Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Plzeň, 2020

Pavel Andrlík

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

Bakalářská práce

Generování obrazových dat pro účely trénování hlubokých neuronových sítí

Vedoucí: Ing. Marek Hrúz, Ph.D.

Plzeň, 2020

Vypracoval: Pavel Andrlík - A17B0538P

PROHLÁŠENÍ

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

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

V Plzni dne *3.7.2020* …….

vlastnoruční podpis

Poděkování

Chtěl bych poděkovat mému vedoucímu práce panu Ing. Marku Hrúzovi Ph.D. za možnost podílet se na projektu pro Českou televizi. Také děkuji za pomoc při vytváření a implementaci algoritmu a rád bych vyzdvihl jeho hluboké odborné znalosti v dané problematice. Dále bych chtěl poděkovat panu Ing. Martinu Bulínovi M.Sc., který mi pomáhal s drobnými, ale častými implementačními obtížemi.

Abstrakt

Cílem bakalářské práce je vytvoření a implementace algoritmu pro generování obrázků s texty, imitující texty zpravodajských relací. Tyto obrázky budou následně sloužit pro trénování umělých neuronových sítí pro rozpoznávání textů v obraze.

V první části práce jsou analyzovány zpravodajské relace, respektive je zjišťována struktura a rozložení obrazových dat. Dále jsou analyzovány používané texty a znaky včetně používaných fontů. Zjišťováno je také, jaká se nejčastěji objevují slova, speciální nebo v běžném jazyce neobvyklé znaky a symboly a kombinace těchto neobvyklých znaků s používanými slovy nebo druhy textů, jako jsou například jména, místa, povolání, politické strany a podobně.

V druhé části se zabývám konkrétním návrhem algoritmu, jeho zobecněním pro možnost použití u různých stanic provozujících zpravodajské relace a v poslední řadě jeho implementací v konkrétním programovacím jazyce.

V poslední části je experimentálně ověřena kvalita dosažených výsledků. Jaccardův index (také IoU) rozpoznaných oblastí s textem s použitím stávající natrénované sítě je průměrně 0,7. Dále jsou tyto výsledky diskutovány a podrobněji rozebrány. Úplně na závěr je diskutováno navrhnutí možných vylepšení a budoucí práce navázané na tuto práci.

Klíčová slova

generování dat, syntetická data, rozpoznávání textu, zpravodajské relace, neuronové sítě, obrazová data

Abstract

The purpose of this bachelor thesis is to create and implement an algorithm for generating images with texts imitating the texts of news sessions. These images will be used to train artificial neural networks for text-in-image recognition.

In the first part of the work, the news sessions are analyzed, more precisely the structure and distribution of image data is discovered. In addition to, the used texts and characters, including the used fonts, are analyzed. Ascertained are also the most common words, special or unusual characters and symbols in common language, and combinations of these unusual characters with words or types of text used, such as names, places, professions, political parties, and so on.

The second part is focused on a specific design of the algorithm, its generalization for the possibility of using for various news sessions and finally its implementation in a specific programming language.

In the last part, the quality of the achieved results is experimentally verified. The Jaccard index (also IoU) of recognized areas with text using the existing trained network is on average 0.7. These results are then discussed and examined. Finally, the proposal of possible improvements and future work related to this work is discussed.

Key words

data generation, synthetic data, text recognition, news sessions, neural network, image data

Obsah

Kapitola 1

Úvod

Tato bakalářská práce se zabývá vytvářením trénovacích obrazových dat pro hluboké konvoluční neuronové sítě používané na Západočeské univerzitě v Plzni k rozpoznávání textu ve zpravodajských relacích.

1.1 Motivace

Hlavní motivací pro tuto práci je potřeba vytváření trénovacích obrazových dat, které by umožnily zvýšit úspěšnost rozpoznávání hlubokých konvolučních neuronových sítí pro rozpoznávání textu. Dosáhnout vyšší úspěšnosti rozpoznávání by se mělo pomocí specifických pozic, velikostí, podbarvení, rozložení a typů textů v trénovacích datech. Zpravodajské relace mají svá specifika, kvůli kterým nedosahují běžně trénované hluboké konvoluční neuronové sítě tak vysoké úspěšnosti. Jedná se zejména o kombinace znaků a textů, které se v běžné řeči a běžných textech vyskytují velice zřídka nebo vůbec.

Cílem této bakalářské práce je zlepšit stávající úspěšnost používané sítě prostřednictvím generování obrazových dat z cílové domény pro její natrénování. Stávající neuronová síť je trénována na příliš obecných datech, a tudíž dovede rozpoznávat loga, některé ručně psané texty i běžné fonty, kdy právě zmíněná schopnost rozpoznávat i obecné texty brání bezchybnému rozpoznávání textů. Cílem je tedy zlepšit úspěšnost rozpoznávání specifických obrazových textových dat, a tak síť adaptovat.

Činnosti spojené s realizací bakalářské práce byly rozděleny do několika oblastí, kterými byly (i) seznámení se s principy rozpoznávání textů v obraze pomocí neuronových sítí, (ii) nastudování moderních technik generování textů v obraze, (iii) návrh a vytvoření algoritmu pro generování obrazových dat imitujících zpravodajské relace, (iv) experimentální ověření kvality.

1.2 Použité technologie

V této kapitole jsou popsány softwarové technologie použité v bakalářské práci. Veškeré použité technologie jsou typu Open-Source.

1.2.1 Python

Python [1] je vysokoúrovňový dynamický interpretovaný jazyk, který byl navržen Guidem van Rossumem. Tento jazyk získává v posledních letech na popularitě, protože je Open-Source, nabízí velkou řadu knihoven, dá se dobře a rychle naučit a dobře se v něm orientuje. V roce 2018 se díky své zvyšující se popularitě zařadil mezi nejpopulárnější jazyky na světě. Nabízí zároveň instalační balíčky pro většinu používaných platforem (Windows, Linux, macOS, Unix, Android). Aktuálně nejnovější verzí je Python 3.8. Standardní implementace Pythonu je v jazyce C a je vyvíjená Python Software Foundation. Zároveň existují i další implementace jazyka pro specifické potřeby, jako např. PyPy pro masivní paralelní programování.

Nabízí možnost programovat objektově, procedurálně, funkcionálně i imperativně. Kvůli interpretovanému kódu dosahuje Python často nevyrovnaných výkonů. Pokud se kód spouští s implementací PyPy nebo Cython, je možné optimalizací kódu dosáhnout i více než řádového zlepšení a při maximální optimalizaci kódu na výkon lze dosáhnout téměř stejné rychlosti jako u kódu psaného přímo v jazyce C.

Důvody, proč byl pro implementaci zvolen Python, jsou následující:

- velké množství Open-Source knihoven, které ušetří hodně času při práci s obrazovými daty
- snadná implementace kódu a absence starostí jako například správa paměti a dalších nízkoúrovňových starostí
- přenositelnost kódu mezi zařízeními
- oproti Javě, což je asi největší konkurence, nepotřebuje Python vfVM

1.2.1.1 PIL - Python Imaging Library

Python Imaging Library (zkráceně PIL) [2] [3], v novějších verzích známý také jako Pillow, je volná knihovna pro Python, která nabízí řadu funkcí pro práci s obrazovými daty. Byla vydaná v roce 2009 a umožňuje

- práci s jednotlivými pixely
- masking a zprůhledňování
- filtrování obrazu, jako vyhlazování, rozmazání nebo hledání hran
- vylepšení obrazu, jako je zaostření, úprava jasu, kontrastu nebo barvy
- přidání textu do obrázku a mnoho dalšího

Právě poslední funkce budu pro svoji práci využívat především. Pillow knihovna umožňuje pro přidání textu do obrazu zvolit mnohé, jako například font, velikost, průhlednost, barvu, stínování, natočení textu, mezery, směr, zarovnání, jazyk, a mnoho dalších. V mé práci využívám následujících možností

- barva
- výplň
- font
- zarovnání.

Pillow disponuje také funkcí na vrácení velikosti textu v pixelech, kdy vstupními parametry, které využívám, jsou font a text.

Jako podklad na pozadí pro text využívám funkci na vkládání obdélníků a funkci na skládání obrazů přes sebe pro dosažení průhlednosti požadovaných vkládaných objektů.

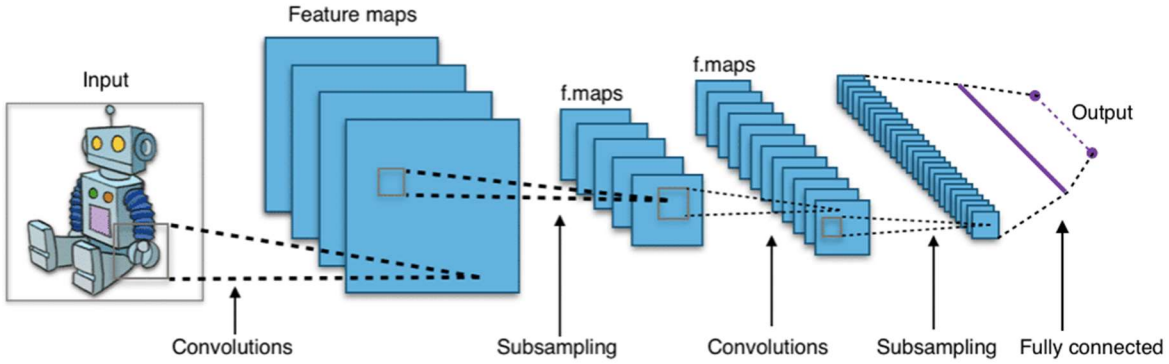
Kapitola 2

Principy rozpoznávání textů v obraze pomocí neuronových sítí

Rozpoznávání textů v obraze pomocí neuronových sítí je realizováno především konvolučními neuronovými sítěmi. Alternativně mohou být doplněny o předzpracování a o zpracování výstupu.

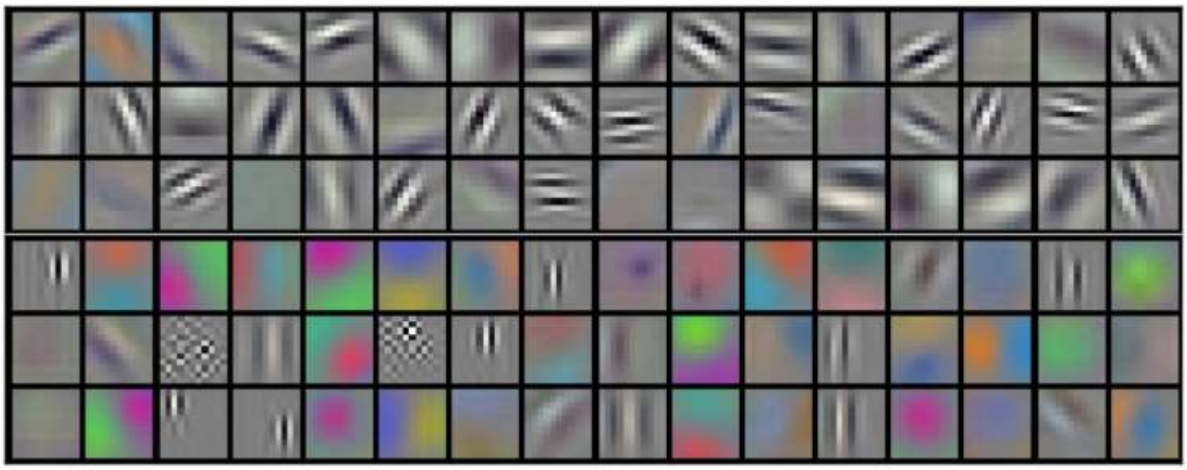
2.1 Konvoluční neuronové sítě

Konvoluční neuronová síť je speciálním případem neuronových sítí [4]. Neuronové sítě jsou klasifikátory, které je možné natrénovat na různorodé specifické úlohy analýzy dat. To zahrnuje zpracování dat, rozpoznávání objektů v obraze, shlukování dat, rozpoznávání řeči a mnoho dalších.



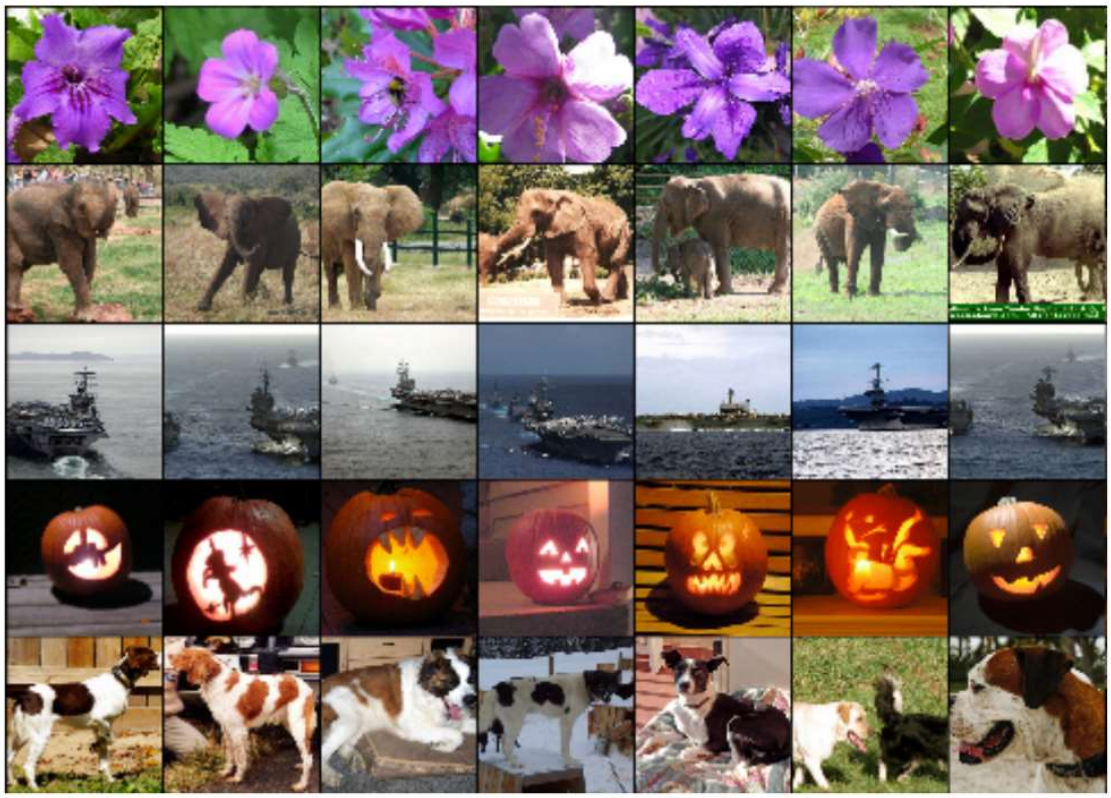
Obrázek 1- Ukázka zjednodušené funkce konvolučních neuronových sítí, kdy se pomocí konvoluce postupně získávají příznakové mapy, až dokud se nezíská pouze příznakový vektor. [5]

Konvoluce obrazu spočívá v zobrazení skupiny informací do jednoho prvku pomocí přenásobení konvoluční maskou. Tato operace se nazývá „pooling“. Výsledek operace konvoluce se nazývá “Feature map”, neboli příznaková mapa a můžeme na ni znovu aplikovat konvoluci. Poté, co obrázek projde všemi vrstvami konvoluční neuronové sítě, se na poslední vrstvu, jež je vektor, aplikuje poslední pooling a zjistí se nejpravděpodobnější rozpoznaný znak.



Obrázek 2 – První vrstva konvoluční neuronové sítě trénované na datasetu ImageNet [6] zobrazující barevné a tvarové filtry, které kooperují s funkcí tyčinek v lidském oku. [6]

Jednou z průlomových prací v oblasti rozpoznávání obrázků pomocí konvolučních neuronových sítí byla práce ImageNet Classification with Deep Convolutional Neural Networks [6], která vysvětlila význam první konvoluční vrstvy, jakožto barevných a tvarových filtrů. Naučené filtry, které se vytvořily pomocí trénování s učitelem pouze z dat, jsou podobné, jako tyčinky v oku člověka. Zajišťují prvotní zpracování, nalezení příznaků důležitých pro další zpracování, resp. rozpoznávání v konvolučních vrstvách a mohou nám říci, jaké prvky jsou v datech klíčové pro identifikaci, resp. rozpoznávání. Toto zjištění mimo jiné koresponduje s teorií evoluce.



Obrázek 3 – Ukázka podobnosti obrázků díky L2 vzdálenosti příznakových vektorů. Levý první sloupeček jsou obrázky, k nimž hledáme nejpodobnější na základě L2 vzdálenosti. Obrázky jsou seřazeny směrem vpravo, jak se zmenšuje jejich podobnost vůči vzoru v levém sloupečku. [6]

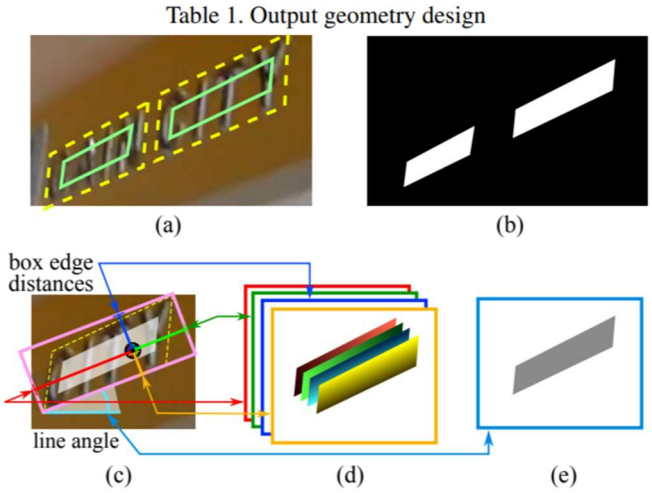
Obrázek 3 ukazuje, že díky blízké L2 vzdálenosti příznakových vektorů podobných obrázků ve vektorovém prostoru tvořeném příznakovými vektory, je možné velmi dobře určovat podobnost obrázků. Podobnost se neurčuje pouze pomocí barevných a tvarových statistik podobně jako dříve, ale právě pomocí L2 vzdálenosti příznakových vektorů, které obsahují extrahované klíčové sémantické příznaky.

2.2 End-to-end metoda pro vícejazyčné zpracování textu ve scéně

V této bakalářské práci je využita metoda [7] založená na konvolučních neuronových sítích. V dalších podkapitolách bude tato metoda představena.

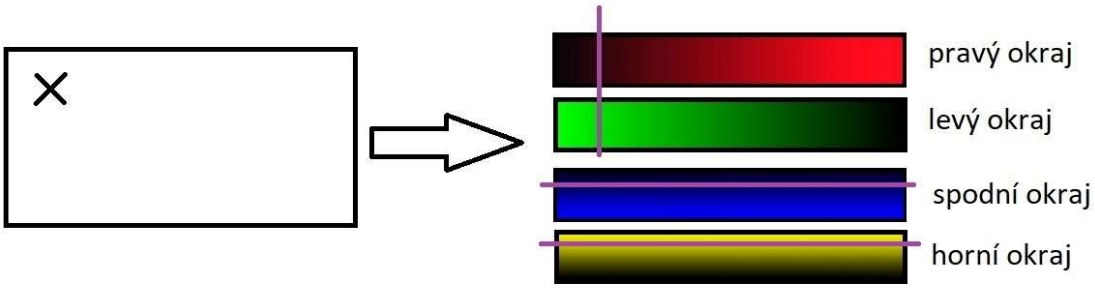
2.2.1 Lokalizace textu ve scéně

Lokalizace textu v obraze se provádí také konvoluční neuronovou sítí. Ta z obrazu text přímo nečte, pouze detekuje čtyřúhelník, ve kterém se text nachází.



Obrázek 4 – Ukázka lokalizace textu ve scéně v práci EAST [8]. Části: (a) žlutě je vyznačena původní oblast s textem, zeleně zmenšená oblast s textem; (b) ohodnocená oblast s textem; (c) generování RBOX mapy; (d) mapa vzdáleností jednotlivých pixelů ke stranám čtyřúhelníku; (e) úhel natočení. [8]

Obrázek 4 ukazuje lokalizaci textu ve scéně pomocí EAST [8] detektoru. Pro detekovanou oblast s textem viz. Obrázek 4 (a) se nejprve vytvoří tzv. "RBOX map". Ta obsahuje čtyři mapy vzdáleností. Každá mapa reprezentuje vzdálenost k jednotlivým stranám čtyřúhelníku. Vzdálenost bodu od strany, je reprezentována barvou, která obsahuje informaci o vzdálenosti bodu ke straně. Na základě těchto map se zredukuje velikost oblasti s textem (Obrázek 4 (a) zelená oblast), aby se v oblasti nacházely pouze užitečné informace. K boxu se přiřadí i informace úhlu pro zpracování při pozdějším čtení.



Obrázek 5 - Podrobnější vysvětlení, jak se poloha bodu v prostoru reprezentuje pomocí čtyř barevných map. Fialové přímky znázorňují, jak je bod v prostoru reprezentován v daných vzdálenostních mapách.

2.2.2 Čtení textu

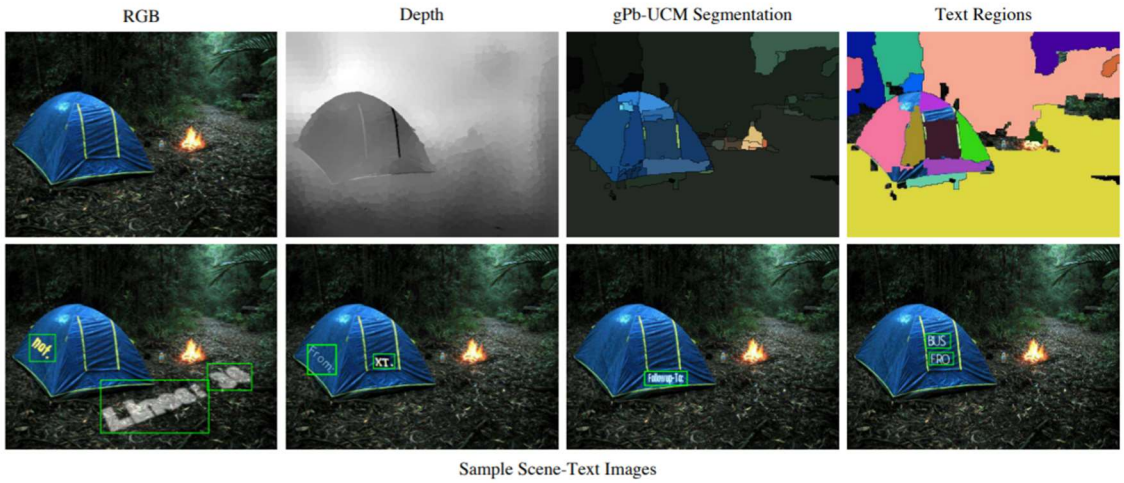
Čtení textu je realizováno plně konvoluční neuronovou sítí (tj. bez "fully connected" vrstev), což umožňuje zpracování libovolně dlouhého vstupu. Konvolucí tvoříme příznakové mapy až do doby, než z nich vytvoříme příznakový vektor. Během tohoto procesu, kdy přes obraz "klouže" konvoluční jádro, mohou vzniknout duplicity, pokud se znak objeví ve dvou nebo více bounding boxech. Tyto duplicity se mohou dostat až do příznakového vektoru, ze kterého se získávají znaky a ve výstupu se tyto duplicity mohou

objevit např. ve formě “AAUTTO”. Parametry sítě jsou optimalizovány pomocí ztrátové funkce “CTC loss function” [9], která odstraní duplicity a mezery nahradí znakem “_”.

Kapitola 3

Moderní techniky generování textů v obraze

3.1 Trénování syntetickými daty



Obrázek 6 - Postup vytvoření syntetických neuspořádaných textů ve scéně. První řádek ukazuje postup zpracování vstupního obrázku, kdy se nejprve provede hloubková analýza pomocí konvoluční sítě [10], následně se provede gPb-UCM segmentace [11] a nakonec obrázek rozdělí jednotlivé oblasti, do kterých je možné generovat text. [12]

Inspirací bakalářské práce byl generátor syntetických dat [12], který vkládá texty do homogenních částí scény. Na výchozí RGB obrázek se aplikuje hloubková analýza realizovaná konvoluční neuronovou sítí [10] a vytvoří se hloubková mapa. Na základě té se obraz segmentuje pomocí "gPb-UCM Segmentation" [11] a výsledkem je původní obrázek rozdělený na segmenty, které eliminují potřebu generovat texty na podkladu.

Texty jsou na homogenních plochách dobře rozpoznatelné a většinou vcelku věrně imitují reálné nápisy.

Pro potřeby rozpoznávání zpravodajských dat je ovšem takovéto generování textu nevhodné, protože zpravodajská data mají stejné typy písma a pro stejné kontexty se vždy vyskytují na stejných pozicích a na jednobarevném mírně průhledném podkladu. Tyto rozdíly vedou k chybám ve čtení textu, protože byla síť trénovaná na nevhodných, respektive příliš obecných datech, která nereflektují vlastnosti obrazových dat zpravodajských relací.

Kapitola 4

Algoritmus pro generování obrazových dat imitující zpravodajské relace

4.1 Analýza zpravodajských relací

Na úvod nutno podotknout, že tento projekt je vytvářen ve spolupráci s ČT, proto byla i analýza primárně prováděna na datech od ČT. Analýza zpravodajských relací se sestává z několika částí, jimiž jsou (i) šablony zpravodajských relací, (ii) analýza pozic grafických prvků, (iii) analýza zobrazovaných textů, (iv) analýza četností jednotlivých šablon. Závěrem je porovnání s relacemi ostatních televizních stanic.

Šablony zpravodajských relací

Následujících 5 obrázků (Obrázek 7 až Obrázek 11) zobrazuje šablony, které jsou ve zpravodajství ČT vždy zobrazovány stejně. U některých jako je šablona vizitka a šablona titulky, jsou některé grafické prvky shodné, v tomto konkrétním případě je to jméno a povolání.

Obrázek 7 - Šablona 1 - "vizitka"

Obrázek 8 - Šablona 2 - "živě"

Obrázek 9 - Šablona 3 - "titulek"

Obrázek 10 - Šablona 4 - "zdroj"

Obrázek 11 - Šablona 5 - "titulky"

Obrázek 12 ukazuje příklady ostatních nestálých obrazových dat s texty. Tato data se vymykají veškerým šablonám a není je možné žádným způsobem předpovídat. Veškerá nestálá obrazová data jsou vytvářena na míru pro jednotlivé zprávy ve zpravodajství. V datech nebyly nalezeny žádné podobnosti, podle kterých by se data daly rozumně sjednoceně generovat.

Obrázek 12 - Ostatní nestálá obrazová data

4.1.1 Analýza pozic grafických prvků

V následující tabulce jsou uvedeny relativní vzdálenosti klíčových bodů jednotlivých šablon. Každý obdélník je definovaný standardně dvěma body, levý horní roh a pravý dolní roh. Tato reprezentace obdélníků byla zvolena, protože je používána v knihovně Pillow (PIL) [3]. Pozice jsou zadané v rozmezí 0-1. Pozice v pixelech se vypočte **pozice * rozlišení osy**.

Šablona 1 - vizitka	levý horní roh		pravý dolní roh	
	x0	y0	x1	y1
první řádka	0.075	0.840277	0.9234375	0.90277777
čas	0.075	0.904	0.1578	0.95
text zpráv	0.15859375	0.90416666	0.9234375	0.95

Tabulka 1 - Šablona 1 - "vizitka"

Šablona 2 - živě	levý horní roh		pravý dolní roh	
	x0	y0	x1	y1
první řádka	0.240625	0.052777	dopočítávané	dopočítávané

Tabulka 2 - Šablona 1 - "živě"

Šablona 3 - titulek	levý horní roh		pravý dolní roh	
	x0	y0	x1	y1
první řádka	0.075	0.8125	0.9234375	0.9027777

Tabulka 3 - Šablona 1 - "titulek"

Šablona 4 - zdroj	levý horní roh		pravý dolní roh	
	x0	y0	x1	y1
první řádka	dopočítávané	dopočítávané	0.92109375	0.10694444

Tabulka 4 - Šablona 1 - "zdroj"

Šablona 5 - titulky	levý horní roh		pravý dolní roh	
	x0	y0	x1	y1
první řádka	0.075	dopočítávané	0.9234375	0.8222222
čas	0.075	0.904	0.1578	0.95

Tabulka 5 - Šablona 1 - "titulky"

4.1.2 Analýza zobrazovaných textů

V textech zobrazovaných během zpravodajství se mimo běžných textů objevují specifické znaky (“/”, “°C”, “%”, ...) a specifické druhy textů (jména, místa, politické strany, ...). Objevují se také čísla, ať už celá nebo desetinná, oddělená desetinnou čárkou, za kterými mohou následovat zkratky ‘mil.’, ‘mld.’ a další. Dále jako poslední se objevují čísla oddělená dvojtečkou, reprezentující čas, skóre, poměr a další.

Právě s těmito specifickými situacemi by neuronová síť natrénovaná na obecných datech mohla mít potíže z důvodu záměny s jinými texty nebo znaky.

- Seznam specifických znaků:
  - /, %, :, °C, (,), +

- Seznam zkratek:
    - m, mm, cm, km, mil., mld., km/h, m/s, r.
    - světové měny
    - zkratky politických stran
- Seznam specifických textů:
    - jména, příjmení, adresy, internetové adresy, firmy, politické strany, státy

4.1.3 Četnosti

Provést analýzu četnosti, jejíž kvantitativní zjištění by byla objektivní, by bylo časově velmi náročné a pro potřeby vytvoření algoritmu i zbytečné. U neuronových sítí při trénování z tzv. “big data” nezáleží na přesném poměru trénovacích příkladů, ale spíše na jejich velkém počtu. Úplně tedy postačí kvantifikace typu “časté”, “standardní”, “méně časté” a “výjimečné”.

Častými jsou:

- jména, příjmení, “:” v podobě zobrazovaného času

Standardními jsou:

- adresy, internetové adresy, firmy, politické strany, “/”, měny

Méně častými jsou:

- %, °C, (,), +, zkratky (m, km, mil., mld., r.), státy

Výjimečné jsou:

- zkratky (km/h, m/s, cm, mm a další)

4.1.4 Porovnání s relacemi ostatních televizních stanic

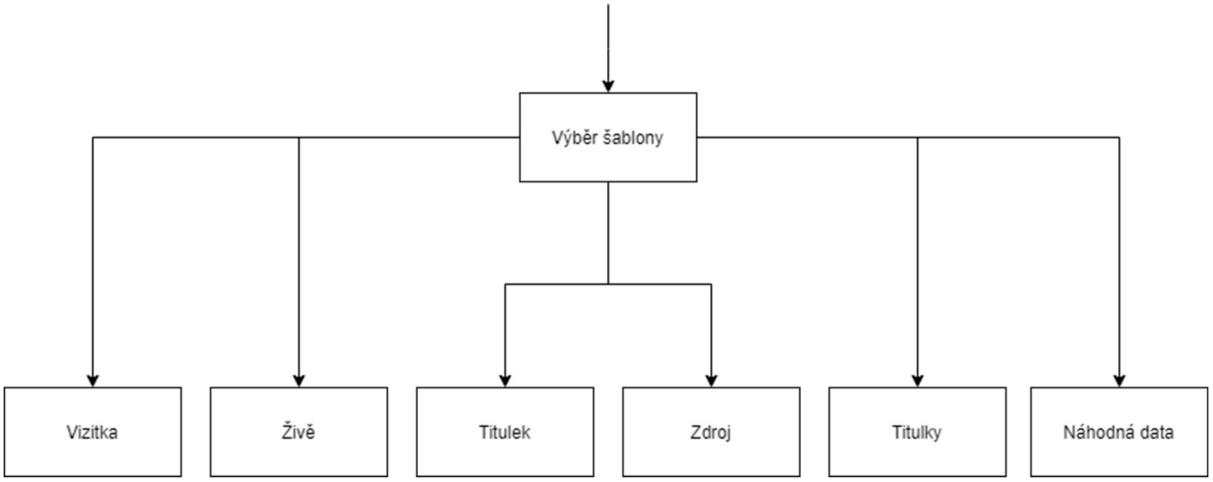
V porovnání s ostatními televizními relacemi vykazují tyto šablony podobnosti. Těmi jsou šablona živě, šablona zdroj, šablona titulky a z části šablona vizitka. U šablony titulek lze také najít podobnost.

4.2 Návrh algoritmu

Návrh algoritmu postupuje dle analýzy a zjištěných šablon.

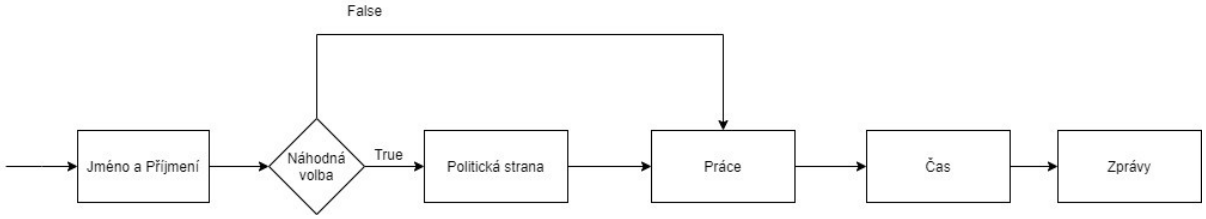
Hlavní kostra algoritmu

Výběr šablony:



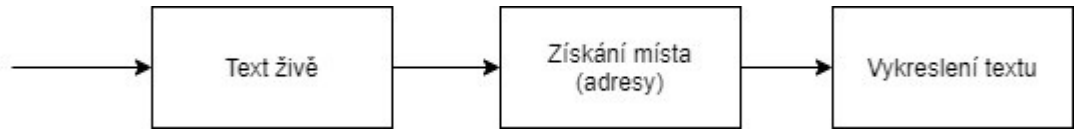
Obrázek 13 - Výběr šablony

Vizitka:



Obrázek 14 - Část pro vytvoření "vizitky"

Živě:



Obrázek 15 - Část pro vytvoření "živě"

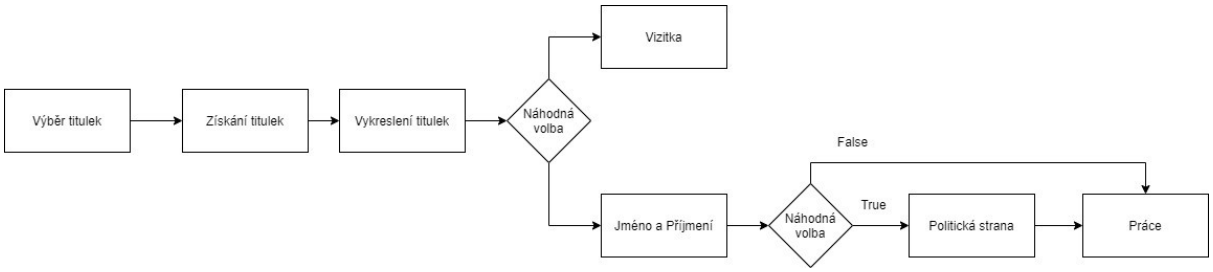
Titulek:

Obrázek 16 - Část pro vytvoření "titulku"

Zdroj:

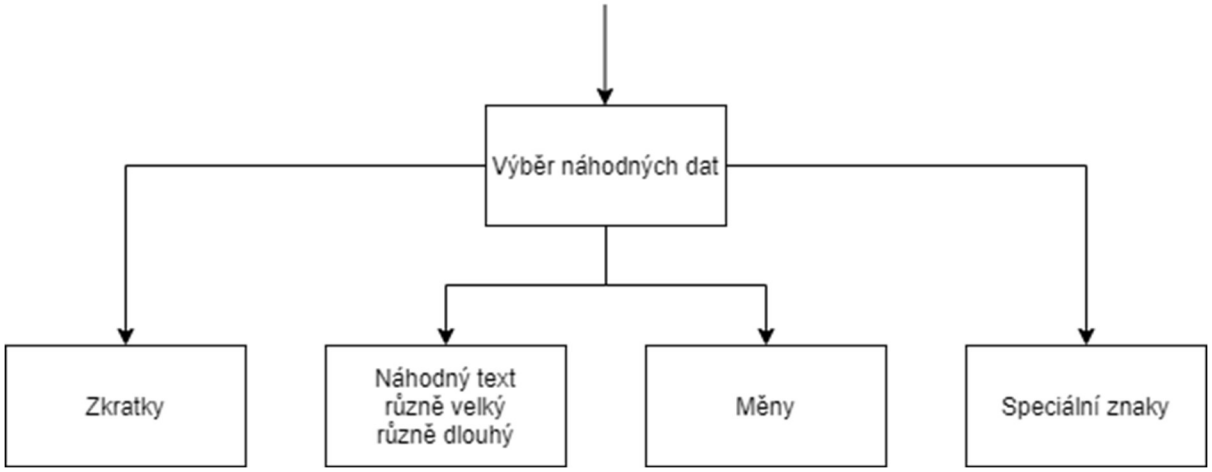
Obrázek 17 - Část pro vytvoření "zdroje"

Titulky:



Obrázek 18 - Část pro vytvoření "titulek"

Náhodná data:



Obrázek 19 - Část pro vytvoření náhodných dat

Anotace:

Obrázek 20 – Vytvoření anotace

4.3 Implementace algoritmu

Implementace algoritmu byla provedena, jak bylo zmíněno v kapitole použité technologie, pomocí programovacího jazyka Python.

Kód je strukturován do dvou tříd a následně do jednotlivých metod. Je zamýšlen jako kód pro “odbornou veřejnost”, z čehož plyne i existence vlastního konfiguračního souboru, který umožní uživateli téměř úplné nastavení pozic grafických prvků. Díky logickému a systematickému členění do jednotlivých funkcí jsou možné i zásahy do programu a jeho případné úpravy pro velmi odlišné zpravodajské relace.

Dvě hlavní třídy jsou

1. Sample
2. AITGM

Sample

Třída Sample má dva hlavní úkoly. Uchovat a separovat data aktuálně generovaného obrázku. Instance třídy Sample díky tomu uchovává veškerá data k obrázku a z toho důvodu je v ní umístěna i funkce pro vytvoření anotace.

Metody třídy se stručným popisem:

- load_image

  Načte do paměti pozadí, do kterého se budou vkládat grafické prvky.

- annotation

  Po dobu vytváření obrazového data uchovává ve strukturované formě pozice a texty vložených grafických prvků.

- annotation_save

  Vygeneruje uložená strukturovaná data do souboru typu “txt”.

AITGM

Druhá třída AITGM (Artificial intelligence training generator module) vytváří instance třídy Sample. Jedna instance třídy Sample pro jeden obrázek. Třída AITGM ke své funkci potřebuje konfigurační soubor, ze kterého jsou načítány veškerá data o umístění souborů s texty, pozice grafických prvků, rozlišení obrazových dat, barvy textů a pozadí a další doplňkové informace nutné k běhu programu.

Dále tato třída do jednotlivých postupně generovaných instancí třídy Sample vloží pozadí a jednotlivé grafické prvky podle logiky v kapitole 4.2 Návrh algoritmu.

Funkce třídy se stručným popisem:

- load_vars

  Načte proměnné z konfiguračního souboru.

- load_cfg

  Načte konfigurační soubor.

- new_sample

  Založí novou instanci třídy Sample a náhodně vybere šablonu. Poté vygeneruje anotační soubor.

- get_one

  Vrátí náhodnou řádku ze souboru v potřebném formátu.

- get_text

  Vrátí náhodně dlouhý text ze zadaného rozmezí ze souboru.

- load_file

  Načte soubor a předupraví pro další funkce.

- get_wild_text

  Vrátí náhodný text ze zadaného rozmezí pro generování náhodného textu.

- get_date

  Vrátí náhodné datum.

- get_politic

  Vrátí náhodnou politickou stranu.

- getplace

  Vrátí náhodnou kombinaci jednoho nebo více prvků míst v tomto pořadí: stát, město, ulice.

- live

  Vygeneruje do obrázku data dle šablony živě.

- subtitles

  Vygeneruje do obrázku titulky.

- source

  Vygeneruje do obrázku data dle šablony zdroj.

- report_title

  Vygeneruje do obrázku data dle šablony titulek.

- name_job

  Vygeneruje do obrázku jméno, příjmení, potencionálně politickou stranu a práci dle návrhu algoritmu. Část šablony vizitka.

- crawl_text

  Vygeneruje do obrázku zprávy. Část šablony vizitka.

- resized_ct

  Vygeneruje do obrázku velikostně změněné zprávy. Část šablony vizitka.

- time_generator

  Vygeneruje do obrázku čas. Část šablony vizitka.

- synt_wild

  Vygeneruje do obrázku náhodná data, pokud je to možné (je dostatek místa, nekolidují grafické prvky, ...). Vrátí True, když funkce skončila úspěšně a text byl vygenerován, nebo False, když se nasčítal dostatek chyb signalizujících problém (zacyklení, kolize, …).

- wild_text

  Kontroluje, jestli se provedla funkce synt_wild a případně ji pouští znovu, dokud se neprovede.

- get_text_color

  Vrátí barvu textu ze zvolených barev v konfiguračním souboru.

- get_back_color

  Vrátí barvu pozadí ze zvolených barev v konfiguračním souboru.

- check_colors

  Zkontroluje, jestli nejsou barvy vložené jako parametry dostatečně odlišné.

Kapitola 5

Dosažené výsledky

Obrázek 21 – Ukázka většiny společně vygenerovaných syntetických dat

Obrázek 21 ukazuje společně vykreslené téměř všechny prvky kromě titulku, který by se kryl s vizitkou. Uprostřed se nachází oblast s náhodným textem, která má za úkol simulovat nestálá obrazová data vyskytující se ve zpravodajství, viz. Kapitola 4.1 – „Analýza zpravodajských relací“.

Obrázek 22 - Porovnání originálu a vytvořené kopie pro titulek zprávy

Obrázek 23 - Porovnání originálu a vytvořené kopie pro titulky

Obrázek 24 - Porovnání originálu a vytvořené kopie pro vizitku

Na porovnáních můžeme vidět věrnost vygenerovaných prvků vzhledem k originálu. ČT poskytla unikátní privátní font „TV Sans Screen“, používaný výhradně v pořadech z produkce ČT, díky čemuž jsou kopie zcela věrné originálu.

Kapitola 6

Experimentální ověření kvality a budoucí práce

6.1 Rozpoznané oblasti s textem

Pro rozpoznání oblastí s textem byl použit E2E-MLT [7] detektor. Na obrázcích Obrázek 25 až Obrázek 27 lze vidět, že rozpoznané oblasti se liší minimálně a v rozpoznaném textu lze spatřit drobné rozdíly, ale i naprosté shody. Shodu vidíme například na Obrázek 25, kde na originálu bylo rozpoznáno „www.cts4.cz“ a na vytvořené kopii také „www.cts4.cz“, ačkoli mělo být rozpoznáno „www.ct24.cz“. Rozdíl je naopak u slova „případů“, kde na originále je rozpoznáno „h pripadù“ a na kopii „hpripadüz“. Metoda na rozpoznání [7] je parametrická a tyto chyby jsou jedním z důsledků nastavení parametrů, a i důsledkem trénování, které neobsahovalo český jazyk.

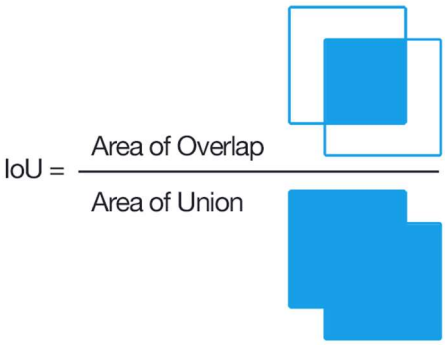
Obrázek 25 - Porovnání rozpoznaných oblastí u titulku

Obrázek 26 - Porovnání rozpoznaných oblastí u titulků

Obrázek 27- Porovnání rozpoznaných oblastí u vizitky

6.2 Jaccard index

Experimentální ověření kvality je provedeno pomocí Jaccardova indexu. Porovnává se průnik vygenerované oblasti s textem a rozpoznané oblasti s textem se sjednocením těchto oblastí. Jaccardův index se spočte jako průnik oblastí vydělený jejich sjednocením.



Obrázek 28 – Výpočet Intersetion over Unions (Jaccardův index) [13]

Pro vygenerované obrázky uvedené v příloze jsem vytvořil kód, který vypočte Jaccardův index oblastí zapsaných při generování textů a oblastí rozpoznaných pomocí E2E-MLT [7] detektoru. Statistika přesnosti rozpoznávání textu bude spočtena v navazující práci.

Výsledky pro obrázky v příloze:

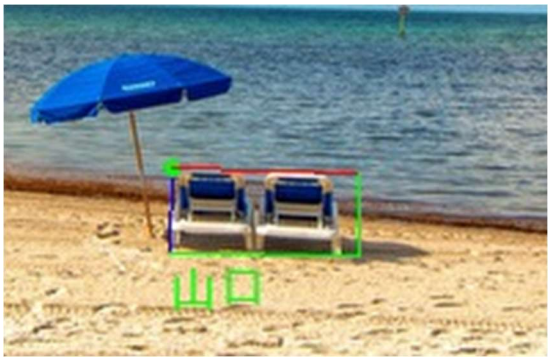
Název	Sjednocení [px]	Průnik [px]	Jaccardův index
img1	2780	1905	0.6852
img2	7446	5390	0.7238
img3	6823	5018	0.7354
img4	3780	3105	0.8214
img5	40120	31953	0.7964
img6	33707	28055	0.8323
img7	9668	6260	0.6474
img8	25303	19985	0.7898
img9	7725	5942	0.7691
img10	5715	3875	0.6780
img11	14184	10801	0.7614

img12	24928	20998	0.8423
img13	6453	4878	0.7559
img14	5953	4584	0.7700
img15	31295	25952	0.8292
img16	8829	4589	0.5197
img17	47695	38101	0.7988
img18	6070	4370	0.7199
img19	20447	16114	0.7880
img20	4906	3772	0.7688
img21	30399	25339	0.8335
img22	3848	3461	0.8994
img23	9340	5284	0.5657
img24	14367	10934	0.7610
img25	16728	12658	0.7566

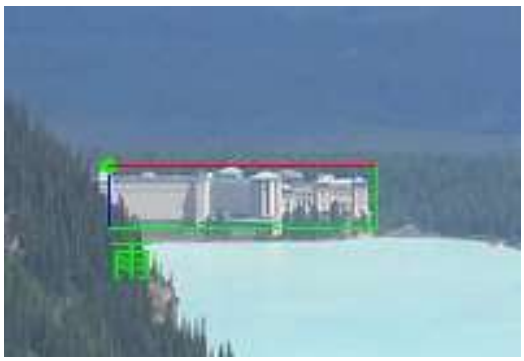
Tabulka 6 - Jaccardovo skóre pro jednotlivé obrázky z přílohy

Nízké hodnoty u obrázků „img16“ a „img23“ jsou způsobené falešným rozpoznáním nejspíše čínských znaků, které se na obrázcích vůbec nevyskytují. Obě falešná rozpoznání jsou ukázané na Obrázek 29 a Obrázek 30.

Celkově výsledky výše uvedeného testu průniků oblastí říkají, že aktuální rozpoznávání textů [7] není úplně špatné, nicméně pro český jazyk a standardizované texty má nedostatky, jako překrývaní oblastí, nepřesnost oblastí způsobené vlivem naučených dat a další.



Obrázek 29 – Chyba u přílohy img16, způsobena snahou sítě rozpoznávat i špatně čitelné texty



Obrázek 30 – Chyba u přílohy img23, způsobena snahou sítě rozpoznávat i špatně čitelné texty

6.3 Budoucí práce

Tato bakalářská práce bude sloužit jako základ pro diplomovou práci v navazujícím studiu. Případné nedostatky v generování dat se projeví až později, kdy budou syntetická data z této práce použita pro trénování neuronové sítě. Cílem bude zlepšit úspěšnost stávající sítě [7] v rozpoznávání českého jazyka a zpravodajských relací. V této práci nebyla změřena přesnost rozpoznávání textů, která je i pouhým pohledem špatná. Ta bude změřena a porovnána v budoucí navazující práci.

Prozatím se jako možné vylepšení jeví funkce „get_text“ třídy AITGM, která by mohla vracet texty začínající začátkem věty. Toto vylepšení by ovšem nemuselo být až tak užitečné, jak se může zdát, protože ve stávajícím provedení, by měla teoreticky necelistvost slov přispívat ke generalizaci sítě. Další možné vylepšení by mohlo být ve funkci „syntwild“, která vykresluje do obrázků náhodná data. Zde by se data mohla generovat ve více stylistických variantách a ve více barevných kombinací než nyní.

Kapitola 7

Závěr

Cílem této bakalářské práce bylo vytvořit generátor syntetických obrazových dat, který by imitoval zpravodajské relace. Důvodem je zlepšení stávajícího rozpoznávání, které dělá velké množství chyb, protože je trénované na jiná data a jazyky, než je potřeba.

Již při analýze dat se objevila komplikace v podobě nestálých dat, která není možno syntetizovat z důvodu různé polohy, velikosti písma a zkratek nebo symbolů. Muselo být proto počítáno s dodatečnou potřebou upravení řešení s ohledem na úspěšnost trénované sítě. Při implementaci se tudíž objevilo složité rozhodnutí ohledně struktury algoritmu. Protože nebylo možné experimentálně ověřit funkčnost řešení s ohledem na některá nestálá data, musel být kód psán takovou formou a takovým stylem, aby se v něm dalo rychle vyznat a dal se jednoduše upravit i jinak než formou změny konfiguračního souboru.

Vytyčeného cíle bylo s ohledem na limitující faktor navazující práce úspěšně dosaženo. Statistika úspěšnosti rozpoznávání oblastí v syntetických datech se stávající sítí je dle Jaccardova indexu (také IoU) průměrně kolem hodnoty 0,7. Statistika na přesnost rozpoznávaných textů bude provedena v navazující diplomové práci, kde bude přetrénován stávající model [7] a bude možné generátor dat vylepšit o znalosti navázané na trénování sítě.

Seznam obrázků

Seznam tabulek

Reference

[1] R. Van, G. a. Drake a L. Fred, „Python 3 Reference Manual,“ 2009. [Online].

[2] P. Umesh, „Image Processing in Python,“ 2012. [Online].

[3] A. Clark, „Pillow (PIL Fork) Documentation,“ 2015. [Online].

[4] L. Yann, B. Léon, B. Yoshua a H. Patrick, „Gradient-based learning applied to document recognition,“ v *Proceedings of the IEEE.*, 1998.

[5] N. Daniel, „What are convolutional neural networks,“ 28 December 2019. [Online]. Available: https://www.unite.ai/what-are-convolutional-neural-networks/.

[6] A. Krizhevsky, I. a. H. Sutskever a E. Geoffrey, „ImageNet Classification with Deep Convolutional Neural Networks,“ v *Advances in neural information processing systems.*, 2012.

[7] M. Bušta, Y. Patel a J. Matas, „E2E-MLT - an Unconstrained End-to-End Method for Multi-Language Scene Text,“ v *Asian Conference on Computer Vision.*, Springer, Cham, 2018.

[8] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He a J. Liang, „EAST: An Efficient and Accurate Scene Text Detector,“ v *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.*, 2017.

[9] G. Alex, F. Santiago a G. Faustino, „Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,“ v *Proceedings of the 23rd international conference on Machine learning.*, 2006.

[10] L. Fayao, S. Chunhua a L. Guosheng, „Deep Convolutional Neural Fields for Depth Estimation from a Single Image,“ v *Proceedings of the IEEE conference on computer vision and pattern recognition.*, 2014.

[11] P. Arbelaez, M. Maire, C. Fowlkes a M. Jitendra, „Contour Detection and Hierarchical Image Segmentation,“ v *IEEE transactions on pattern analysis and machine intelligence*, 2011.

[12] G. Ankush, V. Andrea a Z. Andrew, „Synthetic Data for Text Localisation in Natural Images,“ v *Proceedings of the IEEE conference on computer vision and pattern recognition.*, 2016.

[13] R. Adrian, „https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/,“ [Online].

Přílohy

Příloha 1 - img2

Příloha 2 - img3

Příloha 3 - img5

Příloha 4 - img6

Příloha 5 - img14

Příloha 6 - img16

Příloha 7 - img23

Příloha 8 - img25