařazení.	11
2.3	Nejčastěji využívaná nelineární jádra Kernel Triku pro metodu SVM.	13
5.1	Četnost jednotlivých druhů (tříd) obsažených v základním datasetu.	37
5.2	Získané hodnoty kvality klasifikace jednotlivých metod transformace obrazu.	39
5.3	Získané hodnoty při klasifikaci rozdílných vstupních dat pomocí CNN. Experiment podporující využití segmentace.	41
5.4	Četnost jednotlivých druhů hub obsažených v datasetu před a po augmentaci.	42
5.5	Získané hodnoty při klasifikaci rozdílných vstupních dat pomocí CNN. Experiment podporující využití augmentace.	43
5.6	Tabulka reprezentující cross-validací získané hodnoty přesnosti a hodnotící funkce. První řádek reprezentuje velikost hrany vstupního obrázku. Poslední řádek reprezentuje průměr získaný z 10 iterací cross-validace.	43
5.7	Získané hodnoty při klasifikaci rozdílných vstupních dat pomocí CNN. Experiment podporující využití RGB snímků.	44
5.8	Získané hodnoty při klasifikaci segmentovaných snímků pomocí klasifikátoru CNN a SVM.	45

Seznam zkratek

CNN - Convolutional Neural Networks

DNN - Deep Neural Networks

SVM - Support Vector Machine

ReLU - Rectified Linear Units

RBM - Restricted Boltzmann Machines

SGD - Stochastic Gradient Descent

CPU - Central Processing Unit

GPU - Graphic Processing Unit

RGB - Red Green Blue

OS - Operační Systém

UI - User Interface

Kapitola 1

Úvod

V současnosti, zejména díky dostupnosti spotřební elektroniky, je zcela běžné narazit na různé podpůrné systémy umělé inteligence, mající za úkol asistovat, pomáhat, popřípadě nahradit člověka v běžných, ale i specifických vědeckých úkonech. Jedním z těchto systémů jsou i takzvané systémy expertní. Ty jsou využívány především ve zdravotnictví, ale zcela běžně je možné na ně narazit i v dalších oborech jako jsou mechanika, technická diagnostika a finančnictví. Jejich hlavním úkolem je poskytování rad, rozhodnutí, nebo doporučení tak jak by je doporučil člověk/expert kdyby byl dotázán.

Dalším fenoménem souvisejícím se strojovým učením, jakožto částí umělé inteligence a zároveň state-of-the-art metodou pro klasifikaci obrazu, jsou konvoluční neuronové sítě. V současnosti jsou využívány v mnoha rozdílných odvětvích, přičemž nejčastěji je můžeme nalézt v oblastech souvisejících s rozpoznáváním obrazu a analýzou videa, respektive s počítačovým viděním. Konkrétně se jedná o úlohy, jako je rozpoznávání textu, obličejů, kompozice snímku a dalších informací, zjistitelných z analyzovaného obrazu či videa. Škála jejich využití je opravdu široká, nicméně v kontrastu této práce jsou zajímavé právě díky jejich schopnostem v oblasti rozpoznávání obrazu.

1.1 Motivace

Dle údajů českého statistického úřadu vyrazí každý rok za sběrem hub 70 % obyvatel České republiky. Velké oblibě houbaření se těší i ostatní země střední a východní Evropy. Počet druhů vysokých hub, které v těchto zemích mohou být nalezeny, je odhadován na více jak 3000. Z toho je přibližně 300-400 druhů považováno za nejedlé nebo jedovaté, a méně než 10 za smrtelně jedovaté [1]. I přes nízké procento z celkového počtu druhů, které se dá považovat za jedovaté, dochází každý rok v České republice průměrně k 300 případům přiotrávení a několika úmrtím způsobených jejich konzumací. Velká část těchto případů je přikládána výhradně nedostatečné mykologické znalosti při jejich sběru. Nasnadě je tedy otázka, jak co nejlépe zvýšit povědomí běžných houbařů o jednotlivých druzích, popřípadě zjednodušit jejich rozpoznávání. Protože je tato práce zpracovávána na závěr studia řídicích a rozhodovacích systémů, můžeme možnost dodatečného vzdělávání ihned vyloučit a soustředit se zejména na podpůrné počítačové systémy.

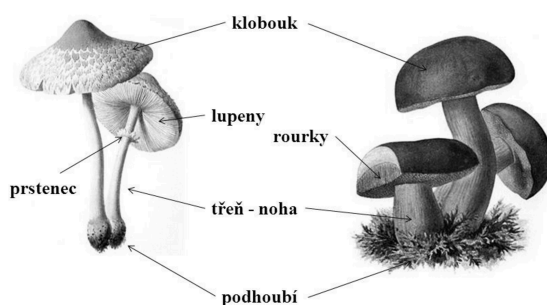
Z pohledu umělé inteligence lze k otázce zjednodušení rozpoznávání přistupovat různými způsoby. Jedním z nich je možnost již zmíněná v úvodu a tou je sestavení expertního systému. V současnosti existuje několik webových projektů, které se rozpoznáváním za pomoci raných expertních systémů zabývají. Jejich největší výhodou je schopnost objektivně a relativně rychle vyřešit daný problém, a to s větší dostupností, než by takovéto činnosti byl schopen člověk. Bohužel jejich ovládání je ve většině případů pro běžného uživatele zbytečně složité a velmi zdlouhavé. Tyto a další nevýhody expertních systémů nám umožní odstranit řešení spočívající ve využití konvolučních neuronových sítí. Jejich vysoká flexibilita nám nabízí přijatelné řešení, na základě kterého lze očekávat kvalitní výsledky za využití jednoduché uživatelské interakce. Touto interakcí je myšleno pořízení snímku plodnice jakýmkoliv zařízením opatřeným fotoaparát a jeho nahráním do natrénované sítě, která je následně schopna uživateli prezentovat výsledky klasifikace.

1.2 Popis problému

Rozpoznávání hub, při existenci velkého množství druhů a vysoké podobnosti mezi nimi, je velice složitou úlohou i pro zkušeného mykologa. Při přenesení do-
tématiky zpracování obrazu za předpokladu, že plodnice jednoho druhu mohou
během svého životního cyklu nabývat rozličných tvarů, barev a podob, nabývá
tato problematika ještě složitější rozměr. Je tedy nutné podívat se na problema-
tiku rozpoznávání hub nejprve z obecného pohledu a až následně na jeho možné
řešení pomocí některých z konvenčních postupů a metod počítačového vidění.
V závislosti na zmíněné složitosti rozpoznávání je potřeba brát v potaz, že vý-
sledný počítačový systém nemusí být vždy 100 % přesný. Je třeba na něj nahlížet
spíše jako na užitečnou pomůcku, která má za úkol zjednodušit uživateli rozpo-
znávání jednotlivých druhů, nikoliv jako na bezchybný zdroj informací.

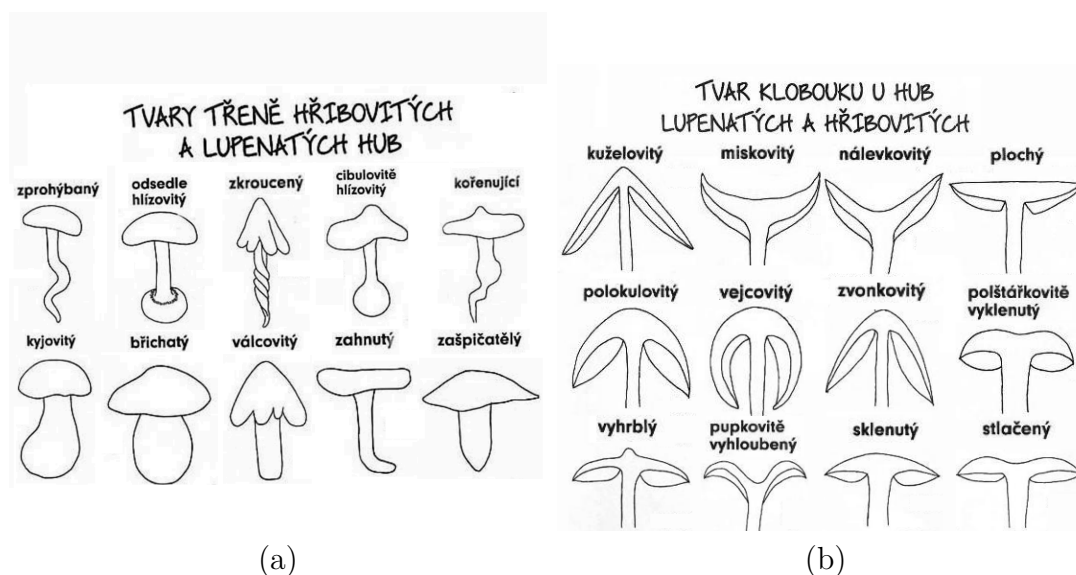
1.2.1 Určovací znaky plodnic a obecné způsoby jejich roz- poznávání

Jelikož navrhovat systém pro rozpoznávání obrazu bez alespoň základních
znalostí dané tematiky je výrazněji komplikované, bude nejprve představena obe-
cná mykologická teorie potřebná pro porozumění problematice rozpoznávání hub.
V této části bude pojednáváno o základních vlastnostech tzv. vysokých hub, které
jsou charakteristické nápadnými plodnicemi, a také o jejich obecných způsobech
rozlišování.



Obrázek 1.1: Obecné znaky plodnic vysokých hub. Převzato z [2].

Vlastní tělo houby se většinou skládá z bezbarvé spleti tenkých vláken nazývaných hyfy. Soubor těchto hyf je označován jako mycelium, nebo také podhoubí. To za příznivých podmínek vytváří plodnice, které slouží výhradně k rozmnožování a tedy šíření hub. Rozmnožování je umožněno prostřednictvím sporů, uvolňovaných z výtrusných lůžek. Ty se mohou v závislosti na druhu vyskytovat na velmi rozdílných místech. Na základě jejich tvaru můžeme plodnice rozdělit do dvou skupin, a to konkrétně na ty s rourkami a s lupeny. Obecné příklady obou druhů jsou viditelné na Obrázku 1.1 spolu s ostatními částmi plodnice. Mezi další významné části se řadí třen a klobouk. Ty mohou nabývat různorodých tvarů a díky tomu výrazně napomáhají s určováním konkrétních druhů. Různé druhy třeně a klobouků jsou znázorněny na Obrázku 1.2.



Obrázek 1.2: Různé tvary třeně (a) a klobouků (b) u plodnic vysokých hub. Převzato z [2].

Další určovací znaky plodnic, respektive hub, se z velké části týkají prostředí a ročního období, ve kterém se běžně nacházejí. Plodnice rostoucí mimo jejich běžné prostředí a časové období je možné nalézt velice zřídka. I když jsou tyto znalosti v praxi minimálně využívané, jedná se o důležité informace, které mohou rozlišit, zdali se jedná o houbu jedlou, popřípadě nejedlou. V neposlední řadě je možné získat další údaje z barvy výtrusového prachu, vůně a chuti plodnice.

Z pohledu počítačového vidění můžeme brát v potaz pouze ty příznaky, které jsou viditelné na první pohled a tedy zachytitelné pomocí digitálního snímku. Je zcela zřejmé, že informace ohledně ročního období, prostředí, chuti a vůně je takto možné získat pouze velice složitě. Z tohoto důvodu je úloha rozpoznání hub v kontextu umělé inteligence výrazně ztížena. Mezi nejrelevantnější informativní příznaky, potřebné pro klasifikaci, se v tomto případě řadí zejména:

- barva a tvar klobouku,
- barva a tvar třeně,
- existence prstence na třeni,
- druh výtrusného rouška udávající jestli jde o houbu s rourkami, nebo lupeny.

Důležitost jednotlivých příznaků je zřejmá, nicméně počítačový systém, který se stará o řešení úlohy, by měl být dostatečně robustní při klasifikaci. Měl by tedy být schopen klasifikovat i takový snímek, kde chybí jeden, nebo více příznaků. Nutno podotknout, že v některých případech může mít absence některých základních znaků fatální důsledky na výsledek klasifikace, a z tohoto důvodu je doporučeno poskytnout klasifikátoru co největší množství informací.

1.2.2 Předzpracování dat

Přestože jakékoliv úpravy formou předzpracování dat evokují značnou ztrátu informací, mohou mít zároveň velmi pozitivní vliv na výsledek řešené úlohy. Je třeba se snažit o takové úpravy, které potlačí nepodstatnou část dat a zároveň zvýrazní tu potřebnou. V současnosti, u většiny běžných úloh v oblasti počítačového vidění, dochází alespoň k minimálním úpravám zpracovávaných dat. Mezi ty se řadí především:

- jasové a geometrické transformace,
- morfologické operace,
- detekce nebo zvýrazňování hran,
- segmentace.

Po prvotním výběru informativních příznaků zmíněných v předešlé podkapitole by se mohlo jevit, že předzpracování dat v problematice rozpoznávání hub by mohlo mít pouze negativní vliv na výsledek. Je tedy nutné experimentálně ověřit, zdali po různých úpravách dojde ke zlepšení, popřípadě zhoršení ve výsledcích.

1.2.3 Klasifikace

Nejpodstatnější částí potřebné v úlohách rozpoznávání obrazu je klasifikace. Jejím účelem je nejprve stanovit podobnost mezi trénovacími daty a následně poskytnout informace ohledně zařazení neviděných dat do některé z tříd. K trénování klasifikátorů můžeme přistupovat dvěma způsoby, a to jako k trénování s učitelem a bez učitele.

Při aplikování některé z metod trénování bez učitele není známa žádná apriorní informace o vstupních datech, a proto se klasifikátor mezi nimi snaží najít určitou podobnost. Tím je myšleno, že vstupní data jsou například na základě Euklidovské vzdálenosti příznakových vektorů rozdělena do několika tříd. Na druhou stranu při učení s učitelem má klasifikátor k dispozici apriorní informaci ohledně jejich zařazení, a cílem klasifikátoru je určitým způsobem zobecnit pohled na jednotlivé třídy tak, aby minimalizoval chybné zařazení u neviděných dat. Pro klasifikaci a trénování s učitelem je možné v úloze rozpoznávání hub využít některé již konvenčně využívané postupy a metody. Mezi ty se řadí především klasifikátor Support vector machines (SVM) a konvoluční neuronové sítě. Obě metody budou blíže popsány v jedné z následujících kapitol a zároveň podrobeny klasifikačnímu experimentu.

Kapitola 2

Klasifikace obrazových dat

V úlohách rozpoznávání obrazu je problém klasifikace definován jako problém identifikace neznámých dat k některému ze souboru známých kategorií. Algoritmus provádějící klasifikaci se nazývá klasifikátor. Při jeho trénování se v úlohách rozpoznávání obrazu, jakožto součásti strojového učení, pracuje výhradně s expertně označenými daty (apriorní informací). Ta nám udávají k jaké kategorii jednotlivé snímky patří, a tím nám zařazují učení klasifikátoru do úloh klasifikace s učitelem. Dále lze k úloze rozpoznávání obrazových dat přistupovat tradičním způsobem s extrakcí nízkoúrovňových příznaků, nebo jako k tzv. Deep learningu zaměřenému na generování vysokoúrovňových příznaků.

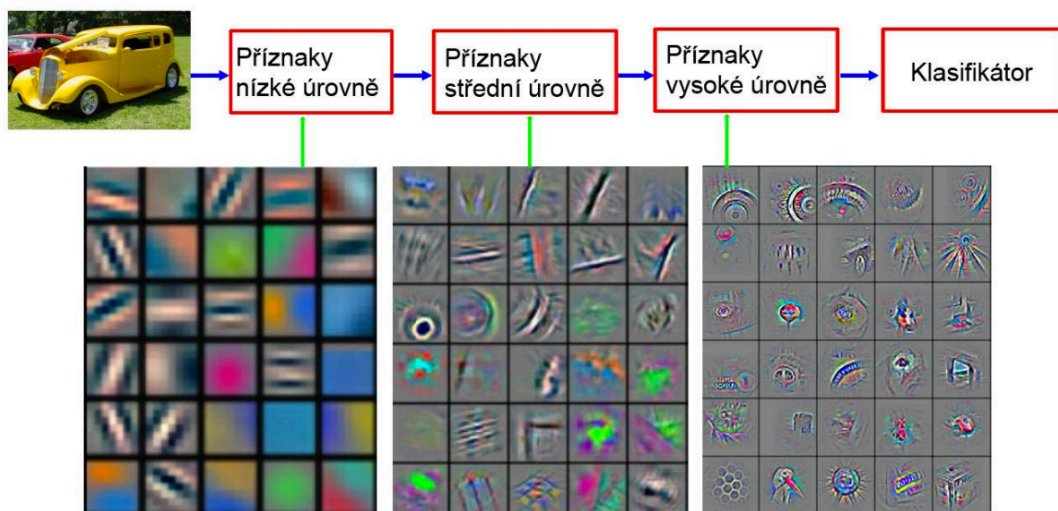
2.1 Tradiční přístup

Tradiční způsob klasifikace obrazových dat založený na extrakci příznaků lze rozdělit do několika kroků. Nejprve je možné, nikoli nutné, využít některých z postupů předzpracování dat tak, aby došlo ke zvýraznění podstatných částí obrazu. Následně je z takto upraveného obrazu (odstranění šumu, normalizace atd.) možné získat charakteristické znaky ve formě příznaků. K jejich extrakci lze využít některé z množství metod (SIFT, SURF, HoG atd.). Všechny nalezené příznaky spolu tvoří vektor příznaků, který je využíván v posledním kroku, a to klasifikaci. Ta vektoru příznaků přiřadí jednu z výstupních tříd.

2.2 Deep learning

Při zpracování obrazových dat se nejčastěji pracuje s jeho maticovou formou. Z té je následně možné vygenerovat různé druhy příznaků, pomocí kterých můžeme analyzovaná data lépe popsat. K podrobnému popisu je důležité mít mimo nízkourovňových příznaků i ty vysokoúrovňové. K jejich generování slouží množina algoritmů strojového učení zvaná Deep learning. Ta k jejich získání používá modely architektur složené z několika nelineárních transformací. Hlavním cílem Deep learningu je tedy automaticky vyhledávat příznaky určené pro popis, nebo detekci obrazových abstrakcí.

Jelikož je každá úloha svým způsobem specifická, je důležité generovat příznaky všech úrovní, tak jak je zobrazeno na Obrázku 2.1. Z nich je poté možné určit konkrétní abstrakci. Pro příklad můžeme vycházet z níže uvedeného obrázku, reprezentujícího automobil. Specifickou kombinací získaných příznaků je možné určit, že konkrétní abstrakce je "žlutý automobil". Další možností ale může být i "červený automobil", nebo "louka". Ačkoliv využití Deep learningu ke generování příznaků je velice cenný a funkční nástroj, je nutné zmínit i jeho nevýhodu. Ta spočívá v tom, že vygenerované příznaky nemusí být v kontextu úlohy dostatečně relevantní a zároveň není možné je jednoduše k těm relevantním nasměrovat.



Obrázek 2.1: Schéma klasifikace využívající Deep learning a vizualizace některých úrovní příznaků. Převzato z [3].

2.3 Hodnocení kvality klasifikace

Pro zhodnocení kvality klasifikace nelze vycházet pouze z procentuální hodnoty reprezentující správné zařazení dat, ale je důležité brát v potaz i hodnoty získané pro jednotlivé třídy. Výsledná procentuální hodnota klasifikátoru nemusí vždy reflektovat funkční klasifikátor, a to zejména díky nerovnoměrnému zastoupení dat. Pokud je tedy v některé třídě výrazně méně dat než v ostatních a zároveň u ní dojde k relativně velkému procentu nekorektní klasifikace, bude tato chyba potlačena velkou úspěšností ostatních tříd. Z tohoto důvodu jsou zavedeny čtyři druhy zaklasifikování viditelné v Tabulce 2.1.

Actual Value	0	0	1	1
Predicted	0	1	0	1
	True negative (TN)	False positive (FN)	False negative (FP)	True positive (TP)

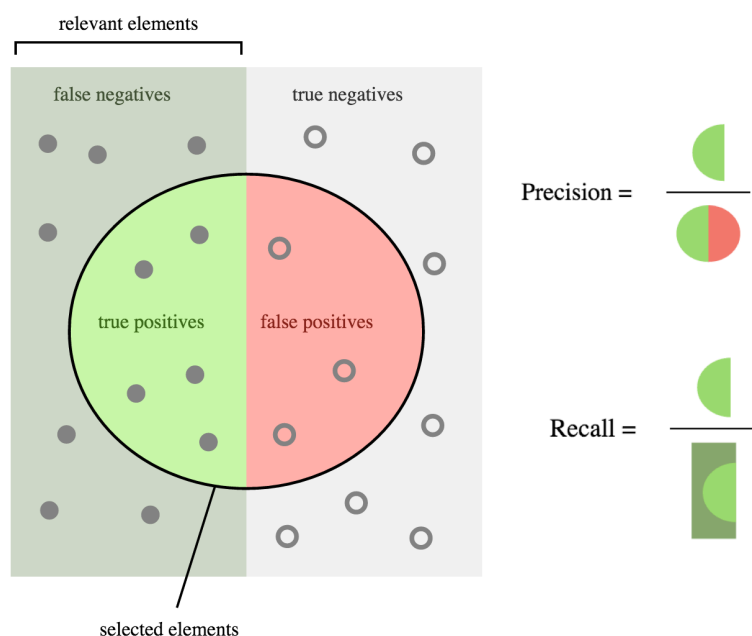
Tabulka 2.1: Čtyři způsoby zaklasifikování umožňující směřovatější vyhodnocení kvality klasifikátoru.

2.3.1 Precision a recall

Jelikož pracovat s pouhým průměrem není při analýze výsledků racionální, rozlišují se pro predikovaná data čtyři způsoby zaklasifikování. Tyto způsoby jsou viditelné v Tabulce 2.1. Častěji se ale setkáváme s hodnotami precision (přesnost) a recall (úplnost), z jejichž pomocí jsme schopni úspěšnost klasifikace vyjádřit jednoznačněji. Jejich výpočet závisí na četnosti správných a chybných klasifikací. Hodnota precision odpovídá pro binární úlohu klasifikace poměru všech správně klasifikovaných dat jedné třídy ku všem klasifikovaným k této třídě. Na druhou stranu recall dodává informace ohledně počtu správně klasifikovaných ku všem, které měly být správně klasifikované. Pro lepší představu je jejich výpočet znázorněn formou Obrázku 2.2 a Vzorců ve 2.1, kde P reprezentuje hodnotu precision a R hodnotu recall.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad F_1 = 2 \frac{PR}{P + R} \quad (2.1)$$

Aplikací těchto metrik na klasifikační úlohu obdržíme hodnoty, které nám popisují kvalitu klasifikátoru výrazně lépe než průměr. Protože posuzování kvality systému pomocí dvou čísel není dostatečně zřetelné, byla stanovena ještě jedna metrika zvaná jako F1 skóre. Tato metrika kombinuje výsledky precision a recall tak, že kvalitu klasifikátoru popisuje mnohem věrohodněji než jejich pouhý aritmetický průměr. Pomocí vzorce ve 2.1 získáme jednu konkrétní hodnotu, díky níž lze velice snadno a rychle porovnat úspěšnost více systémů.



Obrázek 2.2: Vyjádření hodnot precision a recall formou grafického zobrazení. Převzato z [4].

2.3.2 Matice záměn

Další možný způsob posouzení kvality klasifikace je pomocí matice záměn (Tabulka 2.2), která sleduje počty správně a nesprávně zařazených dat. Pomocí této matice je relativně snadno zjistitelné, mezi kterými třídami dochází k chybám nejčastěji, a tím odhalit potenciální klasifikační problém. V řadě úloh může být důležitý i typ chyby, kterého se systém dopustil (FP vs. FN). To platí zejména v kontextu rozpoznávání hub, kdy záměna jedlé houby za nejedlou nemá na rozdíl od opačné záměny fatální následky.

Skutečnost, že různé chyby mohou být jinak závažné, lze předejít využitím tzv. matice cen. Ta uvádí ceny pro různé typy rozhodnutí. Čím větší chyba, tím vyšší cena. Při hodnocení znalostí se tedy nemusí brát v úvahu prostý počet chybných rozhodnutí, ale i případný vliv na uživatele.

Zařazení	Klasifikace	
	Jedlá	Nejedlá
Jedlá	0.95	0.05
Nejedlá	0.02	0.98

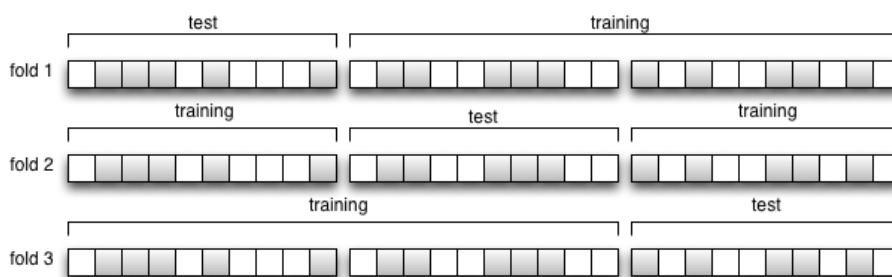
Tabulka 2.2: Příklad normalizované matice záměn v binární úloze. Hodnoty v tabulce reprezentují procentuální zařazení.

2.4 Rozdělení dat

Jedním z předpokladů správně fungujícího klasifikátoru je jeho obecné popsání jednotlivých tříd. Při jeho trénování se hledá takové nastavení, aby minimalizovalo chybné klasifikace u dat obsažených v trénovací sadě. Je však nutné brát v potaz i možnost přetrénování, kdy jsou parametry klasifikátoru uzpůsobeny a nastaveny výhradně ke klasifikaci již viděných dat. To znamená, že klasifikátor dokáže velmi dobře klasifikovat trénovací data, avšak selhává při klasifikaci dat neviděných. Dochází tak ke ztrátě schopnosti zobecňovat. Z tohoto důvodu se zavádí takzvaná zpětná validace, která nám částečně umožňuje snížit riziko přetrénování. Datová sada je rozdělena do trénovací a testovací části. Následně se provádí trénování na trénovací části a poté ověření kvality klasifikace na testovacím setu. Toto ověření však nemusí být vždy směrodatné. Pokud na sebe data budou částečně navazovat a budou i relativně podobná, může se dosáhnout dobrých testovacích výsledků, které bohužel nebudou vypovídat o kvalitě klasifikátoru. Z tohoto důvodu se často využívá křížová validace, u které dochází k opakovanému rozdělování dat a trénování.

2.5 Cross-validation

Křížová validace se provádí pro posouzení natrénovaného klasifikátoru, který by měl co nejvíce zobecňovat model jednotlivých tříd. Tato metoda je založena na opakovaném dělení trénovacích dat do trénovací a testovací část, s kterými je následně prováděno trénování. Tímto získáme několik výsledků, které nám při jejich porovnání udávají, jak se tento klasifikátor bude chovat v praxi, respektive jak bude přesný při klasifikaci neviděných dat. Porovnání je možné provést například za pomoci průměru hodnot získaných z jednotlivých iterací. Pokud bude směrodatná odchylka vykazovat extrémní rozdíly, je zřejmé, že natrénovaný klasifikátor jednotlivé třídy dostatečně nezobecnil, a z tohoto důvodu bude mít problémy při klasifikaci neviděných dat.



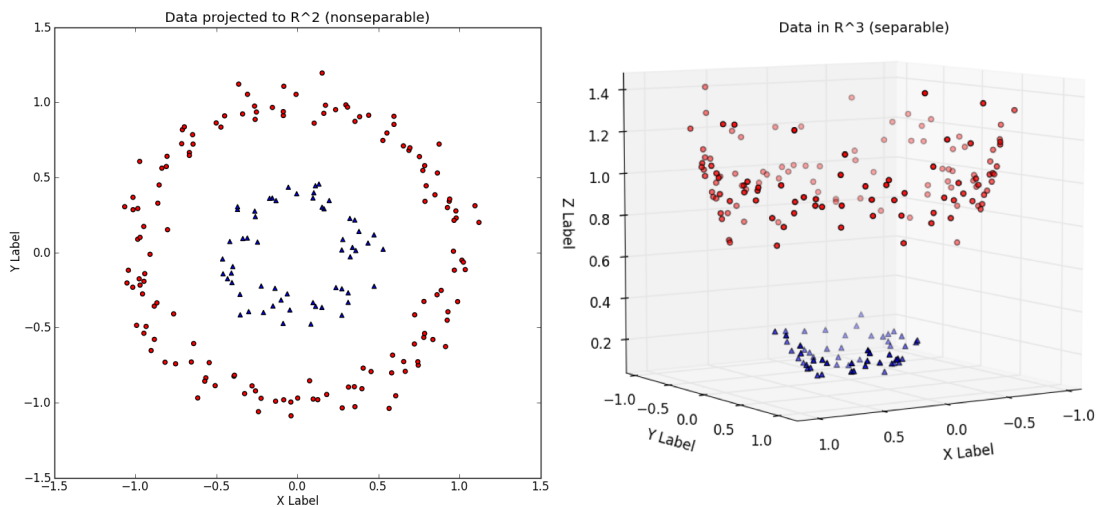
Obrázek 2.3: Ilustrace možných rozdělení dat do trénovací a testovací sady.

2.6 Support Vector Machine

Support Vector Machine (SVM) je jedna z často využívaných klasifikačních metod. Umožňuje najít nadrovinu, která od sebe vzájemně oddělí dvě třídy tak, aby jejich vzdálenost od nalezené nadroviny byla co největší a tím došlo k optimálnímu rozdělení. Jedná se tedy o typickou úlohu binárního dělení. SVM je ale možné využít také ke klasifikaci do více tříd. V tomto případě se nehledá jedna, ale několik nadrovin, které se snaží rozdělit jednu konkrétní třídu od všech ostatních, popřípadě vždy dvě třídy od sebe. Získat ale takové nadroviny je možné pouze u lineárně separovatelných množin. Bohužel nalézt takové množiny v některé z úloh reálného světa je možné pouze v ojedinělých případech. Jelikož matematické řešení nelineárně separovatelných úloh je velice složité, zavádí se takzvaný Kernel Trick.

2.6.1 Kernel Trick

Cílem klasifikační metody SVM je lineární rozdělení trénovací množiny do dvou nebo více tříd. Jak již bylo řečeno, velice často není tento postup realizovatelný. Za tímto účelem se využívá tzv. Kernel Trick, který nám umožňuje transformovat příznaky do typicky mnohem vyšší dimenze, která lineární separaci umožňuje. Grafické znázornění této transformace je zobrazeno na Obrázku 2.4.



Obrázek 2.4: Grafické znázornění Kernel Tricku, respektive transformace dvourozměrných nelineárně separovatelných dat do vyšší dimenze, která nám umožňuje lineární rozdělení. Převzato z [5].

Kernel Trick se provádí pomocí specifické funkce (kernelu) z Tabulky 2.3 a podpůrných vektorů. Ty hledají takové body na okrajích rozdělující nadroviny jsoucí nejbližší. Výsledný klasifikační model je poté tvořen vektory (support vectors) z těchto bodů. Typicky používanými jádry jsou Gaussovské, polynomiální a Sigmoidální.

Jádro	Předpis	Poznámka
Sigmoidální	$K(x_i, x_j) = K(x_i x_j + \theta)^d$	d a θ voleno uživatelem
Polynomiální	$K(x_i, x_j) = \tanh(\eta x_i x_j + \theta)$	η a θ voleno uživatelem
Gaussovské	$K(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2} \ x_i - x_j\ ^2\right)$	σ voleno uživatelem

Tabulka 2.3: Nejčastěji využívaná nelineární jádra Kernel Triku pro metodu SVM.

Jejich vhodnou volbu lze u metody SVM docílit správného namapování trénovacích vektorů do prostoru, ve kterém jsou následně aplikovány principy hledání klasifikační nadroviny, jenž má maximální možný odstup od nově namapovaných tříd. Na druhou stranu volbu funkce jádra nelze zobecnit. Obvykle se testují iterativně, přičemž je vybráno jádro vykazující nejmenší chybu klasifikace. Často se ovšem doporučuje toto testování odstartovat s jádrem Gaussovským.

Kapitola 3

Konvoluční neuronové sítě

V této části práce se budu zabývat speciálním druhem neuronových sítí, který zejména díky vývoji v oblastech počítačového hardwaru a mikroelektroniky zažívá v posledních letech renesanci. Tento speciální druh neuronových sítí se nazývá konvoluční (dále jako CNN¹) a využívá se zejména v oblasti počítačového vidění, konkrétně rozpoznávání obrazu.

3.1 Historie

První zmínky o konvoluci související s využitím neuronové sítě, se datují již od roku 1989, kdy v článku [6] prezentoval Y. LeCun a kolektiv využití Backpropagation algoritmu k získání hodnot konvolučních jader tak, aby konvoluce vstupního obrazu s natrénovanými jádry fungovala jako extrakce pouze potřebných obrazových příznaků vyšší úrovně. Tyto příznaky byly následně za pomoci neuronové sítě použity ke klasifikaci znaků z ručně psaných poštovních směrovačích čísel.

V následujících letech byl koncept CNN představen podrobněji obzvláště prostřednictvím Y. LeCuna v článcích zaměřujících se již konkrétně na problematiku CNN [7], [8], [9]. Bohužel výsledky a možnosti využití byly velice závislé na základě již zmíněné výpočetní náročnosti. Zlom přišel až po velkém úspěchu týmu z univerzity v Torontu, který v roce 2010 vyhrál známou soutěž ILSVRC (ImageNet Large Scale Visual Recognition Challenge) zaměřující se na klasifikaci enormního množství dat (více jak 1,2 milionu obrázků a 1000 tříd). Spolu s tímto

¹Convolutional Neural Network

úspěchem publikovali článek [10], v němž popsali nový přístup k trénování CNN využívající některých již dříve popsaných postupů a zejména využití dvou grafických karet, kterým výrazně zkrátily dobu trénování. Podrobnějšímu rozboru použitých přístupů se věnuje jedna z následujících sekcí.

V posledních měsících mohlo širokou veřejnost zaujmout speciální použití CNN v programu AlphaGO. Ten byl navržen a implementován za účelem porážení světového velmistra hry GO² a jako první počítačový program uspěl. Dalším příkladem může být i využití v oblasti umění, konkrétně malby. Tam umožňují transformaci obrazových dat do podoby, jakou by ho ztvárnil některý z proslulých malířů.



Obrázek 3.1: Příklad využití neuronové sítě k tématickému přemalování obrázku. Převzato z [10].

3.2 Charakteristika

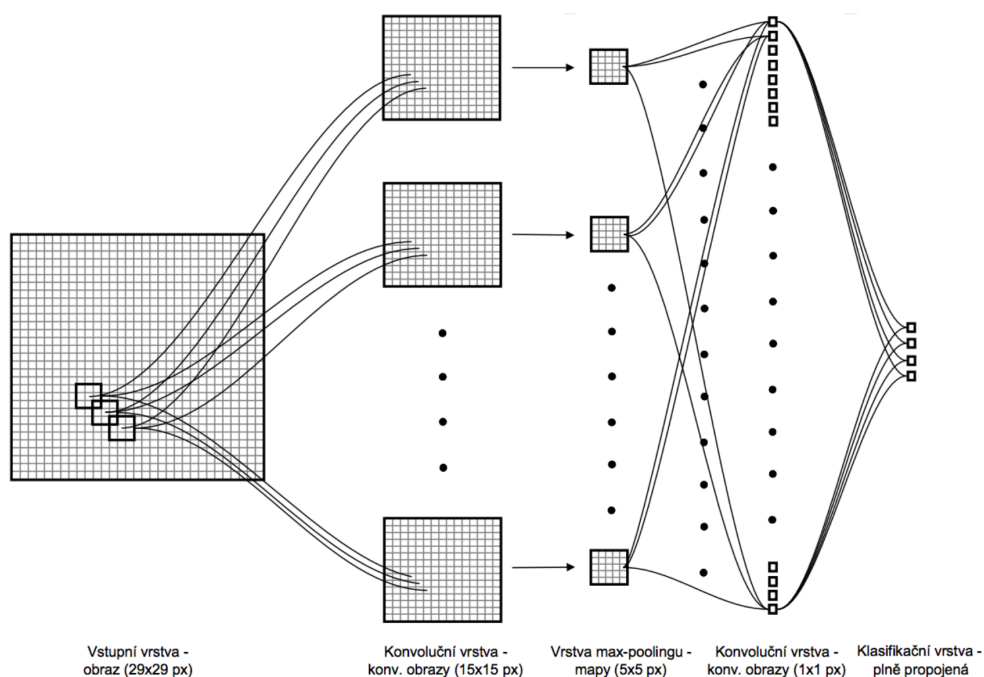
CNN jsou nejnovějším typem dopředných neuronových sítí vytvořeným a využívaným především v oblasti počítačového vidění. Jejich koncept je postavený na biologickém základu, konkrétně na biologické funkci mozku kočky. Ve studii [12], kde zkoumali zrakové centrum mozku odhalili, že na některé tvary objektů reagují vždy stejné shluky neuronů. Při podrobnější analýze zjistili, že jednotlivé neurony řazené hierarchicky do kaskád zrakového centra mozku opakovaně reagují na konkrétní příznaky vyskytující se v obraze. Fungují tedy jako detektory.

Základním kamenem CNN je použití konvoluce k získání vstupních hodnot pro neurony v následujících vrstvách sítě, na rozdíl od neuronových sítí, kde se k zís-

²Hra vysoké obtížnosti a nesčetných herní možností považována za velice obtížnou zkoušku pro umělou inteligenci.

kání těchto hodnot využívají aktivační funkce jednotlivých neuronů. Na vstupu CNN tedy dochází ke zpracování obrazových dat namísto příznakového vektoru. Tato vstupní vrstva se dle její funkce jmenuje konvoluční a je možné ji pro získání příznaků s větší informativní hodnotou zřetězit. Často jsou ale mezi ně vkládány takzvané poolingové vrstvy. Ty jsou schopny zajistit jistou invarianci vůči posunu a drobným zkreslením obrazu. Zároveň s tím redukuje množství "nepotřebných" dat. Jedná se tedy o pomyslné filtry, které propouští pouze podstatnou část informace.

Hlavním cílem je natrénovat síť (respektive hodnoty kernelů využívaných ke konvoluci) tak, aby reagovala na některé důležité změny v obraze. Tímto získáme neurony, které budou přesně jako v mozku kočky reagovat na konkrétní objekty. Na jejich základě je následně možné klasifikovat neznámá data.



Obrázek 3.2: Ilustrace možných rozdělení dat do trénovací a testovací sady.

3.3 Architektura

Architektura CNN je formována pomocí několika rozdílných vrstev, které převádějí vstupní data na výstup reprezentující informaci ohledně umístění do tříd. Ačkoliv je možné se setkat s mnohými speciálně navrženými vrstvami, jen ně-

které z nich jsou využívány téměř ve všech případech. A právě o nich a jejich podrobnějším popisu bude pojednávat následující část této sekce.

3.3.1 Konvoluční vrstva

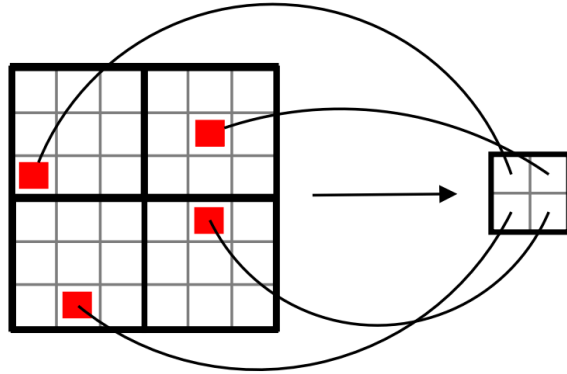
Konvoluční vrstva je určena počtem a velikostí konvolučních jader a jejich hodnotou posunu. Počet konvolučních jader určuje počet výskytů vrstvy, respektive množství detektorů, které posléze vstupují do vrstvy následující, přičemž každý z nich je uzpůsoben k vyhledávání jiných druhů nespojitosti. Volba jejich velikost by měla být závislá na velikosti vstupního obrazu. Čím detailnější je, tím větší by konvoluční jádro mělo být. Hodnota posunu definuje posun jader po vstupním obraze ve směru osy x a y . Důležitou myšlenkou je dodržení překryvu při konvoluci a nalezení maximální aktivační hodnoty reprezentující hledané příznaky (featurey).

3.3.2 Poolingová vrstva

Pooling představuje proces podvzorkování vstupního konvolučního obrazu. Jeho zařazení je výhodné z důvodu rychlejší konvergence při trénování. Zároveň s tím dochází k nalezení podstatně jiných a robustnějších příznaků. V současnosti se využívají dva způsoby poolingů (max a average), jejichž princip je téměř stejný. Jak je viditelné na Obrázku 3.3, je výstup konvoluční vrstvy rozdělen do několika nepřekrývajících se regionů. Ty jsou definované velikostí konvolučního jádra. Následně je v těchto regionech vyhledáváno maximum pro max-pooling, a nebo počítán průměr pro average-pooling. Výstupem je tedy mapa obsahující nalezená maxima, nebo vypočítané průměry. Dochází tak k výrazné redukci dat.

3.3.3 Fully Connected vrstva

Předposledním článkem CNN je vždy Fully Connected vrstva a běžně se využívá i jejich sériové zřetězení. V případě klasifikace není její počet neuronů nijak omezen, pouze s jednou výjimkou a tou je vrstva poslední. U té by jejich počet měl vždy odpovídat množství klasifikačních tříd. Pokud se tedy nejedná o poslední vrstvu CNN není počet neuronů omezen a záleží vždy na konkrétní úloze. Neurony uvnitř ní jsou stejně jako u běžných neuronových sítí propojeny ke všem výstupům z předešlé vrstvy.



Obrázek 3.3: Ilustrace max-poolingu. Obraz je rozdělen do 4 stejných regionů. Nalezená maxima jsou zvýrazněna červenou barvou.

3.3.4 Loss vrstva

Takzvaná Loss vrstva je využívána až jako poslední část CNN a výhradně specifikuje jaký je rozdíl mezi predikovanými a skutečnými labely. K výpočtu hodnoty Loss funkce využívá dle povahy úlohy některý ze známých přístupů:

- Softmax Loss - je využíván pro predikci jedné třídy z množství vzájemně se vylučujících tříd

$$p'_{nj} = \frac{e^{x_{nj}}}{\sum_{j=1}^K e^{x_{nj}}}, \quad j = 1, \dots, K. \quad (3.1)$$

- Sigmoid cross-entropy Loss - je využíván pro predikci tříd formou pravděpodobností

$$E = -\frac{1}{n} \sum_{n=1}^N [p_n \log p'_n + (1 - p_n) \log(1 - p'_n)]. \quad (3.2)$$

- Euclidean Loss - je využíván u regrese ke skutečným labelům

$$E = \frac{1}{2N} \sum_{n=1}^N \|y'_n - y_n\|_2^2. \quad (3.3)$$

3.4 Trénování

V problematice trénování neuronových sítí se setkáváme zejména algoritmem zpětného řetězení (Backpropagation), který v některých případech může být obohacen o pre-training nebo fine-tuning. Pomocí pre-trainingu je možné inicializovat

váhy v určité vrstvě. Fine-tuning se stará o obdobný problém, ale u úloh s malým množstvím trénovacích dat. Jakmile jsou váhy neuronů nastaveny, je možné je optimalizovat pomocí algoritmu zpětného řetězení (Backpropagation).

3.4.1 Pre-training

Pre-training nám umožňuje nastavit váhy určité vrstvy v neuronové síti. Ty můžou být následně relativně rychle optimalizované pomocí trénovacího Backpropagation algoritmu. Literatura popisuje mnoho algoritmů, ale nejčastěji je využívána metoda RBM (Restricted Boltzmann Machines). S její pomocí dojde k trénování první skrytou vrstvy, která se následně využije k získání hodnot pro další skryté vrstvy. Tento proces je následně opakován tolikrát, kolikrát je potřeba k získání potřebných hodnot.

3.4.2 Fine-tuning

Jelikož CNN vyžadují z důvodu snížení možného přetrénování co nejširší dataset potřebný pro trénování, je velkým problémem u mnohých úloh právě množství dat, které je pro trénování možné využít. Jedním z možných řešení je natrénovat síť na podobné úloze, pro kterou máme dostupný větší dataset a následně ji přetrénovat pomocí datasetu konkrétní úlohy s nedostatečným množstvím dat. Tento přístup se nazývá fine-tuning a umožňuje nám aplikovat CNN i na problémy popsateľné pouze malými trénovacími sety.

3.4.3 Backpropagation

Nejčastěji využívaným algoritmem k učení CNN, ale i běžných neuronových sítí je algoritmus Backpropagation. Jelikož se tato práce zabývá výhradně aplikací CNN v úloze reálného světa a její možné implementace, bude jeho princip popsán pouze stručně bez přesné matematické definice.

Princip funkce tohoto algoritmu je možné rozdělit do třech na sebe navazujících částí:

1. Dopředné (feed-forward) šíření vstupního signálu trénovacího vzoru.
2. Zpětné šíření chyby.
3. Aktualizace váhových hodnot na spojeních.

Před začátkem učení je, k nastavení parametrů sítě, možné využít některé z metod pre-trainingu, nebo kompletně náhodného nastavení. Tím je stanovena výchozí hodnota chyby z které algoritmus následně zjistí směr, ve kterém chyba z tohoto "bodu" klesá nejrychleji. Směr reprezentuje vektor, který nazýváme gradient. Postupně proti směru jeho růstu upravujeme váhy neuronové sítě, a tím dochází ke minimalizaci chyby klasifikace. Funkce přepočítávající parametry sítě pomocí hodnoty gradientu se následně opakuje do té doby, dokud dochází ke konvergenci chyby k některému z lokálních minim.

Snaží se tedy najít takové parametry sítě, které minimalizují vybrané ztrátové kritérium, přičemž je možné využít některé již zmíněné v sekci 3.3.4. týkající se Loss vrstvy.

3.5 Hledání optimálních parametrů

Pro nalezení optimálních parametrů sítě, které nám zaručí nalezení lokálního minima ztrátové funkce je možné využít mnoha iterativních metod. V této části bude popsána především metoda Stochastic Gradient Descent (SGD), která nám zaručuje velmi pozitivní vliv při konvergenci k některému minimu, ale i metody využitelné především v úlohách, kde použití SGD nevyhovuje.

3.5.1 Stochastic Gradient Descent

Nejčastěji se, z důvodu nižší výpočetní náročnosti při získávání optimálních parametrů neuronové sítě, využívá metoda SGD. Její matematický předpis je stejný jako u metody Gradient Descent, nicméně nevychází z celého datasetu. Rozděluje ho do přijatelných dávek (anglicky batches), a proto není tak paměťově náročná. Aktualizované hodnoty vah jednotlivých neuronů jsou poté vypočítány jako

$$\omega^{t+1} = \omega^t - \gamma_t \sum_{i=1}^n \nabla E_i(\omega^t), \quad (3.4)$$

kde γ reprezentuje hyperparametr učícího kroku (Learning Rate), který nám při jeho relativně malé hodnotě garantuje pozitivní vliv na minimalizaci hodnotící funkce.

Momentová metoda

Přestože použití SGD při trénování CNN má pozitivní vliv na minimalizaci hodnotící funkce, byl představen i novější přístup využívající tzv. Momentum. Ten dosahuje výrazně lepší konvergence u většiny úloh hlubokých neuronových sítí (DNN³). Na Momentum lze nahlížet jako na vážený průměr současné hodnoty gradientu s kumulovanou hodnotou jeho minulých hodnot. Momentum současného kroku lze vypočítat jako

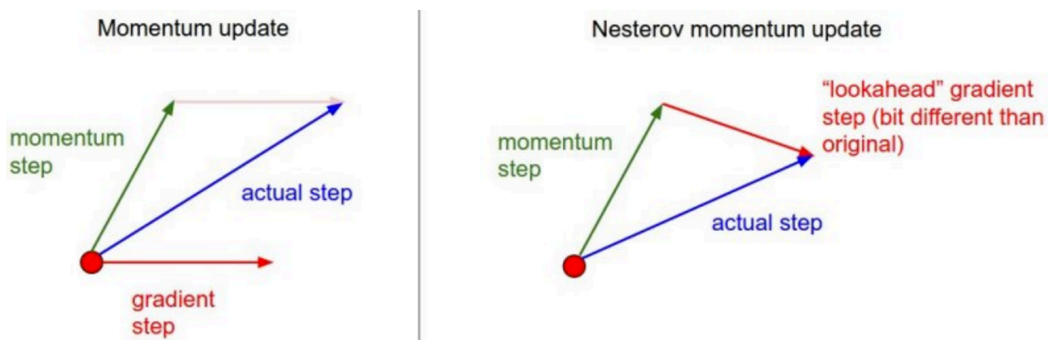
$$\Delta\omega^t = -\gamma_t \nabla E(\omega^t) + \alpha \Delta\omega^{t-1}, \quad (3.5)$$

kde α představuje hyperparametr momenta, který nám udává jak důležitá je hodnota minulých hodnot gradientů (hodnota volena v rozmezí 0.8 - 1.0). Na základě hodnoty Momenta a jeho hyperparametru je možné v průběhu trénování aktualizovat váhy neuronů pomocí Vzorce 3.6.

$$\omega^{t+1} = \omega^t + \Delta\omega^t = \omega^t - \gamma_t \nabla E(\omega^t) + \alpha \Delta\omega^{t-1} \quad (3.6)$$

Nesterova Momentová metoda

Z důvodu velké popularizace neuronových sítí se v poslední době začali objevovat různé modifikace původní Momentové metody, mezi které patří i Nesterovo Momentum. Nesterovo Momentum vychází z bodu, kam původně směřovalo a následně tuto polohu upraví pomocí jeho hodnoty gradientu. Tím je zajištěna teoreticky rychlejší konvergence pro konvexní funkce a v praktickém využití lepší výsledky než u běžného Momenta. Rozdíl je patrný z Obrázku 3.4.



Obrázek 3.4: Grafické znázornění získávání hodnoty Momenta a Nesterova Momenta. Převzato z [13].

³Deep Neural Networks

3.5.2 Adaptivní metody

Předešlé metody byly charakteristické nastavováním hyperparametrů globálně pro celou neuronovou síť. Tento přístup je volen, protože jejich nastavování pro každý neuron je velmi výpočetně náročné. Oproti tomu metodám, které využívají schopnost tyto hodnoty adaptivně měnit, je přislibována velká budoucnost. Jelikož je možné na ně v některých případech narazit, budou stručně popsány právě v této části.

Adagrad

Adagrad je adaptivní učící se metoda navržená Duchim v [14].

$$\sigma_i^{t+1} = \sigma_i^t + \|\nabla E(\omega_i^t)\|_2^2 \quad (3.7)$$

$$\omega_i^{t+1} = \omega_i^t - \frac{\gamma_t \nabla E(\omega_i^t)}{\sqrt{\sigma_i^{t+1} + \epsilon}}, \quad (3.8)$$

kde ϵ musí být velmi malé číslo z důvodu vyhnutí se dělení nulou.

Nevýhodou při použití metody Adagrad je přílišná agresivita v oblasti Deep learningu, z důvodu využití monotónní hodnoty učení (Learning Rate). Zmíněná agresivita je charakterizována hodnotou gradientu, která výrazně konverguje k nule a tím dojde k předčasnému zastavení učení.

RMSProp

Metoda RMSProp je momentálně považována za jednu z nejefektivnějších adaptivních metod, nicméně stále nebyla formálně představena. Jediný zdroj informací pramení z přednášek Geoffa Hintonu [15]. Ten metodu popisuje jako úpravu metody Adagrad tak, aby nedocházelo k agresivnímu poklesu hodnoty učení. Konkrétně se využívá klouzavý průměr umocněného gradientu.

$$\sigma_i^{t+1} = \alpha \sigma_i^t + (1 - \alpha) \|\nabla E(\omega_i^t)\|_2^2 \quad (3.9)$$

$$\omega_i^{t+1} = \omega_i^t - \frac{\gamma_t \nabla E(\omega_i^t)}{\sqrt{\sigma_i^{t+1} + \epsilon}} \quad (3.10)$$

Adam

Adam [16] je nedávno představená metoda, která upravuje metodu RMSProp o přidání momenta. Z následujících vzorců je viditelné, že aktualizace hodnot vychází z "hladší" verze gradientu.

$$m^{t+1} = \beta_1 m^t + (1 - \beta_1) \nabla E(\omega_i^t) \quad (3.11)$$

$$v^{t+1} = \beta_2 v^t + (1 - \beta_2) \|\nabla E(\omega_i^t)\|_2^2 \quad (3.12)$$

$$\omega_i^{t+1} = \omega_i^t - \frac{\gamma_t m^{t+1}}{\sqrt{v^{t+1}} + \epsilon} \quad (3.13)$$

3.6 Best practice

Jak již bylo zmíněno v úvodu této kapitoly, velký zlom zažily CNN v roce 2010, kdy došlo k publikaci [10] v závislosti na výsledcích získaných v rámci soutěže ImageNet. Techniky, které byly při návrhu oceněné sítě využity se následně staly neodmyslitelnou součástí všech dalších CNN. Nejdůležitější objevy a techniky budou popsány právě v této sekci.

3.6.1 Aktivační funkce

V minulosti se výstupní hodnota neuronu u běžných neuronových sítí modelovala pomocí Sigmoidální, nebo lineární funkce. Respektive se výstup neuronu f modeloval jako funkce jeho vstupu x například pomocí

$$f(x) = \tanh(x) \quad \text{a} \quad f(x) = \frac{1}{1 + e^{-x}}. \quad (3.14)$$

Bohužel využití této nelineární saturační funkce má při využití v CNN velký vliv na dobu trénování, a tím nám výrazně omezuje jejich využití. Z tohoto důvodu se v [10] zabývali možnou alternativou. Tou je modelování neuronu pomocí nelineární nesaturovatelné funkce

$$f(x) = \max(0, x). \quad (3.15)$$

Při jejím použití vzniknou tzv. Rectified Linear Units (ReLU). Ty mají lineární odezvu na kladné hodnoty vstupů neuronu a nulovou hodnotu pro ty záporné. Tím je dosaženo výrazného urychlení v problematice trénování. Dle provedeného experimentu se čas potřebný pro získání modelu čtyřvrstvé neuronové sítě pracující s přibližně 25% chybou zkrátit až 6krát. Na základě tohoto zjištění může být konstatováno, že při použití ReLU je trénování CNN výrazně rychlejší.

3.6.2 Overlapping Pooling

V části 3.3.2 byl krátce představen běžně využívaný přístup k pooling. Ten vychází z rozdělení výstupu z konvoluční vrstvy do několika nepřekrývajících se částí. Z těch se následně dle druhu poolingové vrstvy získá výstupní hodnota. Jelikož většina předchozích prací vycházela zejména z tohoto postupu, nabízelo se několik jeho možných modifikací. Mezi ty patří i Overlapping pooling, u něž dochází k překrývání jednotlivých regionů. Při jeho použití docházelo při klasifikaci IMAGENet dat ke zlepšení klasifikace okolo 0.5 % a zároveň s tím i větší odolnosti vůči přetrénování.

3.6.3 Local Response Normalization

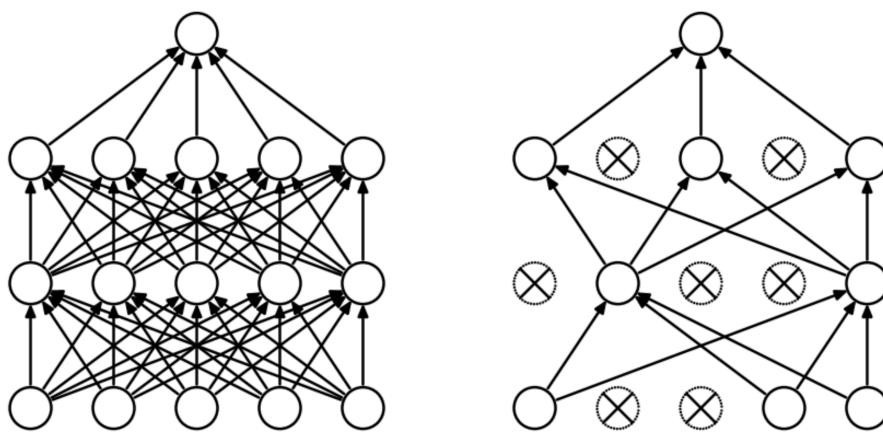
Jelikož se mezi zajímavé vlastnosti ReLU jednotek řadí i nesaturovatelnost, není nutné jejich vstupy na dozdíl od běžných neuronových sítí normalizovat. Jejich učení probíhá pouze v tom případě, pokud je součet jeho vstupních vah kladný. Přesto při experimentování s datasetem IMAGENet bylo použití normalizace využito a tím zároveň docíleno snížení chybovosti sítě o 1.4 %. Toto zlepšení lze ale přikládat i jisté invarianci sítě vůči změnám osvětlení, která je získána právě využitím normalizace.

3.6.4 Data Augmentation

Nejjednodušší a nejběžněji používaná metoda redukce přetrénování je tzv. augmentace dat, respektive vytváření umělých dat, pomocí kterých je možné zvětšit trénovací množinu. Takto vzniklá data mohou nabývat mnoha rozličných forem a do jisté míry závisejí na konkrétní úloze. Nejčastěji se však můžeme setkat s použitím některých geometrických transformací, které nám umožňují obraz různě pootáčet, přetáčet (horizontálně a vertikálně), měnit mu barvy apod.

3.6.5 Dropout

DropOut je regularizační technika sloužící převážně k redukcí možného přetrénování. Přichází se zcela novým a velice jednoduchým přístupem, který je dle výsledků prezentovaných v [17] extrémně efektivní. Během trénování dochází k náhodnému výběru určité části neuronů. Ty jsou následně z vrstvy odstraněny. Doporučený počet odstraněných neuronů je dle pozorování stanoven na 50 - 80 %. Nutno ale podotknout, že má i výrazný podíl na kvalitě nalezených příznaků.



Obrázek 3.5: Levý obrázek reprezentuje běžnou neuronovou síť se dvěma skrytými vrstvami. Na pravém obrázku je zobrazena stejná architektura s využitím Dropoutu. Převzato z [17].

Kapitola 4

Segmentace

Segmentace obrazu slouží především k rozdělení vstupních dat do částí, neboli segmentů, které mají určitou spojitost s prvky reálného světa a zároveň jako celek tvoří segmentovaný obraz. Hlavním cílem segmentace v mnoha úlohách je odlišit od sebe podstatnou a nepodstatnou část informace obsaženou v obraze. Z toho plyne, že pro správnou funkci segmentace je zapotřebí značné množství informací o vstupních datech, bez nichž je samotná úloha segmentace výrazně složitější a často vede k takzvané částečné segmentaci. Z výše uvedeného je zřejmé, že ne vždy dojde ke kompletní segmentaci, a proto jsou zavedeny termíny částečné a kompletní segmentace.

Kompletní segmentace

- Jednoznačná korespondence vysegmentovaných objektů s objekty v segmentovaném obraze.
- Dobré výsledky bez vyšších znalostí pro konstantní objekty na konstantním jasu pozadí.
- Při složitějších úlohách je zapotřebí znalostí o řešeném problému.
- Příklad úlohy - segmentace krevních buněk a textu.

Částečná segmentace

- Vysegmentované oblasti jsou homogenní vzhledem k určitým zvoleným vlastnostem - jas, barva, textura apod.
- Obecně se vysegmentované oblasti mohou překrývat.
- Pro získání použitelných výsledků je nutné využít vyšší úroveň zpracování, popřípadě expertních znalostí.
- Příklad úlohy - segmentace scénérie reprezentující pohledu z okna – výsledek neodpovídá jednomu objektu.

4.1 Metody segmentace

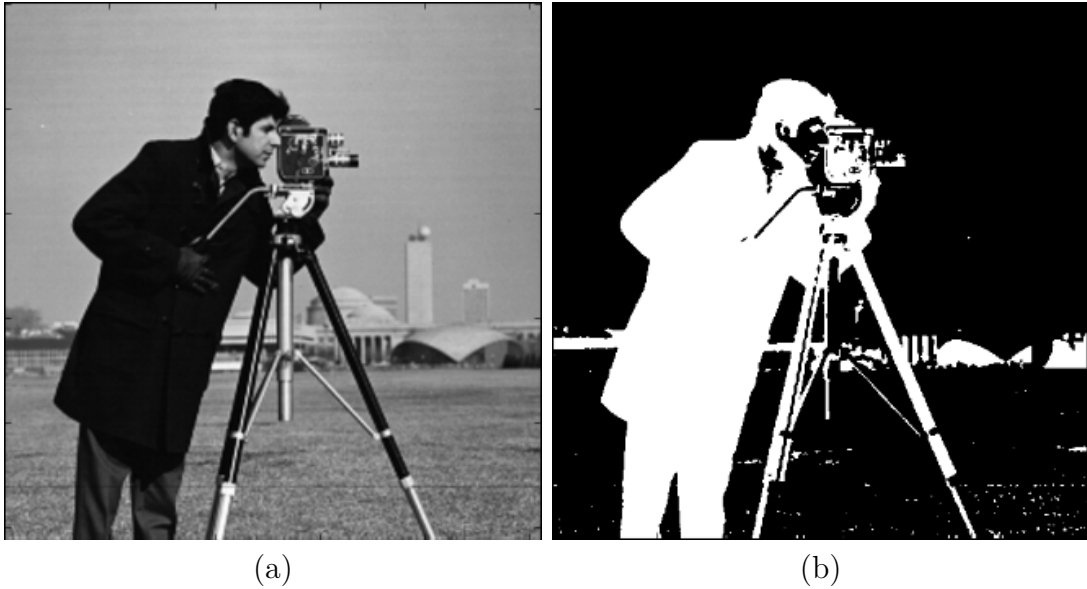
4.1.1 Prahování

Prahování je jednou z nejstarších metod používaných pro segmentaci obrazu. Nicméně i přes její stáří se stále řadí mezi metody nejpoužívanější, a to zejména díky její jednoduchosti, snadné implementaci a nízké hardwarové náročnosti, která umožňuje její využití v reálném čase. Na druhou stranu jí lze využít pouze v určitých typech úloh, kde je evidentní rozdíl v úrovních jasů mezi pozadím a objekty. Prahování je tedy založené na odrazivosti a absorpci světla povrchů a s tím spojenou úroveň jasů jednotlivých pixelů. Metoda prahování spočívá především ve volbě konkrétní hodnoty práhu. Ten můžeme zvolit následujícími způsoby:

- Interaktivně na základě znalosti histogramu.
- Automaticky na základě např. Otsuovy metody blíže popsané v [18].

Po zjištění hodnoty práhu jsme již schopni dle Vzorce 4.1 vytvořit binární obraz, ze kterého je patrné, jaké části odpovídají pozadí, respektive objektu.

$$g(i, j) = \begin{cases} 1, & \text{pro } f(i, j) \geq T \\ 0, & \text{pro } f(i, j) < T \end{cases} \quad (4.1)$$



Obrázek 4.1: Výsledek prahování obrázku (a), prezentovaný jako binární obraz (b), kde černá barva reprezentuje pozadí a bílá popředí.

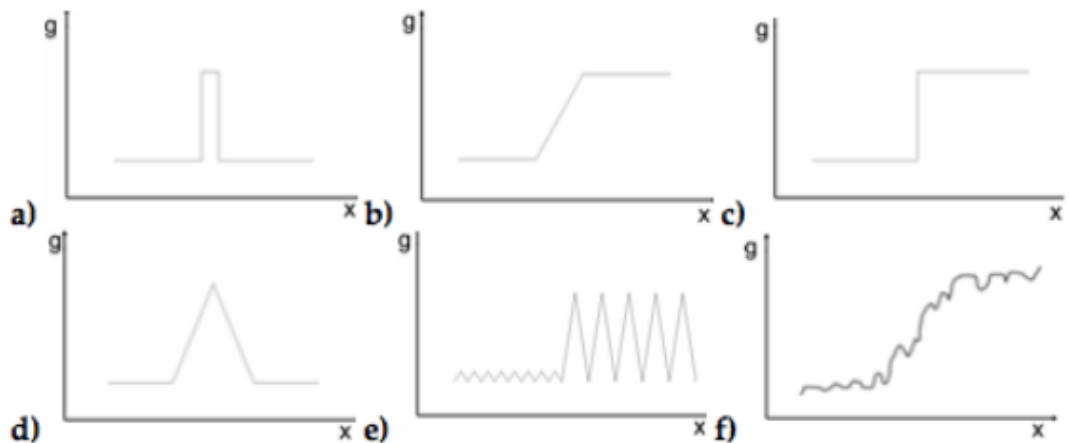
4.1.2 Segmentace na základě detekce hran

Před samotným vysvětlením funkce segmentace vycházející z detekce hran je nezbytné si pojmem hrana ujasnit a následně charakterizovat v kontextu zpracování obrazu.

Hrana

Hrana je oblast obrazu, v níž dochází k určité nespojitosti v jasu, popřípadě v barvě a textuře. Z některých nespojitostí zobrazených na Obrázku 4.2 je zřejmé, že hrana může nabývat různých podob, a proto může být v určitých případech složité lokalizovat všechny důležité hrany.

Úlohu segmentace na základě detekce hran lze rozdělit do dvou částí. Získávání obrazu hran a jejich následného spojování, při kterém dochází k vytváření jednotlivých segmentů.



Obrázek 4.2: Ukázka různých tvarů hran, kde g reprezentuje úroveň jasu a x jednotlivé pixely v řadě.

Detekce hran

Proces detekce hran se dá rozdělit na tři fáze: filtrování, ekvalizace a detekce. V první fázi, tedy filtraci, můžeme pomocí vhodného filtru odstranit šum vzniklý při vzorkování obrazu, kvantování, rozmazání či nevhodném nastavení kamery. Následně je možné zvýraznit ty oblasti, kde je významná změna intenzity jasu. K tomu slouží například metoda CLAHE [19]. Nakonec jsou detekovány a lokalizovány body, kde je změna intenzity nejvýznamnější. Tuto detekci nám umožňují dnes běžně používané metody [20], [21].

Segmentace z obrazu hran

Při segmentaci z obrazu hran se vychází z obrazu hran získaným některým hranovým operátorem. Pro úspěšnost segmentace je zapotřebí splnění následujících předpokladů:

- Není opomenuta žádná významná hrana.
- Není detekována oblast, která hranou není.
- Rozdíl mezi polohou skutečné a získanou hrany by měl být co nejmenší.

Je zjevné, že v některých případech je téměř nemožné splnit výše uvedené předpoklady, nicméně je možné chybovost segmentace minimalizovat apriorní informací o segmentovaném obraze.

Pro segmentaci z obrazu hran se využívají například metody:

- sledování hranice,
- aktivní kontury,
- hougova transformace,
- určování hranice na základě znalosti o její poloze.

4.1.3 Hledání oblastí

Pokud je obraz postihnutý šumem a není v něm možné s potřebnou přesností vyhledávat hrany, je možné pro segmentaci obrazu využít jednu z metod prohledávání oblastí, které jsou odolnější proti šumu a zároveň vyhledávají segmenty přímo s využitím kritéria homogenity, které je založeno na vlastnostech jako úroveň jasů, textura, barva apod.

Metoda spojování oblastí

Metoda spojování oblastí vychází z jednoduchého principu, kdy na jednotlivé pixely (popřípadě oblasti 2×2 a větší) je pohlíženo jako na oblasti samostatné. Následně u těch oblastí, které vyhovují kritériu homogenity, dochází k jejich spojení. Pokud již není možné spojit žádné dvě oblasti, dojde k ukončení spojovacího procesu.

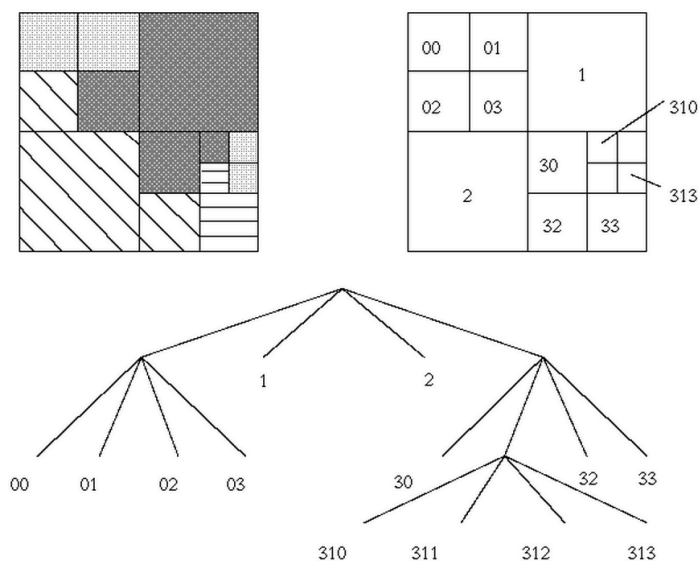
Metoda dělení oblastí

V metodě dělení oblastí se vychází ze stejného principu jako u předchozí popísané metody s tím rozdílem, že na začátku se na obrázek pohlíží jako na jeden segment, který je postupně dělen do té doby, dokud daná oblast není homogenní.

Metoda spojování a dělení oblastí

Tato metoda využívá vlastností obou již popsaných metod. Před vlastní segmentací je potřeba definovat strukturu, kterou budeme používat pro správu oblastí, kdy se nejčastěji vychází z rozkladu do takzvaných čtvrtinových stromů. Jedná se o stromovou strukturu, v níž se oblast při přechodu do další úrovně dělí na čtyři čtvrtinové oblasti (viz. Obrázek 4.3).

Další požadovanou vlastností, kterou musíme definovat, je již dříve zmíněné kritérium homogenity, které hraje klíčovou roli v chování celého algoritmu a mělo by vycházet ze znalosti úlohy a vlastností segmentovaného obrazu. Kritéria pro dělení a štěpení nemusí být nutně stejné, ovšem musíme mít na paměti, že by mohlo dojít k zacyklení výpočtu. Při realizaci je potřeba postupovat tak, aby bylo zacyklení zabráněno.



Obrázek 4.3: Příklad použití quad-tree struktury.

4.1.4 Srovnávání se vzorem

Srovnávání se vzorem je jedna ze základních segmentačních metod, která se využívá k zjištění výskytu a lokace konkrétní části obrazu. Vzor je reprezentován formou masky. Ta je postupně přikládá na všechna místa v porovnávaném obraze, kde je dle jednoho z následujících kritérií vypočítána míra souhlasu, respektive vzájemná korelace. Ta může být matematicky definována např. jako:

$$C_1(u, v) = \frac{1}{1 + \sum_{(i,j) \in V} |f(i+u, j+v) - h(i, j)|}, \quad (4.2)$$

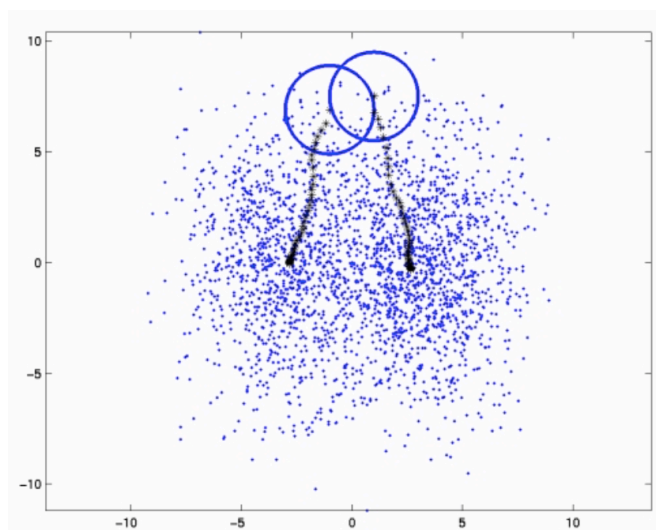
$$C_2(u, v) = \frac{1}{1 + \sum_{(i,j) \in V} (f(i+u, j+v) - h(i, j))^2}, \quad (4.3)$$

kde testujeme souhlas obrazu f se vzorem h umístěným v poloze (u, v) pro každou polohu vzoru v obraze.

Nutno podotknout, že pokud bude hledaný objekt výrazně potočený, s jinou velikostí, nebo geometrickým zkreslením, je potřeba vypočítat vzájemnou korelaci pro všechny pootočení, velikosti apod. Z toho důvodu je rychlost výpočtu výrazně zpomalena. Pro urychlení je možné využít metodu, která nejprve obraz zkoumá s podstatně hrubším rozlišením a v místě získaných lokálních maxim pak přesněji doměří polohy, pro které je vypočítána největší hodnota vzájemné korelace.

4.1.5 Mean-Shift

Mean-Shift je obecný algoritmus pro iterativní nalezení lokálního maxima hustoty vzorků. Jméno vystihuje princip, a proto se v každém kroku posuneme ve směru váženého průměru vzorků (těžiště) tak, jak je zobrazeno na Obrázku 4.4



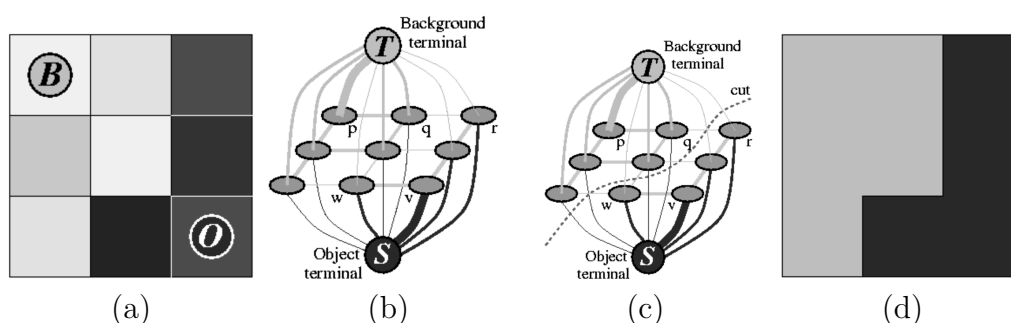
Obrázek 4.4: Příklad konvergence k lokálnímu maximu při použití metody Mean-Shift.

Při použití pro shlukování postupujeme ve dvou krocích:

1. Z každého vzorku spustíme Mean-Shift a zapamatujeme si lokální maximum, do něhož dokonvergoval.
2. Shlukujeme ty vzorky, které dokonvergovaly do stejného maxima (s určitou tolerancí). Oblast, z které vzorky konvergují do stejného bodu, říkáme basin of attraction.

4.1.6 GrabCut

Metoda GrabCut vychází z teorie grafů. Její princip spočívá v převedení obrazových dat na orientovaný graf, ve kterém se dle ohodnocení jednotlivých hran vypočítává minimální cena jeho řezu. Na základě tohoto řezu je poté možné stanovit, jaká část odpovídá segmentovanému objektu. Celý princip je patrný z následujícího obrázku.



Obrázek 4.5: Příklad funkce metody GrabCut, kde (a) reprezentuje označení pozadí a popředí na obrazových dat a (d) výsledný binární obraz, kde černá barva reprezentuje pozadí a bílá popředí. Převzato z [22].

Největší výhodou metody GrabCut při segmentaci je možné interaktivní řešení segmentace, kdy je uživatelem/expertem stanoveno, které části reprezentují pozadí a které odpovídají segmentovanému objektu.

Popis algoritmu

1. Označení seedů - stanovení popředí a pozadí v obraze.
2. Převedení obrazu na orientovaný graf.
3. Ohodnocení hran grafu.
4. Výpočet minimálního řezu grafem.
5. Provedení řezu a tím oddělení objektu od pozadí.

Kapitola 5

Návrh řešení a implementace

V této kapitole bude pojednáno o praktické stránce problému rozpoznávání plodnic vysokých hub. Rozdělení do dílčích sekcí plně koresponduje s jednotlivými již zmíněnými teoretickými postupy. Dojde k popsání experimentů zaměřených na optimalizaci dat obsažených v datasetu a nalezených úskalí. Následující část se zabývá konkrétní formou předzpracování trénovacích dat a její případnou nutností v rámci této úlohy. Dále budou popsány klasifikační experimenty založené na konvolučních neuronových sítích, přičemž dojde i k porovnání s dříve velmi používanou metodou SVM. Na závěr bude popsána integrace vytvořené neuronové sítě do některé možné formy použitelné v reálném světě.

Před popisem provedených experimentů je nutné zmínit, že všechny hodnoty Accuracy (přesnosti) a Loss (hodnotící funkce), uvedené ve zbytku práce, reprezentují střední hodnotu dat získaných z posledních 10 % trénování, respektive testování u 5 realizací cross-validace. Dále je nutné uvést, že většina experimentů popsanych v této části byla provedena na černobílých snímcích, vyjma toho, kde dochází k porovnávání datasetů s černobílými a barevnými daty. Současně s tím byly všechny experimenty, na základě doporučení vedoucího práce, provedeny na konvoluční neuronové síti s neměnnou architekturou. Vytvoření této sítě bylo nultým experimentem provedeným mimo rozsah této práce. Při něm byly metodou pokus-omyl, založené na některých již používaných sítích, nalezeny takové parametry sítě, které umožnili natrénovat síť s přesností klasifikace větší než 50 %.

5.1 Dataset

Jak již bylo v úvodu práce zmíněno, v úlohách rozpoznávání obrazu se setkáváme výhradně s metodami strojového učení za pomoci jisté apriorní informace, respektive označení jaká třída se na obrazových datech vyskytuje. Jedná se tedy o přístup využívající takzvané učení s učitelem, jenž k natrénování potřebují obsáhlou množinu zahrnující označená data (dataset). Je tedy zřejmé, že k vytvoření systému, jehož úkolem je rozhodnout na základě obrazových dat ohledně druhu plodnice, bude takovýto dataset zapotřebí. Protože žádný takový volně šiřitelný set neexistuje, bylo nutné ho svépomocí vytvořit.



Obrázek 5.1: Ilustrace ideálního formátu trénovacích a testovacích dat pro různé druhy hub.

5.1.1 Vzorová data

Jelikož je důležité, aby analyzovaný snímek obsahoval co největší množství informací potřebných pro rozpoznávání, bylo stanoveno, jak by měl každý jeden snímek v ideálním případě vypadat. Při tom byly zúročeny již získané teoretické informace prezentované v sekci 1.2.1, která pojednává o určovacích znacích plodnic vysokých hub. Snímek by tedy měl splňovat následující znaky:

- Zobrazení pouze jedné plodnice (pokud se nejedná o skupinový druh).
- Viditelnost výtrusného rouška.
- Určitá kvalita fotografie - ostrost, světlost, barevnost apod.

- Viditelnost celé plodnice.

Pro lepší ilustraci jsou na Obrázku 5.1 zobrazena ideální data splňující všechny výše zmíněné znaky.

5.1.2 Získání dat

Tvorba datasetu je z důvodů jeho potřebné velikosti časově velmi náročná úloha. Při jeho tvorbě, v kontextu zadané práce, byla velkým přínosem spolupráce s panem Rolandem Baranovičem, který spravuje web www.nahuby.sk, s jehož souhlasem bylo možné využít již částečně označenou databázi obrázků. Ta obsahovala něco kolem 300 tříd a přibližně 2000 obrázků. Z důvodu malé četnosti dat u většiny tříd a v některých případech i zcela nepoužitelných obrázků byla vybrána množina 10 tříd s největší četností. Dalším atributem pro výběr byla i snaha o dodržení co nejširšího spektra druhů a znaků. Dále byl kladen důraz na zajištění podobnosti mezi některými druhy, a tím zvýšení obtížnosti rozpoznávání. Vybrané druhy, četnost jejich snímků a charakteristické znaky jsou viditelné v Tabulce 5.1.

Druh plodnice	Četnost	Charakteristické znaky
Amanita Caesara	120	Lupeny, Jedlá
Amanita Muscaria	142	Lupeny, Nejedlá
Amanita Rubescens	129	Lupeny, Jedlá
Boletus Appendiculatus	105	Rourky, Jedlá
Boletus Calopus	102	Rourky, Nejedlá
Cordyceps Militaris	117	Speciální, Jedlá
Leccinum Scabrum	103	Rourky, Jedlá
Mycena Renati	146	Lupeny, Nejedlá
Schizophyllum Commune	82	Speciální, Nejedlá
Verpa Bohemica	183	Speciální, Jedlá

Tabulka 5.1: Četnost jednotlivých druhů (tříd) obsažených v základním datasetu.

5.2 Předzpracování dat

Předzpracování dat je důležitá část klasifikační úlohy k níž dochází před samotným trénováním. Ačkoliv by se mohlo zdát, že předzpracování nemusí mít

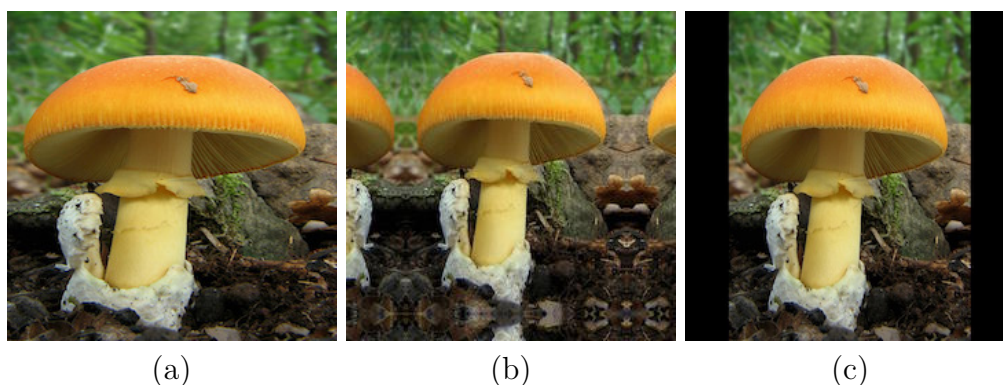
znatelný vliv na výsledky natrénovaného klasifikátoru, je tomu ve většině případů právě naopak. Z tohoto důvodu byl vliv některých druhů předzpracování otestován formou experimentů.

5.2.1 Transformace dat

Jelikož je při trénování CNN nutné využívat stejně velkých čtvercových snímků obsažených v trénovací i testovací sadě, bylo potřeba zajistit jejich transformaci tak, aby této podmínce vyhovovala.

V získané trénovací množině bylo obsaženo přibližně stejné množství snímků pořízených s horizontálním a vertikálním natočením snímacího zařízení, přičemž plodnice nebyla vždy přesně v prostředku snímku. Úlohou tedy bylo vybrat takovou transformaci, popřípadě metodu, která by jednotlivé snímky převedla do čtvercového formátu tak, aby nedošlo k odstranění plodnice ze snímku a zároveň došlo k co nejmenšímu ovlivnění kvality klasifikace u CNN. Konkrétně byly otestovány 3 druhy transformací potřebné k dosažení požadovaného čtvercového tvaru snímku pomocí:

- roztažení,
- zrcadlení,
- přidání černých okrajů.



Obrázek 5.2: Různé druhy transformace obrazu do čtvercové podoby. Obrázek (a) reprezentuje roztažení, obrázek (b) zrcadlení a v poslední řadě (c) reprezentuje nastavení snímku o černé okraje.

Dle získaných výsledků viditelných v Tabulce 5.2 je patrné, že nejlepší odezvu měla 3. možnost, jenž přidávala černé okraje do obdélníkových snímků. Další zajímavostí zjistitelnou ze získaných hodnot byl velký vliv zrcadlení na výsledky trénování. To lze přičinit vzniku nových a zřejmě více "zajímavých" příznaků právě v oblasti, kde docházelo k zrcadlení. Na vyhledávání těchto příznaků byla CNN následně natrénována. Jelikož však měli pouze minimální informativní hodnotu, bylo při predikci velice složité zjistit k jaké třídě analyzovaná data patří.

Formát dat	Test Accuracy	Test Loss
Nesegmentovaná - černé okraje	66.1%	1.645
Nesegmentovaná - zrcadlení okrajů	33.9%	3.74
Roztažení snímku	53.7%	2.31

Tabulka 5.2: Získané hodnoty kvality klasifikace jednotlivých metod transformace obrazu.

5.2.2 Normalizace

Jednou z nejčastějších forem předzpracování v kontextu klasifikace je normalizace. Ta reprezentuje změnu hodnot jednotlivých pixelů obrazu z rozsahu 0-255 na hodnoty 0.0 - 1.0. Tato transformace zajišťuje zrychlení při trénování a hlavně zamezuje konvergenci do některého lokálního optima. Zmíněné chování bylo potvrzeno formou experimentu, kde byla CNN natrénována pomocí normalizovaných a nenormalizovaných dat. U těch nenormalizovaných došlo ve všech případech ke konvergenci do lokálního maxima. Byly vyzkoušeny i různé druhy nastavení hyperparametru Learning Rate, avšak bez znatelného rozdílu. Hodnotící funkce vždy dokonvergovala do lokálního minima reprezentujícího přesnost kolem 12 - 15 %. Na druhou stranu při použití dat normalizovaných byly opakovaně získány výsledky přesahující 60 %. Výsledkem tohoto experimentu byla dokázána nutnost normalizace dat v úloze rozpoznávání plodnic vysokých hub.

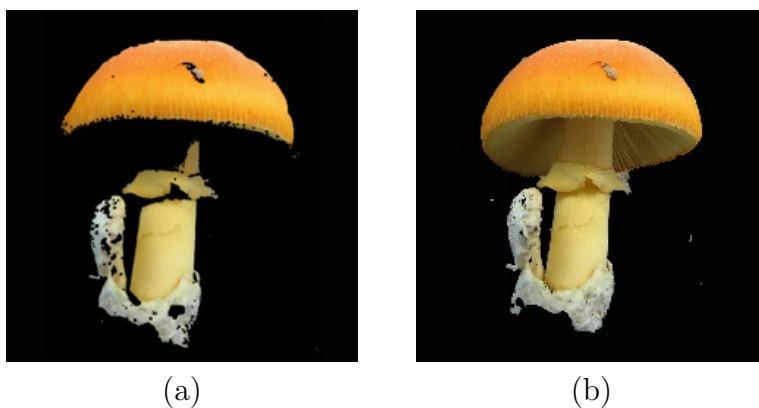
5.3 Segmentace

Na základě výše získaných hodnot bylo očividné, že zařazení segmentace jakožto součásti předzpracování by mohlo mít pozitivní vliv na trénování neuronové sítě. Právě touto tezí se další experiment zabývá.

Nejprve bude nezbytné, na základě teoretických informací, vybrat nejlépe vyhovující metodu. Následně je potřeba ověřit její segmentační schopnost pomocí porovnání s jednou konvenční metodou. Na závěr budou segmentací upravená data použita ke klasifikačnímu experimentu, který by měl potvrdit, nebo vyvrátit pozitivní vliv na řešenou úlohu.

5.3.1 Výběr metody

Důležitou součástí segmentace je výběr metody. Těch v současnosti existuje velké množství a jejich stručnému výčtu a popisu funkce je věnována Kapitola Segmentace. Při výběru konkrétní metody je v kontextu této diplomové práce důležité přihlížet zejména na co největší robustnost při hromadném zpracování dat pro neuronovou síť, ale také k již získaným teoretickým informacím o plodnicích hub. Na základě velmi vysoké různorodosti jednotlivých plodnic stejného druhu je možné rovnou vyloučit metodu srovnávání se vzorem a prahování. Na základě zmíněné rozdílnosti jednotlivých plodnic je možné vyřadit i metody hledání oblastí, u kterých by mohlo dojít k nevyhovění kritériu homogenity, a tím chybné segmentaci. Další popsaná metoda vycházela z detekce hran. Tu je nutné vyřadit na základě možné chybovosti při získávání hran, způsobené například cizím tělesem v obraze (stébla trávy, plž lezoucí po plodnici apod.).



Obrázek 5.3: Výsledky segmentace při použití metody GrabCut (a) a prahování (b).

Jelikož je pro správnou segmentaci potřeba i jistá apriorní informace o tvaru a postavení plodnice ve snímku, bylo rozhodnuto ve prospěch metody GrabCut,

která právě z těchto hodnot vychází. Další její výhodou je možné využití ve finální aplikaci. V té je důležité pracovat s formátem analyzovaného snímku, který se co nejvíce přibližuje formátu trénovacímu. Uživatel tak díky iterativnímu označování popředí a pozadí získá optimální segment plodnice, který bude výrazně snadněji klasifikovatelný.

5.3.2 Testování metody

V této části budou představeny výsledky dvou testů vybrané segmentační metody GrabCut. První se zaměřuje na rozdíl v její přesnost s porovnáním stále často využívané metody prahování. Z Obrázku 5.3 je patrné, že při použití GrabCutu získáme téměř optimální segment plodnice. Na druhou stranu i při použití prahování bylo dosaženo relativně přijatelného výsledku, nicméně je potřeba zmínit, že segment zcela postrádá výtrusné rouško a zároveň k jeho získání takto kvalitního segmentu bylo potřeba i velkého množství postprocesingu.

Další experiment spočíval v hodnocení klasifikace při použití datasetu obsahujícího vysegmentované plodnice vycentrované na černém pozadí. To bylo vybráno pro jeho neutralitu, nicméně obdobný výsledek je možné získat i při použití jiné barvy. Je pouze nutné brát ohled na zajištění optického oddělení popředí a pozadí snímku. Získané výsledky experimentu byly pro lepší porovnání jednotlivých metod preprocesingu dosazeny do Tabulky 5.3 spolu s již dříve získanými hodnotami.

Výsledkem tohoto experimentu je zjištění, že pro klasifikaci vysokých hub má segmentace dat velice pozitivní vliv na výsledky. Při porovnání s předchozí nejlepší metodou předzpracování došlo k výraznému zlepšení, a to o více jak 13 %.

Formát dat	Test Accuracy	Test Loss
Nesegmentovaná - černé okraje	66.1%	1.645
Nesegmentovaná - zrcadlení okrajů	33.9%	3.74
Roztažení snímku	53.7%	2.31
Segmentovaná	79.3%	0.955

Tabulka 5.3: Získané hodnoty při klasifikaci rozdílných vstupních dat pomocí CNN. Experiment podporující využití segmentace.

5.4 Augmentace

Na základě nedostatku trénovacích dat, a tím způsobeným přetrénování sítě, bylo potřeba navýšit velikost datasetu pomocí augmentace. Augmentovaná data byla vytvořena pomocí náhodného otočení v rámci malého úhlu ($0-15^\circ$) a drobného posunutí ve směru obou os (viz Obrázek 5.4). Došlo tedy k vytvoření 3 syntetických snímků ke každému reálnému. Konkrétní hodnoty, o které byl dataset zvětšen, je možné vyčíst z Tabulky 5.4.

Druh plodnice	Četnost	Četnost aug. dat	Celkem
Amanita Caesara	120	360	480
Amanita Muscaria	142	426	568
Amanita Rubescens	129	387	516
Boletus Appendiculatus	105	315	420
Boletus Calopus	102	306	408
Cordyceps Militaris	117	351	468
Leccinum Scabrum	103	309	412
Mycena Renati	146	438	584
Schizophyllum Commune	82	246	328
Verpa Bohemica	183	549	732

Tabulka 5.4: Četnost jednotlivých druhů hub obsažených v datasetu před a po augmentaci.



Obrázek 5.4: Zobrazení augmentovaných dat vytvořených pomocí rotace a posunutí reálného segmentu plodnice.

Velikost datasetu tak byla navýšena na 4 násobek, avšak bylo potřeba prověřit i vliv na snížení přetrénování sítě. Z tohoto důvodu byla síť natrénovaná těmito

daty podrobena klasifikačnímu experimentu. Jeho výsledky jsou zobrazeny v Tabulce 5.5 a prezentují nám tak navýšení přesnosti o více jak 5% a snížení hodnoty Lossu na polovinu.

Formát dat	Test Accuracy	Test Loss
Nesegmentovaná - černé okraje	66.1%	1.645
Nesegmentovaná - zrcadlení okrajů	33.9%	3.74
Roztažení snímku	53.7%	2.31
Segmentovaná	79.3%	0.955
Segmentovaná + augmentovaná	84.5%	0.567

Tabulka 5.5: Získané hodnoty při klasifikaci rozdílných vstupních dat pomocí CNN. Experiment podporující využití augmentace.

5.5 Velikost dat

Do této doby byly díky preprocessingu získány relativně přijatelné hodnoty klasifikace, a proto je doporučováno se stále věnovat úpravám vstupních dat na rozdíl od nastavování parametrů neuronové sítě. Jelikož ve všech předchozích experimentech bylo používáno čtvercových snímků o velikosti hrany 100px, bude právě v této sekci vliv velikosti vstupních dat na hodnotící funkci experimentálně ověřen.

25px		50px		75px		100px		150px	
Acc.	Loss	Acc.	Loss	Acc.	Loss	Acc.	Loss	Acc.	Loss
86.3%	0.548	88.1%	0.408	90.0%	0.341	86.9%	0.455	76.1%	1.138
83.4%	0.605	87.1%	0.420	89.1%	0.365	82.6%	0.635	77.9%	0.980
82.3%	0.595	87.2%	0.423	87.4%	0.458	86.1%	0.493	76.9%	1.091
82.6%	0.729	89.2%	0.406	87.9%	0.426	85.0%	0.538	78.0%	0.962
87.0%	0.481	89.8%	0.339	90.9%	0.313	87.5%	0.424	76.0%	1.145
85.0%	0.514	92.0%	0.278	89.0%	0.335	86.0%	0.545	78.3%	0.947
85.5%	0.586	90.6%	0.320	87.5%	0.434	75.9%	1.421	75.2%	1.345
85.6%	0.555	89.4%	0.354	85.1%	0.531	84.0%	0.654	77.8%	0.940
85.8%	0.525	91.0%	0.303	90.0%	0.349	84.2%	0.628	75.6%	1.245
84.8%	0.541	92.1%	0.299	84.5%	0.567	85.1%	0.519	78.0%	0.970
84.8%	0.569	89.6%	0.365	88.1%	0.412	84.3%	0.631	77.0%	1.076

Tabulka 5.6: Tabulka reprezentující cross-validaci získané hodnoty přesnosti a hodnotící funkce. První řádek reprezentuje velikost hrany vstupního obrázku. Poslední řádek reprezentuje průměr získaný z 10 iterací cross-validace.

Experiment byl prováděn na datasetu s augmentovanými a segmentovanými daty. Došlo k porovnání natrénování CNN na čtvercových obrázcích s velikostí hrany 25, 50, 75, 100 a 150px. Takto získané hodnoty jsou prezentované v Tabulce 5.6, kde poslední řádek je průměr hodnot získaných pomocí 10 iterací cross-validace. Je viditelné, že nejlepší odezvy bylo dosaženo při použití snímků o velikosti 50x50px. Zároveň s tím bylo získáno zlepšení v přesnosti o přibližně 5 % a snížení hodnotící funkce o téměř polovinu.

5.6 RGB vs Gray

Jak bylo popsáno v úvodu této kapitoly, všechny výše popsané experimenty vycházeli z černobílých dat. K tomuto postupu bylo přistoupeno záměrně z důvodu snadnější konvergence při trénování. Jelikož již byly získány přijatelné hodnoty (s výraznou pomocí preprocesingu), bylo možné provést finální experiment. Ten se zaměřuje na porovnání hodnot získaných pro RGB obraz s těmi již známými. Po provedení 5 iterací cross-validace a zprůměrování získaných hodnot Accuracy a Lossu byla tato hodnota přidána spolu s již známými hodnotami do Tabulky 5.7. Z té je možné vyčíst další významný pokrok, ke kterému při použití RGB snímků došlo. Konkrétně se jedná o nárůst přesnosti o dalších 10 % na hodnotu přesahující 95 % a opětovné snížení Lossu na polovinu.

Formát dat	Test Accuracy	Test Loss
Nesegmentovaná - černé okraje	66.1%	1.645
Nesegmentovaná - zrcadlení okrajů	33.9%	3.74
Roztažení snímku	53.7%	2.31
Segmentovaná	79.3%	0.955
Segmentovaná + augmentovaná (gray)	84.5%	0.567
Segmentovaná + augmentovaná (RGB)	95.1%	0.235

Tabulka 5.7: Získané hodnoty při klasifikaci rozdílných vstupních dat pomocí CNN. Experiment podporující využití RGB snímků.

5.7 CNN vs SVM

Ačkoliv klasifikace obrazových dat za pomoci CNN je považována za state-of-the-art přístup, bylo rozhodnuto, že tato teze bude ověřena klasifikačním experimentem. Konkrétně dojde k porovnání výsledků CNN s různými aplikacemi metody SVM, jenž je stále hojně využívána a pro mnohé úlohy dostačující. Pro trénování SVM klasifikátoru bylo využito několika přístupů a jejich kombinací:

1. Příznakový vektor sestavený z hodnot jasu snímku.
2. Příznakový vektor založený na metodě LBP.
3. Příznakový vektor založený na metodě HOG.
4. Použití lineárního SVM.
5. Použití Gaussovského SVM.

Zároveň s tím došlo k trénování klasifikátoru na datasetu, který obsahoval segmentovaná data. Z Tabulky 5.8 je viditelné, že předpoklady založené na rozdílu mezi nízkoúrovňovými a vysokoúrovňovými příznaky, a popsané v úvodu 2. kapitoly, byly správné. Popis pomocí nízkoúrovňových příznaků nedosahuje ani poloviční přesnosti získané při použití CNN, a proto je možné řešení pomocí lineárního klasifikátoru SVM vyloučit.

Klasifikátor	Test Accuracy
CNN	79.3%
Lineární SVM + vektor	22.1%
Lineární SVM + LBP	37.2%
Lineární SVM + HOG	35.9%
Gaussovské SVM + vektor	20.7%
Gaussovské SVM + LBP	36.1%
Gaussovské SVM + HOG	35.4%

Tabulka 5.8: Získané hodnoty při klasifikaci segmentovaných snímků pomocí klasifikátoru CNN a SVM.

5.8 Klasifikace pomocí CNN

Do této části bylo prostřednictvím výše uvedených experimentů zjištěno, jaký formát by měla mít testovací data. Tato sekce se bude zabývat zejména klasifikací. Konkrétně použitým frameworkem pro práci s CNN, nastavením jejích parametrů a výslednou architekturou.

5.8.1 Caffe framework

Pro práci s konvoluční neuronovou sítí jsem použil veřejně dostupný framework Caffe [23]. Ten definuje neuronovou síť pomocí modelového schématu, který vychází z modelu *cuda-convnet* použitého v [10]. Jeho základem jsou vrstvy a spojení mezi nimi. S informacemi pracuje ve formě blobů, představujících jednotné rozhraní paměti. Bloby jsou reprezentovány vícerozměrným polem (nejčastěji čtyřrozměrným), které obaluje zpracovávaná data a zajišťuje synchronizaci mezi CPU nebo GPU. Toto pole v sobě nese data, jejich deriváty v podobě batchů(dávek) a parametry modelu. Rozměry každého blobu jsou poté $kernelNumber \times imageDimension \times imageWidth \times imageHeight$.

5.8.2 Výběr parametrů sítě

V současné době popisuje literatura [3], [15], [13], [23] mnoho různých parametrů, které u konvoluční neuronové sítě mohou být optimalizovány. Mezi ně patří především:

1. počet konvolučních a poolingových vrstev,
2. architektura konvoluční části - počet jader, jejich velikost a posun,
3. architektura poolingové části - typ poolingů, velikost jádra a posun,
4. počet skrytých vrstev a jejich neuronů,
5. typ aktivační funkce,

6. typ regularizace,

7. Learning Rate,

8. Momentum.

Při konstrukci sítě navrhované pro rozpoznávání plodnic vysokých hub bylo čerpáno z již zmíněné odborné literatury, ve které bylo optimální nastavení některých parametrů popsáno. V tomto konkrétním případě se jedná o parametry 5 - 8, které jsou doporučované nastavit takto:

- Typ aktivační funkce - ReLu.
- Typ regularizace - DropOut.
- Learning Rate - skoková funkce jdoucí od 0.1 k 0 (0.1 -> 0.01 -> 0.001 ...).
- Momentum - 0.9.

U zbylých parametrů (1 - 4) bylo rozhodnuto na základě inspirace u již vytvořených (kvalitativně odpovídajících) neuronových sítí (například [10]) a informací popsaných v teoretické části práce, respektive v podkapitole 3.3. Konkrétně byla navrhována síť obsahující do série napojené 2 konvolučně poolingové páry následované dvěma Fully Connected vrstvami, přičemž parametry jim nastavené budou probrány následovně.

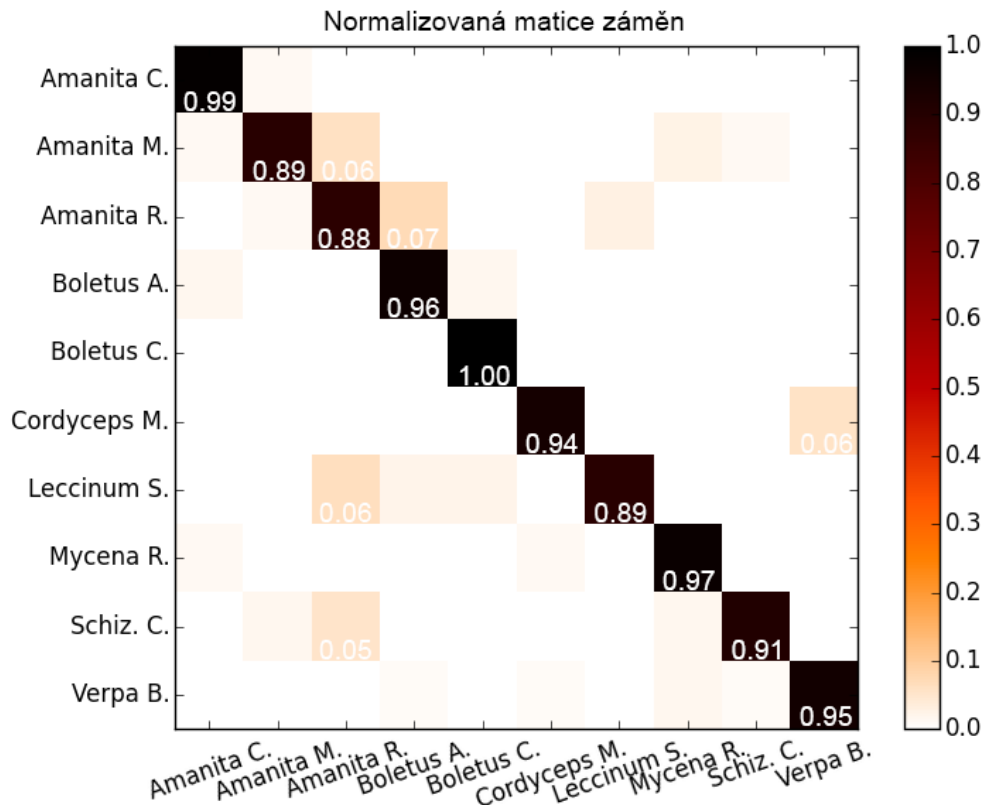
První konvoluční vrstva přivádí na vstup RGB snímek o velikosti 50×50 px. Na tento obrázek je aplikována konvoluce pomocí 96 jader o velikosti 3×3 s krokem o velikosti jedna. Následuje max-poolingová vrstva, která vždy z regionu o velikosti 3×3 vybírá maximální hodnotu. Jelikož krok okénka je má velikost 2, dochází k částečnému překrývání a jedná se tedy o Overlapping pooling popsaný v sekci 3.6.2. U druhé konvoluční vrstvy došlo k obměně parametrů. Počet konvolučních jader narostl na hodnotu 128 a jejich velikost se zvětšila na 4×4 s tím, že krok je o velikosti 2. Čtvrtou vrstvou je opět vrstva max-poolingu, která přesně odpovídá té první. Předposlední částí jsou 2 Fully Connected vrstvy. První má 1000 neuronů a dochází u ní k 50% DropOutu. Druhá jakožto výstupní vrstva má počet neuronů odpovídající počtu tříd. V tomto případě 10. Úplně na závěr je Softmax Loss vrstva, která se stará o procentuální vyhodnocení přesnosti u jednotlivých tříd.

V teoretické části této práce bylo zmíněno hned několik metod pro výpočet optimálních parametrů (vah neuronů) sítě. Jelikož nastavení trénovací metody je v rámci frameworku Caffe umožněné obdobně jako nastavování hyperparametrů bude použitý přístup zmíněn již v této části. Dle výčtu zmíněných metod z podkapitoly 3.5 a parametrů, které je nutné nastavit dle sekce 5.8.2, může být zřejmé, že byla vybrána Momentová metoda vycházející z dříve velice používané metody SGD. Tento výběr byl opět postaven na již dříve prezentovaných a zmíněných článcích a zadání úlohy plně vystačoval. Nebylo tedy zapotřebí využívat některou z adaptivních metod, která by mohla vést k výraznému prodloužení doby trénování.

5.8.3 Trénování

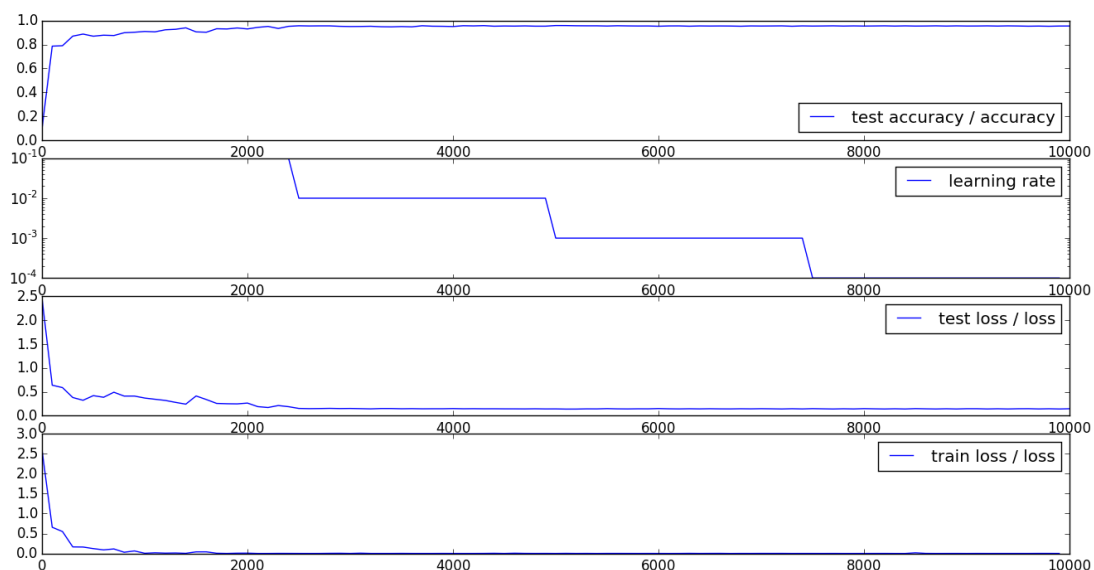
Trénování navrhnuté neuronové sítě bylo prováděno výhradně na serverech výpočetního centra Metacentrum. Ty spravují výkonné počítače umožňující trénování i na nejvýkonnějších grafických kartách. Díky tomu bylo možné zkrátit čas trénování z řádu hodin na několik minut, kdy byl finální čas výrazně závislý na velikosti vstupního obrazu. Před samotným trénováním bylo však potřeba připravit data tak, aby vyhovovala frameworku Caffe. Ten pracuje výhradně s binárními daty uloženými ve formátu H5. K jejich převodu byl napsán a využit script, který na základě vstupních parametrů reprezentuje potřebnou velikost jednotlivých snímků a volbu mezi formátem RGB a grayscale, vygeneruje jeden soubor, který obsahuje všechna data v binární podobě. Takto připravený soubor je následně nutné pomocí druhého vytvořeného scriptu rozdělit do dvou částí (train a test), jejichž poměr závisí na zvolené hodnotě. V této konkrétní úloze bylo využito rozdělení v poměru 70/30, kde 70 % dat odpovídalo trénovací množině a 30 % testovací. Při rozdělování byl kladen důraz na zastoupení všech tříd a podobnou četnost dat u obou subsetů.

Na takto předpřipravených datech již bylo možné spustit trénování CNN s vybranými hyperparametry za předpokladu, že byla správně nadefinována architektura sítě ve formě textového souboru. Pro lepší ilustraci ohledně formátu zápisu je možné nahlédnout do příloženého CD, kam byl tento soubor vložen. Výsledkem trénování jsou vždy minimálně dva soubory. Jeden s příponou ".caffemodel", plně odpovídající natrénované síti použitelné pro predikci, a druhý s příponou ".log", obsahující všechny informace o průběhu trénování. Druhý zmíněný byl velice důležitý při analýze výsledků, protože z něj bylo možné získat průběhy hodnot



Obrázek 5.5: Normalizovaná matice záměn znázorňující mezitřídní chybovost klasifikace. Maximální hodnota 1 reprezentuje 100% úspěšnost klasifikace.

Accuracy a Loss na testovacích i trénovacích datech a také průběh Learning Ratu. Všechny tyto hodnoty, získané u výše popsané sítě s použitím optimálních dat popsaných v podkapitole 5.6, byly vyneseny do grafu reprezentovaného Obrázkem 5.6. Z průběhu je možné vypočítat výsledné hodnoty přesnosti a ztrátové funkce. Rozdíl mezi hodnotami test a train Loss upozorňují na částečné přetrénování, avšak s minimálním dopadem na funkci výsledné neuronové sítě. Že se nejedná o výrazné přetrénování je zřetelné z obou testovacích průběhů. Pokud by k němu docházelo, nebyla by u hodnot Accuracy a Loss viditelná konvergence, ale například pokles, respektive nárůst. Další způsob hodnocení kvality, který je možný získat z hodnot uvnitř ".log"souboru je sestavení matice záměn. Ta nám znázorňuje mezi jakými třídami dochází k největší záměně. Z Obrázku 5.5, který matici záměn představuje, je pozorovatelné, že v nejhorším případě dochází pouze k chybným klasifikacím v řádech jednotek procent.



Obrázek 5.6: Grafické znázornění hodnot Accuracy a Loss získané v průběhu klasifikace.

5.9 Návrh a zpracování mobilní aplikace

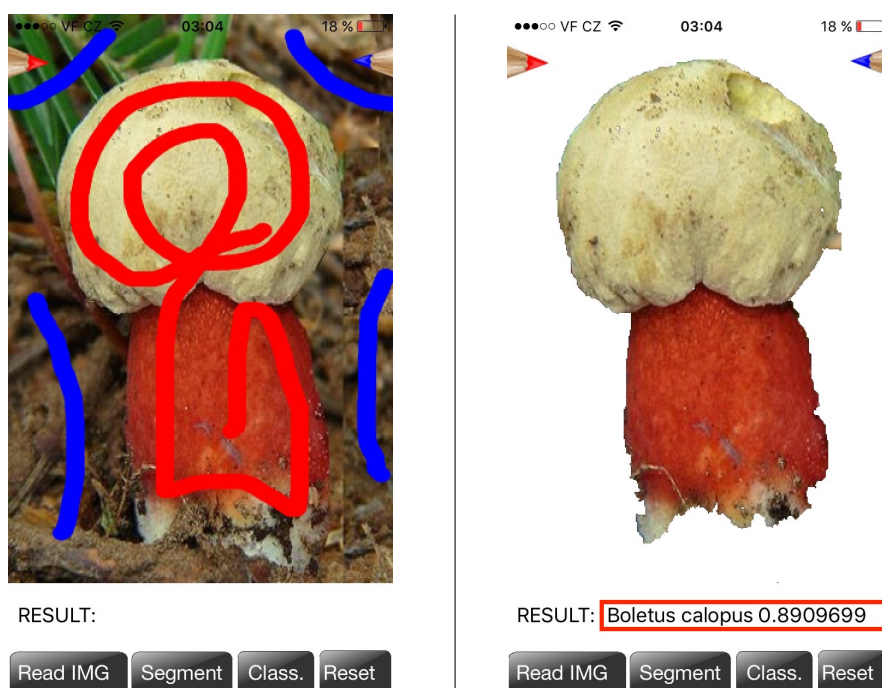
5.9.1 Navržené řešení

Závěrečný úkol této práce byl zaměřen na prozkoumání možných řešení pro použití natrénované CNN v reálném světě. Po důkladném zvážení se jako jediný přijatelný způsob nabízí její integrace do mobilní aplikace. Ta díky možnosti volné šířitelnosti a současné dostupnosti mobilních zařízení umožní její aplikovatelnost, v kterýkoliv čas a na kterémkoliv místě, i zcela běžnému uživateli. Dalším kladem je jednoduchá ovladatelnost při získávání snímku neznámé plodnice, její segmentaci a následné klasifikaci. K pořízení snímku je možné využít mimo jiné i fotoaparát zabudovaný v mobilním zařízení. Pro segmentaci, založené na metodě GrabCut, je potřeba označit popředí a pozadí na získaném snímku. K tomu je možné využít běžného dotyku. Poslední bod, a to klasifikace, je již plně automatizovaný a nevyžaduje žádný uživatelský zásah.

Další otázkou byla volba mobilní platformy. Ačkoliv přibližně 2/3 mobilních zařízení na trhu využívají operačních systémů Android, bylo vzhledem k osobním možnostem nutné se zaměřit pouze na vývoj aplikace určené pro OS iOS.

5.9.2 Výsledná aplikace

Vytvořená aplikace je pouze tzv. Single View aplikací. Pracuje tedy pouze na jedné stránce. Uživatelské rozhraní (UI¹) je poměrně strohé, nicméně jeho účel světlí prostředkům. Z Obrázku 5.7 je viditelné, že UI obsahuje celkem 6 tlačítek, kde 2 tlačítka slouží pro označování popředí/pozadí a zbylé k zresetování označení, nahrání analyzované fotografie, vyhodnocení segmentace a získání predikovaného druhu plodnice. Obdržený label třídy a pravděpodobnost, s jakou k ní byl vytvořený segment klasifikován, jsou zobrazeny vedle popisu "RESULT:".



Obrázek 5.7: Uživatelské rozhraní vytvořené iOS aplikace. Levý obrázek ilustruje označení popředí a pozadí pro segmentační metodu GrabCut. Pravý obrázek zobrazuje vysegmentovanou plodnici a výsledek klasifikace.

5.9.3 Použité technologie

Pro vývoj aplikace pro platformu iOS, která na základě jedné fotografie umožňuje rozpoznat plodnici vysoké houby, bylo využito vývojového prostředí Xcode. To je v současnosti nejpoužívanějším nástrojem pro vývoj iOS aplikací, jehož součástí je i řada simulátorů, na kterých je možné aplikaci otestovat. Další jeho

¹User Interface

výhodou je velice jednoduchá možnost importování frameworků třetích stran. Mezi ty se řadí například knihovna openCV, kterou bylo nutné využít pro implementaci segmentační metody GrabCut.

Pro psaní kódu aplikace bylo využito staršího programovacího jazyku ObjectiveC, na rozdíl od dnes častěji používaného Swiftu. Toto rozhodnutí bylo učiněno zejména proto, že práce s knihovnami openCV a Caffé je velice zjednodušena díky jejich přímé podpoře C++. Není tedy třeba využívat wrapperu, který by částí kódu převáděl do jazyka Swift.

Kapitola 6

Možnosti rozšíření a vylepšení

Jelikož praktická část této diplomové práce byla relativně velmi rozsáhlá, nedovolovala opravdu hlubokou analýzu jednotlivých částí. Z tohoto důvodu je možné dojít k alespoň minimálnímu rozšíření nebo vylepšením téměř ve všech bodech.

První, pravděpodobně nejpodstatnější, vylepšení je možné učinit prostřednictvím rozšíření datasetu. V rámci této práce se pracovalo s datasetem obsahujícím pouze 10 tříd. Přestože získané výsledky reprezentují 95% úspěšnost na testovací sadě, je takto natrénovaná neuronová síť v praxi téměř nevyužitelná. Další předpoklad, který se váže na zmíněné rozšíření je možnost výrazně většího zobecnování hledaných příznaků, a tím snížení hodnotící funkce a teoreticky i zvýšení procentuální úspěšnosti. Dataset by bylo možné rozšířit jak do počtu tříd, tak přidáním dalších augmentovaných dat, vzniklých například pomocí modifikace světlosti, ostrosti a barevnosti.

Další část, ve které je velký prostor pro vylepšení, souvisí s možným experimentováním ohledně architektury a hyperparametrů konvoluční neuronové sítě. V rámci této práce byl zaujat spíše konzervativní postoj, který vycházel především z již ověřených metod a postupů, a nedošlo tak k žádným průlomovým zjištěním. Nasnadě je také otázka využití některé z popsanych adaptivních metod pro nastavování parametrů sítě.

Poslední část, kterou lze eventuálně rozšířit, je mobilní aplikace jako finální produkt. Jak bylo zmíněno, její kompatibilita je pouze s operačním systémem iOS, a tak je možnost jí využít pouze u menší části všech zařízení. Pokud by však byla vytvořena stejná aplikace i pro OS Android, popřípadě Windows, bylo by téměř zaručené, že výsledný produkt by oslovil větší množství uživatelů cílové

skupiny. V neposlední řadě je třeba zmínit i možný prostor v úpravách designu uživatelského rozhraní vytvořené aplikace.

Kapitola 7

Závěr a zhodnocení

Hlavním cílem této diplomové práce bylo navrhnout a implementovat systém, který běžnému člověku umožní snadněji rozpoznávat plodnice vysokých hub. Při jeho návrhu byly zohledněny současné trendy v oblasti umělé inteligence, které řešení nasměrovali k využití konvolučních neuronových sítí. Z tohoto důvodu se velká část práce věnovala teoretickým informacím, na kterých mohl být finální produkt založen. Praktická část se zpočátku zabývala tvorbou datasetu. Experimentální formou bylo otestováno několik různých metod předzpracování a zpracování obrazu. Nejpřijatelnějších výsledků klasifikace bylo dosaženo při použití vysegmentovaných plodnic umístěných na pozadí s neměnnou barvou, která zaručovala jejich optické oddělení. Neuronová síť natrénovaná na segmentovaných datech bylo dosaženo 79.3% přesnosti klasifikace podpořené pěti iteracemi cross-validace. Takto natrénovaná síť již odpovídala některým zavedeným standardům, nicméně stále vykazovala velké přetrénování. Z tohoto důvodu bylo rozhodnuto o přidání augmentovaných dat, která přetrénování zabraňují. Došlo tedy k zvětšení datasetu o trojnásobek, kdy augmentovaná data byla vytvořena pomocí náhodné rotace a náhodného posunu v obou osách. Současně s navýšením datasetu došlo i k zvětšení přesnosti, a to konkrétně o 5.2 % na hodnotu 84.5 %. Další otázkou, ohledně tvaru vstupních dat, byla jejich velikost. O té bylo opět rozhodnuto pomocí experimentu, jehož výsledkem bylo doporučení velikosti 50×50 px. S touto velikostí byla dosažena přesnost 89.5 %. Závěrečný experiment týkající se trénovacích dat byl zaměřen na porovnání černobílých a barevných snímků. Jelikož RGB snímek nese podstatně větší množství informací o obraze, byl tomuto experimentu přikládán velký význam. Ten byl potvrzen navýšením přesnosti o 10.6 % na finálních 95.1 %. V dalším bodě byla popsána architektura použité

sítě, které byla z velké části inspirována již dříve popsány sítěmi. K sestavení sítě byl využit framework Caffe a veškeré trénování bylo z důvodu výpočetního výkonu prováděno na serverech Metacentra. Poslední část práce se zabývá návrhem a tvorbou použitelného řešení pro běžného uživatele (houbaře). Z důvodu velké dostupnosti a možností, které nám mobilní platforma umožňuje, byla vytvořena aplikace pro operační systém iOS. Ta jako výsledný produkt vyžadující minimální příspěvek uživatele, umožňuje výrazné zjednodušení rozpoznávání hub v průběhu jejich sběru.

Literatura

- [1] Český rozhlas: Zaměřeno na jedovaté houby [Online]
Available at:
http://www.rozhlas.cz/priroda/rostliny_houby/_zprava/647682, 2009

- [2] Důmazahrada.cz: OBRAZEM: JAK URČOVAT HOUBY [Online]
Available at:
<http://www.dumazahrada.cz/zahrada/2012/9/6/jak-rozpoznavat-houby>,
2012

- [3] LECUN Y., RANZATO M. A.: Deep Learning Tutorial
Talks by Members of CBLL, CIML, 2013

- [4] Wikipedia: Precision and recall [Online]
Available at:
https://en.wikipedia.org/wiki/Precision_and_recall, 2016

- [5] ERIC K.: Everything You Wanted to Know about the Kernel Trick [Online]
Available at:
http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html,
2016

- [6] LECUN Y., BOSER B., DENKER J. S., HENDERSON D., HOWARDR. E.,
HUBBARD W., JACKEL L. D.: Backpropagation Applied to Handwritten Zip
Code Recognition. Neural Computation
MIT Press Journals, 1989

- [7] LECUN Y., WU Q. Z., JACKEL L. D.: On-line recognition of limited-
vocabulary Chinese character using multiple convolutional neural networks.
International Symposium on circuits and systems, IEEE, 1993

- [8] LECUN Y., BENGIO Y.: Word-level training of a handwritten word recognizer based on convolutional neural networks.
Proc. of the International Conference on Pattern Recognition, IEEE, 1994
- [9] LECUN Y., BENGIO Y.: Convolutional Networks for Images, Speech, and Time-Series
The Handbook of Brain Theory and Neural Networks, MIT Press, 1995
- [10] KRIYHEVSKY A., SUTSKEVER I., HINTON G.E.: ImageNet Classification with Deep Convolutional Neural Networks
Advances in Neural Information Processing Systems 25, Curran Associates Inc., 2012
- [11] Mashable.com: New algorithm gives photos Picasso-style makeovers [Online]
Available at:
<http://mashable.com/2015/08/29/computer-photos>, 2015
- [12] HUBEL D. H., WIESEL T. N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.
The Journal of physiology, IEEE, 1962
- [13] Stanford CS: CS231n: Convolutional Neural Networks for Visual Recognition. [Online]
Available at:
<http://cs231n.github.io/>, 2016
- [14] DUCHI J., HAZAN E., SINGER E.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization
J. Mach. Learn. Res., JMLR.org, 2011
- [15] HINTON G.: Neural Networks for Machine Learning [Online]
Available at:
http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_1ec6.pdf, 2014
- [16] KINGMA D. P., BA J.: Adam: A Method for Stochastic Optimization
CoRR , CoRR, 2014
- [17] SRIVASTAVA N., KRIZHEVSKY H., KRIZHEVSKY A., SUTSKEVER I., SALAKHUTDINOV R.: Dropout: A Simple Way to Prevent Neural Networks

from Overfitting

Journal of Machine Learning Research, IEEE, 2014

- [18] OTSU N.: A threshold selection method from gray-level histograms
IEEE Trans. Sys., Man., IEEE, 1979
- [19] ZUIDERVELD K.: Contrast Limited Adaptive Histogram Equalization
P. Heckbert: Graphics Gems IV, Academic Press, 1994
- [20] CANNY J.: A Computational Approach To Edge Detection
IEEE Trans. Pattern Analysis and Machine Intelligence, 1986
- [21] HARRIS C.: A combined corner and edge detector
Proceedings of the 4th Alvey Vision Conference. pp. 147–151, 1988
- [22] BOYKOV Y.Y., JOLLY M.P.: Interactive graph cuts for optimal boundary
and region segmentation of objects in N-D images
Proceedings Eighth IEEE International Conference on Computer Vision, IEEE,
2001
- [23] JIA, Y.: Caffe: An open source convolutional architecture for fast feature
embedding [Online]
Available at:
<http://caffe.berkeleyvision.org>, 2013

Kapitola 8

Přílohy

Na přiloženém CD se v první řadě nachází elektronická verze této diplomové práce. Dále je přiložen vytvořený model neuronové sítě pojmenovaný "*mushrooms.caffemodel*". Ten reprezentuje finální podobu neuronové sítě využitelné ke klasifikaci. V poslední řadě je ve složce *APP* vložen Xcode projekt, z kterého je možné vygenerovat ".ipa" file použitelný k instalaci vytvořené aplikace.

K nalezení je tedy:

- elektronická verze diplomové práce,
- *caffemodel* použité neuronové sítě,
- Xcode projekt využitelný k instalaci vytvořené aplikace na iOS zařízení.