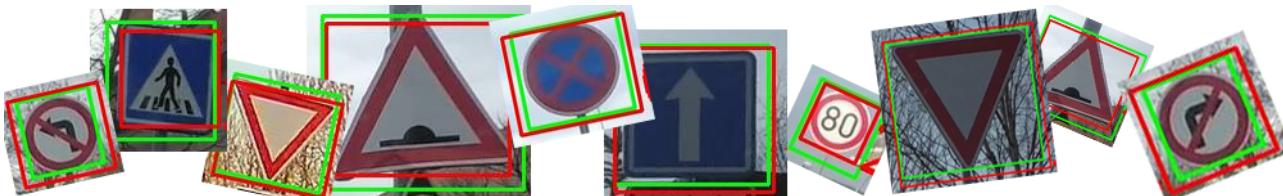


Detekce dopravních značek a semaforů

Tomáš Chocholatý*



Abstrakt

Článek se zabývá detekcí dopravních značek v obrazu. Cílem této práce je vytvoření vhodného detektoru pro detekci a rozpoznání dopravního značení v reálném provozu. Problematika detekce je řešena pomocí konvolučních neuronových sítí (CNN). Za účelem trénování neuronových sítí byly vytvořeny vhodné datové sady, které se skládají ze syntetického i reálného datasetu. Pro zhodnocení kvality detekce byl vytvořen program kvantitativního vyhodnocování.

Klíčová slova: Detekce dopravních značek — Konvoluční neuronové sítě — Dektekce objektů v obrazu

Přiložené materiály: [Demonstration Video](#)

*xchoch07@fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology
herout@fit.vutbr.cz, vedoucí bakalářské práce, Faculty of Information Technology, Brno University of Technology

1. Úvod

Rozpoznávání dopravního značení je jedním z hlavních témat spojených s rozvojem autonomního řízení automobilů. V reálné dopravě jsou nároky na detekci dopravních značek ztíženy nejen potřebou rychlé a spolehlivé detekce, ale také potřebou detekce i za zhoršených podmínek způsobených vlivem vnějšího prostředí.

V současné době existují systémy TSR [1] (Systémy pro rozpoznávání dopravního značení), které spadají do pokročilých systémů asistence řidiče (ADAS). Tyto systémy mají však řadu omezení. Jsou schopny detekovat pouze značky omezující rychlosť nebo značku zakazující předjíždění. Většina těchto systémů pro detekci používaných dnes v reálném provozu pracuje na starším způsobu detekce, který je založený na segmentaci zpracovávaného obrazu [1]. S modernějším způsobem detekce pomocí CNN [2] se dnes už můžeme setkat u prototypů autonomních aut.

Dopravní značky jsou navrhovány tak, aby měly vysoký kontrast vůči okolí a byly pro řidiče automobilu

dobře zpozorovatelné. Jejich základní tvary v kombinaci s vysoce kontrastními barvami zaručují jejich dobrou viditelnost. U dopravních značek je také snaha odlišit od sebe jednotlivé tvary tak, aby nedošlo k jejich nechtěné záměně, což by mohlo vést ke vzniku dopravní nehody.

Pro detektor by mělo být dopravní značení díky vysokému kontrastu dobré detekovatelné. V reálném prostředí však existuje mnoho vnějších faktorů, které mohou kvalitu detekce značek značně ovlivnit.

Během dne se střídají různé světelné podmínky, od špatné viditelnosti za tmy či šera, až do přílišného osvětlení, což má za následek vybledlé barvy a tzv. „přepálení bílé“. Zhoršené světelné podmínky způsobují i jiné faktory. Mezi ně můžeme zařadit dešť, mlhu či sníh. Tyto faktory snižují dohledovou vzdálenost a proto je nezbytné, alespoň do jisté míry zaručit i za těchto ztížených podmínek co nejlepší detekci značek.

Světelné podmínky nejsou však jediným zdrojem problémů. Při detekci bylo nutné zohlednit i různé staří značek. Dopravní značky jsou během své životnosti

43 vystaveny různým povětrnostním podmínkám a jejich
44 vzhled se tak může lišit v závislosti na jejich stáří.
45 Mohou být vybledlé, dochází ke změně odstínu barev,
46 nesou známky různých mechanických poškození.

47 Detektor by neměl být ovlivněn těmito faktory
48 a měl by s touto skutečností počítat a tím zvyšovat
49 kvalitu a míru detekce.

50 Při trénování modelu detektoru byly proto připraveny
51 i záznamy obsahující výše popsané problémy.

52 Pro vlastní detektor jsem použil nástroj YOLO
53 s frameworkm Darknet, který nakonec s použitou ar-
54 chitekturou sítě Tiny YOLO vykazoval nejlepší poměr
55 rychlosti a kvality detekce [3].

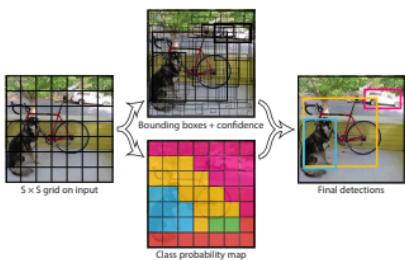
56 2. YOLO

57 Pro tento projekt byla použita architektura sítě YOLO
58 (You Only Look Once) s frameworkm Darknet [4].

59 Tato architektura sítě sleduje celý obraz v době
60 trénování a testování, takže jeho předpovědi jsou ovli-
61 vněny globálním kontextem v obraze. Tento sjedno-
62 cený model má několik výhod ve srovnání s ostatními
63 detekčními systémy, má však specifické nároky na
64 vhodný dataset. O této problematice bude pojednáno
65 níže v sekci 4.

66 Yolo na rozdíl od R-CNN (které používá dvě sítě,
67 konkrétně síť pro oblast návrhu sítě pro generování
68 regionálních návrhů a síť využívající tyto návrhy k de-
69 tekci objektů), využívá pouze jednu síť, která je zod-
70 povědná jak za detekci tak i klasifikaci. Dosahuje tak
71 větší rychlosti, než již zmiňované R-CNN.

72 Při detekci algoritmus YOLO rozdělí vstupní obraz
73 na mřížku o velikosti $S \times S$ [4]. V případě, že se
74 střed objektu vyskytuje v dané buňce, je následně tato
75 buňka zodpovědná za detekci pro tento objekt. Každá
76 buňka následně predikuje určitý počet rámečků a skóre
77 důvěry pro tyto rámců. Skóre důvěry je pravděpodobnost
78 s jakou si je detektor jistý, že se daný objekt v této
oblasti vyskytuje [5].



79 Obrázek 1. Yolo - detekce objektů [4]

80 3. Validace

81 Pro zhodnocení modelů bylo vytvořeno kvantitativní
82 vyhodnocování. To probíhá nad vlastní testovací da-

tovou sadou, která obsahuje reálné snímky z dopravy. 83
Pro testovací datovou sadu bylo pořízeno 899 fotek, 84
které obsahují 1563 anotovaných značek. 85

86 Pro účely kvantitativního vyhodnocování jsem vy-
87 tvořil program v jazyce Python, který vyhodnocuje 88
úspěšnost detekce a na základě získaných údajů vykreslí 89
ROC křivku, a grafy, které vypovídají o úspěšnosti 90
modelu.

91 Pro míru přesnosti detekce je použito IoU [6] (In-
92 tersection over Union). IoU je metrika používaná
93 k přesnosti detektoru objektů. Pro výpočet IoU po-
94 třebujeme dvě sady ohraničujících boxů:

- 95 1. originální rámeček, který přesně definuje, kde
96 se hledaný objekt nachází
- 97 2. rámeček, který vznikl predikcí detektoru

98 Podle vzorce $\text{IoU}(A, B) = \frac{A \cap B}{A \cup B}$ se vypočítá hodnota
99 IoU. Hraníční hodnota IoU byla pro tento detektor
100 zvolena 0,5. Vše, co je detekováno s úspěšností větší
101 jak 0,5, je považováno za true positive, naopak vše
102 s přesností nižší než je tato hranice je definováno jako
103 false negative.

104 IoU dokáže vyhodnotit přesnost detekce, ale ne-
105 poskytuje informace potřebné ke globálnímu zhodno-
106 cení kvality modelu nad validačním datasetem. Pro
107 tento účel byla použita ROC křivka [7]. ROC křivka je
108 nástroj pro hodnocení a optimalizaci binárního klasi-
109 fikačního systému, který ukazuje vztah mezi speci-
110 ficitou a senzitivitou daného testu nebo detektoru pro
111 všechny přípustné hodnoty prahu.

112 ROC křivka nám tedy udává poměr mezi mírou
113 detekce a počtem falešných detekcí.

114 Celá validace probíhá následujícím způsobem. Nej-
115 dříve je spuštěn framework Darknet s příznakem valid. 116 Tento program vygeneruje pro každou třídu soubor.
117 Ten obsahuje názvy obrazů z testovacího datasetu, kde
118 byla detekována tato třída, míří se kterou si je detektor
119 jistý, že se jedná o danou třídu a souřadnice predik-
120 vaného rámce. Validační program poté porovná tyto
121 záznamy s očekávanými detekcemi. O úspěšnou de-
122 tekci se jedná tehdy, když hodnota IoU je větší jak
123 0,5. Na základě výsledků je vykreslen graf, který
124 znázorňuje ROC křivku. Na osu Y se vynese True
125 positive rate a na osu X je vynesena počet false positive
126 (FP) pro celý testovací dataset.

127 Podle hodnoty skóre F1 [8] (celkové měřítko pře-
128 snosti modelu) je nakonec vybrán nejlepší práh pro
129 detekci a ten je následně použit pro konečnou validaci,
130 kdy obrazy s predikovanými i originálními bound-
131 boxy jsou uloženy.

132 Pro vizualizaci výsledku jsem do frameworku Dark-
133 net doimplementoval export videa s detekovanými
134 značkami.

4. YOLO - trénování a tvorba datasetu

Pro trénování konvoluční neuronové sítě jsem hledal vhodnou datovou sadu.

Většina datových sad, které jsou dnes volně k dispozici, není příliš vyhovujících. Ukázku takového datasetu můžeme vidět na obrázku 2. Jak je zřejmé, tato datová sada obsahuje ve velkém množství pouze ořezané značky. První pokusy trénování detekčního modelu na tomto datasetu nedopadly úspěšně.



Obrázek 2. Ukázka volně dostupného datasetu [9]

Yolo se při tréninku učí z celého obrázku a zahrnuje také velikost objektu, jeho umístění v obraze, okolí a další aspekty. V této datové sadě, jejíž ukázku vidíme na obrázku 2, tyto aspekty však chybí. Výsledkem bylo, že detektor identifikoval pouze ty značky, které stejně jako na trénovacím datasetu zabíraly více jak 80% plochy vstupního obrazu a zbytek značek nebyl klasifikován, tudíž se jedná o false negative.

Z nedostatku volně dostupných vhodných reálných datových sad jsem vytvořil svůj vlastní syntetický dataset.

V programovacím jazyce Python jsem vytvořil generátor datasetu. Ten převeze několik vzorků reálných značek a postupně je vkládá do připravené sekvence snímků, které byly pořízeny z palubní kamery jedoucího vozidla. Cílem bylo napodobit běžné umístění značek v provozu. Značky z reálného provozu byly před spuštěním generátoru upraveny tak, že okolí značek je obořeno černou barvou, která je následně při vkládání vyklíčována. Okraje značek jsou ve výsledném obrázku ostré a nedochází k prolínání vrstev. Za účelem dosažení co nejreálnějšího výsledku byly značky vkládány na náhodná místa s náhodnou velikostí.

Původní plán byl generovat velikost značky v závislosti na umístění v obraze. Čím by byla značka více uprostřed, tím by byla menší. Více ke krajům by se velikost značky zvětšovala, což by simulovalo přibližování a mijení značky z palubní kamery auta. Ve skutečném provozu se však ukázalo, že tuto teorii nelze do takové míry předpokládat, protože plně neodpovídá všem umístěním dopravních značek.

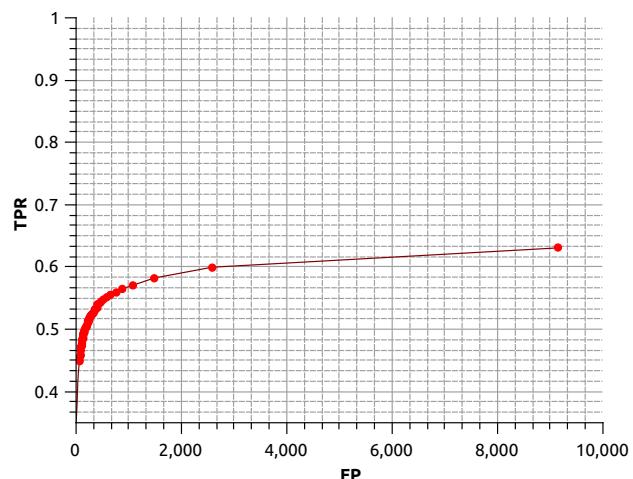
Po použití takto vygenerovaného vlastního datasetu však nebyly výsledky příliš uspokojivé. Oproti původnímu datasetu s ořezanými značkami se v tomto případě zlepšila schopnost detekce. Pokrok nastal ve

schopnosti detekovat značky různých velikostí. Většina značek, které neměly dobrý kontrast a světlé podmínky, stále však nebyla detekována. Z těchto důvodů byla v dalším kroku do generátoru přidána simulace vnějších vlivů. Vkládaným značkám byly upravovány gamma korekce, světlost, kontrast, šum a rozmažání. To mělo za úkol simulovat situace, kdy se značka nachází ve stínu, nebo je detekce prováděna v šeru. Naopak přidání světlosti a kontrastu simulovalo dopadající sluneční paprsky na značku, které se dále odrázejí, ale také vyblednutí značky a různé odchylky v odstínech barev. Dále byly na vkládaných značkách provedeny deformace prostřednictvím otáčení značky kolem vertikální a horizontální osy. Ukázku generování konečné verze datasetu lze vidět na obrázku 3.



Obrázek 3. Ukázka vlastního syntetického datasetu

Po trénování modelu s touto verzí datasetu se výsledky výrazně zlepšily. Nadpoloviční většina z validačního datasetu byla detekována, což je patrné na ROC křivce na obrázku 4.

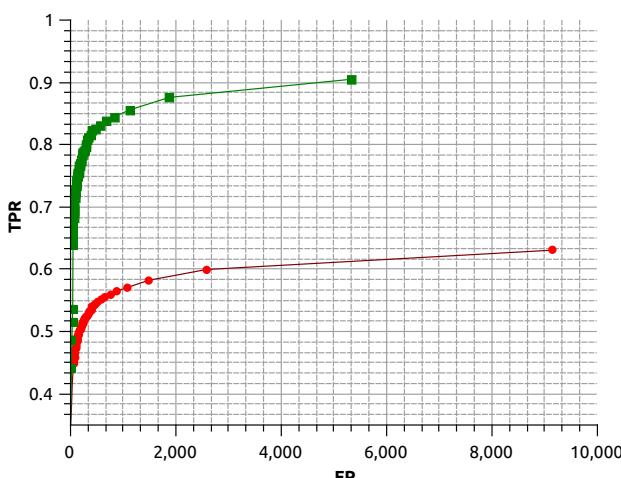


Obrázek 4. ROC křivka

Další trénování modelu nad syntetickým datasetem lepší výsledky již nepřineslo a to ani po rozšíření datasetu dalším generováním.

Bylo tedy nutné dataset nějakým způsobem inovovat. Aby byla míra detekce uspokojivější, rozšířil jsem syntetický dataset o anotované snímky z reálného

datasetu. Malá část datasetu byla převzata od Ing. Tomáše Svobody [10], zbytek reálného datasetu byl pořízen ze záznamů palubní kamery. Záznamy byly pořízeny v různých lokalitách, aby obsahovaly větší rozmanitost prostředí. Dataset obsahuje záznamy z Jihlavы a okolí, Prahy, Brna a Zlína. Ty byly následně ručně anotovány. Celkem bylo anotováno cca 8000 snímků. Právě přidání části reálného datasetu přineslo tížené výsledky. Počet úspěšně detekovaných značek se rasantně zvýšil, zatímco počet false negativů a false pozitivů byl výrazně nižší. Porovnání ROC křivek před a po přidání reálného datasetu do syntetického lze vidět na obrázku 5. Zelená křivka znázorňuje úspěšnost po přidání reálného datasetu.



Obrázek 5. ROC před a po přidání reálného datasetu

zlepšit.

V pokračujícím vývoji bude snaha o zrychlení detektoru. V současné době se detekce na grafice Nvidia GeForce 1050Ti pohybuje okolo 15 fps. Cílem je doslat tuto hodnotu na 25-30 fps, což by se dalo označit, že aplikace pracuje v reálném čase. Dalším důležitým cílem práce bude otestovat, které vlastnosti syntetického datasetu nejvíce zkvalitňují míru detekce, dále otestovat různé deformace současného datasetu a sledovat jejich vliv na kvalitu následné detekce. Cílem bude zjistit, co nejvíce ovlivňuje zvýšení kvality detekce při trénování a na tyto prvky se zaměřit a zvýšit tak kvalitu generovaných syntetických datasetů do budoucna.

6. Poděkování

Tato práce byla vytvořena pod vedením Prof. Ing. Adama Herouta, Ph.D. Tímto bych mu rád poděkoval za veškerou pomoc během práce a cenné rady. Také bych rád poděkoval metacentru Cesnet za poskytnutí výpočetních zdrojů potřebných pro trénování modelů detektoru.

Literatura

- [1] Technavio. Automotive traffic sign recognition system market to grow at a CAGR of around 30% by 2020. <https://www.technavio.com/pressrelease/automotive-traffic-sign-recognition-system-market-grow-cagr-around-30-2020>.
- [2] ujjwalkarn. An Intuitive Explanation of Convolutional Neural Networks. <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>.
- [3] Jonathan Hui. Real-time Object Detection. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088.
- [4] pjreddie. YOLO: Real-Time Object Detection. <https://pjreddie.com/darknet/yolo/>.
- [5] Mauricio Menegaz. Understanding YOLO. <https://grail.cs.washington.edu/wp-content/uploads/2016/09/redmon2016yol.pdf>.
- [6] Dr. Adrian Rosebrock. Deep learning for computer vision with python. Pyimagesearch.
- [7] WikiSkripta. ROC křivka . https://www.wikiskripta.eu/w/ROC_k%C5%99ivka.

5. Závěr

Trend bezpečnosti a snaha o autonomní řízení dnes v automobilovém průmyslu prudce roste. Poptávka po systémech rozpoznávání dopravních značek na trhu vyplývá z rostoucí potřeby omezit dopravní nehody. Je tedy velkým zájem tyto systémy do budoucna zlepšovat.

Práce dokazuje, že pouze syntetický dataset, generovaný tímto způsobem, se nevyrovná datasetu z reálného prostředí. Je potřeba ruční anotace, která, jak lze vidět v článku, má významný vliv na míru a kvalitu detekce. Spojení obou datasetů přineslo lepší výsledky, neboť syntetický dataset není omezen velikostí datové sady, která se může takto vygenerovat, což zaručuje, že v datasetu jsou zastoupeny značky všech různých velikostí a pozic. Oproti tomu, reálný dataset je omezen jeho rozsahem, ale do modelu zase přináší vlastnosti, které výrazně přispívají k lepšímu detekování.

Výsledná aplikace na testovací sadě dosahuje maximální úspěšnosti 90,5% a byla testována i v reálném provozu. Aktuálně je schopna detektovat 34 tříd dopravních značek. V současné době stále probíhá trénování tohoto modelu, což by mělo výsledky ještě mírně

- 288 [8] Wikipedia. F1 score. https://en.wikipedia.org/wiki/F1_score.
- 289 [9] Institut für Neuroinformatik. German Traffic Sign Benchmarks. <http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>.
- 290 [10] Bc. Tomáš Svobada. Detekce, lokalizace
291 a rozpoznání dopravních značek. https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=117978.
- 292
- 293
- 294
- 295
- 296
- 297
- 298