Dokumentace k projektu do předmětu KNN
25. května 2021

License plate recognition

Anh Le Hoang, Radek Pištělák, Jan Šorm
xlehoa00/xpiste05/xsormj00@stud.fit.vutbr.cz
Fakulta Informačních Technologií, Brno

1 Úvod

V posledních letech bylo představeno několik nových návrhů modelů pro rozpoznávání textů v reálných scénách. V této práci se snažíme o jejich reimplementaci, zkombinování těchto několika různých postupů a následné experimentování s takto získánými různými modely na datové sadě registračních značek aut, se kterou pracovali a je i veřejně k dispozici v práci [1].

2 Příbuzné práce

Tato práce výchází z celkem tří příbuzných prací, které jsme studovali. První práce [1] se zabývá stejně jako naše přímo rozpoznáváním registračních značek s nízkou kvalitou a nízkým rozlišením. Jejich přístup je založen na konvoluční neurální síti, která holisticky zpracovává celý obraz a vyhýbá se segmentaci znaků poznávací značky. K tomu využívá celkem 8 výstupů, které se znaží určit znak na odpovídající pozici. V případě, že je značka kratší než 8 znaků, je nutné ji dále vyplnit pomocí speciálního znaku, dokud nedosáhne požadované pevné výstupní velikosti. Reimplementací sítě popsané v této práci se nám povedlo vytvořit základní řešení, se kterým jsme pak porovnávali naše další experimenty.

Druhá práce [2] se zabývá libovolným textem ve scéně a zaměřuje se především na různě zkreslené a zakřivené texty. Jejich model pak kromě sítě pro rozpoznávání obsahuje i lokalizační síť a následnou transformaci metodou Thin plate spline. Podobný problém je pak řešen i ve třetí práci [3]. Po části sítě pro vytvoření příznaků, následuje obousměrná rekurentní vrstva a poté attention vrstva. Cílem attention je určit, z jakých vstupů se má čerpat informace pro postupné určení znaků.

3 Model

Námi implementovaný model se skládá ze dvou částí: lokalizační sítě s tranformací TPS (*Thin plate spline*) a z rozpoznávací sítě, přičemž pomocí parametrů si lze nastavit, zda se má využít první část. Dále si pak lze pomocí parametrů zvolit i to, jaká ze dvou rozpoznávacích sítí se má použít. V následujících třech částech si rozebereme samostatně lokalizační síť, transformaci TPS a obě rozpoznávací sítě. Varianty byly natrénovaný na trénovací části datové sady **ReId**[1] obsahující 95331 anotovaných značek.

3.1 Lokalizační síť

Lokalizační síť, jejíž jednotlivé vrstvy můžete vidět v Tabulce 1, má za úkol nalézt v obrázku horní a spodní hranici kontrolních bodů, za pomoci kterých pak může proběhnout TPS transformace. V našem modelu jsme reimplementovali síť, která je popsána v článku [3], a drobně upravili její výstup. Na výstupu sítě není žadná aktivační funkce a je nutné nastavit počáteční váhy poslední vrstvy na 0 a biasy na hodnoty odpovídající souřadnicím počátčeních kontrolních bodů z rozmezí hodnot $\langle 0, 1 \rangle$.

Vrstvy	**Konfigurace**	**Výstup**
Vstup	obrázek	100×32
Konv1	c: 64 k: 3×3	100×32
BN1	-	100×32
Pool1	k: 2×2 s: 2×2	50×16
Konv2	c: 128 k: 3×3	50×16
BN2	-	50×16
Pool2	k: 2×2 s: 2×2	25×8
Konv3	c: 256 k: 3×3	25×8
BN3	-	25×8
Pool3	k: 2×2 s: 2×2	12×4
Konv4	c: 512 k: 3×3	12×4
BN4	-	12×4
APool	$512 \times 12 \times 4 \rightarrow 512 \times 1$	512
FC1	$512 \rightarrow 256$	256
FC2	$256 \rightarrow 2F$	2F
Upr. tvaru	-	$2 \times F$

Tabulka 1: Architektura lokalizační sítě

3.2 Transformace TPS

Transformace TPS (*Thin plate spline*) dostává na vstupu obrázek a kontrolní body získané z lokalizační sítě a výstupem pak je upravený obraz.

K tomu jsou využity dvě konstantní matice, které vycházejí z cílových pozic kontrolních bodů. Hlavní matematickou operací pro jejich určení je $\phi(r)$, což je RBF (*radial basis function*) aplikovaná na euklidovskou vzdálenost mezi p a c_k. Součástí je pak také transformační matice, která se získá vynásobením právě jedné ze zmíněných konstantních matic a rozšířené matice obsahující kontrolní body z lokalizační sítě. Pro pronásobení transformační matice a druhé konstantní matice obsahující RBF mezi všemi možnými body obrázku a cílovými kontrolními body se získá mřížka, která určuje kam se mají pixely ve vstupním obrázku přesunout ve výsledném obrázku. Výsledná mřížka a vstupní obrázek jsou vstupy pro funkci **grid_sample**, která pak upraví vstupní obrázek na základě mřížky. Více jsou jednotlivé vztahy rozepsány v článku [2], na jejichž základě jsme tuto transformaci implementovali.

3.3 Rozpoznávací síť

Množina znaků kterou sítě mohou rozpoznat obsahuje celkem 35 znaků obsahující všechny možné číslice a celou abecedu bez O, aby nedošlo k možné záměně s číslici 0.

Využíváme dvě možné architektury rozpoznávacích sítí. První z nich sloužila jako naše baseline a byla implementovaná na základě informací popsaných v článku [1]. Její strukturu lze vidět v Tabulce 2. Součástí vstupní abecedy je ještě speciální znak '#', který se využívá jako výplňovací znak pro značky obsahující méně než 8 znaků. Tento znak se opakovaně vkládá na 4. pozici, dokud text nenabývá délky 8. Síť obsahuje celkem 8 výstupů, kde každá vrací 36 hodnot, odpovídající pravděpodobnostem, že se znak s odpovídající hodnotou nachází na dané pozici. Síť je natrénovaná s chybovou funkcí cross-entropy.

Druhou rozpoznávací síť jsme pak reimplementovali na základě informací z článku [3] a její strukturu lze vidět v Tabulce 3. Výška výstupu této části odpovídá 1, aby se pak mohla tato dimenze odstranit a délka pak bude odpovídat jednomu časovému kroku pro následovné obousměrné LSTM. Zde jsme pak převzali mechanismus attention z následujícího repozitáře[1]. Součástí abecedy jsou poté dva speciální znaky značící startovací znak s indexem 35 a poté také ukončovací znak s indexem 36 a výsledná velikost odpovídá hodnotě 37. Výstupem sítě je tvar 9×37, kde 9 odpovídá maximální možné délce řetězce včetně ukončovacího znaku. Síť je natrénovaná s chybovou funkcí cross-entropy, kde během trénování je ignorován výstup s indexem odpovídající počátečnímu znaku.

Vrstvy	**Konfigurace**		**Výstup**
Vstup	obrázek		200×40
Konv1 + BN	c: 16	k: 3×3	200×40
Konv2 + BN	c: 16	k: 3×3	200×40
Pool1	k: 2×2	s: 2×2	100×20
Konv3 + BN	c: 32	k: 3×3	100×20
Konv4 + BN	c: 32	k: 3×3	100×20
Pool2	k: 2×2	s: 2×2	50×10
Konv5 + BN	c: 64	k: 3×3	50×10
Konv6 + BN	c: 64	k: 3×3	50×10
Pool3	k: 2×2	s: 2×2	25×5
Upr. tvaru	-		8000
$8 \times$ FC1	-		128
$8 \times$ FC2	-		36

Tabulka 2: Architektura baseline rozpoznávací sítě

Vrstvy	**Konfigurace**		**Výstup**
Vstup	obrázek		100×32
Konv1	c: 64	k: 3×3	100×32
Pool1	k: 2×2	s: 2×2	50×16
Konv2	c: 128	k: 3×3	50×16
Pool2	k: 2×2	s: 2×2	25×8
Konv3	c: 256	k: 3×3	25×8
Konv4	c: 256	k: 3×3	25×8
Pool3	k: 1×2	s: 2×2	25×4
Konv5	c: 512	k: 3×3	25×4
BN1	-		25×4
Konv6	c: 512	k: 3×3	25×4
BN2	-		25×4
Pool4	k: 1×2	s: 1×2	25×2
Konv7	c: 512	k: 2×2	24×1
	s: 1×1	p: 0×0	

Tabulka 3: Architektura rozpoznávací sítě 2.

4 Vyhodnocení

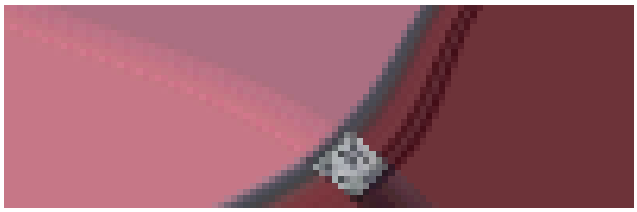
Pro vyhodnocení byly použity celkem 3 datové sady, kde první z nich je testovací část datové sady **ReId**[1] obsahující celkem 76412 značek a poté další dvě, které byly použity v článku pro

[1] https://github.com/clovaai/deep-text-recognition-benchmark

odstranění rozmazání[4], kde každá obsahuje 721, avšak jedna z nich zvaná **blur** obsahuje rozmazaný obrázek a druhá zvaná **deblur** obsahuje výstup sítě pro odstranění rozmazání. Obrázky o velikosti 264×128 v datových sadách jsou dále ořezány a výsledný výřez pak obsahuje střed obrázku o rozměrech 234×74. Vyhodnocení bylo provedeno i na neořezané datové sadě **deblur**, která má v názvu přidanou hvězdičku.

Jako experimenty jsme provedli vyhodnocení nad kombinacemi využívající dvou rozpoznávacích sítí, kde první z nich obsahuje 8 oddělených výstupů a mezi architekturou využívající princip attention. Jsou zde poté vyhodnoceny varianty, které byly natrénovány s augmentací dat, která kolem středu otáčí obrázek značky náhodně v rozmezí $\langle -10, 10 \rangle$ stupňů. Dále byly také vyhodnoceny varianty obsahující zarovnávací síť. Všechny varianty byly natrénovány s optimalizačním algoritmem **Adam** a koeficientem učení rovným 0.001.

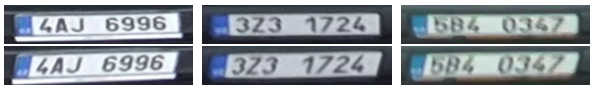
Při použití daného koeficientu učení pro attention síť se zarovnáním se nám nepodařilo výsledný model natrénovat kvůli tomu, že výstupem zarovnávací sítě byl nečitelný obrázek, kvůli příliš velké úpravě vah při zpětném průchodu. Při použití koeficientu učení rovným 10^{-5} se nám podařilo dosáhnout značně lepších výsledků zarovnání během trénování. Ukázky při použití vyššího učícího koeficientu lze vidět na obrázku 1 a na obrázku 2 lze vidět výstupy s nižším koeficientem z druhé epochy. Ukázky ořezávání lze vidět na obrázcích 3, 4, 5 a 6.



Obrázek 1: Obrázek zarovnání s učícím koeficientem rovným 0.001 z druhé epochy.

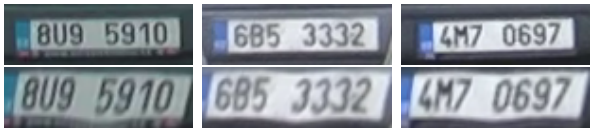
Obrázek 2: Obrázek zarovnání s učícím koeficientem rovným 10^{-5} z druhé epochy.

Výsledky vyhodnocení lze vidět v tabulce 4 obsahující procento správně určených značek z celkového počtu, která obsahuje i výsledky referenční sítě[1]. Dále pak lze vidět vypočtenou Levenštejnovu vzdálenost nad testovací části datové sady **ReId**.



Obrázek 3: Na prvním řádku jsou obrázky před zarovnáním a na druhém je baseline ořez.

Obrázek 4: Baseline ořez. s augmentací.



Obrázek 5: Attention ořez.

Obrázek 6: Attention ořez. s augmentací.

5 Závěr

Podařilo se nám naimplementovat a vyhodnotit několik různých architektur, které však s porovnáním s referenčními výsledky dopadly hůře. Příčinou tohoto může být fakt, že pro trénování byla použitá lepší augmentace dat a jak lze vidět, výsledky baseline se zvýšily i při použití základní rotace. Při porovnání výsledků mezi naimplementovanými architekturami dopadla nejlépe síť baseline využívající zarovnávací síť a augmentace dat na datové sadě **ReId**, avšak výsledky sítě využívající attention se zarovnáním by mohly být vylepšeny, protože výsledný model nebyl dostatečně dotrénován kvůli nedostatku času. Pro srovnání základní síť attention dosáhla nejlepších výsledků v 56. epoše, zatímco varianty se zarovnáním byly natrénovany pouze na 40 epochách. Ukázalo se, že výsledky zarovnání jsou také závislé na učícím koeficientu a možné výsledky by šly zlepšit pomocí nastavení nižšího učícího koeficientu.

Tabulka 4: Tabulka přesností jednotlivých sítí na datových sadách v procentech. Datové sady s hvězdičkou značí přesnosti rozpoznávání bez ořezávání.

Architektury	**ReId**	**blur**	**deblur**	**deblur***
baseline [1]	98.3	55.4	91.0	-
baseline reimp.	95.64	24.69	50.62	5.13
baseline s rotací	96.23	35.64	60.06	17.89
baseline se zarov.	97.34	30.79	57.28	11.51
baseline se zarov. i rot.	97.69	33.01	59.50	17.75
attention	95.88	43.41	81.55	33.15
attention s rotací	96.97	54.37	86.00	37.45
attention se zarov.	93.04	16.78	34.26	0
attention se zarov. i rot.	96.16	35.09	60.33	0

Tabulka 5: Tabulka přehledu Levensteinových vzdáleností na datové sadě **ReId** pro jednotlivé varianty spuštění.

Architektury	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8 a více**
baseline	73079	2150	589	232	145	104	87	22	4
baseline s rotací	73529	1944	411	196	126	96	85	32	2
baseline se zarov.	74378	1312	323	146	82	86	72	13	0
baseline se zarov. i rot.	74644	1170	238	139	96	52	59	14	0
attention	73262	2264	489	186	104	48	52	7	0
attention s rotací	73942	1777	376	132	82	39	58	6	0
attention se zarov.	71096	3997	844	270	116	49	35	4	1
attention se zarov. i rot.	73475	2166	458	192	70	22	26	2	1

Reference

[1] *J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík and P. Zemčík, "Holistic recognition of low quality license plates by CNN using track annotated data,"2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1-6, doi: 10.1109/AVSS.2017.8078501., Online: https://ieeexplore.ieee.org/document/8078501*

[2] *B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao and X. Bai, "ASTER: An Attentional Scene Text Recognizer with Flexible Rectification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 9, pp. 2035-2048, 1 Sept. 2019, doi: 10.1109/TPAMI.2018.2848939., Online: https://ieeexplore.ieee.org/document/8395027*

[3] *J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh and H. Lee, "What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis,"2019, Online: https://arxiv.org/abs/1904.01906*

[4] *HRADIŠ Michal, KOTERA Jan, ZEMČÍK Pavel a ŠROUBEK Filip. Convolutional Neural Networks for Direct Text Deblurring. In: Proceedings of BMVC 2015, Swansea, The British Machine Vision Association and Society for Pattern Recognition, 2015, ISBN 1-901725-53-7.*