BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2019
http://excel.fit.vutbr.cz

Segmentace obrazových dat pomocí hlubokých neuronových sítí

Bc. Radek Pazderka

Abstrakt

Tento článek se věnuje problému segmentace scény z dopravního prostředí řešený pomocí hlubokých neuronových sítí. Popisuje celý proces vývoje. Konkrétně se jedná o přípravu datové sady a její rozšíření o nová vygenerovaná data pomocí GTAV. Následně je uveden model sítě DeepLavV3+ a jeho proces trénování. V závěru se tento natrénovaný model optimalizuje pomocí TensorRT na GPU a následně testuje jeho kvalita.

Klíčová slova: Segmentace obrazu, konvoluční neuronová síť, analýza dopravy, DeepLabv3+

*xpazde14@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Úvod

V dnešní době informací je stále větší požadavek na zpracovávání nasbíraných obrazových dat jako je například klasifikace, detekce, sledování objektů ve videu nebo např. segmentace celé scény obrazu popř. získání hloubkové mapy obrazu a mnohé další problémy. K takovému množství dat již nestačí lidská síla a musí se začít využívat počítače. K řešení těchto problémů se v dnešní době používají neuronové sítě. Neuronové sítě se inspirovaly lidským mozkem. Mozkové buňky neboli neurony jsou v lidském mozku mezi sebou propojené pomocí synapsí. Každé propojení neuronů má určitou sílu. Když do neuronu ze vstupních neuronů projde silný vzruch, tak se daný neuron aktivuje. Stejný koncept je i v neuronových sítí. Každý neuron v počítačovém světě je definovaný jako suma vstupních neuronů vynásobena silou propojení neboli váhou. Když tento součet dosáhne určité hodnoty je aktivační funkcí neuron aktivován.

V tomto článku se zaměříme na konvoluční neuronové sítě, které provádí segmentaci celého obrazu (dále jen SS-CNN). Jinými slovy vytvoříme neuronovou síť, která bude klasifikovat každý pixel vstupního obrazu. Výstupem této neuronové sítě bude maska vstupního obrazu, která bude uvádět přesné pozice a tvary všech definovaných tříd.

Výsledky této práce se použijí v praxi k segmentaci obrazu z dopravní situace, převážně z parkovišť. Bude se jednat o součást systému, který monitoruje a zjišťuje zaplněnost daného parkoviště. Neuronová síť, která bude výstupem této práce bude pomáhat systému v lepším pochopení a popsání situací vzniklých na silnici.

Může se jednat o detekci stojících aut v zakázaných zónách, spočítání podélně parkujících aut, určení volných míst u podélného parkování nebo případně detekování zácpy křižovatky, apod.

2. Definice problému

Cílem této práce je vytvoření SS-CNN, u které se bude brát co největší důraz na kvalitu výstupu bez ohledu na časovou náročnost výpočtu. K vytvoření takovéto sítě bude potřeba:

- Připravit co nejkvalitnější datovou sadu, nejlépe složenou z pohledů na vozovku z výšky do deseti metrů, na které bude síť používána.
- Navrhnout strukturu sítě, která bude schopná naučit se všechny objekty a bude schopná se přizpůsobit na nejrůznější pohledy na vozovku.
- Vytvořit systém, který bude mít na vstupu datovou sadu a strukturu sítě a dokáže síť optimálně trénovat, testovat nebo validovat.

Bonusem této práce bude akcelerace výsledného natrénovaného modelu na grafickou kartu pro dosažení optimálního výpočtu.

3. Existující řešení

Mezi *state of the art* modely sítí pro sémantickou segmentaci se řadí DeepLabv3+ [1], který se vyznačuje ASPP (Atrous Spatial Pyramid Pooling) modulem s dilatačními konvolucemi, který dává důraz na větší okolí bodu a tím lépe pochopí kontext scény. Díky tomuto modulu si zvětšíme receptivní pole [2]. Je to důležité pro to, abychom dokázali správně segmentovat velké plochy (silnice, budovy, oblohu). Tento model byl využit k segmentaci scény z pohledu jedoucího vozidla.

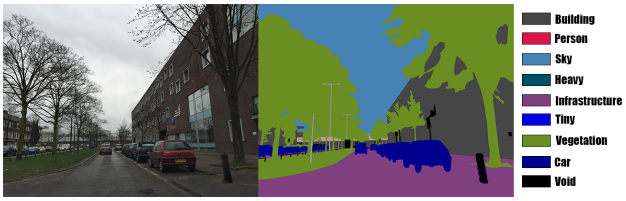
Existuje mnoho dalších sítí pro sémantickou segmentaci, které jsou optimalizované na jiné problémy. Např. model sítě U-Net [3] byl určen pro medicínské účely, pro segmentaci buněk. Další modely mohou být PSPNet [4], MobileUNet, FC-DenseNet apod.

4. Vlastní řešení

Jak bylo řečeno v kapitole 2 bude potřeba vytvořit tři základní věci, bez kterých se žádná neuronová síť neobejde. Jedná se o datovou sadu, strukturu neuronové sítě a prostředí pro trénování sítě. V této kapitole si tyto části ve stručnosti popíšeme a uvedu zde postup, kterým jsem postupoval.

4.1 Datová sada

Datová sada je jednou z nejdůležitějších částí při trénování neuronových sítí. Proto se v této práci velice dbá na to, aby byla datová sada co nejlépe připravena. Datová sada by měla obsahovat snímky scény z dopravního prostředí, k nim korespondující masky a celá datová sada by měla obsahovat soubor s popisky jednotlivých barev v masce. Na Obrázku 1 je uvedená datová sada.



Obrázek 1. Ukázka datové sady.

Datovou sadu jsem rozdělil na dvě části. Datová sada v **první části** slouží pro předtrénování sítě[1]. Neboli se jedná o natrénování obecné sítě. Datová sada se skládá z volně přístupných datových sad - KITTI [5], CamVid [6], CityScapes [7], Mapillary-Vistas[8] a GTAV-street [9]. Všechny tyto datové sady jsem sloučil do jedné datové sady, která obsahuje devět tříd. Seznam všech tříd je uveden na Obrázku 1. Všechny tyto dílčí datové sady obsahují pohled z jedoucího vozidla. My však potřebujeme anotované datové sady z vyššího pohledu. Ideálně z pouliční lampy, případně obrázky z nízko letícího drona, který snímá vozovku. Tyto datové sady bohužel nejsou volně dostupné, tak si musíme pomoci jinak. Ruční anotování obrázků je velice časově náročná činnost, tak nám musí pomoci známá hra GTAV. Pomocí ní jsem si vygeneroval anotovanou datovou sadu, kterou jsem nazval GTAV-parking. Tato datová sada patří do **druhé části** datové sady určené pro dotrénování sítě (fine-tune).

Získání datové sady z GTAV. Firma Artin s.r.o mi poskytla kostru programu pro anotování textur, získání stencil bufferu a originálního snímku ze hry GTAV. Stencil buffer obsahuje masku, která obsahuje informace o lidech, vozidlech, vegetacích a obloze[2]. Zbytek textur musíme anotovat ručně.

Mým úkolem bylo tento zapůjčený software modifikovat k vlastnímu použití. Jednalo se o:

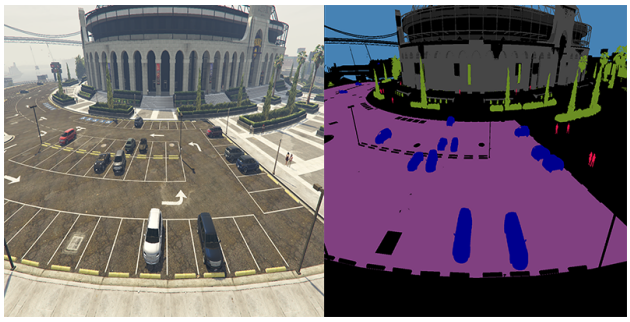
- Anotace nových textur - textury budov, infrastruktury, atd.
- Práce s kamerou, která danou dopravní situaci v určené výšce a pod určitým úhlem obkrouží a zaznamená data.
- Po každé dokončené trajektorii vygeneruje nové rozestavení aut na parkovišti nebo se s kamerou

[1] Datová sada, která zachycuje podobný obecný problém, který řešíme.

[2] http://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/

přemístí do jiné lokace pro monitorování dalších dopravních situací. A celý proces opakuje.

- Neuronová síť musí umět segmentovat tři typy dopravních prostředků. Do prvního typu patří motorky, skútry, kola. Do druhého typy patří osobní auta a dodávky[3]. A třetí typ jsou těžké dopravní prostředky jako jsou kamiony, autobusy, vlaky apod. Bohužel ze stencil bufferu nedokážeme rozeznat o jaký typ se jedná, musel jsem si ze hry dopočítat bounding box všech vozidel a následným postprocessem tyto typy ve stencil bufferu rozlišit.
- Sjednocení obrázku s anotacemi a stencil bufferu. Výsledná ukázka vygenerovaného snímku je na Obrázku 2.



Obrázek 2. Ukázka datové sady vygenerované z GTAV.

4.2 Struktura sítě

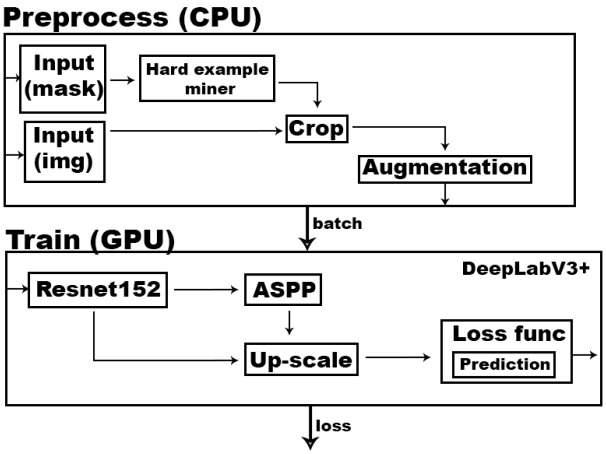
Struktury sítí pro sémantickou segmentaci jsou převážně typu kodér-dekodér (Encoder-Decoder), které v části kódování zmenšuje rozlišení vstupního obrázku a zvětšuje počet kanálů, neboli se jedná o získávání mapy aktivací (feature map). Využívají se k tomu známé feature extractory, které se dobře umístily v celosvětové soutěži ImageNet [10]. Toto malé rozlišení je nutné zvětšit do původní velikosti pomocí dekonvolucí nebo bilineární interpolace, abychom získali masku vstupního obrázku.

V této práci jsem zvolil *state of the art* model sítě DeepLabV3+. Tento model vyžaduje feature extractor pro získání mapy aktivací. Zvolil jsem Resnet152. Jedná se o velice robustní klasifikační síť, která má 152 vrstev a tím dosahuje vysoké přesnosti. Model sítě DeepLabV3+ je uveden na Obrázku 3.

4.3 Trénování

Pro trénování neuronových sítí jsem využil framework Tensorflow. Celý projekt je napsaný ve skriptovacím jazyce Python.

Proces trénování začíná u předzpracování, do kterého vstupuje obrázek a jeho maska. Pomocí masky se vybere oblast, která má pro nás největší informační hodnotu. Jedná se o oblast, ve které se vyskytuje velké množství objektů, které síť např. dlouho neviděla. Není vhodné síti předávat stále vzorky pozadí[4] z důvodu toho, že by se málo trénovala na popředí[5]. Tuto práci má na starosti *Hard example miner*, uvedený na Obrázku 3. Po získání souřadnic ideální oblasti, se tato oblast vyřízne z masky i ze vstupního obrázku. Po vyříznutí se data augmentují. Je to z důvodu toho, aby se síť učila na stále jiných obrázcích a tím byla schopná lépe se přizpůsobit reálným podmínkám. Augmentace může obraz nepatrně rozmazat nebo naopak zaostřit, přidat šum, otočit obrázek, apod. Celý tento proces běží paralelně na CPU a připravuje přesně takovou strukturu, kterou očekává vstup trénování. Jedná se o batch neboli 4-dimensionální tensor (N, C, W, H)[6]



Obrázek 3. Ilustrace procesu trénování.

Druhá část slouží k trénování neuronové sítě. Jak bylo již řečeno, model sítě jsem zvolil DeepLabv3+ s Resnet152 pro extrakci mapy aktivací. Výstupem této části je hodnota loss, která obecně udává rozdíl mezi predikcí sítě a očekávaným výstupem. Cílem trénování je minimalizace loss funkce.

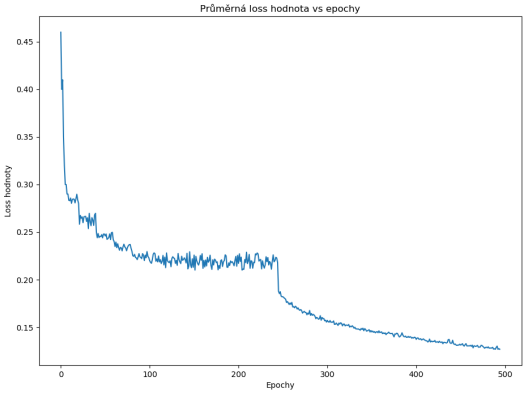
Na grafu 4 je uvedena závislost loss hodnoty na počtu epoch trénování. Je vidět, že od hodnoty od epochy 250 razantně klesly. Bylo to z důvodu přidání *Hard example minera* do procesu předzpracování obrázků, jak bylo popsáno výše.

[3] obecně auta do 3,5 tuny

[4] budovy, obloha, silnice, vegetace

[5] auta, nákladní auta, motorky, lidé

[6] N-počet obrázků, C - počet kanálů, W - šířka, H - výška



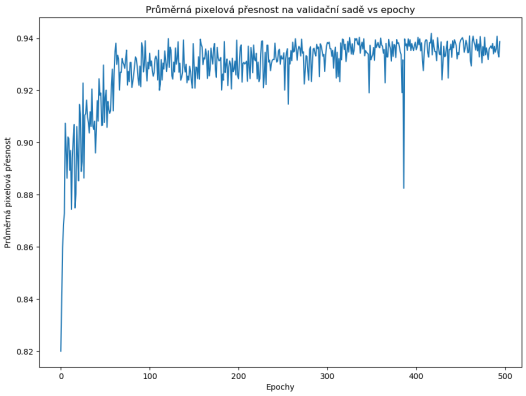
Obrázek 4. Graf závislosti loss hodnot na počtu epoch.

Celou síť jsem trénoval na grafické kartě RTX 2080 titan, kde jedna epocha (čítající 36 000 anotovaných obrázků) trvala 32 minut. Celé trénování trvalo přibližně 11 dní.

4.4 Testování

V této sekci se zaměříme na testování sítě na validační datové sadě. Jedná se o datovou sadu, která nebyla přidaná do trénovací datové sady. Konkrétně se jedná o náhodně vybraných 1000 obrázků ze sloučené datové sady[7].

První metrikou určující kvalitu sítě je metrika úspěšnosti. Jedná se o procentuální vyjádření správně klasifikovaných pixelů. Graf 5 zobrazuje vztah úspěšnosti klasifikace na počtu epoch.

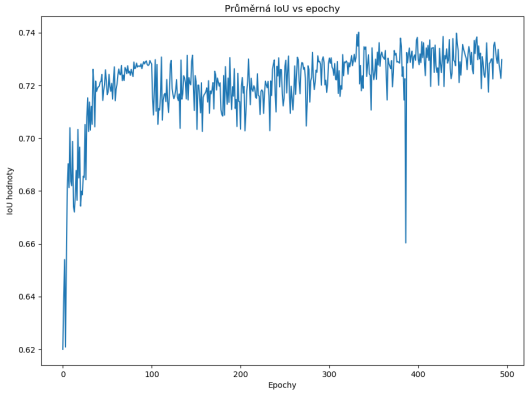


Obrázek 5. Graf závislosti přesnosti klasifikace pixelů na počtu epoch.

Druhá metrika IoU (Intersection over Union) určuje procentuální překrytí mezi predikcí a ground truth maskou, jak je uvedeno v rovnici 1. Graf závislosti metriky IoU na počtu epoch je uveden na Obrázku 6.

[7] Obsahuje jak reálná, tak i umělá data.

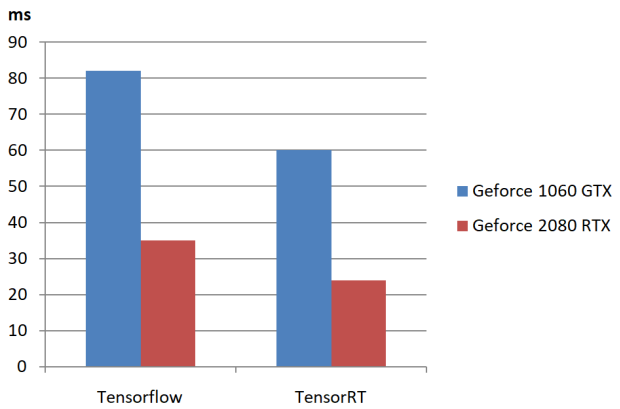
$$IoU = \frac{GT\,maska \bigcap predikce}{GT\,maska \bigcup predikce} \tag{1}$$



Obrázek 6. Graf závislosti IoU na počtu epoch.

4.5 Optimalizace

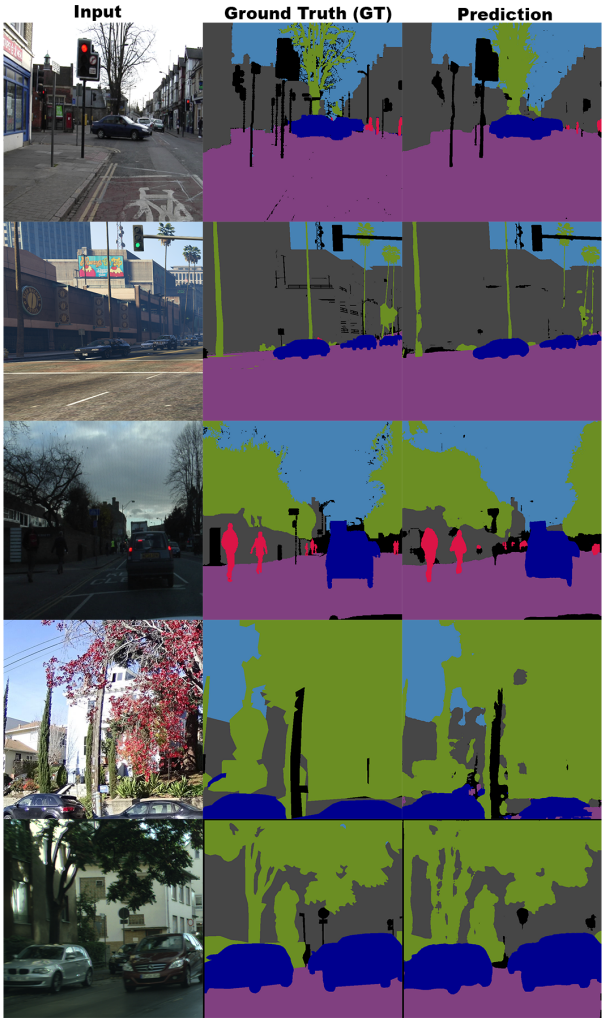
TensorRT je platforma, která slouží k optimalizaci modelů neuronových sítí pro rychlé vyhodnocení na grafické kartě. Je velice vhodné po natrénování modelu sítě, tuto síť optimalizovat pomocí TensorRT. Jak je uvedeno na grafu 7 bylo docíleno vysokého zrychlení vyhodnocení za cenu nepatrného snížení přesnosti.



Obrázek 7. Optimalizace pomocí TensorRT.

5. Experimenty

Testování kvality výstupu sítě na snímcích z pohledu jedoucího auta jsou velice kvalitní, jak je demonstrováno na Obrázku 8.



Obrázek 8. Ukázky výstupů sítě.

Při pohledech z vyšších pozic kvalita výstupu značně klesá. Tento problém se musí řešit rozšířením datové sady o nové snímky z daných pohledů.

6. Přínos

Výstup této práce má velice kvalitní výsledky segmentace z pohledu jedoucího auta. Časová náročnost predikce je 25ms na GPU pomocí TensorRT, která je velice uspokojující. Tato rychlost umožňuje real-time zpracování, což je velice žádoucí pro např. samořídící auta, roboty a jiné přístroje, které potřebují monitorovat svoje okolí a na základě získaných dat reagovat.

Na druhou stranu je síť bohužel zatím specializovaná na tento pohled. S posunutím kamery do vyšší výšky predikce ztratí svojí kvalitu. Tento problém budu řešit přidáním nových dat pomocí generátoru datové sady z GTAV.

7. Závěr

Tento článek se věnoval celému procesu vývoje neuronové sítě sloužící k segmentaci scény. Od vytváření datové sady, přes proces trénování až k optimalizacím a experimentům s neuronovou sítí.

Síť dosahuje 94% přesnosti klasifikace pixelů, 74% IoU s rychlostí výpočtu 82 ms bez optimalizací a 25 ms s optimalizacemi s TensorRT.

Budoucí práce bude spočívat v obohacení datové sady o nová data a dotrénování sítě s těmito daty.

Poděkování

Nejprve bych rád poděkoval firmě RCE s.r.o za odbornou pomoc a za možnost využití jejich hardwaru pro práci s neuronovými sítěmi. Dále bych rád poděkoval firmě Artin s.r.o, která mi poskytla program pro generování anotovaných textur ze hry GTAV. Dále bych rád poděkoval svému vedoucímu diplomové práce Ing. Jaroslavu Rozmanovi, Ph.D. za jeho věcné připomínky na konzultacích.

Literatura

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[2] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. *CoRR*, abs/1701.04128, 2017.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[4] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[6] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30:88–97, 2009.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017.

[9] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.