Rozpoznávání dopravních značek

Shrnutí existujících přístupů, návrh a řešení projektu

Kočica F. — xkocic01*, Strych T. — xstryc05, Láníček A. — xlanic04

Tým č.13

8. května 2020

Úvod

Tato práce pojednává o *state of the art*[1] přístupu k rozpoznání státních poznávacích značek[2] nízké kvality, a to pouze za pomoci konvoluční neuronové sítě, která zpracovává celý snímek (t.j. bez segmentace jednotlivých znaků) a implementace takovéhoto modelu s využitím existujících datových sad z reálného prostředí doplněné o anotace. Dále tato práce pojednává o možnostech rozšíření takovéhoto systému za účelem zvýšení úspěšnosti a testování těchto hypotéz.

V první kapitole lze nalézt shrnutí existujících přístupů ke klasifikaci PZ. V dalších kapitolách postupně návrh architektury neuronové sítě, implementační detaily, popis experimentů s modelem a v poslední řadě testování, vyhodnocení a závěrečné shrnutí dosažených výsledků.

1 Existující přístupy

Některé z dnešních systémů pro rozpoznání poznávacích značek zahrnují segmentaci znaků jakožto jeden z kroků *pipeline*. Problém je, že pokud jsou znaky špatně segmentovány, selže následně i klasifikace znaků i přes to, že klasifikátor je značně robustní a úspěšný. Dalšími problémy (které ovšem stěžují všechny úlohy počítačového vidění) jsou např. rozmazanost, špatné světelné podmínky, šum, atd. Proto se dnes do popředí (již zmíněné tzv. *state of the art*) dostávají tzv. *end-to-end* systémy, které se neskládají ze složité *pipeline* skládající se z několika modulů, ale pouze jedné (konvoluční) neuronové sítě, která řeší všechny problémy "v jednom".

Tento moderní přístup byl k rozpoznání poznávacích značek použit v následujících pracích.

V práci [3] se autoři zaměřili na rozpoznání PZ s velmi nízkou kvalitou. Použili přístup s jednou konvoluční neuronovou sítí bez pomoci segmentace a dosáhli velmi nízké chybovosti 0.4/1.7, kde chybovost je uvedena jako znak/PZ, se kterou předčili ostatní volně dostupné i komerční systémy na různých datasetech. Poslední konvoluční vrstva sítě je připojena ke osmým větvím plně propojených vrstev, kde každá má 36 výstupů (použití pro PZ s osmi znaky, kde 36 výstupů je pro 26 znaků abecedy a 10 číslic). Každá větev rozpoznává právě jedno písmeno z poznávací značky. Uvedená úspěšnost potvrzuje, že je síť schopna se naučit i pozice jendotlivých znaků a není tedy třeba segmentace. Dále je schopna se naučit takové vlastnosti, které jsou robustní vůči efektům snižujícím úspěšnost (rozmazanost snímku, atd.).

V práci [2] použili autoři dvě konvoluční sítě za sebou. První, se čtyřmi vrstvami pro detekci PZ, a druhou, s devítí vrstvami pro následnou klasifikaci znaků. S použitím *sequence labeling* metody umožňující rozeznání celé

* Vedoucí týmu.

[1] Nejlepší přístup pro řešení dané problematiky.

[2] Dále jen PZ.

PZ bylo možné taktéž vynechat segmentaci znaků. Testy prokázaly, že s dostatečným množstvím dat je metoda velmi úspěšná, ale kvůli komplexnosti modelů také velmi pomalá a tudíž nedokáže pracovat v reálném čase.

Práce [1] se zabývala podobným problémem - rozpoznáváním čísel z panoramatických snímků Google Street View, kde byly jednotlivé snímky zpracovávány analogickým přistupem k [3]. I v tomto případě byly všechny potřebné kroky k úspěšnému rozpoznání číslic (lokalizace, segmentace a rozpoznání) integrovány v rámci hluboké konvoluční neuronové sítě. Zásadním závěrem této studie je, že pro efektivní řešení těchto problémů se ukazuje jako zcela zásadní hloubka sítě (konfigurace vykazující nejvyšší úspěšnost 97,84% byla složena z 11 vrstev). Mechanismus autoři popsali tak, že v prvotních vrstvách je řešen lokalizační a segmentační problém a díky tomu je připravena reprezentace vstupního obrázku pro pozdější vrstvy v takové podobě, že se mohou soustředit pouze na rozpoznávání. Autoři však dále zdůrazňují, že taková hloubka (>5 konvolučních vrstev) vyžaduje *velmi rozsáhlou sadu dat* pro efektivní natrénování.

2 Návrh

Tato kapitola se zabývá popisem námi navrženého řešení a experimentů, kterými se budeme snažit docílit lepšího (úspěšnějšího) modelu. Dále jsou zde zmíněny datové sady, které v práci použijeme.

2.1 Datové sady

K dispozici máme tyto sady použité v práci [3]:

- ReId – obsahuje 182.336 barevných PZ různé délky, rozmazání a mírného překrývání se.
- HDR – dátová sada byla zachycena kamerou DSLR a obsahuje tři různé expozice. Skládá se z 652 snímků PZ. Tato sada obsahuje i mírně pootočené snímky HDR, které se v sadě ReId nevyskytují.

Trénování provedeme na trénovací částí datové sady ReId (105.924 snímků) a vyhodnocení na testovací části ReId (76.412 snímků) a sadě HDR (652 snímků).

2.2 Architektura neuronové sítě

Použijeme síť popsanou v práci [3] a provedeme řadu experimentů s cílem zvýšení úspěšnosti sítě.

Data augmentace. Vzhledem k tomu, že trénovací sada neobsahuje pootočené značky, zatímco testovací sada (HDR) ano, provedeme augmentaci trénovacích dat, aby se model naučil klasifikovat také pootočené značky, ale i například rozmazané značky, klasifikaci i za špatných světelných podmínek atp.

Více vlastností. Metoda už tak dosahovala *state of the art* výsledků, ale přidáním dalších vnitřních (*hidden*) vrstev, neuronů do těchto vrstev, nebo více vlastností do plně propojených vrstev, čímž se síť dokáže naučit více vlastností a měla by tedy dosáhnout lepších výsledků ovšem vykoupenou nižší rychlostí klasifikace a trénování.

Ladění (ang. *tuning*) dalších hyperparametrů modelu. To je například změna velíkostí různých vrstev, přidání dropout, modifikace velikost batche, půlící vrstvy, aktivačních funkcí, provedení či odebrání batch normalizace a tak podobně.

3 Implementace

Všechny varianty naší neureonové sítě byly trénovány, stejně jako v referenční práci[3], na platformě tensorflow. Použili jsme k trénování prostředí colab (google).

Architektura a parametry modelu. Architektura modelu se skládá ze tří sekvencí konvolučních vrstev. Poslední vrstva je připojena k osmi větvím plně propojených vrstev sloužícím ke klasifikaci jednotlivých znaků PZ. Výstupem je tenzor velikosti $8 \times 36 \times 1$ obsahující pravděpodobnosti jednotlivých znaků na jednotlivých místech získaných pomocí aktivační funkce softmax. V případě CTC bylo třeba počítat s logity a proto byla funkce softmax nahrazena za lineární.

K trénování všech modelů byl použit optimizátor Adam s faktorem učení (ang. *learning rate*) rovno 0.001 a velikosti "dávky" (ang. *batch size*) 32. Trénování probíhalo po dobu 80 epoch, ale ve většině případů stačilo cca. 50, pak už došlo k přetrénování.

Jako chybová funkce byla použita kategorická cross entropie (či CTC), vhodná pro tuto klasifikační úlohu. Byla použita verze *spars*, kde štítky (ang. *labels*) jsou celá čísla reprezentující třídu (namísto systému *one-hot encoding*).

Každá z konvolučních vrstev je následována batch normalizací za účelem zvýšení rychlosti a úspěšnosti modelu. Každá ze třech (v defaultním modelu) sekvencí konvolučních vrstev je následována pooling vrstvou velikosti $2x2$ (vracející maximum lokální oblasti – nejvýraznější vlastnost) za účelem snížení velikosti vrstvy a množství vlastností potřebných k naučení (ang. *down-sampling*). Jako aktivace za konvolučními vrstvami byla použita funkce `ReLU`, stejně jako v referenční práci, ale bylo experimentováno i s `Leaky-ReLU`, která řeší problém zvaný "*dying ReLU*", protože nepřiřazuje všem záporným hodnotám nulu (neurony se poté nezaseknout v této části). Bias v modelu nebyl použit.

Proměnná délka PZ V rámci našeho modelu bylo nutné se vypořádat s nestejnou délkou PZ v datové sadě. Vzhledem k tomu, že datová sada byla pořízena v českých podmínkách, drtivou většinu (cca 95%) představovaly PZ o délce 7 znaků. Díky této skutečnosti jsme mohli na počátku implementace od proměnné délky abstrahovat a snížit komplexitu modelu za cenu menší přesnosti.

Po dokončení validace zjednodušeného modelu jsme pro implementaci proměnné délky použili stejný přístup jako [3], tedy stanovili limit pro maximální délku na 8 znaků. Chybějící znaky do tohoto počtu byly ve fázi načítání hodnot *ground truth* nahrazeny znakem # v potřebném množství vždy před poslední 4 znaky PZ[3].

Přesnost našich modelů se nicméně doimplementováním proměnné délky zvýšila jen nepatrně, což bylo pravděpodobně dáno již zmiňovanou drtivou převahou českých PZ v datové sadě.

Obrázek 1: Reprezentace PZ s proměnnou délkou

[3]Tento mechanismus kopíruje rozložení znaků na českých PZ.

Model	Chybovost pismen/značiek [%]	GPU [ms]	CPU [ms]
Základní	ReId - 0.53/1.4, HDR - 7.62/24.61	9.02	67.864
CTC + dropout	ReId - **0.33/0.8**, HDR - 6.79/21.05	**8.61**	**60.978**,
s Data Augmentací	ReId - 0.4/1.1, HDR - **3.83/13.31**	9.16	68.165
Augmentace + CTC + dropout + conv	—	—	—

Tabulka 1: Evaluace významných modelů/experimentů na datových setech ReId a HDR. Průměrný čas vyhodnocení byl naměřen na: CPU – 2 jádra Intel(R) Xeon(R) CPU @ 2.20GHz; GPU – Tesla K80. Poslední z modelů, který byl kombinací všech postupů, které zlepšily úspěšnost jsme bohužel nestihli dotrénovat.

4 Experimenty

Augmentace dat. Augmentácia dát bola vykonaná pomocou sekvencie z dôvodu, že sekvencia oproti generátorom garantuje, že sieť bude trénovaná počas epochy na každom vzorku len raz. Na augmentáciu dát bolo použitá externá knižnica s názvom albumentations[4]. Použili sme pri tom tieto transformácie: horizontálne otočenie, náhodná zmena kontrastu obrázku, náhodná gamma korekcia, zmena jasu, transformácie pomocou HSV (Hue,Saturation,Value) a náhodné affínne transformácie: rotácia, posun a zmena veľkosti.
Oproti základnému modelu tento model počas prvých epóch nedosahoval taký výsledok ako základný model pretože obsahuje obrovské množstvo obrázkov. Ale jeho úspešnosť bola oveľa lepšia už o pár epóch neskôr. Oproti základnému modelu sme na dasete ReId dosiahli o 0.1% lepší úspěšnosti a na datasete HDR dokonce o 11.3% lepší úspěšnosti. Markantný rozdiel pri datasete HDR, je spôsobený iným vystrihnutím fotiek a takisto preto, že trénovací dataset ReId neobsahuje pootočené obrázky PZ (které se model naučit klasifikovat díky augmentaci - přesněji pootočení snímků značek).

CTC – *Connectionist temporal classification.* CTC je typ výstupu neuronové sítě a k němu analogicky ztrátové (skórovací) funkce. CTC zkouší všechny možné všechny možné kombinace zarovnání *ground-truth* textu v obrázku a počítá sumu skóre všech těchto zarovnání. Pro použití CTC jsme potřebovali místo pravděpodobností použít logity a tedy jsme změnili aktivační funkci plně propojené vrstvy ze softmax na lineární. Hned po prvním použití CTC v naší implementaci modelu prezentovaného v referenční práci[3] byla od začátku trénování vidět velmi markantní změna. Hodnota chybové funkce byla od samého počátku **značně** nižší, než bez CTC, ovšem to vedlo k velmi rychlému přetrénování modelu a nakonec i velmi podobným výsledkům, jako bez CTC. Proto jsme model se CTC doplnili značným množstvím *dropout* funkcí (tři dropout vrstvy s hodnotou 0.3 umístěné za půlící vrstvy všech třech sekvencí konvolučních vrstev), což vedlo k opravdu zásadnímu snížení chyby modelu o 0.2/0.6 na ReId datové sadě. Při použití dropout u modelu bez CTC úspešnost klesla o několik procent. Obecně nám ale model přišel vyváženější, pravděpodobnosti neoscilovaly s každou další epochou o několik procent nahoru či dolů, ale byly ustálené a klesaly až po přetrénování modelu.

Zvýšení počtu vlastností (ang. *features*) v plně propojených vrstvách. Model se defaultně skládá z osmi větví dvou plně propojených vrstev, kterým

[4] `https://github.com/albumentations-team/albumentations`

Model	Najhoršie znaky/#chyb	Najlepšie znaky/#chyb
Základní	F,G,#/7; 3/6; N/5	1,4,7,9,A,B,C,E,H,J,K,L,M,P,R,S,T,U,Z/0
CTC	F,G/7; 1/3;	4,6,7,9,A,C,E,H,J,K,L,M,R,S,T,U,Z/0
s Data Augmentací	F,G,#/7; N/5,	1,2,4,5,7,8,9,A,B,C,E,H,J,K,L,M,P,R,S,T,U,Z/0

Tabulka 2: Investigace znaků ve kterých jednotlivé modely dělají nejčastěji a naopak nejméně často chyby – Měli jsme v plánu vytvořit augmentací hlavně značky s chybovými znaky, aby se síť tyto znaky lépe naučila a dělala v nich méně chyb – to jsme ovšem už nestihli, pouze obecnou augmentaci na všechny snímky.

Model	1	2	3	4	5	6	7	8	spolu
Základní	5	7	8	0	7	6	7	1	41
CTC	3	8	3	0	7	2	3	1	27
s Data Augmentací	5	7	8	0	7	2	0	1	30

Tabulka 3: Počet chyb na jednotlivých pozicích značek (pozice 1 je první znak PZ, pozice 8 je poslední znak PZ) na datasetu ReId.

bylo přidáno malé množství vlastností, ale chybovost zůstala stejná.

Zvýšení počtu konvolučních vrstev. Model se defaultně skládá ze tří sekvencí třech konvolučních vrstev. Experimentovali jsme přidáním další konvoluční vrstvy do každé ze tří sekvencí. Trénování i doba vyhodnocení se lehce zpomalily a výsledkem byla značně nižší chybovost, proto jsme se rozhodli přidat tuto změnu do nejlepšího modelu, ve kterém bylo použito: CTC, augmentace, dropout a přidání konvolučních vrstev.

Navýšení počtu filtrů v konvolučních vrstvách. Konvoluční vrstvy v modelu se defaultně skládájí z 32/64/128 filtrů. Tato čísla byla zvýšena na 64/128/256 což dvojnásobně zvýšilo velikost modelu a dobu trénování. Výsledkem byla větší chybovost.

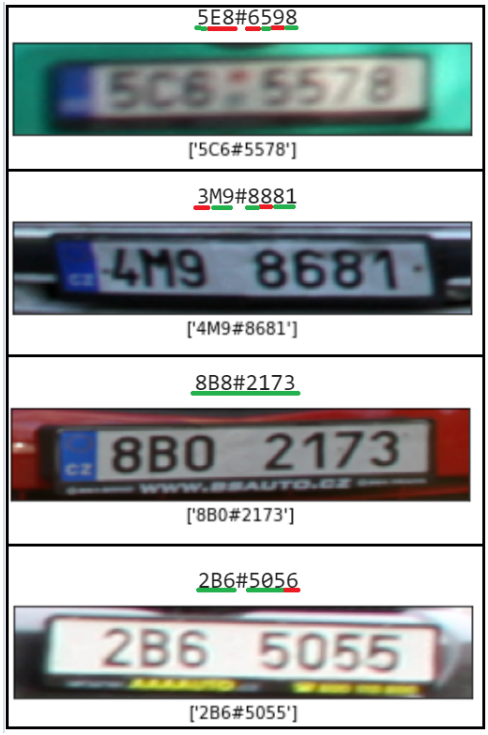
Přidání dropout vrstev. Od počátku se nám zdálo trénování na 80 epoch zmíněných v referenční práci [3] příliš mnoho. Zhruba od 50. epochy se výsledky modelu nijak nezlepšovaly, a proto jsme se rozhodli vyzkoušet přídání jedné vrstvy dropout s hodnotou 0.5 za poslední půlící vrstvu. Chybovost modelu se mírně snížila a proto jsme se rozhodli ji použít i v kombinaci s nejlepším modelem používajícím CTC k získání ještě nižší chybovosti.

Změna batch size. Většina modelů byla trénována s batch size velikosti 32, proto jsme se rozhodli zkusit změnit tento parametr na hodnotu 64, ale výsledky zůstaly téměř stejné, proto jsme dále používali 32 k zachování konzistentnosti vůči dřívějším modelům.

Normalizace snímků před trénováním. Vstup naší neuronové sítě byly snímky o velikosti $200 \times 40 \times 3$, tedy RGB snímky s hodnotami jednotlivých pixelů $0 - 255$. Existují tři typy normalizace: 1) Hodnoty od 0 do 1; 2) Hodnoty od -1 to -1; 3) Hodnoty od ? do ?, ale se středem v 0. My jsme se rozhodli pro první z těchto možností aby data měla stejné "měřítko" (ang. *scale*), což vede ke zrychlení konvergence a zlepšení přesnosti. Tato modifkace modelu neměla žádný účinek, nejspíše proto, že v našich modelech po každé konvoluční vrstvě provádíme batch normalizaci.

5 Testování

Pro testování jsme vytvořili jednoduché utility, které vyhodnocovaly různé modely na datových sadách popsaných v kapitole 2. Počítaly úspěšnost (%), chybovost (písmena/značky), a také na kterých pozicích nejčastěji vznikaly falešně pozitivní vzorky (nesprávné predikce) a také které znaky byly nejčastěji špatně klasifikovány.



Obrázek 2: Příklad několika správně i nesprávně klasifikovaných značek. Vyhodnoceno pomocí základního modelu. Pozn.: Nad PZ výstup modelu (zeleně – správně klasifikované znaky, červeně – špatně klasifikované znaky), pod PZ *ground-truth* hodnota.

Datová sada. Pro trénování a validaci při trénování byl použit dataset ReID[3], který obsahuje snímky SPZ s velmi nízkou kvalitou, a to v poměru 105410 trénovacích vzorků a 76032 vzorků validačních. Pro testování byla použita datová sada HDR[3] čítající 652 snímků PZ.

6 Závěr

Úloha rozpoznání PZ bez segmentace znaků popsaná v práci[3] byla úspěšně reprodukována ve všech ohledech a bylo dosaženo velmi malé chybovosti. Práce byla rozšířena rozmanitými experimenty, které v několika případech přinesly značně nižší chybovost (t.j. datová augmentace, CTC, dropout, více konv. vrstev). Rychlost s referenční prací se nedá smysluplně porovnat, protože jsme používali colab notebook od google s různě výkonnými zdroji (cpu/gpu).

Odkazy

[1] Ian Goodfellow et al. "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks". In: *ICLR2014*. 2014.

[2] Hui Li a Chunhua Shen. "Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs". In: *CoRR* abs/1601.05610 (2016). arXiv: 1601.05610. URL: http://arxiv.org/abs/1601.05610.

[3] Jakub Špaňhel et al. "Holistic recognition of low quality license plates by CNN using track annotated data". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. Srp. 2017, s. 1–6. ISBN: 978-1-5386-2939-0. DOI: 10.1109/AVSS.2017.8078501.

A Rozdělení bodů

Login	Odvedená práce slovně	%
xkocic01*	Neuronová síť, CTC, experimenty, dokumentace	39%
xstrych05	Augmentace, nástroje k vyhodnocení, dokumentace	33%
xlanic04	Proměnná délka, dokumentace	28%

Tabulka 4: Tabulka specifikující odvedenou práci a rozdělení bodů mezi jednotlivé členy týmu.

B Obsah přiloženého média

Adresářová struktura (po rozbalení komprimovaného souboru) je následující:

- `src/` – Obsahuje zdrojové kódy vytvořených modelů ve formátu python notebook. V podadresáři `experimental/` lze nalézt méně významné modely použité pouze jako pokusy.
- `results/` – Obsahuje textové soubory výstupů našich vyhodnocovacích nástrojů pro jednotlivé modely.
- `models/` – Obsahuje soubory exportovaných modelů používané pro testování ve formátu json.
- `utils/` – Obsahuje nástrojé používané zejmnéna k vyhodnocení úspěšnosti, chybovosti a rychlosti modelů.

* Vedoucí týmu.