

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Analýza historických katastrálních map

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan ČERNOGURSKÝ**
Osobní číslo: **A18B0383P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační systémy**
Téma práce: **Analýza historických katastrálních map**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prostudujte metody pro zpracování obrazu pomocí neuronových sítí.
2. Seznamte se s dostupnými knihovnami pro implementaci neuronových sítí.
3. Seznamte se s dodanými mapovými podklady a vytvořte datovou sadu vhodnou pro trénování modelů a vyhodnocení úspěšnosti.
4. Navrhněte a implementujte alespoň jednu metodu, která umožní detekci palcových značek.
5. Provedte vyhodnocení úspěšnosti navržené metody na dodaných datech a srovnání jejích výsledků s metodami klasickými.
6. Zhodnoťte dosažené výsledky a navrhněte možná rozšíření.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Ladislav Lenc, Ph.D.**
Nové technologie pro informační společnost

Datum zadání bakalářské práce: **5. října 2020**
Termín odevzdání bakalářské práce: **6. května 2021**

L.S.

Doc. Dr. Ing. Vlasta Radová
děkanka

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

Poděkování

Rád bych poděkoval Ing. Ladislavu Lencovi, Ph.D. za odborné vedení, cenné připomínky a rady při vedení této bakalářské práce.

Jan Černogurský

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 6. května 2021

Jan Černogurský

Abstract

The bachelor thesis focuses on the issue of image processing using neural networks. The aims of the work are to get acquainted with the available libraries for the implementation of neural networks, and with the supplied historical follows an analysis of the maps and understanding of their issues. In the practical part, the aim is to create data sets from the supplied data, which will be suitable for training and evaluation of neural network models, and to design and implement at least one method that will allow the detection of inch marks in cadastral maps using neural networks. The output is an evaluation of the success of the proposed methods on the supplied data and a comparison of obtained results with classical methods.

Abstrakt

Bakalářská práce se zaměřuje na problematiku zpracování obrazu pomocí neuronových sítí. Cíle práce jsou seznámení se s dostupnými knihovnami pro implementaci neuronových sítí, seznámení se s dodanými historickými mapovými podklady, jejich následná analýza a porozumění jejich problematice. V praktické části je cílem vytvoření datové sady z dodaných podkladů, která bude vhodná pro trénování modelů a vyhodnocení tohoto trénování a následné navrhnutí a implementace alespoň jedné metody, která umožní detekci palcových značek v katastrálních mapách pomocí neuronových sítí. Výstupem je vyhodnocení úspěšnosti navržené metody na dodaných datech a srovnání jejich výsledků s metodami klasickými a následné zhodnocení dosažených výsledků a navrhnutí možných rozšíření.

Obsah

1	Úvod	1
2	Analýza obrazu	2
2.1	Digitální obrázek	3
3	Segmentace	5
3.1	Základní metody segmentace	5
3.2	Regionální metody	5
3.3	Prahovací metody	7
3.4	Hranové metody	10
4	Umělé neuronové sítě	13
4.1	Vrstvy	14
4.2	Učení umělé neuronové sítě	15
4.2.1	Data augmentation	16
4.3	Konvoluční neuronová síť (CNN)	16
4.4	Plně propojená konvoluční neuronová síť (FCN)	17
4.5	Long Short Term Memory (LSTM)	19
5	Použité technologie	20
5.1	Python	20
5.2	OpenCV	20
5.3	NumPy	20
5.4	Tensorflow	20
5.5	Inch Evaluator	20
5.6	Keras	21
5.6.1	Vrstvy Keras	21
6	Data a jejich zpracování	23
6.1	Historické katastrální mapy	23
6.2	Původní data	23
6.3	Ground-truth data	24
6.4	Dataseťy	26
7	Detekce palcových značek	27
7.1	Stávající metoda dekece palcových značek	29
7.2	Architektura navrhované metody implementující FCN	29

7.3	Proces zpracování dat	31
7.3.1	Výřez celé hranice mapy	31
7.3.2	Výřez vzorků hranice mapy	31
8	Experimenty	33
8.1	Přístupy k vyhodnocení výsledků	33
8.2	Pixelové metody	33
8.2.1	Metriky	34
8.3	Učení	37
8.4	Predikování	43
9	Dosažené výsledky	44
10	Závěr	46
11	Použité zkratky	47
12	Přílohy	48
13	Příručka	49
13.1	Instalace	49
13.2	Spuštění	49
	Literatura	50

1 Úvod

V posledních letech je stále efektivnější ukládat data a informace na elektronická úložiště a zařízení, jako jsou například počítače. Příkladem mohou být historické dokumenty, jako jsou třeba historické mapy. Velké množství historických dat vzniká skenováním dochovaných historických dokumentů. V rámci projektu se řeší zpřístupnění map stabilního katastru. Mapy jsou ručně kreslené. Hlavní cíl je určit pozici mapového listu na základě textové informace obsažené v tzv. nomenklatuře. Získat výřez map (mapový rám) a složit vše do jednoho bežešvého celku vhodného pro prezentaci. Mapy jsou uloženy v podobě jednotlivých mapových listů. Množina sousedních mapových listů tvoří katastrální území.

Cílem této bakalářské práce je vytvořit program schopný analyzovat dané mapové podklady a pomocí neuronové sítě automaticky identifikovat tzv. palcové značky, které určují měřítko mapy a jsou rozvňěž použity ke skládání jednotlivých map. Přínosem neuronové sítě by mělo být to, že se tyto značky naučí v mapových listech rozpoznávat a minimalizuje nutnou manuální anotaci mapových listů. Ta by se následně měla omezit pouze na kontrolu. Jelikož jsou mapové listy ručně kreslené a navíc staré více než sto let, mohou mít řadu vad, jakými mohou být například rotace celého obsahu, digitální šum, různé deformace a mnoho dalších. Tyto vady by mohly znepříjemňovat již takto neefektivní manuální skládání.

Přístupů k implementaci neuronové sítě, které by pomohly tento problém vyřešit, je několik. Záměr této bakalářské práce je se nejdříve seznámit s těmito přístupy a následně z nich vybrat nejlépe se jevící řešení a implementovat jej. Po úspěšné implementaci dojde k zhodnocení dosažených výsledků a porovnání navrženého a implementovaného řešení s doposud využívanou metodou, která je založena na klasických metodách zpracování obrazu.

2 Analýza obrazu

Analýza obrazu se používá jako základní nástroj pro rozpoznávání, diferenciaci a kvantifikaci různých typů obrazů, včetně obrazů ve stupních šedi a barevných obrazů, multispektrálních obrazů. Je důležité si uvědomit, že analýza obrazu je součástí širšího pole známého jako zpracování obrazu, kde hlavními základními myšlenkami jsou zlepšení vizuální kvality animace a extrahování užitečné informace nebo funkce. Analýza je založena na různých vlastnostech obrazu, jako jsou barva, lesk, morfologie objektů a textura. Akce zpracování obrazu lze rozdělit do tří podoblastí:

1. Komprese obrazu - snižuje požadavky na paměť odstraněním nadbytečnosti přítomné v obraze, tj. obrazové informace, která není pro lidské oko vnímatelná.
2. Předzpracování obrazu - spočívá ve zlepšení vizuální kvality obrazu snížením šumu, kalibrací pixelů a standardizací, vylepšením detekce hran a zvýšením spolehlivosti kroku analýzy obrazu na základě objektivních a dobře zavedených kritérií. Termín předběžné zpracování obrazu se obecně označuje pro všechny manipulace s obrazem, z nichž každý vytváří nový obraz.
3. Analýza obrazu - obvykle vrací číselné hodnoty a nebo grafické informace o vlastnostech obrazu, které jsou vhodné pro klasifikaci, detekci defektů nebo predikci některých kvalitativních vlastností zobrazovaného objektu. Termín analýza obrazu se používá, když výstupem je číslo nebo rozhodnutí, ne obraz.

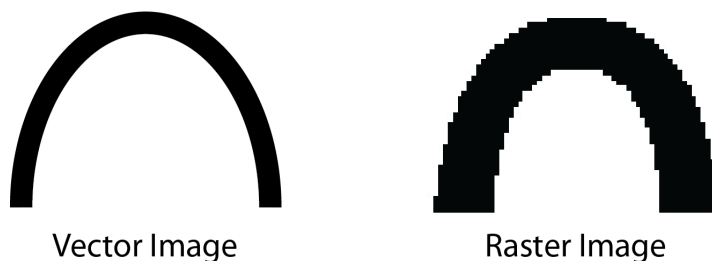
Tyto akce zpracování spolu souvisí a mohou mít pro každou aplikaci jiný účinek nebo výstup.

2.1 Digitální obrázek

Digitální obraz je numerická 2D nebo 3D reprezentace skutečného fyzického objektu nebo scény, ze které můžeme získat přesnou prostorovou (geometrickou) a nebo spektrální (v případě hyperspektrálního obrazu) reprezentaci s dostatečnými detaily (rozlišením) pro zpracování, komprese, skladování, tisk a zobrazení. Může se jednat o digitální obrázek vektorového nebo rastrového typu, v závislosti na tom, zda je rozlišení obrazu pevné nebo ne.

Vektorový obrázek nezávisí na rozlišení, což znamená, že obraz lze zvětšit nebo zmenšit, aniž by to ovlivnilo kvalitu výstupu. Termín digitální obrázek obvykle odkazuje na rastrové obrázky, které se hojně nazývají bitmapovými obrázky. Kromě toho se také používá k datům spojeným s body rozptýlenými v trojrozměrné oblasti, například vytvořenými tomografickým zařízením. V takovém případě se každému údaji říká voxel.

Rastrové obrázky jsou elektronické soubory, které se skládají z diskretních obrazových prvků, nazývaných pixely (zkratka pro obrazové prvky). S každým pixelem je spojeno číslo, které je průměrným zářením (nebo jasnem) relativně malé oblasti ve scéně, což představuje barvu nebo úroveň šedi v jednom bodě obrazu.



Obrázek 2.1: Srovnání vektorového a rastrového obrázku

Zdroj: <https://www.barcode.graphics/>

Nejběžněji používanou metodou pro vytváření rastrových obrázků (tj. Digitalimaging nebo získávání obrázků) je digitální fotografování za použití CCD kamery v procesu zvaném digitalizace. Proces digitalizace vyžaduje mapování obrazu na mřížce a kvantizaci úrovně intenzity. Digitalizace je

proces převodu analogového signálu na digitální signál, známý jako A / D (analogově-digitální) převod. U rastrových obrazů je analogový napěťový signál (z kteréhokoli z několika typů zobrazovacích snímačů), úměrný množství světla odráženého nebo přenášeného digitalizovanou položkou, rozdělen na diskrétní číselné hodnoty. Jinými slovy, proces převádí obraz na řadu malých obrázkových čtvercových prvků nebo pixelů, které jsou buď černé nebo bílé (binární), konkrétní odstín šedé (stupně šedi) nebo barvy. Každý pixel je reprezentován jednou nebo řadou binárních číslic. Měřením barvy (nebo úrovně šedé pro černobílé fotografie) obrazu ve velkém počtu bodů můžeme vytvořit digitální přiblížení obrazu, ze kterého lze rekonstruovat kopii originálu.

3 Segmentace

Segmentace obrazu je v počítačovém vidění kritickým procesem. Zahrnuje rozdělení vizuálního vstupu na segmenty, aby se zjednodušila analýza obrazu. Segmenty představují objekty nebo části objektů a zahrnují sady pixelů nebo „superpixely“. Segmentace obrazu třídí pixely na větší komponenty, což eliminuje potřebu považovat jednotlivé pixely za jednotky pozorování. Cílem segmentace je rozklad obrazu na části, které jsou smysluplné s ohledem na konkrétní aplikaci [4]. Existují tři úrovně analýzy obrazu:

- Klasifikace - kategorizace celého obrázku do třídy jako „lidé“, „zvířata“, „venku“ a podobně.
- Detekce objektů - detekce objektů v obrazu a vytvoření obdélníku kolem nich, například osoby nebo automobily.
- Segmentace - identifikace částí obrazu a porozumění tomu, k jakému objektu patří. Segmentace je základem pro provádění detekce a klasifikace objektů.

Existují další techniky segmentace obrazu, které se v minulosti běžně používaly, ale jsou méně účinné než jejich protějšky s hlubokým učením, protože používají rigidní algoritmy a vyžadují lidský zásah a odborné znalosti.

3.1 Základní metody segmentace

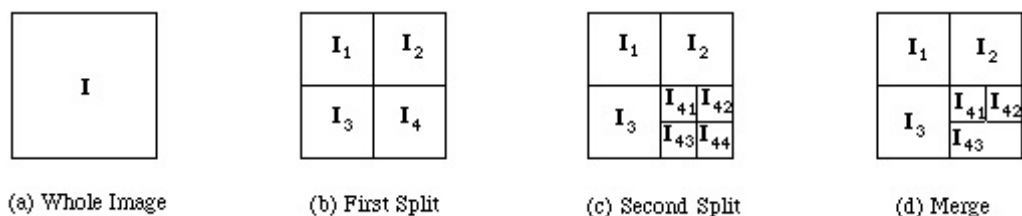
Přístupů k segmentaci obrazu existuje mnoho, avšak pro představení by nám naprosto měly stačit základní metody jako jsou metody regionální, prahové a hranové.

3.2 Regionální metody

Regionální metody (anglicky Region-based methods) jsou metody detekující přímo oblasti v obraze namísto hranic (hran) těchto oblastí. Jsou efektivnější pro zašumělý obraz než jiné metody. Je-li v obraze hodně šumu, hranové operátory obtížně detekují hrany. Hlavním segmentačním kritériem pro detekci oblastí v obraze je homogenita oblastí. Kritériem homogenity mohou být: úroveň šedi, barva, textura, tvar, model, apod.

Split and merge

Metoda dělení a spojování je založena na přístupu rozděl a panuj. Metoda je založena na quad-tree reprezentaci dat, kdy podobraz je rozdělen na čtyři kvadranty, pokud jsou jeho měřené atributy nehomogenní. Pokud však jsou sousední čtverce homogenní, jsou následně spojeny v jednu oblast. Jinými slovy je vstupní obraz postupně dělen na dílčí oblasti (nejčastěji se jedná o čtverce, avšak tvar struktury dílčí oblasti se může lišit, ale vždy musí být předem definován), dokud nejsou dílčí oblasti dostatečně malé pro segmentaci. Potom se k vytvoření konečných výsledků segmentace použije příslušné pravidlo sloučení. Tento proces se dělí na čtyři fáze: rozdělení obrazu na malé subregiony, sloučení podobných subregionů a prostorově přilehlých regionů a eliminování malých regionů.

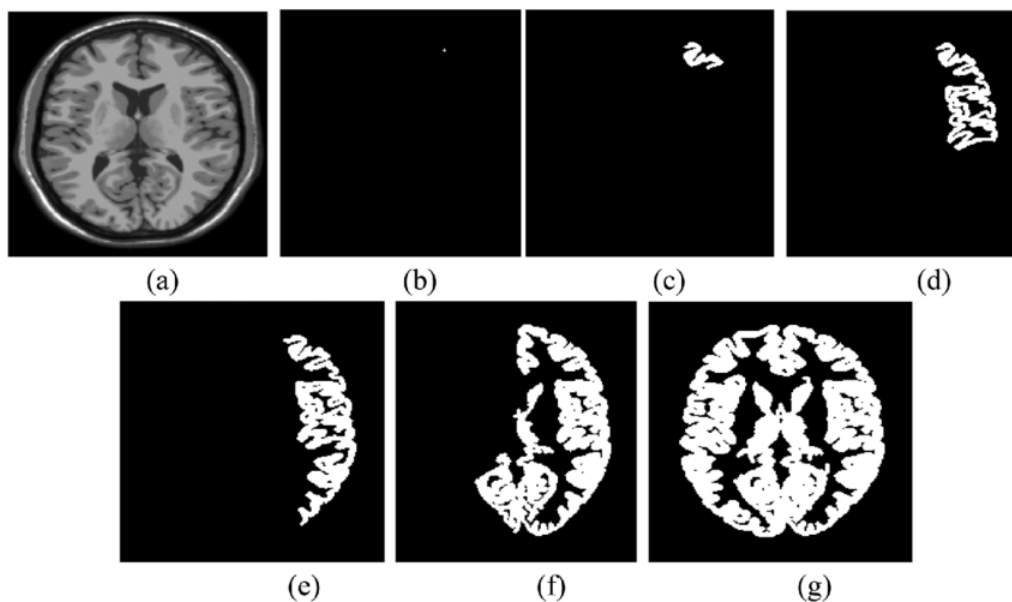


Obrázek 3.1: Split and merge metoda

Narůstání oblasti (Region growing)

Jedná se o metodu s konceptuálně nejjednodušším postupem. Metoda je založená na hledání podobnosti mezi sousedními pixely na určité oblasti. To znamená, že pixely, které mají podobné atributy, jsou seskupeny do jedinečných oblastí. Stejně jako u všech technik segmentace je použití intenzity šedé úrovně nejběžnějším prostředkem přiřazování podobnosti, ale existují i jiné možnosti, jako je varianční, barevná a multispektrální funkce.

Techniky šíření oblasti shlukují pixely, které představují homogenní oblasti v obraze. Oblasti se zvětšují seskupením sousedních pixelů, jejichž vlastnosti, například intenzita, se liší o méně než předem definované proměnné. Každému rozšiřovanému regionu je ve výstupním obrázku přiřazen jedinečný celočíselný štítek. Tato třída algoritmů má tendenci dobře fungovat pro náročné a ne moc přehledné snímky, protože je adaptivní a méně náchylná k účinkům sousednosti, šumu a nejednoznačnosti hranic.



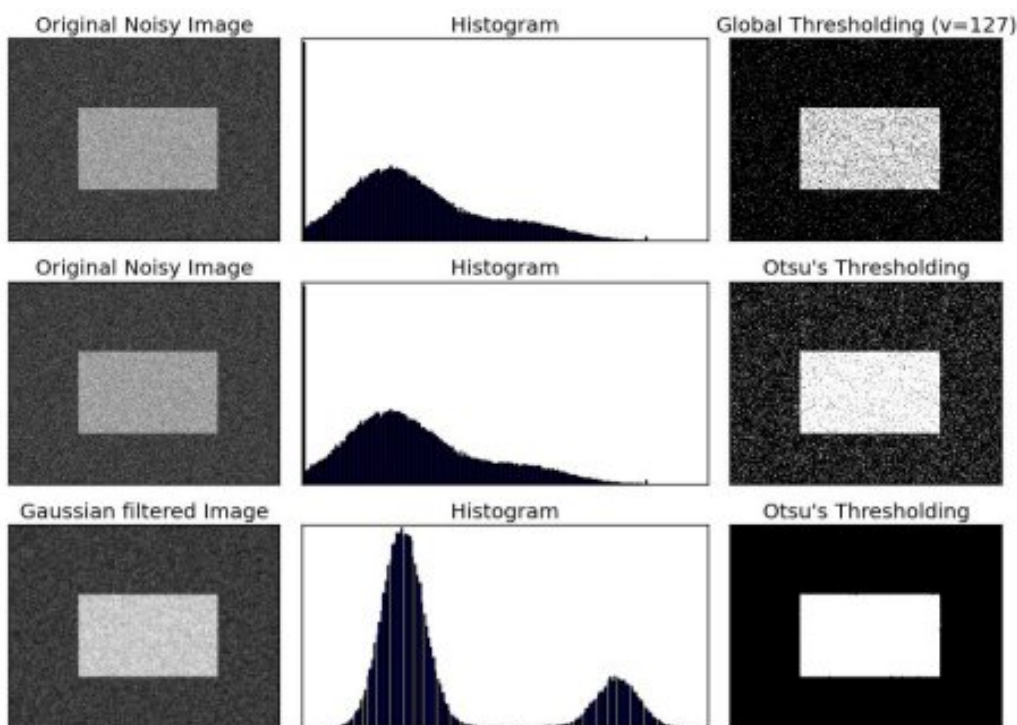
Obrázek 3.2: Ukázka Region growing metody na MRI snímku mozku

Zdroj: <https://www.researchgate.net/>

3.3 Prahovací metody

Jedná se o jednu z nejstarších a nejjednodušších metod segmentací obrazu. Patří mezi nejrychlejší segmentační metody a je možné je využít i v reálném čase. Jedná se o nenáročné metody na hardwarovou výbavu a díky tomu i o nejčastěji využívané metody.

Práh obrazu, který extrahuje objekt z pozadí ve vstupním obrazu, je jednou z nejběžnějších aplikací v analýze obrazu. Například při automatickém rozpoznávání strojově tištěných nebo ručně psaných textů, při rozpoznávání tvarů objektů a při vylepšování obrazu je prahování nezbytným krokem pro předzpracování obrazu. Metody prahování obrazu oddělují dvouúrovňové prahování pixely obrazu na dvě oblasti (tj. Objekt a pozadí); jedna oblast obsahuje pixely s šedými hodnotami menšími než prahová hodnota a druhá oblast obsahuje pixely s šedými hodnotami většími než prahová hodnota. Dále, pokud jsou pixely obrazu rozděleny do více než dvou oblastí, jedná se o víceúrovňové prahování. Obecně se práh nachází ve zjevném a hlubokém údolí histogramu. Pokud však údolí není tak zřejmé, je velmi obtížné určit prahovou hodnotu. V mnoha aplikacích zpracování obrazu se úroveň šedi pixelů patřících k objektu podstatně liší od úrovně šedi pixelů patřících k pozadí. Prahové hodnoty se poté stávají jednoduchým, ale účinným nástrojem k oddělení objektů od pozadí.



Obrázek 3.3: Ukázka tří prahovacích přístupů

Zdroj: <https://docs.opencv.org/>

Postup

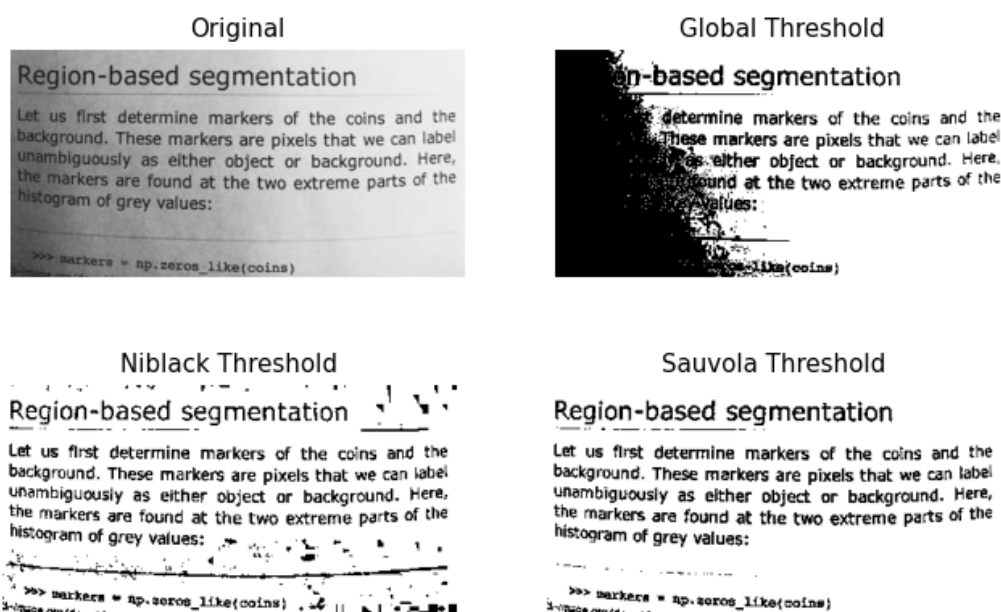
Histogram obsahuje informace o počtu pixelů v obraze s konkrétní hodnotou šedi. Analýzou takového histogramu získáme práh T (Thresholding), respektive množinu prahů T_1, T_2, \dots, T_n , pomocí kterých pak můžeme obraz rozdělit na podoblasti:

$$R(i, j) = \begin{cases} 1 & \text{pro } f(i, j) < T \\ 0 & \text{pro jiné} \end{cases}$$

Získání prahu z histogramu ovšem není triviální úlohou. Histogramy reálných obrazů neobsahují ostré hrany a špičky, ale mohou mít několik vrcholů různé výšky a strmosti. K rozpoznání, jedná-li se o vhodný vrchol (ostrý a hluboký) můžeme použít poměr plochy vrcholu a plochy jeho pláště. Hodnota prahu se určuje jako lokální minimum mezi dvěma největšími lokálními maximy.

Niblack a Sauvola

Prahové hodnoty Niblack a Sauvola jsou techniky lokálního prahování, které jsou užitečné pro obrázky, kde pozadí není jednotné, zejména pro rozpoznávání textu. Místo výpočtu jediné globální prahové hodnoty pro celý obraz se pro každý pixel vypočítá několik prahových hodnot pomocí konkrétních vzorců, které zohledňují průměr a standardní odchylku okolí pixelu (definované oknem soustředěným kolem pixelu).



Obrázek 3.4: Niblack a Savola metody

Zdroj: <https://scikit-image.org/>

Otsu

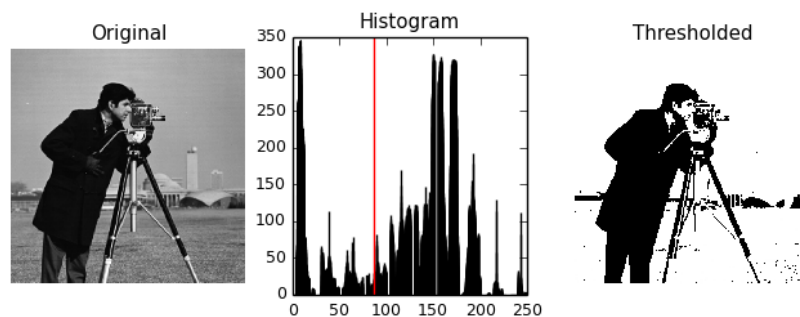
Problém s jednoduchým prahováním spočívá v tom, že musíte ručně určit prahovou hodnotu. Můžeme ručně zkontrolovat, jak dobrý je práh, když vyzkoušíme různé hodnoty, ale je to zdlouhavé a v reálném světě celkem nepraktické. Potřebujeme tedy způsob, jak automaticky určit prahovou hodnotu. Technika Otsu pojmenovaná podle jejího tvůrce Nobuyuki Otsu je dobrým příkladem automatického prahování.

Metoda zpracovává histogram obrazu, segmentuje objekty minimalizací rozptylu na každé z tříd. Obvykle tato technika vytváří vhodné výsledky pro

bimodální obrazy. Histogram takového obrazu obsahuje dva jasně vyjádřené vrcholy, které představují různé rozsahy hodnot intenzity.

Automatické globální prahové algoritmy mají obvykle následující kroky:

1. Zpracování vstupního obrazu
2. Získání histogramu obrazu (distribuce pixelů)
3. Vypočtení prahové hodnoty T
4. Vyměnění obrazových pixelů za bílé v těch oblastech, kde je sytost větší než T , a za černé v opačných případech.



Obrázek 3.5: Ukázka automatického Otsu prahování

Zdroj: <http://scipy-lectures.org/>

3.4 Hranové metody

Detekce hran je základním nástrojem pro segmentaci obrazu. Metody detekce okrajů transformují původní obrázky na okraje jejich obsahu plynoucí ze změn šedých tónů v obrazu. Při zpracování obrazu, zejména v počítačovém vidění, detekce hran zpracovává lokalizaci důležitých variací obrazu na úrovni šedé a detekci fyzikálních a geometrických vlastností objektů scény. Jedná se o základní proces detekující obrysy objektu a hranice mezi objekty a pozadím v obraze.

Detekce hran je nejznámější přístup používaný k detekci významných nespojitostí v hodnotách intenzity. Okraje jsou lokální změny v intenzitě obrazu. Hrany se obvykle vyskytují na hranici mezi dvěma oblastmi. Hlavní rysy lze extrahovat z okrajů obrázku. Detekce hran má hlavní funkci pro analýzu obrazu. Tyto funkce používají pokročilé algoritmy počítačového vidění.

Detekce hran je aktivní oblastí výzkumu, protože umožňuje analýzu obrazu na vyšší úrovni. V úrovni šedé jsou tři různé typy diskontinuit (bod, čára a hrany). Prostorové masky lze použít k detekci všech tří typů nespojitostí v obraze.

Existuje mnoho technik detekce hran využívaných pro segmentaci obrazu. Mezi nejčastěji používané techniky detekce hran založené na diskontinuitě patří Roberts Edge Detection, Sobel Edge Detection, Prewitt Edge Detection, Kirsh Edge Detection a Canny Edge Detection.

Sobel Edge Detection

Sobelova metoda detekce hran pro segmentaci obrazu vyhledá hrany pomocí Sobelovy aproximace k derivaci. Předpovídá hrany v těch bodech, kde je gradient nejvyšší. Technika Sobel zvýrazňuje oblasti s vysokou prostorovou frekvencí, které odpovídají hranám. Obecně se používá k nalezení odhadované absolutní velikosti gradientu v každém bodě na vstupním obrázku ve stupních šedi. Sobelova metoda byla dříve nejoblíbenější metodou detekce hran, dnes již existují efektivnější metody.

Cannyho detektor

V praxi je technika detekce hran Canny jednou ze standardních technik detekce hran. Poprvé ji vytvořil John Canny pro svou magisterskou práci na MIT v roce 1983 a stále svou efektivitou překonává mnoho novějších algoritmů, které byly od té doby vyvinuty. Chceme-li najít hrany oddělením šumu od obrazu před nalezením hran obrazu, je Canny ideální metodou.

Kroky Canny metody:

- Spojení obrazu $f(r, c)$ s Gaussovou funkcí. Tím se získá plynulý obraz.
- Použití operátoru prvního rozdílového gradientu k výpočtu síly hrany, poté se získá velikost a směr hrany.
- Aplikování maximálního a kritického potlačení na velikost gradientu.
- Aplikování prahové hodnoty na obrázek, který nemá maximální potlačení.

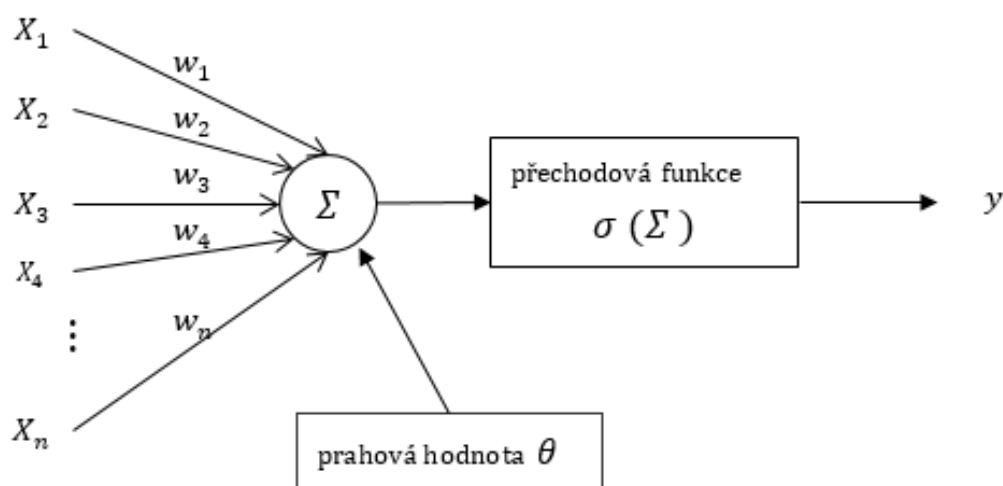


Obrázek 3.6: Ukázka Canny hranové detekce

Zdroj: <https://www.slideserve.com/>

4 Umělé neuronové sítě

Neuronové sítě jsou matematické modely, které k ukládání informací používají učící se algoritmy inspirované mozkiem. V dnešní době se v této oblasti často používá pojem strojové učení, což je vědecká disciplína zabývající se návrhem a vývojem algoritmů, které počítačům umožňují učit se na základě dat, například z dat senzorů nebo dat uložených v databázích. Hlavním zaměřením výzkumu strojového učení je automaticky se naučit rozpoznávat složité vzorce a přijímat inteligentní rozhodnutí na základě dat. Strojové učení tedy úzce souvisí s oblastmi, jako jsou statistika, dolování dat, rozpoznávání vzorů a umělá inteligence. Předností neuronových sítí je schopnost



Obrázek 4.1: Schéma neuronové sítě

Zdroj: <http://po.licka.cz/>

učit se, tedy zapamatovat si kombinace, které vedly k požadovanému výstupu a u nových vstupů se potom obracet na svoji paměť a na základě zkušeností odhadovat nový výsledek. Umělá neuronová síť (Artificial Neural Network – ANN), také známá jako simulovaná neuronová síť (Simulated Neural Network – SNN) či neuronová síť, je skupina umělých neuronů, jenž jsou vzájemně propojeny. Ve většině případů se jedná o adaptivní systém, který mění svou strukturu založenou na vnitřních nebo vnějších informacích, které jí proudí, jinak řečeno, dokáže se autonomně měnit podle získaných dat (dokáže se učit a na základě naučených dat se přizpůsobit datům novým).

Neuronové sítě se skládají z neuronů, které mají libovolný počet vstupů a jen jeden výstup. Citlivost vstupů je určena vahami přiřazenými jednotlivým vstupním proměnným. Každý vstupní prvek je vynásoben danou vahou a od tohoto součinu se odečte práh, který aktivuje výstup neuronu. Následně přechodová funkce transformuje vnitřní potenciál a poté dochází k vytvoření výstupu.

4.1 Vrstvy

Základní strukturou dat v neuronových sítích je vrstva. Vrstva je modul pro zpracování dat, který bere jako vstup jeden nebo více tenzorů a na výstupu je jeden nebo více tenzorů. Některé vrstvy jsou bezstavové, avšak častěji mají jednotlivé vrstvy stav jasně definován.[2] U vícevrstevných neuronových sítí rozlišujeme tři typy vrstev: vstupní, výstupní a skryté (jsou skryty vnějšimu pozorovateli).

Vstupní vrstva

Vstupní vrstva komunikuje s externím prostředím, které představuje vzor neurální sítě. Jeho úkolem je vypořádat se pouze se všemi vstupy. Tento vstup se přenesení do skrytých vrstev, které jsou vysvětleny níže. Vstupní vrstva by měla představovat podmínku, pro kterou trénujeme neuronovou síť. Každý vstupní neuron by měl představovat nějakou nezávislou proměnnou, která má vliv na výstup neuronové sítě.

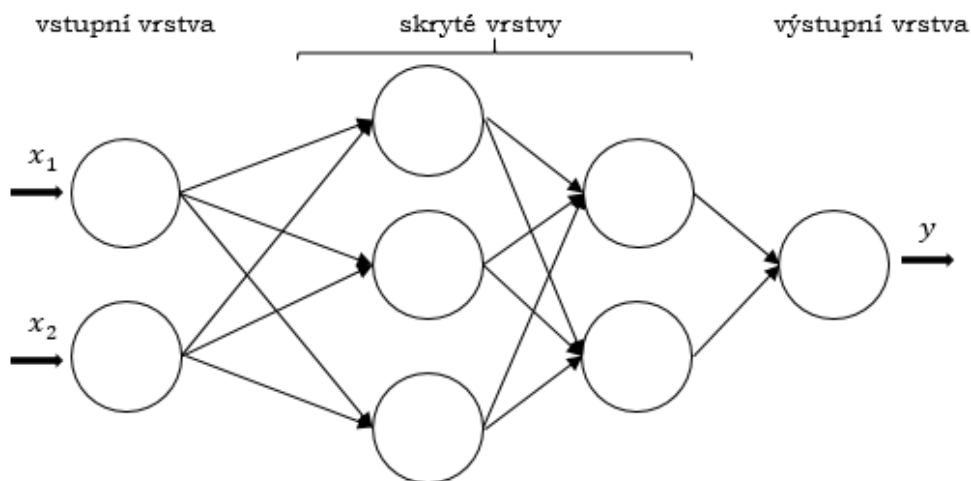
Skrytá vrstva

Skrytá vrstva je soubor neuronů, na které je aplikována aktivační funkce a je to mezivrstva mezi vstupní vrstvou a výstupní vrstvou. Úkolem skryté vrstvy je zpracovávat vstupy získané jeho předchozí vrstvou. Je to tedy vrstva, která je zodpovědná za extrahování požadovaných funkcí ze vstupních dat. Mnoho výzkumů bylo provedeno při vyhodnocování optimálního počtu neuronů ve skryté vrstvě, ale přesto žádný z nich nedokázal určit přesný počet. V neurální síti může být také několik skrytých vrstev. Musíme tedy u každého problému individuálně určit, kolik skrytých vrstev musí být použito. Předpokládejme, že pokud máme data, která lze oddělit lineárně, pak není nutné používat skrytou vrstvu, protože lze aktivační funkci implementovat do vstupní vrstvy, která může problém vyřešit. Ale v případě problémů, které

se zabývají složitými rozhodnutími, můžeme použít 3 až 5 skrytých vrstev na základě stupně složitosti problému nebo požadovaného stupně přesnosti. To však neznamená, že pokud budeme neustále zvyšovat počet vrstev, bude neurální síť poskytovat vyšší přesnost. Nezáleží pouze na počtu skrytých vrstev, ale také na počtu jednotlivých neuronů, ze kterých se neuronová síť skládá. Pokud je počet neuronů ve srovnání se složitostí problémových dat menší, pak ve skrytých vrstvách bude jen velmi málo neuronů, které by adekvátně detekovaly signály ve složité datové sadě.

Výstupní vrstva

Výstupní vrstva neuronové sítě shromažďuje a přenáší informace odpovídajícím způsobem tak, jak bylo navrženo. Vzor prezentovaný výstupní vrstvou lze přímo vysledovat zpět do vstupní vrstvy. Počet neuronů ve výstupní vrstvě by měl přímo souviset s typem práce, kterou neuronová síť prováděla.



Obrázek 4.2: Schéma vícevrstvé neuronové sítě

Zdroj: <http://po.licka.cz/>

4.2 Učení umělé neuronové sítě

Učení neuronové sítě znamená snahu o minimalizaci ztráty. Jinými slovy jde o snahu docílit toho, aby míra úspěšnosti řešení dané úlohy byla co největší. Toho lze docílit vhodnou kombinací parametrů. Trénování umělých

neuronových sítí se dělí na tzv. Trénování s učitelem a Trénování bez učitele. U trénování s učitelem neuronové sítě obecně provádějí kontrolované výukové úkoly, budují znalosti ze souborů dat, kde je předem poskytnuta správná odpověď. Sítě se pak učí laděním, aby samy našly správnou odpověď, čímž se zvýší přesnost jejich předpovědí. U trénování bez učitele neznáme požadovaný výstup, pouze síť zásobujeme daty a necháme jí, ať si sama vytvoří skupiny a příznaky.

4.2.1 Data augmentation

Data augmentation je technika používaná, pokud máme k dispozici pouze omezené množství dat. Vytváříme modifikované verze obrázku jeho transformací. Použití data augmentation často vede k lepším výsledkům než při použití velkého množství různorodých dat. V zásadě metody data augmentation zvyšují variabilitu dat, ale nepřidávají žádné nové informace do datasetu.[10]

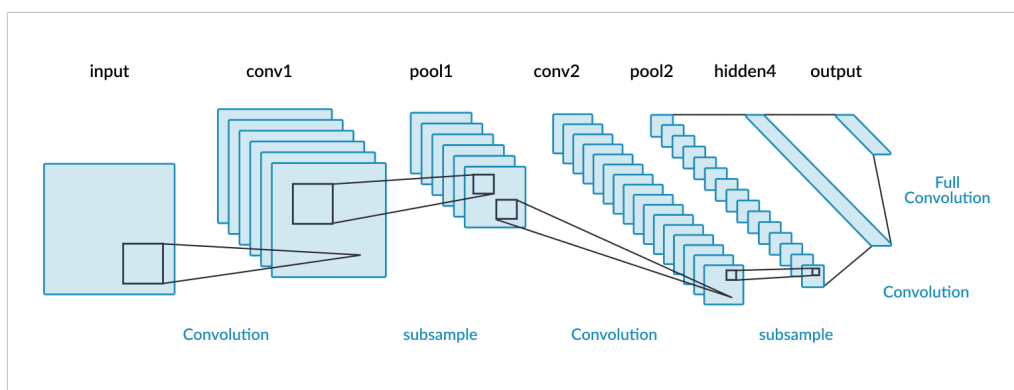
4.3 Konvoluční neuronová síť (CNN)

Konvoluční neuronová síť může mít desítky nebo stovky vrstev, přičemž každá z vrstev se učí detekovat různé vlastnosti obrazu. Filtry se aplikují na každý tréninkový obraz v různých rozlišeních a výstup každého konvolvovaného obrazu se použije jako vstup do další vrstvy. Filtry mohou začínat jako velmi jednoduché funkce, jako je jas a hrany.

Stejně jako tradiční neuronové sítě má i CNN neurony s váhami. Model se tyto hodnoty učí během tréninkového procesu a průběžně je aktualizuje s každým novým příkladem tréninku. V případě CNN jsou však váhy a hodnoty zkreslení stejné pro všechny skryté neurony v dané vrstvě.

Trénování větších a složitějších CNN je velmi hardwarově náročné. Místo procesorů se pro trénování CNN často používají grafické karty, které mají větší výpočetní výkon. K tomu, aby bylo možné trénovat pomocí grafických karet, je potřeba speciálních softwarových a hardwarových architektur. Mezi nejčastěji používané architektury patří například CUDA, jenž slouží ke spouštění programů přímo na grafických kartách.

Neurony v CNN jsou rozděleny do trojrozměrné struktury, přičemž každá sada neuronů analyzuje malou oblast nebo rys obrazu. Jinými slovy, každá skupina neuronů se specializuje na identifikaci jedné části obrazu. CNN používají předpovědi z vrstev k vytvoření konečného výstupu, který představuje vektor skóre pravděpodobnosti, jenž představuje pravděpodobnost, že konkrétní rys patří do určité třídy.



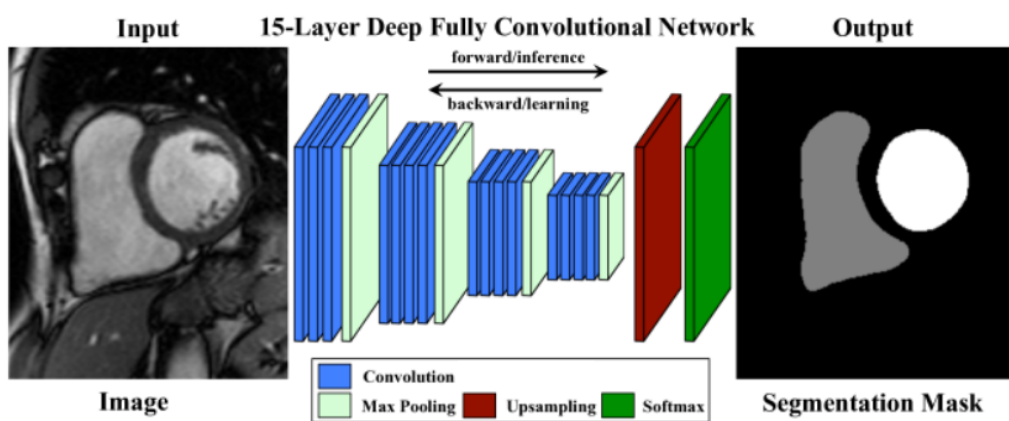
Obrázek 4.3: Schéma konvoluční neuronové sítě

Zdroj: <https://missinglink.ai/>

4.4 Plně propojená konvoluční neuronová síť (FCN)

Plně konvoluční síť využívá konvoluční neurální síť k transformaci obrazových pixelů na kategorie pixelů. Na rozdíl od dříve zavedených konvolučních neuronových sítí transformuje FCN výšku a šířku mapy funkcí mezi lehlé vrstvy zpět na velikost vstupního obrazu prostřednictvím transponované konvoluční vrstvy, takže předpovědi mají vzájemnou korespondenci se vstupním obrazem v prostorový rozměr (výška a šířka). Vzhledem k poloze na prostorové dimenzi bude výstupem kanálové dimenze predikce kategorie pixelu odpovídajícího umístění.

Plně konvoluční síť střídají konvoluční a nelineární operace, aby efektivně klasifikovaly všechny pixely vstupního obrazu v jednom průchodu vpřed. FCN obsahují max-pooling (viz. 5.6.1) a up-sampling. Neobsahují plně propojené vrstvy.

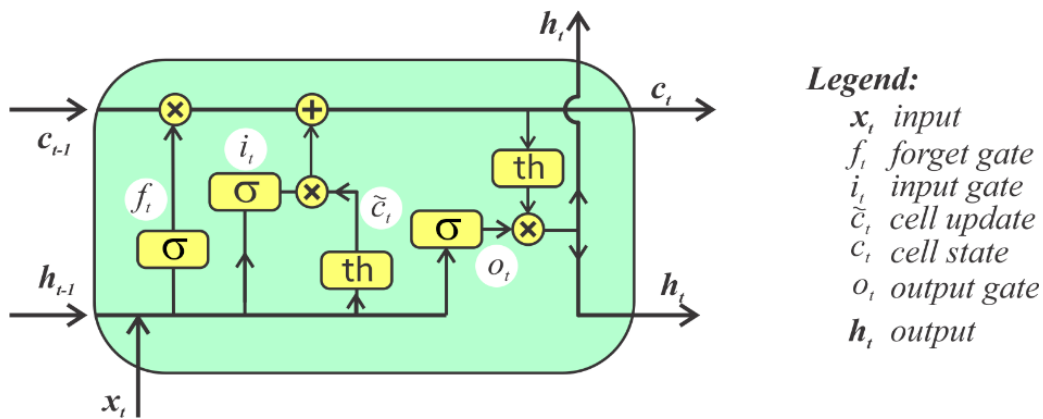


Obrázek 4.4: Schéma patnácti vrstvé FCN

Zdroj: [9]

4.5 Long Short Term Memory (LSTM)

Jedná se o model nebo architekturu, která rozšiřuje paměť rekurentních neuronových sítí. Rekurentní neuronové sítě mají obvykle „krátkodobou paměť“ v tom, že používají trvalé předchozí informace, které mají být použity v aktuální neuronové síti. LSTM zavádí dlouhodobou paměť do rekurentních neuronových sítí. Zmírňuje problém mizejícího přechodu, což je místo, kde se neuronová síť přestane učit, protože aktualizace různých vah v dané neuronové síti se zmenšují. Dělá to pomocí řady „bran“. Ty jsou obsaženy v paměťových blocích, které jsou spojeny vrstvami.



Obrázek 4.5: Schéma LSTM

Zdroj: <https://www.rs-online.com/>

5 Použité technologie

5.1 Python

Python patří mezi nejsnadněji osvojitelné programovací jazyky. Je velmi expresivní a multiplatformní, lze tedy bez problému spustit program napsaný ve Windows i na unixových systémech jako je Linux, Mac OS X. Opravdu silnou stránkou jazyku Python je, že se dodává s již kompletní velmi rozsáhlou standardní knihovnou. V jazyku Python lze programovat v procedurálním, objektivě orientovaném a v menší míře též funkcionálním stylu.[8]

5.2 OpenCV

OpenCV (Open Source Computer Vision Library) je knihovna pro různé platformy, pomocí které lze vyvíjet aplikace počítačového vidění v reálném čase. Zaměřuje se hlavně na zpracování obrazu, snímání a analýzu videa, včetně funkcí, jako je detekce obličeje a detekce objektů.

5.3 NumPy

Numpy je základní knihovna pro vědecké výpočty v Pythonu. Poskytuje vysoce výkonný multidimenzionální objekt pole a nástroje pro práci s těmito poli. Má také funkce pro práci v doméně lineární algebry, Fourierovy transformace a matic.

5.4 Tensorflow

Tensorflow je systém strojového učení, který používá dataflow grafy k reprezentaci jeho výpočtu. Nabízí vývojářům možnost experimentování s trénovacími algoritmy. Tensorflow podporuje velké spektrum aplikací s důrazem na trénování a důsledky hlubokých neuronových sítí. [1]

5.5 Inch Evaluator

Součástí dodaných dat byl program `inch_evaluator.py`, který slouží k finálnímu hodnocení složené predikované mapy.

5.6 Keras

Keras je rozhraní API neuronových sítí na vysoké úrovni vyvinuté se zaměřením na umožnění rychlých experimentů.

5.6.1 Vrstvy Keras

Vrstvy Keras jsou primárním stavebním kamenem modelů Keras. Každá vrstva přijímá vstupní informace, provádí výpočty a nakonec vydává transformované informace. Výstup jedné vrstvy bude proudit do další vrstvy jako její vstup. Keras je založen na minimální struktuře, která poskytuje čistý a snadný způsob vytváření modelů hlubokého učení založených na TensorFlow nebo Theano. Keras je navržen tak, aby rychle definoval modely hlubokého učení.

Conv2D

Conv2D je dvourozměrná konvoluční vrstva. Tato vrstva vytváří konvoluční jádro, které je spojeno se vstupem vrstvy a vytváří tenzor výstupů. Pomocí Conv2D dochází ke zmenšení počtu parametrů, které se mají natrénovat.

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernel

0	-1	0
-1	5	-1
0	-1	0

114	328	-26	470	158
53	266	-61	-30	344
403	116	-47	295	244
108	-135	256	-128	344
314	346	279	153	421

Obrázek 5.1: Conv2D s jádrem 3×3 .

Zdroj: <https://www.pyimagesearch.com/>

Conv2DTranspose

Vrstva Conv2DTranspose slouží k zvětšení rozměru tensorů. Z redukovaného stavu vytvoří pomocí jádra tensor zvětšený.

MaxPooling2D

Max pooling je proces diskretizace na základě vzorku. Cílem je odebrat vzorek vstupní reprezentace (obrázek, výstupní matice se skrytou vrstvou atd.), snížit jeho rozměrnost a umožnit předpoklady o vlastnostech obsažených v binovaných podoblastech. MaxPooling vybere vždy tu nejmenší hodnotu v menší oblasti (matici). Máme-li definovanou velikost poolu (např. 2x2), vstupní tensor zapíše největší hodnotu dané oblasti do tensoru výstupního. Tato vrstva tedy také redukuje rozměr tensoru.

Image Matrix				Max Pool	
2	1	3	1	2	4
1	0	1	4	7	9
0	6	9	5		
7	1	4	1		

Obrázek 5.2: MaxPooling s velikostí oblasti 2×2 .

Zdroj: <https://austingwalters.com/>

6 Data a jejich zpracování

6.1 Historické katastrální mapy

Katastrální mapa (zkráceně katastr) je technický výraz pro řadu záznamů, které ukazují rozsah, hodnotu a vlastnictví (nebo jiný základ pro užívání nebo obsazenost) půdy. Přesněji řečeno, katastr je záznam ploch a hodnot pozemků a vlastníků půdy, který byl původně sestaven pro účely zdanění. V mnoha zemích však již neexistuje žádná daň z pozemků a v praxi katastr slouží dvěma dalším důležitým účelům. Poskytuje pohotový prostředek k přesnému popisu a identifikaci jednotlivých pozemků a slouží jako průběžná evidence práv k pozemkům. Z tohoto důvodu je celkem zřejmé, že je vhodné tyto záznamy vést a uschovávat je. Díky tomu se dodnes zachovalo mnoho historických katastrálních map. Moderní katastr obvykle sestává z řady rozsáhlých map nebo plánů a odpovídajících registrů.

6.2 Původní data

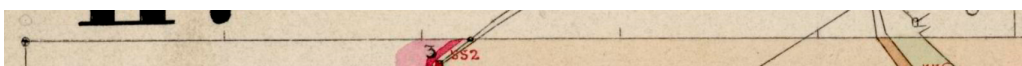
Jako vstupní data slouží dodané německy psané, ručně kreslené, zdigitalizované mapové podklady z poloviny 19. století. Jedná se o jednotlivé historické mapové listy, které mimo jiné obsahují palcové a rohové značky. Palcové a rohové značky nemusí být přesné a také mohou být překresleny jinými mapovými atributy. Celý mapový list také může být lehce zrotován do určité strany, je tedy následně vhodné jej zrotovat do vodorovné polohy. Jelikož se jedná o historické mapové podklady, barevné nepřesnosti nejsou výjimkou (jedná se o starý papír nejednotné barvy).

Každá strana mapového listu obsahuje jednu stranu mapového rámu. Tu představuje rovná čára omezená dvěma rohy, které jsou označeny rohovými značkami. Na jednotlivých stranách se nacházející palcové značky. Jejich množství se odvíjí od délky určité strany. Jednotlivé palcové značky lze rozdělit do dvou kategorií (palcové značky - jedná se o krátké čáry kolmé na hranici mapy, každá pátá palcová značka - jedná se o delší čáry kolmé na hranici mapy (tzv. pětipalce)). V datech se také mohou vyskytovat mapové listy, kterým chybí jedna strana mapového rámu, u takových se tedy použijí pouze zbylé strany obsahující palcové značky.



Obrázek 6.1: Ukázka jednoho mapového listu

Zdroj: KIV

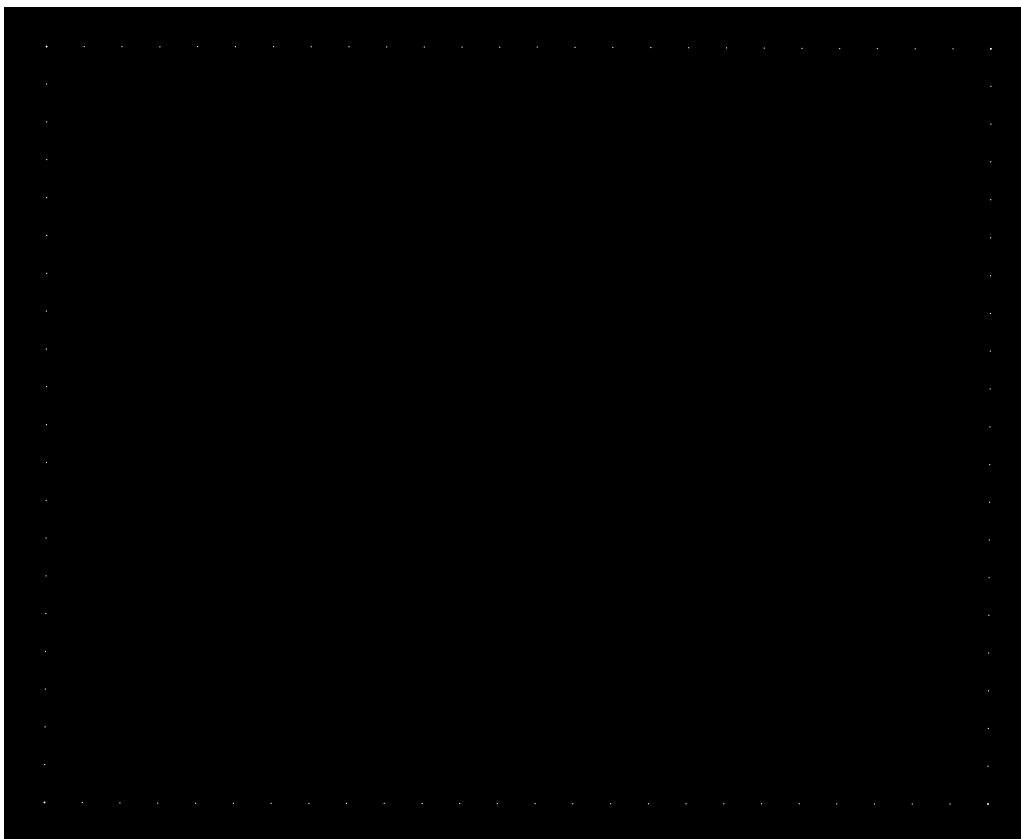


Obrázek 6.2: Přiblížení na palcové značky

6.3 Ground-truth data

Poskytnutá datová sada neobsahuje ground-truth data pro všechny mapové listy. Pro trénování neuronové sítě potřebujeme z původních mapových listů extrahovat pouze palcové a rohové značky, žádné jiné mapové atributy pro nás nejsou důležité. Toho bylo docíleno tak, že byl vytvořen černý obrázek shodné velikosti jako mapový list a byly do něj vloženy bílé kruhy reprezentující jednotlivé palcové a rohové značky. Snaha byla o co nejpřesnější umístění nových bílých značek oproti jejich vzoru v původním mapovém listu, kruhy byly kresleny tak, aby jejich střed ležel na pozici palcové značky. Vkládání bílých značek do nového černého obrázku bylo manuální, mohlo proto dojít

k malým odchylkám oproti původním umístěním jednotlivých značek (tyto odchylky se však pohybují v jednotkách pixelů). Výstupem tohoto procesu je tedy nový černobílý obrázek (maska) představující pouze palcové a rohové značky (viz. Obrázek 6.3), ty jsou na tomto obrázku reprezentovány jako bílé kruhy vždy o stejné velikosti a to rohové o velikosti 15px a palcové o velikosti 10px.



Obrázek 6.3: Ukázka anotovaného mapového podkladu



Obrázek 6.4: Přiblížení anotovaných palcových značek a jedné rohové

6.4 Datasetsy

Výsledné výřezy z původních a z anotovaných dat je nutné rozdělit do datasetů, které budou sloužit k učení a testování neuronové sítě. Data je nutno rozdělit na tři kategorie: trénovací, validační a testovací data. Přičemž je nutné, aby se žádná jednotlivá data neobjevila ve více z těchto kategorií. Pokud by k tomu totiž došlo, mohlo by to výrazně ovlivnit schopnost sítě predikovat výsledná data. Nemělo by se tedy stát, že model validujeme na nějakých datech a tato data budou následně použita k testování modelu. Původních a anotovaných (masky) obrázků je 44, celkem tedy 88 s tím, že původní data jsou ve formátu JPG a masky jsou ve formátu PNG. Rozdělení dat do množin je znázorněno v tabulce 6.1

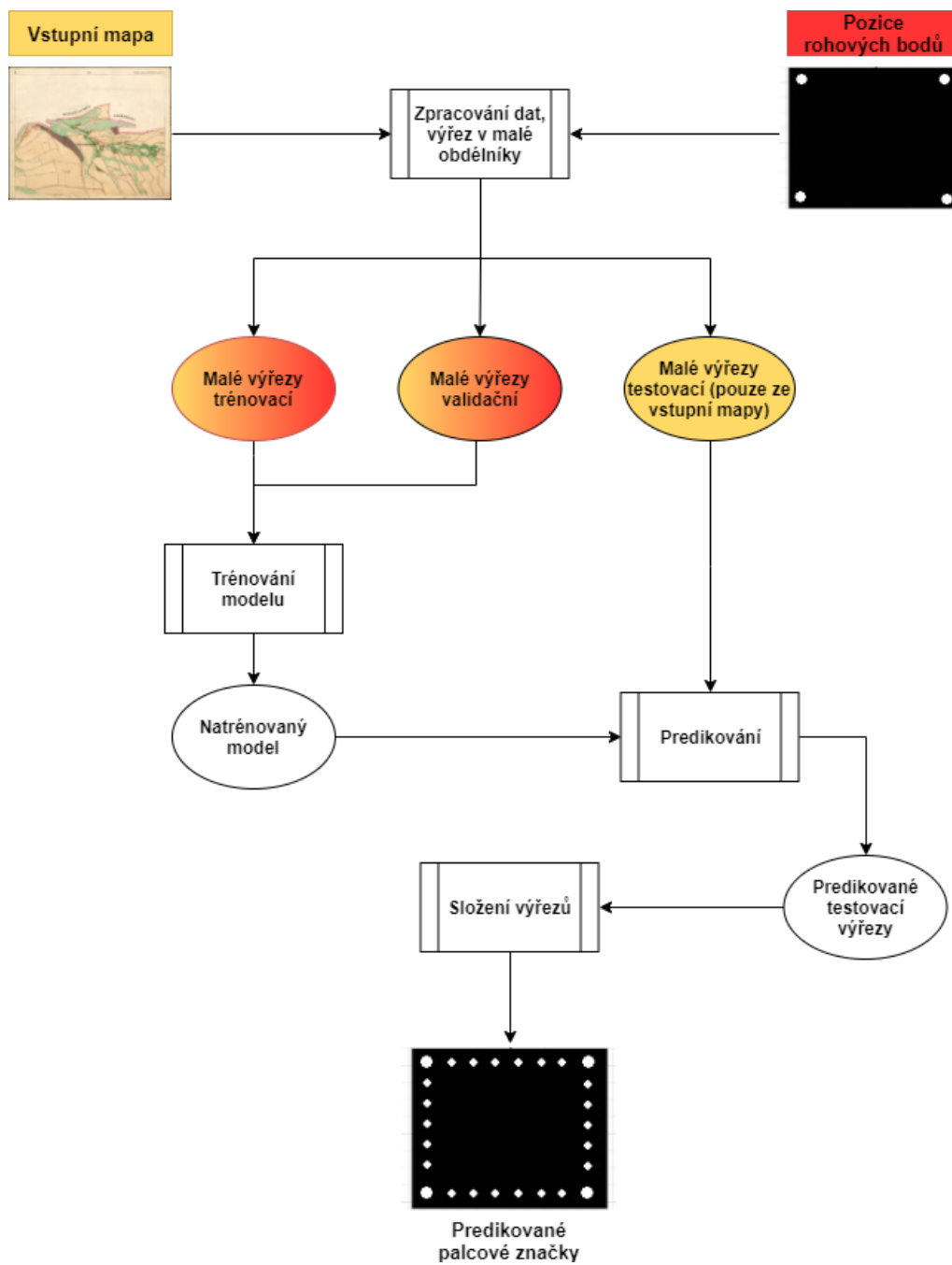
Tabulka rozdělení datasetů			
Kategorie dat	Počet původních	Počet anotovaných	Celkem
Trénovací data	26	26	52
Validační data	6	6	12
Testovací data	12	12	24

Tabulka 6.1: Rozdělení datasetů

Jelikož se jedná o značné množství obrázků v celkem dobrém rozlišení (především původní mapové podklady), výsledné datasety zabírají několik gigabytů. Je následně tedy důležité při načítání a zpracování dat a jejich použití, ošetřit přeplnění operační paměti a předejít tak chybám. Takto velké množství dat má samozřejmě také vliv na rychlost trénování neuronové sítě. Čím větší je vstupní množina dat, tím déle se může neuronová síť trénovat, avšak na druhou stranu větší množství dat může zaručovat vyšší efektivitu tréninku a validace.

7 Detekce palcových značek

K trénování neuronové sítě slouží ořezané obrázky reprezentující jednotlivé strany mapového rámu daných mapových listů (viz. 7.3), ty jsou pro nás oblastí zájmu, jelikož obsahují palcové značky. Při zpracování jednotlivých mapových listů, předpokládáme, že už jsou z předchozího kroku zjištěny pozice rohových značek. Zjistíme tedy, kde se tyto rohy nacházejí a kolik jich je. Pokud jich je méně než čtyři, tak danému mapovému listu chybí jedna hranice (jedná se o výřez z většího mapového listu). V dodaných mapových podkladech se nevyskytují mapové listy, kterým by chybělo více hranic než jedna. Známe-li tedy umístění jednotlivých rohových značek, je následně jednoduché určit oblast, ve které se hranice s palcovými značkami nachází. Může se stát, že je celý obrázek zrotován a mohlo by tak dojít k tomu, že se provede výřez, ve kterém by palcové značky postupně z výřezu unikaly. Tomu může být zamezeno například zrotováním celého obrázku do vodorovné polohy. Abychom tomuto předešli, tak zvolíme dva rohové body, které spojuje společná hranice a vypočteme jejich průměrnou y -ovou souřadnici, jsou-li vedle sebe a nebo x -ovou souřadnici, jsou-li pod sebou. Při následném správném zvolení výšky výřezu dosáhneme toho, že ve výsledném výřezu celé hranice budou všechny palcové i rohové značky a po finální predikci malých výřezů a jejich zpětném složení nebudeme muset obrázek složitě rotovat. Na obrázku 7.1 můžeme vidět schéma použitého postupu.

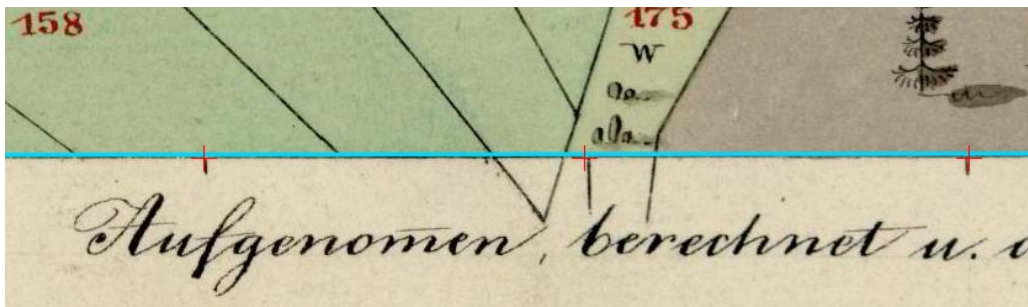


Obrázek 7.1: Diagram procesu zpracování dat, učení modelu a následné predikce

V následujících dvou podkapitolách je popis dvou použitých metod detekce palcových značek a to původní dodané metody a nově navrženého algoritmu, který využívá plně konvoluční síť. Ve třetí podkapitole je popsán proces zpracování dat, se kterými bude následně model trénován.

7.1 Stávající metoda detekce palcových značek

Stávající metoda detekce palcových značek je založená na klasických technikách zpracování obrazu. Využívá prahování a Houghovu transformaci pro nalezení mapového rámu. Následně jsou vytvořeny výřezy podél jednotlivých stran mapového rámu a v nich se opět pomocí prahování hledají čáry kolmé na hranici mapového rámu. Metoda je poměrně citlivá na falešné detekce. Nenalezené (nevýrazné) palcové značky se dopočítávají na základě známé přibližné vzdálenosti mezi značkami.



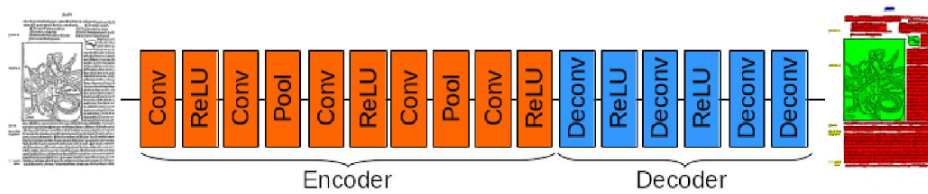
Obrázek 7.2: Nepřesná detekce stávající metody

Na obrázku 7.2 je patrná nepřesná detekce palcových značek, ke které došlo při využití stávající metody. Tyto nepřesnosti jsou významné a cílem nové metody je podobné nepřesnosti eliminovat.

7.2 Architektura navrhované metody implementující FCN

Pro trénování a vyhodnocení algoritmu segmentace jsou vyžadována manuálně anotovaná ground-truth. V těchto základních ground-truth maskách je oblast obvykle označena jako celý blok, který obsahuje pixely na popředí a na pozadí. Trénování modelu vyžaduje původní obraz a očekávaný výstup segmentace, což je ground-truth.

Použitá architektura [11] je zobrazena na obrázku 7.3. Skládá se ze struktury enkodéru a dekodéru založené na vrstvách konvoluce a dekonvoluce. Na vrstvy dekonvoluce lze v zásadě pohlížet jako na inverzní konvoluci. Nastavením kroku rovné velikosti jádra je možné jej použít ke zvýšení rozlišení obrazu o tento faktor. Navrhovaná síť se skládá ze sdružování vrstev s velikostí jádra a krokem 2, což je důvod, proč dvě střední dekonvoluční vrstvy mají také velikost jádra a krok 2, aby se zvětšily rozměry obrazu v souladu. Enkodér se skládá z více konvolučních vrstev než dekodér, protože kodér musí zpracovávat podrobnější informace. Každá jednotlivá konvoluční vrstva má velikost jádra 5 a pro udržení rozlišení obrazu používá odsazení na okraji.[11]



Obrázek 7.3: Proces plně konvoluční neruonové sítě

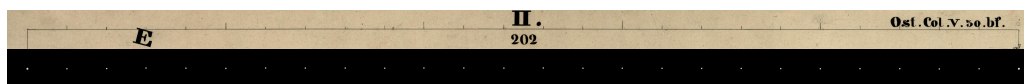
Zdroj: [11]

7.3 Proces zpracování dat

Proces zpracování dat se skládá ze dvou kroků. Nejdříve se vyříznou celé hranice obsahující palcové a rohové značky. Následně se segmentují na menší výřezy, se kterými následně bude model trénován.

7.3.1 Výřez celé hranice mapy

Nejdříve se provede výřez hranice jednoho mapového listu. Jedná se o výřez okolo čáry představující hranici na dané straně. Výřez je o velikosti 150px nad a pod hraniční čarou a 150px vlevo od prvního rohového bodu a 150px vpravo od posledního rohového bodu. Tento výřez je proveden jak pro původní mapový podklad, tak i pro anotovaný mapový podklad reprezentující masku.

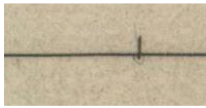


Obrázek 7.4: Porovnání původního a anotovaného ořezu jedné hranice mapového rámu

7.3.2 Výřez vzorků hranice mapy

Po ořezání a otočení všech stran daného mapového podkladu je nutné daný výřez rozdělit na menší vzorky, které lze poté využít jako vstupní data pro trénování modelu neuronové sítě. Jinými slovy dochází k cyklickému ořezávání dlouhého ořezu celé mapové hranice (viz. obrázek 7.4) na menší obrázky, které mají vždy stejnou výšku (100 pixelů), ale jejich délka je vždy větší (200, 300 nebo 400 pixelů - viz. obrázek 7.5), ale pokaždé je stejná pro všechny menší ořezy. Výstupem jsou tedy obdélníky. Jednotlivé malé obdélníky se překrývají se sousedními vždy o deset procent své plochy, kromě posledního, u kterého se může překryv lišit na základě délky celého výřezu. Kvůli takto nastaveným rozměrům se může stát, že se řez malého výřezu provede zrovna v místě, kde se nachází palcová značka a protne a rozdělí jí tak, že se v malém výřezu bude nacházet pouze její část. To však ničemu nevádí, jelikož i takové výřezy dobře poslouží modelu k jeho správnému trénování.

100x200



100x200



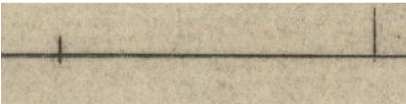
100x300



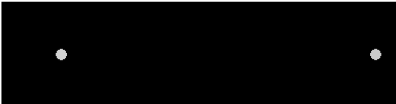
100x300



100x400



100x400



Obrázek 7.5: Původní a ground-truth výřezy vzorků (o všech třech velikostech)

8 Experimenty

Hlavním cílem experimentů je porovnat konzervativní metodu nepoužívající neuronové sítě s novou metodou, která je využívá. Konkrétně využívá již dříve zmíněnou plně konvoluční neuronovou síť. Výstupem experimentů by měl být fakt, zda-li je využití neuronových sítí pro danou problematiku přínosem, či nikoliv. Nejdříve potřebujeme vybrat model, který by se nejlépe naučil rozeznávat a následně predikovat jednotlivé palcové značky. K výběru nám slouží klíčová metrika IoU foreground. Foreground z důvodu, že na popředí se nachází pouze pár pro nás důležitých informací. Obvykle se také průměruje *bg*(Background), avšak v našich obrázcích se vyskytuje velké množství informací na pozadí a popředí má proto pro nás větší vypovídající hodnotu. Snažíme se zároveň vybrat vhodné parametry, se kterými je model realizován. Mezi tyto parametry patří například velikost výřezu jednotlivých prvků v trénovacím, testovacím a ve validačním datasetu, na kterých se model učí a testuje. Dalším důležitým parametrem je počet epoch. Počet epoch je hyperparametr, který definuje, kolikrát bude učící algoritmus pracovat v celé datové sadě trénování. Jedna epocha znamená, že každý vzorek v datové sadě trénování měl příležitost aktualizovat parametry interního modelu. Dále zkoumáme využití metody Early Stopping (viz. 8.3).

8.1 Přístupy k vyhodnocení výsledků

Podle [3] se v současné době pro vyhodnocení úspěšnosti používají dva přístupy: blokově založené a pixelově založené. Každý přístup se liší použitými metrikami, požadavky na vstup a výpočetní náročností. K vyhodnocení úspěšnosti našich dat nám však budou stačit pouze pixelově založené přístupy.

Naš přístup k vyhodnocení výsledků spočívá v porovnání složených obrázků z predikovaných výřezů a jejich ground-truth ekvivalentech. Snaha je samozřejmě o co nejpřesnější shodu mezi těmito dvěma obrázky.

8.2 Pixelové metody

V pixelově založeném přístupu pro vyhodnocení je každý pixel obrázku označen třídami, které reprezentuje. Vstupem metrik jsou tedy dva obrázky (ground-truth a výsledek segmentace) s metadaty obsahujícími informace,

jakých jednotlivých klasifikačních tříd pixel nabývá. Ground-truth obrázek je při vyhodnocení reprezentován jako:

$$L = (\{c \in C\}_i | 1 \leq i < n)$$

C představuje množinu unikátních tříd v dokumentu a n je celkový počet pixelů v obrázku. Stejným způsobem je reprezentován i obrázek získaný segmentací, označme ho jako L' . Během vyhodnocení je tedy porovnáváno přiřazení tříd k pixelu v ground-truth obrázku vůči přiřazení v segmentaci, pixel po pixelu.[5] Při vyhodnocení mohou nastat následující případy:

- True Positive (TP) - pixel v L a v L' má stejnou třídu c_i
- False Positive (FP) - třída c_i byla přiřazena pouze pixelu v L'
- True Negative (TN) - třída c_i nebyla přiřazena ani jednomu pixelu (v L a v L')
- False Negative (FN) - třída c_i nebyla přiřazena pixelu v L' , ale byla přiřazena pixelu v L

8.2.1 Metriky

Precision

Precision (přesnost) je pravděpodobnost, že náhodnému pixelu v obrázku po predikci je správně přiřazena třída c .

Recall

Recall (úplnost) je pravděpodobnost, že se náhodný pixel z ground-truth s třídou c bude vyskytovat i v obrázku získaném predikcí.

MAE


Mean Absolute Error (Střední absolutní chyba) modelu označuje průměr absolutních hodnot každé chyby predikce ve všech instancích testovací sady dat. Predikční chyba je rozdíl mezi skutečnou pozicí palcové značky a predikovanou hodnotou pro danou instanci.

$$mae = \frac{\sum_{i=1}^n abs(y_i - \lambda(x_i))}{n}$$

Hlavním přínosem implementování neuronové sítě by mělo být právě zlepšení v této metrice.

Intersection over Union (IoU)

Intersection over Union (průnik ku sjedocení) je nejpoblárnější metrika hodnocení používaná v testovacích kritériích detekce objektů. Mezi optimalizací běžně používaných ztrát vzdálenosti pro regresi parametrů ohraničovacího rámečku a maximalizací této metrické hodnoty však existuje rozdíl. Optimálním cílem pro metriku je metrika samotná. V případě osově vyrovnaných 2D ohraničovacích polí lze ukázat, že IoU lze přímo použít jako regresní ztrátu.[6] Zkoumáme Jaccarduv koeficient podobnosti (J).

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Obrázek 8.1: Intersection over Union

Zdroj:[7]

Vzorec:

$$J(A, B) = \frac{A \cap B}{A \cup B}$$

Abychom mohli použít Intersection over Union k vyhodnocení (libovolného) detektoru objektu, potřebujeme:

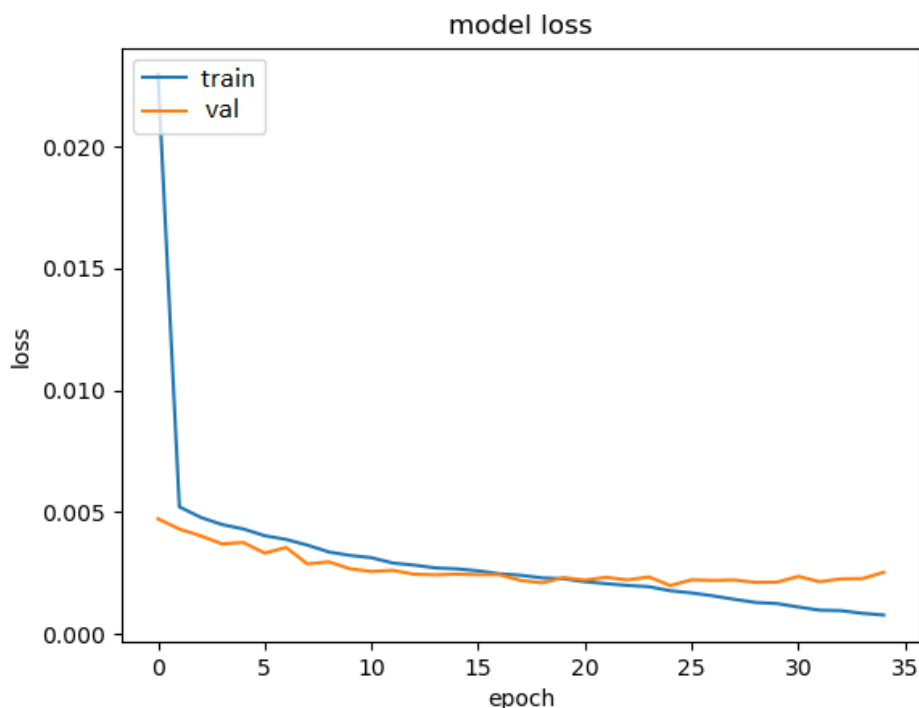
- Hraniční ground-truth ohraničené rámečky z testovací sady, které určují, kde je v obrázku náš objekt.

- Sadu předpovězených ohraničujících rámečků vyhodnocených naším modelem.

8.3 Učení

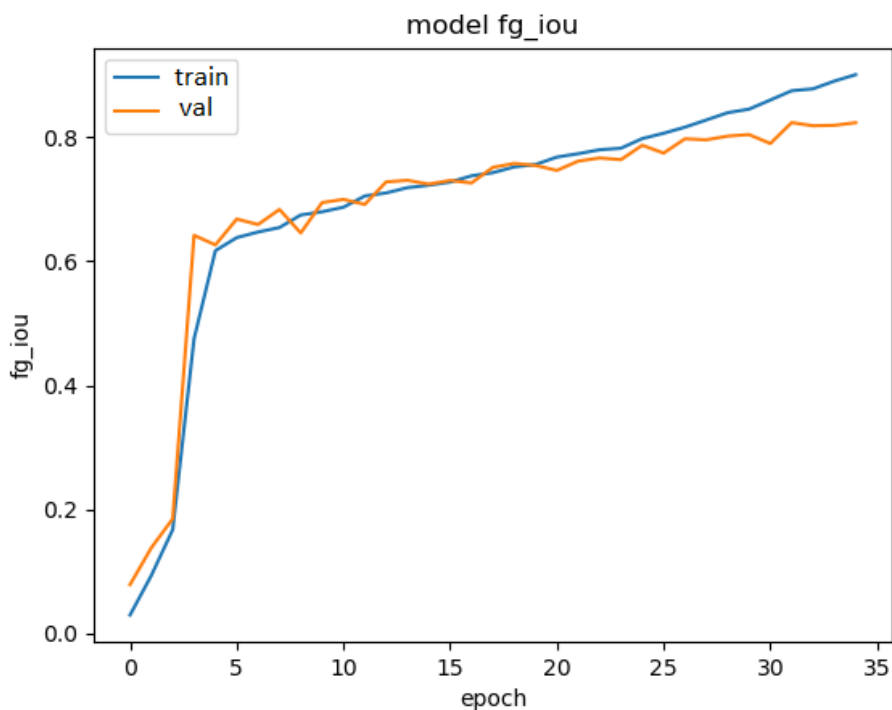
Nejdříve musíme pomocí trénovacích a validačních dat zajistit co největší přesnost následných predikcí sítě. Jinými slovy musíme model co nejlépe naučit. To, jak kvalitně se nám to povede, můžeme ovlivnit množstvím použitých dat a následnými parametry učení modelu. Půjde nám tedy o výběr vhodných hyperparametrů metody. Poté, co se model s danými parametry naučí, srovnáme výsledky učení s výsledky dosaženými s použitím jiné kombinace parametrů. Následně vybereme ten nejlepší model (viz. tabulka 8.1), který poté využijeme pro predikování dat na testovacích datech.

Z průběhu učení modelu lze vygenerovat grafy, které ukazují vývoj jednotlivých metrik v čase (přibývajících epochách). Z těchto grafů jsme schopni určit, zda zadané parametry byly vhodně zvolené a nebo zda by bylo vhodné některý z nich změnit. Pomocí těchto grafů jsme schopni posoudit, jak probíhá dané učení modelu.



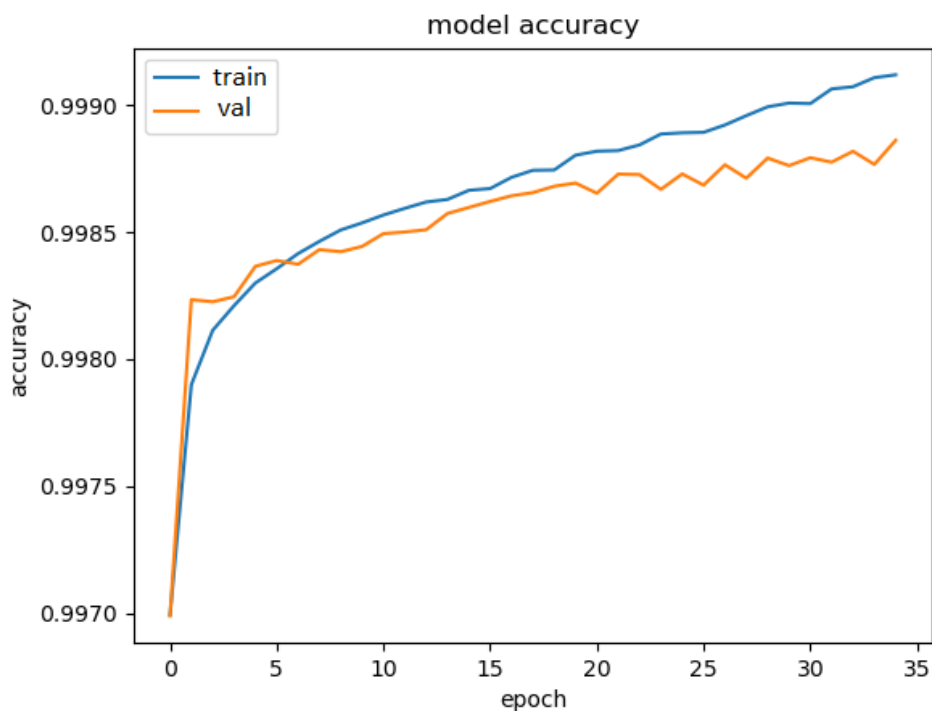
Obrázek 8.2: Graf průběhu trénování modelu s velikostí výřezu 100x300

Z grafu ztráty (Obrázek 8.2) vidíme, že validační ztráta přestává klesat mezi 20. a 25. epochou. Růst validační ztráty značí možné přetrénování modelu a v bodě, kdy dojde k nárůstu, je vhodné trénování zastavit.



Obrázek 8.3: Graf průběhu trénování modelu o velikosti výřezu 100x300 a 35 epochách znázorňující fg_iou

Na grafu představujícím průběh trénování modelu znázorňující fg_iou (Obrázek 8.3) lze vidět, že se hodnota fg_iou u trénovací sady dat stále zlepšuje, ale u validační datové sady v posledních čtyřech epochách roste jen velmi málo oproti trénovací sadě. Lze tedy usoudit, že zvolené množství epoch je dostačující, tomu také odpovídá srovnání s jinými počty epoch (viz. Tabulka výsledného IoU během učení sítě), kde má toto množství epoch nejlepší výsledek.

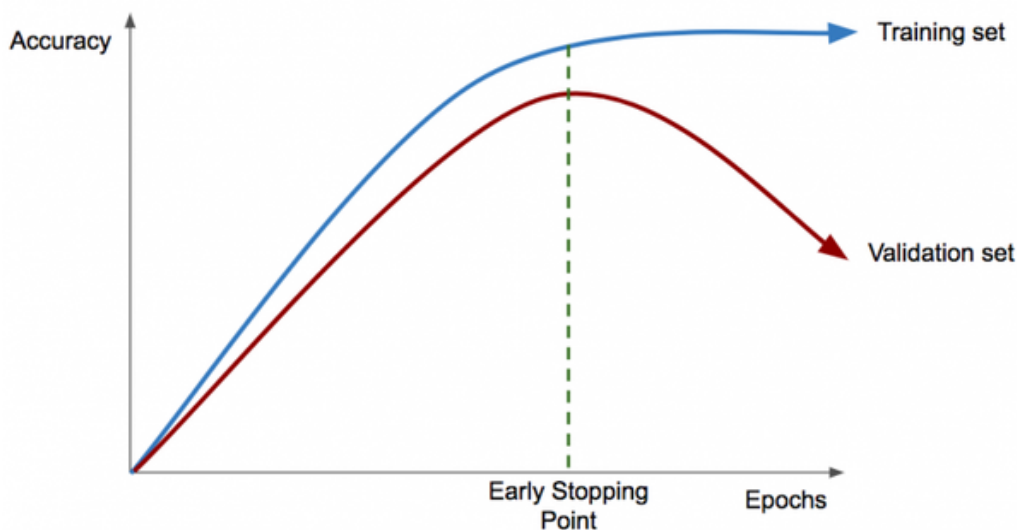


Obrázek 8.4: Graf průběhu trénování modelu o velikosti výřezu 100x300 a 35 epochách znázorňující přesnost

Z grafu přesnosti (Obrázek 8.4) vidíme, že model by pravděpodobně mohl být trénován o trochu více, protože trend přesnosti u obou datových sad v posledních několika epochách stále stoupá. Můžeme také vidět, že model se dobře naučil jak tréninkovou datovou sadu tak i validační, což ukazuje srovnatelné vlastnosti obou sad.

Early Stopping

Problém s trénováním neuronových sítí spočívá ve výběru počtu epoch, které se mají použít. Příliš mnoho epoch může vést k přetrénování sítě na trénovací sadě a tím ke snížení schopnosti generalizace na neviděných datech. Příliš málo může mít za následek nedostatečné natrénování modelu. Jednoduše řečeno, model se může naučit jen velmi málo a nebo se může učit zbytečně moc, obojí má za následek neefektivní následné predikování modelu. Early Stopping je metoda, která umožní určit vhodný počet tréninkových epoch a ukončit trénink, jakmile se výkon modelu přestane zlepšovat. To, zda se výkon stále zlepšuje, monitorujeme pomocí validační ztrátové funkce (Validation loss). Validační ztráta je stejná metrika jako ztrátová funkce při trénování, ale nepoužívá se k aktualizaci vah. Vypočítává se stejným způsobem - spuštěním sítě vpřed po vstupech x_i a porovnáním síťových výstupů y^i se základními hodnotami y_i pomocí ztrátové funkce.



Obrázek 8.5: Graf znázorňující využití metody Early Stopping

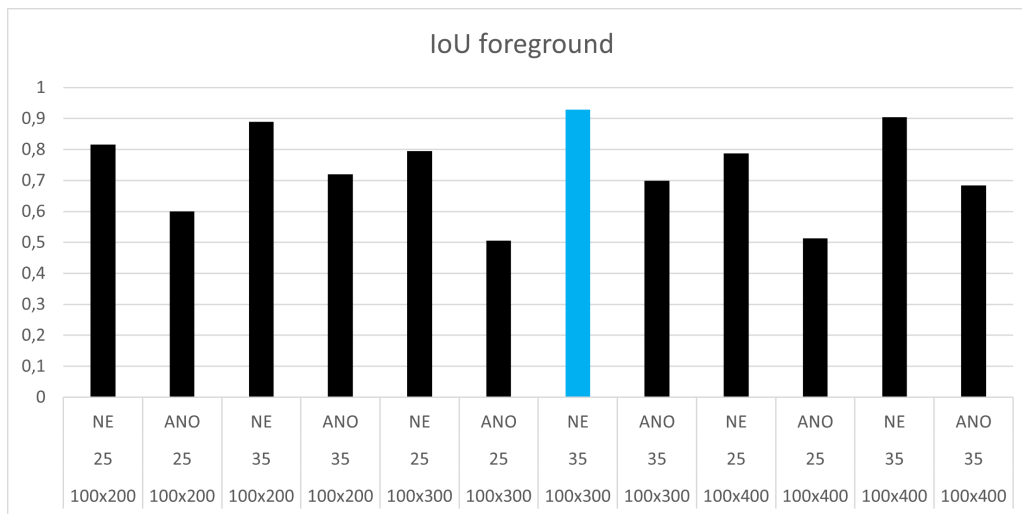
Zdroj: <https://www.quora.com/>

Výsledky učení modelu

Na základě předchozích experimentů byly zvoleny počty epoch 25 a 35 a bylo provedeno srovnání s metodou Early Stopping. Tabulka 8.1 a obrázek 8.6 obsahují výsledky učení modelu na jednotlivých sadách výřezů.

Tabulka výsledného IoU během učení sítě			
Velikost výřezu	Počet epoch	Early Stopping	IoU foreground
100x200	25	NE	0,8159
100x200	25	ANO	0,5995
100x200	35	NE	0,8890
100x200	35	ANO	0,7199
100x300	25	NE	0,7956
100x300	25	ANO	0,5052
100x300	35	NE	0,9288
100x300	35	ANO	0,6984
100x400	25	NE	0,7871
100x400	25	ANO	0,5133
100x400	35	NE	0,9042
100x400	35	ANO	0,6846

Tabulka 8.1: Srovnání jednotlivých parametrů použitých během trénování modelu a jejich výsledků



Obrázek 8.6: Graf srovnání jednotlivých pokusů

Na základě výsledků lze tvrdit, že se model vždy naučil lépe s větším množstvím epoch a bez aplikování Early Stoppingu. Z tabulky výsledků vyplývá to, že se model na základě metriky IoU pro foreground nejlépe naučil při velikosti výřezu 100x300, 35 epochách a bez Early Stoppingu. Tento model má velice dobré výsledky.

Z tabulky srovnání jednotlivých parametrů použitých během trénování modelu a jejich výsledků a z grafů průběhu učení vybraného modelu lze konstatovat, že vybrané parametry pro trénink modelu byly vhodně zvoleny a naučený model je tedy připravený k predikování výsledných dat na testovací datové sadě.

8.4 Predikování

Naučený model můžeme využít k predikování jednotlivých výřezů na testovací sadě. Výsledné predikované výřezy je následně potřeba poskládat zpět na jejich původní pozici v daném mapovém podkladu. Je nutné, aby toto zpětné složení bylo přesné a nestalo se, že by predikovaný výřez byl vložen na špatnou pozici. Došlo by tím k narušení vzoru ground-truth a následné metody pro vyhodnocení úspěšnosti by takto špatně složené obrázky vyhodnotily velmi špatně. Náš model predikuje palcové a rohové značky celkem přesně (odchylky se pohybují maximálně v jednotkách pixelů). Může se však stát, že model předpoví palcovou nebo rohovou značku i pro místo, na kterém se ve skutečnosti nenachází (viz. obrázek 8.7). Stává se tak z důvodu, že na hranicích mapových listů se mohou objevit i jiné atributy. Nejčastěji se jedná o různé fyzickogeografické, socioekonomické a doplňkové prvky (komunikace, nápisy, hranice oblastí a podobně). Následný složený obrázek z predikovaných výřezů tak může obsahovat více značek než ground-truth obrázek. Pokud bychom následně chtěli například skládat tyto obrázky pomocí palcových značek, je nutno na toto myslet a ošetřit to. Například faktem, že v mapových listech se jednotlivé palcové značky nacházejí vždy v podobných vzdálenostech (okolo 310 pixelů). Také se samozřejmě může stát, že model nějakou značku nepředpoví, na dodaných datech k tomu však nedocházelo ve větší míře. Stávalo se ale, že jednotlivé bílé body reprezentující jednotlivé značky nebyly ve stejných velikostech jako v ground-truth datech (body byly menší než v ručně anotovaných datech).



Obrázek 8.7: Ukázka predikce palcové značky pro jeden výřez



Obrázek 8.8: Predikce validní a nevalidní palcové značky pro jeden výřez

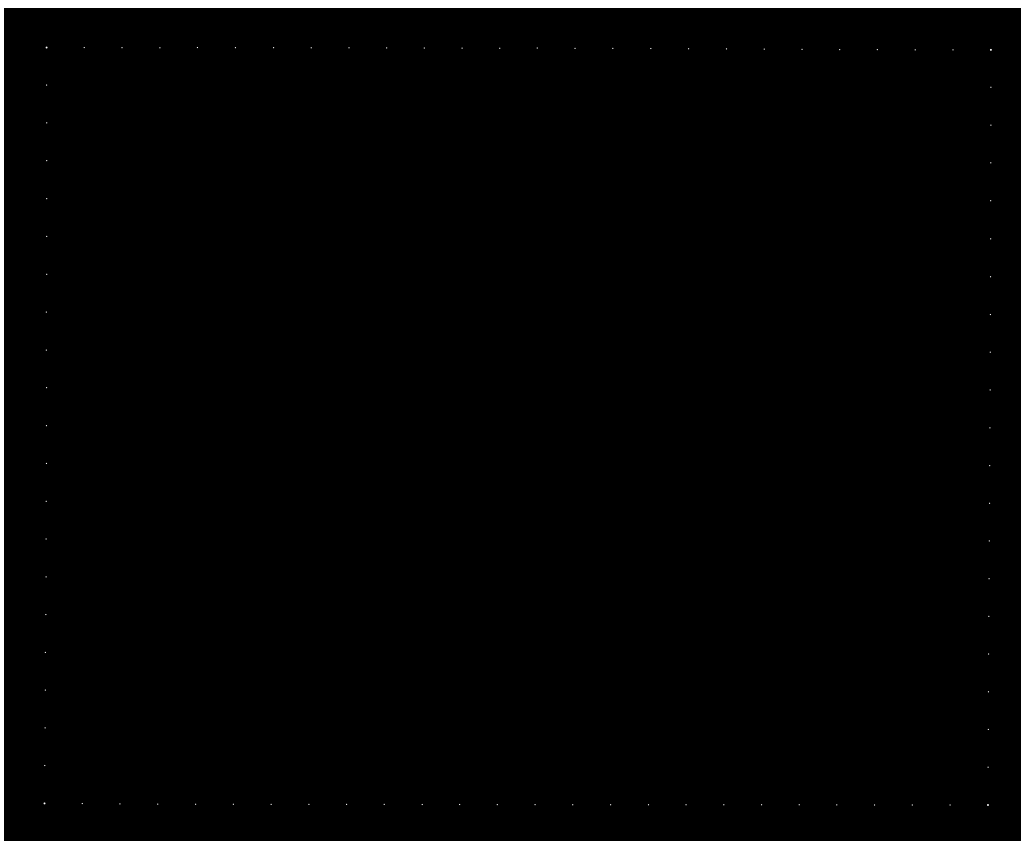
9 Dosažené výsledky

V tabulce 9.1 můžeme vidět srovnání stávající konzervativní metody dodané vedoucím práce a nově navržené metody využívající neuronovou síť. Výsledky obou těchto metod byly vyhodnoceny stejným programem zkoumajícím různé metriky mezi něž patří i námi zkoumané metriky (viz 8.2.1). Jediný parametr, který se pro vyhodnocování mění, je tolerační rádius. Toleranční rádius je oblast zájmu okolo palcové značky v ground-truth obrázku. Čím menší je, tím je menší tolerance pro odchylku vytvořeného obrázku jednou z metod a ground-truth obrázku.

Tabulka porovnání staré metody s metodou využívající FCN				
Metoda	Tolerační rádius	Precision	Recall	MAE
Původní	2	44,8	34,5	1,23
Původní	3	93,7	93,1	1,86
Původní	5	95,8	96,0	1,90
Původní	10	97,4	97,6	1,98
FCN	2	87,15	95,68	1,80
FCN	3	84,68	92,86	1,80
FCN	5	87,41	96,01	1,81
FCN	10	87,58	96,18	1,82

Tabulka 9.1: Srovnání jednotlivých parametrů použitých během trénování modelu a jejich výsledků

Z tabulky porovnání staré metody s metodou využívající konvoluční neuronovou síť lze vidět klesající trend u metrik *Precision*, *Recall* a *MAE* se zmenšujícím se toleračním rádiem u obou metod. Avšak u nově navržené metody využívající konvoluční neuronovou síť je toto klesání výrazně menší než u konzervativní metody. Nová metoda dosahuje výrazně lepší hodnoty i pro naše zkoumání klíčové metriky *MAE*. Na základě tohoto faktu lze konstatovat, že využití neuronových sítí pro zpracování obrazu v tomto případě může být vhodnou (lepší) alternativou oproti klasickým metodám pro detekci palcových značek v dodaných mapových podkladech.



Obrázek 9.1: Vizualizace hodnocení složeného obrázku z predikovaných výřezů



Obrázek 9.2: Přiblížený výřez z vizualizace hodnocení

Obrázky 9.1 a 9.2 znázorňují hodnocení již složených obrázků (z predikovaných výřezů) porovnané s jejich ekvivalentním ground-truth obrázkem. Barevné kruhy představují tolerační rádius okolo palcové značky na ground-truth obrázku. Čím zelenější barva daného kruhu reprezentující tolerační rádius, tím větší přesnosti predikce dané palcové značky bylo docíleno a naopak blíží-li se barva červené, přesnost predikované palcové značky je menší.

10 Závěr

Cílem práce bylo prostudovat metody pro zpracování obrazu pomocí neuronových sítí, seznámit se s dostupnými knihovnamí pro implementaci neuronových sítí, seznámení se s dodanými mapovými podklady a jejich zpracováním vytvořit základní datové sady vhodné pro trénování modelů neuronové sítě a následné vyhodnocení dosažených výsledků tohoto trénování. Za použití natrénovaného modelu měla být vytvořena metoda, která by umožnila detekci palcových značek na dodaných datech s následným vyhodnocením a srovnáním této metody s metodami doposud využívanými k řešení tohoto problému.

Těchto cílů bylo dosaženo. Seznámil jsem se s metodami pro zpracování obrazu pomocí neuronových sítí a s postupy jejich realizování. Následně jsem úspěšně vytvořil a implementoval metodu využívající neuronovou síť pro detekci palcových značek na dodaných mapových podkladech. Díky této metodě lze predikovat jednotlivé palcové značky a vytvořit tak data podobající se ground-truth datům. Při následných pracích například se skládáním mapových listů není nutné na jednotlivých mapách manuálně vyhledávat a označovat palcové ani rohové značky. Došlo ke zhodnocení navržené a implementované metody a k vyhodnocení jejích výsledků, podle kterých můžeme model označit za úspěšný, jelikož je srovnatelný a v některých klíčových aspektech lepší než doposud využívaná metoda, která nevyužívá neuronových sítí.

Jako možné rozšíření se nabízí analýza jiných druhů neuronových sítí, které by popřípadě mohly mít lepší výsledky. Lze využít například LSTM, tedy modelu nebo architektury, která rozšiřuje paměť rekurentních neuronových sítí. Výhoda LSTM je, že dokáže zohlednit prostorové rozložení palcových značek a tím může eliminovat falešné detekce. Jelikož samotné zpracování mapových podkladů, aby byly vhodné pro učení modelu, je celkem pracné a při větším množství dat může být časově náročnější, nabízí se možnost vynechat tento krok a naučit model pracovat s celými mapovými listy. Avšak toto řešení nemusí být tak efektivní, vzhledem k tomu, že navrhované modely neuronových sítí vykazují lepší výsledky za použití většího množství dat o menších velikostech, což dodané mapové podklady nespĺňují.

11 Použité zkratky

CUDA - Compute Unified Device Architecture

CCD - Charge-Coupled Device

API - Application Programming Interface

T - Treshold

GT - Ground Truth

JPG - Joint Photographic Group

PNG - Portable Network Graphics

12 Přílohy

Priložené DVD obsahuje text práce ve formátu **pdf** spolu s jeho zdrojovými soubory ve složce **text**. Implementace bakalářské práce je umístěna ve složce **src**. Data jsou umístěna ve složce **data**. Struktura přiloženého DVD je podrobněji popsána v souboru **readme.txt**, který se nachází v kořenovém adresáři.

13 Příručka

13.1 Instalace

Vytvořený program vyžaduje přítomnost nainstalovaného Pythonu v počítači a byl vytvořen ve verzi 3.8. Nutný je také nainstalovaný Tensorflow a to ve verzi 2.1.0 (minimálně však 2.0). Následně je nutná přítomnost softwaru Keras, jeho verze se však odvíjí od používané verze Tensorflow, na stránce <https://docs.floydhub.com/guides/environments/> je seznam kompatibilních verzí. Nutností je také knihovna OpenCV-Python a NumPy. Závislosti potřebné pro běh programu jsou uloženy v souboru **requirements.txt**. Závislosti se nainstalují s využitím příkazové řádky a balíčkového manažera PIP následovně: `pip install -r requirements.txt`

Využit lze také utility Anaconda nebo Miniconda, ve kterých si lze jednotlivý potřebný software instalovat.

13.2 Spuštění

Celý projekt plní dvě funkce: trénování modelu a následné predikování pomocí již naučeného modelu.

Příkaz pro trénování modelu má následující formát:

```
python trainer.py -e <počet epoch> -m <odsazení výřezu> -w  
<délka výřezu> -es <True/False - použití Early Stoppingu>, kde  
parametry -e, -m, -w, -es jsou volitelné a nepovinné.
```

Máme-li natrénovaný model, můžeme jej využít pro predikci následujícím příkazem:

```
python predictor.py <cesta ke složce obsahující dvě podsložky s  
testovacími daty> <délka výřezu>
```

Chceme-li následně zhodnotit přesnost predikovaných obrázků, můžeme využít přiložený program **inch_evaluator.py** příkazem:

```
python inch_evaluator.py <cesta ke ground-truth obrázkům> <cesta  
k predikovaným obrázkům>
```

Literatura

- [1] ABADI, M. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* [online]. TensorFlow, 2015. [cit. 2021/02/02]. Dostupné z: <https://www.tensorflow.org/>.
- [2] CHOLLET, F. *Deep Learning with Python*. 2018. ISBN 978-1-61729-443-3.
- [3] CLAUSNER, C. – PLETSCHACHER, S. – ANTONACOPOULOS, A. Scenario driven in-depth performance evaluation of document layout analysis methods. *2011 International Conference on Document Analysis and Recognition*, s. 2011. ISSN 1404–1408.
- [4] HARALICK, R. M. – SHAPIRO, L. G. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*. 1985, 29, 1, s. 100–132. ISSN 0734-189X. doi: [https://doi.org/10.1016/S0734-189X\(85\)90153-7](https://doi.org/10.1016/S0734-189X(85)90153-7). Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0734189X85901537>.
- [5] LIŠKA, M. *Segmentace historických obrazových dokumentů*. 2019.
- [6] REZATOFIGHI, S. H. et al. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, s. 658–666.
- [7] ROSEBROCK, A. *Intersection over Union (IoU) for object detection* [online]. 2016. [cit. 2021/03/09]. Dostupné z: <https://www.pyimagesearch.com/>.
- [8] SUMMERFIELD, M. *Python 3. Computer Press*. 2013. ISBN 978-80-251-2737-7.
- [9] TRAN, P. A Fully Convolutional Neural Network for Cardiac Segmentation in Short-Axis MRI. *ArXiv*. 2016, abs/1604.00494.
- [10] VOBECKÝ, A. *Data augmentation for neural networks training*. 2019.
- [11] WICK, C. – PUPPE, F. Fully convolutional neural networks for page segmentation of historical document images. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, s. 287–292. IEEE, 2018.