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

Diplomová práce

Segmentace stran rukopisných dokumentů

Plzeň 2020 Bc. Josef Baloun

Místo této strany bude zadání práce.

Prohlášení

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

V Plzni dne 19. května 2020

Bc. Josef Baloun

Poděkování

Děkuji *doc. Ing. Pavlu Královi, Ph.D* za cenné rady, věcné připomínky a vstřícnost při konzultacích. Děkuji také projektu *Bavorsko-česká síť digitálních historických pramenů (Porta fontium)* za poskytnutí dat potřebných pro vypracování této práce.

Abstract

Page layout analysis plays an important role in the process of document retrieval. It can still be a challenging task for historical handwritten documents due to their diverse structure and possible quality degradation. In this thesis, an overview of possible methods for solving this problem is presented and a dataset composed of the pages of handwritten chronicles is created. This thesis also presents a prototype of the system for page layout analysis. The segmentation and classification into text, image and background classes are solved as a pixel-labeling problem. The prototype is based on a fully convolutional neural network inspired by U-Net. The best results were achieved when the prototype was set to the processing of entire pages of documents, the loss function was weighted and the training set was automatically augmented.

Abstrakt

Analýza stran dokumentů hraje významnou roli v procesu jejich elektronického zpřístupnění. Dokonce i v současné době může představovat nelehkou výzvu pro historické ručně psané dokumenty vzhledem k jejich různorodé struktuře a možné degradaci kvality. V rámci této práce je vypracován přehled možných metod pro řešení tohoto problému a vytvořena datová sada složená ze stran ručně psaných kronik. Dále je navržen prototyp systému pro analýzu stran dokumentů. Segmentace a klasifikace do tříd text, obrázek a pozadí jsou řešeny označením každého obrazového bodu strany dokumentu vhodnou třídou. Základem prototypu je plně konvoluční neuronová síť založená na síti U-Net. Nejlepších výsledků bylo dosaženo s prototypem, pro který bylo nastaveno zpracování celých stran dokumentů, bylo provedeno váhování chybové funkce a byla automaticky rozšířena trénovací množina.

Obsah

1 Úvod

V současné době je patrná značná snaha o digitalizaci a elektronické zpřístupnění dokumentů ve většině oblastí. Této snahy se týká i projekt *Bavorsko-česká síť digitálních historických pramenů* [21], jehož cílem je za pomoci rozsáhlé digitalizace a webové prezentace spojit do jednoho virtuálního celku v minulosti násilně roztržené archiválie státních archivů České republiky a Bavorska.

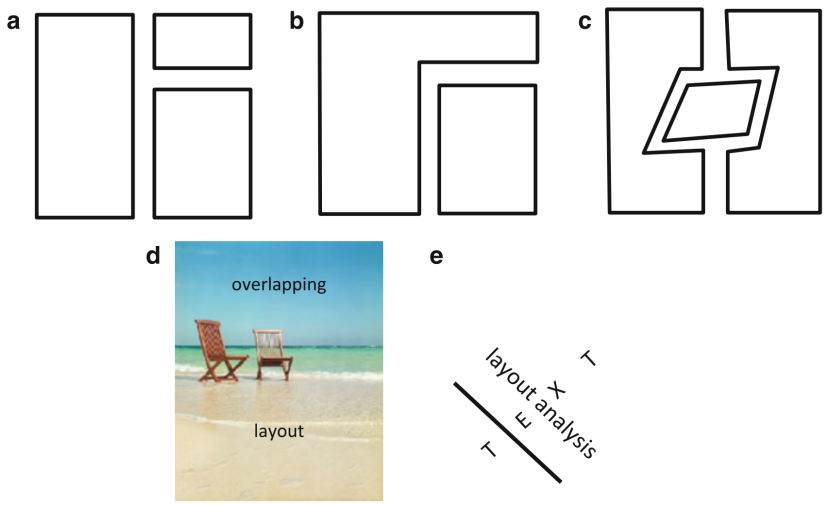
Základním krokem k elektronickému zpřístupnění je naskenování stran dokumentů. Dalším krokem může být analýza stran dokumentů (angl. *page layout analysis*), která se skládá ze segmentace strany dokumentu na homogenní komponenty a jejich klasifikace např. na bloky textu a obrázky. Dále může být na textové bloky aplikováno optické rozpoznávání znaků (*OCR*) a jejich převod do strojově čitelné podoby. Původní stránka dokumentu tak bude převedena do značkované, strojově čitelné podoby umožňující efektivní vyhledávání v jejím obsahu. Je snadné si představit, kolik práce by to ušetřilo historikům. V současné době je pro moderní tištěné dokumenty tento proces na velice dobré úrovni, ale problém nastává u starších a ručně psaných dokumentů vzhledem k jejich různorodé struktuře a degradované kvalitě.

Tato práce se týká zmiňované analýzy stran ručně psaných dokumentů, které mohou vhodně reprezentovat např. kroniky z 19. století. První část práce popisuje úvod do problému segmentace. Je zde uveden přehled metod, kterých je možné využít při řešení segmentace stran rukopisných dokumentů (rozdělení na text, obrázky a pozadí). Následuje seznámení s historickým portálem Porta fontium, poskytnutými daty a dalšími dostupnými datovými sadami pro segmentaci stran. Další část se týká vytvoření nové datové sady založené na stranách kronik poskytnutých z portálu Porta fontium. Následuje část týkající se popisu prototypu systému pro segmentaci stran. V této části je představen navržený systém pro označení každého obrazového bodu strany dokumentu příslušnými třídami (angl. *pixel-labeling*), architektura použité plně konvoluční neuronové sítě, postup vyhodnocení, výběr provedených experimentů, dosažené výsledky a jejich zhodnocení včetně dalších možných rozšíření.

2 Možnosti segmentace stran

Segmentací stran se rozumí úloha extrahování homogenních komponent z obrázku stránky dokumentu. Homogenní komponenty mohou představovat např. textové bloky, řádky textu, tabulky a obrázky. Úloha segmentace stran nezahrnuje klasifikaci komponent, ale je důležité pochopit, že tyto úlohy nelze oddělit. Pro obě tyto úlohy existuje anglický termín *page layout analysis* [17].

Problém segmentace stran má za sebou dlouhou historii sahající do konce 70. let, kdy se řešilo rozpoznávání znaků (dále jako *OCR*, angl. Optical Character Recognition). Pro úspěšné zvládnutí rozpoznávání znaků bylo třeba tyto znaky extrahovat, což je potřeba dodnes. Při zpracování různorodých stran se začalo ukazovat, že problém není snadno řešitelný a že existují strany s různou obtížností. To vedlo ke klasifikaci rozvržení strany (angl. *page layout*) dle obrázku 2.1 na obdélníkové (angl. *rectangular*), manhattanské (angl. *Manhattan*), nemanhattanské (angl. *non-Manhattan*) a překrývající se (angl. *overlapping*) [17]. Pro rukopisný text lze předpokládat nejobtížnější překrývající se rozvržení strany.



Obrázek 2.1: Třídy rozvržení strany: obdélníkové (a), manhattanské (b), nemanhattanské (c) a překrývající se (d)(e) [17]

Do současné doby bylo vynaloženo nejvíce úsilí na segmentaci tištěných stran dokumentů [17]. Pro strojové zpracování se však tištěný text od rukopisného významně liší. U rukopisného textu je problematická např. binarizace a nelze se spolehnout na homogenní zarovnání a strukturu textu.

Některé z metod zpracování tištěného textu proto nemusí fungovat pro text rukopisný. Důvodem může být problematické rozdělení rukopisné strany na popředí a pozadí. Kvůli tomu nemusí fungovat např. metody spoléhající na binarizaci a spojené komponenty.

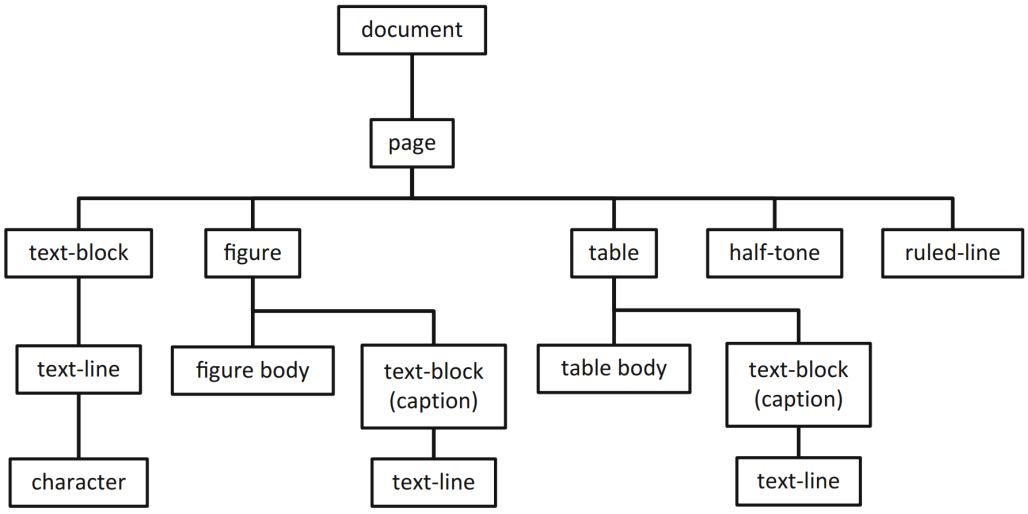
Aby byl výstup segmentační metody použitelný pro OCR, měla by metoda umožňovat separaci pozadí, textu a případně obrázků [25].

Na základě studia metod vhodných pro segmentaci stran jsou metody rozděleny na konzervativní metody a konvoluční neuronové sítě.

2.1 Konzervativní metody

Konzervativní metody zahrnují metody založené např. na spojených komponentách [17]. Zde se předpokládá snadné rozdělení pozadí od popředí. Další metody zahrnují extrakci příznaků a jejich použití pro případnou klasifikaci [18].

V literatuře jsou zmiňované přístupy shora dolů, zdola nahoru a hybridní, které jsou jejich kombinací. U těchto přístupů se využívá hierarchického rozdělení stran (viz obrázek 2.2). Přístup shora dolů spočívá v extrakci větších komponent a jejich dělení na menší komponenty. Např. textové bloky jsou rozděleny na řádky, které jsou poté rozděleny na znaky. Přístup zdola nahoru je opakem přístupu shora dolů a spočívá v extrakci menších komponent a jejich spojováním do větších komponent. Např. jsou extrahovány znaky, které jsou následně spojovány do textových bloků či řádků. V prostudované literatuře převládá přístup zdola nahoru.



Obrázek 2.2: Hierarchie strany dokumentu [17]

2.1.1 Segmentace a klasifikace strany na základě spojených komponent

Metoda pro segmentaci a klasifikaci stran s využitím spojených komponent a přístupu zdola nahoru je prezentována v práci *Page segmentation and classification utilising a bottom-up approach* [12]. Metoda zahrnuje digitalizaci, korekci natočení, segmentaci a klasifikaci na text nebo grafiku.

Prvním krokem je digitalizace a binarizace obrazu pomocí 300 dpi skeneru. Druhým krokem je korekce natočení strany. V tomto kroku je využito segmentace pomocí spojených komponent do bloků (viz třetí krok). Z bloku je nejdříve nalezena nejspodnější spojená komponenta a k ní hledány další komponenty tak, aby tvořily „řádek“ spojených komponent. Z těchto řádků jsou určeny převládající řádky a na jejich základě je poté vypočten úhel s horizontem a tedy úhel sklonu strany.

Třetím krokem je segmentace, která se skládá ze dvou částí. První část je tvorba spojených komponent, jejímž cílem je vytvořit hraniční obdélníky (angl. bounding rectangles) kolem jednotlivých komponent [12]. Vytvořené obdélníky jsou základem pro další analýzu. Druhou částí je seskupení komponent (obdélníků). Seskupují se sousední komponenty podobných rozměrů. Z tohoto důvodu jsou komponenty rozřazeny do kategorií malá, střední a velká na základě zadaných prahů pro jejich velikost. Seskupování poté probíhá v rámci dané kategorie [12] a vznikají bloky.

Posledním krokem je klasifikace, která vychází z rozdílné struktury grafického a textové bloku. Textové bloky jsou tvořeny řádky, které mají téměř rovnoměrné rozestupy. Výšky znaků textu v bloku jsou podobné. Dále je brán v úvahu poměr černé ku bílé a zjištění, že grafické bloky obsahují vnořené komponenty. Na základě toho je blok označen jako textový či grafický.

2.1.2 Segmentace strany pomocí skeletizace pozadí

Metoda segmentace stran prezentovaná v práci *Page Segmentation Based on Thinning of Background* [16] využívá skeletizaci pozadí, což umožňuje segmentaci libovolně natočené strany s nemanhattanským rozvržením strany (viz obrázek 2.1). Vstupem metody je opět binarizovaný obrázek strany.

Postup (viz obrázek 2.3) začíná skeletizací pozadí podle čtyř-sousednosti (jako sousední pixel se bere levý, pravý, horní a dolní). Z výsledku skeletizace pozadí jsou odstraněny „slepé cesty“. Pro jednotlivé spojené komponenty popředí tím vzniknou ohraničení, která vymezují jejich oblasti (viz obrázek 2.3.b).

Další krok odstraňuje zbytečné hranice a dochází ke spojování oblastí (viz

Obrázek 2.3: Segmentace pomocí skeletizace pozadí: (a) strana dokumentu, (b) výsledek skeletizace po odstranění „slepých cest“, (c) výsledek segmentace [16]

obrázek 2.3.c). Bere se v úvahu minimální vzdálenost hranice od popředí D a rozdíl průměrů šířek sousedních bloků W. Dále jsou stanovené prahové hodnoty t_D a t_W. Vzhledem k velkému počtu hranic je jejich odstranění provedeno na základě jednoduché rovnice 2.1.

$$t_D \cdot W + t_W \cdot D \leq t_D \cdot t_W \tag{2.1}$$

2.1.3 Rozšíření metody využívající skeletizace

Možnosti klasifikace segmentovaných oblastí jsou prezentovány v práci *Page Segmentation and Content Classification for Automatic Document Image Processing* [26]. Pro segmentaci je v této práci využita předchozí metoda založená na skeletizaci pozadí [16], kterou se snaží vylepšit a urychlit např. zmenšením velikosti vstupního obrázku. Dále jsou zde prezentovány čtyři metody klasifikace do dvou tříd: text nebo obrázek.

První metoda pro klasifikaci je založena na vzájemné korelaci (angl. cross-correlation) signálu. Jako signál jsou brány jednotlivé sloupce regionu. Porovnávány jsou vždy sousední sloupce. Vzájemná korelace celého regionu je poté určena dle vzorce 2.2, kde $\oplus$ značí operaci XOR, M je výška regionu, N je šířka regionu a $p(i, j)$ je hodnota pixelu i-tého sloupce a j-tého řádku.

$$C_r = 1 - \frac{2}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [p(i, j) \oplus p(i + 1, j)] \tag{2.2}$$

Pokud region dosahuje $C_r \geq 0,86$ je prohlášen za obrázek. V opačném případě je prohlášen za text.

Druhá klasifikační metoda využívá Kolmogorovovy komplexity, kde je míra složitosti sekvence dána délkou nejkratšího binárního programu, který

danou sekvenci dokáže produkovat [26]. Hodnota komplexity je spočtena dle vzorce 2.3, kde N je délka sekvence a $c(N)$ je komplexita spočtená algoritmem navrženým Kasparem a Schusterem [15]. Přičemž platí $0 \leq KC \leq 1$.

$$KC = \frac{1}{N} c(N) log_2 N \tag{2.3}$$

Binarizovaný obrázek (region) je převeden na jednodimenzionální sekvenci hodnot obrazových bodů čtením po řádkách a po sloupcích. Tím se získají dvě sekvence. Výsledná komplexita se získá průměrem komplexit těchto dvou sekvencí. Na základě experimentů autoři ukázali, že obrázek má nízkou komplexitu a text vysokou.

Pro třetí metodu je použita třívrstvá neuronová síť. Vstupní vrstvu tvoří 40 neuronů, skrytou vrstvu 20 neuronů a výstupní vrstva je tvořena 2 neurony. Vstupem sítě je 40 napočítaných příznaků. Výstup tvoří dvě hodnoty odpovídající třídě text a obrázek.

Poslední metodou je hierarchický přístup, který využívá předešlých tří metod. Zjednodušeně je obrázek regionu nejprve zpracován neuronovou sítí. Pokud není výstup neuronové sítě přesvědčivý, jsou použity zbylé metody.

2.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě v poslední době překonaly nejlepší metody (*state-of-the-art*) ve většině úloh vizuálního rozpoznávání. Přestože konvoluční neuronové sítě jsou známé již delší dobu, jejich úspěch byl limitován velikostí trénovacích sad a také hardwarovými nároky, které již dnes nejsou podstatným problémem [23].

Segmentace rukopisných historických dokumentů je velmi náročnou úlohou, protože tyto dokumenty oproti moderním tištěným dokumentům mají zpravidla horší kvalitu, nepravidelnou strukturu apod. Vhodným způsobem zpracování v tomto případě může být právě využití konvolučních neuronových sítí, které se dokáží naučit vhodné příznaky přímo z obrazových bodů a není tak nutné dělat předzpracování obrázku [8, 25].

K problému segmentace pomocí konvolučních neuronových sítí se často přistupuje k jako tzv. *pixel-labeling* problému, kde je každý pixel označen vhodnou třídou.

Následuje úvod do neuronových sítí a výběr možných a dle mého názoru vhodných architektur pro řešení segmentace stran. Výběr se skládá z konvoluční neuronové sítě a dvou plně konvolučních neuronových sítí.

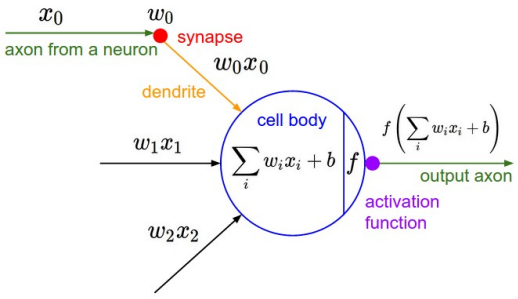
2.2.1 Úvod do neuronových sítí

Cílem této sekce je vysvětlit základy neuronových sítí, což je klíčové pro další pochopení práce. Prezentované informace v této sekci jsou čerpány převážně z [5, 11].

Matematický model neuronu

Matematický model neuronu (dále jako *neuron*, viz obrázek 2.4) představuje základní stavební blok neuronové sítě.

Skládá se ze vstupů x, vah w a parametru b, který představuje *práh* (angl. bias). Váhy w a práh b jsou parametry neuronu, které se nastavují při trénování sítě.



Obrázek 2.4: Matematický model neuronu [11]

Výstup neuronu může být přiveden na vstup dalších neuronů a tím vzniká neuronová síť. Výstup neuronu je spočten dle vzorce 2.4, kde f je aktivační funkce.

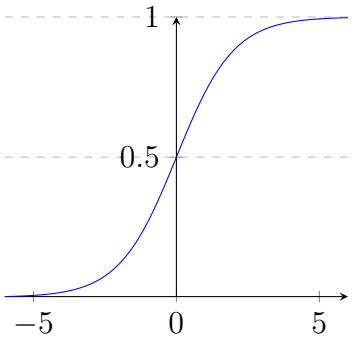
$$output = f(\sum_{i=0}^{n} w_i \cdot x_i + b) \tag{2.4}$$

Aktivační funkce

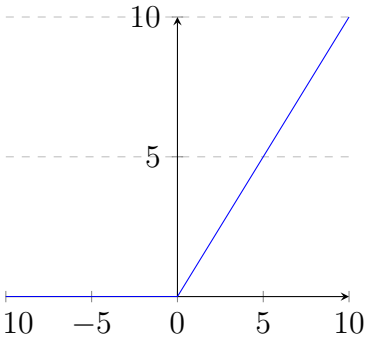
Aktivační funkce modeluje chování biologického neuronu, kde musí vstup přesáhnout určitou mez, aby mohl neuron tzv. vystřelit signál na výstup. Proto je důležité, aby aktivační funkce nebyla lineární. Příkladem aktivační funkce je dle vzorce 2.5 *sigmoida* (viz obrázek 2.5) a dle vzorce 2.6 *ReLU* (viz obrázek 2.6).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.5}$$

$$f(x) = \max(0, x) \tag{2.6}$$



Obrázek 2.5: Sigmoida jako aktivační funkce [5]



Obrázek 2.6: ReLU jako aktivační funkce [5]

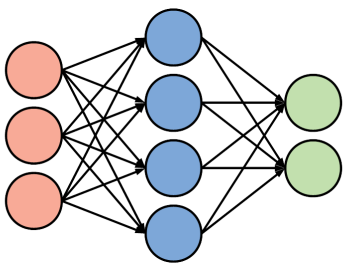
Další funkcí je *softmax*, která se používá pro klasifikaci. Výstupem je pravděpodobnost jednotlivých tříd. Výstup je spočten nad množinou neuronů (obvykle poslední výstupní vrstva sítě) podle vzorce 2.7, kde n značí počet tříd, y_i představuje konkrétní třídu, x značí vstupy, w a b potom představuje váhy a prahy.

$$P(y_i|x, w, b) = \frac{e^{{w_i}^T x_i + b_i}}{\sum_{j=1}^{n} e^{{w_j}^T x_j + b_j}} \tag{2.7}$$

Vrstvy neuronové sítě

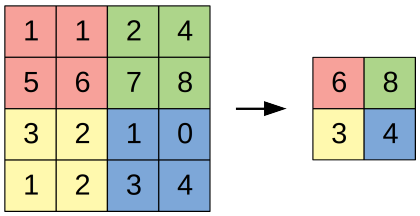
Neurony jsou uspořádány do vrstev, které obsahují neurony stejného typu. Vstupem vrstvy může být výstup předchozí vrstvy. Používané vrstvy jsou následující.

Plně propojená vrstva (viz obrázek 2.7) je vrstva, ve které je každý neuron propojen s každým neuronem v předchozí vrstvě. Vzniká velké množství spojení a tedy parametrů vrstvy (vah). Tyto parametry je nutné natrénovat a uchovávat v paměti počítače.



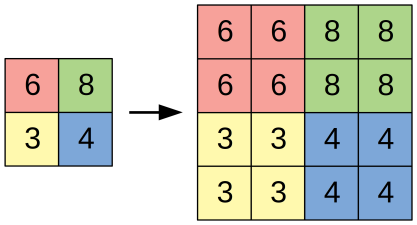
Obrázek 2.7: Ilustrace plně propojené vrstvy [5]

Max-poolingová vrstva (viz obrázek 2.8) redukuje počet parametrů, tj. snížení výpočetní náročnosti sítě. Toho je docíleno výběrem maximální hodnoty ze zvolené oblasti, díky tomu může max-poolingová vrstva také redukovat šum a propagovat pouze silné a důležité spojení (hodnoty). Vrstva nevyžaduje trénování, protože neobsahuje parametry, které by bylo třeba trénovat.



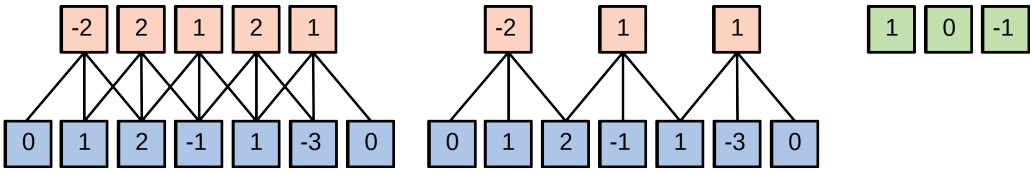
Obrázek 2.8: Princip max-poolingové vrstvy [11]

Upsamplingová vrstva (viz obrázek 2.9) má opačný účel než max-poolingová vrstva. Funguje opačným způsobem a zvětšuje dimenzi. Jedna hodnota vstupu je vyplněna na zvolenou oblast. Tato vrstva podobně jako max-poolingová vrstva nevyžaduje trénování.



Obrázek 2.9: Princip upsamplingové vrstvy

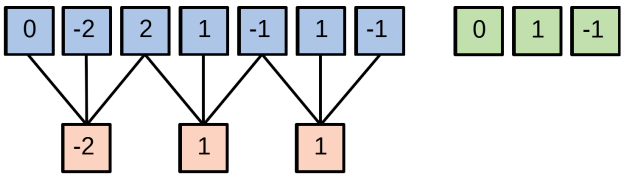
Konvoluční vrstva je základem konvolučních neuronových sítí. Neuron je zde propojen pouze se zvolenou oblastí a jeho natrénované parametry (tvoří *filtr*) jsou sdíleny mezi neurony napříč celou vrstvou (viz obrázek 2.10).



Obrázek 2.10: Možné propojení neuronů v konvoluční vrstvě (vstup modře, výstup červeně, filtr zeleně) [5, 11]

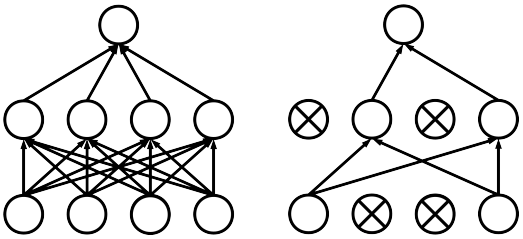
U vrstvy je volena oblast (*velikost filtru*), posun (*stride*) a počet filtrů. Pokud je použito více filtrů je jejich výstup skládán na sebe. Užitečnou možností je tzv. *padding*, který doplní okraje nulami. Lze tak pro daný vstup dosáhnout stejné velikosti výstupu.

Dekonvoluční vrstva neprovádí dekonvoluci. Název dekonvoluční vrstva je běžně používaný, ale jsou vhodnější názvy např. anglicky *convolution transpose*. Vrstva dělá opak konvoluční vrstvy. Je zde filtr, který je váhován vstupem a kopírován na výstup (viz obrázek 2.11). Vrstvu je možné použít podobně jako upsamplingovou vrstvu pro zvětšení rozměrů. Rozdíl je, že dekonvoluční vrstva se zvětšení rozměrů učí.



Obrázek 2.11: Princip dekonvoluční vrstvy: Natrénovaný filtr (zeleně) je váhován vstupem (červeně) a vrácen na výstup (modře).

Dropoutová vrstva (viz obrázek 2.12) slouží jako technika, která dokáže zabránit přetrénování sítě. Tato vrstva je aktivní pouze během trénování sítě a jejím jediným parametrem je pravděpodobnost p, že neuron nebude aktivní. Deaktivace neuronu zabraňuje, aby se síť příliš upnula na vzory během trénování.



Obrázek 2.12: Princip dropoutové vrstvy: model sítě před (vlevo) a po (vpravo) deaktivaci neuronů (kruhy představují aktivní neurony, kruhy s křížem neaktivní neurony a šipky aktivní spojení mezi neurony) [5]

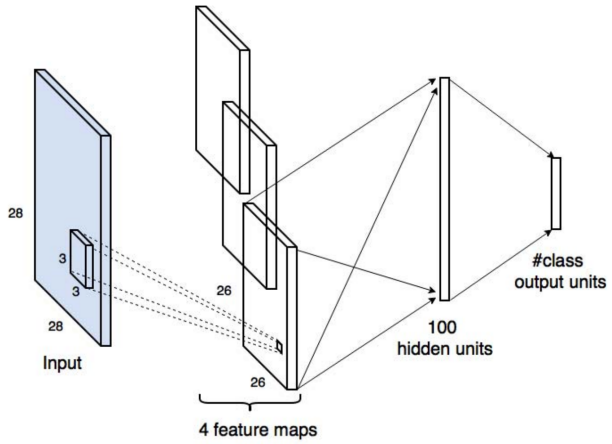
Trénování neuronové sítě

Trénování má za cíl optimální nastavení parametrů sítě. Nejčastěji je řešeno zpětným šířením chyb – *backpropagation* (viz [11]), které spočívá ve vy-

hodnocení vstupu a porovnání s požadovaným výstupem. Následně je možné spočítat, jak které neurony ovlivnily výslednou chybu a upravit jejich parametry tak, aby byla chyba zmenšena. K tomu je třeba mít připravenou množinu trénovacích dat, která obsahuje vstupy a k nim požadované výstupy – *ground-truth* (viz obrázek 4.4).

2.2.2 Konvoluční neuronová síť

V práci *Convolutional Neural Networks for Page Segmentation of Historical Document Images* [8] je segmentace uvažována jako tzv. *pixel-labeling* problém. Pro značení pixelů je zde navržena konvoluční neuronová síť. Hlavní myšlenkou je naučit síť extrahovat příznaky a na jejich základě klasifikovat pixely do požadovaných tříd.



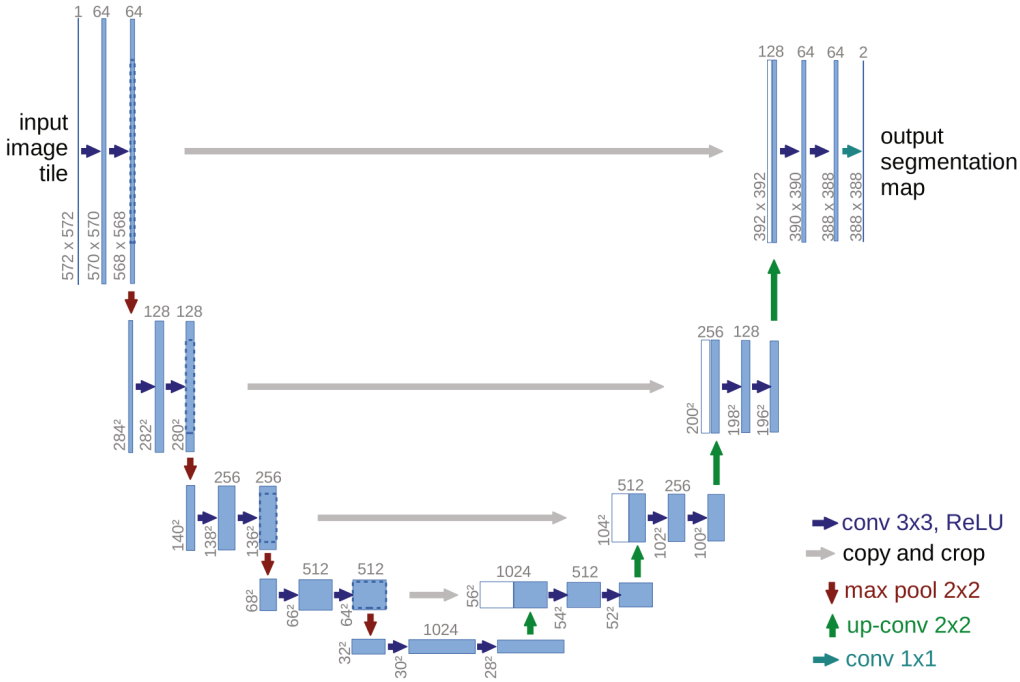
Obrázek 2.13: Architektura konvoluční neuronové sítě [8]

Architektura sítě (viz obr. 2.13) je poměrně jednoduchá. Vstupem sítě je obrázek ve stupních šedi s rozměry 28×28 pixelů. Následuje jedna konvoluční vrstva se čtyřmi filtry velikosti 3×3. Výstupem konvoluční vrstvy je 26×26×4 hodnot, které tvoří vstup plně propojené vrstvy se 100 neurony. Jako poslední je plně propojená vrstva s M neurony, kde M je počet tříd. Poslední vrstva používá *softmax* aktivační funkci a predikuje tak pravděpodobnosti jednotlivých tříd. U ostatních vrstev je použita $ReLU$ aktivační funkce. Za výslednou třídu je zvolena třída s nejvyšší pravděpodobností.

Pro urychlení procesu je nejprve použit algoritmus pro detekci tzv. *superpixelů* [8]. Superpixel představuje část obrazu, která obsahuje pixely patřící ke stejnému objektu např. znak. Z tohoto superpixelu je zvolen prostřední pixel a kolem něho vybrána oblast 28×28 pixelů, která je přivedena na vstup sítě. Výsledek je poté přiřazen celému superpixelu.

2.2.3 Plně konvoluční neuronová síť U-Net

V práci *U-Net: Convolutional Networks for Biomedical Image Segmentation* [23] je prezentována síť pro segmentaci biomedicínských obrázků. Název sítě vychází z její architektury (viz obrázek 2.14), která připomíná písmeno U. Síť U-Net byla původně navržena pro segmentaci biomedicínských obrázků, ale v současné době existuje řada jejích modifikací pro různé účely včetně segmentace dokumentů.



Obrázek 2.14: Architektura sítě U-Net [23]

Architektura U-Net vychází z typického použití konvolučních sítí pro klasifikaci obrázku do dané třídy. Podobně jako v předchozí metodě založené na konvoluční neuronové síti [8] je díky posuvnému okénku (angl. sliding-window) možné klasifikovat i jednotlivé pixely podle jejich okolí. Tento přístup umožňuje lokalizaci objektů a vzhledem k počtu pixelů jsou trénovací data ve smyslu počtu trénovacích vzorů mnohem rozsáhlejší než počet obrázků. Zároveň jsou zde ale dva problémy.

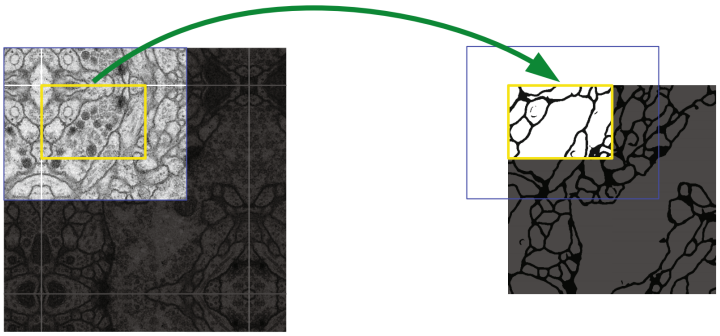
Prvním problémem je, že pro každý pixel musí být samostatně predikována daná oblast. Jelikož se oblasti překrývají, dojde k velké redundanci výpočtů a zvýšené časové náročnosti.

Druhým problémem je nutný kompromis mezi přesností lokalizace a využitím kontextu. Malé oblasti na vstupu nemají dostatečný kontext. Velké oblasti mají zase problém s přesnou lokalizací, protože vyžadují více max-

poolingových vrstev, což snižuje přesnost lokalizace.

Pro řešení těchto problémů je v této práci použita plně konvoluční neuronová síť, která bere v úvahu příznaky z více vrstev. To vede k dobré lokalizaci i možnosti většího kontextu.

Architektura sítě U-Net (viz obr. 2.14) se skládá z kontrakční (vlevo) a expanzivní (vpravo) části. Kontrakční část odpovídá klasické architektuře konvoluční sítě. Tvoří ji opakující se bloky ze dvou 3×3 konvolučních vrstev bez paddingu a 2×2 max-poolingové vrstvy. Po každé max-poolingové vrstvě je zdvojnásoben počet filtrů. Expanzivní část se skládá z upsamplingové vrstvy následované 2×2 konvoluční vrstvou (v obrázku 2.14 jako „up-conv“). Tím se zvětší rozměry a redukuje počet filtrů na polovinu. K těmto hodnotám jsou po oříznutí přidány příznaky „mělčí“ vrstvy (tzv. *skip-connections*). Následují dvě 3×3 konvoluční vrstvy. Jako poslední vrstva je použita 1×1 konvoluční vrstva mapující příznaky do požadovaného počtu tříd.



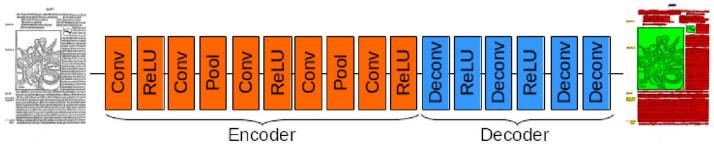
Obrázek 2.15: Overlap-tile strategy: Pro predikci ve žlutém rámečku je potřeba obrázek v modré oblasti. Případná chybějící data jsou získána zrcadlením (viz okraje levého obrázku). [23]

Síť používá pouze konvoluční vrstvy bez paddingu. Používají se tedy pouze hodnoty, pro které je dostupný celý kontext. Důvodem této strategie je umožnění tzv. overlap-tile strategy (viz obrázek 2.15).

2.2.4 Plně konvoluční neuronová síť (FCN)

Adaptace výše popsané sítě U-Net [23] pro segmentaci historických dokumentů je prezentována v práci *Fully Convolutional Neural Networks for Page Segmentation of Historical Document Images* [25].

Tato síť FCN však na rozdíl od sítě U-Net neobsahuje tzv. *skip-connections* (viz „copy and crop“ vazby v obrázku 2.14). Autoři tvrdí, že důležité jsou pouze informace kódující velké oblasti, a proto nejsou *skip-connections* potřeba.



Obrázek 2.16: Architektura sítě FCN [25]

Architektura sítě FCN (viz obr. 2.16) je tvořena *enkodér* a *dekodér* částí. Část enkodér tvoří 2×2 max-poolingové vrstvy a konvoluční vrstvy s 5×5 filtrem. U konvolučních vrstev je použit padding a počet filtrů od první vrstvy odpovídá hodnotám 40, 60, 120, 160 a 240. Část dekodér tvoří dekonvoluční vrstvy. První dvě dekonvoluční vrstvy mají filtr 2×2 pro zvětšení rozměrů odpovídající vstupu. Počet filtrů v dekonvolučních vrstvách je 240, 120, 60 a 6. Výstupem je tedy predikce pixelů do 6 tříd.

Vstupní obrázek je nejprve zmenšen v odpovídajícím poměru tak, aby se vešel do oblasti 260×390 pixelů. Případné nevyužité místo je vyplněno bílou barvou. Poté je v jednom průchodu celý obrázek zpracován sítí, čímž se získá jeho maska. Maska je následně zvětšena na velikost původního obrázku.

3 Data

Vhodná data jsou stěžejním bodem této diplomové práce, protože jsou potřebná pro trénování a vyhodnocení systému pro segmentaci stran. Práce se zabývá segmentací stran rukopisných dokumentů na text, obrázky a pozadí. Z tohoto důvodu jsou vyžadovány strany rukopisných dokumentů obsahující text a obrázky s odpovídající anotací.

3.1 Porta fontium

Porta fontium [21] je webová platforma (informační systém) umožňující bezplatné vyhledávání archiválií ze státních archivů České republiky a Bavorska.

Na přeshraničním projektu spolupracují Státní oblastní archiv v Plzni a Generální ředitelství státních bavorských archivů v Mnichově. Projekt by měl umožnit vyhledávání digitalizovaných dokumentů.

Bavorští partneři projektu provádějí tyto aktivity:

- digitalizace archivních souborů
- vědecké zpracování (zpřístupňování) listin kláštera Waldsassen
- zpřístupňování sbírky fotografií (identifikace objektů)
- integraci digitalizovaných objektů a zároveň propojení se společnou internetovou prezentací realizovaného projektu
- spolupráce na obsahové a technické výstavbě společné prezentace projektu
- pořádání společných pracovních setkání během doby trvání projektu

Český vedoucí partner realizuje následující aktivity:

- digitalizace archivních fondů
- vědecké zpracování listin, aktového materiálu a knih Archivu města Cheb z provenience kláštera ve Waldsassenu
- integrace digitalizovaných materiálů, jejich popis metadaty

3.1.1 Cíle

Historie a vzájemné soužití Čechů a Němců jsou úzce provázané. Jejich soužití bylo poznamenáno několika událostmi dějin. Za jednu takovou událost je považováno násilné rozdělení značného množství archiválií v důsledku 2. světové války. Tyto materiály jsou uloženy v českých a bavorských státních archivech.

Hlavním cílem projektu je tedy opětovné spojení k sobě patřících archiválií do virtuálního celku a vytvoření digitální reprodukce, která bude prostřednictvím webového serveru poskytnuta široké veřejnosti, vědeckému světu i regionálním badatelům.

K dalším cílům patří rozvoj česko-německých vztahů, vytvoření vzorové prezentační základny pro odbornou meziarchivní komunikaci, komunikaci historických pracovišť a také dalších odborných a vědeckých zařízení.

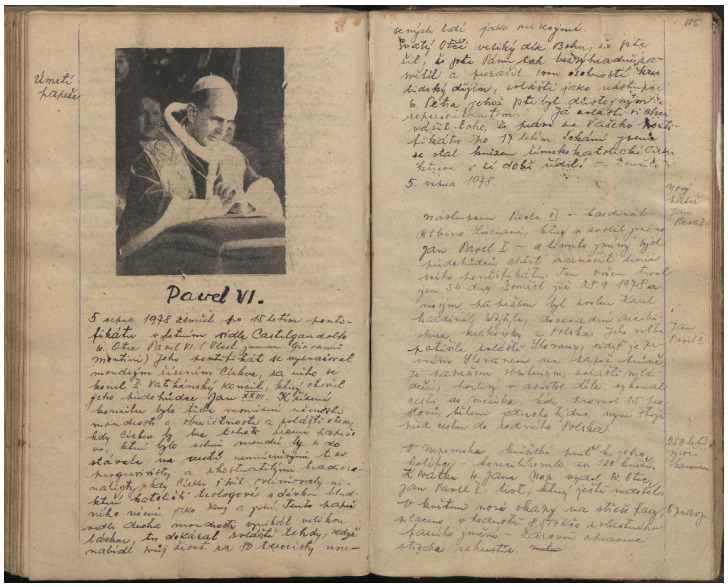
3.1.2 Podmínky užití

Digitální kopie archiválií a metadata jsou zpřístupněny pro jakékoli nekomerční účely za podmínky, že digitální reprodukce archiválií nebo digitální repliky archiválií budou užívány v databázi Porta fontium. Užívání těchto kopií mimo databázi je umožněno „pro svou osobní potřebu, pro účely vědecké nebo vyučovací v rozsahu odůvodněném sledovaným nevýdělečným účelem za podmínky, že je uveden konkrétní zdroj těchto kopií včetně internetové adresy“ [21].

3.1.3 Poskytnutá data

Vedoucím práce poskytnutá data obsahují kvalitně digitalizované strany dokumentů. Bohužel vzhledem ke stáří dokumentů, které pochází převážně z 19. a 20. století, se na některých dokumentech vyskytují nedokonalosti jako např. rozpité skvrny od inkoustu, různá zahnutí stránky způsobená pravděpodobně vlhkostí nebo průsvit či otisk textu z jiných stran. Celkově je však kvalita stran na dobré úrovni (viz obrázek 3.1).

Bohužel k digitalizovaným stránkám dokumentů nejsou poskytnuta metadata vhodná pro segmentaci na text, obrázky a pozadí. Stroji tedy na základě poskytnutých dat nelze podat informace o rozmístění obrázků ani textových bloků.



Obrázek 3.1: Ukázka poskytnutých stran dokumentů z portálu Porta fontium [21]

3.2 Dostupné datové sady pro segmentaci stran dokumentů

Vzhledem k chybějící anotaci v poskytnutých datech byly hledány další datové sady vhodné pro segmentaci stran. Tyto datové sady jsou shromážděné na webových stránkách Pattern Recognition & Image Analysis Research Lab [22]. Pro přístup k datovým sadám je nutné mít zřízený účet a schválený přístup ke konkrétní datové sadě. Datové sady jsou zde dostupné pro osobní i vědecké účely a jejich hlavní zástupci budou uvedeni dále:

Layout Analysis Dataset

Datová sada Layout Analysis Dataset [1] byla vytvořena zejména pro vyhodnocení metod analýzy rozvržení stran (angl. layout analysis). Obsahuje 478 stran, což ji řadí mezi největší z prostudovaných datových sad. Ke každé straně je poskytnuta anotace s kompletním rozvržením stránky. Anotace je ve formátu PAGE (viz sekce 4.4).

Datová sada obsahuje tištěné dokumenty (ukázka viz obrázek 3.2) se širokou škálou různých rozložení stran. Většina stran obsahuje text s obrázky. Dokumenty pochází převážně z moderních novin a časopisů. Je zde tedy zastoupeno velké množství fontů a různé způsoby vkládání obrázků.

Obrázek 3.2: Ukázka strany dokumentu z datové sady Layout Analysis Dataset [1]

IMPACT Project Dataset

IMPACT project (Improving Access to Text) je velký projekt financován Evropskou komisí. Jeho cílem je zlepšení přístupu k historickému textu a redukce překážek, které brání hromadné digitalizaci evropského kulturního dědictví [14].

V rámci projektu by měla být vytvořena datová sada [19] obsahující množinu přesahující 600 tisíc obrázků dokumentů původem z Evropských knihoven včetně České republiky. V současné době však stále není dokončen a na webových stránkách Pattern Recognition & Image Analysis Research Lab tak není dostupný.

RDCL2019

RDCL2019 [10] je datová sada použitá pro soutěž v analýze stran dokumentů s komplexním rozvržením *ICDAR2019 Competition on Recognition of Documents with Complex Layouts*. Obsahuje 85 neanotovaných a 15 anotovaných stran z tištěných dokumentů a vychází z datové sady Layout Analysis Dataset. Anotace je ve formátu PAGE.

Obrázek 3.3: Ukázka strany dokumentu z datové sady HNLA2013 [4]

HNLA2013

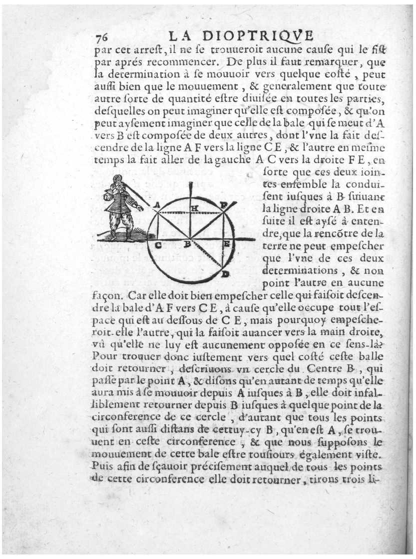
HNLA2013 [4] je datová sada použitá pro soutěž v analýze stran historických novin *ICDAR2013 Competition on Historical Newspaper Layout Analysis*. Datová sada obsahuje 50 neanotovaných a 8 anotovaných stran historických novin, ukázka viz obrázek 3.3. Anotace je ve formátu PAGE. Datová sada byla vytvořena v rámci projektu IMPACT.

HBR2013

HBR2013 [3] je datová sada použitá pro soutěž v analýze stran historických knih *ICDAR2013 Competition on Historical Book Recognition*. Datovou sadu tvoří historické knihy a obsahuje 200 neanotovaných a 6 anotovaných stran, ukázka viz obrázek 3.4. Anotace je ve formátu PAGE. Datová sada byla vytvořena v rámci projektu IMPACT.

HDLAC2011

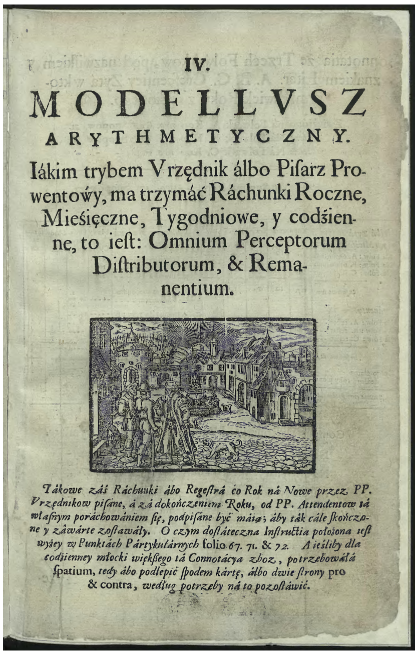
HDLAC2011 [2] je datová sada použitá pro soutěž v analýze stran historických dokumentů *ICDAR2011 Historical Document Layout Analysis Competition*. Zde se jedná o datovou sadu pro rozvržení historických dokumentů. Obsahuje 100 neanotovaných a 6 anotovaných stran, ukázka viz obrázek 3.5. Většina stran je opět tištěných. Anotace je ve formátu PAGE. Datová sada byla vytvořena v rámci projektu IMPACT.



76 LA DIOPTRIQVE

par cet arreſt, il ne ſe trouueroit aucune cauſe qui le fiſt par aprés recommencer. De plus il faut remarquer, que la determination à ſe mouuoir vers quelque coſté, peut auſſi bien que le mouuement, & generalement que toute autre ſorte de quantité eſtre diuiſée en toutes les parties, desquelles on peut imaginer qu'elle eſt composée, & qu'on peut aysement imaginer que celle de la bale qui ſe meut d'A vers B eſt composée de deux autres, dont l'vne la fait deſcendre de la ligne A F vers la ligne C E, & l'autre en meſme temps la fait aller de la gauche A C vers la droite F E, en ſorte que ces deux ioin-tes enſemble la conduiſent iuſques à B ſuiuant la ligne droite A B. Et en ſuite il eſt ayſé à entendre, que la rencōtre de la terre ne peut empeſcher que l'vne de ces deux determinations, & non point l'autre en aucune façon. Car elle doit bien empeſcher celle qui faiſoit deſcendre la bale d'A F vers C E, à cauſe qu'elle occupe tout l'eſpace qui eſt au deſſous de C E, mais pourquoy empeſcheroit-elle l'autre, qui la faiſoit auancer vers la main droite, vû qu'elle ne luy eſt aucunement opposée en ce ſens-là? Pour trouuer donc iuſtement vers quel coſté ceſte balle doit retourner, deſcriuons vn cercle du Centre B, qui paſſe par le point A, & diſons qu'en autant de temps qu'elle aura mis à ſe mouuoir depuis A iuſques à B, elle doit infalliblement retourner depuis B iuſques à quelque point de la circonference de ce cercle, d'autant que tous les points qui ſont auſſi diſtans de cettuy-cy B, qu'en eſt A, ſe trouuent en ceſte circonference, & que nous ſuppoſons le mouuement de cette bale eſtre touſiours également viſte. Puis afin de ſçauoir préciſement auquel de tous les points de cette circonference elle doit retourner, tirons trois li-

Obrázek 3.4: Ukázka strany dokumentu z datové sady HBR2013 [3]



IV.

MODELLVSZ

ARYTHMETYCZNY.

Iákim trybem Vrzędnik álbo Piſarz Prowentowy, ma trzymáć Ráchunki Roczne, Mieśięczne, Tygodniowe, y codźienne, to ieſt: Omnium Perceptorum Diſtributorum, & Remanentium.

Tákowe záś Ráchunki ábo Regeſtrá co Rok ná Nowe przez PP. Vrzędnikow piſane, á zá dokończeniem Roku, od PP. Attendentow zá właſnym poráchowániem śię, podpiſane być máią; áby ták cále ſkończone y záwárte zoſtawáły. O czym doſtáteczna Inſtructia położona ieſt wyżey w Punktách Pártykulárnych folio 67. 71. & 72. *A ieśliby dla codźienney młocki większego tá Connotácya zboz, potrzebowáłá* ſpatium, *tedy ábo podlepić ſpodem kártę, álbo dwie ſtrony* pro & contra, *według potrzeby ná to pozoſtáwić.*

Obrázek 3.5: Ukázka strany dokumentu z datové sady HDLAC2011 [2]

4 Příprava datové sady

Jelikož se nepodařilo najít odpovídající datovou sadu, která by byla zároveň vhodně anotována pro účely této práce, byla připravena vlastní datová sada. Jako zdroj dat byly použity strany kronik z portálu Porta fontium [21].

4.1 Požadavky na datovou sadu

Základním požadavkem na datovou sadu je, aby obsahovala obrázky stran dokumentů s ručně psaným textem a obrázky. V [1] zmiňují tři hlavní vlastnosti, které charakterizují kvalitní datovou sadu jak z hlediska obsahu, tak z hlediska použitelnosti. Jsou to:

1. realističnost – datová sada musí obsahovat reprezentativní vzorek reálných dat
2. detailnost – datová sada musí obsahovat detailní informace umožňující důkladné vyhodnocení
3. flexibilní struktura – procházení datové sady by mělo být snadné a mělo by umožňovat vyhledávání a výběr podskupin se specifickými vlastnostmi

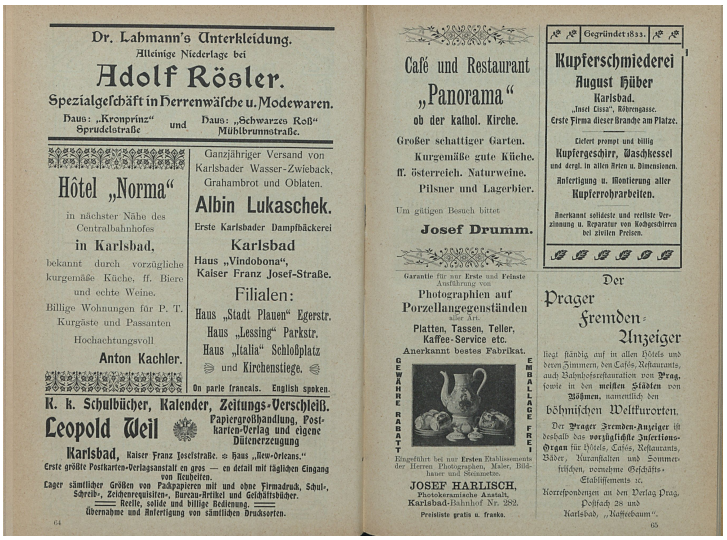
4.2 Použitelná data

Vzhledem k tomu, že je požadována segmentace stran rukopisů, jsou vhodnými daty stránky z kronik Porta fontium. Vedoucím práce bylo poskytnuto 10 vyhovujících stran či dvojstran (dále jako *stran*) kronik z fary Budětice a fary Petrovice u Sušice (příklad viz obrázek 3.1). Dále bylo poskytnuto celkem 199 stran z kronik Blovice, Chudenice a Hroznětín. Z těchto kronik však obsahovalo obrázek pouze 8 stran a to z kroniky Blovice.

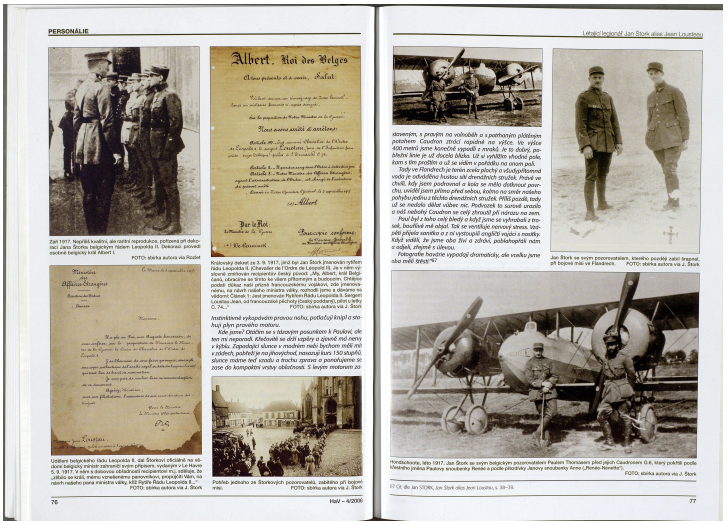
K dispozici bylo také 10 stran periodik (ukázka viz obrázek 4.2), 10 stran adresářů (ukázka viz obrázek 4.1) a 10 stran s fotografiemi a tištěnými popisky. Tyto strany však obsahují výhradně tištěný text a rozložením strany neodpovídají definovaným požadavkům.

Z prostudovaných datových sad (viz sekce 3.2) nebyly nalezeny strany s ručně psaným textem a většina stran není anotována. Při ručním procházení bylo vytipováno 14 stran (příklady viz obrázky 3.3, 3.4 a 3.5), které by rozvržením a stylem mohly odpovídat stranám z kronik.

Celkem bylo z pěti různých kronik shromážděno 18 stran obsahujících obrázky i text a velké množství stran obsahujících pouze text bez obrázků. Dále bylo shromážděno celkem 34 tištěných stran adresářů, stran s fotografiemi a stran z prostudovaných datových sad.



Obrázek 4.1: Ukázka strany adresáře z portálu Porta fontium [21]



Obrázek 4.2: Ukázka strany periodika z portálu Porta fontium [21]

4.3 Systém pro anotaci Aletheia

Aletheia [9] je systém použitý pro anotaci stran dokumentů, který byl částečně vyvinut v rámci projektu IMPACT a je dostupný na stránkách Pat-

tern Recognition & Image Analysis pro osobní a výzkumné účely. Hlavním cílem tohoto systému je efektivita, přesnost, flexibilita a použitelnost. Umožňuje anotovat regiony, řádky textu, jednotlivá slova či písmena a spoustu dalšího. Poskytuje i automatické metody pro anotaci včetně rozpoznávání znaků. Tyto automatické metody pracují korektně pro moderní tištěné dokumenty. Pro starší, ručně psané dokumenty (ukázka viz obrázek 3.1) nebyly v době vypracování této práce automatické metody použitelné a práci neulehčily. Aletheia pracuje s formátem PAGE (viz sekce 4.4), do kterého anotaci ukládá, případně ji čte.

Aletheia umožňuje přesnou a detailní definici regionů, které jsou klíčové pro tuto práci. Podporuje jedenáct typů regionů: text, obrázek, grafika, čára řádky, graf, oddělovač, tabulka, matematika, šum, rámeček a neznámý. Pro některé typy existují ještě podtypy značící logickou funkci regionu. Pro textový region je to odstavec, nadpis, číslo stránky apod. Mohou být zadány i další doplňující atributy regionu např. jazyk, font a text pro textový region.

4.3.1 Nástroje

Pro anotaci regionů nabízí Aletheia několik užitečných manuálních a poloautomatických nástrojů, které práci zrychlí a zpřesní. Většina nástrojů vyžaduje binarizovaný obrázek strany a využívá spojených komponent.

Klíčovým krokem pro následnou práci je binarizace. Pro binarizaci je na výběr mezi manuálním prahováním, Otsu algoritmem, Sauvola algoritmem a adaptivní binarizací. Rozdíl je v kvalitě a především v rychlosti binarizace.

Dalším užitečným nástrojem je odstranění šumu. Během binarizace starších dokumentů většinou dojde v závislosti na použitém algoritmu k vytvoření většího či menšího množství šumu. Odstranění šumu probíhá pomocí spojených komponent a ručně zadaného prahu. Pokud je spojená komponenta menší než zadaný práh, je spojená komponenta z binarizovaného obrazu odstraněna. Další možností je ruční výběr spojených komponent představujících šum a jejich následné smazání.

Regiony je možné označovat manuálně a to jako obdélník nebo polygon. Užitečné nástroje pro urychlení a zpřesnění označení regionu jsou *To Coarse Contour* a *To Fine Contour*. *To Coarse Contour* je doporučen pro text. Výsledná hranice regionu je hrubší a obsahuje méně bodů. *To Fine Contour* je vhodný např. pro obrázky složitých tvarů. Je jemnější, ale obsahuje velké množství bodů a je tak náročnější pro následné výpočty. Oba nástroje jsou volány nad vytvořeným regionem a pracují na podobném principu. Na základě zadaných parametrů je region zmenšován dokud hranice regionu nenarazí na černý bod v binarizovaném obrázku. Pro tyto nástroje je tedy

důležitá kvalitní binarizace bez šumu. Dále je možné regiony upravovat manuálně vložením či odstraněním bodu.

4.4 PAGE formát

```
<?xml version="1.0" encoding="UTF-8"?>
<PcGts
  xmlns="http://schema.primaresearch.org/.../2019-07-15"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="...">
  <Metadata>
    <Creator>Baloun Josef</Creator>
    <Created>2020-02-20T11:13:27</Created>
    <LastChange>2020-02-20T11:51:37</LastChange>
  </Metadata>
  <Page
    imageFilename="mesto-blovice-1837-1954_0860.jpg"
    imageWidth="5777" imageHeight="4479">

    <TextRegion id="r0" type="paragraph">
      <Coords points="5462,91 5590,91 5590,179 5462,179"/>
      <TextEquiv>
            <Unicode></Unicode>
      </TextEquiv>
    </TextRegion>

    <ImageRegion id="r1">
      <Coords points="4802,3723 4832,2794 3577,2771 3546,3700"/>
    </ImageRegion>
  </Page>
</PcGts>
```

Obrázek 4.3: Příklad PAGE formátu

PAGE (Page Analysis and Ground-truth Elements) formát je dle [9] flexibilní, rozšiřitelný a vyspělý formát, který umožňuje přesnou a detailní anotaci. PAGE formát [20] je používán v rámci projektu IMPACT a na mezinárodních soutěžích ICDAR (International Conference on Document Analysis and Recognition). Využívá jako základní technologii jazyk XML, který umožňuje přímou integraci a to nejen pro výzkumné projekty. Existuje také řada nástrojů podporujících PAGE formát (např. nástroje pro poloautomatickou anotaci, vyhledávání, vyhodnocení, prohlížení atd.).

PAGE formát umožnuje precizní popis jakéhokoliv elementu, který může být nalezen na stránkách dokumentu. V rámci stránky jsou všechny elementy

reprezentovány regionem určitého typu. Nejběžnější typy regionu jsou text, obrázek, grafika, čára řádky, graf, oddělovač, tabulka, matematika, šum, rámeček a neznámý. Textový region je možné dále dělit na řádky, jednotlivá slova či znaky.

Region je popsán identifikátorem a souřadnicemi polygonu, který reprezentuje hranici regionu. Kromě identifikátoru a souřadnic, které nalezneme v každém regionu, jsou dostupná další specifická metadata v závislosti na typu regionu. Textový region může obsahovat informace například o barvě textu, barvě pozadí a specifikaci podtypu (nadpis, odstavec atd.). Jelikož je rozpoznávání textu hlavním cílem analýzy dokumentů, je zde možné uchovávat textový obsah pro všechny textové elementy. Dále je zde podpora pro specifikaci pořadí čtení.

Příklad PAGE souboru je na obrázku 4.3. Kořenový element `PcGts` obsahuje elementy `Metadata` a `Page`. Element `Metadata` obsahuje např. autora, datum a čas vytvoření a další údaje. Element `Page` obsahuje anotaci stránky. Mezi jeho atributy je např. název souboru a rozměry obrázku. Element `Page` obsahuje elementy regionů. Prvním elementem je `TextRegion`, který představuje textový region, který je typu `paragraph` a jeho identifikátor je `r0`. Souřadnice regionu jsou definované elementem `Coords`. Případný text regionu je definovaný elementem `TextEquiv`. Dalším elementem je `ImageRegion` s identifikátorem `r1`, který představuje region obrázku a je definován opět elementem `Coords`.

PAGE formát si prošel určitým vývojem a existuje více verzí, které se mohou lišit např. ve způsobu uložení souřadnic. Nicméně názvy elementů, jejich atributů a struktura je samovysvětlující.

4.5 Postup anotace

Pro anotaci je použit nástroj Aletheia (viz sekce 4.3). Postup anotace obrázku strany (případně dvojstrany) dokumentu prováděný v této práci je následující:

1. vyplnění metadat strany
    - vyplnění autora anotace
    - případná poznámka (např. úprava již anotované strany)

2. binarizace strany
    - pro většinu případů použita rychlá Otsu binarizace
    - při neúspěchu Otsu binarizace na problematických stranách je použita adaptivní binarizace
3. odstranění šumu po binarizaci
    - redukce šumu pomocí nástroje využívajícího spojené komponenty a jejich prahování (viz sekce 4.3), práh volen pro každou stranu individuálně
    - následně manuální výběr a mazání komponent představujících šum
4. označování textu
    - označeny regionu textu (čísla stránek, poznámky, popisky fotografií a obrázků, odstavce apod.), v souvislém textu jsou anotovány odstavce jako jeden region
    - postup:
        (a) manuální označení formou polygonu
        (b) použití nástroje *To Coarse Contour* (viz sekce 4.3)
        (c) kontrola a případná manuální editace regionu
5. označování obrázků a ostatních grafických prvků (razítka apod.)
    - postup pro přibližně obdélníkové tvary:
        (a) manuální označení formou polygonu
        (b) kontrola a případná manuální editace regionu
    - postup pro složitější tvary:
        (a) manuální označení formou polygonu
        (b) použití nástroje *To Fine Contour* (viz sekce 4.3)
        (c) kontrola a případná manuální editace regionu
6. nastavení atributů regionů
    - text nastaven jako TextRegion
    - obrázek nastaven jako ImageRegion
    - ostatní grafické prvky nastaveny jako GraphicRegion

4.6 Vytvořená datová sada

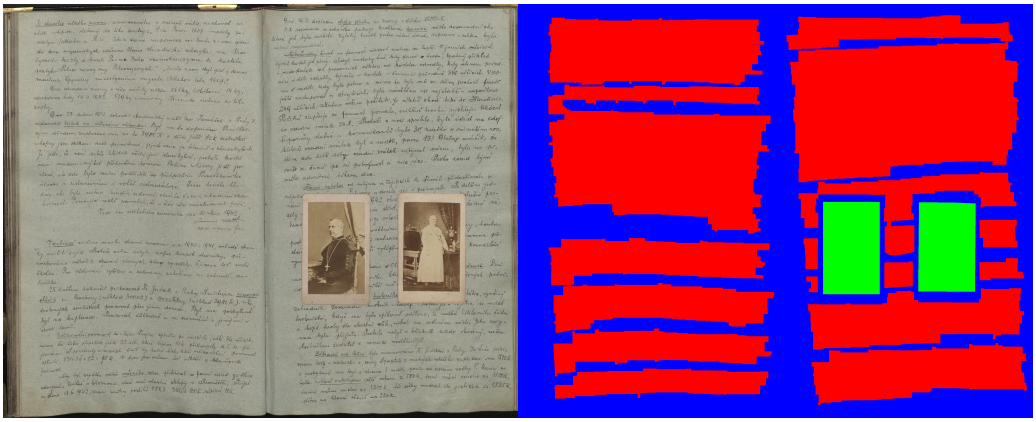
Celkem bylo anotováno 58 obrázků stran poskytnutých z portálu Porta fontium [21]. Tyto obrázky zahrnují 10 stran adresářů, 10 stran s fotografiemi a tištěnými popisky, 18 vyhovujících stran kronik s obrázky a 20 stran kronik obsahujících pouze text. Dále bylo anotováno, případně upravena anotace u 14 tištěných stran z prostudovaných datových sad (viz sekce 3.2).

Do datové sady nebyly zahrnuty strany periodik. Důvodem je jejich moderní vzhled a jiné rozvržení stran, kdy je jako obrázek vložena strana odlišného dokumentu (viz obrázek 4.2). Tyto strany však obsahují značné množství obrázků. Vhodné obrázky (celkem 29) byly vyříznuty pro další zpracování.

Požadavky na datovou sadu dle sekce 4.1 jsou považovány za splněné díky výběru vhodných stran z pěti kronik a detailní anotaci pro segmentaci stran na text, obrázky a pozadí. Použití formátu PAGE [20] pro anotaci stran rozšiřuje možnosti využití této datové sady díky množství nástrojů, které práci s tímto formátem podporují.

Velikost strany obrázku v datové sadě je omezena na 1024 pixelů, menší strana je v daném poměru původního obrázku. Důvodem je časová a hardwarová náročnost pro pozdější zpracování.

K danému obrázku je z PAGE souboru s anotací vygenerován PNG soubor s požadovaným výsledkem segmentace (angl. *ground-truth*). PNG soubor má stejné rozměry a obsahuje 3 kanály R, G a B, kde jsou uloženy informace pro segmentaci (viz obrázek 4.4). Kanál R představuje text na daném pixelu, kanál G představuje obrázek či grafický prvek na daném pixelu a kanál B představuje pozadí na daném pixelu (doplněk R a G kanálů).



Obrázek 4.4: Ukázka jedné dvojstrany z vytvořené datové sady: vlevo dvojstrana kroniky, vpravo *ground-truth*

PNG soubor obsahuje hodnoty 0 a 255. Hodnota 0 značí, že daný pixel nepatří do kategorie dané kanálem např. se nejedná o text. Hodnota 255 značí, že daný pixel patří do kategorie dané kanálem např. se jedná o text. Daný pixel může být tedy zároveň text i obrázek, což je vhodné např. u obrázku který je popsán textem.

Datová sada je rozdělena na testovací, trénovací a validační části, které jsou použity k experimentům pro zjištění optimálních parametrů systému. Protože jsou data určená pro trénovací množinu různorodá (strany bez obrázků, samostatné obrázky, rukopisné a tištěné strany dokumentů), je členění trénovací množiny podrobnější. Struktura datové sady je podle složek následující:

- test
  - testovací část dat
  - 8 stran kronik s obrázky
- valid
  - validační část dat
  - 4 strany kronik s obrázky
- train
  - trénovací část dat
  - 6 stran kronik s obrázky
- train_imgs
  - trénovací část dat
  - 29 obrázků vyříznutých z periodik
- train_printed
  - trénovací část dat
  - 34 stran s tištěným textem z různých zdrojů
- train_text
  - trénovací část dat
  - 20 stran kronik bez obrázků

5 Prototyp systému pro segmentaci stran

Na základě prostudované literatury je navržen prototyp systému pro segmentaci stran. Základem systému je plně konvoluční neuronová síť. Plně konvoluční sítě jsou zvoleny, protože na problému segmentace historických dokumentů dosahují nejlepších výsledků z prostudovaných metod.

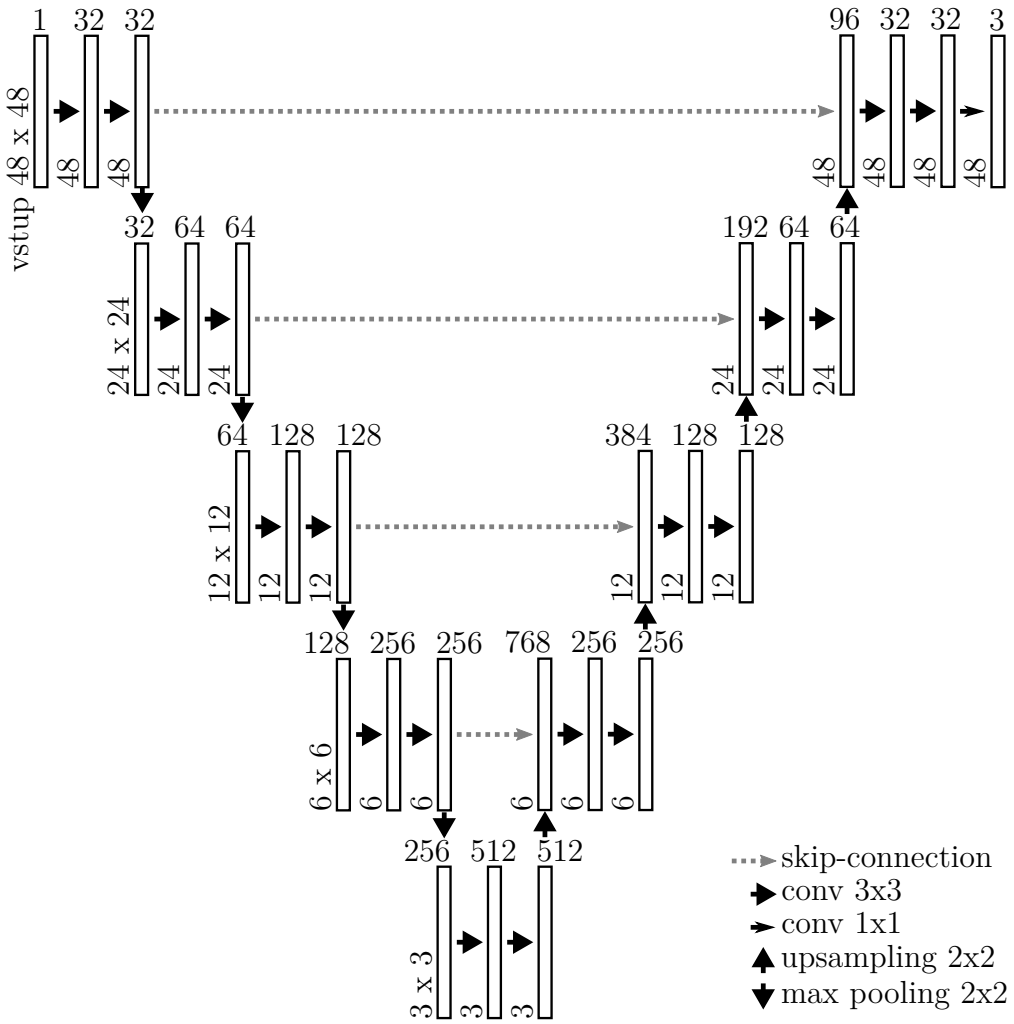
K problému segmentace je přistupováno jako k pixel-labeling problému, kde může být jeden pixel zařazen do více tříd, protože se předpokládá překrývající se rozvržení strany. Tato volba vychází i z vlastností datové sady, kde je na stranách dokumentů možné nalézt fotografie, které jsou popsané textem. Pro tento případ je požadováno, aby byl pixel klasifikován zároveň jako obrázek i text. Pro experimenty je využita vytvořená datová sada (viz sekce 4.6).

5.1 Architektura sítě

Architektura sítě je inspirována sítěmi U-Net [23] a FCN [25]. Síť FCN se liší od U-Net hlavně v tom, že neobsahuje tzv. skip-connections, které považují autoři za zbytečné pro segmentaci stran. Díky tomu ale může mít síť FCN problém v podobě nutného kompromisu mezi přesností a kontextem (viz sekce 2.2.3). Pokud jsou skip-connections opravdu zbytečné, lze předpokládat, že je natrénovaná síť bude ignorovat. Z tohoto důvodu vychází architektura sítě především z U-Net.

Síť je tvořena dle obrázku 5.1 kontrakční (vlevo) a expanzivní (vpravo) částí. Vstupem sítě je obrázek ve stupních šedi. Na obrázku 5.1 je zobrazen nejmenší požadovaný rozměr vstupu 48×48 pixelů. Rozměr je volen tak, aby v nejnižším konvolučním bloku byl rozměr alespoň 3×3. Další požadavek na rozměr sítě je, aby byl dělitelný 16. Požadavek plyne z použití max-poolingových a upsamplingových vrstev. Pro max-poolingovou vrstvu musí být vždy vstupní rozměr dělitelný dvěma, jinak dojde při následném upsamplingu k nekonzistenci rozměrů a následné chybě. Jelikož jsou použity čtyři max-poolingové vrstvy plyne dělitelnost 16 ze sekvence $16, 8, 4, 2, 1$. Validní rozměry pro vstup sítě jsou tedy dle vzorce 5.1.

$$x = 48 + i \cdot 16, \text{kde } i = 0, 1, 2, \ldots, \infty \tag{5.1}$$



Obrázek 5.1: Architektura sítě (příklad pro nejmenší požadovaný vstup 48×48): Box představuje vícekanálovou mapu příznaků. Rozměr je značen na levé straně boxu a počet kanálů je značen nad boxem.

Za vstupem následuje kontrakční část, která se skládá z konvolučních bloků následovaných 2×2 max-poolingovou vrstvou. Konvoluční bloky jsou složeny ze dvou konvolučních vrstev s ReLU aktivační funkcí, filtry velikosti 3×3 a je zde použit padding. První konvoluční vrstva má 32 filtrů a po každé max-poolingové vrstvě je jejich počet zdvojnásoben.

Na kontrakční část navazuje expanzivní část, která je tvořena konvolučními bloky následovanými 2×2 upsamplingovou vrstvou. Po každé upsamplingové vrstvě je počet filtrů v konvolučních vrstvách zmenšen na polovinu. Výstup upsamplingové vrstvy je spojen s výstupem konvolučního bloku předchozí úrovně kontrakční části. Tyto hodnoty jsou vstupem dalšího konvolučního bloku.

Poslední vrstvou sítě je konvoluční vrstva s aktivační funkcí sigmoid. Velikost filtru je 1×1 a počet filtrů odpovídá počtu tříd (text, obrázek a pozadí). Vstupem vrstvy jsou natrénované příznaky, které jsou převedeny na výstup. Výstupem je predikce celého vstupního obrázku, kde jednotlivé kanály odpovídají predikci pro konkrétní třídu. Díky použití sigmoid aktivační funkce je každá třída klasifikována zvlášť a je tedy možné, aby byl pixel klasifikován např. jako text a obrázek zároveň.

Ve fázi trénování jsou využity dropoutové vrstvy s parametrem $p = 0,2$ (pravděpodobnost 20 %, že je neuron deaktivován). Dropoutová vrstva je vždy mezi dvěma konvolučními vrstvami, které tvoří konvoluční blok.

Sdílení parametrů v rámci konvoluční vrstvy vede k výhodné vlastnosti plně konvolučních neuronových sítí, které umožňují na vstup sítě přivést obrázek s téměř libovolnými rozměry. Při volbě rozměrů je ale důležité brát ohled na hardwarovou náročnost a splňovat podmínky dané architekturou.

5.2 Načtení vstupního obrázku

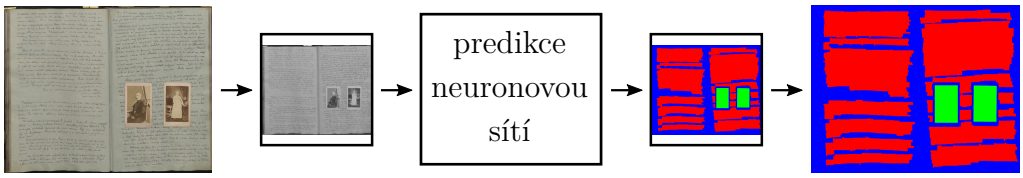
Vzhledem k vlastnostem plně konvoluční neuronové sítě není nutné specifikovat konstantní vstupní rozměr. Je však vhodné, aby rozměr odpovídal rozměrům používaným během trénování sítě (viz sekce 5.5). Toho je docíleno stanovením maximálního rozměru pro vstup sítě. Větší rozměr obrázku je převeden na stanovený maximální rozměr a druhý rozměr obrázku je dopočten v odpovídajícím poměru (viz obrázek 5.2).

Po zmenšení obrázku může nastat situace, že rozměry neodpovídají požadavkům na vstup sítě (viz sekce 5.1). To lze řešit doplněním okrajů (padding) nebo zvětšením či zmenšením tak, aby výsledné rozměry odpovídaly vzorci 5.1.

Během vyhodnocení výsledků pro již natrénovanou síť jsou oba přístupy srovnatelné a dosahují podobné úspěšnosti. Načítání obrázků ve fázi trénování je problematičtější (viz sekce 5.5.3) a jsou zde patrné rozdíly v úspěšnosti. Využití paddingu způsobuje problematickou segmentaci a klasifikaci např. černých ploch v obrázcích. Proto je řešena úprava rozměrů pro vstup sítě zmenšením či zvětšením stejně jako při trénování, není-li řečeno jinak (jiné postupy viz sekce 5.5.3).

5.3 Postup segmentace vstupního obrázku

Postup zpracování (viz obrázek 5.2) začíná načtením vstupního obrázku ve stupních šedi a úpravou jeho rozměrů pro vstup sítě dle sekce 5.2. Násle-



Obrázek 5.2: Postup vyhodnocení prototypem: Vstupní obrázek je zmenšen (box kolem zmenšeného obrázku představuje maximální rozměr pro vstup sítě) a zpracován ve výstup, který je následně zvětšen na původní rozměry obrázku.

duje predikce segmentace pomocí natrénované sítě založené na U-Net (viz sekce 5.1). Výstup sítě tvoří hodnoty poslední vrstvy s aktivační funkcí sigmoid. Tím se pro každou požadovanou třídu získá maska se stejnými rozměry jako rozměry vstupu sítě. Tyto masky se nejdříve zvětší na původní rozměry obrázku a následně jsou prahovány hodnotou 0,5. Výsledkem segmentace jsou binární masky pro jednotlivé třídy, jejichž rozměry odpovídají původnímu obrázku.

5.4 Vyhodnocení úspěšnosti

Pro vyhodnocení úspěšnosti jsou zpracovávány jednotlivé obrázky s jejich ground-truth. Pro vstupní obrázek je získána predikce segmentace podle sekce 5.3. Následně se porovnává ground-truth s výsledky predikce v původní velikosti. Porovnává se vždy po jednotlivých třídách, např. je nejprve zvlášť vyhodnocena úspěšnost pro text, poté je vyhodnocena pro obrázek a tak dále. Pro jeden obrázek a pro konkrétní metriku (viz sekce 5.4.1) jsou spočteny celkem čtyři hodnoty – pro třídy text, obrázek, pozadí a jejich průměr.

Postup vyhodnocení na testovací nebo validační sadě dat je podobný. Nejprve jsou zpracovány jednotlivé obrázky dané sady výše popsaným postupem. Poté jsou výsledky pro jednotlivé obrázky zprůměrovány v rámci konkrétní metriky a třídy. Např. je průměrována hodnota pro accuracy u třídy text přes všechny obrázky dané sady.

5.4.1 Použité metriky

Aby bylo možné prototyp systému objektivně vyhodnotit, je nutné zvolit vhodné metriky. Protože je k úloze přistupováno jako k pixel-labeling problému, je možné aplikovat pixelově založené přístupy. Jelikož je každý pixel klasifikován zda patří do dané třídy (0 – nepatří do třídy, 1 – patří do třídy), je možné využít např. metrik používaných pro vyhodnocení úspěš-

nosti binární klasifikace. V pracích [8, 18, 25] jsou popsány úpravy metrik pro *accuracy*, *precision*, *recall* a *F1 score* pro naši úlohu. Dalšími možnostmi jsou *Intersection over Union* a *Foreground Pixel Accuracy*. V této práci jsou vyhodnoceny všechny metriky, protože mohou mít jinou vypovídací schopnost.

Pro výpočet je požadován predikovaný výstup a správný výstup (ground-truth) pro konkrétní třídu např. text. Vyhodnocení probíhá porovnáváním predikce s ground-truth na úrovni pixelů. Při vyhodnocení mohou pro pixel nastat čtyři případy:

- TP (True Positive) – pixel je správně predikován jako 1
- TN (True Negative) – pixel je správně predikován jako 0
- FP (False Positive) – pixel je chybně predikován jako 1
- FN (False Negative) – pixel je chybně predikován jako 0

Pro výpočet dále popsaných metrik představují TP, TN, FP a FN počet pixelů patřících do dané množiny.

Accuracy počítá zastoupení správných predikcí dle vzorce 5.2. Jde vlastně o podíl správných predikcí se všemi predikcemi. Výsledek je možné chápat jako pravděpodobnost, že daný pixel bude správně klasifikován. Nejlepší hodnota je 1. Ta je dosažena v případě, že je každý pixel predikován správně. Naopak při chybné predikci všech pixelů je dosaženo nejhorší hodnoty 0.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.2}$$

Precision představuje přesnost a hodnota je počítána dle vzorce 5.3. Lze ji chápat jako pravděpodobnost, že je pixel predikovaný jako 1 predikován správně. Hodnoty se pohybují v rozmezí jako u accuracy. Je dobré si uvědomit, že např. při predikci jediného pixelu jako 1 a ostatních jako 0 je možné dosáhnout $precision = \frac{1}{1+0} = 1$.

$$precision = \frac{TP}{TP + FP} \tag{5.3}$$

Recall představuje úplnost a hodnota je spočtena dle vzorce 5.4 a je možné ji chápat jako pravděpodobnost, že je pixel označený v ground-truth jako 1 predikovaný správně. Hodnoty se opět pohybují od 0 do 1. Může nastat případ, že budou predikovány všechny pixely jako 1, potom bude výsledný $recall = \frac{n}{n+0} = 1$.

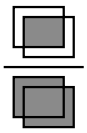
$$recall = \frac{TP}{TP + FN} \tag{5.4}$$

F1 score představuje harmonický průměr precision a recall. Hodnota F1 score je spočtena podle vzorce 5.5. Protože pro některé případy u precision a recall mohou být vypočtené hodnoty zavádějící, je vhodné uvažovat zároveň obě tyto metriky. Hodnoty jsou opět v rozmezí od 0 do 1.

$$F1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.5}$$

Intersection over Union (dále jako *IoU*) lze nalézt také pod pojmy jako Jaccard index nebo Jaccard similarity coefficient. IoU je využíváno pro porovnání podobnosti množin (viz vzorec 5.6). Toho lze využít, je-li na výstup nahlíženo jako na množinu. Potom je žádoucí, aby si predikovaná A a správná B množina byly co nejvíce podobné (viz obrázek 5.3). Pixelově založená mo-

$$IoU = \frac{A \cap B}{A \cup B} \tag{5.6}$$

$IoU =$ 

Obrázek 5.3: Vizualizace Intersection over Union

difikace je spočtena dle vzorce 5.7 a je podobná accuracy s tím rozdílem, že se neberou v úvahu *True Negatives*. Stejně jako v předchozích případech se hodnoty pohybují od 0 do 1.

$$IoU = \frac{TP}{TP + FP + FN} \tag{5.7}$$

Foreground Pixel Accuracy (dále jako *FgPA*) je prezentována v [25] a vychází z myšlenky prezentované na obrázku 5.4, kde je uvažován výsledek segmentace pouze pro popředí. Výpočet FgPA [25] je prakticky výpočet accuracy nad množinou pixelů popředí. Je počítán dle vzorce 5.8, kde je pro pixel na pozici x definováno δ_x a b_x:

- δ_x
  - hodnota 1 – souhlasí predikce s ground-truth
  - hodnota 0 – v ostatních případech

- b_x
  - hodnota 1 – pixel na pozici x je pixel popředí
  - hodnota 0 – v ostatních případech

Obrázek 5.4: Myšlenka Foreground Pixel Accuracy: Důležitá je segmentace popředí. [25]

Pro výpočet FgPA je však nutné zjistit pixely popředí a k tomu už může být potřeba i řešení segmentace. Pro řešení problému rozlišení popředí od pozadí je v této diplomové práci použita adaptivní binarizace navržená pro dokumenty [24]. Tím je zajištěna věrná aproximace popředí a není nutné pro každý obrázek manuálně vytvářet další informace o pixelech popředí. FgPA nabývá hodnot od 0 do 1 a v rámci práce je vyhodnoceno pro všechny třídy. Otázkou je, nakolik relevantní je např. pro pozadí úspěšnost na pixelech popředí.

$$FgPA = \frac{\sum_x b_x \cdot \delta_x}{\sum_x b_x} \tag{5.8}$$

5.5 Optimalizace parametrů systému

Stěžejním bodem pro výslednou funkčnost systému je úspěšné natrénování sítě, které je ovlivněno velkým množstvím nastavitelných parametrů a také zvolenými možnostmi pro jednotlivé části systému jako např. načtení vstupu sítě, trénování na celých stranách nebo pouze na jejich částech, rozšíření trénovací množiny, využití váhování chybové funkce.

Všechny části systému je nutné zvolit tak, aby bylo dosaženo co nejlepších výsledků. Bohužel návod k řešení neexistuje. Řešení je proto založeno na heuristice a experimentech, pro které je důležité zvolit vhodnou metriku pro porovnávání jednotlivých modelů.

5.5.1 Volba chybové funkce a její optimalizace

Za chybovou funkci (angl. *loss function*) je zvolena Binary Cross-Entropy loss [13] (také jako Sigmoid Cross-Entropy loss), protože je v poslední vrstvě

použita aktivační funkce sigmoid a zároveň se v práci jedná vlastně o binární klasifikaci, kde je pixel predikován jako patří/nepatří do dané třídy. Výpočet vychází ze vzorce 5.9, kde p je predikce a y je správná hodnota z ground-truth pro odpovídající třídu daného pixelu.

$$loss = -\frac{1}{N} \cdot \sum_{i=1}^{N} y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i) \tag{5.9}$$

Optimalizace chybové funkce je zajištěna iterativní úpravou vah pomocí algoritmu Adam [7], který je doporučován v [11]. Parametry jsou ponechány výchozí, protože souhlasí s doporučovanými [7, 11]. Adam je adaptivní algoritmus, který uchovává rychlost učení pro každý parametr (váhu) sítě a tyto rychlosti upravuje podle vývoje trénování. Mezi hlavní výhody patří výpočetní efektivita, malé paměťové nároky a nízká potřeba lazení hyper-parametrů algoritmu. V porovnání s ostatními lze Adam považovat za efektivní, protože dosahuje velice dobrých výsledků a rychleji.

5.5.2 Stanovení doby trénování

Vhodná délka trénování je důležitá pro dosažení dobrých výsledků a zároveň zamezení přetrénování sítě. Protože není známo, jaká je optimální délka trénování této sítě na daných datech s konkrétními parametry, je využito techniky *early stopping*. Early stopping je technika, používaná pro ukončení tréninku, pokud se dále nezlepšuje monitorovaná metrika.

Zde se objevuje problém, jakou metriku zvolit. Možností by bylo vytvoření složené metriky z váhovaných metrik prezentovaných v sekci 5.4.1, ale otázkou je, jaké váhy zvolit. Na druhou stranu je v [18] zmíněna metrika IoU (viz sekce 5.4.1) jako vhodná pro segmentaci. IoU porovnává podobnost množin a v určitém smyslu vlastně představuje vizuální podobnost. Zároveň v rámci experimentů měla subjektivně největší vypovídací schopnost o kvalitě segmentace. Proto je jako monitorovaná metrika zvolena IoU.

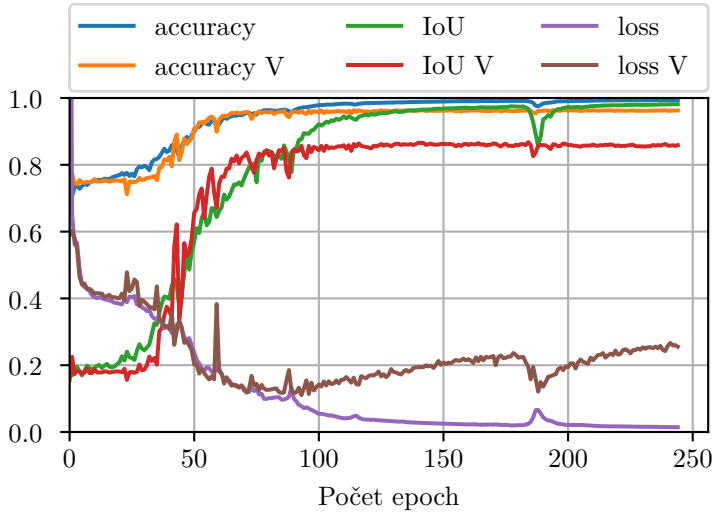
Během trénování je po každé epoše vyhodnocena metrika IoU na validační sadě (viz IoU V na obrázcích 5.5, 5.6 a 5.8). Pokud se tato metrika nezlepšuje po dobu alespoň 30 epoch (v případě menší množiny trénovacích dat až 60 epoch) je trénování ukončeno. Během trénování jsou váhy ukládány a pro vyhodnocení jsou použity ty s nejlepší dosaženou hodnotou IoU na validační sadě.

5.5.3 Analýza možností načítání

Pro vyhodnocení je nutné obrázek nejprve zmenšit pro vstup sítě. Důvodem jsou časové a hardwarové nároky. Navíc je pravděpodobné, že rozlišení obrázků stran přesahující např. 5000×4000 pixelů je zbytečně vysoké pro segmentaci na text, obrázky a pozadí. Za dostatečné lze předpokládat rozlišení 260×390 pixelů používané pro vstup sítě FCN [25] (viz sekce 2.2.4). Dalším důvodem je různorodost rozlišení, která může vést k situacím, kdy je stejné písmeno jednou vysoké 200 pixelů a podruhé např. 20 pixelů. To může způsobit problémy během trénování i testování (vyhodnocení úspěšnosti) sítě. Úpravou rozměrů lze tedy docílit i částečného sjednocení např. velikosti písmen. Kvůli hardwarovým nárokům je pro použitou grafickou kartu NVIDIA GeForce GTX 770 a zvolenou architekturu na základě experimentů stanoveno 512×512 pixelů jako horní omezení vstupu (a zároveň výstupu) pro fázi trénování.

Padding

První možností načítání vstupu je zmenšení obrázku tak, aby odpovídal danému limitu. Následně lze doplnit prázdná místa bezvýznamovými nulami (*padding*). Stejným způsobem je upraveno i ground-truth. Takto upravený obrázek i ground-truth je možné vyhodnotit celé najednou. Průběh trénování



Obrázek 5.5: Průběh trénování s využitím metody *padding* pro načítání vstupu: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

je zobrazen na obrázku 5.5 a dosažené výsledky jsou rozepsány v tabulce 5.1.

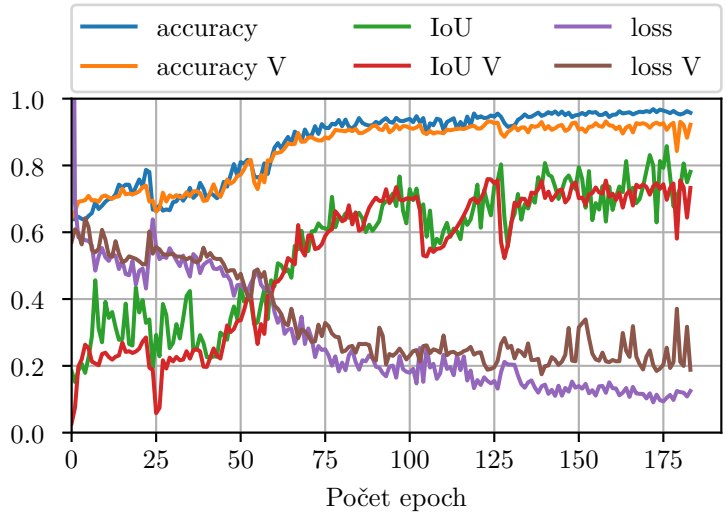
S touto možností byly prováděny první experimenty, kde se však ukázalo, že vzhledem k vyplňování prázdných míst černou barvou může u některých obrázků docházet k chybné segmentaci černých ploch.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.946	0.941	0.914	0.927	0.865	0.989
Obrázky	0.971	0.804	0.879	0.834	0.721	0.904
Pozadí	0.920	0.909	0.942	0.925	0.860	0.899
Průměr	0.946	0.885	0.912	0.895	0.815	0.931

Tabulka 5.1: Dosažené výsledky na validační sadě s využitím metody *padding* pro načítání vstupu

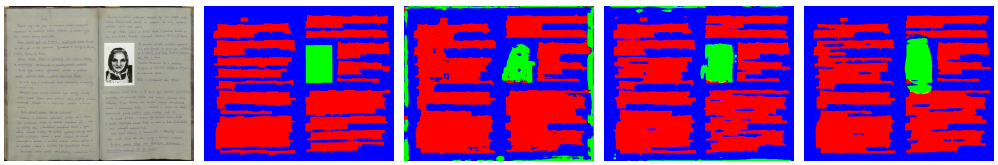
Crop

Dalším řešením je využití vlastnosti sdílených parametrů konvolučních vrstev a trénovat síť pouze na částech stran. Tím se lze vyhnout doplňování prázdných míst a s tím spojeným problémům. Vstupní obrázek je nejprve načten v původním rozlišení (sjednocená maximální možná velikost datové sady je 1024×1024 pixelů). Z načteného obrázku je následně vyříznuta (*crop*) oblast 512×512 pixelů. Stejným způsobem je vyříznuta odpovídající oblast



Obrázek 5.6: Průběh trénování s využitím metody *crop* pro načítání vstupu: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

v ground-truth. Ačkoliv jsou pro fázi trénování (průběh viz obrázek 5.6) použity oblasti s rozměry 512×512 pixelů, je možné následně predikovat celé obrázky v původním rozlišení 1024×1024 pixelů, protože konvoluční vrstva parametry sdílí a ty jsou zároveň natrénované na strany s tímto rozlišením.



(a) (b) (c) (d) (e)

Obrázek 5.7: Příklady predikce: vstupní obrázek (a), ground-truth (b), predikce s využitím metody *crop* (c), predikce s využitím metody *padding* (d) a predikce s využitím metody *resize* (e)

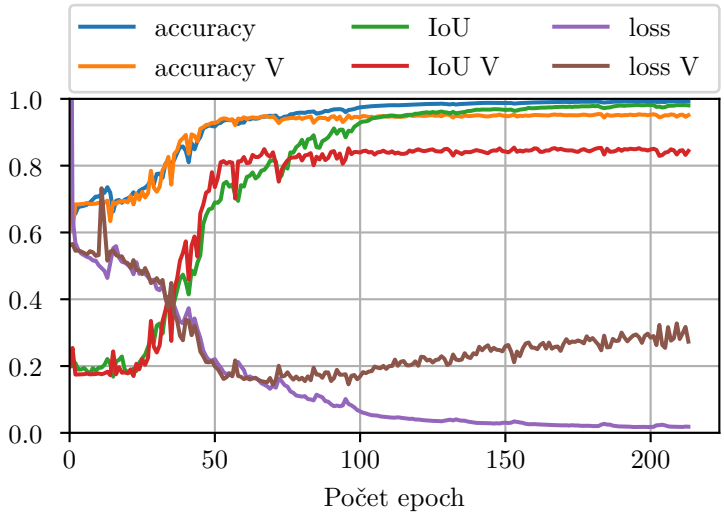
Během trénování však mohou nastat situace, kdy je vyříznuta například oblast, která neobsahuje obrázky nebo text. Potom mohou být váhy upraveny nevhodně. Dalším problémem může být využití paddingu u konvolučních vrstev v architektuře sítě (viz 5.1). Způsob trénování u metody *crop* je totiž podobný tzv. overlap-tile strategy (viz obrázek 2.15), kde se padding nepoužívá. Pro predikci celých stran je ale padding považován za vhodný. První důvod je stejná velikost výstupu. Druhý skrytý důvod je, že padding může poskytovat informaci o kontextu ve smyslu pozice u okraje stránky. Díky tomu je možné s větší pravděpodobností predikovat pixel na okraji stránky např. jako pozadí. V případě *crop* může být během trénování toto chování potlačeno, čímž je možné vysvětlit horší výsledky (viz tabulka 5.2) a velkou chybovost při okrajích stránky (viz obrázek 5.7.c).

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.947	0.940	0.916	0.927	0.866	0.985
Obrázky	0.948	0.646	0.899	0.729	0.593	0.865
Pozadí	0.899	0.934	0.871	0.900	0.819	0.848
Průměr	0.931	0.840	0.895	0.852	0.759	0.899

Tabulka 5.2: Dosažené výsledky na validační sadě s využitím metody *crop* pro načítání vstupu

Resize

Další možností načítání vstupu je podobně jako u metody padding zmenšení obrázku tak, aby odpovídal danému limitu. Pro splnění podmínek da-



Obrázek 5.8: Průběh trénování s využitím metody *resize* pro načítání vstupu: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

ných architekturou sítě je upraveno rozlišení obrázku. Výpočet rozlišení pro vstup sítě se skládá ze dvou kroků:

1. úprava rozlišení – rozlišení delší strany obrázku je zmenšeno na horní limit 512 pixelů a rozlišení druhé strany obrázku je zmenšeno v odpovídajícím poměru

2. korekce rozlišení – rozlišení je upraveno tak, aby splňovalo požadavky dané architekturou dle vzorce 5.1 v sekci 5.1 – v tomto kroku je možná mírná deformace obrázku, ale maximální možná změna rozměru je pouze ±8 pixelů

Po výpočtu požadovaného rozlišení jsou obrázek i ground-truth na toto rozlišení převedeny.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.942	0.934	0.909	0.921	0.855	0.982
Obrázky	0.985	0.884	0.931	0.903	0.827	0.985
Pozadí	0.934	0.936	0.939	0.937	0.882	0.989
Průměr	0.953	0.918	0.926	0.920	0.855	0.985

Tabulka 5.3: Dosažené výsledky na validační sadě s využitím metody *resize* pro načítání vstupu

Během trénování (průběh viz obrázek 5.8) jsou na vstup přiváděny obrázky s různým rozlišením. Tím je umožněna predikce celé strany a lze využít informací poskytovaných díky paddingu v konvolučních vrstvách. Na obrázku 5.7 je patrný šum na okraji stran u předchozích metod. Metoda *resize* tento problém eliminuje. Dále není nutné rozměry doplňovat konkrétní hodnotou a nevznikají s tím související problémy.

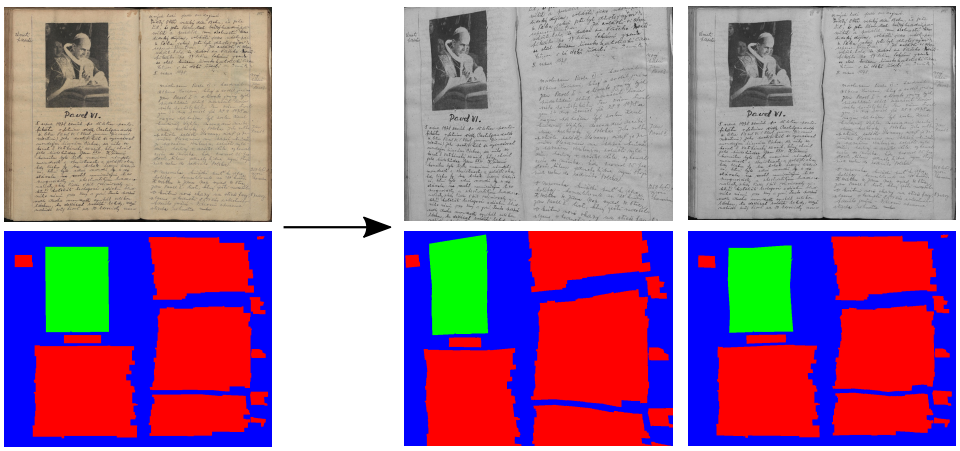
Na základě vlastností a výsledků (viz tabulka 5.3) se jako nejvhodnější varianta ukázalo řešení založené na změně rozlišení vstupního obrázku (*resize*).

5.5.4 Možnosti augmentace trénovacích dat

Augmentace dat je technika sloužící k rozšíření trénovacích dat z již dostupných dat. Pro dobrou výkonnost je klíčové mít dostatečně velkou datovou sadu. Datové sady např. pro klasifikaci obrázků běžně obsahují stovky vzorů. Zde však máme v trénovací sadě k dispozici pouze šest obrázků stran kronik s obrázky. Vzhledem k nízkému počtu obrázků v trénovací sadě je augmentace vhodná pro rozšíření trénovací množiny.

Z průběhu trénování na obrázku 5.8 je navíc patrné, že ačkoliv se chybová funkce (*loss*) na trénovací sadě zmenšuje, na validační sadě se naopak zvětšuje. Jedná se zřejmě o přetrénování sítě. Vedle využití dropoutové vrstvy je augmentace další možností jak přetrénování sítě zabránit, nebo ho alespoň redukovat.

U běžných obrázků se pro augmentaci používá např. posun, rotace, zrcadlení a výřezy. Pro obrázky stran dokumentů však nemusí být všechny možnosti vhodné. Např. zrcadlení nemusí být vhodné vzhledem k čitelnosti textu.



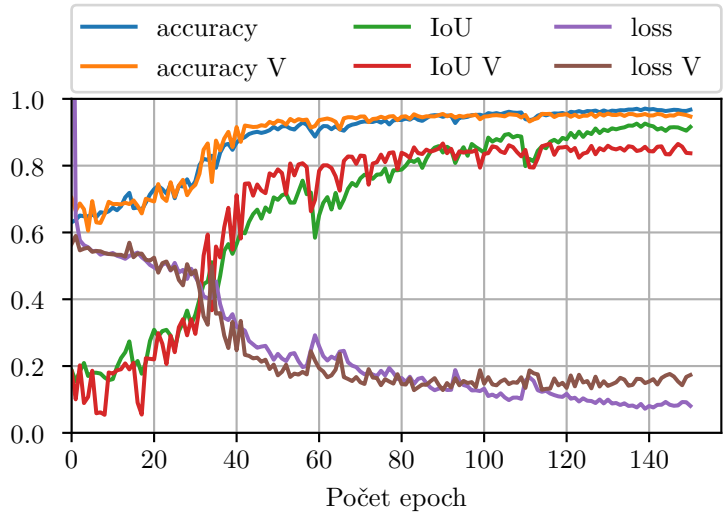
Obrázek 5.9: Vizualizace výsledku augmentace

Pro augmentaci je využita knihovna Augmentor [6]. Nastavení je voleno tak, aby z výsledku augmentace bylo stále patrné, že se jedná a stranu dokumentu. Na základě provedených experimentů a vizuální kontroly jsou použity metody *random_distortion*, *skew* a *rotate*. Kde metoda *random_distortion* provádí náhodné elastické zkreslení řízené mřížkou a magnitudou. Metoda *skew* provádí náhodné zkosení pomocí perspektivní transformace obrázku. Metoda *rotate* je použita k náhodnému otočení obrázku v daném rozmezí. Po velkém množství experimentů s knihovnou jsou parametry zvoleny následovně:

- random_distortion:
  - probability=0.8 – pravděpodobnost aplikování
  - grid_width=8 – mřížka 8×8
  - grid_height=8 – mřížka 8×8
  - magnitude=(minsize // 100) – představuje sílu zkreslení a je spočtena celočíselným dělením na základě menšího rozměru obrázku, aby byl vizuálně podobný výstup pro různě velké obrázky během experimentování
- skew:
  - probability=0.4 – pravděpodobnost aplikování
  - magnitude=0.18 – hodnota pro maximální zkosení
- rotate:
  - probability=0.9 – pravděpodobnost aplikování
  - max_left_rotation=5 – maximum pro levou rotaci je 5 stupňů
  - max_right_rotation=5 – maximum pro pravou rotaci je 5 stupňů

Výsledek s tímto nastavením je naznačen na obrázku 5.9. Augmentace je prováděna až během fáze trénování, kdy jsou data průběžně načítána odpovídajícím způsobem jako při použití metody *resize* v sekci 5.5.3. Během tohoto procesu je stejný postup augmentace aplikován na obrázek i ground-truth zároveň.

Z průběhů trénování bez augmentace (viz obrázek 5.8) a s augmentací (viz obrázek 5.10) je patrné, že při trénování s augmentací nedochází k tak velkému odchýlení chybové funkce na validační a trénovací sadě. Na základě výsledků (viz tabulka 5.4) lze pozorovat mírné zlepšení.



Obrázek 5.10: Průběh trénování s využitím *augmentace* trénovací sady: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

Na druhou stranu je zde větší množství chyb u okrajů stránky, které jsou pravděpodobně způsobené augmentací. Díky augmentaci se při trénování objevují vzory, kde je text či obrázek přímo u okraje vstupu sítě. To může způsobovat podobné problémy jako u *crop* v sekci 5.5.3.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.944	0.927	0.924	0.924	0.862	0.980
Obrázky	0.986	0.926	0.920	0.922	0.856	0.986
Pozadí	0.935	0.943	0.936	0.939	0.885	0.985
Průměr	0.955	0.932	0.927	0.928	0.867	0.984

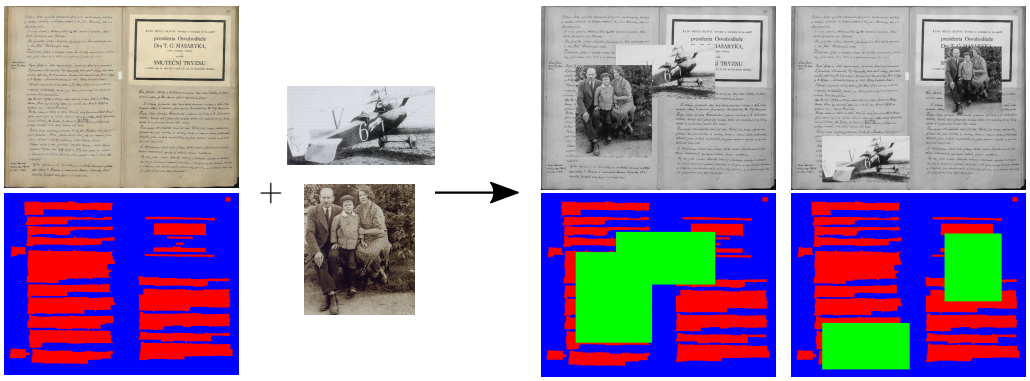
Tabulka 5.4: Dosažené výsledky na validační sadě s využitím *augmentace* trénovací sady

5.5.5 Možnosti automatické tvorby nových trénovacích dat

Další možnost rozšíření vhodných trénovacích dat spočívá ve využití stran kronik obsahujících pouze text. Tyto strany je možné rozšířit o dostupné obrázky získané v rámci přípravy datové sady (viz sekce 4.6). Tvorba nových trénovacích dat je znázorněna na obrázku 5.11 a postup je následující:

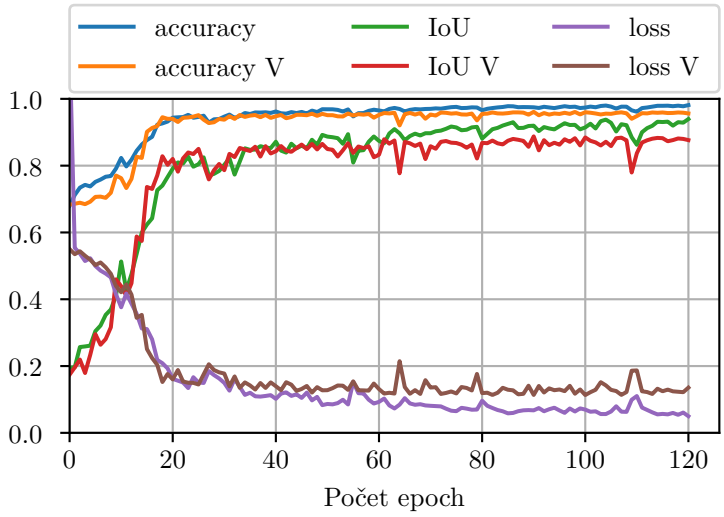
1. načtení obrázku a ground-truth způsobem odpovídajícím metodě *resize* v sekci 5.5.3
2. náhodná volba počtu vkládaných obrázků v rozmezí od jednoho do tří
3. vložení obrázků a úprava ground-truth, kde postup pro každý vkládaný obrázek je následující:
   (a) náhodná volba obrázku pro vložení
   (b) náhodná změna rozlišení vkládaného obrázku tak, aby jeho větší rozměr:
      - nepřesahoval 60 % menšího rozměru obrázku stránky
      - byl větší než 20 % menšího rozměru obrázku stránky, pokud to jeho originální rozlišení umožňuje
   (c) náhodná volba umístění oblasti pro vložení obrázku, která je alespoň 8 pixelů od okraje obrázku stránky
   (d) vložení obrázku na zvolenou oblast stránky
   (e) úprava zvolené oblasti v ground-truth

Tento postup je podobně jako u augmentace použit až během fáze trénování, kdy jsou data načítána průběžně.



Obrázek 5.11: Vizualizace tvorby nových trénovacích dat

U průběhu trénování (viz obrázek 5.12) je docíleno podobného efektu jako s augmentací. Nedochází tedy k nárůstu chybové funkce na validační sadě v době, kdy je na trénovací sadě chyba snižována. V porovnání s předchozími možnostmi je dosaženo nejlepších výsledků (viz tabulka 5.5).



Obrázek 5.12: Průběh trénování s využitím vytváření nových trénovacích dat: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

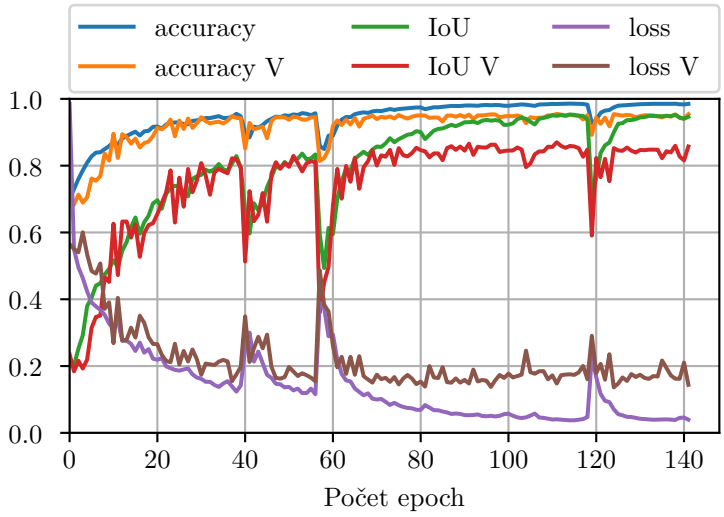
	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.950	0.943	0.923	0.932	0.875	0.991
Obrázky	0.990	0.932	0.956	0.943	0.893	0.994
Pozadí	0.943	0.946	0.948	0.946	0.899	0.991
Průměr	0.961	0.940	0.943	0.940	0.889	0.992

Tabulka 5.5: Dosažené výsledky na validační sadě s využitím vytváření nových trénovacích dat

5.5.6 Rozšíření o tištěná data

Další experiment spočívá v rozšíření trénovací množiny o stránky dokumentů s tištěným textem, které jsou součástí připravené datové sady (viz sekce 4.6).

Průběh trénování (viz obrázek 5.13) vykazuje výkyvy v průběhu epoch a oproti předchozím případům je poměrně nestálý. Důvodem může být velká různorodost trénovacích dat. Ačkoliv tvoří trénovací množinu z velké části data odlišná (viz např. obrázky 3.3 a 4.1) od dat validačních, jsou dosažené výsledky prezentované v tabulce 5.6 velmi dobré, ale oproti předchozím případům se v predikcích objevuje větší množství šumu.



Obrázek 5.13: Průběh trénování s rozšířením trénovacích dat o strany s tištěným textem: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.943	0.944	0.898	0.919	0.854	0.986
Obrázky	0.988	0.962	0.903	0.930	0.870	0.990
Pozadí	0.936	0.921	0.963	0.941	0.889	0.988
Průměr	0.955	0.942	0.921	0.930	0.871	0.988

Tabulka 5.6: Dosažené výsledky na validační sadě s rozšířením trénovacích dat o strany s tištěným textem

5.5.7 Možnosti váhování chybové funkce

Další experiment spočívá ve váhování chybové funkce. Změní-li se chybová funkce, změní se i výsledek její optimalizace při trénování. Toho lze využít např. při nerovnoměrně rozložené trénovací sadě, kde je jedna třída zastoupená mnohem větším počtem vzorů. Nebo lze zajímavějším vzorům přiřadit větší význam. V rámci této práce jsou provedeny experimenty s váhováním kanálů a oblastí oddělujících jednotlivé komponenty.

Váhování kanálů

Ve většině obrázků stran dokumentů je různé zastoupení pixelů představujících např. obrázek. Většinou je počet pixelů spadajících do třídy obrázek menší než počet pixelů, které do této třídy nespadají. To může během tré-

nování sítě vést k tomu, že síť predikuje třídu obrázek s menší pravděpodobností, což ale nemusí být požadované chování. Řešením může být váhování chybové funkce.

Chybová funkce je váhována podle jednotlivých kanálů tak, aby bylo váhované zastoupení tříd rovnoměrné. Výpočet vah vychází z ground-truth, který se skládá z jednotlivých kanálů (představuje ground-truth pro jednu konkrétní třídu). Každý kanál obsahuje hodnoty 0 a 1, které značí příslušnost k dané třídě. Proto je každý kanál zpracován samostatně.

Výpočet vah pro kanál je proveden na základě ground-truth podle vzorce 5.10, kde w_0 představuje váhu pro třídu 0, w_1 představuje váhu pro třídu 1, c_0 představuje počet pixelů označených v daném kanálu ground-truth jako 0 a c_1 představuje počet pixelů označených v daném kanálu ground-truth jako 1.

$$w_0 = \frac{c_0 + c_1}{2 \cdot c_0} \tag{5.10a}$$

$$w_1 = \frac{c_0 + c_1}{2 \cdot c_1} \tag{5.10b}$$

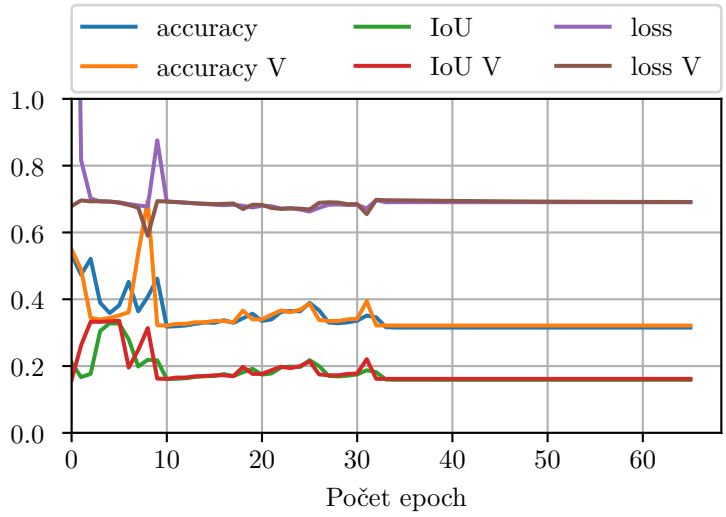
Příklad je uveden pro obrázek který obsahuje 1000 pixelů, kde je 250 pixelů označených jako obrázek (c_1) a 750 pixelů označených není (c_0). Váhy jsou spočteny jako $w_0 = \frac{750+250}{2 \cdot 750} = \frac{2}{3}$ a $w_1 = \frac{750+250}{2 \cdot 250} = 2$. Tím je dle vzorce 5.11 docíleno rovnoměrného váhovaného zastoupení tříd $c_0 \cdot w_0 = 750 \cdot \frac{2}{3} = 500 = 250 \cdot 2 = c_1 \cdot w_1$.

$$c_0 \cdot w_0 = c_0 \cdot \frac{c_0 + c_1}{2 \cdot c_0} = \frac{c_0 + c_1}{2} = c_1 \cdot \frac{c_0 + c_1}{2 \cdot c_1} = c_1 \cdot w_1 \tag{5.11}$$

Výsledným efektem je, že chybné označení pixelu jako obrázek má menší význam než jeho neoznačení, které vede k větší chybě. Ta je při při zpětném šíření chyby ve fázi trénování propagována zpět a hraje tedy větší roli při optimalizaci parametrů sítě. Tím je tedy řešeno nerovnoměrné zastoupení tříd, protože více pixelů s menší chybou může mít stejný vliv jako menší počet pixelů s větší chybou.

Při praktickém použití se však tímto způsobem nepodařilo síť uspokojivě natrénovat (viz obrázek 5.14). Neuspokojivé výsledky (viz tabulka 5.7) tedy nejsou překvapivé i vzhledem k tomu, že síť predikuje prakticky pro každý pixel stejné výsledky.

Po časově náročných kontrolách kódu, testování správnosti implementace a dlouhých úvahách byla jako nejpravděpodobnější příčina určena absence limitu tohoto váhování. Vzory v trénovací množině obsahují obrázek jen na zlomku plochy. Poté mohou být váhy příliš vysoké a působit optimalizátoru chybové funkce problémy. Výsledkem může být optimalizace, kde je



Obrázek 5.14: Průběh trénování s váhováním chybové funkce po kanálech bez limitu: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené V) části

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.387	0.383	1.000	0.552	0.383	0.415
Obrázky	0.100	0.094	1.000	0.170	0.094	0.396
Pozadí	0.481	1.000	0.012	0.023	0.012	0.771
Průměr	0.322	0.492	0.671	0.249	0.163	0.527

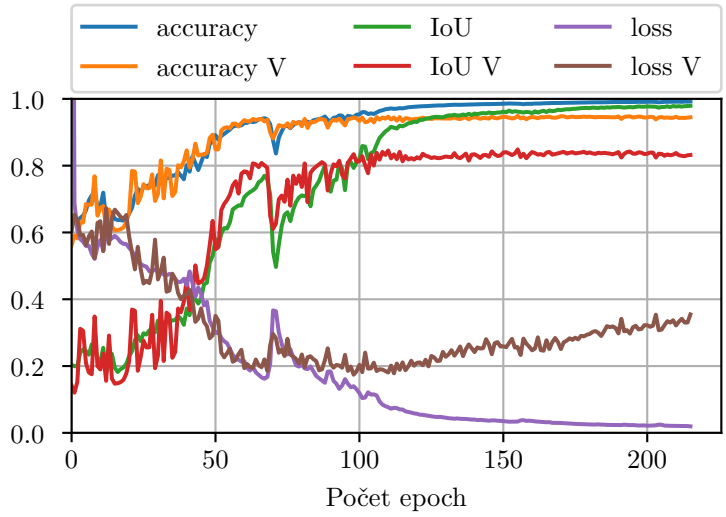
Tabulka 5.7: Výsledky na validační sadě s váhováním chybové funkce po kanálech bez limitu

predikován stejný výsledek jako optimum k chybové funkci, která je však rovnoměrně váhovaná. To může být důvod, proč je každý pixel označen jako obrázek, přestože je obrázků ve vzorech menší množství.

$$0,67 \leq w \leq 2 \tag{5.12}$$

Kvůli těmto důvodům byl zvolen limit pro vypočtené váhy dle vzorce 5.12. Zvolené hodnoty vycházejí z výše popsaného příkladu a limitují hodnoty, které by byly spočteny při velmi nerovnoměrném zastoupení tříd pro obrázek: méně než $\frac{1}{4}$ nebo více než $\frac{3}{4}$ plochy obrázku.

Po této úpravě již bylo možné síť natrénovat (průběh viz obrázek 5.15). Dosažené výsledky (viz tabulka 5.8) jsou však srovnatelné s výsledky bez váhování (viz tabulka 5.3).



Obrázek 5.15: Průběh trénování s váhováním chybové funkce po kanálech s limitem: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

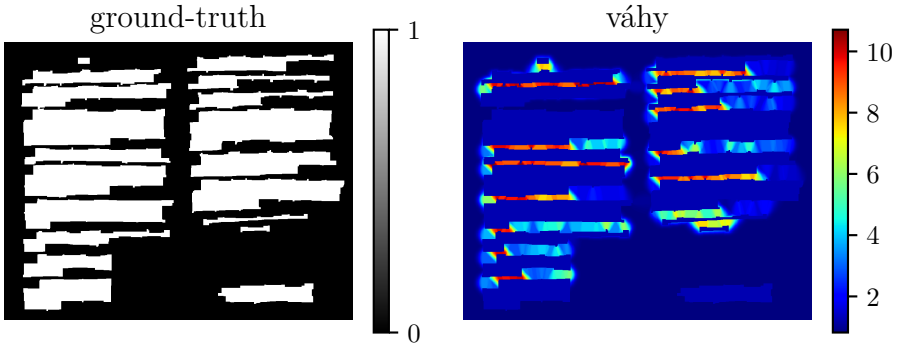
	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.936	0.941	0.885	0.910	0.838	0.980
Obrázky	0.983	0.918	0.905	0.910	0.835	0.986
Pozadí	0.928	0.916	0.953	0.934	0.876	0.984
Průměr	0.949	0.925	0.914	0.918	0.850	0.983

Tabulka 5.8: Dosažené výsledky na validační sadě s váhováním chybové funkce po kanálech s limitem

Oddělení komponent

Tato část se snaží čelit problému nedostatečného oddělování komponent jako např. jednotlivých odstavců textu. V predikované segmentaci se objevuje slévání jednotlivých komponent, což zamezuje jednoduché možnosti určení oblasti odstavce např. pomocí určení vnější hranice komponenty, protože dojde k tomu, že v jedné oblasti bude více odstavců. Problém se prohlubuje i vzhledem k ručně psaným dokumentům, které nemají pevně danou strukturu. Navíc zde chybí i další vertikální či horizontální oddělovače, které je možné spatřit u tištěných dokumentů a kterých by tedy bylo možné využít. Další problém plyne i z anotovaných obrázků stran dokumentů, kde se oblasti jednotlivých odstavců mohou překrývat, protože se překrývají jejich řádky či některá písmena.

Řešení problému je proto z výše popsaných důvodů velice obtížné. Je však možné řešit oddělení alespoň nepřekrývajících se komponent a to díky využití váhování chybové funkce, které lze provést podobně jako v práci [23], kde je zvýšena váha oblastem, které oddělují jednotlivé komponenty.



Obrázek 5.16: Příklad vypočtených vah pro oddělení komponent (zároveň jsou pro lepší vizualizaci váhované třídy)

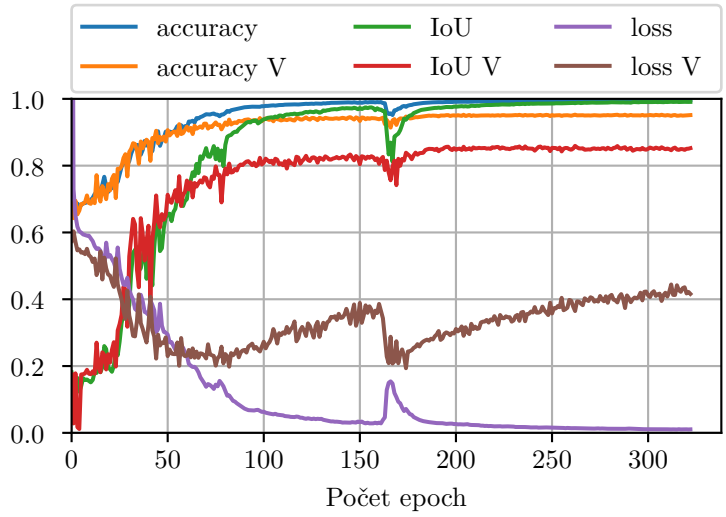
Výpočet vah je prováděn podobně jako u váhování kanálů vždy pro jednotlivý kanál. Pro vysvětlení postupu výpočtu vah je nutné nejprve nadefinovat několik pojmů. Na základě obrázku 5.16 se pozadím rozumí černá oblast v *ground-truth* a komponentou se rozumí souvislá bílá oblast v *ground-truth* představující např. jeden odstavec.

Výpočet vah je založen na výpočtu prezentovaném v práci [23]. Váhy jsou spočteny na základě vzorce 5.13, kde x představuje pozici pixelu, $w_c(x)$ je původní váha, $d_1(x)$ představuje vzdálenost k nejbližší komponentě a $d_2(x)$ představuje vzdálenost k druhé nejbližší komponentě. Pro parametr w_0 je zvolena hodnota 10 a parametr σ je navýšen na hodnotu 10 oproti hodnotě prezentované v práci [23] kvůli možným větším mezerám mezi komponentami. Spočtená váha pro pixely oddělující komponenty je přidána pouze pixelům pozadí. Toho je docíleno násobením $(1-\mathrm{gt}(x))$, kde $\mathrm{gt}(x)$ představuje hodnotu pixelu v *ground-truth*.

$$w(x) = w_c(x) + w_0 \cdot \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right) \cdot (1 - \mathrm{gt}(x)) \tag{5.13}$$

Za $w_c(x)$ je možné dosadit hodnotu 1 nebo hodnoty získané váhováním kanálu pro vyrovnání zastoupení tříd. Takto získané váhy pro jednotlivé pixely kanálu je možné vidět na obrázku 5.16.

Pro trénování v této části je za $w_c(x)$ dosazena hodnota 1 a výpočet vah pro jednotlivé obrázky je proveden pro kanály představující třídy text



Obrázek 5.17: Průběh trénování s váhováním chybové funkce pro oddělení komponent: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

a obrázek. Výpočet je prováděn až v rámci průběžného načítání obrázků během trénování sítě.

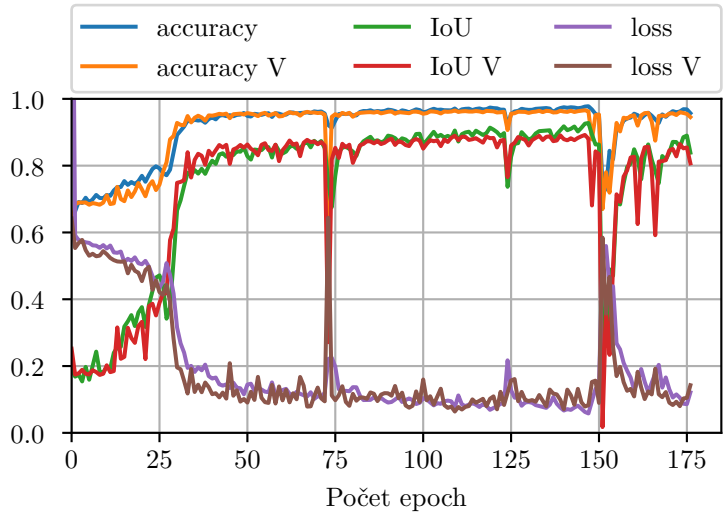
Dosažené výsledky (viz tabulka 5.9) jsou lepší než výsledky bez váhování chybové funkce (viz tabulka 5.3). Zhoršení však nastalo u *recall*, což odpovídá vzhledem k situaci, že zde dochází během predikce častěji k oddělování v místech, kde by podle ground-truth být nemělo.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.939	0.959	0.875	0.914	0.843	0.991
Obrázky	0.987	0.962	0.888	0.919	0.853	0.991
Pozadí	0.932	0.918	0.959	0.937	0.882	0.989
Průměr	0.953	0.946	0.907	0.923	0.859	0.990

Tabulka 5.9: Dosažené výsledky na validační sadě s váhováním chybové funkce pro oddělení komponent

5.5.8 Možnosti vyššího vstupního rozlišení

V této části je proveden experiment s vyšším vstupním rozlišením obrázku strany, protože se očekává, že výstup s vyšším rozlišením bude více detailní a proto by mohl dosáhnout lepší úspěšnosti. V předchozích případech



Obrázek 5.18: Průběh trénování s omezením vstupního rozměru na 1024 pixelů: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené *V*) části

je rozměr obrázku omezen 512 pixely. Zde je omezení změněno na 1024 pixelů. Vzhledem k paměťové náročnosti byla síť trénována na procesoru, proto byla doba trénování 14 hodin.

Ve fázi trénování (průběh viz obrázek 5.18) bylo využito trénovací sady a její rozšíření o kroniky obsahující pouze text, které jsou následně doplněny o obrázky dle sekce 5.5.5. Dále je aplikována augmentace dle sekce 5.5.4. Dosažené výsledky jsou prezentovány v tabulce 5.10. Pro porovnání se stejným postupem trénování při omezení rozměrů na 512 pixelů jsou přiloženy výsledky v tabulce 5.11.

Na základě porovnání výsledků v tabulkách 5.10 a 5.11 není dosaženo významného zlepšení a zároveň se zvětšuje výpočetní náročnost. Dále je z důvodu zvýšeného rozlišení možné snížení kontextu, protože architektura sítě zůstává stejná, ale oblast pro stejný kontext je dvakrát větší.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.960	0.949	0.945	0.947	0.900	0.992
Obrázky	0.987	0.936	0.923	0.927	0.865	0.993
Pozadí	0.951	0.956	0.952	0.953	0.911	0.990
Průměr	0.966	0.947	0.940	0.942	0.892	0.992

Tabulka 5.10: Dosažené výsledky s omezením vstupního rozměru na 1024 pixelů

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.952	0.940	0.934	0.936	0.881	0.991
Obrázky	0.988	0.946	0.939	0.941	0.888	0.993
Pozadí	0.943	0.953	0.941	0.946	0.898	0.990
Průměr	0.961	0.946	0.938	0.941	0.889	0.991

Tabulka 5.11: Dosažené výsledky s omezením vstupního rozměru na 512 pixelů

5.6 Analýza dosažených výsledků

Výsledky dosažené v rámci experimentů jsou shrnuty v tabulce 5.12, kde *padding*, *crop* a *resize* představují metody načítání vstupu pro trénování sítě dle sekce 5.5.3. Z výsledků je jednoznačně nejlepší metoda *resize*, která oproti ostatním metodám načítání vstupu umožňuje eliminaci problému se šumem na okraji stran.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Padding 5.5.3	0.946	0.885	0.912	0.895	0.815	0.931
Crop 5.5.3	0.931	0.840	0.895	0.852	0.759	0.899
Resize 5.5.3	0.953	0.918	0.926	0.920	0.855	0.985
Resize (referenční) 5.5.3	0.953	0.918	0.926	0.920	0.855	0.985
Augmenace 5.5.4	0.955	0.932	0.927	0.928	0.867	0.984
Tvorba nových dat 5.5.5	0.961	0.940	0.943	0.940	0.889	0.992
Tištěná data 5.5.6	0.955	0.942	0.921	0.930	0.871	0.988
Váhování kanálů 5.5.7	0.949	0.925	0.914	0.918	0.850	0.983
Oddělení komp. 5.5.7	0.953	0.946	0.907	0.923	0.859	0.990
Vstup 512 pixelů 5.5.8	0.961	0.946	0.938	0.941	0.889	0.991
Vstup 1024 pixelů 5.5.8	0.966	0.947	0.940	0.942	0.892	0.992

Tabulka 5.12: Souhrn dosažených výsledků (průměr): Pro identifikaci metody je dostupné klíčové slovo (slova) a sekce, kde je experiment proveden. Porovnávání je možné v rámci daného bloku (odděleny horizontální čarou), protože mezi bloky nemusejí být dodrženy stejné podmínky.

Resize (referenční) je shodný s *resize* a slouží jako referenční metoda pro ostatní metody v bloku tabulky 5.12.

Augmenace je metoda pro rozšíření trénovací sady prezentovaná v sekci 5.5.4. V porovnání s referenční metodou je patrné mírné zlepšení a lze tak metodu doporučit.

Tvorba nových dat představuje metodu pro rozšíření trénovací množiny o automaticky generovaná data dle sekce 5.5.5. Oproti referenční metodě je

dosaženo výrazného zlepšení a je tedy vhodná.

Tištěná data v tabulce 5.12 značí rozšíření trénovací množiny o strany tištěných dokumentů dle sekce 5.5.6. Díky zlepšení oproti referenční metodě lze využít rozšíření trénovací množiny o tištěné strany.

Váhování kanálů je metoda pro vyrovnání rozdílů v zastoupení tříd prezentovaná v sekci 5.5.7, která se ukázala jako problematická a nelze ji doporučit vzhledem ke zhoršení oproti referenční metodě.

Oddělení komp. v tabulce 5.12 představuje metodu váhování chybové funkce pro lepší oddělování komponent dle sekce 5.5.7. Dle výsledků dosahuje mírného zlepšení, ale oproti referenční metodě lépe odděluje komponenty, proto je metoda vhodná.

Vstup 512 pixelů a *vstup 1024 pixelů* v tabulce 5.12 značí velikosti vstupního rozlišení během trénování sítě dle sekce 5.5.8. Z dosažených výsledků je patrné mírné zlepšení při vyšším rozlišení, ale zároveň dochází k velkému nárůstu výpočetní složitosti.

5.7 Finální model

Protože je velice obtížné vyzkoušet všechny možné kombinace vzhledem k jejich velkému množství, je při výběru vhodného způsobu vycházeno z provedených experimentů. Na jejich základě bylo vyzkoušeno několik kombinací vhodných pro trénování a jako nejlepší kombinace se ukázala následující.

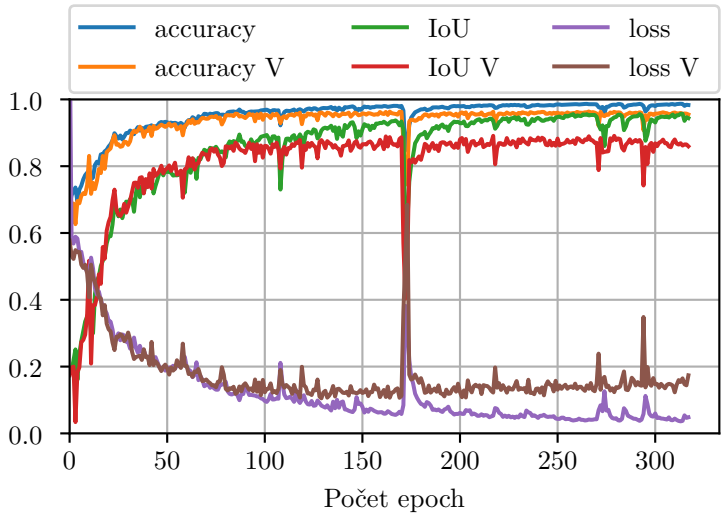
Klíčovou částí systému je plně konvoluční neuronová síť, jejíž architektura je podrobněji popsána v sekci 5.1. Trénování probíhá na trénovací množině a je rozšířeno o kroniky obsahující pouze text, které jsou následně doplněny o obrázky dle sekce 5.5.5.

Načítání vstupu je řešeno jako v případě metody *resize* v sekci 5.5.3. Vstupní obrázek je tedy načten ve stupních šedi a jeho rozměry převedeny v odpovídajícím poměru tak, aby se obrázek vešel do oblasti 512×512 pixelů. Následně je provedena korekce rozměrů z důvodů daných architekturou sítě (viz sekce 5.1). Odpovídajícím způsobem je načteno i ground-truth obrázku strany dokumentu. Během trénování jsou tedy síti poskytovány vzory s různými rozměry.

Na načtený obrázek je aplikována upravená augmentace dle sekce 5.5.4. Úprava spočívá ve vynechání zkosení (*skew*) a rotace (*rotate*). Pro augmentaci je použito pouze náhodné elastické zkreslení (*random_distortion*). Důvodem je zamezení možných problémů zmiňovaných v sekci 5.5.4, které způsobují větší chybovost u okrajů stran. Problém spočívá pravděpodobně ve změně okraje strany v důsledku následného oříznutí po rotaci či zkosení.

Proto je použito pouze náhodné elastické zkreslení, protože zde k tomuto jevu nedochází. Díky tomu je možné využít informaci poskytovanou díky paddingu v konvolučních vrstvách jako kontextu ve smyslu pozice u okraje strany a například ve větší míře ignorovat v této oblasti přítomný šum.

Následuje váhování chybové funkce pro oddělení komponent podle sekce 5.5.7. Váhy jsou spočteny na základě augmentovaného ground-truth pro kanály představující text a obrázek. Výpočet je proveden na základě vzorce 5.13, kde je za $w_c(x)$ dosazena hodnota 1, pro parametr w_0 je zvolena hodnota 5 a parametr σ zůstává na hodnotě 10.



Obrázek 5.19: Průběh trénování finálního modelu: hodnoty po jednotlivých epochách pro *accuracy*, *IoU* a chybovou funkci (*loss*) na trénovací a validační (označené V) části

Takto nastavené trénování (průběh viz 5.19) dosáhlo v porovnání s ostatními možnostmi nejlepších výsledků na validační sadě dle tabulky 5.13. Po vizuální stránce navíc nejlépe odděluje komponenty a obsahuje minimální množství šumu.

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.953	0.972	0.899	0.933	0.877	0.991
Obrázky	0.989	0.913	0.976	0.942	0.891	0.995
Pozadí	0.949	0.950	0.955	0.952	0.908	0.991
Průměr	0.964	0.945	0.943	0.942	0.892	0.992

Tabulka 5.13: Dosažené výsledky finálního modelu na validační sadě

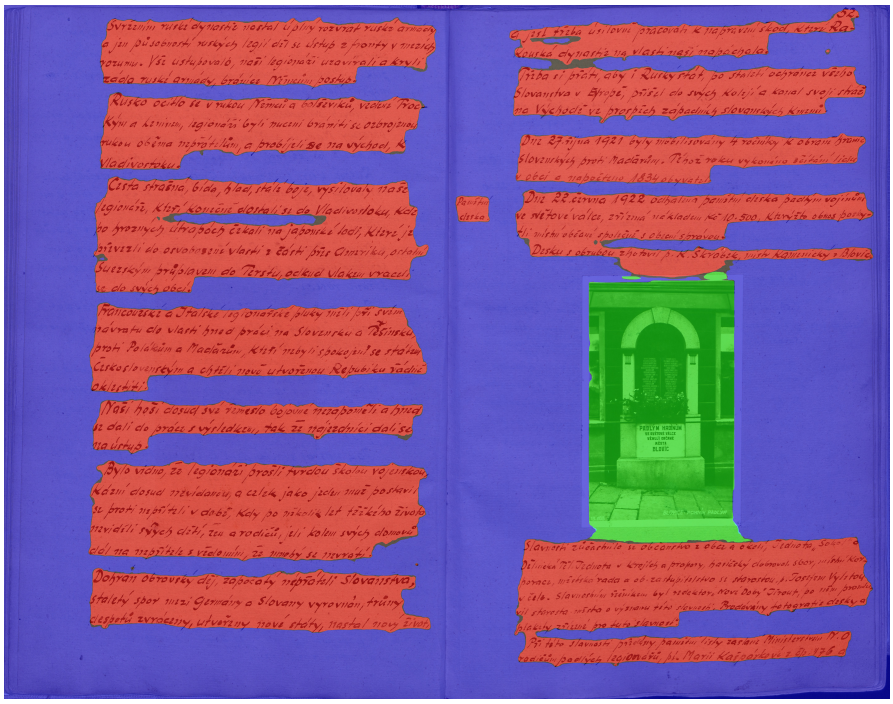
5.8 Výsledky

Nejlepších výsledků na validační sadě bylo dosaženo s finálním modelem popsaným v sekci 5.7. Proto byl vybrán pro vyhodnocení na testovací sadě, kde bylo dosaženo skvělých výsledků (viz tabulka 5.14).

	Accuracy	Precision	Recall	F1 score	IoU	FgPA
Text	0.960	0.966	0.907	0.934	0.878	0.989
Obrázky	0.991	0.935	0.979	0.955	0.916	0.990
Pozadí	0.960	0.958	0.971	0.964	0.931	0.993
Průměr	0.970	0.953	0.952	0.951	0.908	0.991

Tabulka 5.14: Dosažené výsledky finálního modelu na testovací sadě

Systém je navržen pro kroniky a byl trénován výhradně na stranách kronik, proto je výstup segmentace na těchto stranách velmi dobrý podobně jako na obrázku 5.20, kde je možné vidět úspěšné oddělení odstavců textu na vhodně strukturované straně kroniky.



Obrázek 5.20: Ukázka výstupu prototypu pro stranu kroniky z testovací sady: červeně text, zeleně obrázek a modře pozadí

Překvapením je úspěšná predikce na tištěných dokumentech (viz obrázky 5.21 a 5.22). Pro tyto dokumenty lze dosáhnout velmi dobrých výsledků, přestože nebyl systém pro segmentaci na tyto strany navržen a natrénován.

Obrázek 5.21: Ukázka výstupu prototypu pro stranu periodika z portálu Porta fontium [21]: červeně text, zeleně obrázek a modře pozadí

Obrázek 5.22: Ukázka výstupu prototypu pro stranu časopisu z datové sady Layout Analysis Dataset [1]: červeně text, zeleně obrázek a modře pozadí

5.9 Použité technologie

Pro implementaci práce je použita *Anaconda*[1] jako distribuce programovacího jazyka Python kvůli jednoduché správě balíků (knihoven). Práce je tedy implementována v jazyce Python a to ve verzi 3.7.

Pro manipulaci s obrázky, jejich načítání, ukládání, změnu rozlišení apod. je využita open source knihovna počítačového vidění *OpenCV*[2]. Pro další zpracování a výpočty je použita knihovna pro zpracování obrázků *scikit-image*[3] a knihovny pro vědecké výpočty *SciPy*[4] a *NumPy*[5]. Pro augmentaci je využita knihovna *Augmentor*[6] [6].

Implementace sítě je provedena za pomocí knihovny *Keras*[7] a *TensorFlow*[8]. Keras je vysokoúrovňové rozhraní pro neuronové sítě napsané v jazyce Python, které se snaží umožnit rychlé experimentování. K tomu využívá modularity, kde je model sítě chápán jako spojení modulů – např. vrstvy neuronové sítě, aktivační funkce a optimalizátory trénování apod. Může pracovat nad knihovnami TensorFlow, CNTK nebo Theano. TensorFlow je open source knihovna pro strojové učení a neuronové sítě. Umožňuje provádět výpočty na více procesorech nebo grafických kartách, což je vhodné pro urychlení výpočtů. Knihovnu TensorFlow používají společnosti jako Google, ebay nebo airbnb. Lze ji tedy považovat za funkční a ověřenou.

Pro ověření funkčnosti jsou použity jednotkové testy *unittest*[9], které jsou dále doplněny o případnou vizuální kontrolu pomocí knihovny pro vizualizaci *Matplotlib*[10]. Knihovna je zároveň použita pro jednoduché grafické rozhraní prototypu systému pro segmentaci.

[1] `https://www.anaconda.com/`
[2] `https://opencv.org/`
[3] `https://scikit-image.org/`
[4] `https://www.scipy.org/`
[5] `https://numpy.org/`
[6] `https://github.com/mdbloice/Augmentor`
[7] `https://keras.io/`
[8] `https://www.tensorflow.org/`
[9] `https://docs.python.org/3/library/unittest.html`
[10] `https://matplotlib.org/`

5.10 Zhodnocení a možná rozšíření

Pro segmentaci a rozdělení strany dokumentu na obrázek, text a pozadí bylo v práci využito *pixel-labeling* přístupu spolu s plně konvoluční neuronovou sítí založenou na síti *U-Net* [23]. Dosažené výsledky prezentované v sekci 5.8 jsou velice slibné. Dobrých výsledků je dosaženo dokonce i se sítí natrénovanou na pouhých šesti stranách kronik (viz sekce 5.5.3). Nelze tedy jinak, než doporučit tento přístup spolu s využitím plně konvolučních neuronových sítí k řešení tohoto problému.

Velkou výhodou implementovaného řešení je možnost segmentace prakticky libovolných rozměrů strany. Zajímavou vlastností sítě je i schopnost generalizace. Přestože je síť trénovaná na starších ručně psaných kronikách, dokáže dobře segmentovat i moderní tištěné strany (viz obrázky 5.21 a 5.22).

Architektura sítě popsaná v sekci 5.1 se díky zpracování celých stran a využití paddingu v konvolučních vrstvách dokáže velmi dobře vypořádat se šumem na okraji obrázků stran. Ten může být způsoben např. viditelností okrajů stran při naskenování větší oblasti, než je potřeba. Uspokojivá je i rychlost zpracování jedné strany, které je při vhodném rozlišení obrázku a využití výkonné grafické karty téměř okamžité. Hraje zde však roli i původní rozlišení obrázku a s tím související úpravy, takže je pak výsledná doba segmentace spíše v řádu vteřin, ale i to je dobrý výsledek.

Kapacita implementované sítě se ukázala jako dostatečná a bylo by s ní pravděpodobně možné dosáhnout dalšího členění jako např. rozlišení tištěného a psaného textu. Podobně jako v práci *Fully Convolutional Neural Networks for Page Segmentation of Historical Document Images* [25] by dále mohla být navržená síť pravděpodobně využita i pro klasifikaci textových bloků např. na odstavce a nadpisy. Zajímavým experimentem by mohlo být využití sítě pro OCR, kde by jednotlivé kanály výstupu představovaly znaky abecedy. Po zpracování kanálů by bylo možné určit znak a jeho pozici a díky tomu převést obrázek ručně psaného dokumentu do strojově čitelné podoby. Byla by však nutná datová sada, kde by bylo anotováno každé písmeno textu.

Výstup v podobě binárních masek je možné dále zpracovat pro konkrétní využití. Lze např. vymaskovat vstupní obrázek pomocí logického součinu a získat tak pouze text nebo obrázky. Dále je možné určit oblasti např. pomocí nalezení vnější hranice nalezené komponenty, případně lze danou oblast vyříznout.

Systém by bylo možné dále vylepšit například optimalizací prahování, redukcí šumu ve výstupu segmentace nebo využitím postupu odpovídajícího anotaci. Ve výstupu segmentace v podobě binární masky by byly nejprve odstraněny malé komponenty představující šum. Následně by byla nalezena

vnější hranice komponenty. Na základě binarizovaného vstupního obrázku a nalezené vnější hranice by byla hranice upravena podobným způsobem jako při použití nástroje *To Coarse Contour* nebo *To Fine Contour* během postupu anotace dle sekce 4.5.

Využít by se dala i vysoká úspěšnost na pixelech popředí (FgPA) pro zlepšení binarizace obrázku, kterou lze využít například pro nástroj *To Coarse Contour*. Vymaskování výstupu segmentace s binarizovaným obrázkem by ve velké míře odstranilo šum a umožnilo kvalitnější zpracování pro metody využívající binarizovaný obrázek (např. využití nástroje *To Coarse Contour* v předchozím odstavci).

Ke zlepšení by mohla vést i změna architektury tak, aby byl dostupný větší kontext, který by mohl pomoci lepšímu oddělování jednotlivých komponent (např. odstavců). Ke zlepšení by vedlo pravděpodobně i vyšší rozlišení vstupu, které ale povede k mnohem větší výpočetní náročnosti.

Rozšíření datové sady a vhodnější augmentace by s velkou pravděpodobností také zlepšily úspěšnost a použitelnost sítě.

6 Závěr

Cílem práce je návrh a implementace prototypu systému pro segmentaci stran na text, pozadí a obrázky. Proto jsou v první části práce prostudovány možnosti řešení problému a dostupné datové sady. Dále je čtenář seznámen s historickým portálem Porta fontium.

Bohužel nebyla nalezena vhodně anotovaná datová sada pro řešení dané úlohy, proto následuje část týkající se vytvoření nové datové sady. Jako základ datové sady posloužily stránky z pěti různých kronik poskytnutých portálem Porta fontium. Stránky jsou vybrány tak, aby představovaly reprezentativní vzorek reálných dat. Detailní anotace je provedena s využitím nástroje Aletheia. Nástroj pracuje s formátem PAGE, který je podporován řadou dalších nástrojů. Jsou tedy splněny požadavky na realističnost, detailnost a flexibilní strukturu. Celkem je anotováno 38 stran kronik a dalších 34 stran s tištěným textem z různých zdrojů.

Další kapitola popisuje navržený a implementovaný prototyp systému pro segmentaci a rozdělení stran dokumentů na text, obrázky a pozadí spolu s experimenty pro nalezení optimálních parametrů tohoto prototypu. Problém je řešen jako pixel-labeling problém, kde je každý obrazový bod dokumentu klasifikován do tříd: text, obrázek a pozadí. Předpokládá se rozložení, kde se mohou jednotlivé komponenty překrývat. Jeden pixel tedy může být zároveň zařazen do více tříd, protože se ve stranách dokumentů mohou objevovat obrázky popsané textem apod. V práci jsou provedené experimenty např. s rozšířením trénovacích dat a váhováním chybové funkce pro lepší oddělování segmentovaných komponent. Na základě experimentů je systém odladěn a vyhodnocen. Dále jsou navržena jeho další možná rozšíření.

Použitá plně konvoluční neuronová síť se osvědčila díky skvělým dosaženým výsledkům, schopnosti generalizace a potřeby nízkého počtu trénovacích dat. Navržený systém je schopen zpracovat obrázky stran téměř libovolných rozměrů a jeho výstupem jsou binární masky pro každou klasifikovanou třídu. Tyto masky je možné dále zpracovat pro konkrétní využití např. vymaskováním vstupního obrázku, nalezením vnějších hranic segmentovaných komponent či vyříznutím požadovaných oblastí. Díky tomu je možné výstup využít např. pro strojovou anotaci stran nebo poskytnutí textových bloků pro OCR.

7 Slovník pojmů a zkratek

- accuracy – metrika pro vyhodnocení úspěšnosti
- Aletheia – systém pro anotaci
- augmenace – metoda pro rozšíření trénovací sady
- bias – práh
- crop – vyříznutí části obrázku
- dropout – deaktivace neuronu se zadanou pravděpodobností
- early stopping – ukončení trénování pokud se model během trénování dále nezlepšuje
- epocha – představuje cyklus nad všemi vzory trénovací sady během trénování sítě
- FgPA – Foreground Pixel Accuracy, metrika pro vyhodnocení úspěšnosti
- F1 score – metrika pro vyhodnocení úspěšnosti
- ground-truth – požadovaný výsledek
- IoU – Intersection over Union, metrika pro vyhodnocení úspěšnosti
- klasifikace – zařazení do tříd
- loss funkce – chybová funkce
- OCR – optické rozpoznávání znaků
- padding – doplnění okrajů bezvýznamovými hodnotami
- page layout – rozvržení strany dokumentu
- page layout analysis – segmentace stran na komponenty a jejich klasifikace
- pixel – obrazový bod
- pixel-labeling – označení každého obrazového bodu třídou

- precision – metrika pro vyhodnocení úspěšnosti
- recall – metrika pro vyhodnocení úspěšnosti
- ReLU – aktivační funkce neuronu
- resize – změna velikosti obrázku
- segmentace – extrahování homogenních komponent
- sigmoida – aktivační funkce neuronu
- skip-connection – přidaná vazba v architektuře sítě
- softmax – aktivační funkce neuronu
- state-of-the-art – nejlepší dostupné řešení
- strana – obrázek strany či dvojstrany dokumentu
- To Coarse Contour – nástroj systému Aletheia
- To Fine Contour – nástroj systému Aletheia

Literatura

[1] Antonacopoulos, A. et al. A Realistic Dataset for Performance Evaluation of Document Layout Analysis. In *2009 10th International Conference on Document Analysis and Recognition*, s. 296–300, July 2009. doi: 10.1109/ICDAR.2009.271.

[2] Antonacopoulos, A. et al. Historical Document Layout Analysis Competition. In *2011 International Conference on Document Analysis and Recognition*, s. 1516–1520, Sep. 2011. doi: 10.1109/ICDAR.2011.301.

[3] Antonacopoulos, A. et al. ICDAR 2013 Competition on Historical Book Recognition (HBR 2013). In *2013 12th International Conference on Document Analysis and Recognition*, s. 1459–1463, Aug 2013. doi: 10.1109/ICDAR.2013.294.

[4] Antonacopoulos, A. et al. ICDAR 2013 Competition on Historical Newspaper Layout Analysis (HNLA 2013). In *2013 12th International Conference on Document Analysis and Recognition*, s. 1454–1458, Aug 2013. doi: 10.1109/ICDAR.2013.293.

[5] Baloun, J. *Prohledávání dokumentů podle automaticky extrahovaných vzorů* [online]. Plzeň, 2018. [cit. 2020-03-28]. Západočeská univerzita v Plzni. Dostupné z: `https://portal.zcu.cz/stag?urlid=prohlizeni-prace-detail&praceIdno=75853`.

[6] Bloice, M. D. – Roth, P. M. – Holzinger, A. Biomedical image augmentation using Augmentor. *Bioinformatics*. 04 2019, 35, 21, s. 4522–4524. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz259. Dostupné z: `https://doi.org/10.1093/bioinformatics/btz259`.

[7] Brownlee, J. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning* [online]. [cit. 2020/4/6]. Dostupné z: `https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/`.

[8] Chen, K. et al. Convolutional Neural Networks for Page Segmentation of Historical Document Images. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01, s. 965–970, Nov 2017. doi: 10.1109/ICDAR.2017.161.

[9] Clausner, C. – Pletschacher, S. – Antonacopoulos, A. Aletheia - An Advanced Document Layout and Text Ground-Truthing System for

Production Environments. In *2011 International Conference on Document Analysis and Recognition*, s. 48–52, Sep. 2011. doi: 10.1109/ICDAR.2011.19.

[10] Clausner, C. – Antonacopoulos, A. – Pletschacher, S. ICDAR2019 Competition on Recognition of Documents with Complex Layouts - RDCL2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, s. 1521–1526, Sep. 2019. doi: 10.1109/ICDAR.2019.00245.

[11] *CS231n Convolutional Neural Networks for Visual Recognition* [online]. [cit. 2020/3/28]. Dostupné z: http://cs231n.github.io/.

[12] Drivas, D. – Amin, A. Page segmentation and classification utilising a bottom-up approach. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 2, s. 610–614 vol.2, Aug 1995. doi: 10.1109/ICDAR.1995.601970.

[13] Godoy, D. *Understanding binary cross-entropy / log loss: a visual explanation* [online]. [cit. 2020/4/6]. Dostupné z: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a.

[14] *IMPACT project* [online]. [cit. 2020/03/17]. Dostupné z: http://www.impact-project.eu/.

[15] Kaspar, F. – Schuster, H. G. Easily calculable measure for the complexity of spatiotemporal patterns. *Phys. Rev. A*. Jul 1987, 36, s. 842–848. doi: 10.1103/PhysRevA.36.842. Dostupné z: https://pdfslide.net/documents/easily-calculable-measure-for-the-complexity-of-spatiotemporal-patterns.html.

[16] Kise, K. – Yanagida, O. – Takamatsu, S. Page segmentation based on thinning of background. In *Proceedings of 13th International Conference on Pattern Recognition*, 3, s. 788–792 vol.3, 1996.

[17] Kise, K. *Page Segmentation Techniques in Document Analysis*, s. 135–175. Springer London, London, 2014. doi: 10.1007/978-0-85729-859-1_5. Dostupné z: https://doi.org/10.1007/978-0-85729-859-1_5. ISBN 978-0-85729-859-1.

[18] Liška, M. *Segmentace historických obrazových dokumentů* [online]. Plzeň, 2019. [cit. 2020-03-17]. Západočeská univerzita v Plzni. Dostupné z: https://portal.zcu.cz/stag?urlid=prohlizeni-prace-detail&praceIdno=79568.

[19] Papadopoulos, C. et al. The IMPACT Dataset of Historical Document Images. In *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*, HIP '13, s. 123–130, New York, NY, USA, 2013. Association for Computing Machinery. doi: 10.1145/2501115.2501130. Dostupné z: https://doi.org/10.1145/2501115.2501130. ISBN 9781450321150.

[20] Pletschacher, S. – Antonacopoulos, A. The PAGE (Page Analysis and Ground-Truth Elements) Format Framework. In *2010 20th International Conference on Pattern Recognition*, s. 257–260, Aug 2010. doi: 10.1109/ICPR.2010.72.

[21] *Porta fontium* [online]. Bavorsko-česká síť digitálních historických pramenů. [cit. 2020/03/06]. Dostupné z: http://www.portafontium.cz/.

[22] *PRImA* [online]. Pattern Recognition & Image Analysis Research Lab. [cit. 2020/03/06]. Dostupné z: https://www.primaresearch.org/.

[23] Ronneberger, O. – Fischer, P. – Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N. et al. (Ed.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, s. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[24] Sauvola, J. – Pietikäinen, M. Adaptive document image binarization. *Pattern Recognition*. 2000, 33, 2, s. 225 – 236. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(99)00055-2. Dostupné z: http://www.sciencedirect.com/science/article/pii/S0031320399000552.

[25] 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, April 2018. doi: 10.1109/DAS.2018.39.

[26] Yip, S. K. – Chi, Z. Page segmentation and content classification for automatic document image processing. In *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No.01EX489)*, s. 279–282, May 2001. doi: 10.1109/ISIMP.2001.925388.

A Obsah přiloženého DVD

K diplomové práci je přiloženo DVD s elektronickou verzí této práce, vytvořenou datovou sadou, zdrojovými soubory s implementací prezentované práce a potřebné výstupy a soubory pro případné ověření dosažených výsledků. DVD obsahuje následující soubory:

1. anotace.zip
    - archiv s anotovanými obrázky (viz kapitola 4)
    - obsah:
        - pro každý obrázek je poskytnuto ground-truth v PAGE formátu a z něho vygenerovaný png soubor
        - názvy složek značí původ obrázků stran dokumentu, například v podsložce porta/adresare jsou stránky adresářů z portálu Porta fontium
2. dataset.zip
    - archiv s připravenou datovou sadou (viz sekce 4.6)
    - obsah:
        - je poskytnuto ground-truth v podobě png souboru
        - názvy složek a jejich obsah odpovídají sekci 4.6
3. models.zip
    - archiv s výstupy pro ověření prezentovaných výsledků
    - kořenový adresář obsahuje:
        - jupyter – složka obsahující jupyter notebook[1] soubory s vyhodnocením a vizualizací výsledků
        - weights – složka obsahující váhy sítě prezentované v práci a průběh trénování
        - readme.txt – soubor popisující obsah archivu a význam názvů souborů

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

4. src.zip

    - archiv se zdrojovými kódy a prototypem aplikace pro segmentaci stran (viz příloha B)
    - kořenová složka obsahuje:
        - readme.txt soubor s podrobnějšími informacemi
        - soubor s Anaconda[2] prostředím použitým při vývoji a vyhodnocení
        - soubor s požadavky prototypu aplikace pro segmentaci stran
        - další soubory viz readme.txt

5. text.zip

    - archiv se zdrojovými soubory LaTeX pro tento dokument

6. poster.zip

    - archiv s požadovaným posterem

7. Baloun_Josef_2020_DP.pdf

    - elektronická verze tohoto dokumentu

[2] `https://anaconda.org/`

B Uživatelská příručka

Součástí archivu *src.zip* (viz příloha A) je prototyp aplikace pro segmentaci stran pod názvem *pagenter* (ze slov *page* a *segmenter*). Pro aplikaci jsou v kořenové složce klíčové:

- pagenter_data – složka s výchozími daty pro *pagenter* (vstupní obrázky, váhy sítě, složka pro případné ukládání a ukázky výstupu)
- pagenter.py – prototyp aplikace pro segmentaci stran
- pagenter_settings.py – nastavení aplikace pro segmentaci stran
- pagenter_requirements.txt - požadavky aplikace pro segmentaci stran (potřebné balíky Python)

B.1 Instalace

Pro aplikaci je doporučen Python ve verzi 3.7. Pro spuštění jsou vyžadovány balíky tensorflow, keras, matplotlib, opencv a numpy. Tyto balíky lze nainstalovat pomocí správce balíků *pip*[1] zadáním následujícího příkazu v kořenové složce:

```
pip install -r pagenter_requirements.txt
```

V případě zájmu je možné změnit výchozí nastavení v přehledně komentovaném souboru *pagenter_settings.py*, který obsahuje:

- cestu pro soubor s váhami sítě (./pagenter_data/weights/...)
- omezení velikosti pro vstup sítě (512)
- cestu ke složce pro případné ukládání výstupu segmentace (./pagenter_data/output)
- výchozí cestu ke složce se vstupními obrázky pro segmentaci (./pagenter_data/input)
- výběr přípon souborů pro filtrování vstupních souborů (.jpg a .tif)

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

B.2 Spuštění

Aplikaci je vhodné spustit z kořenové složky, protože v nastavení (soubor pagenter_settings.py) jsou jako výchozí zadány relativní cesty. Aplikace má jeden volitelný parametr, který představuje cestu ke složce se vstupními obrázky. Ostatní nastavení je převzato ze souboru *pagenter_settings.py*.

Při volání bez parametru je načtena výchozí cesta ke složce se vstupními obrázky pro segmentaci, která je definovaná v souboru *pagenter_settings.py*:

```
python pagenter.py
```

Při volání s parametrem je výchozí cesta ke složce se vstupními obrázky pro segmentaci nahrazena cestou zadanou jako parametr:

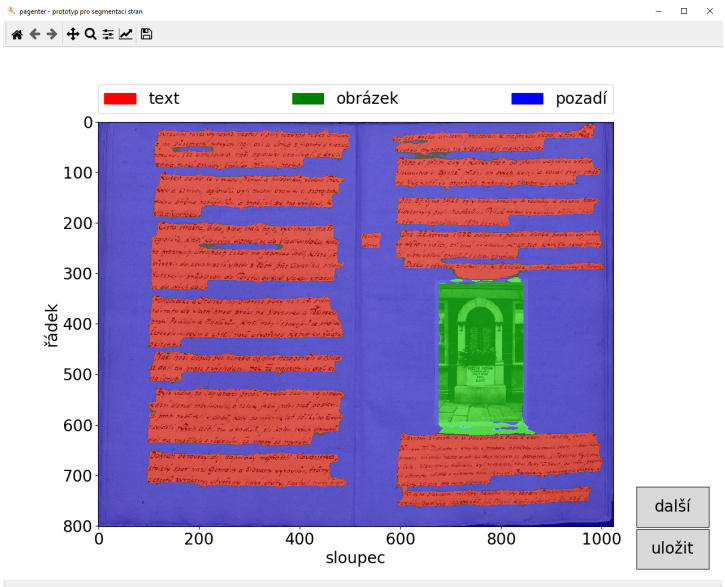
```
python pagenter.py ./mydata/folder
```

Po úspěšném spuštění se zobrazí jednoduché GUI[2] (viz obrázek B.1). V případě neúspěšného spuštění se řiďte výpisem nebo chybovým hlášením.

B.3 Ovládání

Po otevření GUI (viz obrázek B.1) je zobrazen první segmentovaný obrázek. Pomocí panelu nástrojů (horní část, ale závisí na verzi balíku matplotlib) je možné přiblížit detaily apod. Pro ovládání jsou určena dvě tlačítka *další* a *uložit*.

[2]GUI – grafické uživatelské rozhraní



Obrázek B.1: GUI pagenter

Další obrázek je zpracován a zobrazen po stisku tlačítka *další*. V případě, že nejsou v dané vstupní složce další obrázky k dispozici je tlačítko *další* skryté.

Uložení výsledku segmentace je provedeno kliknutím na tlačítko *uložit*. V tom případě dojde k uložení původního obrázku, výstupu segmentace a jejich vizualizace (viz obrázek B.1) do složky pro ukládání, která je definovaná v souboru *pagenter_settings.py*.

Ukončení aplikace se provede zavřením okna s GUI.