VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ANALÝZA ROZLOŽENÍ STRAN TEXTOVÝCH DOKUMENTŮ POMOCÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ

CONVOLUTIONAL NETWORKS FOR DOCUMENT LAYOUT ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE DAVID ENDRYCH
AUTHOR

VEDOUCÍ PRÁCE Ing. OLDŘICH KODYM,
SUPERVISOR

BRNO 2018

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

20900

Student: **Endrych David**
Program: Informační technologie
Název: **Analýza rozložení stran textových dokumentů pomocí hlubokých neuronových sítí**
Convolutional Networks for Document Layout Analysis
Kategorie: Zpracování obrazu
Zadání:

1. Prostudujte základy konvolučních sítí a analýzy rozložení stran textových dokumentů.
2. Vytvořte si přehled o současných metodách analýzy rozložení stran textových dokumentů pomocí konvolučních sítí.
3. Vyberte nebo navrhněte metodu aplikovatelnou na analýzu rozložení stran textových dokumentů.
4. Obstarejte si databázi vhodnou pro experimenty.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
- http://www.image-net.org/challenges/LSVRC/2013/results.php

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Kodym Oldřich, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 1. listopadu 2018

Abstrakt

Cílem této bakalářské práce je vytvořit nástroj pro analýzu rozložení stran textových dokumentů. Problém je řešen pomocí konvolučních neuronových sítí. Architekturou zvolenou v téhle bakalářské práci je architektura U-Net. Pro trénování modelu sítě se používá chybová funkce cross entropy s použitím mapy vah. Pomocí hledání spojitých komponent dochází k získávání regionů odstavců. Experimenty se vyhodnocují pomocí objektové metriky Symmetric Best Dice. Z experimentů vyplynulo, že je lepší používat všechny hrany odstavců než se zaměřovat pouze na vertikální hrany odstavců. Dále experimenty ukazují, že trénovací strategie vzorkování batche a adaptativní rozlišení pomáhají ke zlepšení výsledků analýzy. V experimentech je také popsána aplikace separátorů, která je užitečná při analýze vícesloupcových dokumentů.

Abstract

The goal of this thesis is to create a tool for analyzig the page layouts of text documents. The problem is solved by convolution neural networks. The architecture chosen in this thesis is the U-Net architecture. The cross entropy error function with weight map is used for train the network model. Paragraph regions are obtained throught connected component analysis. Experiments are evaluated using the Symmetric Best Dice object metric. Experiments have shown that it is better to use all paragraph edges than to focus only on vertical paragraph edges. In addition, experiments show that batche sampling strategies and adaptive resolution help to improve analysis results. The experiments also describe the application of separators, which is useful in analyzing multi-column documents.

Klíčová slova

počítačové vidění, hluboké neuronové sítě, analýza rozložení stran, segmentace obrazu, U-Net, umělá inteligence

Keywords

computer vision, deep neural networks, page layout analysis, image segmentation, U-Net, artificial intelligence

Citace

ENDRYCH, David. *Analýza rozložení stran textových dokumentů pomocí hlubokých neuronových sítí.* Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Oldřich Kodym,

Analýza rozložení stran textových dokumentů pomocí hlubokých neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Oldřicha Kodyma. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
David Endrych
14. května 2019

Poděkování

Chtěl bych poděkovat panu Ing. Oldřichovi Kodymovi za odborné vedení práce, konzultace, trpělivost a cenné rady, které mi pomohly tuto práci vytvořit.

Obsah

Kapitola 1

Úvod

Tato práce se zabývá analýzou rozložení stran textových dokumentů pomocí konvolučních neuronových sítí. Cílem této práce je vytvořit nástroj, který analyzuje vstupní obrázek dokumentu a vrací informace o rozložení strany textového dokumentu. Motivací k řešení tohoto problému je aplikace v post-processingu detekce řádků. Při detekci řádků může docházet ke spojení nebo ke spojení sousedních řádků. Pomocí informací získaných z analýzy rozložení stran se můžou tyhle chyby v detekci opravit.

V kapitole 2 je vysvětlena problematika analýzy rozložení stran textových dokumentů. Jsou zde popsány čtyři existující řešení, které se touhle problematikou zaměřují. Jako první je uvedeno řešení pomocí hledání vzorů, dále následuje analýza rozložení stránek řeckých novin a jednorozměrná konvoluční neuronová síť pro rychlé zpracování dokumentů a segmentační model dhSegment pro obecné řešení segmentačních úloh, jako je extrakce stránky, detekce řádků nebo právě analýza rozložení stran.

Kapitola 3 se zaměřuje na návrh řešení nastoleného problému. Je zde vysvětlena využitá architektura neuronové sítě U-Net, která se využívá pro segmentaci obrazu, chybová funkce cross entropy, která se využívá při učení modelu včetně využití mapy vah pro zlepšení učení sítě. Potom následuje popis postupu zpracování dokumentů s využitím natrénovaného modelu včetně použitého post-processingu. Na konci kapitoly je popsána metrika úspěšnosti Symmetric Best Dice, která se využívá pro vyhodnocování úspěšnosti experimentů a nástroje, které jsou použity pro implementaci.

Kapitola 4 se zaměřuje na použitou datovou sadu Impact dataset, která se v této práci využívá pro trénování a vyhodnocení úspěšnosti modelu neuronové sítě. V kapitole je vysvětleno, jak data v datové sadě vypadají, co datová sada obsahuje nebo z čeho se datová sada skládá. Potom jsou zde popsané úpravy, které byly na datové sadě aplikovány pro účely trénování modelu.

Kapitola 5 se zaměřuje na proběhlé experimenty v rámci této práce. Zde jsou jako první popsány experimenty, které se zaměřují na formulaci úlohy. Jedná se o binární segmentaci, kde dochází k rozlišovaní mezi odstavcem a zbytkem obsahu. Dále k těmhle klasifikačním třídám přibude třída reprezentující hrany odstavců. Následující experiment se zaměřuje na učení pouze vertikálních hranic odstavců, aby zjednodušil učení modelu neuronové sítě. Další část experimentů se zaměřuje na trénovací strategie. Zde první experiment má za cíl zlepšit výběr batche při trénování neuronové sítě. Druhý experiment je zaměřen na adaptivní rozlišení, kde je snaha, aby data, které jdou na vstup sítě byli na sebe více podobná. Poslední experiment se zaměřuje na zlepšení rozdělování více sloupcových dokumentů, převážně novin pomocí aplikace vertikálních separátorů.

Kapitola 6 obsahuje diskuzi o dosažených výsledcích.

Kapitola 2

Analýza rozložení stran textových dokumentů

Tato kapitola se zaměřuje na problematiku analýzy rozložení stran textových dokumentů. Zde představím aktuální metody řešení tohoto problému. Jako první popíši řešení založené na hledání vzorů v obrázku dokumentu a segmentaci novinových článků pomocí Bayesovských modelů. Dále v kapitole představím model pro rychlou analýzu dokumentů, který využívá jednorozměrnou konvoluční neuronovou síť a řešení analýzy pomocí obecného modelu hlubokého učení dhSegment.

2.1 Úvod do problematiky

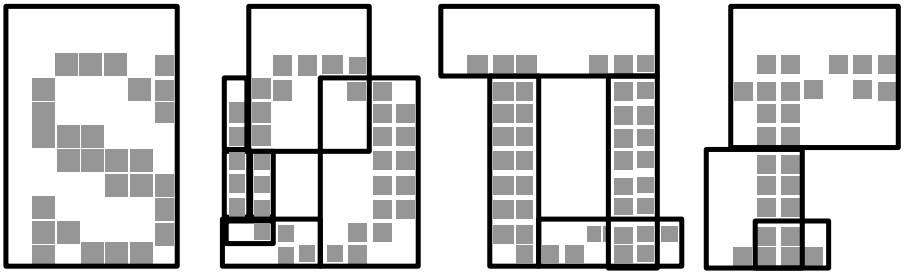
V dnešní době stále více dochází k digitalizaci textových dokumentů. To vede ke stále častějšímu používání OCR (Optical Character Recognition), které slouží pro přepis textu z naskenovaných obrázku do digitální formy. Díky získaným přepisům můžeme informace o dokumentech daleko lépe uchovávat, můžeme v nich vyhledávat nebo dokonce editovat. Důležitým krokem k extrakci informací z dokumentů je analýza rozložení stran. Cílem analýzy je identifikovat jednotlivé regiony dokumntu, které následně můžeme klasifikovat a seřazovat. Informace získané analýzou jsou použity pro OCR k určení textových regionů, které se mají zpracovávat. Také můžou být použity pro detekci duplikátů dokumentů nebo k indexaci dokumentů podle jejich struktury.

Metody pro analýzu textových dokumentů můžeme rozdělit do několika skupin [16]:

- Metody založené na klasifikací regionů nebo bloků
- Metody založené na klasifikaci jednotlivých pixelů
- Metody založené na klasifikaci spojitých komponent (Connected component)

Metody založené na klasifikaci regionů nebo bloků segmentují dokument do jednotlivých bloků, které klasifikují do jednotlivých tříd, jako je např. titulek, odstavec, obrázek apod. Metody založené na klasifikaci pixelů přiřazují jednotlivým pixelům klasifikační třídy individuálně, a potom vytvářejí z hypotéz jednotlivé regiony. Poslední skupina metod je založená na klasifikaci komponent souvislosti. Tyto metody využívají místní informace k vytváření objektových hypotéz, které se kombinují, přezkoumávají, a nakonec se klasifikují.

Dále se metody můžou dělit podle směru:



Obrázek 2.1: Ukázka hledání vzorů ve slově sour [17].

- Přístup zdola nahoru
- Přístup shora dolů

Přístup zdola nahoru iterativně analyzuje dokument podle jednotlivých pixelů. Metody založené na tomhle přístupu typicky zpracují, jako první dokument do spojených regionů černé a bílé barvy, potom jsou tyto regiony seskupovány do slov, následně do jednotlivých řádků textu, a nakonec do textových bloků. Oproti tomu přístup shora dolů se iterativně snaží rozřezat dokument do jednotlivých sloupců a bloků na základě bílých mezer a geometrických informací [2].

2.2 Řešení pomocí vzorů

Prvním možným řešením je pomocí hledání vzorů. Tuto metodu představují v článku [17] Phillip E. Mitchell a Hong Yan. Hlavní komponentou této metody je segmentační algoritmus, který identifikuje v dokumentu text a oblasti grafického obsahu. Jako první algoritmus hledá a lokalizuje základní vzory dokumentu a následně na nich provádí klasifikaci pomocí charakteristiky run-length [8], analýzy šíření a sousedních vztahů. Run je definován jako řetězec po sobě jdoucích pixelů v určitém směru (typicky v 0°, 45°, 90° a 135°), které jsou stejné úrovně šedi. Run-length je definován, jako počet opakujících pixelů v tomto řetězci. Další věc, kterou má algoritmus na starost je řazení obsahu dokumentu podle pořadí čtení dokumentu. Algoritmus zvládá jak jednosloupcové, tak i vícesloupcové dokumenty.

V první fázi algoritmus vyhledává vzory, které ukládá do spojitého seznamu z důvodu rychlé přístupnosti v době klasifikace. Jako první algoritmus vyhledává 9x9 oblasti, které obsahují černý pixel. Vzory jsou hledány pomocí spojování sousedních oblastí. Oblasti jsou spojeny, pokud jsou od sebe vzdáleny nejvíce jeden pixel od sebe. Vzory pro textové oblasti jsou většinou jednotlivé znaky nebo slova v závislosti na velikosti textu. Na obrázku 2.1 lze vidět ukázku hledání vzorů na slově sour.

Dalším krokem je klasifikace vzorů. Cílem klasifikace je rozhodnout, jestli jsou jednotlivé vzory text nebo grafický obsah. Vzory jsou klasifikovány podle několika vlastností, podle velikosti, počtu černých pixelů a run-length statistiky. Na začátku klasifikace jsou všechny vzory prohlášeny za textový obsah a od toho se následně odvíjí pravidla. Příklady klasifikačních pravidel lze vidět v tabulce 2.1.

Potom následuje další část klasifikace, a to je klasifikace kontextu. Jelikož vzory se vyskytují většinou ve skupinách, tak je potřeba udělat znovu klasifikaci obsahu. Tato klasifikace se odvíjí od klasifikací sousedních vzorů. Příklady pravidel jsou vidět v tabulkách 2.2 a 2.3.

Klasifikační pravidlo	**Vysvětlení**
Malé vzory nebo malý počet černých pixelů	Malé vzory např. interpunkce nebo šum jsou odstraněny a klasifikace probíhá až později podle kontextu.
Velké vzory.	Identifikuje grafiku, obrázky.
Vzor je dlouhý a úzký.	Lokalizace čar nebo řádkové grafiky.
Velká plocha nebo velký poměr černých pixelů na bílé.	Jedná se o grafický obsah.
Vysoký průměr nebo maximální run-length černé barvy.	Textové vzory.

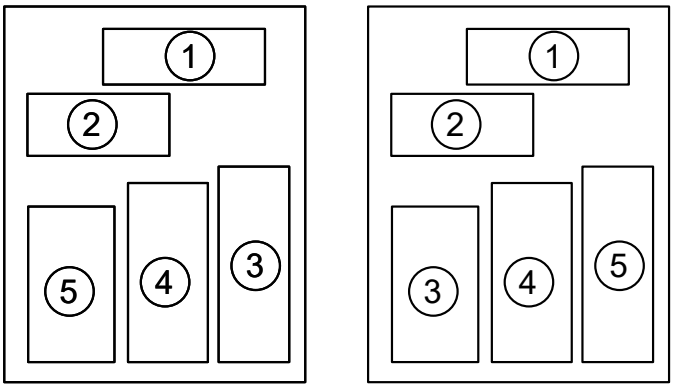
Tabulka 2.1: Klasifikační pravidla pro vzory

Kontextové pravidlo	**Vysvětlení**
Standardní odchylka velikosti oblasti je větší než 1.0 a jsou zde alespoň 4 grafické vzory.	Nerovnoměrná velikost vzoru značí, že vzor je grafický obsah.
Vzor leží uvnitř velkého grafického vzoru a jsou zde alespoň 4 grafické vzory.	Vzory uvnitř jiných velkých grafických vzorů jsou obvykle také grafický obsah.
Grafické vzory přečíslují textové vzory v oblasti alespoň v poměru 2:1 a je zde méně než 5 textových vzorů.	Text většinou není izolován okolo, tak velkého počtu grafických vzorů.

Tabulka 2.2: Kontextová pravidla pro textové vzory

Kontextové pravidlo	**Vysvětlení**
Standardní odchylka velikosti oblasti je menší než 1.1 a jsou zde alespoň 3 textové vzory.	Textové regiony jsou charakteristické pravidelnými velikostmi.
Textové vzory přečíslují grafické vzory v oblasti alespoň v poměru 2:1 a jsou zde méně než 3 grafické vzory.	Grafický obsah nebývá izolován okolo, tak velkého počtu textových vzorů.

Tabulka 2.3: Kontextová pravidla pro grafické vzory



Obrázek 2.2: Rozdíl mezi jednoduchým řazením bloků a řazením pomocí algoritmu jemného řazení (soft ordering). [17].

Po klasifikaci vzorů dochází k vytváření bloků, které reprezentují odstavce, titulky nebo grafický obsah dokumentu. Pokud nejsou od sebe moc vzdáleny, tak dochází k jejich seskupování. Limity jsou definovány pomocí rovnic

$$hlimit = 1,1 * avh \tag{2.1}$$

a

$$vlimit = 0,8 * avh, \tag{2.2}$$

kde *avh* je průměrná výška vzorů, *hlimit* je limita v horizontálním směru a *vlimit* je limita ve vertikálním směru.

Následně dochází k řazení bloků. Nejednoduší varianta řazení bloků by byla řadit bloky podle horních hran odstavců a následně podle levého okraje. To ale tvoří nepřirozené řazení odstavců, jelikož způsobuje spoustu chyb v případě více sloupcových dokumentů. Flexibilnější algoritmus by měl dovolovat sloupce, které jsou vedle sebe řadit zleva doprava nehledě na horní hraně bloku. Algoritmus jemného řazení (soft ordering) jde dále a to tak, že dovoluje řazení podle levého okraje bloku v závislosti na výšce bloku. Limita, která určuje, zda mohou být dva vzory mimo pořadí (nebo se překrývat) za pomocí jemného řazení je definována pomocí funkce sigmoid:

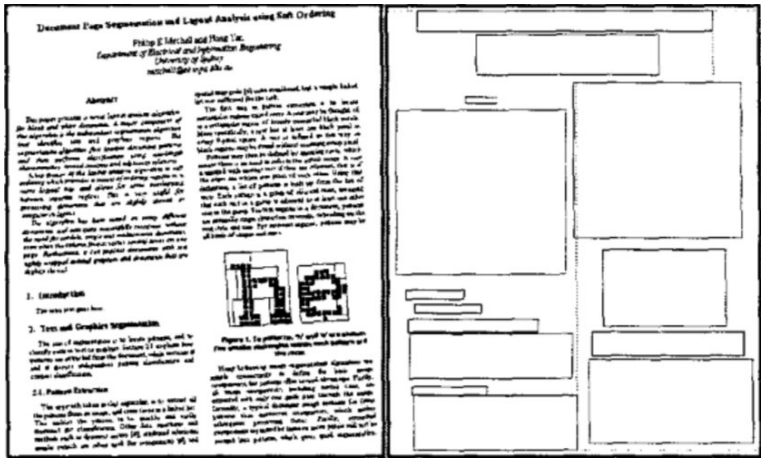
$$limit = \frac{avh}{1 + \frac{4}{avh}\epsilon^{avh-minh}}, \tag{2.3}$$

kde *avh* je průměrna výška vzoru a *minh* je minimální výška bloků, které jsou řazeny. Na obrázku 2.2 lze vidět rozdíl mezi jednoduchým řazením bloků a jemným řazením.

Na obrázku 2.3 lze vidět výsledek analýzy dokumentu pomocí metody hledání vzorů.

2.3 Segmentace článků novin pomocí Bayesovských modelů

Autoři článku [25] Giorgos Sfikas, Georgios Louloudis, Nikolaos Stamatopoulos a Basilis Gatos prezentují algoritmus pro segmentaci obrázku novin do jednotlivých článků. Algoritmus se skládá ze dvou částí. V první části se detekují regiony obsahující text, nadpisy



Obrázek 2.3: Výsledek analýzy pomocí hledání vzorů [17].

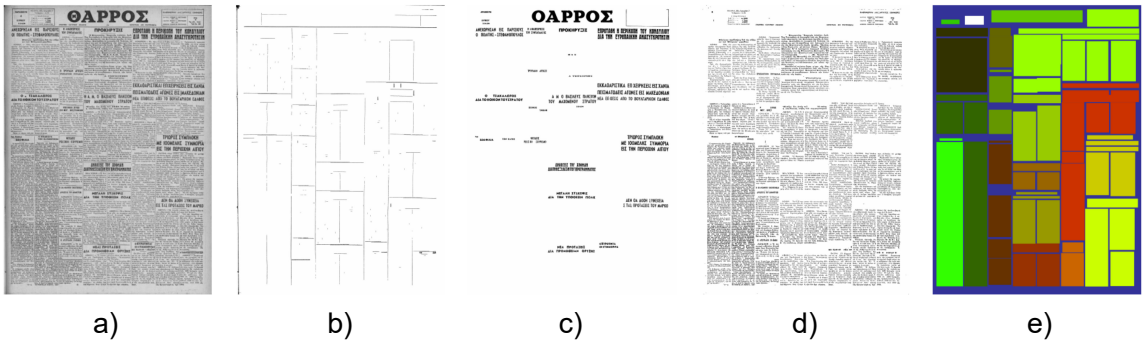
a zbývající obsah. V další části se tyto nalezené regiony spojují a vytvářejí se z nich regiony článků. Pro testování modelu byla použita historická datová sada regionálních řeckých novin Tharros, které jsou vydávány od roku 1899.

Jako první se provede analýza komponent souvislosti (Connected Component Analysis) na binárních datech vstupního obrázku. Komponenty souvislosti jsou používány po celou dobu běhu algoritmu jako základní prvek, který utváří regiony. Komponenty souvislosti s dominantní hlavní osou nad jejich vedlejší osou jsou označeny za separátory. Na separátorech a komponent souvislosti, které jsou až příliš malé vzhledem k průměrné velikosti komponent souvislosti se provádí shluková analýza pomocí Fully Bayesian Gaussian Mixture Model (FBGMM) [25]. Autoři článku určili pro každou komponentu souvislosti tři vlastnosti, které se v FBGMM využívají a to šířku, výšku a tloušťku komponenty souvislosti. Tloušťka je vypočítaná jako maximální transformace vzdálenosti od zbývajících komponent souvislosti. Po vyhlazení výsledku shlukové analýzy vzniknou komponenty reprezentující obsah, titulek a ostatní komponenty. Pokud se tyhle komponenty překrývají s některým z nalezených separátorů, dojde k rozdělení komponenty. Po odstranění pixelů reprezentující nadpisy a separátory, dojde k analýze komponent souvislosti, díky které se získají odstavce reprezentující regiony obsahu.

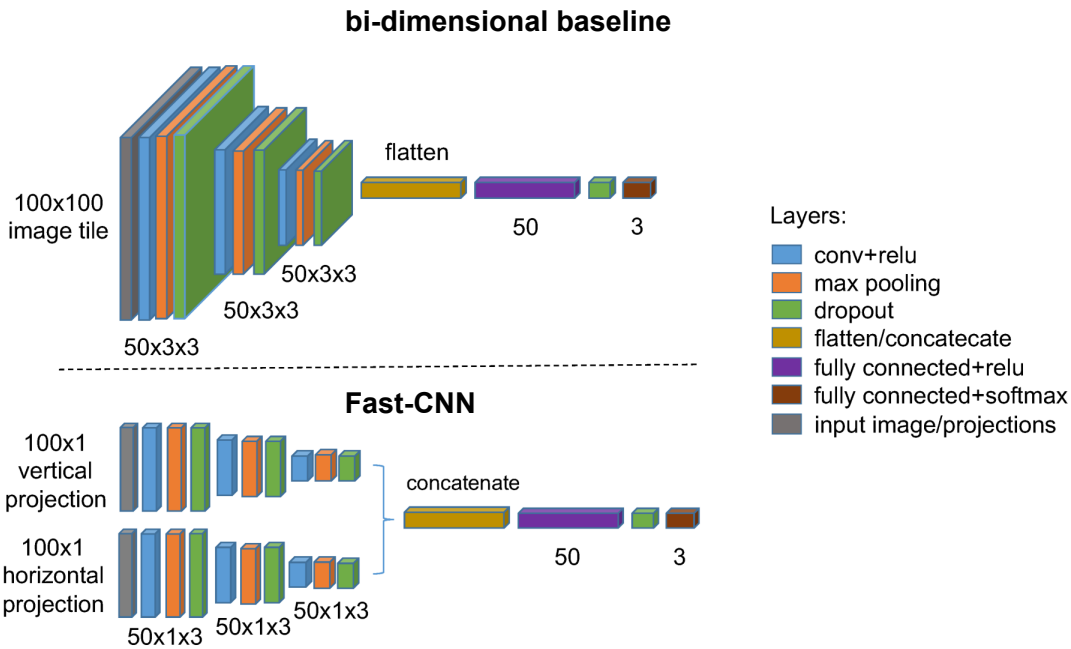
Hledání článků se provádí na základě předchozí segmentace, která rozdělí obrázek na regiony nadpisů a regiony obsahu. Sousední regiony nadpisů se spojí do skupiny. Následně dojde k přiřazení regionů obsahu ke skupině nadpisů. Přiřazení regionu probíhá tak, že se hledá skupina nadpisů, která je nad daným regionem obsahu. Skupina nadpisů a k něm přiřazené regiony tvoří dohromady hledané články. Výsledek a postup analýzy lze vidět na obrázku 2.4.

2.4 Rychlá analýza dokumentů založena na konvolučních sítí

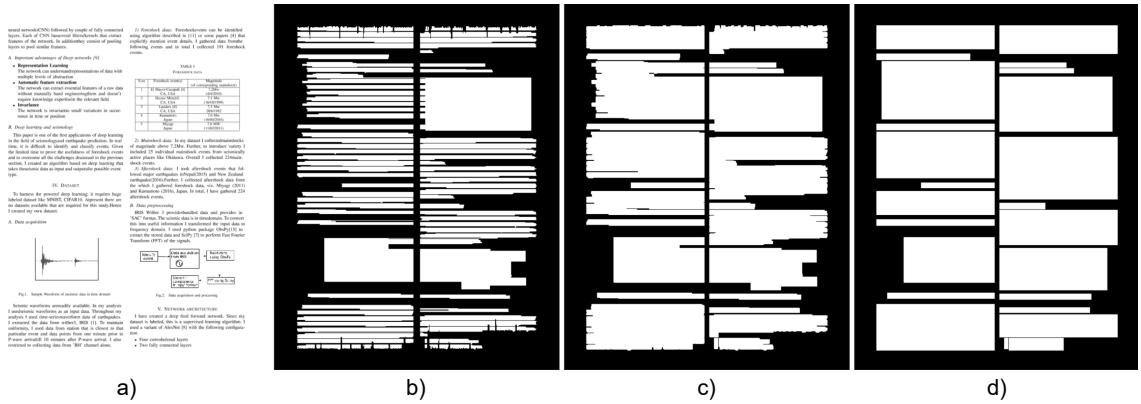
Dário A. B. Oliveira a Matheus P. Viana se pro hledání regionů obsahu v článku [26] rozhodli použít jednorozměrnou konvoluční neuronovou síť (Fast-CNN). Ta je složená ze dvou větví, každá zpracovává jednu 1D projekci, následně se spojí a poslední společná vrstva provede klasifikaci. Baseline metoda se kterou autoři článku porovnávají metodu Fast-CNN, je založena na 2D konvoluční síti. Obě architektury jsou zobrazeny na obrázku 2.5.



Obrázek 2.4: Analýza textového dokumentu pomocí Bayesovských modelů [3]. a) Vstupní obrázek textového dokumentu. b) Nalezené separátory. c) Nalezené nadpisy. d) Nalezené regiony obsahu. e) Výsledek analýzy



Obrázek 2.5: Srovnání popsaných architektur sítí pro analýzu dokumentu. Nahoře architektura bi-dimensional baseline. Dole architektura jednorozměrné konvoluční sítě (Fast-CNN) [26].



Obrázek 2.6: Postupné zpracování dokumentu [26]. a) Vstupní obrázek dokumentu převedený do černobílé barvy. b) Výsledek run-length algoritmu. c) Po použití dvou 3x3 dilatací. d) Výsledné bloky obsahu.

Pro získání datasetu pro učení modelu použili 300 dokumentů ze stránky ArXiv.[1] Regiony obsahu získali autoři pomocí poloautomatické metody. Tato metoda je založena na run-length algoritmu, který detekuje regiony, u kterých je šance, že obsahují informace. Následně se pomocí dilatace a hledání největších spojených komponent získají bloky obsahu. Tenhle postup a získaný výsledek lze vidět na obrázku 2.6. Po získání regionů došlo k manuálnímu určení klasifikačních tříd. Autoři práce se rozhodli rozlišit tři klasifikační třídy text, tabulku a obrázek. Tímhle procesem získali dataset, který obsahuje 99 bloků tabulek, 2995 bloků obrázků a 4533 textových bloků.

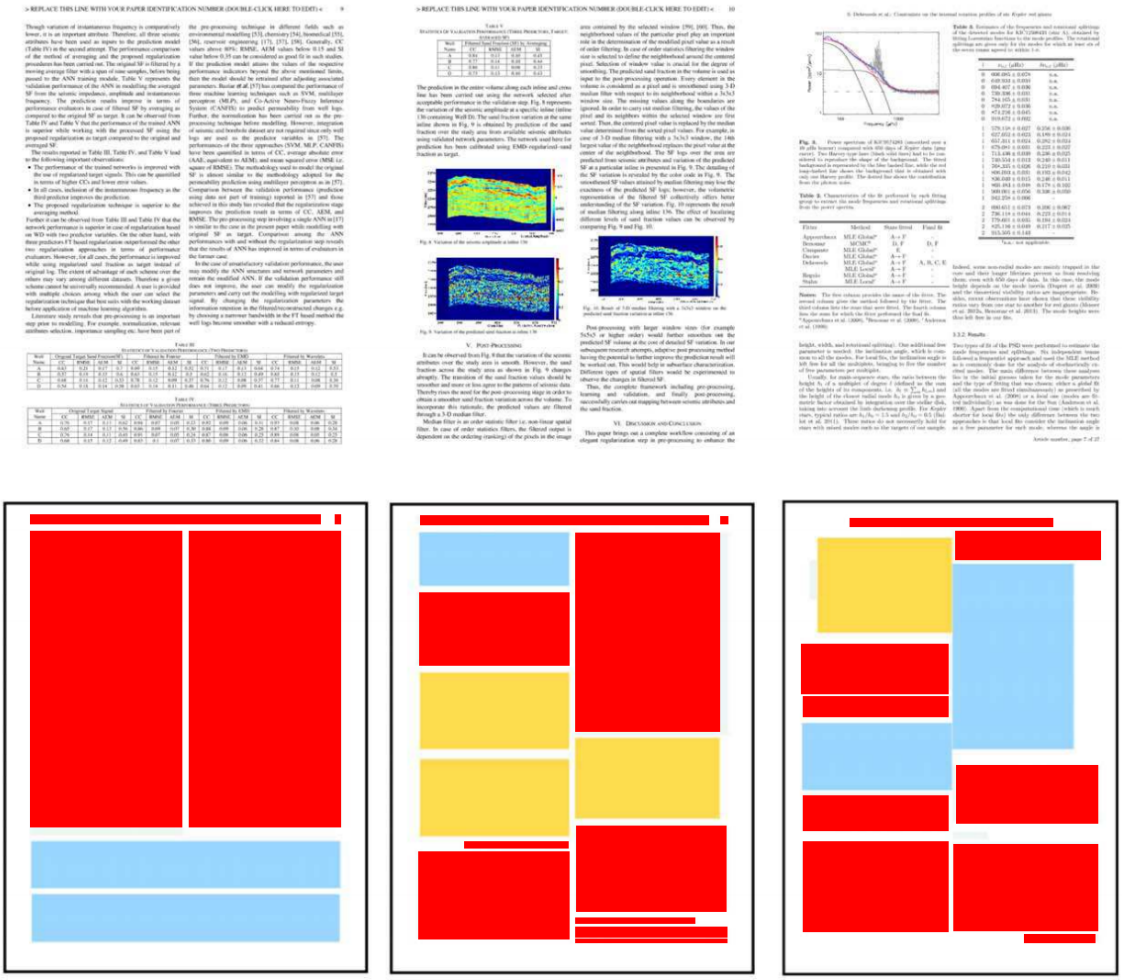
Autoři pro trénování konvoluční sítě využívají podmnožinu datasetu, která obsahuje 90 vzorků každé třídy. Na vybraných datech provedli augmentaci dat pomocí vzorkování 100x100 pixelů posuvného okna s krokem 30. Pomocí tohoto procesu vytvořili databázi, která obsahovala 6092 dílů 100x100 pixelů od každé klasifikační třídy. Dále se dataset rozdělil na dvě části 80% pro trénování a 20% pro testování přesnosti natrénovaných modelů.

Po natrénování modelů došlo k vyhodnocovaní na kompletním datasetu. Implementovanou metodu v práci porovnávali s řešením bi-dimensional baseline. Fast-CNN i bi-directional baseline dosáhly podobné úspěšnosti, Fast-CNN však bylo podstatně rychlejší při trénování i vyhodnocování. Na obrázku 2.7 lze vidět ukázku segmentace.

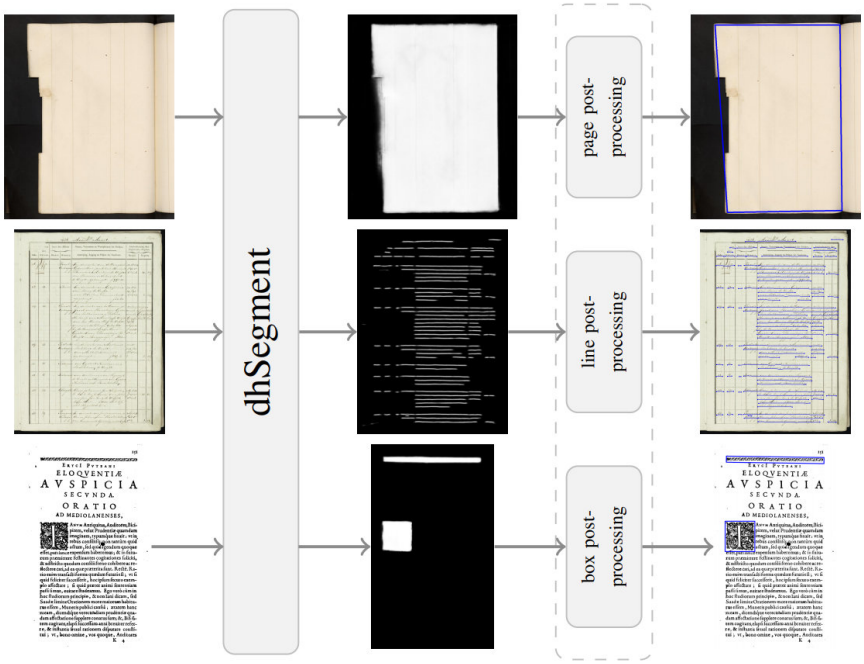
2.5 dhSegment: Segmentace dokumentů pomocí obecného hlubokého učení

Autoři Sofia Ares Oliveira, Benoit Seguin a Frederic Kaplan v článku [3] představují architekturu konvoluční neuronové sítě pro obecnou segmentaci dokumentů dhSegment. Systém je postaven na dvou po sobě jdoucích částech, které lze vidět na obrázku 2.8. První částí je plně konvoluční neuronová síť, která predikuje pravděpodobnosti jednotlivých klasifikačních tříd pro každý pixel obrázku, který dostane na vstup. Tím vzniká pravděpodobnostní mapa, která se využije v druhé části systému. Zde dochází k transformaci map pravděpodobností na výsledný výstup úlohy.

[1] `https://arxiv.org/`



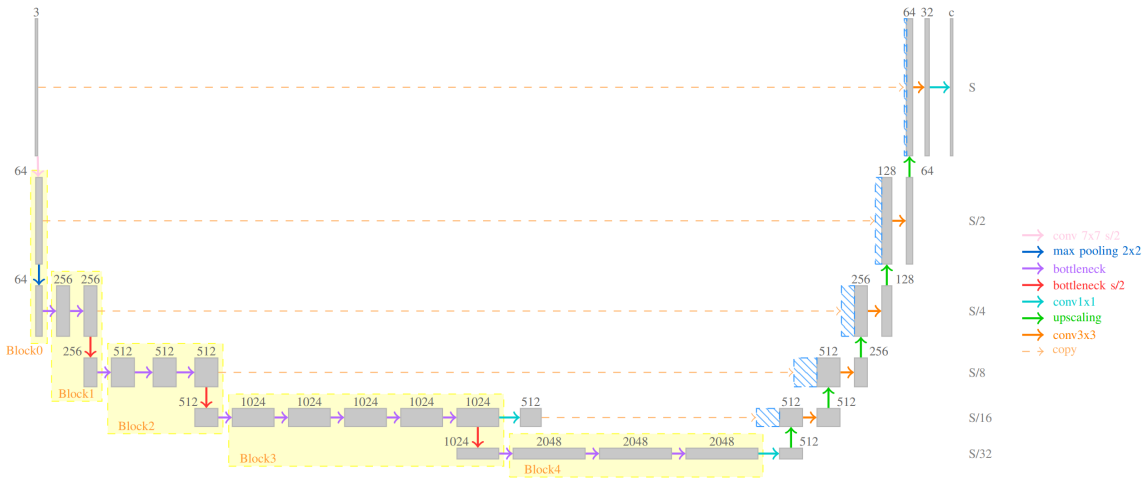
Obrázek 2.7: Výstup segmentace pomocí modelu Fast-CNN [26]. Žlutá barva značí textové bloky, modrá tabulky a žlutá barva značí obrázky.



Obrázek 2.8: Přehled systému dhSegment. V první fázi dojde k získání pravděpodobnostní mapy ze vstupního obrázku. V druhé části se dělá transformace pravděpodobnostní mapy na požadovaný výstup systému. [3].

Architektura sítě dhSegment je zobrazena na obrázku 2.9. Architektura dhSegment je založena na architektuře U-Net, která bude popsána v kapitole 3. Autoři však jako jeho první část využili ResNet-50 [9] předtrénovaný na ImageNetu [6]. To pomáhá, aby byla síť dokázala více generalizovat. Další část sítě se nazývá expansive path. Tato část mapuje mapy funkcí (feature maps) v nízkém rozlišení zpět na původní rozlišení. Expansive path je složena z pěti bloků a závěrečné konvoluční vrstvy, která přiřazuje klasifikační třídu každému pixelu. Každý blok, který reprezentuje dekonvoluci je složen konkatenací zvětšení předchozího kroku a odpovídající kopií mapy funkcí první části sítě. Jakmile dojde ke spojení bloků následuje 3x3 konvoluce s následným ReLu. Z důvodu snížení počtu parametrů sítě a využití paměti dochází v závěrečných krocích sítě ke snížení počtu kanálů map funkcí na 512 pomocí 1x1 konvoluce, které jsou znázorněny na obrázku 2.9 tyrkysovou šipkou.

Cílem této publikace bylo ukázat efektivnost a generalizaci modelu dhSegment. Z toho důvodu autoři použili pro zpracování výsledků modelu pouze základní post-processing. Jako první se používá prahování, které je použito k získání binární mapy z výstupu výše popsané neuronové sítě. Dalším post-processing, kteří autoři použili bylo použití morfologických operací, konkrétně se jedná o použití eroze a dilatace. Dalším post-processingem, který byl v práci využit byla analýza spojitých komponent. Tento post-processing využili k filtrování malých spojených komponent, které zůstávají po prahování a morfologických operací. Dále pro získávání souřadnic z detekovaného regionu používají vektorizaci tvarů (shape vectorization). Díky tomuhle kroku je možné získat z binárního obrázku body polygonů. I přesto většinou dochází k získávání tzv. bounding boxů (reprezentovány pomocí čtyř souřadnic). Při trénování sítě použili autoři L2 regularizaci [5], optimalizační algoritmus Adam [14] a batch normalizaci [11]. Obrázky, které jsou posílány na vstup do neuronové sítě jsou



Obrázek 2.9: Architektura neuronové sítě dhSegment [3]. Žlutá část přestavuje architekturu ResNet-50 [9].

ořezány a zmenšeny na 300x300 pixelů, také jsou k obrázkům přidány volné kraje, aby se předešlo problémům s okrajem stránek. Pro zlepšeni generalizace je použita také augmentace dat [20]. Jedná se zde o změnu rozlišení, rotaci a zrcadlení vstupů do sítě. Tyto úpravy se aplikují až při trénování sítě.

Účinnost modelu dhSegment je otestována na pěti různých typech segmentačních úloh pro analýzu dokumentů. Jedná se o extrakci stránky, detekci řádků, analýzy rozložení stránky, detekci ornamentů a extrakci fotek. Extrakce stránek má za cíl určit, která oblast reprezentuje obsah dokumentu a tím odfiltrovat např. okraje stránky, které by mohli v dalších typech analýzy (detekce řádků, analýza rozložení stránky) způsobovat falešné detekce. Metoda dosáhla state of art výsledků ve všech úlohách.

Kapitola 3

Navrhované řešení

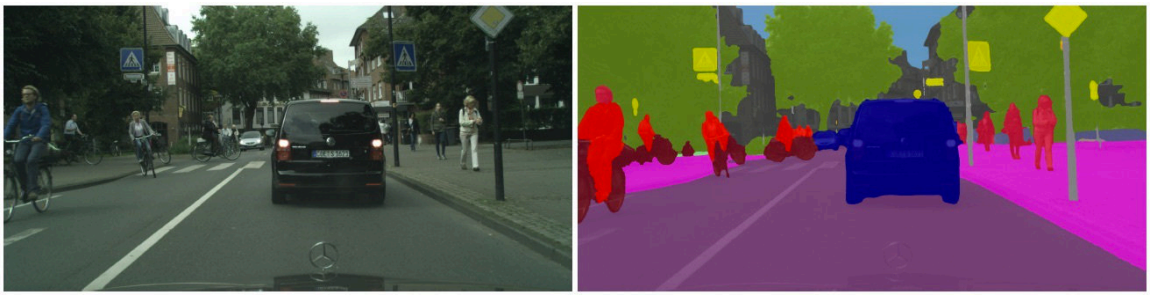
Tato kapitola se zabývá návrhem řešení mé práce. Nejprve vysvětlím architekturu neuronové sítě U-Net, kterou jsem se rozhodl použít pro vyřešení zadaného problému. Potom popíšu chybovou funkci cross entropy včetně vysvětlení rozšíření chybové funkce o mapu vah. Dále se bude navazovat vysvětlením postupu zpracování textového dokumentu s využitím natrénovaného modelu neuronové sítě včetně použitého post-processingu a nakonec dojde v kapitole na popis vybrané metriky úspěšnosti Symmetric Best Dice (SBD) a k výběru technologií použitých pro implementaci.

3.1 Architektura neuronové sítě

Pro moji neuronovou síť jsem se rozhodl použít architekturu U-Net [22], kterou vytvořili Olaf Ronneberger, Philipp Fischer a Thomas Brox. Jedná se o jednu se základních architektur konvolučních neuronových sítí určené k segmentaci obrazu, kdy nám nestačí pouze rozhodnout do které třídy patří obraz jako celek, ale chceme pro každý pixel vstupních dat rozhodnout do které klasifikační třídy daný pixel patří, jak je možné vidět na obrázku 3.1. U-Net si našel uplatnění při zpracování obrazu v biomedicíně např. pro segmentaci snímku mozku [7].

U-Net je založený na plně konvoluční neuronové síti. Skládá se ze dvou částí, které lze vidět na obrázku 3.2. V první části sítě, která se nazývá contracting path dochází k postupnému zmenšování vstupního obrázku. Díky tomu je síť schopna získávat informace o kontextu z větší oblasti obrázku. Tato část se skládá z opakovaných aplikací dvou 3x3 konvolučních vrstev. Po každé z vrstev následuje aktivace pomocí ReLu a vrstva 2x2 max poolingu s krokem dva. Tato vrtva způsobuje postupné zmenšování vstupního obrázku. Při každém zmenšení zdvojnásobujeme počet kanálů. Druhá část sítě, které se říká expansive path upřesňuje lokalizaci. V této části dochází v jednotlivých krocích k opětovnému zvětšování rozlišení pomocí 2x2 up-convolution. Potom dojde k připojení vrstev z contracting path k vrstvám na odpovídajícím stupni expansive path. V poslední vrstvě dochází k mapování vektoru kanálů z 64 na počet odpovídající počtu klasifikačních tříd.

Pro svůj segmentační model jsem se rozhodl udělat určíté úpravy. Jako první jsem zvolil menší vstupní obrázek z důvodu menších nároků na paměť a zrychlení učení modelu. Místo rozlišení 572x572 jsem se rozhodl použít rozlišení 300x300. Toto je velikost patche, která je dost malá, ale obsahuje dostatek informací pro vyvození informace o textových regionech. Dalším rozdílem je použití jiného typu paddingu u 2D konvolucí, tím se zajistí, že se při konvolucích nezmenšují data a to zajistí, že výsledné data mají stejné rozlišení, jako data



Obrázek 3.1: Ukázka segmentace obrazu. Vlevo je zobrazen vstup do sítě a vpravo výstup. Pro každý pixel je určena klasifikační třída např. modrá pro auto, červená je člověk nebo žlutá dopravní značka [15].

vstupní. Další změnou bylo zmenšení počtu kanálů sítě. V článku autoři začínali na 64 kanálech, které postupně zdvojnásobovali při max poolingu. Z důvodu zmenšení nároků na paměť a zrychlení učení sítě jsem se rozhodl použít 8 kanálů, které, stejně jako je tomu v článku, postupně zdvojnásobuji.

3.2 Popis trénování

Důležitou součástí každé neuronové sítě je chybová funkce. Tato funkce udává, jak moc se výsledek sítě liší od požadovaného výsledku. Nejznámější chybové funkce, které se používají v neuronových sítí [12] jsou Mean Squared Error (MSE), Mean Squared Logarithmic Error (MSLE), Cross Entropy nebo Hinge Loss. Pro moji síť jsem se rozhodl využít chybovou funkci cross entropy. Cross entropy se nejvíce využívá v binární klasifikaci (v případě, kdy rozhodujeme mezi dvěma klasifikačníma třídami). Cross entropy je definováno jako

$$H(y,p) = \sum_i y_i \log(p_i), \tag{3.1}$$

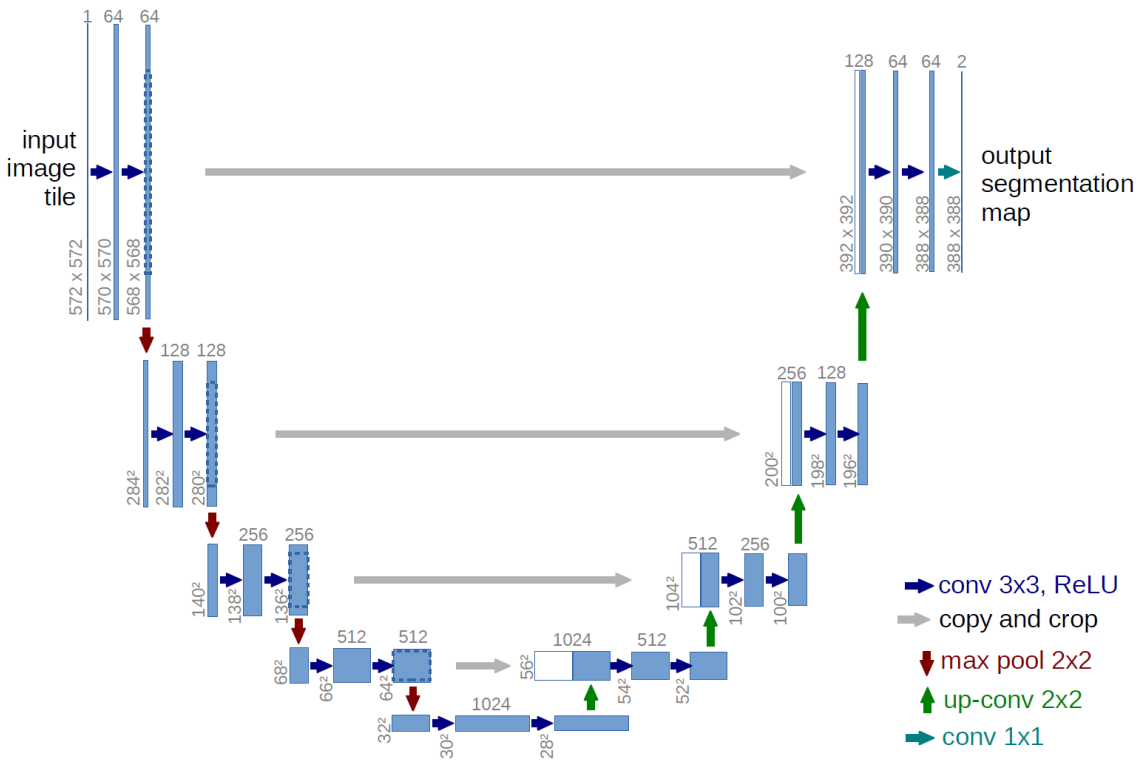
kde p představuje pravděpodobnost třídy a y je odpovídající hodnota z ground truth. Pokud děláme klasifikaci mezi více klasifikačními třídami můžeme použít upravenou chybovou funkci, která se nazývá Multi-Class Cross Entropy. Která se počítá pomocí následující rovnice

$$H(y,p) = -\frac{1}{n}\sum_{i=1}^{n}[y^{(i)}\log(p^{(i)}) + (1-y^{(i)})\log(1-p^{(i)})], \tag{3.2}$$

kde p představuje jako v předchozím případě pravděpodobnost třídy a y je odpovídající hodnota z ground truth.

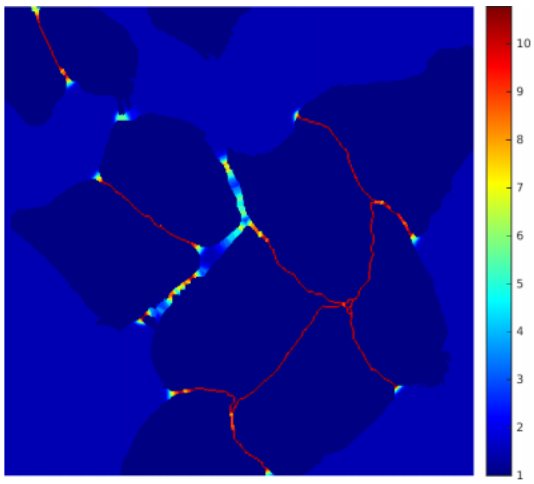
Váhování chybové funkce použijeme v případě, kdy chceme větší váhu na loss data pointu, který je v datové sadě vzácnější. V cross entropy vynásobíme loss každého pixelu příslušnou váhou, tzn. že musíme vyrobit "mapu vah". Toto využili např. v originálním článku o U-Netu [22] pro naučení hranic mezi jednotlivými buňkami. Pro trénovací data si vytvořili mapy vah, kdy v závislosti na velikosti hranic mezi buňkami určili hodnotu. Tím přinutili síť, aby dávala na malé hranice buněk co největší důraz a díky tomu nedošlo k ignorování této informace v průběhu učení. Na obrázku 3.3 lze vidět ukázku mapy vah.

Model neuronové sítě se trénuje ve 150 000 krocích s velikostí batche 8. Rozlišení vstupního obrázku dokumentu je 300x300 pixelů. Pro trénování modelu neuronové sítě jsem použil optimalizační algoritmus Adam [14] a batch normalizaci [11].



Obrázek 3.2: Architektura U-Net [22].

.



Obrázek 3.3: Ukázka mapy vah z článku [22]. Červená barva zobrazuje místa, kde mají váhy největší hodnotu, jedná se o nejužší hranice mezi buňkami.

3.3 Zpracování dokumentu

Potom co je natrénovaný model neuronové sítě je možné začít zpracovávat jednotlivé dokumenty. Jako první je potřeba načíst obrázek požadovaného dokumentu, který se má analyzovat. Jelikož je model naučený na zmenšených vstupech, je potřeba jako první vstupní obrázek zmenšit. Ve většině případech se jedná o zmenšení v konstantním poměru. V případě adaptativního rozlišení, na které se více zaměřuji v kapitole 5 je potřeba spočítat medián výšky řádků, který se využije pro výpočet hodnoty poměru zmenšení.

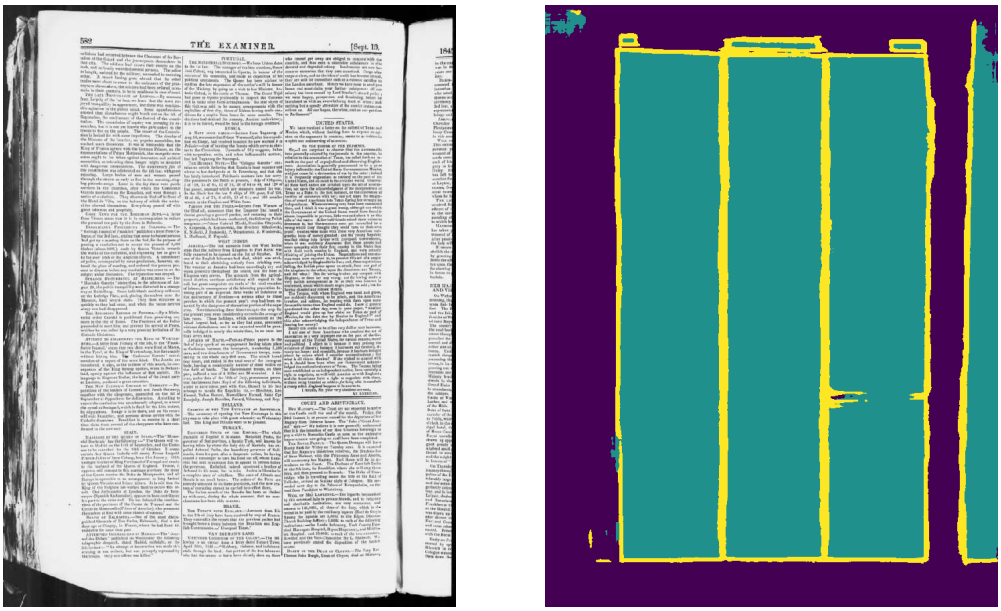
Jako další je potom na řadě poslání vstupního obrázku dokumentu do modelu neuronové sítě. Ta vrací na výstup jednotlivé pravděpodobnostní mapy klasifikačních tříd. Třídu pixelu určíme podle maxima pravděpodobnosti na výstupu neuronové sítě. Na obrázku 3.4 lze vidět vstupní obrázek dokumentu a k němu odpovídající segmentační mapu. Dalším potřebným krokem je zjistit ze segmentační mapy, které pixely označené klasifikační třídou pro odstavec jsou spojeny a tvoří tak spolu textový region. Pro řešení tohoto problému je možné použít hledání spojitých komponent. Vstupem pro hledání spojitých komponent je binární obrázek, ten se získá tak, že se vezme pravděpodobnostní mapa odstavců, od které se odečte pravděpodobnosti mapa hran. Hledání spojitých komponent určí, které pixely jsou propojeny, a tudíž náleží stejnému regionu a které naopak patří, do jiných regionů. Jakmile se ví, které pixely patří, do kterého regionu, může se přejít k získávání souřadnic regionů. Tato část se provádí pomocí hledání bounding boxů jednotlivých roztříděných regionů z minulého kroku. Potom co jsou nalezeny souřadnice všech regionů, může se přejít k důležité části a tou je post-processing, ten je detailněji popsán v následují podkapitole. Výsledek hledání spojitých komponent a určování souřadnic lze vidět na obrázku 3.5. Nakonec je možné získané souřadnice regionů uložit. Ty jsem se rozhodl ukládat ve formátu PAGE XML [21].

3.4 Post-processing

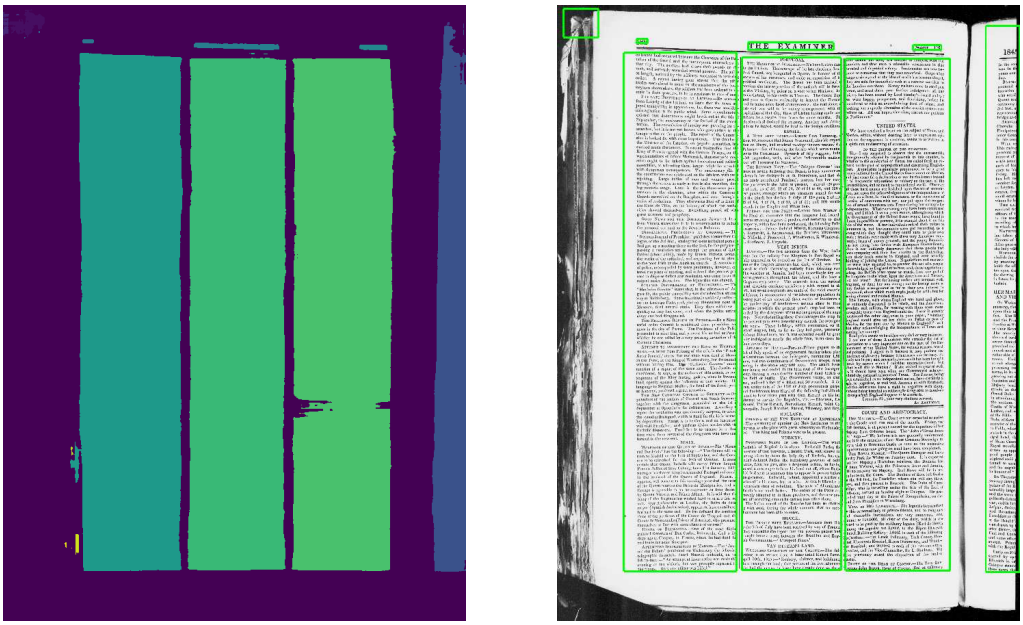
Jak bylo zmíněno v předchozí podkapitole, důležitou součástí zpracování dokumentu je post-processing. Ten přichází na řadu po získání souřadnic z nalezených regionů. Jelikož po hledání spojitých komponent vzniká velký počet malých odstavců je potřeba tyto artefakty filtrovat. To se dá docílit poměrně jednoduchým způsobem. Zda má odstavec zůstat nebo být zahozen rozhoduje to, jestli je šířka větší než předem zvolený práh. Po hledání spojitých komponent také vznikají odstavce, které jsou uvnitř jiného většího odstavce. Zde je řešení stejně jako v předchozím případě jednoduché. Stačí mezi sebou porovnávat jednotlivé souřadnice dvou odstavců. Pokud dojde k vyhodnocení, že je jeden odstavec uvnitř jiného, tak dojde k zahození vnitřního odstavce. Na obrázku 3.6 lze vidět vstup a výstup výše popsaného post-processingu.

3.5 Metrika úspěšnosti

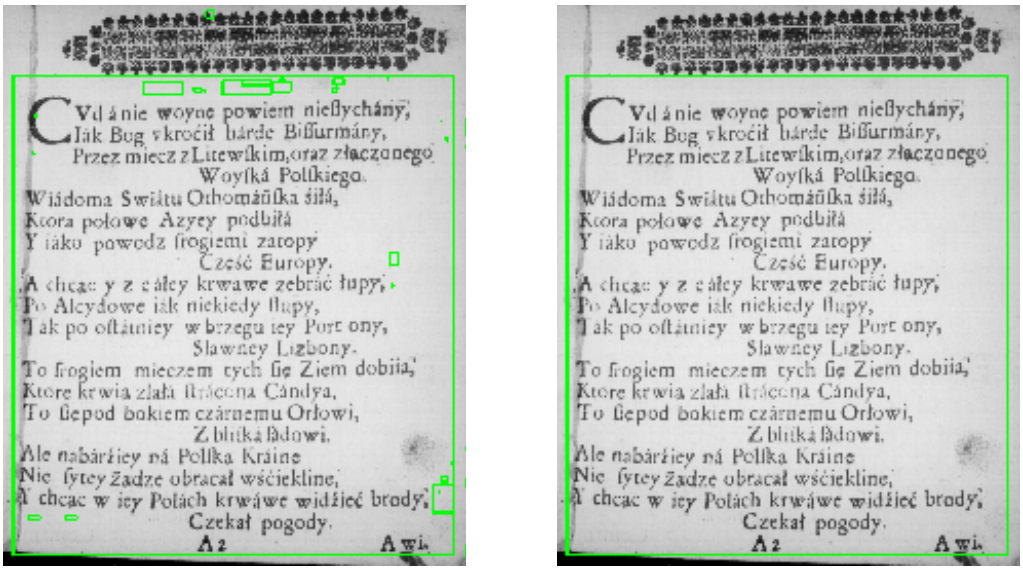
Důležitým krokem je ověření úspěšnosti experimentů. Pro moje experimenty jsem se rozhodl měřit úspěšnost predikce rozložení stránky pomocí objektové metriky Symmetric Best Dice (SBD) [24]. Tato metrika měří pro každý detekovaný objekt překryv s nejbližším ground truth objektem, a naopak pro každý ground truth objekt překryv s nejbližším detekovaným



Obrázek 3.4: První část zpracování dokumentu. Nalevo vstupní obrázek a napravo vytvořená segmentační mapa. Fialová barva značí pozadí tyrkysová odstavec a žlutá hrany odstavce.



Obrázek 3.5: Nalevo lze vidět výsledek po hledání spojitých komponent a napravo zelené obdélníky značí výsledné hranice nalezených odstavců.



Obrázek 3.6: Nalevo lze vidět stav před použitím post-processingu a napravo jsou vidět regiony po post-processingu.

objektem. Pro výpočet SBD se využívá metrika Best Dice (BD), která je definována jako:

$$BD(L^a, L^b) = \frac{1}{M} \sum_{i=1}^{M} \max_{1 \leq j \leq N} \frac{2|L_i^a \bigcap L_j^b|}{|L_i^a| + |L_j^b|}, \tag{3.3}$$

kde $|.|$ označuje plochu odstavce (počet pixelů). L_i^a pro $a \leq i \leq M$ a L_j^b pro $a \leq j \leq N$ jsou odstavce objektové segmentace ze sady odstavců L^a a L^b. M a N značí počet odstavců v těchhle sadách. SBD je definováno jako:

$$SBD(L^{ar}, L^{gt}) = \min \{BD(L^{ar}, L^{gt}), BD(L^{gt}, L^{ar})\}, \tag{3.4}$$

kde L^{gt} je ground truth a L^{ar} je výsledek algoritmického řešení.

3.6 Implementace

Pro tuto práci jsem se rozhodl použít programovací jazyk Python [23], knihovnu TensorFlow [1] pro implementaci modelu neuronové sítě. Podobných knihoven, které slouží pro zjednodušení implementace modelů neuronových sítí existuje celá řada. Mezi další nejznámější knihovny pro tvorbu modelů neuronových sítí patří např. knihovna Caffe [13], PyTorch [19] nebo Keras [4].

Kapitola 4

Datová sada

Tato kapitola se zabývá datovou sadou použitou pro natrénování modelů neuronové sítě a vyhodnocování jejich úspěšnosti. Jako první se zaměřím na datovou sadu historických dokumentů Impact dataset, kde popíši, jak daná datová sada vypadá, co obsahuje a jak je členěná. Potom popíši úpravy, které jsem použil na anotace datové sady ke zlepšení učení neuronové sítě a přípravu dat pro neuronovou síť.

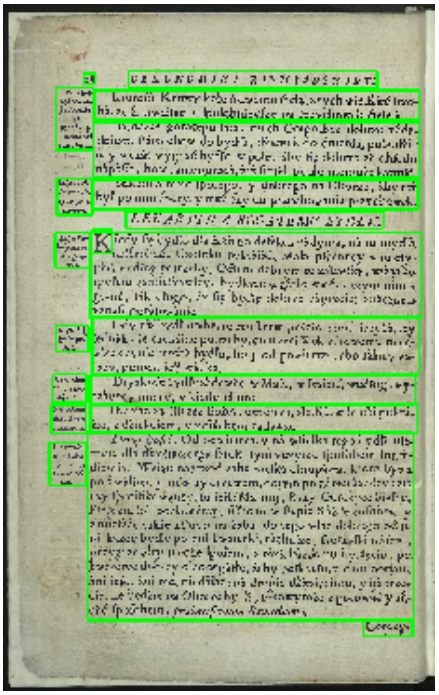
4.1 Impact dataset

Pro mou práci jsem se rozhodl zvolit datovou sadu historických dokumentů Impact dataset [18]. Tento dataset obsahuje přes 600 000 obrázků dokumentů z hlavních evropských knihoven. Ke zhruba 45 000 obrázkům těchto dat existuje přepis obsahující informace o rozložení odstavců a přepis textu. Všechny dokumenty jsou naskenovány v dobré kvalitě ve vysokém rozlišení. Anotace regionů jsou ve formě souřadnic obsaženy v PAGE XML souborech [21]. Na obrázku 4.1 můžete vidět ukázkovou stránku z datasetu.

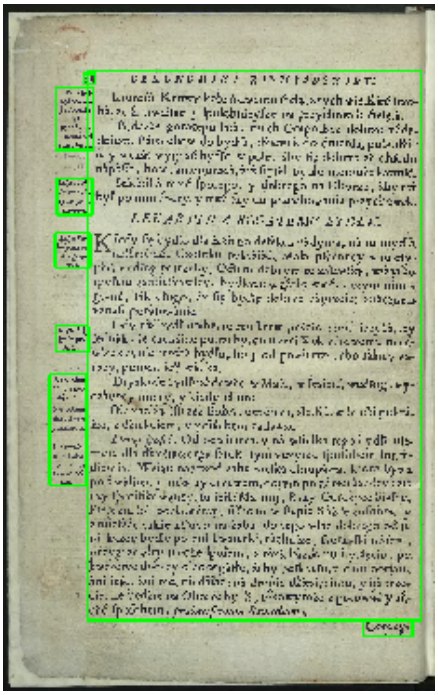
Dataset je rozdělen do 172 knih, tyto knihy jsem se rozhodl rozdělit na trénovací a testovací sadu. Dvě třetiny knih jsem určil jako trénovací sadu a zbytek knih jako testovací. Jelikož je velký nepoměr v počtu stránek jednotlivých knih, hlavně jsou velké rozdíly mezi knihami s jednoduchým rozložením stránek (jedno-sloupcové) a těmi složitějšími. Noviny většinou obsahují pouze pár stránek, a proto kdyby nedošlo k výběru podmnožiny stran daných knih, tak by v datasetu byl velký nepoměr a neuronová síť by se učila téměř celou dobou na jednoduchých rozloženích stránek. Z toho důvodu jsem se rozhodl z každé knihy použít pouze náhodných dvacet stránek.

4.2 Úpravy datové sady a příprava dat

Jelikož jednou z aplikací analýzy rozložení stran je správné seřazení detekovaných řádků, tak jsem se z důvodu nekonzistentních anotací a nejasné definici horizontální hranice omezil na detekci souvislých sloupců textu. Z tohoto důvodu jsem se rozhodl odstavce, které jsou těsně nad sebou spojit. Toto spojování z důvodu komplikovaných layoutů není vždy dokonalé a v datové sadě tím pádem vznikají artefakty. Také jelikož anotace v datové sadě nebyli konzistentní, občas byly regiony obtaženy pomocí polygonů natěsno, a naopak někdy jednoduše pomocí bouding boxu, kdy velká část byla prázdná, rozhodl jsem se anotace sjednotit a vše jsem upravil, aby byly anotace ve formě bounding boxů. Výsledek těchto úprav lze vidět na obrázku 4.2. Tyto úpravy zjednodušují učení neuronové sítě, důležitost



Obrázek 4.1: Ukázka stránky z Impact datasetu. Zelené polygony značí anotované textové regiony.



Obrázek 4.2: Ukázka stránky z Impact datasetu po úpravě anotací. Zelené polygony značí anotované textové regiony.

sjednocování je hlavně v případě experimentu, kde se provádí segmentace do tří tříd viz kapitola experimenty. Poslední úpravou datové sady bylo přidání pozic a výšky řádků do anotací datové sady.

Kapitola 5

Experimenty

V kapitole jsou popsány experimenty, které proběhly v rámci této práce. Jako první dojde popsání experimentů, které se týkají formulace úlohy. Experimenty s formulací zahrnují:

- Binární segmentaci
- Segmentaci rozšířenou o detekci hran
- Segmentaci, kde se používají pouze vertikální hrany odstavců

V další části kapitoly dojde k experimentům se zlepšením trénování neuronové sítě. Tyhle experimenty zahrnují:

- Vzorkování batche
- Adaptativní rozlišení

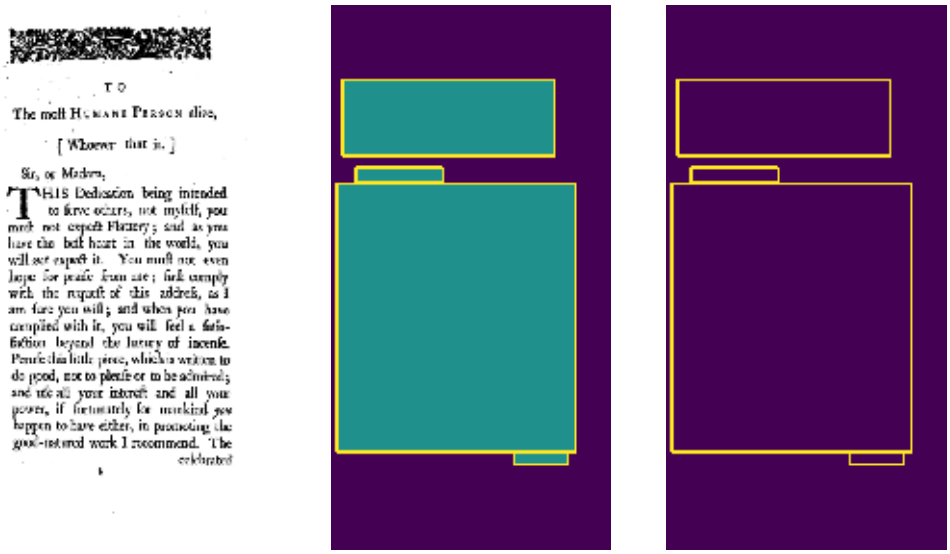
Na konci kapitoly je popsána aplikace vertikálních separátorů.

5.1 Formulace úlohy

Pro trénování neuronové sítě je potřeba mít připravená data. Jelikož jsou data v datové sadě velká, bylo je potřeba zmenšit. Pro většinu experimentů se jednalo o konstantní zmenšení, kde měřítko zmenšení bylo zvoleno tak, aby se do vstupu do sítě, které má 300 pixelů na šířku a 300 pixelů na výšku vešlo co největší plocha dokumentu. Pro rychlý a efektivní výpočet chybové funkce je vhodné mít předem vygenerovanou ground truth masku anotací a vah. Ukázku těchto vstupních dat lze vidět na obrázku 5.1.

Jako první experiment jsem se rozhodl natrénovat model pro řešení binární segmentace dokumentu, kdy se jednotlivým pixelům obrázku dokumentu přiřazují klasifikační třídy odstavec a pozadí. Na obrázku 5.2 lze vidět vstup do sítě, segmentační mapa a výsledek zpracování dokumentu.

Pro zlepšení zpracování více sloupcových dokumentů jsem se rozhodl přidat klasifikační třídu reprezentující hrany odstavců. Tato úprava by měla v případě více sloupcových dokumentů zabraňovat slévání sousedních odstavců. Pro správné natrénování modelu na detekci hran bylo potřeba dodat chybové funkci mapu vah. Pixely hran mají v masce vah 10x větší váhu než ostatní pixely. Pokud by mapa vah nebyla dodaná došlo by k tomu, že by se model nezaměřoval na hrany a došlo by tak k ignorování těchto informací. Na obrázku 5.3 lze vidět vstupní obrázek, segmentační mapa a výsledek zpracování dokumentu.



Obrázek 5.1: Na obrázku lze vidět připravená data pro neuronovou síť. Nalevo lze vidět vstupní obrázek, který jde na vstup do sítě. Potom lze vidět vygenerovanou ground truth a mapu vah.

Obrázek 5.2: Ukázka výsledku experimentu binární segmentace. Jako první lze vidět vstupní obrázek, následuje vzniklá segmentační mapa a výstup zpracování dokumentu.

Obrázek 5.3: Ukázka výsledku experimentu, kdy se model učí navíc i hrany odstavců. Jako první lze vidět vstupní obrázek, následuje vzniklá segmentační mapa a výstup zpracování dokumentu.

Dalším experimentem bylo učení pouze vertikálních hran odstavců. Tím by mohlo dojít k zjednodušení učení vertikálních hran odstavců. Tyhle hrany jsou pro analýzu mnohem důležitější, jelikož hlavním cílem je rozdělit od sebe sousední odstavce. Pokud dojde ke spojení odstavců, které jsou nad sebou, tak se nejedná o takový problém, jako spojení sousedních odstavců. Na obrázku 5.4 lze vidět vstupní obrázek, segmentační mapa a výsledek zpracování dokumentu.

Experiment	**Úspěšnost**
Binární segmentace	0,39463
Všechny hrany odstavců	0,50939
Pouze vertikální hrany	0,47875

Tabulka 5.1: Výsledky experimentů pro model se třemi klasifikačními třídami. Metrikou úspěšnosti je zde SBD.

Z výsledků experimentů plyne, že přidání třetí klasifikační třídy pro hrany odstavce pomáhají při analýze, zatímco zahození horizontálních hranic nepomáhá.

5.2 Trénovací strategie

Pro zlepšení trénovaní modelu neuronové sítě jsem se rozhodl využít dvě úpravy strategie trénování. První z nich je vzorkování batche. To spočívá ve zlepšení výběru obsahu batche. Batch se pomocí této strategie skládá ze dvou částí. První polovina batche je vybrána náhodně, aby nedocházelo k přetrénování modelu a mohlo se tak dostat na všechny data z datové sady. Druhá část se vybírá pomocí pravděpodobnosti výběru. Tyto pravděpodobnosti se počítají pomocí skóre, které se při přípravě dat určuje pro každý snímek v datové sadě. Skóre je určeno v závislosti na počtu odstavců a velikostí plochy kterou odstavce zabírají. Tím dojde při výběru k preferování obrázků dokumentu, které mají více menších odstavců, ale zároveň odstavce zabírají podstatnou část stránky, aby nedošlo k vybrání stránky, která

Obrázek 5.4: Experiment, kdy se model učí pouze vertikální hrany

obsahuje malý textový region a zbytek stránky je prázdný. Skóre lze vypočítat následovně:

$$score = \begin{cases} c/S, & \text{pokud } S > (0.3 * (w * h)) \\ 0 & jinak, \end{cases}$$

kde S značí velikost plochy odstavců, c je počet odstavců, w a h značí šířku a výšku obrázku dokumentu. Potom co se určí skóre pro všechny data v datové sadě, tak dojde k výpočtu pravděpodobnosti, že se obrázek použije při vybírání batche. Tato pravděpodobnost se vypočítá následovně:

$$p_{(a)} = \frac{score_{(a)}}{score_{sum}}, \tag{5.1}$$

kde $p_{(a)}$ je pravděpodobnost výběru obrázku do batche, $score_a$ značí skóre obrázku a $score_{sum}$ je suma skóre všech obrázků v datasetu.

Experiment	**Vzorkování batche**	**Úspěšnost**
Všechny hrany odstavců	Ano	0,50939
	Ne	0,51045
Pouze vertikální hrany	Ano	0,47875
	Ne	0,49373

Tabulka 5.2: Výsledky experimentu vzorkování batche. Metrikou úspěšnosti je zde SBD.

Druhou trénovací strategií je adaptativní rozlišení. V tomhle experimentu jde o sjednocení rozlišení podle velikosti textu. Pro natrénování modelu je zapotřebí si správně přichystat data. Tentokrát místo konstantního zmenšení je potřeba změnit rozlišení vstupních dat podle vypočteného poměru. Z výšek řádků, které jsou součástí anotací v datasetu se následně vypočítá medián, který se použije pro výpočet hodnoty poměru pomocí rovnice:

$$s = m * B \tag{5.2}$$

kde m představuje medián výšky řádků a B reprezentuje základní poměr, který lze zvolit libovolně. Pokud je medián výšky řádků nula, což nastává v případech, kdy stránka neobsahuje žádný text, tak se jako poměr zmenšení zvolí právě základní poměr B. V případě adaptativního rozlišení se musí lehce upravit postup zpracování dokumentu, kdy je potřeba po načtení obrázku dokumentu určit poměr zmenšení stránky. Dále už zpracování dokumentu pokračuje stejně, jak to bylo popsáno v kapitole 3.

Experiment	**Vzorkování batche**	**Úspěšnost**
Adaptativní rozlišení se všemi hranami odstavců	Ano	0,51215
	Ne	0,53761
Adaptativní rozlišení pouze s vertikálními hranami	Ano	0,50056
	Ne	0,51423

Tabulka 5.3: Výsledky experimentu adaptativního rozlišení. Metrikou úspěšnosti je zde SBD.

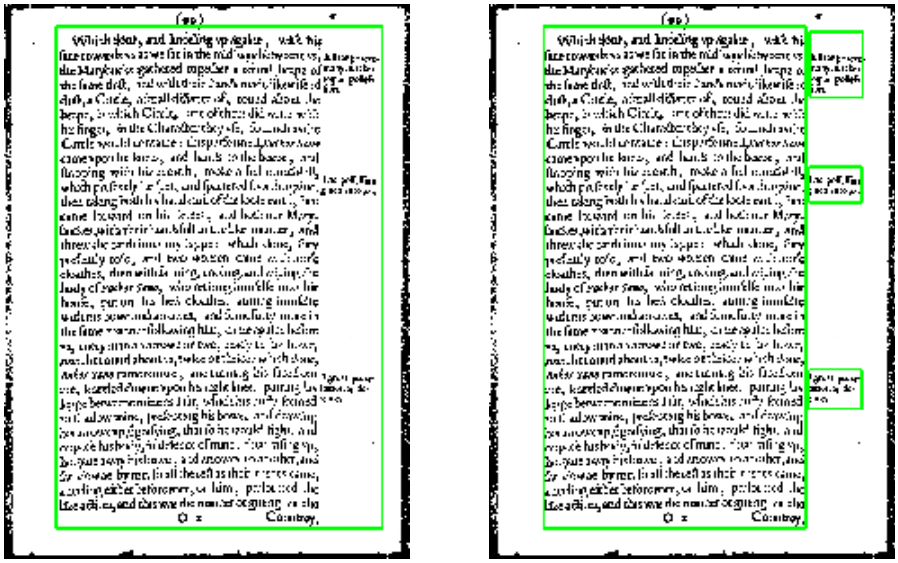
Z experimentů plyne, že při trénování se vyplatí používat jak vzorkování, tak i adaptativní rozlišení.

5.3 Aplikace separátorů

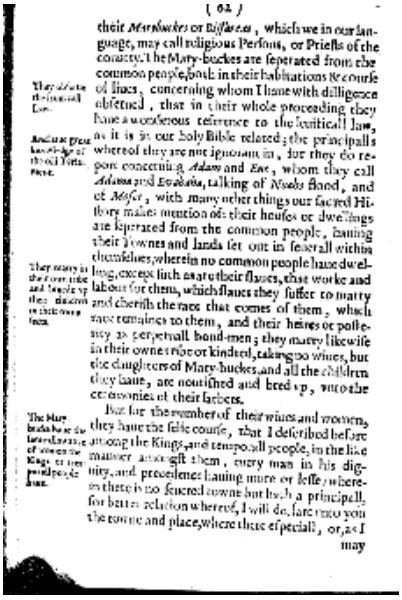
Možným post-processingem je hledání a aplikování vertikálních separátorů. Tento post-processing zlepšuje oddělení vedlejších odstavců, jak jde vidět nalevo na obrázku 5.5. V případě, kdy využijeme aplikaci vertikálních separátorů, získáme výsledek, který můžeme vidět napravo na obrázku 5.5. Hlavní motivací zavedení hledání separátorů je analýza novinových dokumentů. Vzhledem k jejich komplexnosti je zde nejpravděpodobnější, že dojde ke slévání vedlejších odstavců.

Jako první je potřeba provést rotaci obrázku dokumentu, aby byly řádky dokumentu vodorovně. Pro zjištění úhlu, o který se má obrázek rotovat je potřeba znát souřadnice řádků dokumentu. Ty jsou pro účely testování obsaženy v datové sadě. Ze souřadnic řádků lze vypočítat úhel sklonu a velikost jednotlivých řádků. Výsledný úhel rotace se určí, jako průměr úhlů sklonu 50% nejdelších řádků. Alternativně by šlo odhadnout z detekcí hran. Na obrázku 5.6 lze vidět dokument, který je naskenován pod špatným úhlem a výsledek po provedení rotace.

Navržená metoda předpokládá tří labelovou klasifikaci. Ideální je zde model, který hledá pouze vertikální hrany odstavců. Ze segmentační masky je potřeba vytvořit dvě vertikální sumační projekce do 1D. Jedna, která reprezentuje klasifikační třídu hran odstavců a druhá třídu odstavců. Projekce se vytváří pomocí spočítání počtu pixelů, které náleží dané klasifikační třídě. Následuje detekce souřadnic, na kterých dominuje třída hrana, pomocí projekcí. Jelikož dojde k nalezení spousty pozic, které jsou blízko sebe je zapotřebí tyto pozice filtrovat. K tomu slouží Non-Maximum Suppression [10]. Pomocí této metody se dají nalézt nejlepší lokální separátory. Jakmile jsou známy pozice separátorů dojde k aplikaci separátorů do segmentační mapy. Jelikož jsou separátory hledány v rotované verzi je potřeba tuto rotaci také zohlednit při aplikaci do segmentační mapy. Na obrázku 5.7 lze vidět segmentační mapa před a po aplikaci separátorů, zde lze vidět, že by došlo ke slévání bočních poznámek jak je zobrazeno nalevo na obrázků 5.5. Díky aplikaci separátorů dojde k rozdělení hlavního odstavce od bočních poznámek, jak lze vidět napravo na obrázku 5.5. Tato aplikace má ale i negativní efekt jelikož separátor se aplikuje na celou výšku obrázku, tak



Obrázek 5.5: Výsledek zpracování dokumentu. Nalevo bez použití separátorů a napravo s použitím separátorů. Zde díky separátorům dojde k rozdělení hlavního odstavce od bočních poznámek.



(62)

their *Marabuckes* or *Bissareas*, which we in our language, may call religious Persons, or Priests of the country. The Mary-buckes are seperated from the common people, both in their habitations & course of lives, concerning whom I have with dilligence obserued, that in their whole proceeding they have a wonderous reference to the leuitical law, as it is in our holy Bible related; the principalls whereof they are not ignorant in, for they do report concerning *Adam* and *Eue*, whom they call *Adama* and *Euahda*, talking of *Noahs* flood, and of *Moyses*, with many other things our sacred History makes mention of: their houses or dwellings are seperated from the common people, hauing their Townes and lands set out in seuerall within themselues, wherein no common people haue dwelling, except such as are their slaues, that worke and labour for them, which slaues they suffer to marry and cherish the race that comes of them, which race remaines to them, and their heires or posterity as perpetuall bond-men; they marry likewise in their owne tribe or kindred, taking no wiues, but the daughters of Mary-buckes, and all the children they haue, are nourished and bred vp, vnto the ceremonies of their fathers.

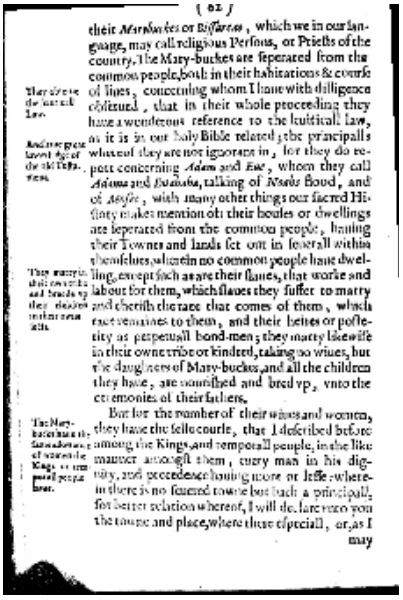
But for the number of their wiues and women, they haue the selfe course, that I described before among the Kings, and temporall people, in the like manner amongst them, every man in his dignity, and precedence hauing more or lesse, wherein there is no seuered towne but hath a principall, for better relation whereof, I will declare vnto you the towne and place, where these especiall, or, as I may

They obserue the leuitical Law.

And are great learned skill of the old Testament.

They marry in their owne tribe and bands vp their daughters in their owne tribe.

The Mary-buckes haue the like gradation of women the Kings, or temporall people haue.



(62)

their *Marabuckes* or *Bissareas*, which we in our language, may call religious Persons, or Priests of the country. The Mary-buckes are seperated from the common people, both in their habitations & course of lives, concerning whom I have with dilligence obserued, that in their whole proceeding they have a wonderous reference to the leuitical law, as it is in our holy Bible related; the principalls whereof they are not ignorant in, for they do report concerning *Adam* and *Eue*, whom they call *Adama* and *Euahda*, talking of *Noahs* flood, and of *Moyses*, with many other things our sacred History makes mention of: their houses or dwellings are seperated from the common people, hauing their Townes and lands set out in seuerall within themselues, wherein no common people haue dwelling, except such as are their slaues, that worke and labour for them, which slaues they suffer to marry and cherish the race that comes of them, which race remaines to them, and their heires or posterity as perpetuall bond-men; they marry likewise in their owne tribe or kindred, taking no wiues, but the daughters of Mary-buckes, and all the children they haue, are nourished and bred vp, vnto the ceremonies of their fathers.

But for the number of their wiues and women, they haue the selfe course, that I described before among the Kings, and temporall people, in the like manner amongst them, every man in his dignity, and precedence hauing more or lesse, wherein there is no seuered towne but hath a principall, for better relation whereof, I will declare vnto you the towne and place, where these especiall, or, as I may

They obserue the leuitical Law.

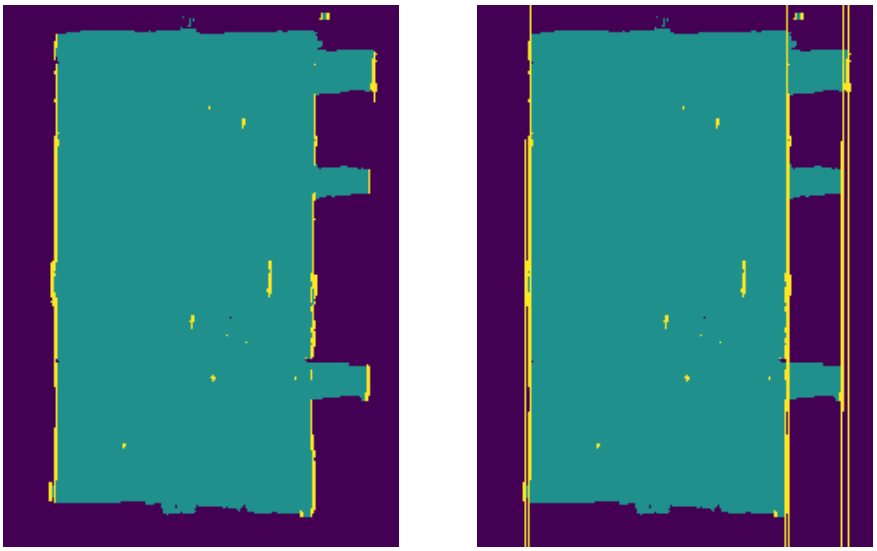
And are great learned skill of the old Testament.

They marry in their owne tribe and bands vp their daughters in their owne tribe.

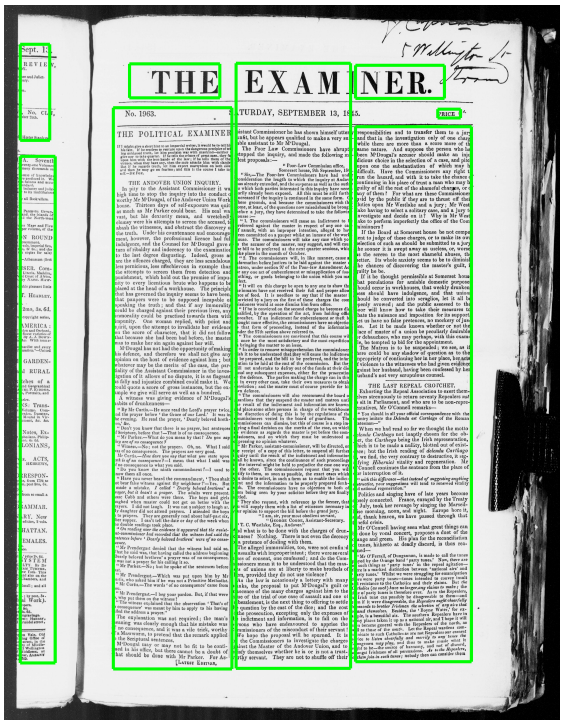
The Mary-buckes haue the like gradation of women the Kings, or temporall people haue.

Obrázek 5.6: Vstupní obrázek před a po provedení rotace podle sklonu detekovaných řádků.

může dojít k rozdělení odstavce, který je například přes celou šířku stránky, jak to jde vidět na obrázku 5.8.



Obrázek 5.7: Segmentační mapa před a po aplikaci vertikálních separátorů.



Obrázek 5.8: Výsledek zpracování novinového článku, kde dochází kvůli aplikaci separátorů k chybnému rozdělení nadpisu na části.

Kapitola 6

Diskuze

Z výsledků experimentů vyplývá, že model neuronové sítě, který se trénuje se všemi hranami odstavce dosahuje větší úspěšnosti než model, který je trénován pouze s vertikálními hranami odstavce a model, který je natrénován na binární segmentaci. Dále z experimentů v sekci 5.2 trénovací strategie vyplývá, že jak vzorkování batche, tak adaptativní rozlišení zlepšilo úspěšnost modelu neuronové sítě. V experimentu na aplikaci separátorů lze vidět, že je tenhle post-processing v mnoha případech vícesloupcových dokumentů užitečný. Má ale také negativní efekty, kdy může kvůli aplikaci po celé výšce dokumentu dojít k rozdělování odstavců. Zde se nabízí úprava, kdy by po nalezení souřadnic vertikálních separátorů docházelo ke zkoumání, ve kterých částech se má separátor aplikovat. To lze zjistit pomocí horizontálních sumačních projekcí, kde by se zkoumalo okolí separátoru. Zde by bylo rozhodující, jestli v dané části dochází k přechodu. Pokud by například vlevo od separátoru v dané části dominoval odstavec a napravo pozadí, tak by došlo k aplikaci separátoru.

Jednou z problémových části práce byla metrika SBD. Ukázalo se, že metrika SBD i přesto, že je běžně používaná v literatuře není pro tuto práci zcela ideální, jelikož dochází ke zkreslení. Tato metrika trpí problémem, že i přes to, že výsledek analýzy vypadá na pohled zcela správně dochází k nízkému ohodnocení. To se děje např. kvůli spojení odstavců, které jsou nad sebou, a i po úpravě datové sady nebyly spojeny. Z tohoto důvodu je například složité rozhodnout, jestli je výhodnější detekce všech hran odstavců nebo jen vertikálních. Kvůli spojení odstavců, které jsou nad sebou dochází ke zhoršení výsledku měření, ale ve skutečnosti toto spojení odstavců v mé úloze nedělá problémy. Pro moji úlohu by se dala tato metrika zaměnit za řádkovou metriku. V téhle metrice by se zkoumalo, jak moc informace o rozložení strany dokumentu pomohly nebo naopak zhoršily post-processingem nalezené řádky dokumentu.

Další možnou formou post-processingu by bylo použití morfologických operací dilatace a eroze.

Kapitola 7

Závěr

Cílem práce bylo vytvořit nástroj, který analyzuje vstupní obrázek dokumentu a vrací informace o rozložení strany textového dokumentu. V kapitole 2 jsem jako první popsal problematiku a rozdělení metod pro analýzu dokumentů. Byly zde popsány čtyři aktuální metody založené na hledání vzorů, konvolučních neuronových sítí a bayesovských modelech.

V další kapitole 3 jsem popsal můj návrh řešení nastoleného cíle. Zde jsem na začátku popsal architekturu modelu neuronové sítě U-Net a chybovou funkci cross entropy, které jsem se rozhodl použít při implementaci řešení. Dále jsem v kapitole popsal postup zpracování dokumentu včetně použitého post-processingu, metriku úspěšnosti SBD pomocí které jsem vyhodnocoval výsledky experimentů a nástroje pomocí kterých jsem implementoval své řešení. Trénování sítě probíhá v 150 000 krocích při velikosti batche 8. Vstup do sítě má rozlišení 300x300 pixelů a při trénování se používá optimalizační algoritmus Adam a batch normalizace.

Kapitola 4 byla zaměřena na popis vybrané datové sady Impact dataset, která byla použita pro trénování modelu a vyhodnocování úspěšnosti experimentů. Zde byly popsány i úpravy anotací vybrané datové sady, které byly v rámci práce aplikovány. Jednalo se o spojení odstavců, které jsou nad sebou, sjednocení anotací odstavců pomocí bounding boxů a přidání pozic a výšky řádků.

Kapitola 5 popisovala experimenty, které byly v rámci práce uskutečněny. Jako první zde byly představeny základní experimenty, které formulovali úlohu. Zde se dosáhlo nejlepších výsledků při segmentaci s použitím všech hran odstavců. V další části kapitoly došlo k popisu experimentů, které byly zaměřeny na zlepšení trénování modelu neuronové sítě. Zde výsledky experimentů ukázali, že se vyplatí při učení používat jak vzorkování batche, tak i adaptativní rozlišení. V poslední části kapitoly došlo na popis aplikace separátorů. Které se ukázalo, jako užitečné v případě dokumentů s více sloupci.

V kapitole 6 došlo k shrnutí výsledků experimentů, tak k diskuzi o možných vylepšeních této práce. Byla zde i diskutována změna aktuální metriky úspěšnosti za řádkovou metriku.

V rámci další práce by mohlo dojít k rozšíření klasifikačních tříd např. o tabulky, obrázky. Další rozšíření by mohla být sémantická analýza textového dokumentu nebo zlepšení post-processingu.

Literatura

[1] Abadi, M.; Agarwal, A.; Barham, P.; aj.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, software available from tensorflow.org. URL http://tensorflow.org/

[2] and: Parameter-free geometric document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 23, č. 11, Nov 2001: s. 1240–1256, ISSN 0162-8828, doi:10.1109/34.969115.

[3] Ares Oliveira, S.; Seguin, B.; Kaplan, F.: dhSegment: A generic deep-learning approach for document segmentation. *arXiv e-prints*, Apr 2018: arXiv:1804.10371, 1804.10371.

[4] Chollet, F.; aj.: Keras. https://keras.io, 2015.

[5] Cortes, C.; Mohri, M.; Rostamizadeh, A.: L2 Regularization for Learning Kernels. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, Arlington, Virginia, United States: AUAI Press, 2009, ISBN 978-0-9749039-5-8, s. 109–116.

[6] Deng, J.; Dong, W.; Socher, R.; aj.: ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] Dong, H.; Yang, G.; Liu, F.; aj.: Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. *CoRR*, ročník abs/1705.03820, 2017, 1705.03820.

[8] Dong, J.; Wang, W.; Tan, T.; aj.: Run-Length and Edge Statistics Based Approach for Image Splicing Detection. In *Digital Watermarking*, editace H.-J. Kim; S. Katzenbeisser; A. T. S. Ho, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ISBN 978-3-642-04438-0, s. 76–87.

[9] He, K.; Zhang, X.; Ren, S.; aj.: Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016, doi:10.1109/cvpr.2016.90.

[10] Hosang, J. H.; Benenson, R.; Schiele, B.: Learning non-maximum suppression. *CoRR*, ročník abs/1705.02950, 2017, 1705.02950.

[11] Ioffe, S.; Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015, s. 448–456.

[12] Janocha, K.; Czarnecki, W. M.: On Loss Functions for Deep Neural Networks in Classification. *CoRR*, ročník abs/1702.05659, 2017, 1702.05659.

[13] Jia, Y.; Shelhamer, E.; Donahue, J.; aj.: Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, New York, NY, USA: ACM, 2014, ISBN 978-1-4503-3063-3, s. 675–678, doi:10.1145/2647868.2654889.

[14] Kingma, D. P.; Ba, J.: Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[15] Kundu, A.; Vineet, V.; Koltun, V.: Feature Space Optimization for Semantic Video Segmentation. In *CVPR*, 2016.

[16] Le, V. P.; Nayef, N.; Visani, M.; aj.: Text and non-text segmentation based on connected component features. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2015, s. 1096–1100, doi:10.1109/ICDAR.2015.7333930.

[17] Mitchell, P. E.; Yan, H.: Document page segmentation and layout analysis using soft ordering. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, ročník 1, Sep. 2000, ISSN 1051-4651, s. 458–461 vol.1, doi:10.1109/ICPR.2000.905375.

[18] Papadopoulos, C.; Pletschacher, S.; Clausner, C.; aj.: The IMPACT Dataset of Historical Document Images. In *Proceedings of the 2Nd International Workshop on Historical Document Imaging and Processing*, HIP '13, New York, NY, USA: ACM, 2013, ISBN 978-1-4503-2115-0, s. 123–130, doi:10.1145/2501115.2501130.

[19] Paszke, A.; Gross, S.; Chintala, S.; aj.: Automatic differentiation in PyTorch. In *NIPS-W*, 2017.

[20] Perez, L.; Wang, J.: The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *CoRR*, ročník abs/1712.04621, 2017, 1712.04621.

[21] Pletschacher, S.; Antonacopoulos, A.: The PAGE (Page Analysis and Ground-truth Elements) format framework. 08 2010, s. 257–260, doi:10.1109/ICPR.2010.72.

[22] Ronneberger, O.; Fischer, P.; Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, ročník abs/1505.04597, 2015, 1505.04597.

[23] Rossum, G.: Python Reference Manual. Technická zpráva, Amsterdam, The Netherlands, The Netherlands, 1995.

[24] Scharr, H.; Minervini, M.; French, A. P.; aj.: Leaf Segmentation in Plant Phenotyping: A Collation Study. *Mach. Vision Appl.*, ročník 27, č. 4, Květen 2016: s. 585–606, ISSN 0932-8092, doi:10.1007/s00138-015-0737-3.

[25] Sfikas, G.; Louloudis, G.; Stamatopoulos, N.; aj.: Bayesian Mixture Models on Connected Components for Newspaper Article Segmentation. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, New York, NY, USA: ACM, 2016, ISBN 978-1-4503-4438-8, s. 143–146, doi:10.1145/2960811.2967165.

[26] Viana, M. P.; Oliveira, D. A. B.: Fast CNN-Based Document Layout Analysis. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017: s. 1173–1180.

Příloha A

Obsah DVD

Obsah přiloženého DVD:

- **data/** - adresář datové sady
- **source/** - adresář se zdrojovými kódy
- **text/** - adresář s textovou částí práce
- **video/** - adresář s videem