

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

## **Bakalářská práce**

# **Analýza a segmentace historických obrazových dokumentů**

Místo této strany bude  
zadání práce.

# Poděkování

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

V Plzni dne 6. května 2020

Filip Tremel

# Prohlášení

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

V Plzni dne 6. května 2020

Filip Tremel



## **Abstract**

Segmentation process of historical document images is important for their conversion into text form. The goal of segmentation process is find areas containing text and save them into a PAGE xml file, which contains all important information about the specific page of the document. Segmentation process consists of many steps. With fully convolution neural networks UNet and Arunet, we create predication masks of areas of interest and separators. As a part of this bachelor thesis, experiments that help to find appropriate parameters of the neural networks will be carried out. The next step will be to implement an extension, which will make it possible to distinguish the types of text blocks and also better separate specific paragraphs.

## **Abstrakt**

Proces segmentace historických obrazových dokumentů je klíčový pro jejich následné převedení do textové podoby. Cílem segmentačního procesu je nalezení oblastí, které obsahují text a jejich uložení do PAGE xml, obsahující veškeré důležité informace o konkrétní stránce dokumentu. Segmentační proces je složen z mnoha kroků. S využitím plně propojených konvolučních neuronových sítí UNet a AruNet, vytvoříme predikované masky oblastí zájmu a oddělovačů jednotlivých částí. V rámci bakalářské práce budou provedeny experimenty s cílem nalézt vhodné parametry neuronových sítí. Dále bude implementováno rozšíření, které umožní odlišit typy textových oblastí a také lépe oddělit jednotlivé odstavce.

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>1</b>  |
| <b>2</b> | <b>Segmentace</b>                                  | <b>2</b>  |
| 2.1      | Základní metody segmentace . . . . .               | 2         |
| 2.1.1    | Prahovací metody . . . . .                         | 2         |
| 2.1.2    | Region-based metody . . . . .                      | 4         |
| 2.1.3    | Hranové metody (Boundary/edge-based methods) . .   | 6         |
| <b>3</b> | <b>Umělé neuronové sítě</b>                        | <b>9</b>  |
| 3.1      | Perceptron . . . . .                               | 9         |
| 3.2      | Vrstvy . . . . .                                   | 10        |
| 3.3      | Učení umělé neuronové sítě . . . . .               | 11        |
| 3.3.1    | Trénování s učitelem . . . . .                     | 11        |
| 3.3.2    | Trénování bez učitele . . . . .                    | 12        |
| 3.4      | Konvoluční neuronová síť - CNN . . . . .           | 12        |
| 3.4.1    | Data augmentation . . . . .                        | 13        |
| 3.5      | Plně propojené konvoluční neuronové sítě . . . . . | 14        |
| 3.5.1    | UNet . . . . .                                     | 14        |
| 3.5.2    | AruNet . . . . .                                   | 14        |
| <b>4</b> | <b>Data</b>  | <b>16</b> |
| 4.1      | Europeana . . . . .                                | 16        |
| 4.2      | Porta fontium . . . . .                            | 16        |
| 4.3      | PAGE . . . . .                                     | 17        |
| 4.3.1    | Struktura PAGE . . . . .                           | 17        |
| 4.4      | Layout Evaluation . . . . .                        | 17        |
| <b>5</b> | <b>Proces segmentace</b>                           | <b>22</b> |
| <b>6</b> | <b>Využité technologie</b>                         | <b>25</b> |
| 6.1      | Python . . . . .                                   | 25        |
| 6.2      | OpenCV-Python . . . . .                            | 25        |
| 6.3      | Tensorflow . . . . .                               | 25        |
| 6.4      | Keras . . . . .                                    | 25        |
| 6.4.1    | Keras vrstvy . . . . .                             | 26        |

|           |   |           |
|-----------|---|-----------|
| <b>7</b>  | <b>Rozdělení stránky na bloky</b>                                   | <b>28</b> |
| 7.1       | Volba modelu . . . . .  | 28        |
| 7.1.1     | Způsob vyhodnocení správnosti modelu . . . . .                      | 28        |
| 7.1.2     | Testované trénovací parametry experimentu . . . . .                 | 29        |
| 7.1.3     | Výsledné hodnoty . . . . .  | 30        |
| 7.1.4     | Zhodnocení . . . . .  | 31        |
| 7.2       | Predikce separátorů . . . . .                                       | 31        |
| <b>8</b>  | <b>Prvotní detekce oblastí zájmu</b>                                | <b>35</b> |
| 8.1       | Postup . . . . .  | 36        |
| <b>9</b>  | <b>Typy oblastí zájmu</b>   | <b>37</b> |
| 9.1       | Hlavička stránky . . . . .  | 37        |
| 9.2       | Sloupec standardního textu . . . . .                                | 37        |
| 9.3       | Speciální objekty . . . . .   | 37        |
| 9.4       | Proces určování typu oblasti . . . . .                              | 38        |
| 9.5       | Hodnoty rozhodující o typu oblasti . . . . .                        | 39        |
| 9.5.1     | Velikost písma . . . . .  | 39        |
| 9.5.2     | Množství znaků . . . . .  | 39        |
| 9.5.3     | Průnik výsledků . . . . .   | 39        |
| <b>10</b> | <b>Detekce spojených odstavců</b>                                   | <b>42</b> |
| 10.1      | Algoritmus detekce a oddělení odstavců . . . . .                    | 42        |
| <b>11</b> | <b>Finální výsledky</b>   | <b>44</b> |
| 11.0.1    | Porovnání výsledků na testovací sadě vůči diplomové práci . . . . . | 44        |
| <b>12</b> | <b>Závěr</b>  | <b>47</b> |
| <b>13</b> | <b>Použité zkratky</b>  | <b>48</b> |
| <b>14</b> | <b>Příručka</b>   | <b>49</b> |
| 14.1      | Instalace . . . . .   | 49        |
| 14.2      | Spuštění . . . . .  | 49        |
|           | <b>Literatura</b>   | <b>50</b> |

# 1 Úvod

Žijeme v moderní době počítačů, mobilů a mnoha jiných chytrých zařízení. Snažíme se veškeré vědění šířit mezi lidstvo a díky internetu je to mnohem jednodušší než v době, kdy bylo možné informace získat pouze z tištěných a psaných knih. Zde ovšem vzniká problém, jak převedeme staré dokumenty, knihy nebo noviny do digitální podoby? Ruční přepis je možný a někdy i nezbytný, avšak existuje i méně náročná cesta jak toho dosáhnout. Snahou projektu Bavorsko-česká síť digitálních historických pramenů<sup>1</sup> je spojit rozdělené archivní fondy, jež se vztahují k sudetoněmeckým dějinám a dějinám bavorského příhraničí do jednoho virtuálního celku.

Pouhé naskenování a uložení kopie není ideální pokud bychom si přáli v textu hledat konkrétní pasáže, oblasti či informace. Takto importované stránky mají řadu vad, jakými může být například digitální šum, pootočení celého textu, jeho deformace a mnoho dalších.

Projekt Moderní zpřístupnění historických pramenů má za cíl obohatit platformu Porta fontium o možnost efektivního dvojjazyčného vyhledávání. Předpokladem pro možnost vyhledávání je převedení dokumentů do textové podoby. To zahrnuje předzpracování, segmentaci a OCR.

V rámci projektu vzniká i OCR demo, webová aplikace pro demonstraci použitých metod. Prvním krokem je binarizace, která má za úkol oddělit text, pozadí a částečně odstranit šum. Segmentací získáme oddělené jednotlivé oblasti zájmu, nezbytně nutné pro následné rozpoznávání textu a obsahu dokumentu.

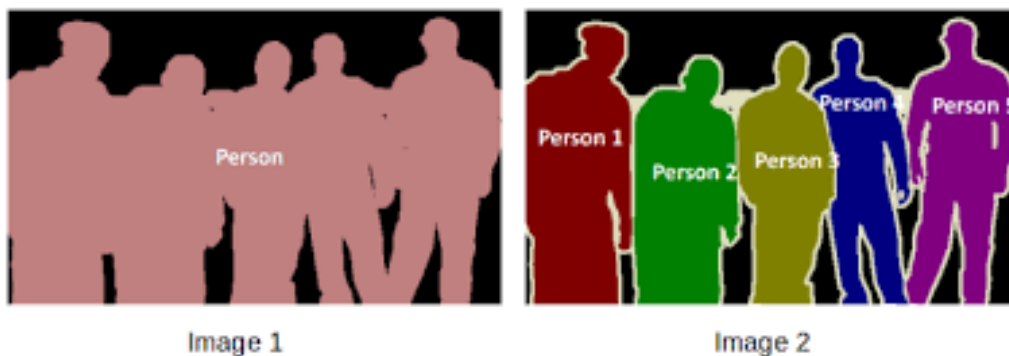
V rámci této bakalářské práce se seznámíme se základními přístupy k segmentaci obrazových dokumentů společně s testovanými datovými množinami následované popisem umělých neuronových sítí. Cílem je navrhnout a implementovat proces segmentace s využitím plně konvolučních neuronových sítí, navazující na diplomovou práci Miroslava Lišky[4], která využívala síť AruNet a UNet pro rozdělení dokumentu na bloky podle separátorů. Segmentační proces popsany v této práci jde ještě dále a snaží se získat jednotlivé odstavce, nadpisy a speciální bloky, které pouhý proces AruNetu, dosud implementovaný, nedokázal oddělit. Od výsledného PAGE souboru je očekávána mnohem lepší a přesnější anotace.

---

<sup>1</sup><http://www.portafontium.cz>

## 2 Segmentace

Proces snažící se oddělit jednotlivé oblasti zájmu nazýváme segmentace. Oblast zájmu představuje část obrázku, dokumentu, která je pro nás užitečná a z důvodu dalšího zpracování, musí být oddělena od ostatních objektů. Tyto separované části obsahují jednotlivé objekty jakými mohou být například konkrétní osoba ze společné fotografie, SPZ získaná z obrázku automobilu nebo v případě historických obrazových dokumentů textový odstavec.



Obrázek 2.1: Segmentace skupiny osob

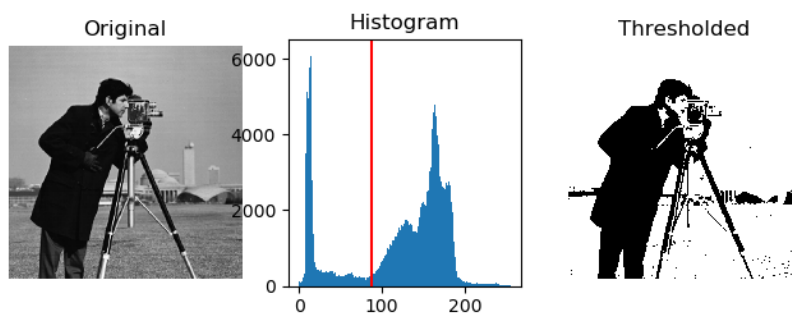
**Zdroj:** <https://cdn.analyticsvidhya.com>

### 2.1 Základní metody segmentace

Existuje mnoho základních přístupů k procesu segmentace, metody prahové, regionální a hranové jsou jenom malým výčtem avšak pro principiální představení naprosto dostatečné.

#### 2.1.1 Prahovací metody

Prahovací metody patří mezi nejstarší a zároveň nejjednodušší metody segmentace obrazu. Z důvodu její nenáročnosti na hardwarovou realizaci je to nejčastěji používaný postup. Díky tomu patří mezi nejrychlejší segmentační metody a je ho možné provádět v reálném čase. Práh představuje hranici dle které se algoritmus rozhoduje při procházení každého pixelu binárního obrázku, zda konkrétnímu pixelu přiřadí hodnotu 0 či 1.



Obrázek 2.2: Thresholding (Prahování)

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

### Postup

1. V prvním kroku si musíme zvolit práh (Threshold -  $T$ ) jako předem určenou konstantu  $T$ .
2. Následně metoda spustí algoritmus, který postupně prochází každý jeden pixel v binárním obrázku. Pokud je hodnota pixelu menší než práh, funkce mu nastaví hodnotu 0, v opačném případě 1.

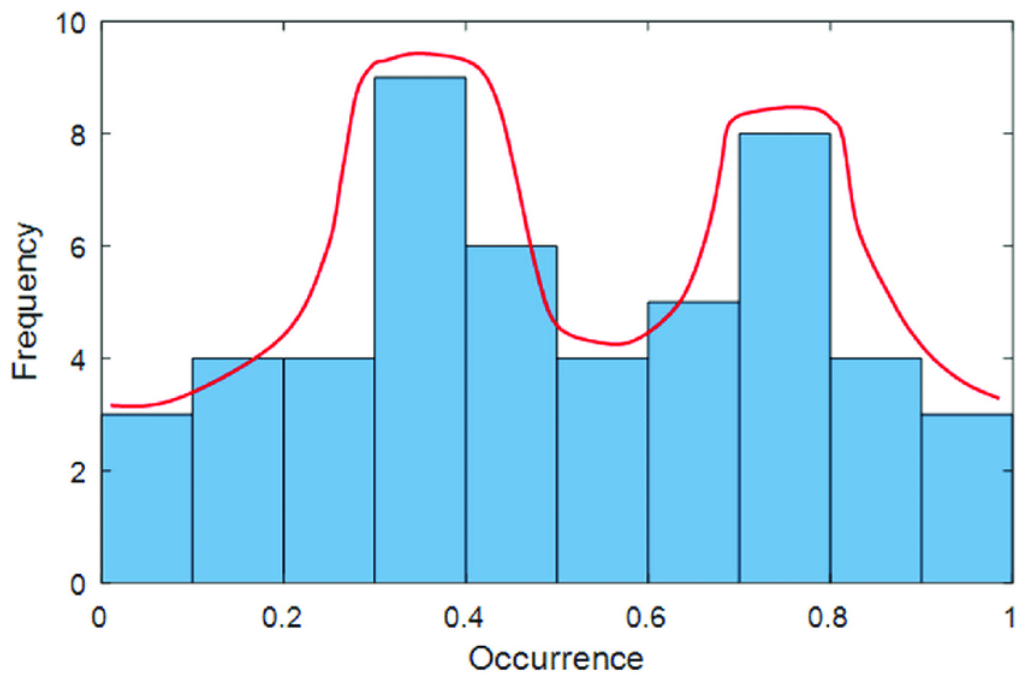
$$g(i, j) = \begin{cases} 1 & \text{pro } f(i, j) \geq T \\ 0 & \text{pro } f(i, j) < T \end{cases}$$

### Hledání prahu

Práh je možné hledat vlastností zvané bimodálnost. Pomocí bimodálního rozložení histogramu se určuje hodnota prahu jako lokální minimum mezi dvěma největšími lokálními maximy. Ovšem občas je dosti obtížné jednoznačně určit kde se lokální minima a maxima nachází. Z tohoto důvodu se histogram vyhlazuje pomocí lokálního průměrování nebo filtruje mediánem.

### Otsu

Otsuova metoda je metoda automatického prahování binárních obrázků. Patří mezi metody globálního prahování. Algoritmus se snaží nalézt takovou optimální hodnotu prahu  $T$ , aby rozptyl  $\delta$  (vážený rozptyl dvou tříd jasů) byl minimalizován. Předpokladem je, že histogram obrázku má bimodální rozdělení.[7]



Obrázek 2.3: Bimodální rozdělení

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

## Sauvola

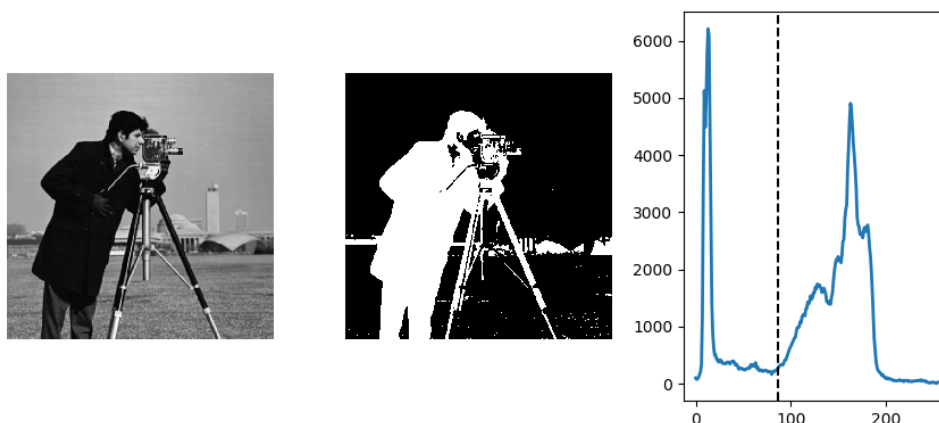
Sauvola technika spočte práh  $T$  pomocí směrodatné odchylky a střední hodnoty. Lokální střední hodnota a směrodatná odchylka přizpůsobují hodnotu prahu dle kontrastu v okolí pixelu. Narozdíl od prahovací metody Otsu patří mezi lokální metody.[7]

### 2.1.2 Region-based metody

Region-based metody jsou souborem metod, které detekují přímo oblasti zájmu v obrazu namísto hran. Tento přístup je vhodný pokud obraz obsahuje velké množství digitálního šumu, protože metody hledající hranice (boundary-based methods) při jeho vysoké koncentraci velmi obtížně hrany detekují. Segmentační požadavek pro regionální metody je homogenita oblastí. Kritériem homogenity je barva, textura nebo úroveň jasu.

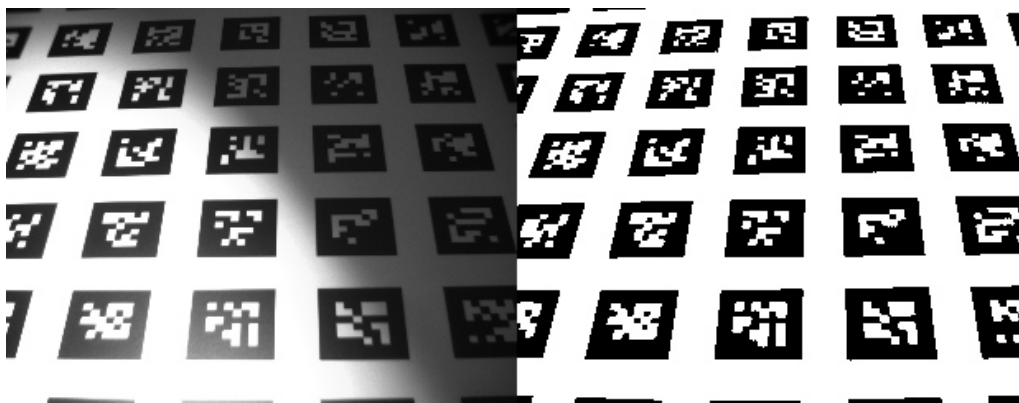
### Region growing

Region growing je metoda narůstání oblastí, kdy sousední pixely s podobnou amplitudou jsou spolu seskupovány. Na začátku je potřeba vybrat alespoň



Obrázek 2.4: Otsu thresholding

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



Obrázek 2.5: Sauvola thresholding

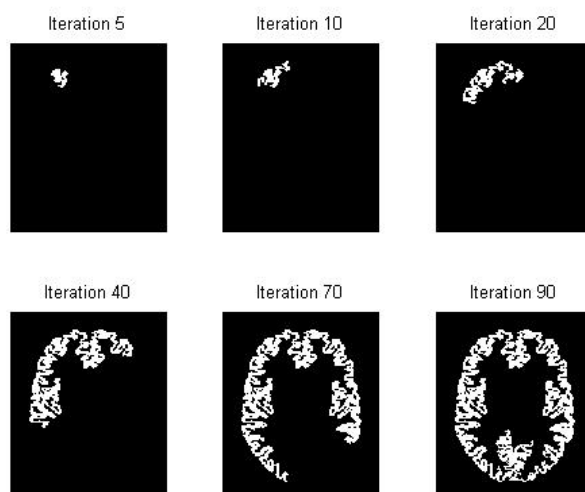
Zdroj: <https://www.mathworks.com>

jeden inicializační pixel, tzv. seed pixel, od kterého se oblast začne šířit a zvětšovat.

### Split and merge

Metoda dělení a spojování regionů vznikla ze dvou starších metod region splitting a region merging. Dělení obrazu na stále menší části (tvar struktury je nutné předem definovat jako vstupní parametr) a pokud mají sousední části stejnou nebo velmi podobnou homogenitu, jsou spojeny dohromady. Nejčastěji používanou strukturou (tzv. quart tree) rozdělí vždy jednu část (obvykle čtverec) na 4 stejně velké podoblasti. Jak již bylo výše uvedeno jedním ze vstupních parametrů, který musíme předem definovat je tvar struktury, nicméně není to jediný parametr. Další vlastností je kritérium ho-

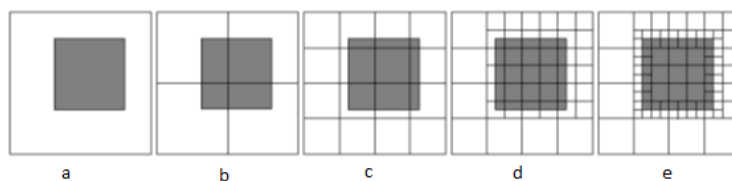




Obrázek 2.6: Region growing

**Zdroj:** <https://ww2.mathworks.cn>

mogenity nebo-li příznaky podle kterých se mají oblasti spojovat. Kritérium homogenity má klíčový dopad na chování algoritmu.



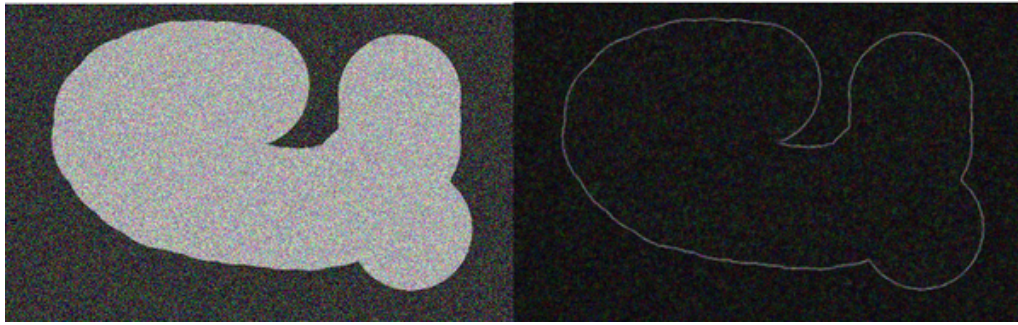
Obrázek 2.7: Split and merge

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

### 2.1.3 Hranové metody (Boundary/edge-based methods)

#### Houghova transformace

Houghova transformace reprezentuje rovinné křivky. Nespornou výhodou oproti ostatním metodám je její robustnost, protože jí relativně nevadí digitální šum ani poškozená data. Tyto rovinné křivky se protínají přesně v bodě, který reprezentuje parametry námi hledané křivky (parametrické vyjádření přímky).



Obrázek 2.8: Hrana při přechodu - Hranové metody

**Zdroj:** <https://upload.wikimedia.org>

### Hranový detektor Canny

Hranový detektor Canny je více-krokový algoritmus, který potlačí digitální šum a vyhledá hrany v obrázku.

1. V obrázku je potřeba snížit šum a nechtěné detaily pomocí Gaussova filtru.

$$g(m, n) = G_{\delta}(m, n) * f(m, n)$$

$$G_{\delta} = \frac{1}{\sqrt{2\pi\delta^2}} \exp\left(-\frac{m^2 + n^2}{2\delta^2}\right)$$

2. Výpočtem gradientu  $g(m, n)$  s využitím některého z gradientních operátorů získám:

$$M(m, n) = \sqrt{g_m^2(m, n) + g_n^2(m, n)}$$

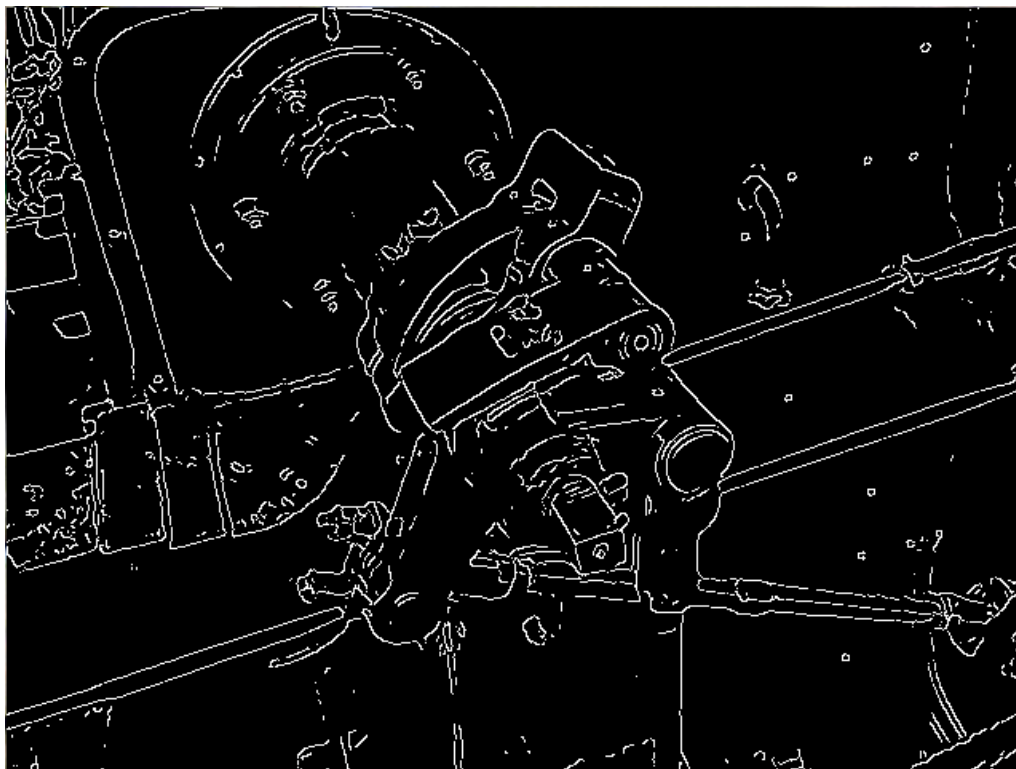
$$\theta(m, n) = \tan^{-1}[g_n(m, n)/g_m(m, n)]$$

3. Práh  $M$  vypočteme z tvaru:

$$M_T(m, n) = \begin{cases} M(m, n) & \text{pro } M(m, n) > T \\ 0 & \text{jindy} \end{cases}$$

4. Pomocí prahování vytvořím dva rozdílné prahy (práh 1 < práh 2), abychom získaly dva binární obrázky T1 a T2

5. Propojení hranových segmentů v obrázku T2 pro vytvoření spojitých hran. To provedeme tak, že každý segment v T2 je třeba sledovat až do jeho konce a poté prohledat a najít nějaký hranový segment v T1 k propojení mezery dokud nedosáhne jiného hranového segmentu v T2

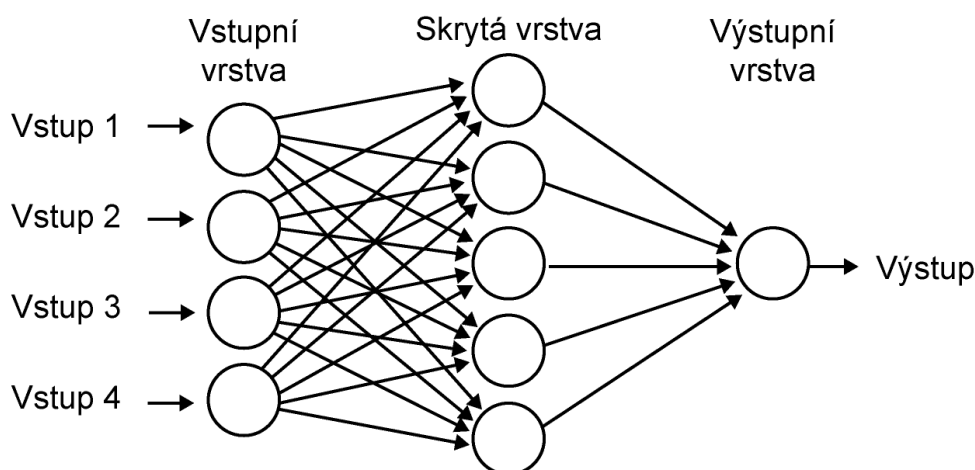


Obrázek 2.9: Ukázka výsledku Canny hranového detektoru

**Zdroj:** <https://upload.wikimedia.org>

## 3 Umělé neuronové sítě

Neuronová síť je výpočetní model, který je složený z umělých neuronů (perceptronů) podobně jako lidský mozek obsahuje buňky nebo-li biologické neurony schopného spolu komunikovat (lidský mozek obsahuje 100 miliard neuronů). Ty si navzájem předávají signály pomocí přenosových funkcí. Spektrum použití neuronových sítí se v dnešní době velice rychle rozrůstá. Od autonomních automobilů schopných automatického řízení, přes bezpečnostní systémy rozeznávající obličeje, až po systém pro rozpoznávání nemocí u lidí a následné doporučení léčby.



Obrázek 3.1: Schéma neuronové sítě

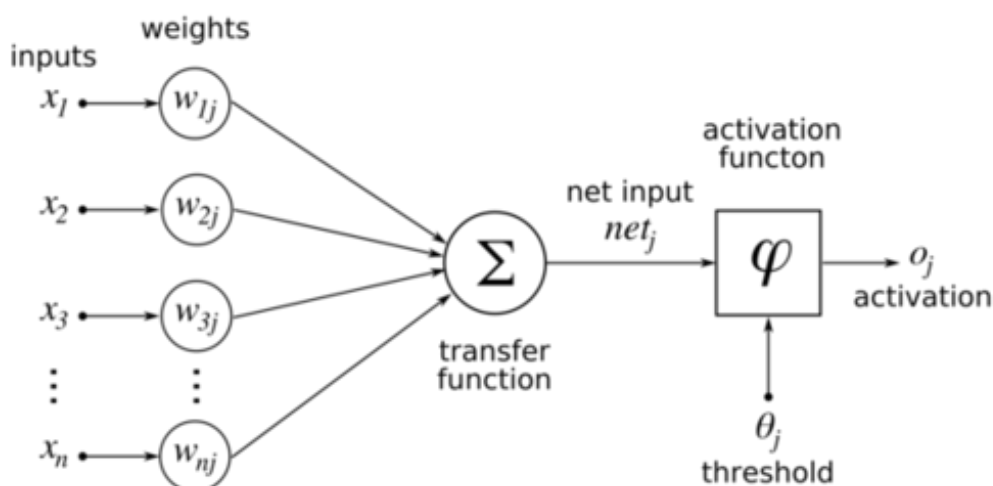
**Zdroj:** <https://portal.matematickabiologie.cz>

### 3.1 Perceptron

Perceptron je matematický model biologického neuronu a zároveň základním prvkem umělé neuronové sítě.

Každý umělý neuron může mít libovolný počet vstupů, ale vždy pouze jediný výstup, který ovšem může být vstup do více než jednoho perceptronu v další vrstvě. Perceptron je vlastně binární klasifikátor, který převede vektor vstupů  $x = [x_1, x_2, x_3, \dots, x_n]$  na výstupní hodnotu funkce  $f(x)$ .

$$f(x) = \begin{cases} 1 & \text{pro } w * x + b > 0 \\ 0 & \text{jindy} \end{cases}$$



Obrázek 3.2: Model perceptronu

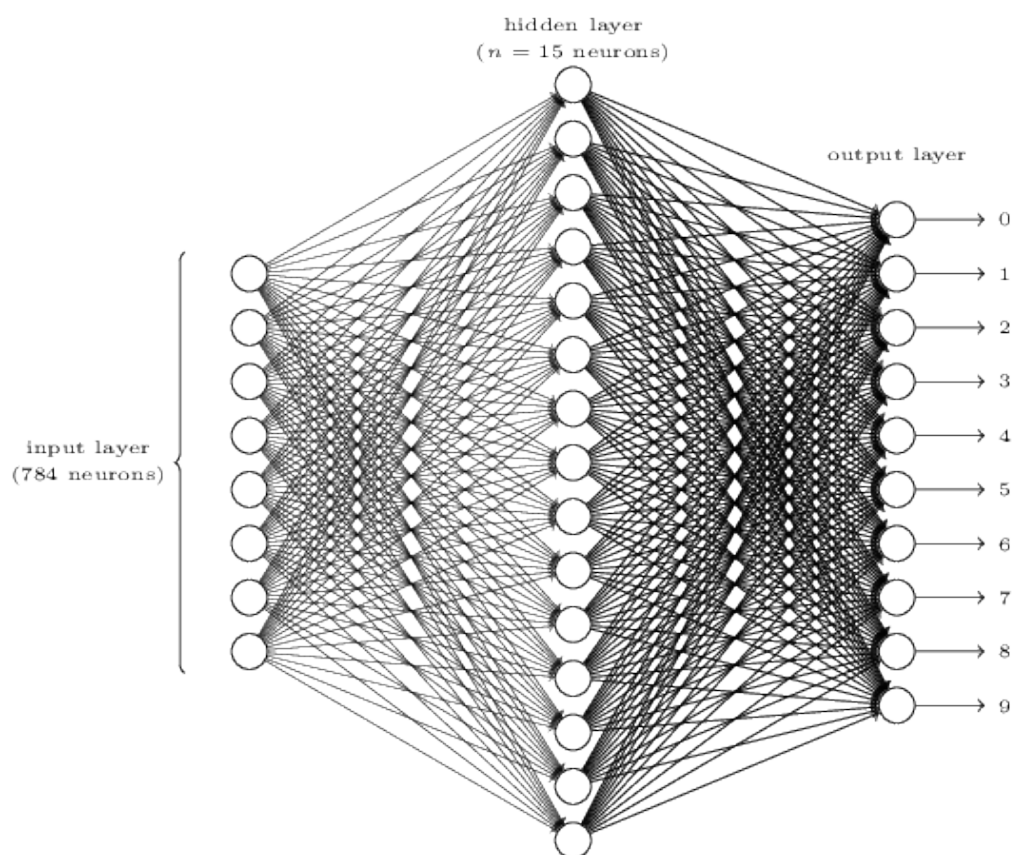
**Zdroj:** <https://nl.wikipedia.org>

$w$  nazýváme vektor vah,  $b$  je definován jako předem určená konstanta. Vektor vah  $w = [w_1, w_2, w_3, \dots, w_n]$  je klíčovou součástí pro správně fungující neuronovou síť.

## 3.2 Vrstvy

Neuronová síť je tvořena skupinami neuronů, které nazýváme vrstvy. Vrstva je modul pro zpracování dat, který přijímá jeden nebo více tenzorů (tenzor je zobecněním pojmu vektor) na vstupu a vrací jeden nebo více tenzorů na výstupu. Vstupní vrstva je obvykle jedna a je první z množiny vrstev. Po ní následují jedna nebo více vrstev skrytých, jejichž parametry si síť při trénování upravuje, aby dosáhla co nejvyšší přesnosti predikce. Poslední vrstvu nazýváme výstupní. Jako příklad můžeme uvažovat neuronovou síť po které požadujeme aby byla schopna rozeznat ručně napsaná čísla 0-9. Vstupní vrstva přijme naskenovaný obrázek číslice. Skryté vrstvy si při učení vytvořily vlastní příznaky pro rozeznávání. Obrázek bude dělen a podle těchto příznaků se vyhodnotí jaká číslice byla do sítě vložena. Výstupní vrstva, ve které bude 10 neuronů očíslovaných 0-9, nám vrátí výsledek jako pravděpodobností rozdělení u každého neuronu v této vrstvě. Ilustrace tohoto příkladu je vidět na obrázku 3.3.

U umělé neuronové sítě rozeznáváme dvě fáze její činnosti. První z nich nazýváme fáze učení nebo trénování neuronové sítě, kdy pomocí trénovacího algoritmu upravuje své parametry vah, v případě konvolučních neuronových sítí filtry, aby produkovala co nejpřesnější a nejkvalitnější výstupy. Druhá



Obrázek 3.3: Příklad neuronové sítě pro rozeznávání ručně napsaných číslic  
**Zdroj:** <https://lh3.googleusercontent.com/>

fáze predikuje výsledky dle modelu.

### 3.3 Učení umělé neuronové sítě

Učení neuronové sítě znamená snahu nalézt takovou kombinaci parametrů, která minimalizuje ztrátovou funkci (Ztráta je hodnota, která reprezentuje míru úspěšnosti řešení dané úlohy.) pro daný soubor příkladů trénovacích dat a jejich cíle. Při procesu učení předkládáme modelu vstup spolu s očekávaným výstupem (ground-truth) a provede se výpočet gradientu parametrů sítě s ohledem na ztrátu. Parametry se poté změny (rychlost změny je definována rychlostí učení) v opačném směru od gradientu.[2]

#### 3.3.1 Trénování s učitelem

Trénování s učitelem je jedna z metod učení neuronové sítě. Předem musíme znát požadovaný výsledek našeho snažení. Neuronové síti vždy předložíme

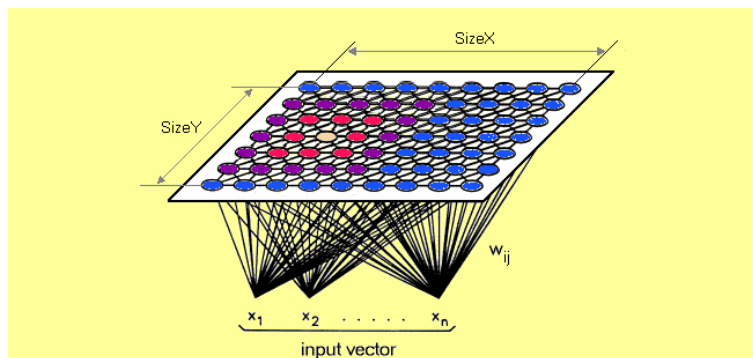
vstupní data a výsledek který jsme obdrželi ze sítě porovnáme s naším požadovaným výsledkem. Následuje spočtení o kolik je nutné upravit hodnoty vah, abychom co nejvíce minimalizovali hodnotu chyby. Tento postup je opakován do té doby než jsme s hodnotou chyby spokojeni, nebo-li dokud získaná chyba není menší než námi definována minimální chyba.

### 3.3.2 Trénování bez učitele

Při trénování bez učitele neznáme požadovaný výstup. Pouze do neuronové sítě vložíme datovou množinu a necháme síť ať si sama vytvoří skupiny a jejich příznaky. Trénování bez učitele je důležité při zpracování obsahu multimédií, kde je požadováno obvykle shlukování nebo rozdělení různých skupin dat.

#### SOM - Self organizing map

Samoorganizační mapy patří mezi umělé neuronové sítě, které jsou trénovány bez učitele. Při učení je generována dvourozměrná mapa. Tyto mapy jsou jedním z nástrojů pro shlukovou analýzu, tedy dělení do několika tříd. Hlavní vlastností samoorganizačních map, uspořádanost mapování se hojně využívá v mnoho oblastech průmyslu. Existuje mnoho samoorganizačních map, mezi nejznámější patří Kohonenova samoorganizační mapa, LVQ nebo třeba rostoucí samoorganizační mapa (GSOM).[9]

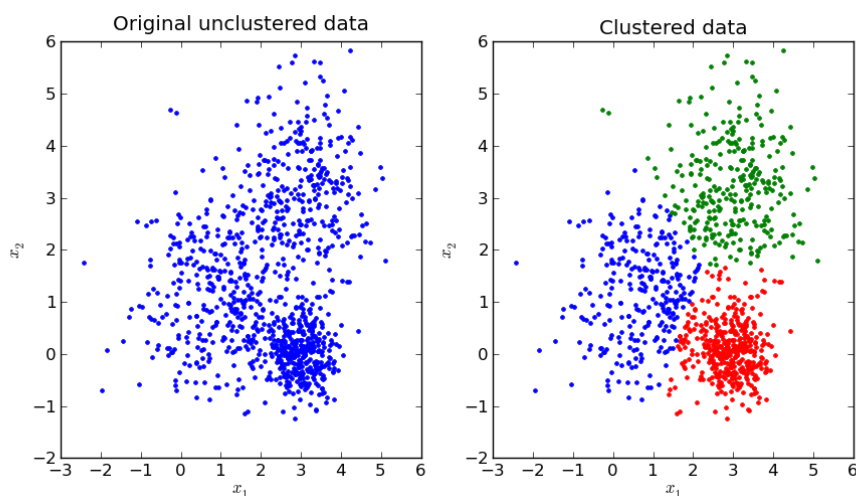


Obrázek 3.4: Samoorganizační mapa - SOM

Zdroj: <https://miro.medium.com/>

## 3.4 Konvoluční neuronová síť - CNN

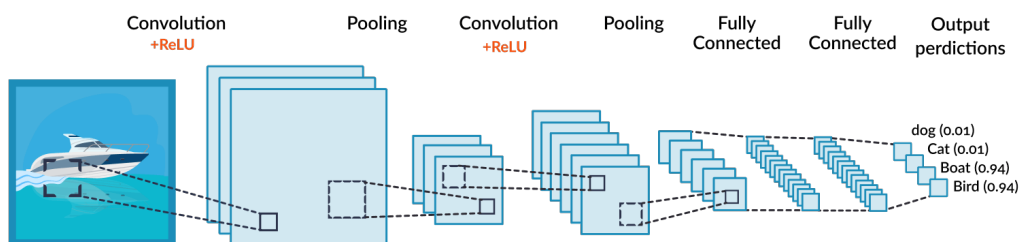
Konvoluční neuronová síť si hledá sama vlastní příznaky pomocí trénování, kdy síti předkládáme sady vstupních dat u kterých známe výsledek. Pou-



Obrázek 3.5: Ukázka shlukovací metody

**Zdroj:** <https://miro.medium.com>

živáme je pro složité úlohy mezi které patří například rozeznávání zvuků, textu, videí a obrázků. Trénování větších a složitějších CNN je hardwarově i časově náročná operace. Z tohoto důvodu se obvykle sítě trénují pomocí grafických karet, které poskytují mnohem vyšší výpočetní výkon než pouhý procesor, pomocí různých hardwarových a softwarových architektur. Mezi tyto architektury patří Compute Unified Device Architecture nebo-li CUDA vyvinutá společností nVIDIA, nebo AMD FireStream pro grafické karty značky AMD. Umožňují spouštět programy napsané v C, C++ a mnoho dalších přímo na GPU.



Obrázek 3.6: Konvoluční neuronová síť

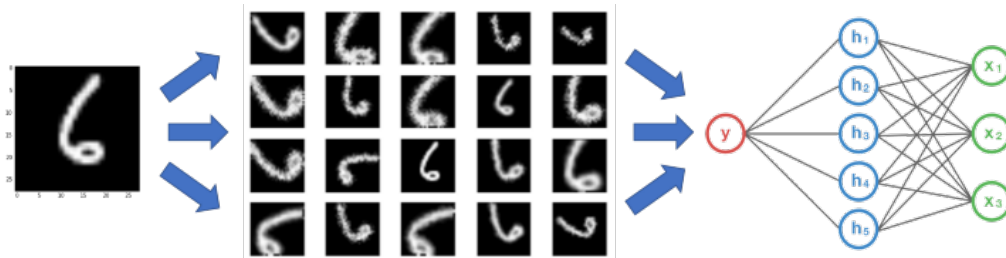
**Zdroj:** <https://missinglink.ai>

### 3.4.1 Data augmentation

Data augmentation je technika používaná pokud máme k dispozici pouze omezené množství obrázků v datasetu. Vytváříme modifikované verze ob-



rázků otáčením či jinou deformací. Použití data augmentation často vede k lepším výsledkům než při použití velkého množství různorodých dat. V zásadě metody data augmentation zvyšují variabilitu dat, ale nepřidávají žádné nové informace do datasetu.[8]



Obrázek 3.7: Data augmentation

**Zdroj:** <https://nanonets.com>

## 3.5 Plně propojené konvoluční neuronové sítě

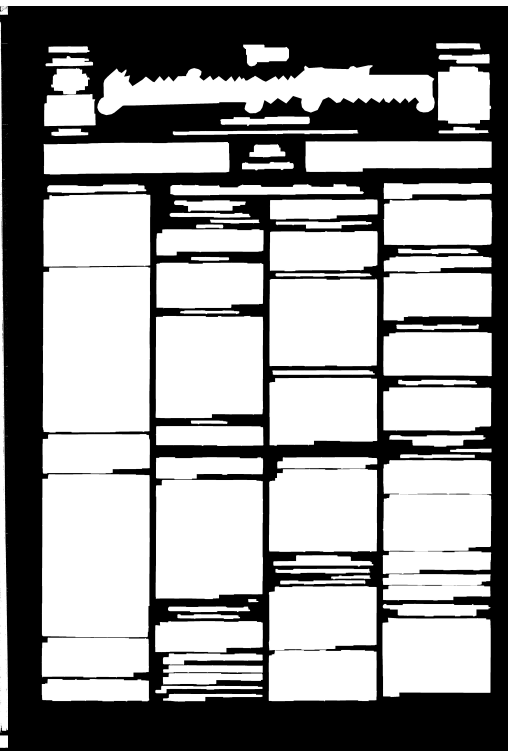
### 3.5.1 UNet

UNet je plně konvoluční neuronová síť určená pro detekci oblastí zájmu a následné vytvoření předběžné binární masky zvýrazňující kde se tyto oblasti nacházejí.

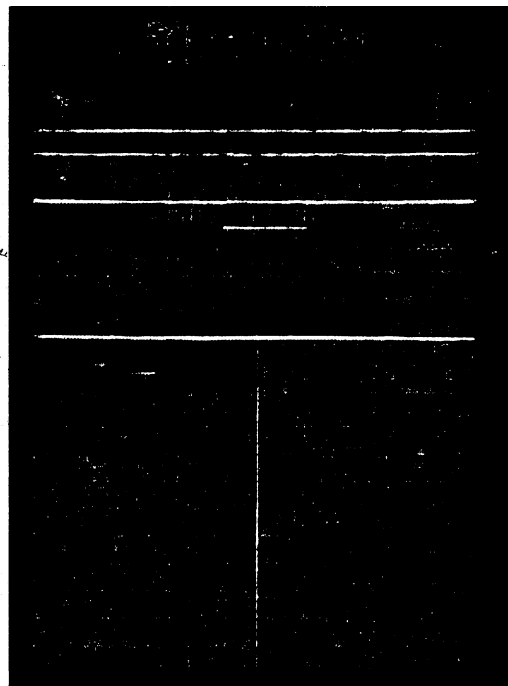
Model je 24-vrstvá plně propojená struktura neuronové sítě, kdy vstupním parametrem je část vstupního obrázku o určité velikosti ve formátu (výška, šířka, hloubka). Standardní velikostí je (256,256,1), obvykle budeme chtít predikovat obrázek s mnohem větším rozlišením, proto musí být vstup rozdělen na mnoho částí a každá se vyhodnotí zvlášť a nakonec se opět složí dohromady jako výstupní maska.[5]

### 3.5.2 AruNet

AruNet, hluboká neuronová síť, byla navržena pro detekci řádků textu v dokumentech. My ji však budeme využívat pro predikci separovacích prvků na obrázku jakými jsou svislé a vodorovné čáry oddělující jednotlivé textové bloky. Architektura sítě byla inspirována architekturou plně konvoluční sítě UNet a je rozšířena o reziduální bloky a mechanismus attention.[3]



Obrázek 3.8: Očekávaný výstup UNetu, binární maska zobrazující oblasti zájmu



Obrázek 3.9: AruNet predikce separátorů

# 4 Data

## 4.1 Europeana

<sup>1</sup>Europeana Collections obsahují více než 50 milionů digitalizovaných položek mezi které patří knihy, dokumenty a mnoho dalších. Ve své kolekci nabízí dataset obsahující zdigitalizované stránky z historických novin. Ke každé stránce je k dispozici ještě PAGE dokument ve formátu xml obsahující veškeré informace o separátorech, textových regionech a mnoho dalších důležitých a užitečných informací. Datová sada Europeana byla využita při trénování konvolučních neuronových sítí UNet a AruNet.

Dataset je složen z 84 stránek historických dokumentů společně s PAGE xml soubory obsahující kompletní anotaci a veškeré informace o každém z dokumentů. Na obrázku 4.1 je jeden z příkladů Europeana dokumentů, celý dataset je možné si prohlédnout v příloze. Dataset Europeana byl při trénování využíván pro prvotní trénovací proces nového modelu a použit pro tzv. Transfer learning - přenos znalostí z jiné, větší datové sady.

## 4.2 Porta fontium

<sup>2</sup>Dataset Ascher Zeitung projektu Porta fontium obsahuje německy psané, zdigitalizované noviny z 19. století. Bohužel data nebyla kvalitně naskenována. Vybledlé oblasti, deformované písmo, nevycentrované stránky jsou jen některé z vad, které dokumenty obsahují. K datasetu je dodáván PAGE dokument obsahující informace o rozložení stránky, umístění textových oblastí nebo obsahu textu a také kompletní přepis dokumentu. Dataset Porta fontium byl při trénování využíván k dotrénování modelu.[4] Soubor příkladů obsahuje 25 různorodých obrázků, 10 určených pro trénovací, 10 pro testovací a 5 pro validační účely. Jednu ze stránek si je možné prohlédnout na obrázku 4.2, zbývající jsou umístěny v příloze.

---

<sup>1</sup><https://www.europeana.eu/>

<sup>2</sup><http://www.portafontium.cz/>

## 4.3 PAGE

<sup>3</sup>PAGE nebo-li Page Analysis and Ground-truth Elements slouží k uložení všech potřebných informací o obsahu stránky a jejím layoutu ve stromové struktuře souboru xml. PAGE je kolekce xml formátů, která byla vyvinuta a je spravována organizací PRIMa Research Lab v University of Salford, UK.

### 4.3.1 Struktura PAGE

Soubor PAGE formátu xml obsahuje následující informace:

1. Metadata o autorovi, datumu vytvoření, názvu obrázku a jeho rozměrech
2. Množinu RegionRefIndexed složící k udržení informace o pořadí bloků
3. Množinu TextRegion obsahující informace o souřadnicích regionu, jeho typ, index a popřípadě i textový obsah
4. Množinu SeparatorRegion obsahující index a souřadnice jednotlivých oddělovačů textových bloků

Společně vytváří kompletní anotaci stránky obrazového dokumentu a s její pomocí je možné vyhodnotit kvalitu provedené segmentace aplikací Layout Evaluation.

## 4.4 Layout Evaluation

<sup>4</sup>Layout Evaluation je nástroj pro grafické zobrazení obsahu PAGE souboru a vyhodnocení rozdílu mezi ground-truth a výsledkem segmentace. V této bakalářské práci je využit pro zhodnocení výsledného vygenerovaného PAGE xml souboru. Obrázek 4.3 přehledně graficky vyobrazuje rozdíly.

### Grafické zobrazení vyhodnocení

- Červená - Barva zobrazující chybějící celou anotovanou oblast nebo alespoň její část
- Světle modrá - False Detection nebo-li chybná detekce signalizuje že jsme anotovali oblast, která chybí v ground-truth.

---

<sup>3</sup><https://www.primaresearch.org/>

<sup>4</sup><https://www.primaresearch.org/tools/PerformanceEvaluation>

- Oranžová - Je označení pro oblast, jejíž malá část chybí, nebo špatně definovaný typ bloku.
- Svislé čáry žlutá barva - Spojené oblasti
- Vodorovné čáry žlutá barva - Rozdělené oblasti

# H. Budelmann, Rathhausmarkt 7, Rathhausstr. 22.

## Grosser Herbst-Ausverkauf reinwollener Kleiderstoffe.

**Portièren**  
das Stück  
von 4 Mk. bis 45 Mk.  
Kosten nach dem Zoll-  
Anschluss 2-6 Mk.  
mehr.

Schwarze  
und farbige  
**Seide**  
für Costime.  
Gedieg. Qualitäten.  
von 1.70-9.50  
das Meter.

### J. Braun

Wegen  
**Zoll-Anschlusses**

Räumung aller ausländischen Fabrikate,  
sowie aller derjenigen Waaren, welche nach erfolgtem  
Anschluss bedeutend theurer werden.

Von den vielen Artikeln nennen heute:

|  |   |  |
|--|---|--|
| <b>Kleiderstoffe</b><br>In allen Farben, Gestreift und glatt.<br>Das Meter 60 Pfg., 1 Mk., 1.50 Mk.,<br>2 Mk. u. s. w.                   | <b>Tricot-Anzüge.</b><br>für Knaben. Jede Grösse. — Baste<br>Waare 5 Mk., 7 Mk. 9 Mk.   | <b>Tischzeug und Leinen.</b><br>Für 6-8-12 Pers. Für Bett-<br>1.20 Mk., 1.75 Mk. d. Meter u. st. Pfg.<br>2.25, 3 Mk., 4.00 Mk. Das 3.25 Mk.                  |
| <b>Winter-Mäntel.</b><br>Aus hochmodernen Stoffen.<br>Elegante Formen.<br>45 Mk., 60 Mk., 75 Mk. u. s. w.                                | <b>Engl. Tuil-Gardinen.</b><br>kosten nach dem Zollanschluss 25 %<br>mehr.  | <b>Halb-Leinen.</b><br>für Bett- und Leibwäsche das Meter<br>32 Pfg., 40 Pfg., 45 Pfg. bis 1.10 Mk.  |
| <b>Regen-Mäntel.</b><br>Absolut wasserdicht. Schöne Formen.<br>Von 4 Mk.—30 Mk.  | <b>Teppich-Rollenwaare.</b><br>kosten nach dem Zollanschluss 1 Mk.<br>1.50 Mk. d. Meter mehr.                                     | <b>Handtücher.</b><br>Gesicht- und Ueberhang-<br>Handtücher.<br>das Dutzend Das Stück<br>6.90, 7.50, 11 Mk. 1.25, 1.50, 2.50,<br>13.00-17.00 Mk. 3.50-10 Mk. |
| <b>Herren-Garderobe.</b><br>Anzüge von bestem Buckskin<br>45 Mk., 55 Mk. u. s. w.<br>Regen-Öcker, neue Stoffe,<br>80 Mk., 80 Mk., 40 Mk. | <b>Abgepasste Teppiche.</b><br>kosten nach dem Zollanschluss 25 %<br>mehr.  |  |
|  | <b>Schlaf- und Reise-<br/>Decken.</b><br>weiss und farbig.<br>1.75 Mk., 2.50 Mk., 5 Mk., 6.25 Mk.,<br>8.50 Mk., 12 Mk. bis 25 Mk. |  |

Schwarze  
und farbige  
**Seide**  
für Costime.  
Gedieg. Qualitäten.  
von Mk. 1.70-9.50  
das Meter.

**Reste  
und  
Coupons**  
von Kleiderstoffen  
und Leinen - Waaren  
in passenden Längen.

**Vollkommener Schutz gegen Fieber**  
durch die  
**Imperial-Genera-  
Lösch-Granaten.**  
Das zuverlässigste und präziseste Stere-  
mittel bei Fieber.  
Stets bereit. — Versagt nie.  
Ist ein Fund in jeder gegen Fieber ohne  
Bitterkeit.  
Vertrieb:  
**F. H. Langsdorff, Rathhausstrasse 10.**

**Ausverkauf!**  
Mit Bestenwillen hat Herrm. Langsdorff, aus  
Opfer, Jagd, Kunst u. Brief-Verfertiger,  
Stereotyp, Pantoscope u.  
ausverkauft werden. Die Güter sind durch  
ausgezeichnete Preise zu erhalten, wenn man zum  
**W. Basilius, Reihstr. 245, Altona.**

Actien-Gesellschaft  
**Schaumweinfabrik Wachenheim.**  
Bester und billigster Schaumwein, Kraut für  
zu beziehen durch Wein- und Delicatessen-  
Handlungen und das  
General-Depot für Hamburg und Umgegend  
grosse Johannisstrasse 10, II.

Ihr mancherlei Nachfragen zu thun!  
**Adolph Ludolphi**  
Herbinandstraße 76. Fernsprecher 666.  
Viel Geschäftsbis  
**Amerikan. Seisaulagen.**  
Wichtige Niederlage der  
Original-Amerik. „Garland“-Sloves  
ausgezeichnet mit Währungs- Besitztümern zu beschaffen.  
**Löhndolfsche Patent-Oelen.**  
118er 700 Centa in Hamburg in Betrieb.

Die billigste Papervereinigung in Deutschland.  
**Berliner Abendpost**  
erschient täglich (wöchentlich sechs Mal).  
Abonnement **1 Mark** vierteljährlich.  
Die „Berliner Abendpost“ ist eine politische Par-  
teipresse mit einer Zeitung der Tages- und Freizeit, der  
raschen und sachlichen Berichterstattung. Tägliche Sitzen-  
und Handbücher. — Im Feuilleton beginnt Anfangs Oktober  
der höchst interessante Roman: „Bosona Matuschok“ von  
C. Dreyer.  
Die „Berliner Abendpost“ gelangt am Morgen in  
Hamburg zur Ausgabe.  
Abonnements für das Oktober-Quartal zu Eins Mark  
weiss man bei der Postamtlich.  
Für Anzeigen: Zeitungspreis 20 Pfg. Berlin SW, Kochstr. 23.  
Auswärts **42,000** Abonnenten.

Künftiger Jahne unter Garantie, 2 und 3 Mark.  
Rich. Paul, Zehnthofstr., Bangl. Strassen 651.

Obrázek 4.1: Ukázka dokumentu ze sady Europeana

### Bekanntmachung.

Den Herren Fleischhauern sowie Privat-schlächtern von Aisch zur Anzeige, daß jeder der genannten, der ein Stück Schlachtvieh, als: Rindvieh, Schweine, Kälber, Schafe zc. schlachtet, bei Vermeidung von Unannehmlichkeiten vorher bei dem Unterzeichneten Meldung zu machen hat.

Johannes Voit  
am Stein Nr. 166.

### Vorläufige Anzeige.

Freunden des Gesangs zur Nachricht kündigt unterzeichneter Verein die Abhaltung einer

**Gesangs-Produktion**

Sonntag den 14. Januar im hiesigen Schießhause, vorläufig an.

Der Männergesangsverein.

### Guter Hafer

wird zu kaufen gesucht.  
Wo? sagt die Expedition d. Bl.

### Verkauf.

50 Centner Heu bester Qualität sind billig zu verkaufen.  
Wo? sagt die Expedition d. Bl.

### Empfehlung.

Steinkohlen in 10 Sorten von bester Pechkohle bis zur Maschinentohle liefern ab Zwickau und Lugau sowohl in Lowrys als auch in größten Quantitäten bei Abschluß und bitte um gef. Auftragserteilung.

Bad Elster, den 1. Januar 1866.

F. Fritsch.

### 5 Stück gläserne Salon-Lustre,

4 à 6 Flammen und 1 à 12 Flammen sind billig zu verkaufen. Wo? sagt die Expedition d. Bl.

Verantwortliche Redaktion Druck und Verlag von Gustav Becker in Aich.

### Danksagung.

Für die uns vielfertig bewiesene aufrichtige Theilnahme an dem Kranken- und Todtenbette unsers Sohnes und Bruders des k. k. Oberarztes

**Dr. Adam Hofmann,**

sowie für die zahlreiche Begleitung bei dem Leichenbegängnisse desselben, sagen wir unsern tiefgefühltesten Dank.

Friedersreuth u. Rospach, 2. Januar 1866.

Die trauernden Anverwandten.

+

+

+

II.

### Stilles Beileid.

Der Todesengel kam herab  
Trotz allem Hoffen, Bitten, Flehn!  
Er führt das Mägdlein in das Grab,  
Wo weinend seine Eltern steh'n.

Was mit der einen Hand es gibt,  
Das Schicksal mit der andern nahm!  
Die Blüthe, die Ihr so geliebt,  
Sank welkend, da die and're kam,

O! sende Himmel Deinen Trost,  
In der gepflüchten Eltern Herz!  
Gib Ihnen jene heil'ge Kost  
Die lindernd heilet jeden Schmerz.

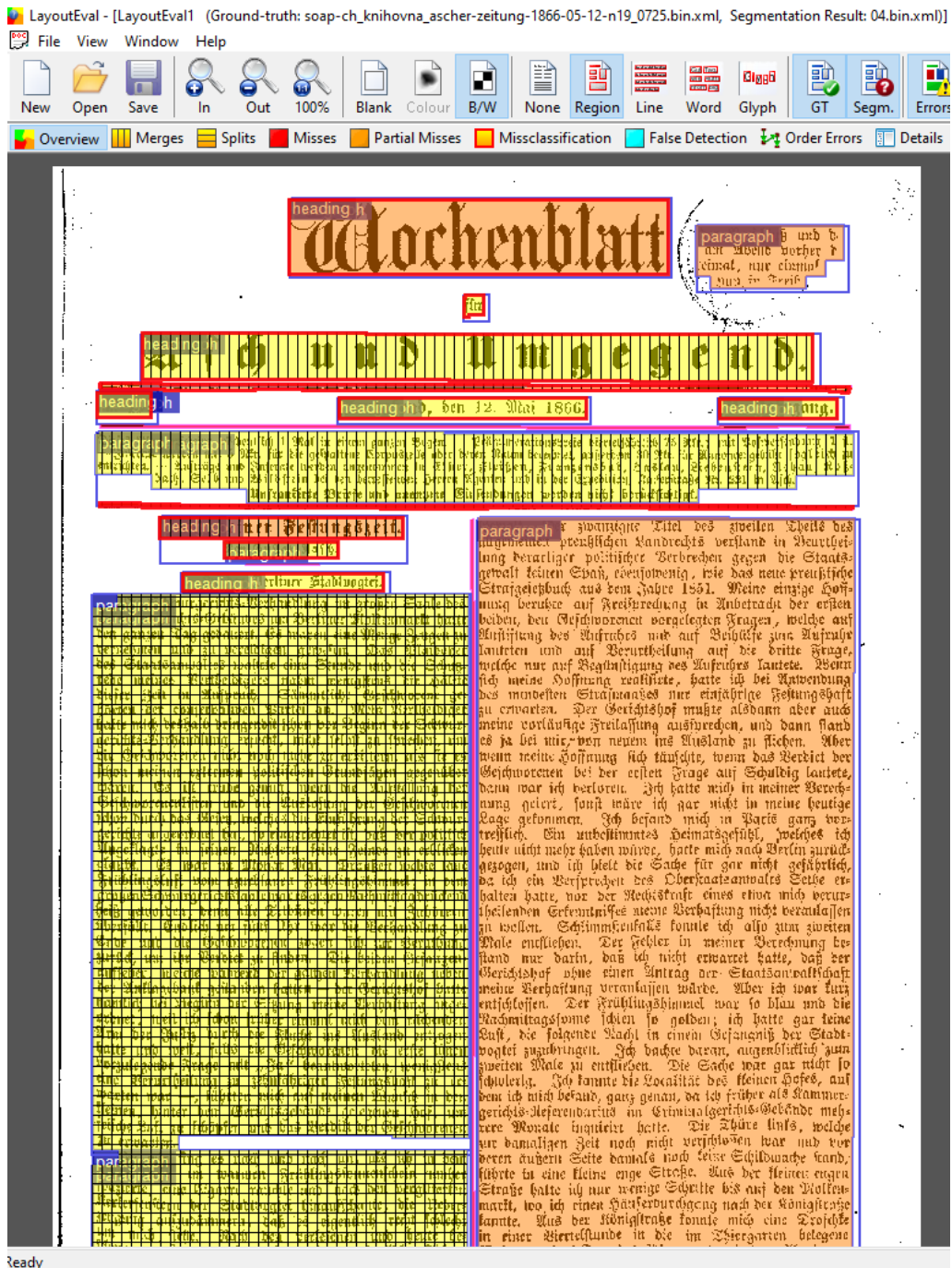
Am 4. Januar.

### Confiance 249.

Die Worte, die Sie mir unter der entblätterten Eiche sagten, habe ich nicht vergessen. Sie haben doch ganz Recht, ich fühl' es klar und tief, mit schrecklicher Gewissheit. — Der Weg wird lang sein, öde und dunkel, aber das Licht, das von Ihnen ausgeht soll mir leuchten. Leben Sie wohl, leben Sie tausendmal wohl, tausendmal wohl! Vergessen Sie mich nicht in Ihrer lieben Seele! Ich werde Ihrer mit ewiger Treue gedenken, werde Ihr Bild mit ewiger Liebe im Herzen tragen und habe nur den Einen Wunsch, Sie einst so glücklich zu sehen, als Sie es verdienen und als ich Sie so gern, ach so gern! gemacht hätte.

G. F. L.

Obrázek 4.2: Ukázka dokumentu ze sady Porta fontium (binarizováno)

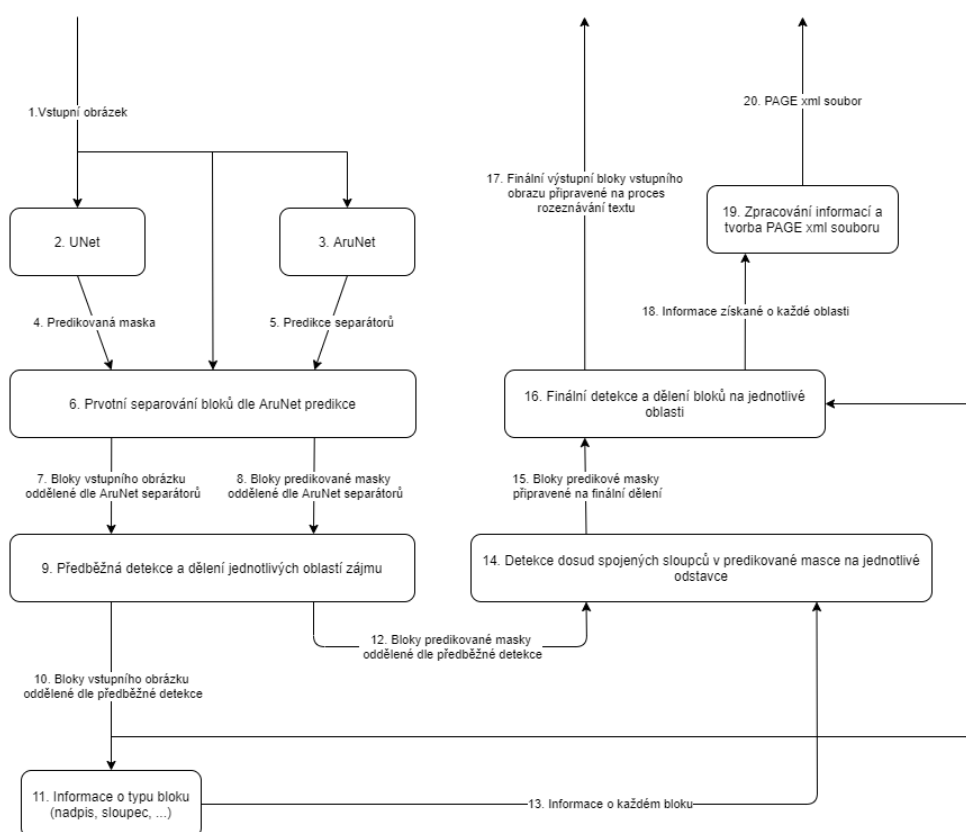


Obrázek 4.3: Ukázka aplikace Layout Evaluation



# 5 Proces segmentace

Samotný proces je složen z několika na sobě závislých kroků, jejichž cílem je vytvořit množinu obrázků jednotlivých oblastí zájmu vstupního obrazového dokumentu a PAGE soubor formátu xml obsahující informace o všech elementech.



Obrázek 5.1: Diagram procesu segmentace

1. **Vstupní obrázek** - Historický obrazový dokument z datové sady Porta fontium upravený procesy binarizace a rotace.
2. **UNet** - Transformace vstupního obrázku na predikovanou masku oblastí zájmu plně propojenou konvoluční neuronovou sítí UNet.
3. **AruNet** - Detekce svislých a horizontálních čar ve vstupním obrázku představující oddělovače jednotlivých oblastí pomocí plně propojené konvoluční neuronové sítě AruNet.

4. **Predikovaná maska** - Binární obrázek bíle zobrazující oblasti zájmu na černém pozadí.
5. **Predikce separátorů** - Binární obrázek bíle zobrazující separátory oddělující oblasti zájmu na černém pozadí.
6. **Prvotní separování bloků dle AruNet predikce** - Vytvoření bloků dle predikce separátorů ze vstupního obrázku a predikované masky.
7. **Bloky vstupního obrázku oddělené dle AruNet separátorů** - Množina bloků vytvořená ze vstupního obrázku obsahující oblasti oddělené dle predikovaných separátorů.
8. **Bloky predikované masky oddělené dle AruNet separátorů** - Množina bloků vytvořená z predikované masky obsahující oblasti oddělené dle predikovaných separátorů.
9. **Předběžná detekce a dělení jednotlivých oblastí zájmu** - Prvotní snaha detekovat jednotlivé oblasti zájmu na již oddělených blocích získaných v předchozím kroku.
10. **Bloky vstupního obrázku oddělené dle předběžné detekce** - Množina bloků vstupního obrázku získaná předběžnou snahou oddělit veškeré oblasti zájmu.
11. **Informace o typu bloku (nadpis, sloupec, ...)** - Na základě velikosti písma, stylu a dalších vlastností snaha určit o jaký typ bloku se jedná, experimenty prováděny na blocích vstupního obrázku získaných v předchozím kroku.
12. **Bloky predikované masky oddělené dle předběžné detekce** - Množina bloků predikované masky získaná předběžnou snahou oddělit veškeré oblasti zájmu.
13. **Informace o každém bloku** - Informace získané v předchozím kroku uložené v dočasných souborech. Důležité pro tvorbu PAGE xml souboru.
14. **Detekce dosud spojených sloupců v predikované masce na jednotlivé odstavce** - Proces detekce dosud spojených odstavců a jiných objektů, které musí být odděleny v blocích predikované masky získané z kroku číslo 9.

15. **Bloky predikované masky připravené na finální dělení** - Bloky predikované masky upravené předchozím krokem připravené na konečné dělení.
16. **Finální detekce a dělení bloků na jednotlivé oblasti** - Vytvoření výstupní množiny bloků vstupního obrázku za pomoci bloků predikované masky získané v kroku 14 a předání informací pro tvorbu xml souboru PAGE.
17. **Finální výstupní bloky vstupního obrazu připravené na proces rozeznávání textu** - Množina bloků vstupního obrázku, upravené pro případný další krok rozeznávání textu.
18. **Informace získané o každé oblasti** - Data obsahující důležité informace o souřadnicích jednotlivých bloků ve vstupním obrázku, jejich typ a mnoho dalšího.
19. **Zpracování informací a tvorba PAGE xml souboru** - Informace získané v kroku 16 je nutno správně zpracovat a zapsat do xml souboru.
20. **PAGE xml soubor** - Cíl našeho snažení, soubor obsahující veškeré informace o každé entitě vstupního obrázku.

# 6 Využité technologie

## 6.1 Python

Python patří mezi nejsnadněji osvojitelné programovací jazyky. Je velmi expresivní a multiplatformní, lze tedy bez problému spustit program napsaný ve Windows i na unixových systémech jako je Linux, Mac OS X. Opravdu silnou stránkou jazyku Python je, že se dodává s již kompletní velmi rozsáhlou standardní knihovnou. V jazyku Python lze programovat v procedurálním, objektivě orientovaném a v menší míře též funkcionálním stylu.[6]

## 6.2 OpenCV-Python

OpenCV je open-source knihovna, jenž se zaměřuje na počítačové vidění a strojové učení. Obsahuje velké množství metod ulehčující manipulaci s multimediálními soubory včetně algoritmů, které mohou být využity pro rozpoznávání obličejů, identifikování objektů, rozeznávání akcí člověka a mnoho dalších.<sup>1</sup>

## 6.3 Tensorflow

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

## 6.4 Keras

Keras je framework pro hluboké učení v Pythonu. Poskytuje jednoduchý způsob, jak definovat a trénovat téměř jakýkoli druh modelu hlubokého učení. Keras byl vyvinut pro rychlé experimentování. Knihovna obsahuje stavební bloky pro vývoj hlubokých učebních modelů. Nepracuje s nízkoúrovňovými operacemi, jako je manipulace s tenzory a derivacemi. Místo toho spoléhá na specializovanou a dobře optimalizovanou tenzorovou knihovnu, v našem případě Tensorflow.[2]

---

<sup>1</sup><https://opencv.org>

### 6.4.1 Keras vrstvy

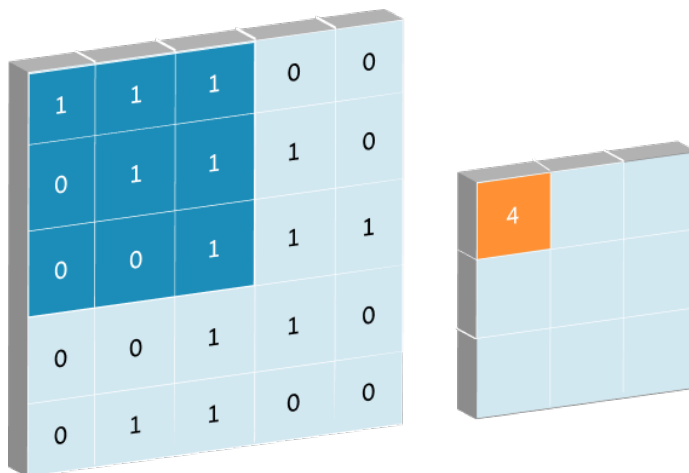
Keras nabízí vrstvy, které jsou vhodné pro různé formáty tenzorů a různé typy zpracování dat.[2] Některé z těchto vrstev, které jsou obsaženy v architektuře sítě UNet je vhodné zde popsat pro lepší pochopení co se při predikci a trénování děje uvnitř této sítě.

#### Input

Vrstva Input inicializuje rozměry tensor objektu, jenž této vrstvě předáme jako vstupní argument.

#### Conv2D

Conv2D je konvoluční vrstva ve dvou rozměrech. Konvoluční vrstva vytvoří jádro, které převede vstupní na výstupní tensor. Převodem dojde k redukci počtu tenzorů a tedy i ke zmenšení počtu parametrů, které je nutné natrénovat.



Obrázek 6.1: Conv2D redukuje pomocí jádra rozměr tensoru

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

#### Conv2DTranspose

Pokud potřebujeme naopak zvětšit rozměr tensoru, Keras nabízí vrstvu Conv2DTranspose s účinkem opačným k Conv2D. Z redukovaného stavu pomocí jádra vytvoří na výstupu tensor zvětšený.

|    | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4  |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5  |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6  |   | 0 | 0 | 1 | 3 | 2 | 1 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7  |   | 0 | 0 | 1 | 3 | 3 | 1 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8  |   | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9  |   | 0 | 0 | 3 | 2 | 3 | 3 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 10 |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 11 |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

|    | Input |   |   |   |   |   |   |   |   | Kernel |  |  | Output |  |  |  |  |  |  |  |  |  |  |  |  |
|----|-------|---|---|---|---|---|---|---|---|--------|--|--|--------|--|--|--|--|--|--|--|--|--|--|--|--|
| 4  |       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 5  |       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 6  |       | 0 | 0 | 1 | 3 | 2 | 1 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 7  |       | 0 | 0 | 1 | 3 | 3 | 1 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 8  |       | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 9  |       | 0 | 0 | 3 | 2 | 3 | 3 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |

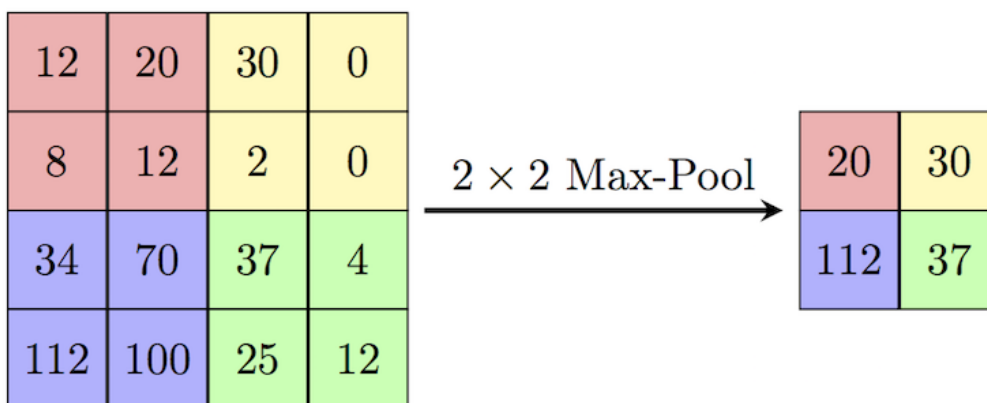
|    | Kernel |  |  | Output |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----|--------|--|--|--------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 6  |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7  |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8  |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9  |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |        |  |  |        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Obrázek 6.2: Conv2DTranspose pomocí jádra zvětšuje rozměr tensoru

Zdroj: <https://miro.medium.com>

### MaxPooling2D

Pooling si můžeme představit jako proces výběru určitých hodnot z malých oblastí v matici vstupního tensoru. Pro max pooling vybere proces z každé oblasti hodnotu největší, pro min pooling hodnotu nejmenší. MaxPooling2D je další z užitečných Keras vrstev. Po nastavení velikosti poolu projde celý vstupní tensor a z každé oblasti vyjme největší hodnotu, kterou zapíše do tensoru výstupního. Stejně jako Conv2D se jedná o vrstvu, která redukuje rozměr tensoru.



Obrázek 6.3: MaxPooling2D vybere největší hodnotu z oblasti

Zdroj: <https://computersciencewiki.org>

# 7 Rozdělení stránky na bloky

Prvním krokem k úspěšnému vytvoření finálního PAGE xml souboru je vytvoření masky, která se co nejvíce podobá ground-truth. Jak již bylo řečeno dříve, maska je binární obrázek zobrazující jednotlivé oblasti zájmu. Oblasti zájmu v obrazovém dokumentu, zahrnují veškerý text co stránka obsahuje. Na její procentuální odchylku od ground-truth má přímý vliv kvalita natrénovaného UNet modelu. Naším cílem je nalézt takové trénovací parametry, aby daná odchylka byla co možná nejmenší.

## 7.1 Volba modelu

Bylo řečeno, že volba parametrů a trénovacích dat má klíčový vliv na správnost a přesnost výsledného predikovacího modelu. Z tohoto důvodu byl proveden experiment, jehož cílem je nalézt nejlepší model sítě UNet. Pro potřeby experimentu byla z testovací sady datasetu Porta fontium vybrána skupina pěti obrazových dokumentů určených pro validační účely. Ve validační sadě jsou obsaženy různé typy layoutů, která se v datech vyskytují. Celou validační sadu si je možné prohlédnout v příloze.

### 7.1.1 Způsob vyhodnocení správnosti modelu

Vyhodnocení bude provedeno pomocí metriky nazývané Jaccardův koeficient.

#### Jaccardův koeficient podobnosti

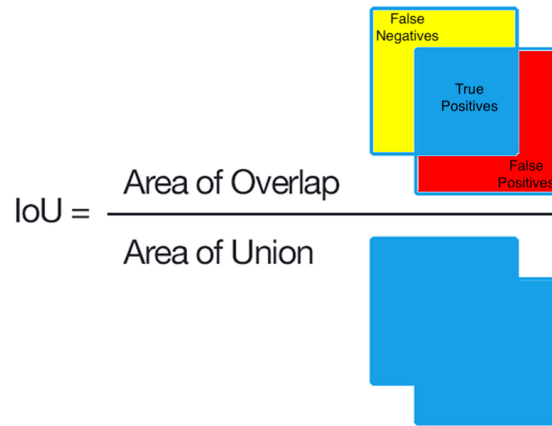
Metrika Jaccardův koeficient podobnosti je používána pro porovnávání podobnosti a odlišnosti množin.[4]

Definujme obrázek predikované masky jako PM a její ground-truth jako GT.

- TruePositives - reprezentuje celkový počet pixelů kde PM i GT nabývají hodnoty 1
- FalseNegatives - reprezentuje celkový počet pixelů kde PM nabývá hodnoty 0 zatímco GT hodnoty 1
- FalsePositives - reprezentuje celkový počet pixelů kde GT nabývá hodnoty 0 zatímco PM hodnoty 1

Poté Jaccardův index (J) podobnosti získáme z tvaru:

$$J = \frac{TruePositives}{FalseNegatives + FalsePositives + TruePositives}$$



Obrázek 7.1: Intersection over union nebo-li Jaccardův koeficient podobnosti

**Zdroj:** <https://i.imgur.com/>

### 7.1.2 Testované trénovací parametry experimentu

Změnou vstupních parametrů je možné docílit značně rozdílných výsledných predikcí. Byl tedy vytvořen experiment, jehož cílem je nalézt takovou kombinaci parametrů, se kterými dosáhneme nejlepší predikované masky vyhodnocené dle metriky Jaccardův koeficient podobnosti. V experimentu byla testována následující variace vstupních parametrů:

1. Trénovací vstupní rozměr (256,256,1), Dataset (Europeana), Vstupní model (/), Výstupní rozměr (256,256,1)
2. Trénovací vstupní rozměr (320,256,1), Dataset (Europeana), Vstupní model (/), Výstupní rozměr (320,256,1)
3. Trénovací vstupní rozměr (256,256,1), Dataset (Porta fontium), Vstupní model (1), Výstupní rozměr (256,256,1)
4. Trénovací vstupní rozměr (320,256,1), Dataset (Porta fontium), Vstupní model (2), Výstupní rozměr (320,256,1)
5. Trénovací vstupní rozměr (320,256,1), Dataset (Porta fontium), Vstupní model (2), Výstupní rozměr (640,512,1)



| Jaccardův koeficient podobnosti |         |       |       |       |       |        |         |
|---------------------------------|---------|-------|-------|-------|-------|--------|---------|
| Model                           | Dataset | 1     | 2     | 3     | 4     | 5      | avg     |
| 1                               | E       | 0,794 | 0,829 | 0,891 | 0,802 | 0,8125 | 0,8257  |
| 2                               | E       | 0,826 | 0,845 | 0,91  | 0,85  | 0,847  | 0,85556 |
| 3                               | E + P   | 0,832 | 0,831 | 0,909 | 0,787 | 0,813  | 0,8344  |
| 4                               | E + P   | 0,878 | 0,89  | 0,94  | 0,902 | 0,896  | 0,9012  |
| 5                               | E + P   | 0,84  | 0,876 | 0,912 | 0,811 | 0,840  | 0,8558  |
| 6                               | E + P   | 0,587 | 0,64  | 0,614 | 0,611 | 0,616  | 0,6136  |
| 7                               | E       | 0,865 | 0,85  | 0,936 | 0,872 | 0,862  | 0,877   |
| 8                               | E       | 0,875 | 0,863 | 0,938 | 0,883 | 0,867  | 0,8852  |
| 9                               | E + P   | 0,834 | 0,851 | 0,906 | 0,824 | 0,81   | 0,845   |
| 10                              | E + P   | 0,792 | 0,790 | 0,874 | 0,827 | 0,807  | 0,818   |

Tabulka 7.1: Hodnoty Jaccardova koeficientu podobnosti validační sady pro každý testovaný model

6. Trénovací vstupní rozměr (320,256,1), Dataset (Porta fontium), Vstupní model (2), Výstupní rozměr (1280,1024,1)
7. Trénovací vstupní rozměr (640,512,1), Dataset (Europeana), Vstupní model (/), Výstupní rozměr (640,512,1)
8. Trénovací vstupní rozměr (640,512,1), Dataset (Europeana), Vstupní model (/), Výstupní rozměr (320,256,1)
9. Trénovací vstupní rozměr (640,512,1), Dataset (Porta fontium), Vstupní model (7), Výstupní rozměr (640,512,1)
10. Trénovací vstupní rozměr (640,512,1), Dataset (Porta fontium), Vstupní model (7), Výstupní rozměr (320,256,1)

Rozměry větší než (1280,1024,1) nebo menší než (256,256,1) dosahovaly veškeré špatných výsledků už při vizuální kontrole, protože predikce naprosto zničí strukturu dokumentu a z toho důvodu nebyli vůbec zařazeny do vyhodnocení.

### 7.1.3 Výsledné hodnoty

Veškeré podklady, ground-truth, predikované masky a skript použitý pro vyhodnocení hodnot tabulky 7.1 jsou k dispozici v příloze.

### 7.1.4 Zhodnocení

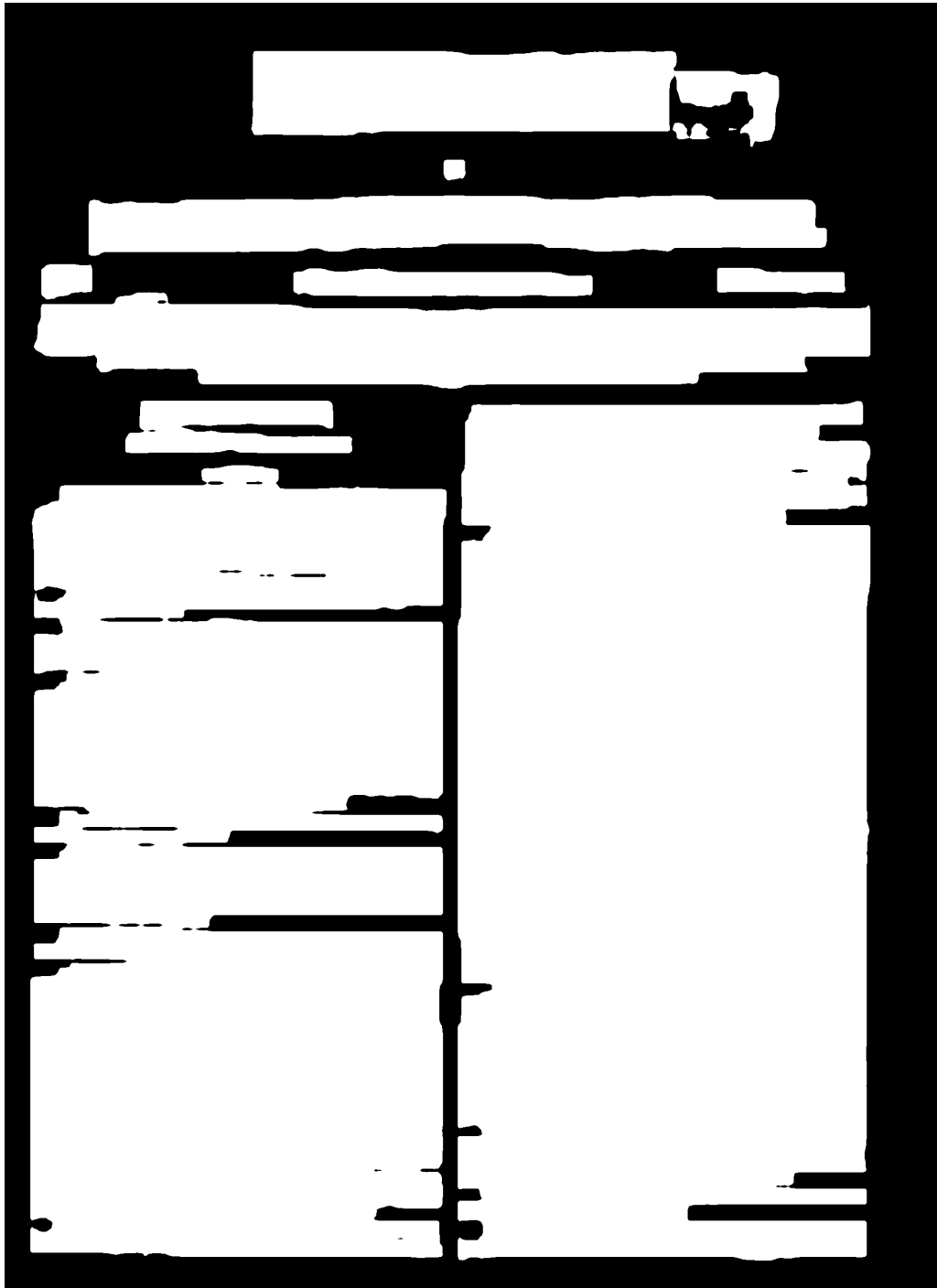
Na základě výsledků získaných metrikou Intersection over union byl pro proces segmentace vybrán model číslo 4 s vysokou přesností přesahující hranici 90 procent. Obrázek 7.2 je zobrazením jedné z validačních masek vytvořené modelem číslo 4. Modely trénované na nižším rozlišení podrobené většímu vstupnímu rozměru nedosahovaly takových hodnot, avšak vytváří zajímavý jev, kdy model snažící se detekovat odstavce, detekuje jednotlivá slova viz obrázek 7.3. Tohoto jevu by bylo možné využít pokud bychom si přáli separovat jednotlivá slova.

## 7.2 Predikce separátorů

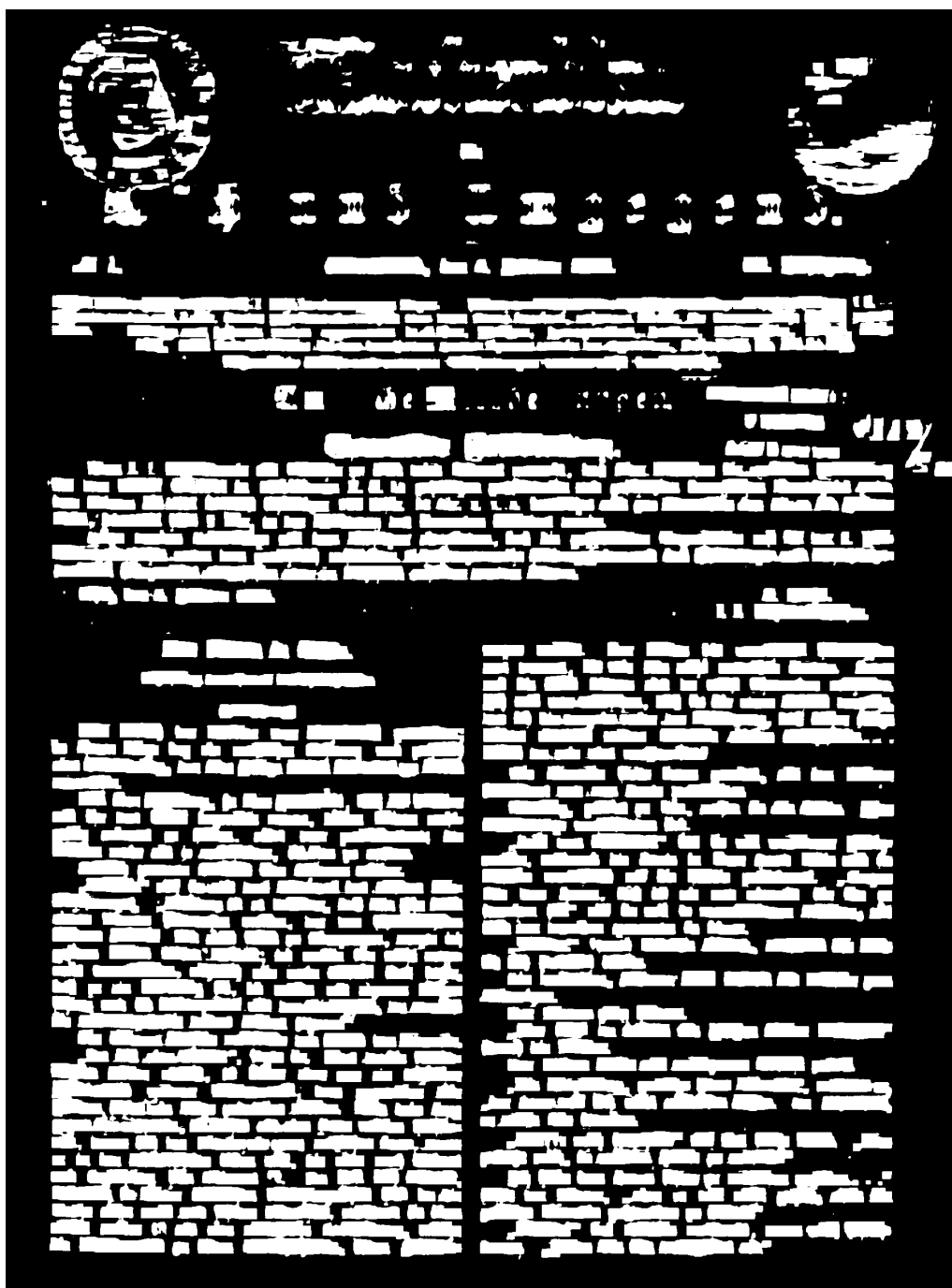
Vstupní historický obrazový dokument obsahuje řadu svislých a vodorovných čar oddělující jednotlivé texty a části. Tyto oddělovače nebo-li separátory jsme schopni detekovat a s jejich pomocí provést prvotní dělení na bloky. Detekci provede plně konvoluční neuronová síť AruNet popsaná v kapitole o umělých neuronových sítích.

Výstupní produkt AruNetu, predikovanou binární masku separátorů, předám algoritmu, který na základě jejich polohy oddělí jednotlivé části. Obrázek 7.4 prezentuje ukázkou výsledné predikované masky získané ze sítě AruNet.

1. Obrázek predikované masky separátorů je podroben morphologické operaci dilatace pro zesílení detekovaných oddělovacích čar.
2. Algoritmus nejprve prochází obrázek a snaží se detekovat horizontální separátory.
3. V okamžiku kdy nějaký najde, pokusí se detekovat, zda jsou pod tímto horizontálním oddělovačem nějaké svislé.
4. Pokud ano, rozdělí je na jednotlivé bloky, v opačném případě oddělí blok v celé šířce stránky.
5. Kroky 2, 3 a 4 se opakují do doby, dokud algoritmus neprojde celou výškou stránky.



Obrázek 7.2: Ukázka predikované masky získané od nejlepšího modelu



Obrázek 7.3: Pokud trénovací rozměr je nižší než rozměr vstupní, může vzniknout podobný jev.

**Wochenblatt**  
für  
**Nisch und Umgegend.**

Nr. 1.      Sonnabend, den 6. Januar 1866.      III. Jahrgang.

**Amtlliche Mittheilungen.**

**Executive-Vertheilung.**

Dem I. I. Bezirksamte als Obdient zu 200 und bekannt gemacht, daß über Kaufleute die executive Vertheilung bei dem Nisch-Gläser in Schilber N. C. 15 geordneten gerichtlichen Verfahren p. c. o. dem 16. u. 30. Jänner 1866, f. d. m. 1866, in der Wohnung des Excenten vollzogen wurde.

Diesem werden Kaufleute mit dem Beschlusse eingeladen, daß die geordneten Gegenstände erst bei der 2. Vertheilungshandlung unter dem gerichtlichen Aufsicht des Obdienten und des Obdienten-Protokoll-Bürogehilfen eingesehen oder in Abschrift erhoben werden können.

I. I. Bezirksrichter.

**Der Mann in Eifer.**  
Erzählung von Hans Hagenstein.

(Fortsetzung)

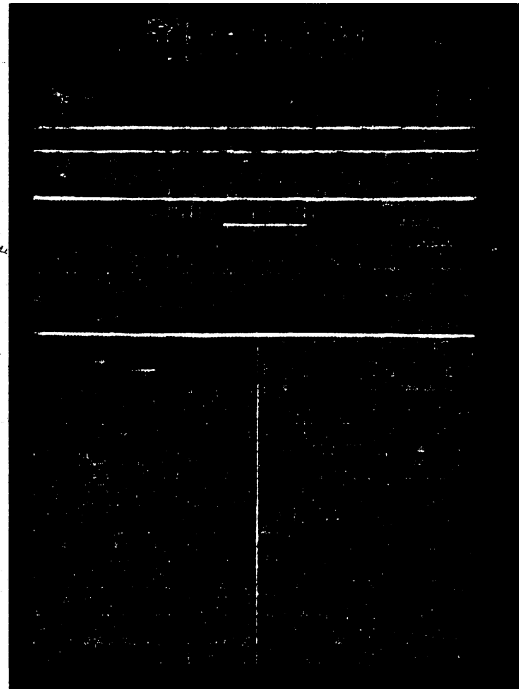
Das Klapp der Tochter war tiefdenklich gewesen; in ihrem Auge lag ein unheiliger Schmerz und zugleich eine Schärfe, die ihre Züge ihren Ausdruck zu geben vermochte.

Wie ihre Stimmung, so ihre Antwort, daß sie Mutter mit ihr das Glück begehren werde, diesen Mann recht artig erkaufen zu lassen, alles was jetzt nöthig sei, die Mutter mehr als je zu unterstützen, sie ihm zu helfen.

„Gutlich gesehn, Eibne ihre Stimme wieder.“

„Mutter!“ rief sie mit dem Ausdruck des tiefsten Entschlusses, „Mutter, ich es bewilligt, daß Sie über mich einem Manne hantieren wollen, der von irgendwelchem Stande ist und mich selbst zu übergeben, magte ich gehen, wenn Sie nicht wollen, ich werde mich nicht ungerne verheirathen, gleich mit dem Obdienten, daß Sie ein Willing sei, ich werde mit eigenen Augen sehen, um ich fortzuwand mit der Obdienten, da werden die Entschlossenheit dieses Mannes zeigen.“

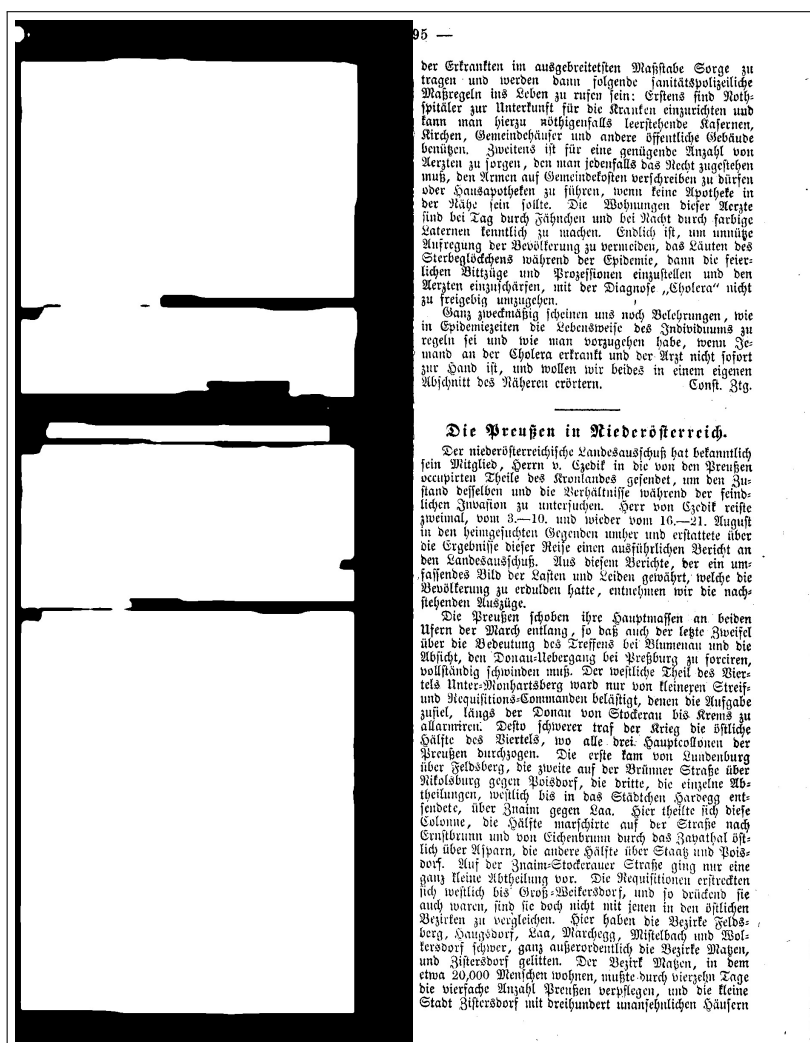
Ein bitterer Schweiß umfloss den Wangen der Hagenstein. „Du bist eine Thörin“, sagte sie. „Was du bei solchem Eifer denkst, ist weder dein noch ein lächerlicher Dummheit, unterer Mannesrecht, dem gegenüber nicht können, wenn ein Mann nicht ganz entzogen kann, wenn er in der Mannhülle leben will, die dich noch führt und Geduld sein Element ist. Du soll dich überreden, Eibne; in deiner kindlichen Unbedachtlichkeit hast du die Sache schon erit und schwer betradet, möddest du dich nicht weiter in die eine allgütige Entscheidung, welche die Mannesrecht zu ihrem Geschieden, ihren Fortw...



Obrázek 7.4: Ukázka predikce jednotlivých oddělovačů v obrázku

# 8 Prvotní detekce oblastí zájmu

Výsledkem předchozího kroku je rozdělení vstupního obrázku pomocí separátorů na velké bloky. Avšak tyto bloky téměř vždy obsahují vícero oblastí zájmu, které ve výsledné situaci musejí být odděleny. Společně s oddělenými bloky vstupního obrázku máme k dispozici ještě stejné bloky získané z predikované masky textu.



Obrázek 8.1: Blok, oddělený pomocí separátorů získaných sítě AruNet (nalevo) a z originálního obrázku (napravo)

## 8.1 Postup

Pro ukázkou máme AruNet blok, který si přejeme rozdělit na jednotlivé oblasti (obrázek 8.1). Blok obsahuje tři velké oblasti zájmu (prozatím berme odstavec jako celý sloupec, oddělování jednotlivých odstavců je postup uvedený v kapitole Detekce spojených odstavců).

1. Načteme libovolný AruNet blok vytvořený z predikované masky a společně s ním i blok stejné oblasti odděleného z původního obrázku.
2. Pro blok predikované masky provedeme jednoduché binární prahování, kdy pokud hodnota je vyšší než hranice, hodnotu pixelu nastavíme na 255 v opačném případě na 0.
3. OpenCV nabízí velmi užitečnou metodu **findContours** pro nalezení kontur, kdy jako parametr vložíme výsledek z kroku předchozího a ona nám vrátí pole souřadnic, rozměrů a jejich hierarchii.
4. Toto pole kontur je procházeno a pokud jsou splněny podmínky popsané níže, je kontura vyříznuta jako samostatný blok dle souřadnic získaných pomocí **findContours** a uložena do složky.
5. Pokud jsou podmínky splněny a blok s maskou je oříznut, stejnými souřadnicemi je upraven i originální blok.

### Podmínky pro vytvoření nového bloku

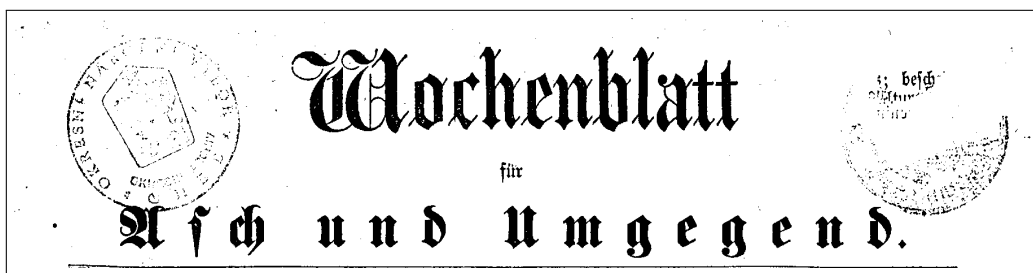
1. Šířka i výška kontury musí dosahovat alespoň hodnoty 25 pixelů, menší kontury jsou nechtěné oblasti nebo zbytkový šum.
2. Kontura nesmí být plně obsažena v kontuře jiné (plocha sjednocení dvou kontur musí být větší než plocha větší kontury)

## 9 Typy oblastí zájmu

Jednotlivé oblasti historických obrazových dokumentů Porta fontium lze rozeznávat dle parametrů jakými jsou například velikost písma nebo množství znaků v oblasti. Tato znalost má využití nejen pro samotné PAGE xml ale i zda můžeme považovat blok za sloupec textu a později se v bloku pokusit detekovat jednotlivé odstavce.

### 9.1 Hlavička stránky

Hlavička obvykle obsahuje hlavní nadpis pro celou stránku (viz. obrázek 9.1). Průměrná velikost znaků je více než dvojnásobná oproti standardní velikosti. Množství znaků dosahuje mnohem menší hodnoty než u bloků obsahujících odstavce.



Obrázek 9.1: Ukázka hlavičky stránky

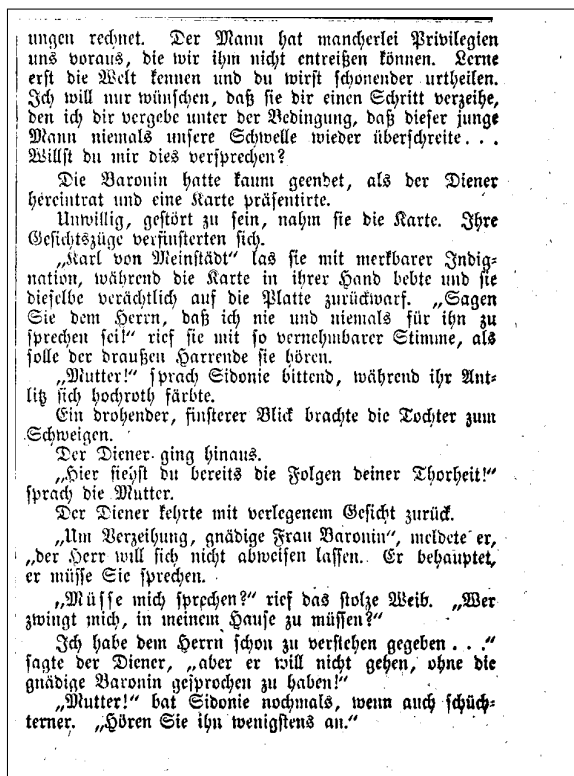
### 9.2 Sloupec standardního textu

Sloupec nebo odstavec textu je složen z velkého množství znaků standardní velikosti písma (viz obrázek 9.2). Na tento typ bloku je použit postup oddělování jednotlivých odstavců.

### 9.3 Speciální objekty

Mezi speciální objekty lze zařadit razítko, rámeček ohraničující text, ručně dopsané poznámky či jiné drobné obrázky. Tento blok je zachován v celku a není ořezáván. Jen ojediněle je jeho obsahem část textu (viz obrázek 9.3, jenž po svém obvodu obsahuje neúplný text).





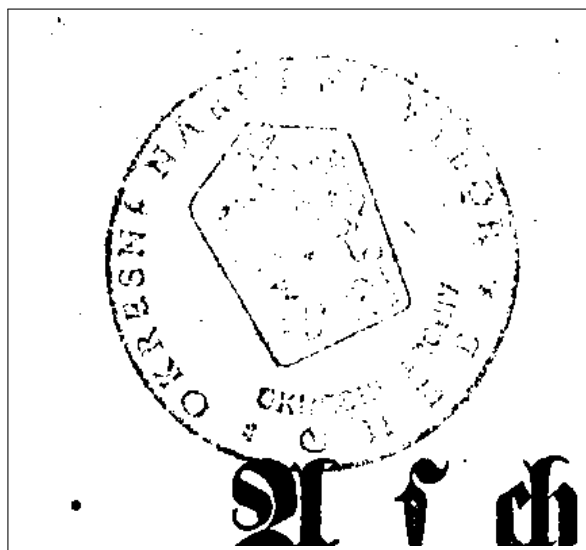
Obrázek 9.2: Ukázka sloupce

## 9.4 Proces určování typu oblasti

Pro další manipulaci s oříznutými oblastmi je nutné rozdělit každý blok získaný v předchozím kroku ze vstupního obrázku na blok obsahující sloupec nebo odstavec a na blok s nadpisem. Důvodem jsou pravidla podle kterých se řídí algoritmus pro hledání odstavců ve sloupcích nebo větších blocích textu. Pokud by jsme takovému algoritmu předložili blok s obsahující pouze nadpis, velmi pravděpodobně by část nesprávně oddělil.

1. Načteme libovolnou část vstupního obrázku vytvořenou v předchozím kroku (Prvotní detekce oblastí zájmu)
2. S pomocí open-source knihovny OpenCV zjistíme celkový počet znaků a jejich velikosti, které se v daném bloku nacházejí.
3. Vypočteme průměrnou výšku znaku.
4. <sup>1</sup>Pokud je průměrná výška alespoň 30 pixelů a celkový počet znaků roven alespoň 100, blok označím jako textový, v opačném případě jako nadpis.

<sup>1</sup>Hodnoty zde použité jsou objasněny v následující sekci



Obrázek 9.3: Ukázka speciálního objektu

## 9.5 Hodnoty rozhodující o typu oblasti

### 9.5.1 Velikost písma

Jedním z jednoduchých způsobů dle kterého lze říci zda je blok nadpisem nebo odstavcem je průměrná velikost znaků. Pro určení mezní hodnoty podle níž je možné určit typ bloku jsem provedl experiment. Spočívá v měření průměrných velikostí znaků u bloků, kde předem víme zda patří do skupiny odstavců či nadpisů.

### 9.5.2 Množství znaků

Podobný přístup měření jako byl použit u velikosti písma je možné využít i pro další parametr, který dopomůže určit typ bloku. Množství znaků v oblasti nadpisu je obvykle řádově menší než u odstavců.

### 9.5.3 Průnik výsledků

Pro označení bloku jako nadpis je postačující splnění alespoň jedné z podmínek. Může se stát, že velikost znaků nadpisu bude nižší než hranice určující typ, avšak druhá hodnota určující počet znaků bude nízká. Tabulka 9.1 obsahuje 22 testovací bloků u kterých byly zjištěny hodnoty počtu znaků a průměrné velikosti znaků. Tabulky 9.2 a 9.3 jsou vyhodnocovací, obsahují minimální, maximální a průměrné hodnoty pro odstavce a nadpisy dle kterých bylo po analýze rozhodnuta pro odhad typu bloku kombinace hodnot

| Index bloku | Typ      | Počet znaků | Průměrná velikost znaků |
|-------------|----------|-------------|-------------------------|
| 1           | Nadpis   | 22          | 45,6                    |
| 2           | Nadpis   | 4           | 21,5                    |
| 3           | Nadpis   | 34          | 17,2                    |
| 4           | Nadpis   | 44          | 20                      |
| 5           | Nadpis   | 14          | 22                      |
| 6           | Nadpis   | 29          | 21                      |
| 7           | Nadpis   | 9           | 14                      |
| 8           | Nadpis   | 19          | 21                      |
| 9           | Nadpis   | 29          | 31,5                    |
| 10          | Nadpis   | 21          | 68,9                    |
| 11          | Nadpis   | 17          | 19,1                    |
| 12          | Odstavec | 3106        | 17,6                    |
| 13          | Odstavec | 624         | 14,3                    |
| 14          | Odstavec | 539         | 17,9                    |
| 15          | Odstavec | 2174        | 17,75                   |
| 16          | Odstavec | 825         | 17,16                   |
| 17          | Odstavec | 646         | 15,5                    |
| 18          | Odstavec | 266         | 15,6                    |
| 19          | Odstavec | 170         | 16                      |
| 20          | Odstavec | 129         | 15                      |
| 21          | Odstavec | 442         | 16,3                    |
| 22          | Odstavec | 622         | 14,5                    |

Tabulka 9.1: Tabulka hodnot kdy pro každý blok bylo zjištěn počet znaků a průměrná velikost všech znaků

100 pro počet znaků a 30 pro velikost písma.

| Počet znaků | Min | Max  | Avg |
|-------------|-----|------|-----|
| Nadpis      | 4   | 44   | 22  |
| Odstavec    | 129 | 3106 | 867 |

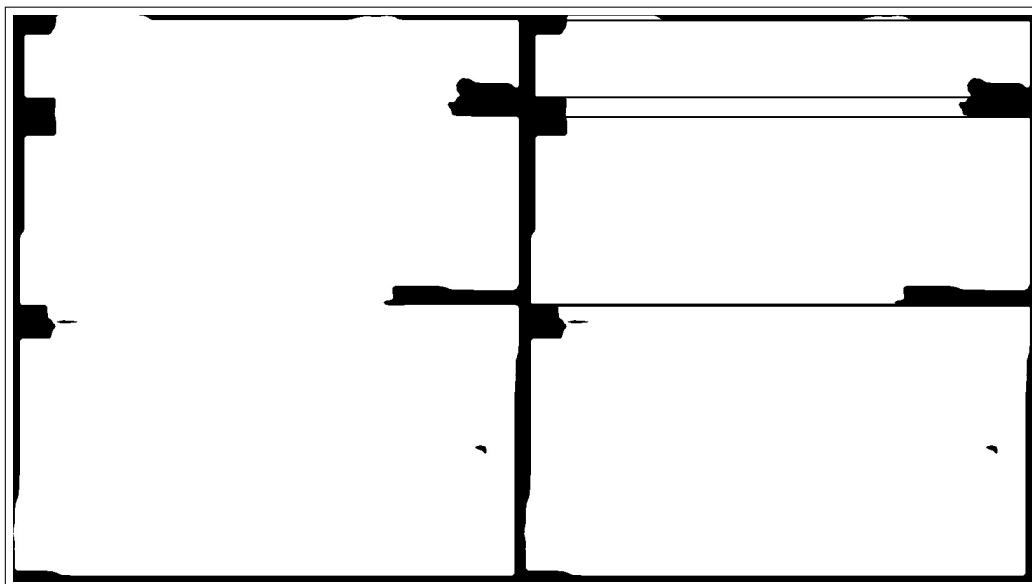
Tabulka 9.2: Vyhodnocovací tabulka pro počet znaků

| Velikost znaků | Min  | Max   | Avg  |
|----------------|------|-------|------|
| Nadpis         | 14   | 68,9  | 25,6 |
| Odstavec       | 14,3 | 17,75 | 16,1 |

Tabulka 9.3: Vyhodnocovací tabulka pro průměrnou velikost znaku

# 10 Detekce spojených odstavců

UNet, neuronová síť jež nám vytvoří predikovanou masku oblastí zájmu, nedokáže správně oddělit jednotlivé odstavce. A proto na bloky definované jako textový blok je nutné aplikovat algoritmus detekující dosud spojené odstavce a části, které musí být odděleny. Na bloku masky spojených odstavců je obvykle možné pozorovat odsazení řádku. Pokud se tato oblast nachází v pravé části sloupce představuje ukončení jednoho odstavce, pokud nalevo, odstavec zde začíná. Na obrázku 10.1 je možné tento jev pozorovat. Naším cílem je detekovat odsazení a oddělit odstavce od sebe.



Obrázek 10.1: Odstavce jsou detekovány s pomocí malých černých oblastí zleva i zprava

## 10.1 Algoritmus detekce a oddělení odstavců

1. Pokud je blok definován jako textový blok, zjistíme si jeho rozměry.
2. Pokud jsou oba z rozměrů bloku větší než jedna desetina rozměrů vstupního obrázku, uložíme si hodnotu pěti procent šířky bloku.
3. Jako první hledám odskočení zleva, procházím matici binárního obrázku po jednotlivých pixelech a pokud hodnoty prvních alespoň pět

procent řádky jsou rovny 0, k dočasné proměnné inicializované na nulu připočtu jedničku. Pěti procentní hodnota byla na základě analýzy zjištěna jako dostatečná vzdálenost pro vyfiltrování chyb predikce.

4. Proces opakuji do té doby než je odskočení ukončeno. Pokud v tento okamžik dočasná proměnná dosahuje hodnoty vyšší než 25, vrátím se na začátek odskočení a všechny hodnoty v celé řádce změním na 0, čímž dosáhnu oddělení odstavce.
5. Dočasnou proměnnou vynuluji a pokračuji dále dokud nenarazím na konec bloku, přičemž stále provádím kroky 3 a 4.
6. Stejný proces provedu na sloupec zprava kde tentokrát hledám ukončení odstavce. Důvodem dvojitého průchodu je situace kdy nový odstavec není správně detekován zprava či oblasti odskočení obsahuje šum zabráňující vytvoření správné predikce masky oblastí zájmu.

# 11 Finální výsledky

Segmentační proces z dat získaných během svého běhu vytvoří PAGE xml dokument, který bude obsahovat kompletní anotaci obsahu. Názvy bloků jednotlivých oblastí zájmu jsou seřazeny dle toho jak při čtení jdou za sebou. Pro závěrečné vyhodnocení správnosti procesu segmentace bude proveden experiment, kdy procesem zpracuji 8 testovacích Porta fontium stránek a výsledné PAGE porovná s ground-truth v aplikaci Layout Evaluation.

## 11.0.1 Porovnání výsledků na testovací sadě vůči diplomové práci

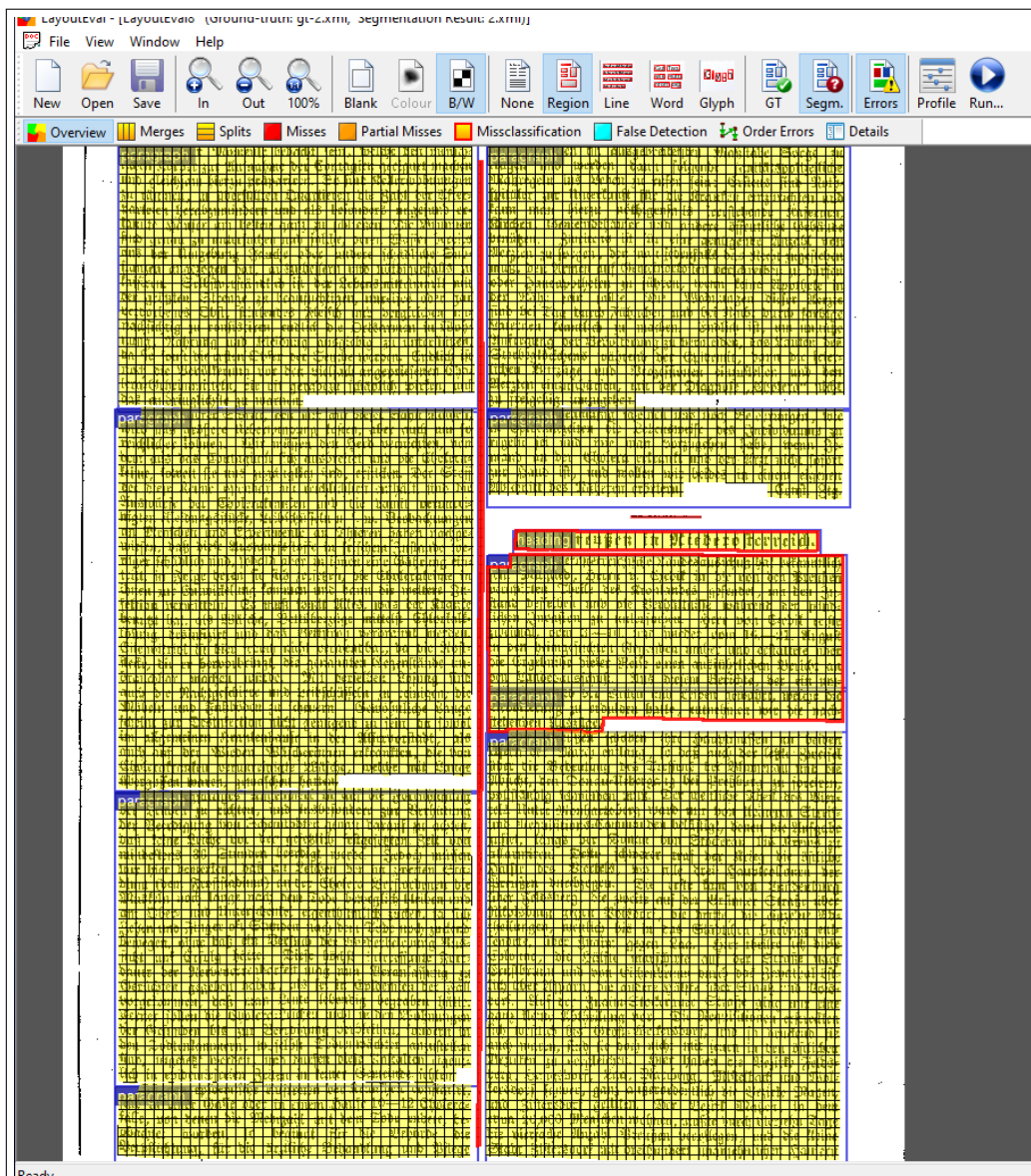
Je nutné provést měření kdy porovnáme výsledky segmentace tohoto procesu s výsledky segmentace diplomové práce Miroslava Lišky. Měření bylo provedeno na 8 testovacích stránkách obrazového dokumentu Porta fontium a výsledné PAGE xml vyhodnoceno aplikací Layout Evaluation, jenž nám poskytla procentuální správnost dané segmentace. Při hodnocení výsledků prosím uvažujte to, že ground-truth anotace je složena ze složitých útvarů, kdežto proces segmentace vytváří obdélníkové anotace. Je tedy samozřejmé že i přes správně vytvořené bloky, vygenerované PAGE bude mít u složitých stránek nižší procentuální správnost. Hodnoty získané během měření jsou sepsány do tabulky 11.1.

Dle získaných výsledků je zřejmé, že proces popsaný, vytvořený a implementovaný v této bakalářské práci dosahuje mnohem vyšší procentuální hodnoty správně oddělených a definovaných oblastí zájmu než výsledky původní segmentace. Rozdíl je více než 20 procentní, ale i přes to, že proces dosahuje průměrné správnosti téměř 90 procent, existují zde možnosti rozšíření, kde by bylo teoreticky možné dosáhnout ještě vyššího hodnocení. Obrázek 11.1 zobrazuje ukázkou grafického vyhodnocení anotace PAGE v aplikaci Layout Evaluation.

| Index stránky | Nová metoda [%] | Stará metoda [%] |
|---------------|-----------------|------------------|
| 1             | 77,95           | 68,93            |
| 2             | 97,63           | 68,63            |
| 3             | 79,06           | 71,25            |
| 4             | 93,69           | 64,42            |
| 5             | 91,20           | 67,05            |
| 6             | 82,94           | 64,67            |
| 7             | 86,39           | 61,56            |
| 8             | 94,56           | 64,38            |
| Avg           | 87,93           | 66,36            |

Tabulka 11.1: Porovnání výsledných hodnot





Obrázek 11.1: Layout Evaluation vyhodnocování PAGE anotace

## 12 Závěr

Cílem práce bylo implementovat metodu segmentace historických dokumentů a vylepšit metodu stávající, která dělí dokument podle separátorů. Na úvod byly prostudovány základní způsoby segmentace a teoretické poznatky o umělých neuronových sítích. Seznámili jsme se se způsoby učení sítí, co je to konvoluční neuronová síť, jak se data generují pokud máme nedostatek trénovacích dat pro jejich učení a uvedl jsem příklady plně propojených konvolučních sítí, které byly využity v této práci. Představení datasetů Europeana a Porta fontium byl krok následující společně s tím co znamená zkratka souboru PAGE a jeho využití v aplikaci Layout Evaluation.

Práce byla navázána na studii Miroslava Lišky ohledně typů plně konvolučních sítí a jejich využití v rámci segmentace. Zde se mi podařilo natrénovat model UNetu dosahující vysoké průměrné procentuální hodnoty 90 procent, který představoval základní kámen celého procesu. Bloky vstupního obrázku získané s pomocí AruNet sítě, byly dále zpracovány a rozděleny na bloky, které přesněji anotují oblasti zájmu. Následoval algoritmus pro detekci nadpisu a spojených odstavců. Pokud proces detekoval blok, kde je sloupec textu složený z několika odstavců, byl rozdělen a výsledné bloky znovu otestovány na typ oblasti. Nakonec informace získané během procesu zpracoval a vytvořil z nich anotační soubor PAGE formátu xml, který byl našim cílem. Metoda byla otestována na testovací sadě Porta fontium a porovnána s výsledky segmentace diplomové práce v aplikaci Layout Evaluation. Oproti předchozí metodě se průměrná procentuální hodnota správné segmentace zvýšila o více než 20 procent. Celý systém je připraven pro jednoduchou integraci do webové aplikace OCR Demo projektu Moderní zpřístupnění historických pramenů.

Jako možné rozšíření se nabízí využít jevu, objeveného při experimentu, jehož cílem bylo nalézt nejlepší model pro predikci masky oblastí zájmu. Tento jev vzniká pokud vstupní rozměr modelu je větší než jeho rozměr trénovací. Model se snaží nalézt odstavce, ale vlivem zvětšení rozlišení dochází k detekci jednotlivých slov. Pro rozsáhlejší experimenty bude potřeba vytvořit ground-truth s anotací slov.

# 13 Použité zkratky

**PAGE** - Page Analysis and Grount-truth Elements

**OCR** - Optical Character Recognition

**XML** - Extensible Markup Language

**GPU** - Graphics Processing Unit

**AMD** - Advanced Micro Devices

**CUDA** - Compute Unified Device Architecture

**GT** - Ground Truth

**PM** - Predicted Mask

# 14 Příručka

## 14.1 Instalace

Proces segmentace byl napsán v programovacím jazyce Python verze 3.7.7. Pro správné fungování je samozřejmostí Tensorflow verze 2.10 nebo alespoň verze 2.0. Keras verze se odvíjí od toho jakou verzi Tensorflow používáte, na stránce <https://docs.floydhub.com/guides/environments/> je seznam kompatibilních verzí. Mezi další důležité knihovny patří OpenCV-Python a Scikit-Image.

Nejjednodušším způsobem je příprava prostředí skrze velmi užitečnou utilitu Anaconda v.3<sup>1</sup>. Do přílohy jsem umístil již předpřipravené prostředí, které stačí nakopírovat do složky envs umístěné v hlavní anaconda3 složce.

**activate nazevprostredi** - aktivace prostředí ve Windows

**source activate nazevprostredi** - aktivace prostředí v Unix

Třetí možností je nainstalovat veškeré závislosti s pomocí textového souboru requirements.txt, jenž je umístěn v příloze.

**pip install -r requirements.txt**

## 14.2 Spuštění

Pokud máte správně připravené prostředí, jediný příkaz, který stačí spustit má tvar:

**python Run.py**

Skript má dva povinné argumenty, prvním je vstupní binarizovaný, nejlépe orotovaný obrázek. Druhý je cesta ke složce, kde se budou ukládat veškeré mezikroky včetně výsledného PAGE xml souboru.

---

<sup>1</sup><https://www.anaconda.com/>

# Literatura

- [1] ABADI, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Dostupné z: <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [2] CHOLLET, F. *Deep Learning v jazyku Python*. Grada, 2019. ISBN 978-80-271-2750-1.
- [3] GRÜNING TOBIAS, S. T. M. J. L. R. L. G. A two-stage method for text line detection in historical documents. *International Journal on Document Analysis and Recognition (IJ DAR)*. 2019, s. 285–302. ISSN 0001-0782. doi: 10.1007/s10032-019-00332-1. Dostupné z: <https://doi.org/10.1007/s10032-019-00332-1>.
- [4] LIŠKA, M. Segmentace historických obrazových dokumentů, 2019.
- [5] 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.
- [6] SUMMERFIELD, M. *Python 3*. Computer Press, 2013. ISBN 978-80-251-2737-7.
- [7] TICHÝ, J. Zpracování scanových dokumentů, 2015.
- [8] VOBECKÝ, A. Data augmentation for neural networks training, 2019.
- [9] ŽÁČEK, V. Kohonenova samoorganizační mapa, 2012.