VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE VOZIDEL V OBRAZE A VIDEU

VEHICLE DETECTION IN IMAGE AND VIDEO

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **DALIMIL ROZPRÝM**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JAKUB ŠPAŇHEL**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Rozprým Dalimil**
Program: Informační technologie
Název: **Detekce dopravních prostředků v obraze a videu**
Vehicle Detection in Image and Video
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s metodami detekce objektů v obraze
2. Pořiďte vhodnou datovou sadu pro detekci různých druhu vozidel v obraze.
3. Vyhledejte metody hlubokého učení zaměřující se na problematiku detekce objektů, primárně na metody podporující detekcí více tříd.
4. Vyberte vhodné metody a experimentujte s nimi.
5. Vhodným způsobem vyhodnoťte vybrané metody a diskutujte dosažené výsledky.
6. Vytvořte plakát a video prezentující vaší práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních tří bodů zadání
- Rozpracovaný 4. bod zadání
- Odevzdání rozepsaného textu práce

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Špaňhel Jakub, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 30. října 2020

Abstrakt

Cílem této práce je porovnání dostupných vícetřídních detektorů při detekci silničních vozidel na vhodně vytvořené datové sadě. Jako vícetřídní detektory byly vybrány neuronové sítě určené k detekci a klasifikaci objektů v obraze. Experimentováno je s detektory Mask R-CNN, YOLOv4 a YOLACT++, které jsou v práci popsány. Výběr detektorů zastupuje různé architektury a přístupy k detekci. Pro účely učení a testování je v práci detailně popsána vytvořená datová sada a její parametry. Detekce je testována na obraze z běžného silničního provozu a samostatně na částečně překrytých objektech. Výsledkem práce je znovupoužitelná a rozšiřitelné datová sada, naměřené výsledky dosažené při detekci a jejich hlubší rozbor.

Abstract

The goal of this thesis is comparison of available multiclass detectors abilities to detect road vehicles on purposely created dataset. As multiclass detectors are chosen neural networks for detection and classification of objects in image. Detectors described in this text and used for experimentation are Mask R-CNN, YOLOv4 and YOLACT++. This selection encompasses multiple different architectures and approaches to object detection. Created dataset used for learning and testing is thoroughly described in this text. Detection capability of detectors is tested on images from casual traffic and separately on partially covered objects. The outcome of this thesis is reusable and expandable dataset, measured performance values and their deeper exploration in this text.

Klíčová slova

detekce objektů, hluboké učení, konvoluční neuronové sítě, Mask R-CNN, YOLOv4, YOLACT++, střední průměrná přesnost

Keywords

object detection, deep learning, convolutional neural networks, Mask R-CNN, YOLOv4, YOLACT++, mean average precision

Citace

ROZPRÝM, Dalimil. *Detekce vozidel v obraze a videu*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jakub Špaňhel

Detekce vozidel v obraze a videu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jakuba Špaňhela. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Dalimil Rozprým
11. května 2021

Poděkování

Chtěl bych poděkovat vedoucímu své práce za vytvoření zadání, poskytnutí odborné podpory, předzpracovaných dat a pravidelných konzultací.

Obsah

Kapitola 1

Úvod

Detekce objektů v obraze je jednou z nezbytných součástí pro pochopení a zpracování obrazu počítačem. Stále se rozvíjejícím řešením tohoto problému jsou detektory postavené na konvolučních neuronových sítích, využívající různé přístupy a architektury k optimalizaci jejich běhu. Cílem práce je otestovat a porovnat vícero dostupných detektorů při použití pro detekování silničních vozidel.

Kapitola 2 poskytuje teoretickou informační základnu pro pochopení fungování jednotlivých detektorů. Zabývá se popsáním procesu detekce od nejnižšího stupně – samotných konvulčních neuronových sítí a postupně se v abstrakci posouvá k popisu architektur detektorů a nakonec i reálných řešení.

Pro účely experimentování s detektory je nezbytné vytvoření a popsání vhodné datové sady, zaměřující se na silniční vozidla. Na ní je pak možné detektory natrénovat a provést testování v kontrolovaných podmínkách. Tímto problémem se zabývá kapitola 3, která detailně rozebírá její strukturu a přípravu, protože charakteristiku datové sady je také třeba brát v úvahu při vyhodnocování a intrepretaci výsledků.

V kapitole 4 je zdokumentován proces experimentování. Popsáno je učení jednotlivých detektorů a následné provádění testů. Pro účely vyhodnocení jsou zavedeny vhodné metriky obecně používané ke kvantifikaci výsledků této problematiky. Po uvedení dosažených hodnot je následně elaborováno s jejich interpretací. Detailnější náhled do charakteru detekce je poskytnut formou vizualizace nasbíraných dat.

Kapitola 2

Umělé neuronové sítě

V posledních letech došlo ke značnému posunu řešení počítačového zpracování obrazu a vidění, zejména díky užití konvolučních neuronových sítí [19]. Obecně jsou, umělé neuronové sítě, výpočetní systémy určené k přeměně vstupních hodnot na požadovaný výstup. Jejich fungování je inspirováno biologickou nervovou soustavou [23].

Základní stavební jednotkou takové sítě je umělý neuron, jehož jediným cílem je emitování své aktivace na základě nějakých vstupů. Matematicky je aktivace i-tého neuronu y_i hodnotou jeho aktivační funkce f, pro sumu součinů jeho vstupů y_{ij} a jejich vah w_{ij} [27]:

$$y_i = f(\sum_{n=1}^{j} w_{ij} y_j).$$

Neurony jsou v síti uspořádány do vzájemně propojených vrstev. Aktivace neuronů v jedné vrstvě se tak přenáší jako vstup do vrstvy následující. Nejjednoduší struktura sítě může být tvořena vstupní vrstvou neuronů pro získání prvních hodnot, skrytou vrstvou a výstupní vrstvou neuronů, ze kterých je možné číst výstupy. Takto jednoduchá struktura by však nebyla schopna řešit složitější problémy, většinou se tak lze setkat s hlubokými sítěmi, jejichž architektura obsahuje více než jednu skrytou vrstvu. [23]

Pro účely počítačového vidění je vstupem sítě reprezentace obrazu – například v případě barevného snímku (modelu RGB) o velikosti 1920×1080 pixelů, vstup tvoří 6 220 800 ($1920 \times 1080 \times 3$) hodnot – jedna pro každou barevnou složku každého pixelu. Forma výstupní vrstvy se pak odvíjí od řešeného problému. Typickým příkladem by mohla být klasifikace daného obrazu, u které by každá z možných tříd měla svůj neuron, jehož aktivace by znamenala přítomnost dané třídy ve vstupním obraze. [23]

2.1 Konvoluční sítě

Konvoluční neuronové sítě při vyhodnocování využívají konvoluce. Tato operace je obzvláště vhodná pro zpracování obrazu, kde lze s její pomocí extrahovat rysy k dalšímu zpracování [23]. K tomu určené vrstvy se nazývají konvoluční a jejich funkce je založená na tzv. *kernelech* (jádrech), tedy maticových filtrech, navržených pro rozpoznávání nějakého konkrétního rysu. Jednoduchým příkladem takového kernelu je Sobelův filtr (nebo také operátor), pro nalezení vertikálních a horizontálních hran, s maticemi tvaru viz obrázek 2.1. [29]

Každý neuron konvoluční vrstvy má své určené receptivní pole ve vstupní reprezentaci, nad kterým provádí skalární součin s maticí kernelu k získání svého výstupu. Toto receptivní pole vždy zahrnuje celou hloubku plochy vstupního obrazu. [23]

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

(a) Varianta pro vertikální hrany.

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

(b) Varianta pro horizontální hrany.

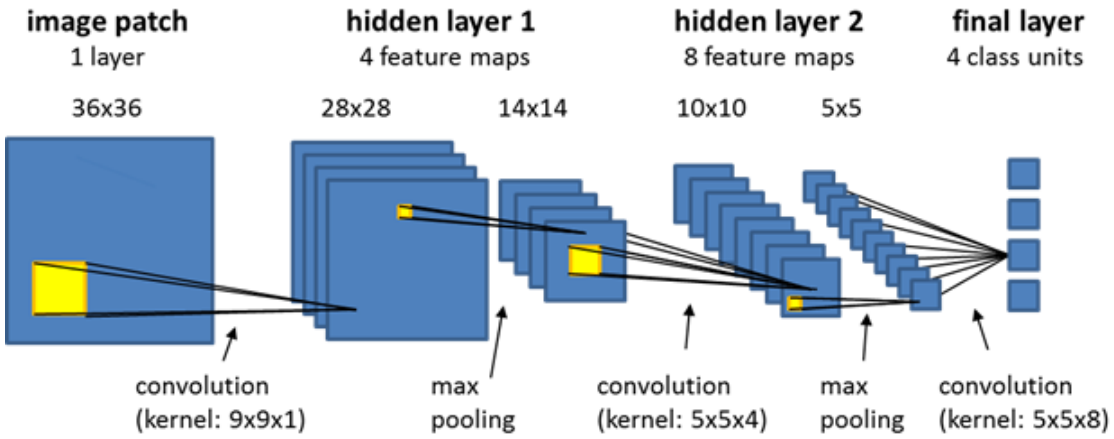
Obrázek 2.1: Matice kernelu, známého jako Sobelův filtr, pro extrakci hran [29].

Výstupem celé konvoluční vrstvy je aktivační mapa (*feature map*) pro každou úroveň hloubky vstupu. Její velikost vychází z rozmístění receptivních polí ve vstupní reprezentaci, s většími maticemi kernelů nebo vzájemnými překrytími receptivních polí se tato mapa změnšuje. [23]

Model konvoluční sítě také typicky obsahuje *pooling* vrstvy, jejichž funkcí je zmenšování vyhodnocované reprezentace a s tím i snižování výpočetní komplexnosti. Tato operace je většinou realizována pomocí *max* filtrů, přenášejících maximální hodnotu z filtrovaného regionu. *Pooling* je ale z podstaty věci destruktivním procesem, při kterém dochází ke ztrátám informace a potenciálně i ztrátě výkonu celé sítě. [23]

V konvoluční síti pro zpracování obrazu se také nachází běžné, plně propojené vrstvy, ve kterých jsou vstupem jednoho neuronu všechny výstupy předcházející vrstvy a analogicky je jeho výstup emitován do všech neuronů vrstvy následující. [23]

Jednoduchý příklad konvoluční sítě pro klasifikaci 4 různých tříd je možné vidět na obrázku 2.2.



Obrázek 2.2: Příklad jednoduché architektury konvoluční neuronové sítě obsahující 2 *pooling* a 3 kovoluční vrstvy s různými velikostmi kernelů. Převzato z [11].

2.2 Hluboké učení

Učením neuronové sítě se rozumí proces úpravy vah vstupů neuronů, za cílem dosažení lepších výsledků při vyhodnocování výstupu. Je zcela nezbytné pro vytvoření modelu schopného rozhodování a plnění potřebných úloh. Název hluboké vychází z typické struktury neuronových sítí obsahujících vícero skrytých vrstev. [23]

Jedná se o učení s učitelem, což pro případ zpracování obrazu znamená využití předpřipravených označených reálných objektů (například *bounding boxů*, tříd nebo pixelových

masek). Je realizováno algoritmem zpětného šíření chyby, kterým se postupně upravují váhy neuronů při opačném průchodu celou sítí, ve snaze minimalizovat chybovou funkci nebo také chybu (dále *loss*). Ta reprezentuje odchýlení predikce od validního výsledku (v tomto případě tedy například detekce objektů). [13]

Při učení může docházet k *overrfittingu*, což je případ příliš velkého přizpůsobení učené sítě učícím datům. *Overfitting* pak typicky vede k horšímu výkonu sítě při vyhodnocování nových, dosud neviděných vstupů. Většinou k němu dochází vlivem nedostatku učících dat, či příliš dlouhého učení. Pro jeho předejití je vhodné během učení sledovat změny hodnot *lossu* a proces adekvátně upravovat. [8]

V kontextu učení sítí pro zpracování obrazu je ještě třeba zavést pojmy *batch*, iterace a epocha, pro měření a popis tohoto procesu. Pro učení je třeba mít vhodná učící data – množinu obrázků (datovou sadu) s označenými správnými detekcemi, na které může detektor trénovat svoji predikci. Detektor se pak postupně učí zpracováváním jednotlivých obrázku, přičemž se jeden tento krok nazývá *step* nebo také iterace. Díky výpočetní akceleraci grafických karet je možné síť vyhodnocovat i pro několik obrázků zároveň. Množina obrázků, co je takto vyhodnocována v jedné iteraci, se nazývá *batch*, který může mít různou velikost dle dostupného výpočetního výkonu. Během učení je běžné, že detektor zpracovává obrázky opakovaně a epocha tak označuje počet iterací potřebných pro jeden průchod celou datovou sadou. Počet iterací v jedné epoše je ještě samozřejmě ovlivněn velikostí *batche*, kterou je třeba vydělit celkový počet obrázků v datové sadě, pro získání počtu iteracích nezbytných k dokončení jedné epochy. [28]

2.3 Zpracování obrazu

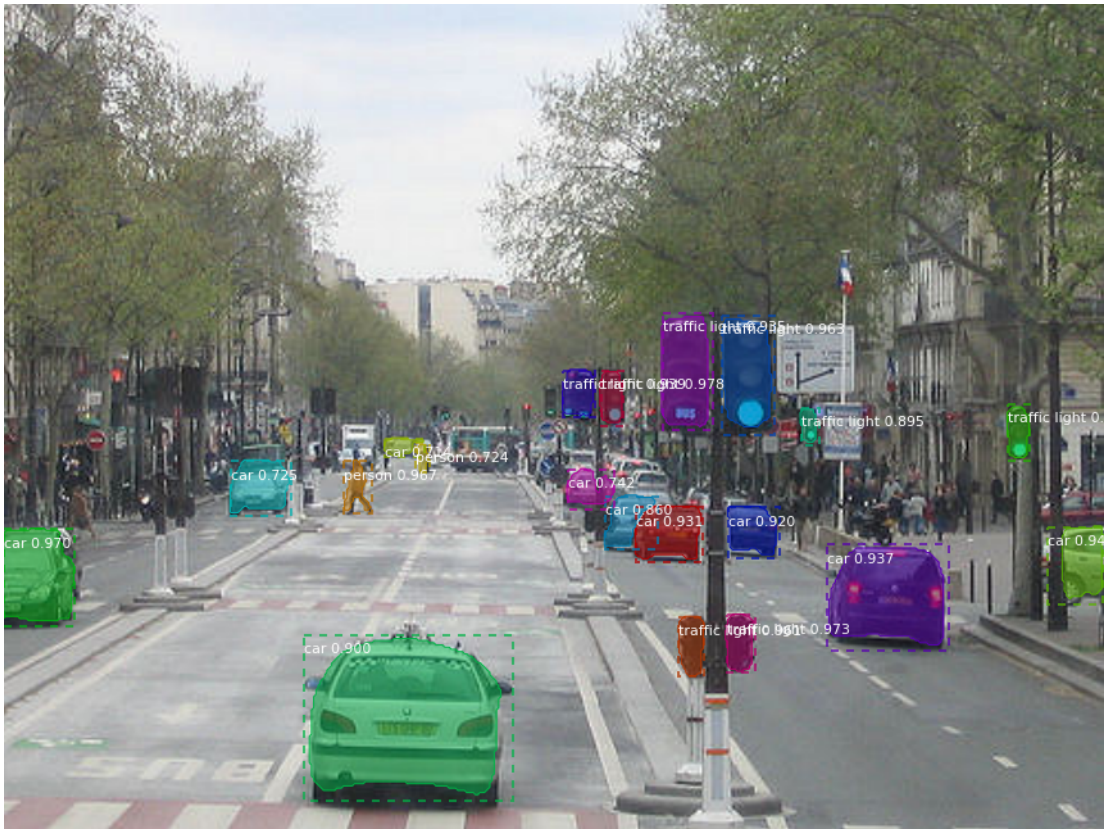
Počítačové vidění je konstrukce explicitního popisu fyzických objektů v obraze [3]. Tuto konstrukci je pak možné rozdělit na kolekci úloh zahrnujících zejména:

- *klasifikaci*, která se zabývá predikcí třídy nebo typu objektů v obraze. Vstupem klasifikace je obraz obsahující jeden objekt, pro který je vybrán jeden, či více různých návěští, reprezentujících příslušné třídy. [9]
- *detekci objektů*, která spočívá v nalezení přítomnosti objektů v obraze a označení jejich polohy *bounding boxem*. Vstupem detekce je obraz, který může obsahovat objekty a výstupem seznam poloh reprezentovaný souřadnicemi (např. souřadnice středového bodu a výška a šířka *bounding boxu*). [9]
- *segmentaci* (nebo také *instanční segmentaci* či *sémantickou segmentaci)*, která slouží k nalezení pixelů obrazu náležících danému objektu, jejím výstupem je tedy pixelová maska pro každý objekt. Segmentací tak lze přesně oddělit jednotlivé instance objektů i v nahuštěném prostoru, jako je například dav lidí, či kolona aut. [9]

Příklad výstupu jednotlivých úloh je možné vidět na obrázku 2.3.

V posledních letech se ukázaly být klíčem k efektivnímu řešení těchto úloh detektory, využívající různé formy a architektury konvolučních neuronových sítí. Jejich základem je páteřní síť sloužící ke zpracování vstupního obrazu a extrakci rysů k dalšímu zpracování. Tato extrakce je tak první fází celého procesu detekce. Páteřní síť je často i největší komponentou celého detektoru počtem vypočítávaných hodnot. [15]

Jeden detektor lze často navázat i na několik různých páteří a naopak jedna architektura páteře, může být využita pro různé detektory [19]. Jedním z příkladů ověřených řešení



Obrázek 2.3: Příklad vizualizace výstupu detektoru Mask R-CNN. Výstup objektové detekce je naznačen *bounding boxy* přerušovanými čarami. Výstup instanční segmentace je barevná plocha uvnitř *bounding boxu*. Výstupem klasifikace jsou návětší tříd u každé detekce (číslo značí hodnotu *confidence* reprezentující pravděpodobnost správnosti detekce). Převzato z [1].

pro extrakci příznaků (nebo také rysů) je síť ResNet. Její název vychází ze slov *Residual Network*, což je neuronová síť, která mimo běžná spojení za sebou jdoucích vrstev, využívá také zkratková či přeskakovací spojení umožňující části vstupu některé vrstvy při vyhodnocování obejít. Její návrh umožňuje využití v různých konfiguracích hloubky, například varianty sítě s 50, 101 či 152 vrstvami, kdy větší síť poskytuje přesnější výsledky. [17]

Jejím evolučním rozšířením je síť ResNeXt, která kromě zkratkových spojení rozděluje některé vrstvy na vícero cest pro paralelní průchod. Suma výstupů těchto paralelních průchodů je pak předána k dalšímu zpracování. ResNeXt vykazuje inkrementální zlepšení výsledků oproti ResNetu. [33]

Dalším příkladem páteře je CSPNet (Cross Stage Partial Network), která se zaměřuje na zlepšení procesu trénování a snížení výpočetní a paměťové náročnosti. Toho se snaží dosáhnout využitím dělení hloubky vstupu na část, co projde konvolučním blokem a část, která se v bloku nezpracovává a je následně připojena až k jeho výstupu. Díky posílení učících schopností by CSPNet měla dosahovat lepších výsledků než ResNet i ResNeXt při použití pro klasifikaci obrazu na datové sadě ImageNet. [32]

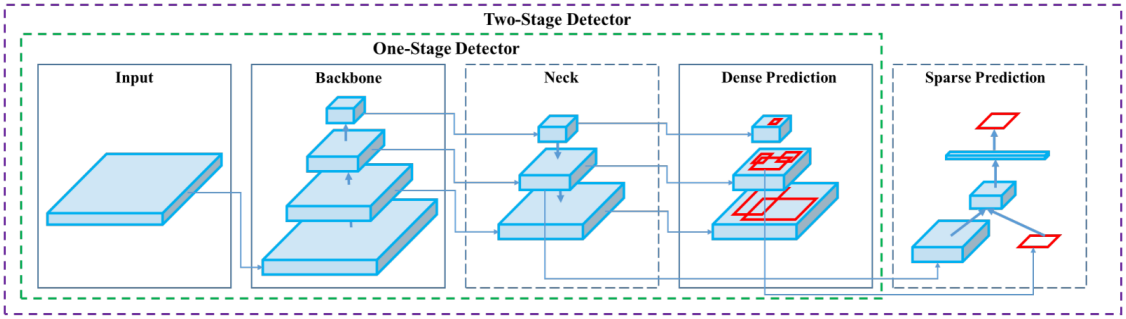
2.3.1 Architektury detekce

Po počáteční extrakci rysů se mohou různé detektory lišit přístupem navázaným na jejich páteř. Typicky je lze rozdělit dle počtu hlavních kroků mezi výstupem páteře a výsledky detekce. Dvě takto získané skupiny by se daly nazvat jako detektory jednoúrovňové a dvouúrovňové. [15]

Dvouúrovňové detektory jako například Faster R-CNN nebo Mask R-CNN v prvním kroku naleznou oblasti s pravděpodobným výskytem objektů pomocí Region Proposal Network (dále jen RPN) [30]. RPN je typicky zaměřena na jednoduchost a efektivitu pro odhadování navrhovaných *bounding boxů* [15]. Výstupem z ní je množina obdelníkových regionů a objektového skóre pro každý z nich [26].

Jednoúrovňové detektory jako například SSD nebo sítě z rodiny YOLO využívají přímo konvoluční síť k predikci *bounding boxů* v jediném kroku, bez navrhovaných regionů získaných z RPN. [15]. Vynechání kroku prvotního odhadování *bounding boxů* RPN, jim typicky umožňuje dosahovat větší běhové rychlosti detekce [24, 25, 5, 21].

V některých případech se může v architektuře detektorů objevit komponenta propojující výstup páteře nebo RPN se vstupem samotné detekce (částí též nazývanou jako hlava detektoru) označovanou jako krk či *neck* [15]. Příkladem této komponenty je hluboká neuronová síť Feature Pyramid Network (dále jen FPN) vycházející z architektury ResNet, která během vyhodnocování postupně snižuje rozlišení vstupu a rysy pro výstup získává kombinováním extrakcí z různých rozlišení [20].



Obrázek 2.4: Schéma porovnání architektury jednoúrovňových a dvouúrovňových detektorů. Jako páteř (*backbone*) může sloužit například RessNet, ResNeXt či CSPNet. Její výstup je dále zpracován krkem sítě (*neck*), jako například FPN. Pro jednoúrovňové detektory proces končí blokem *dense prediction* pro predikci objektů, například hlavou YOLOv3 či SSD. Pro detektory dvoúrovňové blok *dense prediction* představuje RPN, jejíž výstup je nakonec vyhodnocen hlavou jako Mask R-CNN či Faster R-CNN v bloku *sparse prediction*. Převzato z [5].

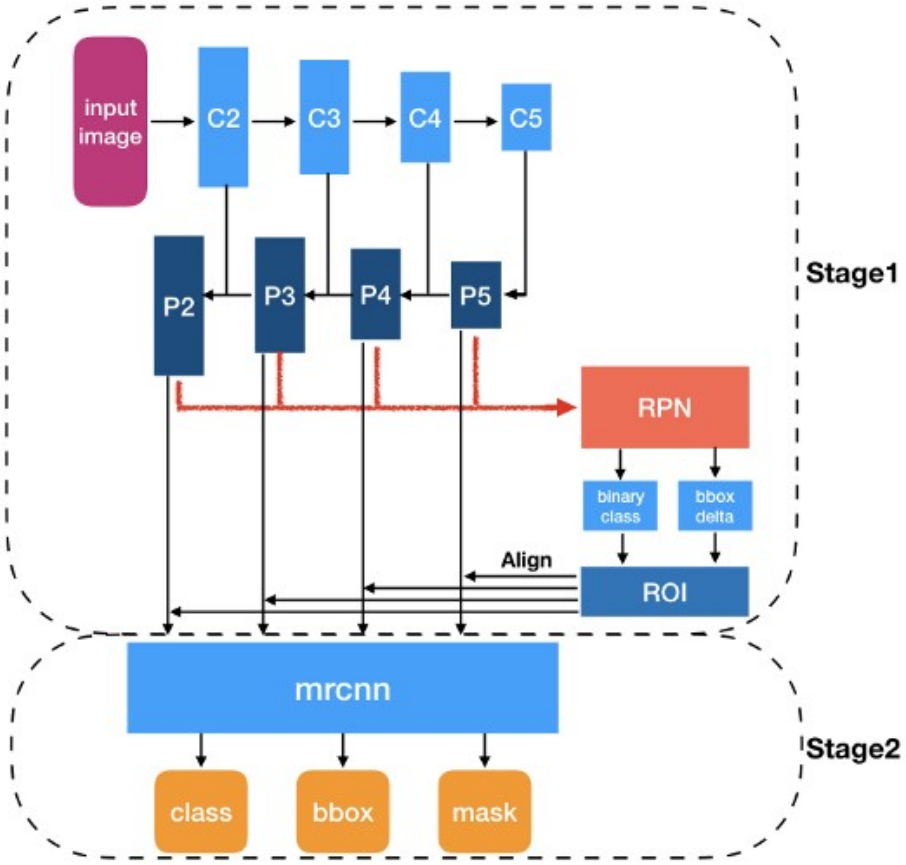
2.3.2 Detektory

Pro účely testování v této práci byl použit detektor YOLOv4 z rodiny YOLO. Síť může provádět detekci objektů a klasifikaci ve vysoké běhové rychlosti umožňující její užití pro detekci v reálném čase (nad 30 FPS). Detektor je postaven na páteři CSPDarknet53, která přímo vychází z architektury CSPNet, nad ni pak jako hlavu přidává síť YOLOv3. Detektor využívá různé techniky k vylepšení přesnosti detekce s minimálním či žádným dopadem na její výpočetní náročnost. Například augmentaci učících dat mozaikováním, kdy je na

vstup dodán obrázek vytvořený z výřezů 4 obrázků datové sady, za účelem zlepšení detekce objektů mimo jejich běžný kontext. [5]

Dalším detektorem testovaným v této práci je Mask R-CNN – dvouúrovňový detektor umožňující instanční segmentaci. Detektor může běžet v různých konfiguracích s výběrem páteře z variant ResNet-FPN, ResNet nebo ResNeXt. ResNet může sloužit jako páteř buďto s 50 nebo 101 vrstvami s extrakcí příznaků z poslední konvoluční vrstvy 4. úrovně sítě a 5. úrovní přidanou do hlavy detektoru. Varianta s FPN obsahuje i 5. vrstvu ResNet v páteři. Schéma architektury sítě je možné vidět na obrázku 2.5. [16]

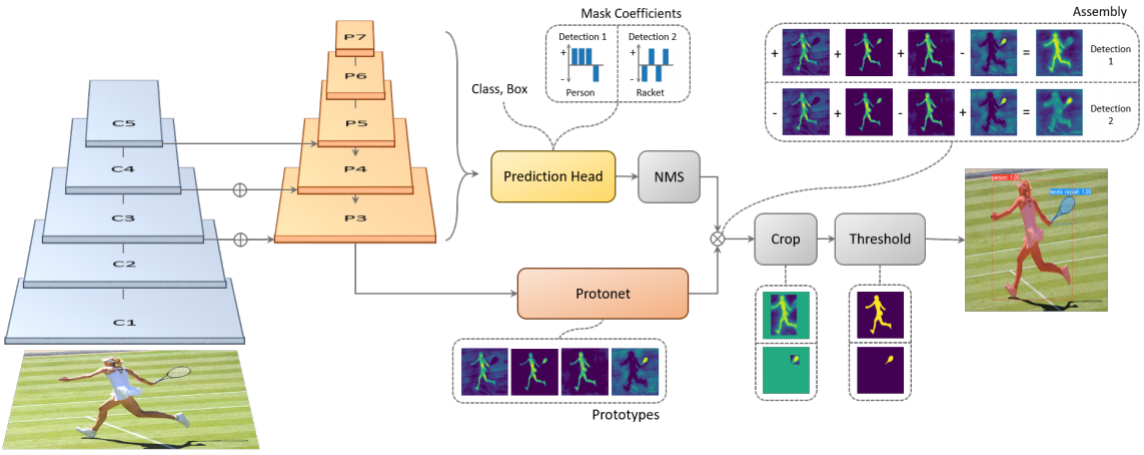
Mask R-CNN předčila své předchůdce svou dosahovanou přesností, nicméně by však měla nabízet jen nízkou běhovou rychlost v porovnání například s YOLOv4. [16]



Obrázek 2.5: Schéma architektury Mask R-CNN. V první úrovni (*Stage1*) jsou rysy z páteře a FPN (bloky *C2-5* a *P2-5*) předány RPN k nalezení regionů s pravděpodobným výskytem objektů (dále jen *RoI*) [16]. *RoIs* jsou pak namapovány zpět na aktivační mapy metodou *RoIAlign*, která minimalizuje ztrátu informace při převodu [16]. V druhé úrovni (*Stage2*) pak proběhne samotná detekce, klasifikace a segmentace. Převzato z [35].

Posledním detektorem testovaným v této práci je YOLACT++, jednoúrovňový detektor schopný provádět instanční segmentaci v reálném čase. Je postaven na páteři ResNet101, kterou však rozšiřuje o deformovatelné konvoluce, umožňující úpravu receptivních polí neuronů učením. Nad páteří je pak ještě připojena FPN. Instanční segmentace je realizována dvěma paralelními cestami – první sestává z FCN pro vytváření prototypových masek pro

celý vstup, druhá pak vytváří predikce vektoru maskových koeficientů. Pro každou instanci je nakonec zkonstruována maska lineární kombinací výstupu obou větví. Vizualizaci architektury je možné vidět na obrázku 2.6. [7]



Obrázek 2.6: Architektura detektoru YOLACT, ze kterého vychází YOLACT++. Modře jsou znázorněny vrstvy páteře propojené s FPN (oranžově), ze které vychází paralelně cesty pro vytváření prototypů masek a koeficientů [7]. Převzato z [7].

Z dalších významných detektorů lze zmínit například jednoúrovňový SSD (Single Shot Multi Box Detector) s páteří MobileNets, umožňující běh detekce s přesností blížící se dvoúrovňovým sítím jako Faster R-CNN, který předčil dosavadní sítě pro detekci v reálném čase (detektor YOLO, předchůdce zmiňovaného YOLOv4). Velkou výhodou je však jeho schopnost běhu na zařízeních s nízkým výpočetním výkonem, umožněná využitím mimo jiné hloubkově rozdělitelné konvoluce. Ta vstup nejprve konvoluje hloubkově po jednotlivých kanálech, výstupy všech kanálů jsou pak bodovou konvolucí spojeny do výsledné reprezentace. [21, 18]

Kapitola 3

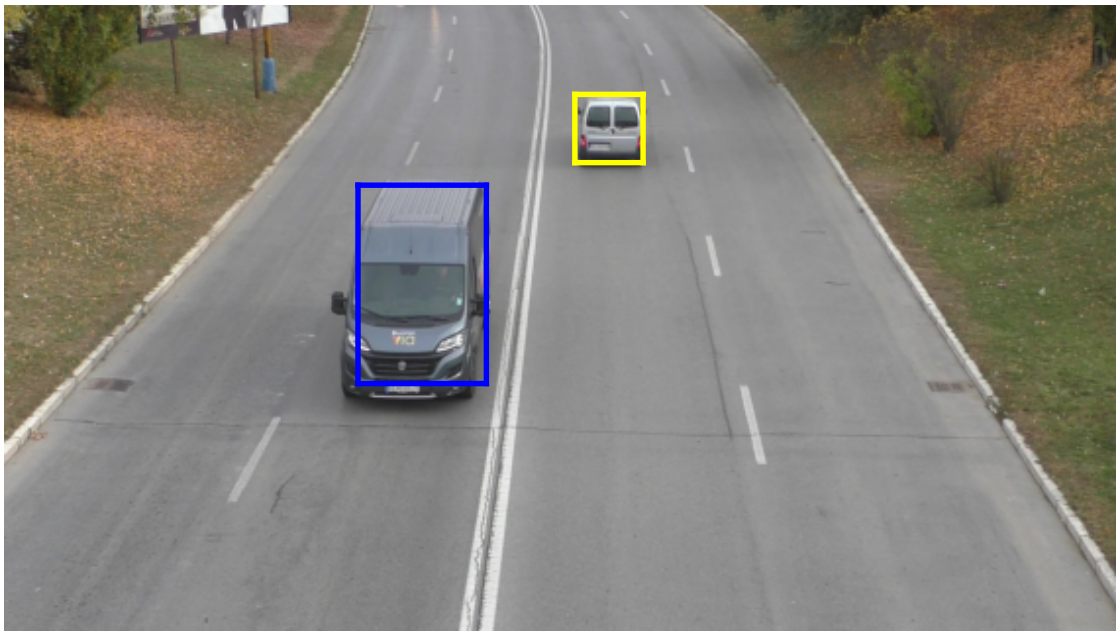
Datová sada

Pro účely učení, testování a vyhodnocování detektorů slouží datová sada skládající se z 114 002 anotovaných obrázků, zachycujících celkem 418 946 instancí dopravních prostředků různých tříd v běžném provozu na několika dopravních komunikacích. Záběry pochází ze 4 různých lokací s různě umístěnou kamerou vůči komunikaci. Všechny obrázky jsou barevné a mají velikost 1920×1080 pixelů. Na obrázcích 3.1, 3.2, 3.3, 3.4, 3.5 a 3.6 jsou příklady snímků z každé lokace s barevně odlišenými *bounding boxy* detekcí pro různé třídy.

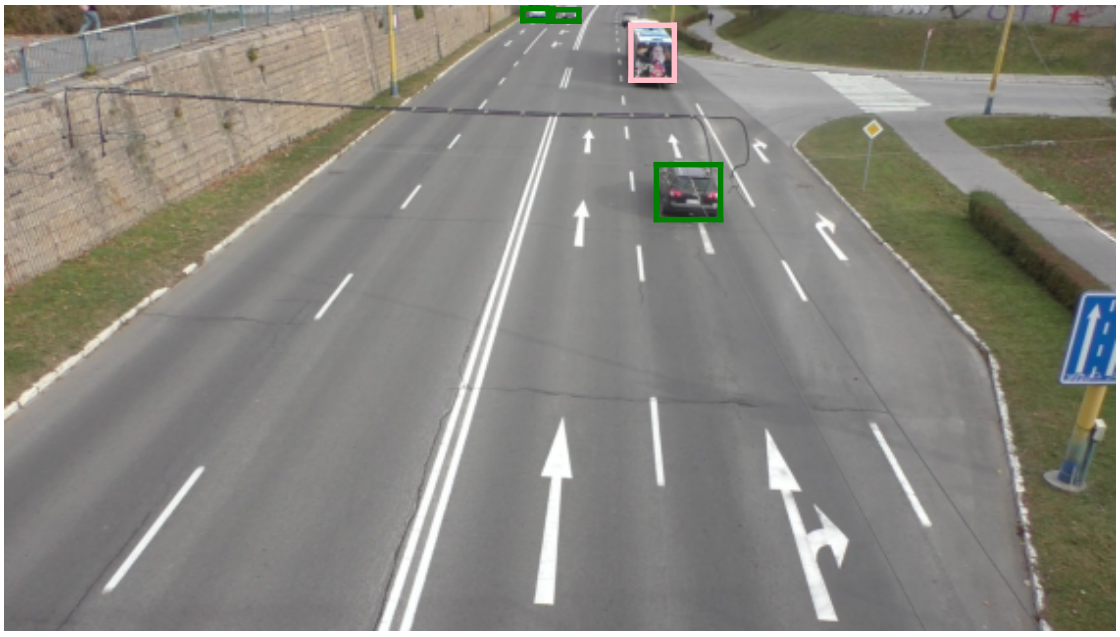
Příprava datové sady spočívala v manuální kontrole předpřipravených detekcí. Ty vznikly z výstupu detektoru Mask R-CNN předtrénovaném na *COCO datasetu*, vytvořením *tracků* jednotlivých instancí využitím *Simple Online Realtime Tracker* pro určení jejich tříd [10, 4]. Hlavní objem práce spočíval v převodu těchto tracků na vhodný formát a opravě jejich nedostatků – zejména úpravě *bounding boxů*, změn typů vozidel, doplnění neexistujících detekcí a případně označení detekcí jako obtížných. Jako časově nejnáročnější se ukázalo označování obtížných detekcí a opravy typů vozidel z důvodu nekonzistence zdrojových dat (vlivem využití *trackování* pro určování typů vozidel se případná chyba při výběru přenese na celou existenci daného objektu).

K průchodu dat a kontrole anotačních souborů byl použit volně dostupný nástroj *labelImg*. Umožňující rychlý průchod při vizuální kontrole a možnost jednoduchého doplnění nedostatků. Jako hlavní z možných formátů výstupu nástroje byl zvolen *PascalVOC* pro jeho jednoduchou strojovou čitelnost [12]. Z tohoto základního formátu pak byly anotace převáděny, vždy dle potřeb konkrétních detektorů či vyhodnocení. [31]

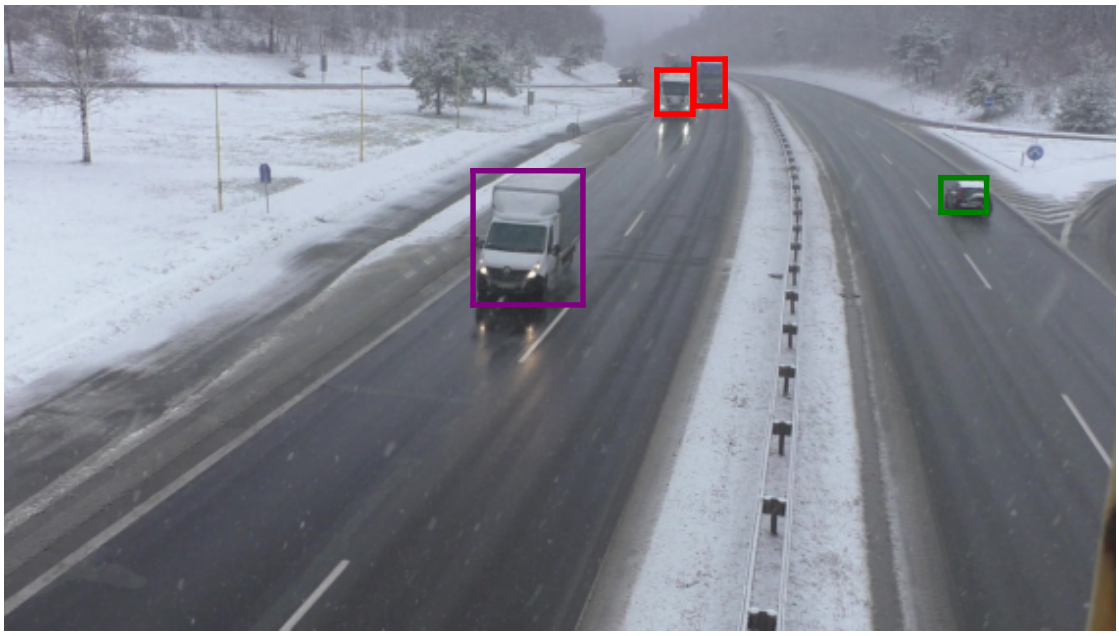
Ke každému obrázku je přiložen anotační soubor obsahující zejména informace o detekovatelných objektech v daném obrázku. Záznam pro daný objekt obsahuje vždy souřadnice jeho minimálního ohraničujícího boxu a název třídy do které je daný objekt zařazen.



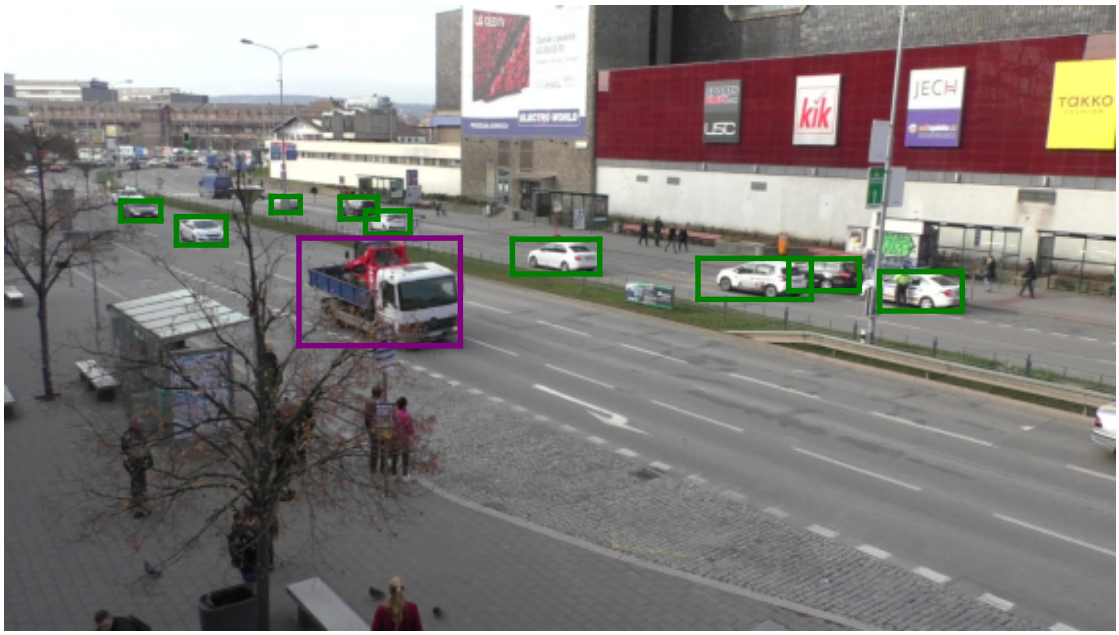
Obrázek 3.1: Příklad obrázku z lokace most Jazdiareň (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace).



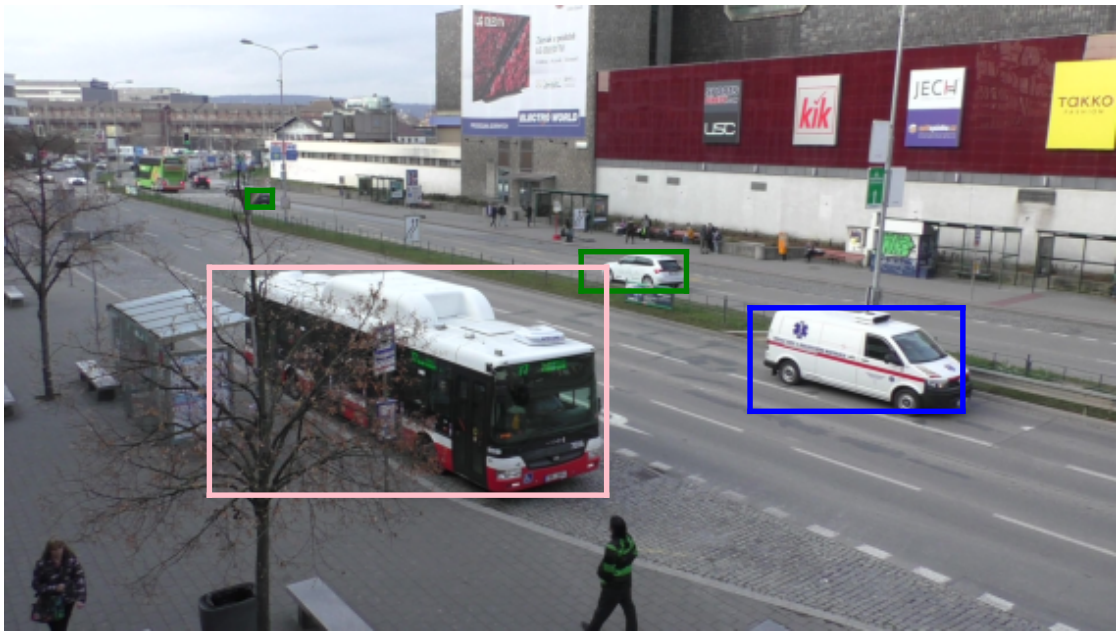
Obrázek 3.2: Příklad obrázku z lokace most Popradská (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace).



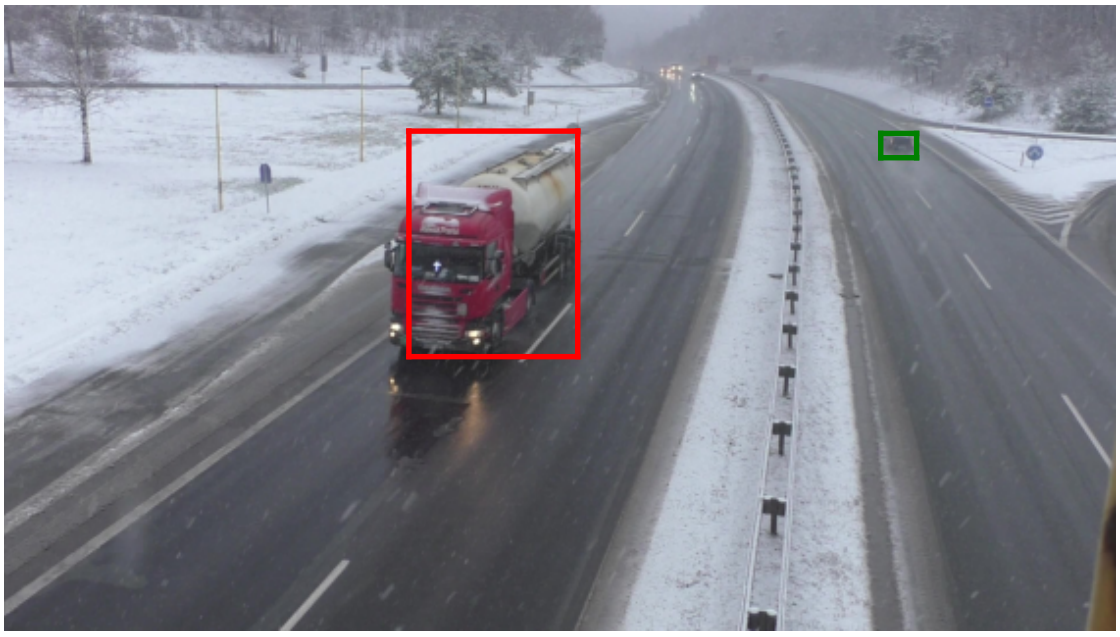
Obrázek 3.3: Příklad obrázku z lokace Zelený Dvor (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace).



Obrázek 3.4: Příklad obrázku z lokace Vaňkovka (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace).



Obrázek 3.5: Příklad obrázku z lokace Vaňkovka (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace).



Obrázek 3.6: Příklad obrázku z lokace Zelený Dvor (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace).

3.1 Klasifikace detekcí

Každá detekce v datové sadě je zařazena právě do jedné třídy, reprezentující typ daného dopravního prostředku. Detekce jsou klasifikovány celkem do 6 tříd:

- Osobní automobil (*car* viz obrázek 3.7a) – dvoustopé motorové silniční vozidlo, určené zejména pro přepravu osob.
- Víceúčelové vozidlo – MPV (*minivan* viz obrázek 3.7c) – Větší osobní automobil než vozidla typu sedan, hatchback a kombi.
- Dodávkový automobil (*van* viz obrázek 3.7b) – dodávka jak v provedení pro přepravu nákladu, tak osob.
- Autobus (*bus* viz obrázek 3.8a) – dálkový autobus nebo autobus městské hromadné dopravy.
- Menší nákladní vozidlo (*minitruck* viz obrázek 3.8b) – dvounápravové nákladní vozidlo s jakýmkoliv provedením nákladního prostoru (skříňové, plachtové, sklápěčkové, ad.).
- Nákladní vozidlo (*truck* viz obrázek 3.7d) – Nákladní vozidlo s jakýmkoliv provedením nákladního prostoru, které má více než 2 nápravy a nákladní vozidla s návěsem (kamiony/tahače).

Z rozložení četnosti jednotlivých tříd zobrazeného v tabulce 3.1 je patrné, že datová sada je třídně nevyvážená, zvláště pak v poměru třídy *car* obsahující osobní vozidla vůči ostatním třídám. Tato nevyváženost je zapříčeněna skladbou vozidel v běžném provozu na daných komunikacích a její odstranění, například podvzorkováním obrázků s detekcemi majoritních tříd, by vyústilo ve ztrátu značné části získaných dat. I nejméně početná třída (*truck*), je však zastoupena v počtu 9 810 instancí, což se ukázalo, jako dostatečné množství pro účely učení a granularity vyhodnocení.

Tabulka 3.1: Rozložení četností tříd v datové sadě

Třída	Absolutní četnost	Přibližná relativní četnost
car	300 383	71,7%
bus	53 239	12,7%
van	25 566	6,1%
minivan	18 118	4,3%
minitruck	11 830	2,8%
truck	9 810	2,3%

(a) *car*

(b) *van*

(c) *minivan*

(d) *truck*

Obrázek 3.7: Příklady vozidel tříd *car*, *minivan*, *van* a *truck*.

(a) *bus*

(b) *minitruck*

Obrázek 3.8: Příklady vozidel tříd *bus* a *minitruck*.

3.2 Rozdělení dat

Datová sada je rozdělena do tří částí dle jejich účelu. Jako první jsou odděleny obrázky s označenými obtížnými detekcemi, zbytek dat je pak rozdělen na část pro učení (85% zbylé části) a pro testování (15% zbylé části). Z důvodu maximalizace množství dat pro učení, nejsou oddělena žádná data do validační části.

Učení

Největší část datové sady (přibližně 82% všech dat) je vyhrazena pro účely učení. Sestává z 93 452 obrázků ze všech lokací, obsahujících celkem 350 076 instancí. V tabulce 3.2 je zobrazeno rozložení četnosti tříd detekcí v této části, které kopíruje rozložení celé sady.

Tabulka 3.2: Rozložení četností tříd v části pro učení

Třída	Absolutní četnost	Přibližná relativní četnost
car	250 733	71,6%
bus	44 647	12,8%
van	21 456	6,1%
minivan	15 169	4,3%
minitruck	9 947	2,8%
truck	8 124	2,3%

Obecné testování

Pro testování je vyhrazeno 16 490 obrázků (přibližně 14,5% všech dat). Tato část datové sady byla vybrána pseudonáhodným navzorkováním 15% dat (po odstranění části pro experimenty na obtížných detekcích) s důrazem na zachování relativních četností počtů obrázků z různých lokací.

Část je určena k inferenčnímu běhu detektorů za účelem analýzy kvality detekce. Z tohoto důvodu je důležité její oddělení od dat využitých k učení sítí, aby nedošlo k znevalidnění výsledků inference.

Testování obtížných detekcí

Součástí datasetu je i 4 060 obrázků obsahujících označené obtížné detekce (obtížné detekce se nachází i v obrázcích napříč celou datovou sadou, pouze v této části jsou však explicitně označené). Jako obtížné detekce jsou označeny objekty, které nejsou viditelné jako celek, lze ale však stále určit, že se jedná o daný dopravní prostředek. Například jde o objekty částečně překryté nějakou překážkou, nebo objekty opouštějící či přijíždějící do záběru kamery.

Tato část datové sady umožňuje samostatné vyhodnocování těchto obtížných detekcí pro hlubší analýzu kvality detekce. Porovnání chování detektorů v krajních případech může při vyhodnocení ukázat zajímavé rozdíly v jejich kvalitě.

Nedostatkem této části je její velikost z důvodu náročnosti označování a kontroly. Toto je patrné z tabulky 3.3, kde je vidět, že relativní četnosti tříd přibližně kopírují hodnoty celé sady. Z absolutních četností však lze vyčíst nízké zastoupení některých minoritních tříd (zejména *van*, *truck*, *minivan* a *minitruck*), což může vést k nepřesnosti a nízké granularitě měřených hodnot. Příklad snímku s dvěmi obtížnými detekcemi lze vidět na obrázku 3.9.

Tabulka 3.3: Rozložení četností tříd v části s označenými obtížnými detekcemi

Třída	Absolutní četnost	Přibližná relativní četnost
car	2 963	68,7%
bus	691	16,0%
van	218	5,1%
truck	206	4,8%
minivan	147	3,4%
minitruck	90	2,1%

Obrázek 3.9: Příklad obrázku s označenými obtížnými detekcemi (*bounding boxy* jsou přidány k obrázku pro ilustraci a nejsou součástí zdrojové reprezentace). Objekt třídy *car* je vyznačen zeleně, třída *van* je vyznačena modře. Obě detekce jsou v tomto případě označeny jako obtížné.

Kapitola 4

Experimenty

Experimenty jsou zaměřeny na ověření schopnosti detektorů lokalizovat objekt pomocí *bounding boxu* a objekt klasifikovat, netestují však schopnost instanční segmentace a predikce masek. K dosažení kvalitních výsledků bylo třeba zvolené detektory před testováním vhodně natrénovat.

Učení a následné testování bylo provedeno na třech různých detektorech s různými charakteristikami a architekturami. Konkrétně sítě Mask R-CNN, YOLACT++ a YOLOv4. Pro tyto účely byly použity dostupné implementace všech tří detektorů [1, 2, 6].

4.1 Učení

Učení probíhalo na celé části dat určené pro trénování (viz sekce 3.2), s průběžným pozorováním dosahovaných hodnot ztrátové funkce pro kontrolu procesu.

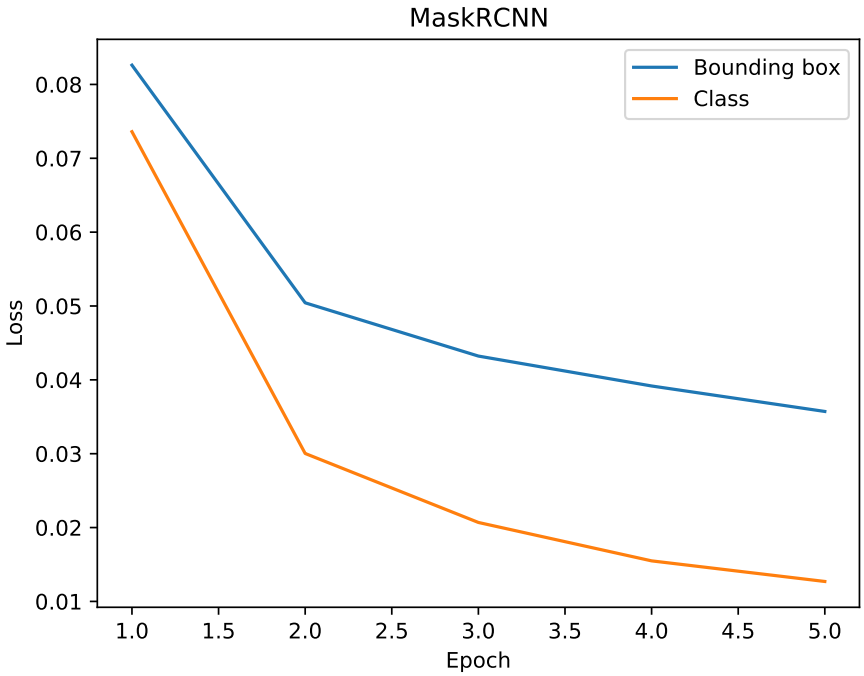
Učení detektoru Mask R-CNN bylo provedeno na celkem 5 epoch a na obrázku 4.1 je vidět průběh hodnot chybové funkce během učení, které mají předpokládaný charakter – počáteční relativně vysoké úrovně s rychle klesající tendencí, která se postupně zpomaluje. Experimenty i učení jsou prováděny a vyhodnocovány pouze pro detekci objektů formou *bounding boxu* a klasifikace, nikoliv pro predikci masek, z tohoto důvodu jsou masky pro učení uměle přidány do datové sady jako celá plocha *bounding boxu*.

Průběh učení detektoru YOLOv4 se mírně odlišuje od očekávání z důvodu jeho kratšího trvání. Graf zanesených hodnot je vidět na obrázku 4.2. Pro zlepšení výkonnosti modelu byla při učení použita vestavěná funkce mozaikování učících dat. Tato skutečnost mohla být jedním z vlivů způsobujícím zastavení přínosu učení a jeho předčasné ukončení, detektor však při detekci vykazuje výkonnost srovnatelnou s ostatními testovanými sítěmi.

Trénování detektoru YOLACT++ proběhlo na celkem 5 epoch, s průběhem dosahovaných hodnot *loss* dle grafu na obrázku 4.3. Ten lze charakterizovat velmi ostrým přiblížením se ke konvergenci a následně jen pomalým (avšak stálým) trendem dalšího klesáním. Stejně jako v případě Mask R-CNN jsou masky pouze uměle nahrazeny celou plochou *bounding boxu*.

4.2 Vyhodnocovací metriky

V této sekci je primárně čerpáno z [34]. Pro interpretaci výsledků detekce a přesné měření její kvality je třeba zavést vhodné nástroje a postupy, umožňující přímé srovnání testovaných detektorů. Ty se většinou zakládají na kvantifikaci nálezů detekce jako pozitivní či



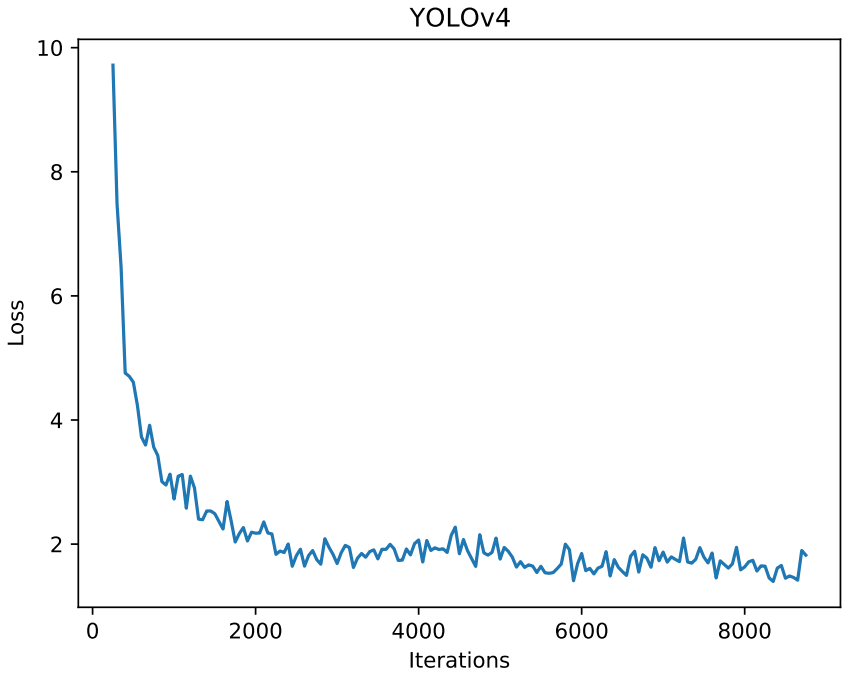
Obrázek 4.1: Hodnoty chybové funkce na konci každé epochy při učení detektoru Mask R-CNN. Ztráta pro lokalizaci *bounding boxů* je zobrazena modře, ztráta pro klasifikaci detekovaného vozidla je zobrazena oranžově.

negativní. Pravý pozitivní nález (dále jen TP) znamená, že detektor úspěšně nalezl objekt v daném prostoru, nález falešný pozitivní (dále jen FP) vyjařuje, že detektor nalezl v daném prostoru objekt, který se tam ve skutečnosti nenachází. Případ skutečného objektu, který však detektor nenalezl a neoznačil, se počítá jako nález falešný negativní (dále jen FN). Pravý negativní nález popisuje případ neoznačení ve skutečnosti neexistujícího objektu, tato veličina má však pro účely vyhodnocení detekce menší význam.

Základními koncepty pro spočtení TP, FP a FN je skóre spolehlivosti (dále *confidence*), které vyjadřuje pravděpodobnost výskytu objektu v *bounding boxu* a překrytí detekovaného *bounding boxu* s *bounding boxem* reálného objektu. Toto překrytí (dále jen IoU) lze vyjádřit jako poměr obsahu průniku a obsahu sjednocení detekovaného boxu B_d a reálného boxu B_r

$$IoU = \frac{area(B_d \cap B_r)}{area(B_d \cup B_r)}.$$

Případy se pak posuzují průchodem detekovaných *bounding boxů* s *confidence* převyšující daný práh a porovnáním shodnosti IoU detekce a reálného *bounding boxu* s daným globálním prahem IoU. Pokud shoda IoU detekce a reálného *bounding boxu* práh přesahuje, počítá se jako TP, v opačném případě pak jako FP. Pro hodnocení detekcí s vícero třídami se berou v potaz pouze reálné *bounding boxy* se shodnou třídou jako detekce. Počet FN se dá pro získání některých metrik vhodně nahradit, není ho tak třeba vypočítávat samostatně.



Obrázek 4.2: Hodnoty chybové funkce při učení detektoru YOLOv4. Hodnoty na ose X představují počet iterací s velikostí *batche* 32 obrázků, konec první a druhé epochy je tak po iteracích 2 921 a 5 842. Pro účely lepší vizualizace není zobrazeno prvních 250 hodnot *loss*, neúměrně vysokých vůči ostatním hodnotám při přibližování se konvergenci. Hodnoty jsou pravidelně navzorkovány s krokem o velikosti 50 hodnot pro menší přeplněnost grafu.

Přesnost a citlivost

Přesnost (dále *precision*) ukazuje validitu výsledků detekce, tedy jak často je detekovaný objekt *TP*, pokud detektor danou oblast označil. Lze ji vyjádřit jako podíl počtu *TP* a jejich součtu s počtem *FP*

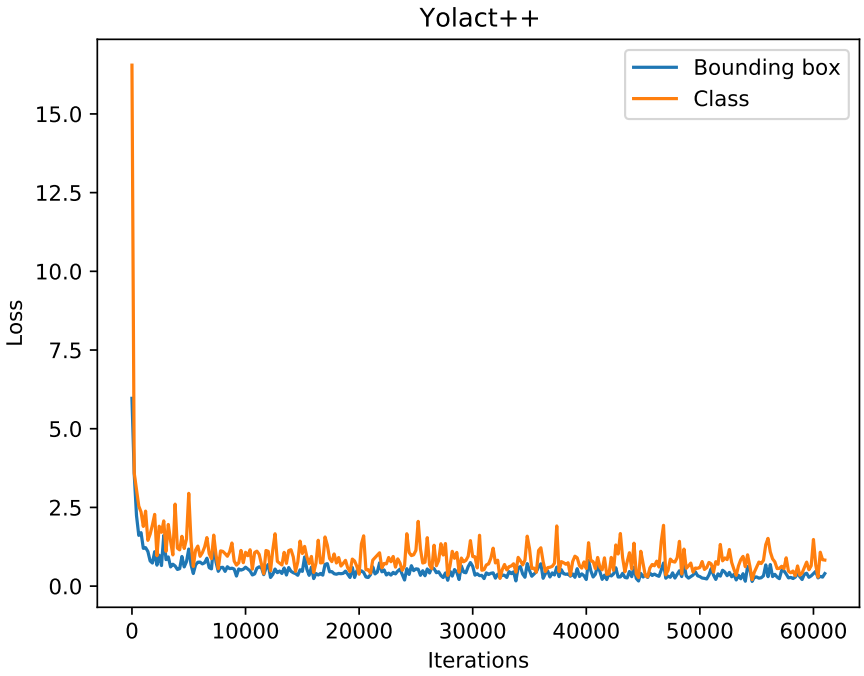
$$precision = \frac{TP}{TP + FP}.$$

Hodnota citlivosti (dále *recall*) popisuje kompletnost výsledků detekce. Vyjadřuje kolik objektů detektor zachytil oproti skutečnému počtu objektů na vstupu. Lze ho vajádřit jako podíl počtu pravých pozitivních nálezů *TP* a jejich součtu s počtem falešných negaticnách nálezů *FN*

$$recall = \frac{TP}{TP + FN}.$$

Součet *TP* a *FN* je maximální počet objektů, které může detektor validně nalézt, tedy jinými slovy – celkový počet reálných objektů. Ten se dá snadno získat z anotovaných dat a není tak třeba spočítat *FN*.

Spíše než tyto samostatné hodnoty je pro vyhodnocení důležitější jejich srovnání na grafu s *precision* jako funkcí *recallu*, vypočítaných pro zvyšující se práh *confidence* a pevný práh *IoU*. Například jako na obrázku 4.4, kde je vidět možné chování těchto dvou hodnot



Obrázek 4.3: Hodnoty *lossu* při učení detektoru YOLACT++. Modře je znázorněn *loss* pro lokalizaci *bounding boxů*, oranžově pak *loss* pro klasifikaci. Osa X značí počet iterací pro velikost *batche* 8, délka jedné epochy je tak 11 682 iterací. Hodnoty jsou navzorkovány z učících dat s krokem mezi vzorky o velikosti 200, pro větší zřetelnost vizualizace.

– snižující se *precision* pro zvyšující se *recall*. Tento efekt je způsoben měnícím se prahem *confidence*, kdy je pro vysokou jistotu detekcí zachycována i vysoká *precision*. S jeho klesáním *precision* upadá, avšak narůstá *recall*, protože se více započítávají nejisté detekce a detektor má širší záběr objektů na vstupu.

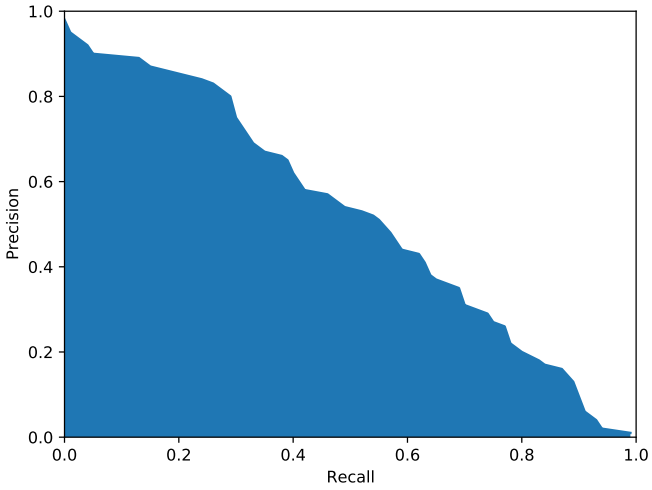
Pro účely vyhodnocení kvality detektorů v této práci je *precision* a *recall* vypočítáván na 101 různých výškách prahu *confidence*, od $0,00$ (včetně) po $1,00$ (včetně) s krokem mezi prahy $0,01$. Tato varianta by měla zajistit dostatečnou granularitu výsledků s příznivými podmínkami pro realizaci vyhodnocovacích výpočtů.

Průměrná přesnost

Průměrná přesnost (dále jen AP) je jednou z hlavních metrik pro vyhodnocení výsledků detekce. Vychází z hodnot interpolované *precision* pro různé hodnoty *recallu*. Ta se interpoluje jako $p_{interp}(r)$ na úrovni *recallu* r a je definována jako nejvyšší *precision* na jakékoliv úrovni *recallu* r' pro $r' \geq r$. Lze ji tedy vyjádřit jako

$$p_{interp}(r) = \max_{r' \geq r} p(r').$$

Úrovně recallu pro výpočet průměrné přesnosti lze zvolit různě. Jedním příkladem je volba 11 hodnot v rovnoměrných rozestupech – tedy 0,0; 0,1; 0,2 až 1,0 nebo, i pro tuto práci zvolený, novější standard s výpočtem hodnot na všech dostupných unikátních úrovních *recallu* pro větší přesnost výsledků.



Obrázek 4.4: Příklad grafu znázorňujícího *precision* jako funkci *recallu*. Modře znázorněná plocha pod touto křivkou by po interpolaci *precision* reprezentovala AP.

AP tedy lze vyjádřit jako plochu pod interpolovanou křivkou *precision* a *recallu* p_{interp} pro n vzestupně seřazených úrovní recallu r_i [34]

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}).$$

Příklad takové plochy je znázorněn modře na obrázku 4.4 (před interpolací *precision*).

Střední průměrná přesnost

Hodnota střední průměrne přesnosti (dále jen mAP) je pravděpodobně jednou z hlavních metrik pro porovnávání kvality různých vícetřídních detektorů. Je dána hodnotou průměru AP detekce pro každou třídu. Pro detektor klasifikujících K různých tříd, s průměrnou přesností detekce i-té třídy AP_i, lze tedy mAP vyjádřit jako

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K}.$$

Pro účely vyhodnocení jsou v této práci využity některé varianty mAP definované pro detekci na datové sadě *COCO - Common Objects in Context* [10]. Konkrétně pak metriky:

- mAP vypočítaná pro práh IoU=0,50 (dále jen $mAP^{IoU=.50}$) – tato hranice IoU je také stanovena pro vyhodnocování detekce *Pascal VOC Challenge* [12].
- mAP vypočítaná pro práh IoU=0,75 (dále jen $mAP^{IoU=.75}$) – striktní metrika vyžadující větší přesnost detekovaných *bounding boxů*. Může být vhodnější než $mAP^{IoU=.50}$ pro porovnání přesných detektorů.
- mAP průměrovaná pro 10 různých prahů IoU=(0.50, 0.55, 0.60, ..., 0.95) (dále jen $mAP^{IoU=.50:.05:.95}$) – primární metrika *COCO Challenge*. Poskytuje vyhodnocení rozsáhlejšího množství dat než $mAP^{IoU=.50}$ a $mAP^{IoU=.75}$.

Pro příklad dalšího možného vyhodnocování detektorů, pak lze uvést třeba další metriky definovány pro *COCO Challenge*, jako mAP^{small}, mAP^{medium} a mAP^{large} rozdělující detekce dle velikosti v obraze [12].

Rychlost

Rychlost detektoru je měřena ve snímcích za sekundu (dále jen *FPS*), tedy na kolika snímcích se stihne provést detekce za jednu sekundu času. Její výpočet vychází z času detekce jednoho snímku. Do času detekce se nepočítá čas potřebný pro získání dat obrázku z uložiště ani jeho převedení do formy vyžadované detektorem, za účelem eliminace vlivu těchto faktorů na výsledky. Nevýhodou tohoto přístupu je, že naměřené hodnoty nemusí odpovídat reálné rychlosti běhu ve skutečném prostředí.

Matice záměn

Matice záměn (dále *confusion matrix*) slouží pro rozbor výsledků klasifikačních úloh. Pro účely této práce je využita varianta pro vícetřídní klasifikaci zobrazující jednotlivé typy reálných detekcí oproti typům z výstupu klasifikace detektoru. Užití *confusion matrix* je zaměřeno na vyhodnocování obtížných detekcí (viz sekce 3.2) pro ověření kvality klasifikace v případech s menším informačním vstupem detektoru. Ilustračním příkladem *confusion matrix* je tabulka 4.1, která je strukturově podobná jako *confusion matrix* využité pro hodnocení vybraných detektorů v této práci. [22]

Tabulka 4.1: Příklad jednoduché *confusion matrix* pro vícetřídní klasifikaci (třídy *car*, *bus* a *truck*). Sloupce (označené jako *true*) reprezentují reálné třídy detekcí, řádky (označené jako *predicted*) reprezentují třídu na výstupu klasifikace. [22]

	true car	*true bus*	*true truck*
predicted car	48	8	4
predicted bus	2	28	14
predicted truck	3	16	33

Po zkonstruování *confusion matrix* z ní lze extrahovat hodnoty *TP*, *TN*, *FP* a *FN* pro klasifikaci. Počet *TP* pro danou třídu je v buňce jejího sloupce a řádku, hodnotu *FP* pak lze spočíst ze zbývajících sloupců pro řádek dané třídy a *FN* ze zbývajích řádků pro sloupec dané třídy. *TN* pro danou třídu jsou všechny hodnoty mimo její sloupec a řádek. [22]

Z těchto hodnot lze vypočítat *precision* a *recall* pro každou třídu (pouze však pro tuto klasifikační úlohu) [22]. Dále je pak možné vyhodnotit *F1–score*, tedy harmonický průměr *precision* a *recallu*, avšak pro klasifikační úlohu s nevyváženou četností tříd (viz tabulka 3.3) je vhodnější využít Matthewsův korelační koeficient (dále jen *MCC*), který bere v potaz i vzájemý poměr jednotlivých případů detekce [14]. Ten lze pro vícetřídní problém vyjádřit jako

$$MCC = \frac{cs - \vec{t} \cdot \vec{p}}{\sqrt{s^2 - \vec{p} \cdot \vec{p}}\sqrt{s^2 - \vec{p} \cdot \vec{p}}},$$

pro $\vec{t} = (t_1, ..., t_k)$ a $\vec{p} = (p_1, ..., p_k)$, kde t_k je počet reálných objektů třídy k, p_k je počet predikcí třídy k, c je celkový počet *TP* klasifikace a s je celkový počet vzorků (součet reálných i predikovaných instancí) [14]. Nejvyšší a nejlepší možná hodnota *MCC* pro vícetřídní

klasifikaci je 1. Hodnoty blízké 0 naznačují, že se úspěšnost klasifikace blíží náhodnému tipování [14].

4.3 Výsledky

Detektory jsou testovány ve dvou kategoriích – obecná detekce na datech odrážejících běžný provoz na pozemních komunikacích a obtížné detekce, pro které se vyhodnocují pouze predikce obtížně detekovatelných objektů (částečně překrytých či vyjíždějích z a přijíždějích do záběru kamery).

4.3.1 Naměřené hodnoty

Obecné testování

Pro získání hodnot ke komplexnímu vyhodnocení běhu detektorů byly testovány spuštěním detekce vždy na celé části datové sady určené pro obecné testování (viz. sekce 3.2). Hlavní výstup každého detektoru (*bounding boxy* s jejich *confidence* a predikovanou třídou) byl uchován a následně vyhodnocen skripty nezávisle na použité implementaci detektoru.

Obecné výsledky dosažené mAP lze vidět v tabulce 4.2, ze které je patrná relativní vyrovnanost výkonnosti detektorů v tomto ohledu. Největší rozdíl v přesnosti dosažené různými detektory v jedné kategorii je pouze 6,1% a to pro $mAP^{IoU=.75}$. Vzájemné výsledky v jednotlivých kategoriích jsou dle očekávání klesající s rostoucí náročností na překryv IoU – všechny sítě dosáhly nejvyšší přesnosti v kategorii $mAP^{IoU=.50}$, nižší v kategorii $mAP^{IoU=.75}$ a nejnižší pro nejpřísnější metriku v tomto ohledu $mAP^{IoU=.50:.05:.95}$.

Tabulka 4.2: Detektory dosažená $mAP^{IoU=.50:.05:.95}$, $mAP^{IoU=.50}$ a $mAP^{IoU=.75}$ na testovací části datové sady (viz sekce 3.2) při běhové rychlosti *FPS*. Seřazeno dle výsledků primární metriky $mAP^{IoU=.50:.05:.95}$.

Detektor	$mAP^{IoU=.50:.05:.95}$	$mAP^{IoU=.50}$	$mAP^{IoU=.75}$	*FPS*
Mask R-CNN	61,1%	79,0%	72,4%	3,44
YOLACT++	58,4%	82,4%	71,8%	4,47
YOLOv4	56,1%	76,9%	66,3%	105,50

Testování obtížných detekcí

Pro vyhodnocení detekce na obtížných případech byly sítě testovány spuštěním vždy na celé části datové sady s označenými obtížnými detekcemi (viz. sekce 3.2). Stejně jako v případě obecného testování byl výstup v podobě *bounding boxů* s *confidence* a tříd vyhodnocen nezávisle na implementaci detektoru. Oproti obecnému testování byly však během vyhodnocení zcela ignorovány výsledky pro detekce bez obtížného příznaku. Uvedené hodnoty jsou tak zcela nezávislé na běžných detekcích (které se nevyhnutelně vyskytují na některých vyhodnocovaných obrázcích).

Dosažené hodnoty jsou zobrazeny v tabulce 4.3. Vlivem záměrné obtížnosti detekce jsou naměřené hodnoty přesností $mAP^{IoU=.50:.05:.95}$, $mAP^{IoU=.50}$ a $mAP^{IoU=.75}$ pro všechny

detektory výrazně nižší než jejich protějšky z obecného testování v tabulce 4.2, je však třeba zmínit, že všechny sítě dosáhli alespoň nějaké měřitelné úrovně a potvrdili schopnost detekce pro tyto případy. Hodnoty mAP zachovávají očekávaný klesající trend pro stoupající striktnost IoU. Oproti obecnému testování lze pozorovat mnohem větší míru variability hodnot, s rozpětím až 32,3% mezi prvním a posledním detektorem pro $mAP^{IoU=.50}$ a $mAP^{IoU=.75}$. Narozdíl od obecného testování je pro všechny kategorie mAP konstantní pořadí detektorů s většími vzájemnými rozestupy (avšak relativně pravidelnými stejně jako v tabulce 4.2).

Tabulka 4.3: Detektory dosažená $mAP^{IoU=.50:.05:.95}$, $mAP^{IoU=.50}$ a $mAP^{IoU=.75}$ na části datové sady pro testování obtížných detekcí (viz sekce 3.2) při běhové rychlosti FPS. Seřazeno dle výsledků primární metriky $mAP^{IoU=.50:.05:.95}$.

Detektor	$mAP^{IoU=.50:.05:.95}$	$mAP^{IoU=.50}$	$mAP^{IoU=.75}$	FPS
YOLACT++	46,8%	71,1%	59,2%	7,44
YOLOv4	33,6%	51,0%	41,0%	48,88
Mask R-CNN	26,9%	38,8%	26,9%	3,23

Mimo mAP je detekce také vyhodnocena ve směru samotné klasifikace, pro kontrolu, zda-li detektory správně predikují typ vozidel speciálně v obtížných případech, kdy na vstupní reprezentaci mohou chybět odlišovací rysy konkrétních tříd. Toto vyhodnocení je realizováno vytvořením matice záměn pro každý detektor s prahy IoU na úrovni 0,75 a *confidence* na 0,5. Z té jsou pak, kromě *precision* a *recallu* pro detailní rozbor, vypočítány hodnoty MCC zanesené v tabulce 4.4. Ty svou výškou potvrzují dobrou schopnost detektorů klasifikovat obtížné detekce a mezi jednotlivými sítěmi jsou pouze očekávatelné rozdíly.

Tabulka 4.4: Hodnoty MCC dosažené při klasifikaci testování obtížných detekcí pro práh překrytí IoU 0,75.

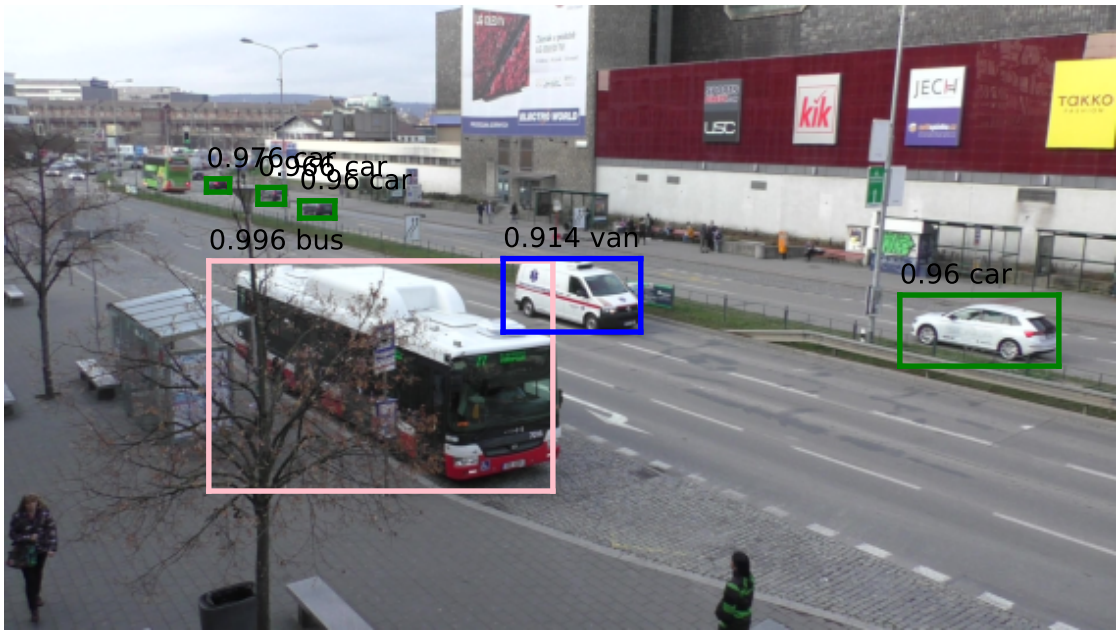
Detektor	MCC
YOLOv4	0,87
Mask R-CNN	0,80
YOLACT++	0,79

4.3.2 Zhodnocení

Mask R-CNN

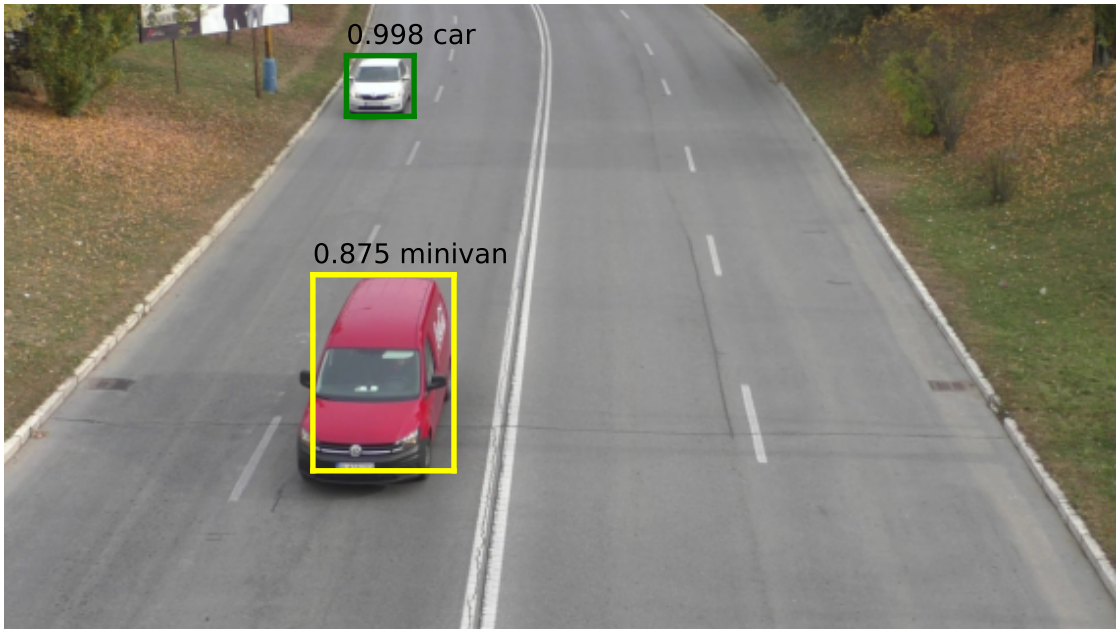
Dá se očekávat, že Mask R-CNN bude dosahovat vysokých stupňů přesnosti, což se potvrdilo prvenstvím detektoru v kategoriích $mAP^{IoU=.75}$, $mAP^{IoU=.50:.05:.95}$ a druhým místem v $mAP^{IoU=.50}$ (viz tabulka 4.2) pro obecné testování. Úspěch ve striktnějších katagoriích potenciálně značí přesnější schopnosti lokalizace *bounding boxů* oproti ostatním detektorům.

Detailnější pohled na výkonnost detektoru poskytují obrázky 4.7 a 4.8 s grafy křivek *precision-recall* pro jednotlivé třídy. Jejich srovnáním lze pozorovat konstantní charakter detekce s očekávaným snížením (avšak velmi mírným) pro úroveň *IoU* 0,75 oproti *IoU* 0,50. Z tvaru křivek vyplývá, že detektor dosahuje menšího snížení *precision* při vyšších úrovních *recallu* a že se *recall* obecně drží na velmi vysokých hodnotách a neklesá k nízkým úrovním, jako je tomu u ostatních sítí (viz obrázky 4.14, 4.15, 4.21 a 4.22).



Obrázek 4.5: Vizualizace výstupu detektoru Mask R-CNN na testovacím obrázku z lokace Vaňkovka. Název třídy a *confidence* je zobrazen pouze pro detekce s *confidence* přesahující hodnotu 0,5.

Pravděpodobně zajímavější jsou však hodnoty *mAP* pro obtížné detekce v tabulce 4.3, kde lze pozorovat, že Mask R-CNN dosahuje výrazně nižší přesnosti oproti ostatním detektorům. Umístění na posledním místě ve všech kategoriích přesnosti – $mAP^{IoU=.50}$, $mAP^{IoU=.50}$ i $mAP^{IoU=.50:.05:.95}$ se znatelným odstupem od ostatních sítí, je velmi polarizující oproti výkonu detektoru při běžném testování. Detailněji lze tuto skutečnost pozorovat na obrázku 4.9, ze kterého je patrné, že si detektor zachovává relativně vysokou *precision*, avšak dosahuje pouze relativně nízkých hodnot *recallu*, což způsobuje výsledné nízké hodnocení *mAP*. Detekce v těchto podmínkách vynechává větší množství objektu v porovnání s ostatními detektory, což značí potenciální nevhodnost detektoru Mask R-CNN pro užití k detekci v podobných případech.

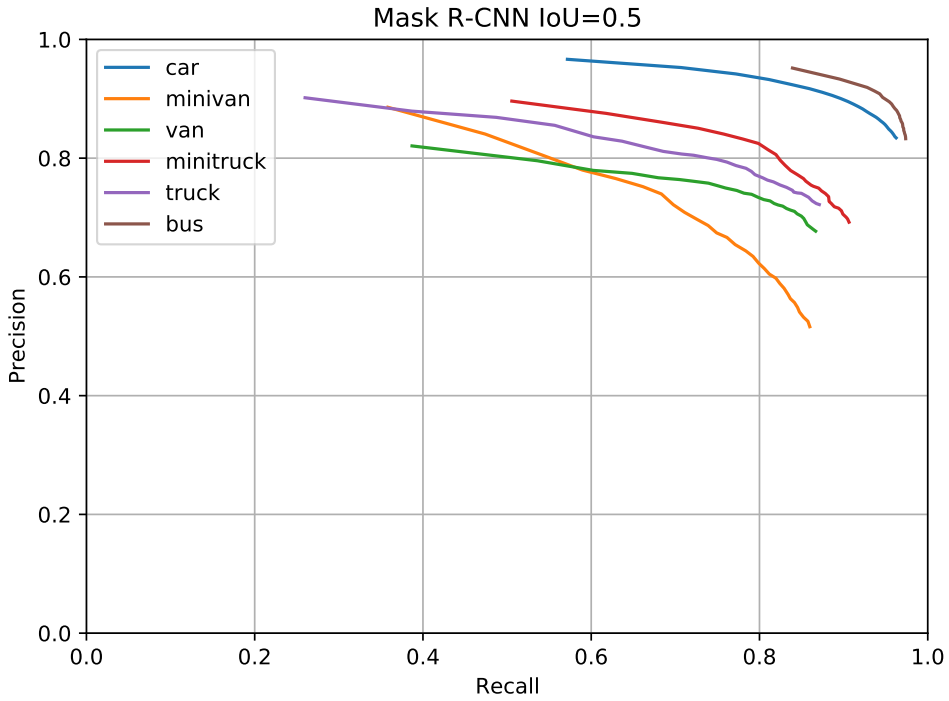


Obrázek 4.6: Vizualizace výstupu detektoru Mask R-CNN na testovacím obrázku z lokace most Jazdiareň. Název třídy a *confidence* je zobrazen pouze pro detekce s *confidence* přesahující hodnotu 0,5.

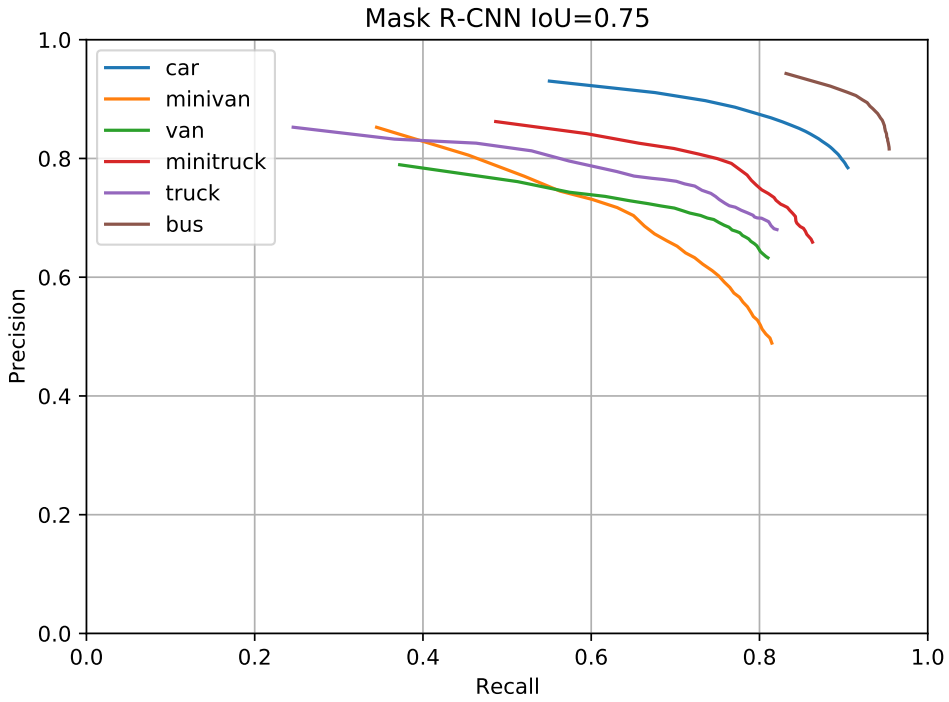
Při pohledu na samotnou klasifikaci obtížných detekcí (tabulka hodnot MCC 4.4), výkonnost Mask R-CNN zhruba odpovídá ostatním detektorům a neodráží rozměr odlišnosti celkové detekce. To lze objasnit i při pohledu na obrázek 4.10, kde je vidět rozložení hodnot *precision* a *recall* pro jednotlivé třídy klasifikace. Mimo odlehlejších případů, jako pro třídu *minitruck*, se hodnoty drží na lepších úrovních a lze tak soudit, že celkový výkon detektoru může být spíše způsoben lolalizací *bounding boxů* než klasifikací. Při pohledu na *confusion matrix* (viz obrázek 4.11) je možné odhalit konkrétnější nedostatky klasifikace, jako například relativně časté zaměňování vizuálně podobných tříd jako například *car* a *minivan*. Zajímavý je poměr záměn tříd *truck* a *minitruck*, kde se dá očekávat určitá nepřesnost, nicméně detektor velmi často špatně predikuje typ *minitruck* pro vozidla *truck*, avšak ne naopak.

Rychlosti detekce 3,44 *FPS* (viz tabulka 4.2) a 3,23 *FPS* (viz tabulka 4.3) zhruba odpovídají rychlosti detekce udávané pro *COCO test dataset*. [16]

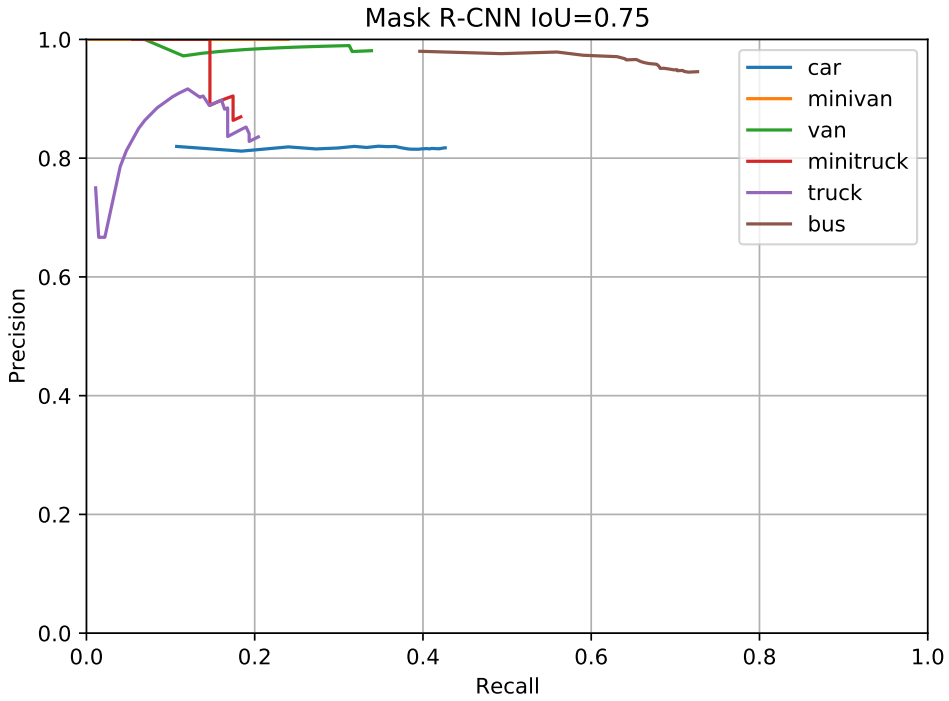
Vizualizace výstupu detektoru je možné vidět na obrázcích 4.5 a 4.6.



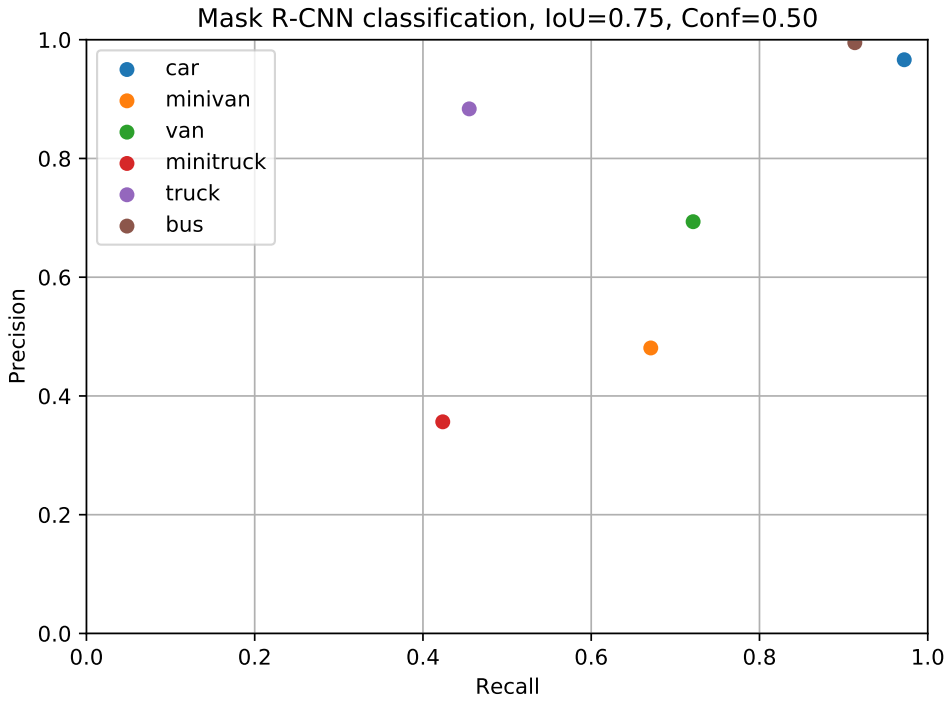
Obrázek 4.7: Graf zobrazující křivky *precision-recall* dosažené detektorem Mask R-CNN pro jednotlivé třídy při obecném testování s prahem překrytí *IoU* na úrovni 0,5.



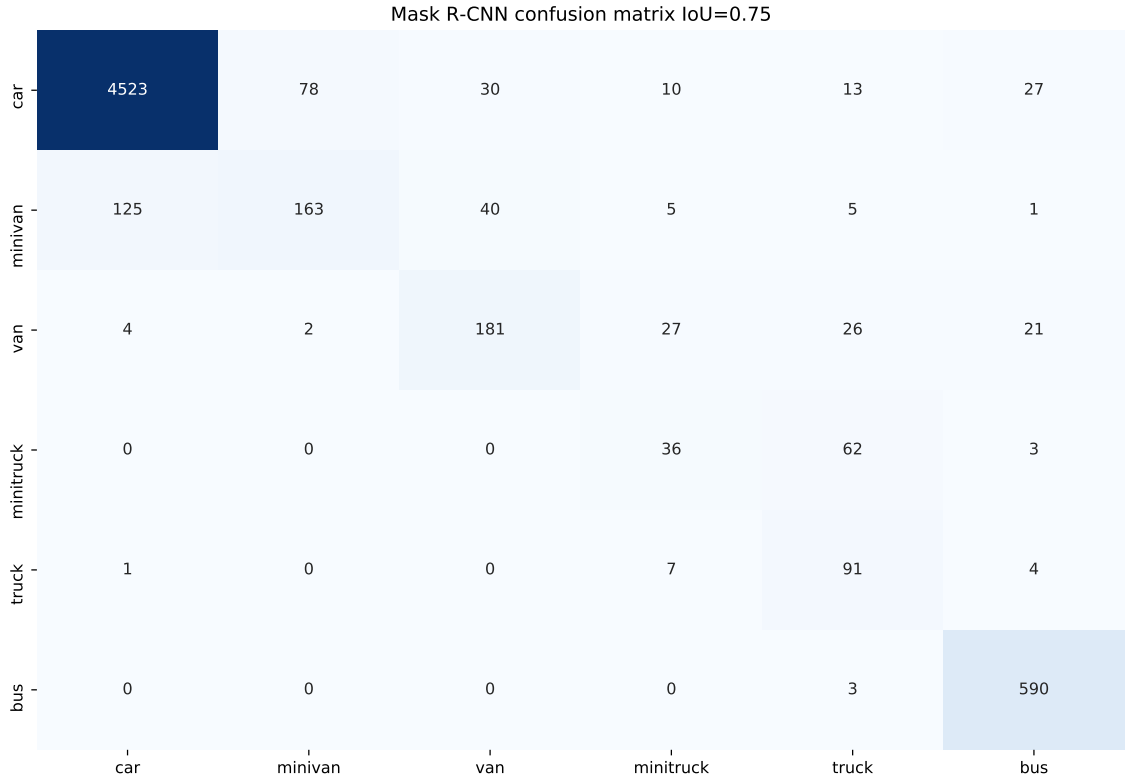
Obrázek 4.8: Graf zobrazující křivky *precision-recall* dosažené detektorem Mask R-CNN pro jednotlivé třídy při obecném testování s prahem překrytí *IoU* na úrovni 0,75.



Obrázek 4.9: Graf zobrazující křivky *precision-recall* dosažené Mask R-CNN pro jednotlivé třídy při testování na obtížných detekcích s prahem překrytí *IoU* na úrovni 0,75.



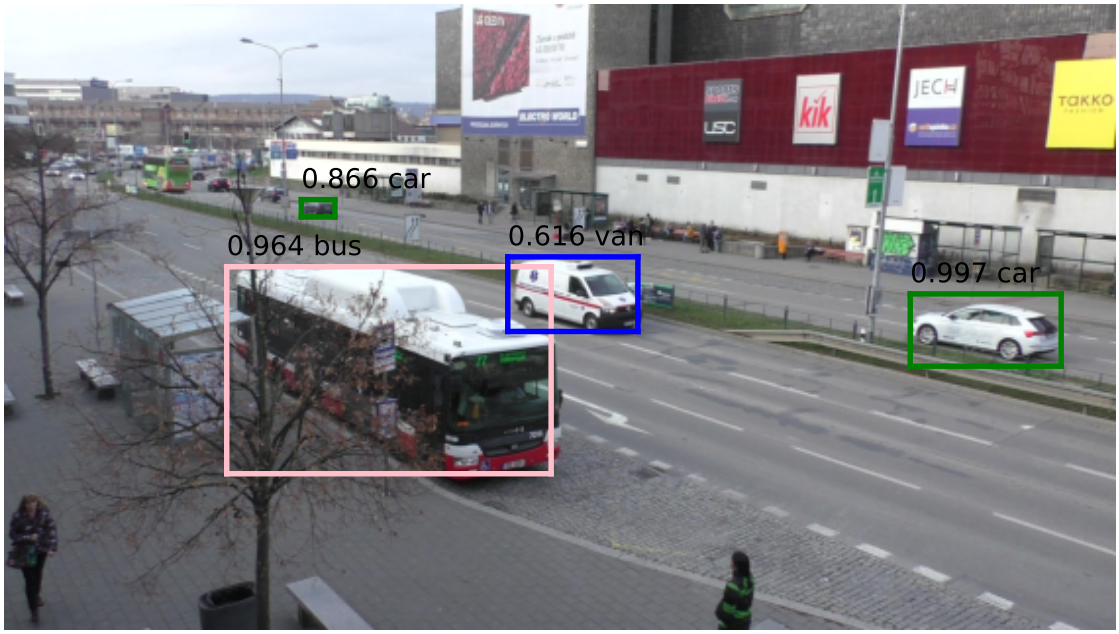
Obrázek 4.10: Hodnoty *precision* a *recall* jednotlivých tříd dosažené detektorem Mask R-CNN při klasifikaci obtížných detekcí. Měřeno pro práh *IoU* 0,75 a *confidence* 0,5.



Obrázek 4.11: *Confusion matrix* detektoru Mask R-CNN při klasifikaci obtížných detekcí. Měřeno pro práh *IoU* 0,75 a *confidence* 0,5. Sloupce znázorňují reálný typ vozidla, řádky označují výstup klasifikace.

YOLACT++

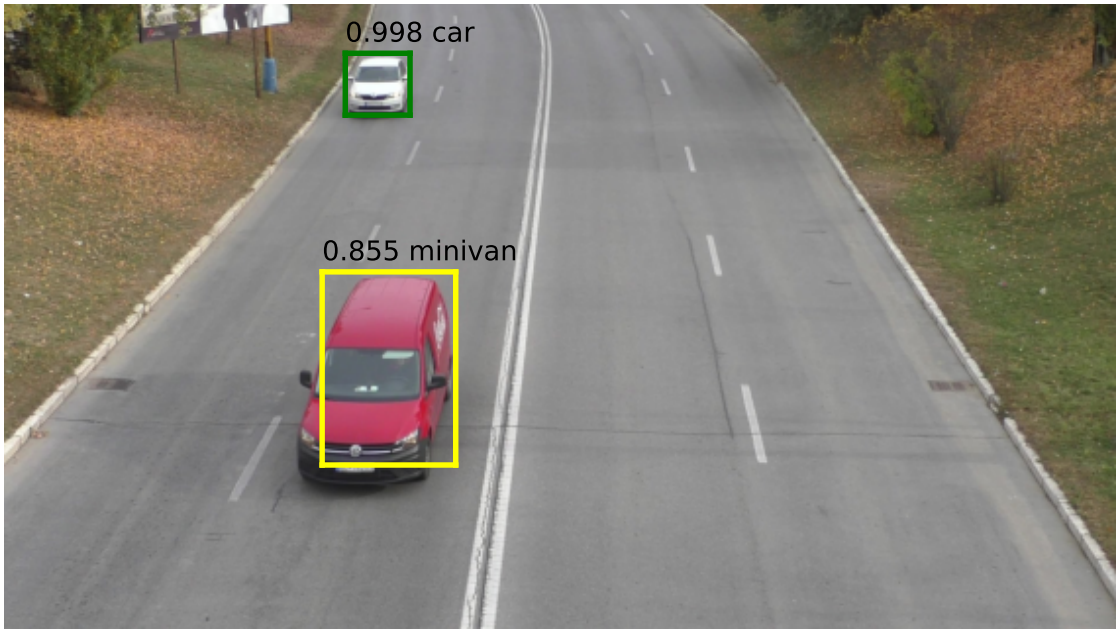
Detektor při testování dosahoval velmi dobrých výsledků mAP, jak lze vidět v tabulce 4.2, umístil se na druhém místě pro striktnější metriky $mAP^{IoU=.75}$ a $mAP^{IoU=.50:.05:.95}$ a pro $mAP^{IoU=.50}$ dosáhl zcela nejvyšší absolutní hodnoty. Tyto výsledky naznačují, že detektor je výkonnější pro nalézání více objektů, za cenu horší přesnosti při lokalizaci *bounding boxů*. Tento charakter detekce naznačuje i detailnější rozbor výsledků v grafech na obrázcích 4.14 a 4.15. Pro práh IoU 0,5 je u většiny tříd možné pozorovat dosažení téměř maximálních hodnot *recallu* (avšak s nízkou *precision*), naproti tomu pro striktnější práh IoU 0,75 lze pozorovat zmizení tohoto efektu a hodnoty na nebo pod úrovní ostatních detektorů (viz obrázky 4.8 a 4.22). Mimo jiné je na chování detektoru také patrné velmi plynulé klesání *precision* pro zvyšující se *recall*.



Obrázek 4.12: Vizualizace výstupu detektoru YOLACT++ na testovacím obrázku z lokace Vaňkovka. *Bounding box* je zobrazen pouze pro detekce s *confidence* přesahující hodnotu 0,5.

V porovnání s ostatními sítěmi se YOLACT++ při obecném testování prokázal jako velmi citlivý detektor se srovnatelnou přesností, to ho činí vhodnějším například pro použití, u kterého je důležitější nalezení všech objektů na vstupu více než jejich přesná lokalizace, například získávání informací o vytížení provozu a další.

Pravděpodobně ještě zajímavější hodnoty lze nalézt v tabulce 4.3, ze které je patrná dominance detektoru ve všech měřených kategoriích mAP pro obtížné detekce. YOLACT++ se umístil na prvním místě pro $mAP^{IoU=.50:.05:.95}$, $mAP^{IoU=.50}$ i $mAP^{IoU=.75}$ se značným odstupem od nižších příček. Oproti obecnému testování došlo u detektoru k nejmenší ztrátě výkonnosti v ohledu mAP pro obtížné detekce (typicky těsně nad hranici přibližně 11%). Při podrobnějším rozboru lze na obrázku 4.16 objevit možnou příčinu těchto výsledků – vysoké hodnoty *recallu* dosahované při detekci oproti ostatním detektorům (viz. obrázky 4.23 a 4.9). *Precision* se však drží na srovnatelných úrovních jako u ostatních detektorů.

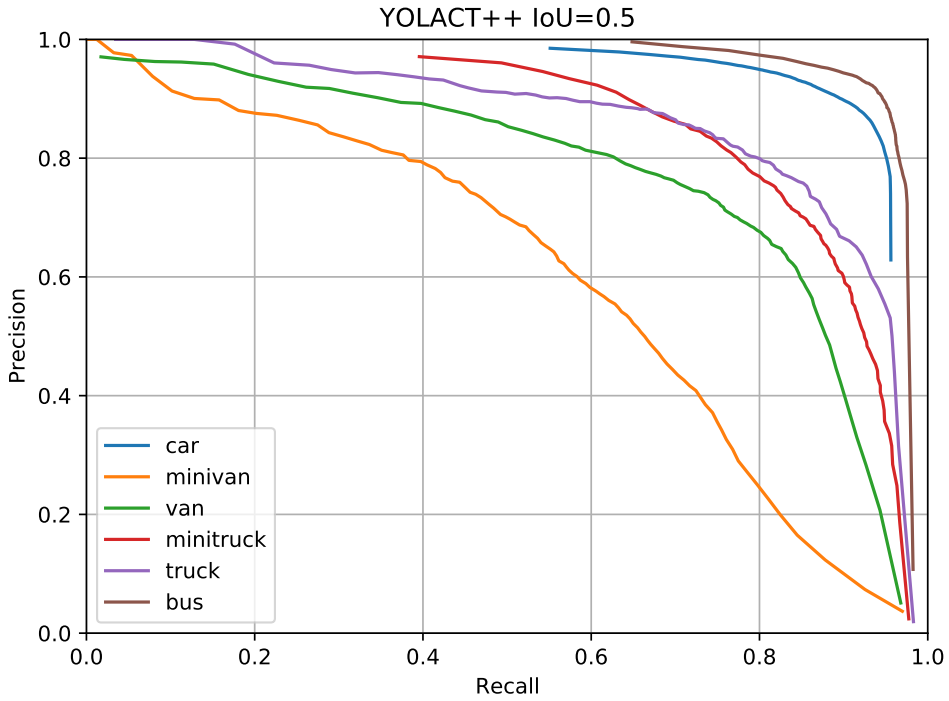


Obrázek 4.13: Vizualizace výstupu detektoru YOLACT++ na testovacím obrázku z lokace most Jazdiareň. *Bounding box* je zobrazen pouze pro detekce s *confidence* přesahující hodnotu 0,5.

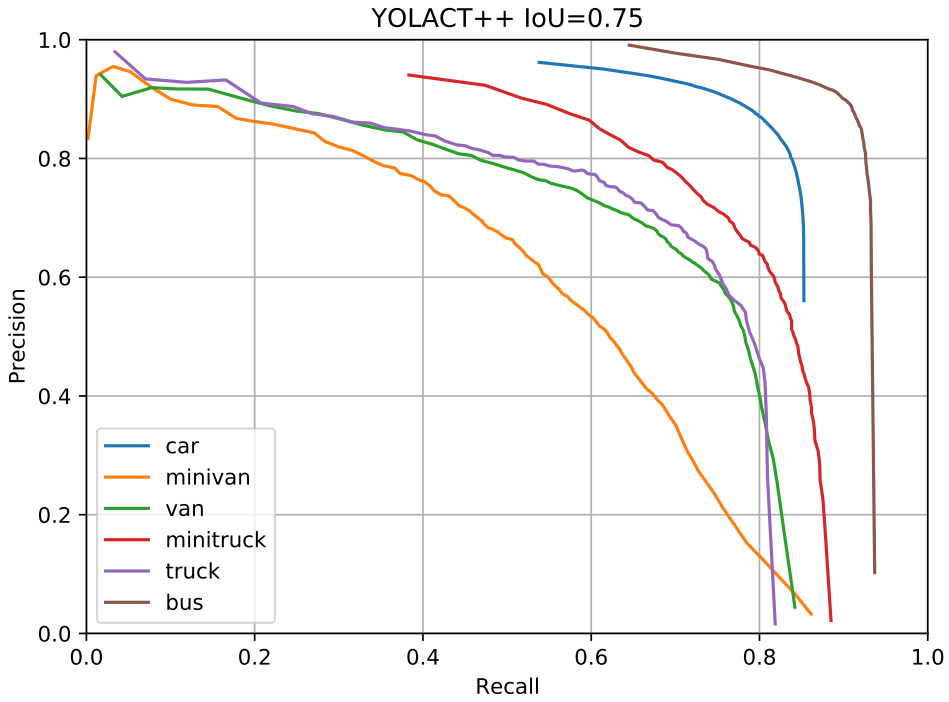
V tabulce 4.4 je vidět, že se detektor paradoxně umístil až na posledním místě pro hodnotu *MCC*, nicméně dosažená úroveň je z absolutního pohledu vysoká. Schopnost klasifikace obtížných detekcí lze důkladněji prozkoumat i pomocí grafu na obrázku 4.17, který ukazuje poměrně široké rozestupy dosažených hodnot mezi jednotlivými třídami. Zejména pak rozdíly pro téměř perfektní výsledek tříd *car* a *bus* oproti ostatním třídám. Na obrázku 4.18 je možné vidět *confusion matrix* klasifikace obtížných detekcích, ze které jsou patrné podobné jevy jako u Mask R-CNN (viz 4.11) – časté záměny tříd *minivan* s *car* (či opačně) a *truck* s třídou *minitruck*. Mimo to měl také YOLACT++ mírně větší problémy s klasifikací vozidel třídy *van*.

V tabulkách 4.2 a 4.3 jsou hodnoty dosažené rychlosti detekce při běžném testování i pro obtížné detekce. Ty sice dle očekávání překonávají rychlosti dosažené Mask R-CNN, nicméně zůstávají hluboko pod očekávanou hranicí 30 *FPS* použitelnou pro běh v reálném čase a dosažené detektorem YOLOv4. [7]

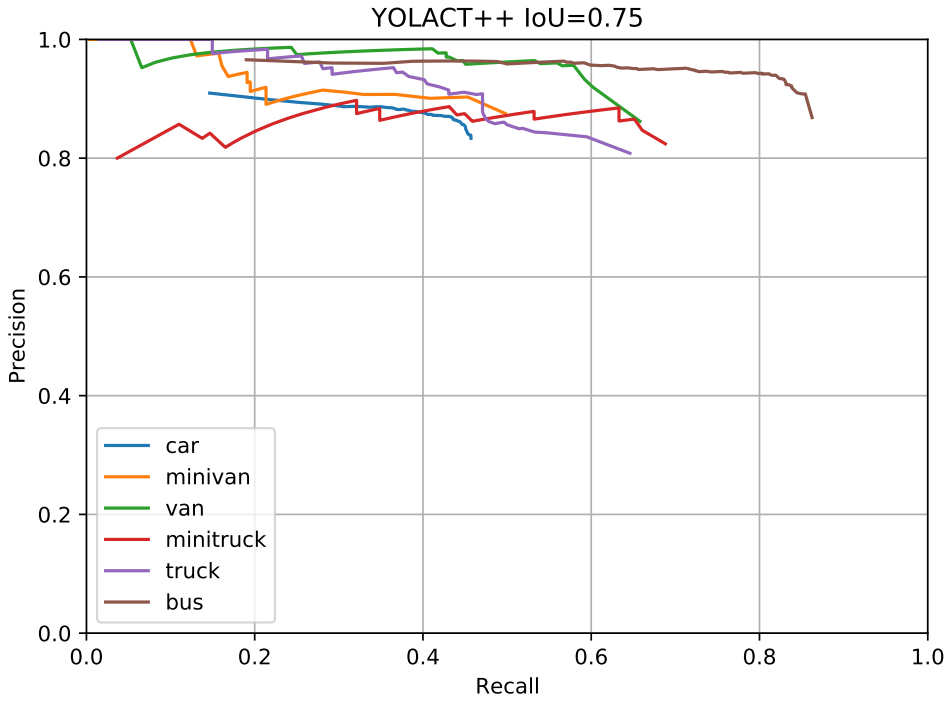
Vizualizace výstupu detektoru je možné vidět na obrázcích 4.12 a 4.13.



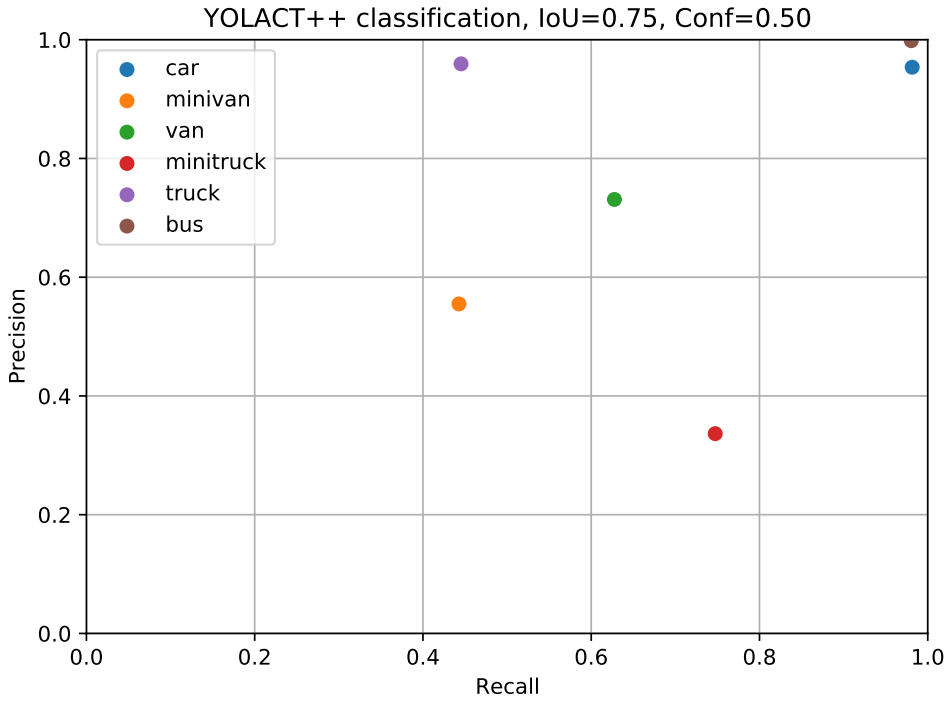
Obrázek 4.14: Graf zobrazující křivky *precision-recall* dosažené detektorem YOLACT++ pro jednotlivé třídy při obecném testování s prahem překrytí *IoU* na úrovni 0,5.



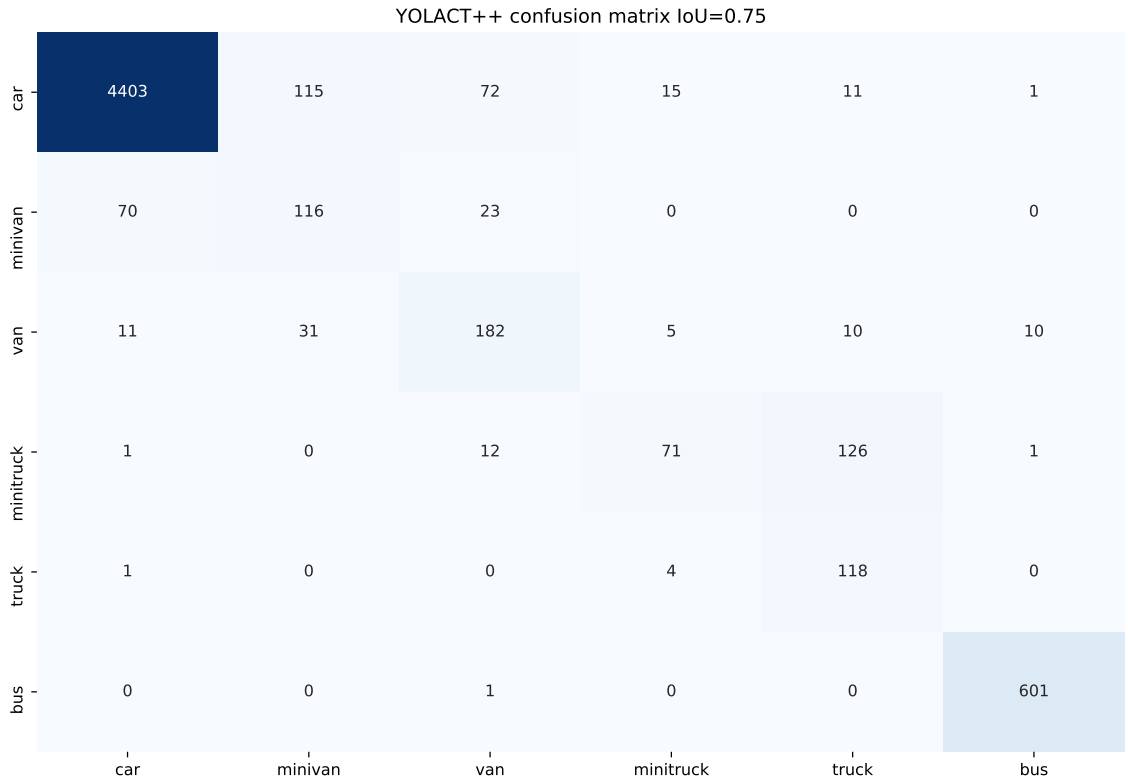
Obrázek 4.15: Graf zobrazující křivky *precision-recall* dosažené detektorem YOLACT++ pro jednotlivé třídy při obecném testování s prahem překrytí *IoU* na úrovni 0,75.



Obrázek 4.16: Graf zobrazující křivky *precision-recall* dosažené detektorem YOLACT++ pro jednotlivé třídy při testování na obtížných detekcích s prahem překrytí *IoU* 0,75.



Obrázek 4.17: Hodnoty *precision* a *recall* jednotlivých tříd dosažené detektorem YOLACT++ při klasifikaci obtížných detekcí. Měřeno pro práh *IoU* 0,75 a *confidence* 0,5.

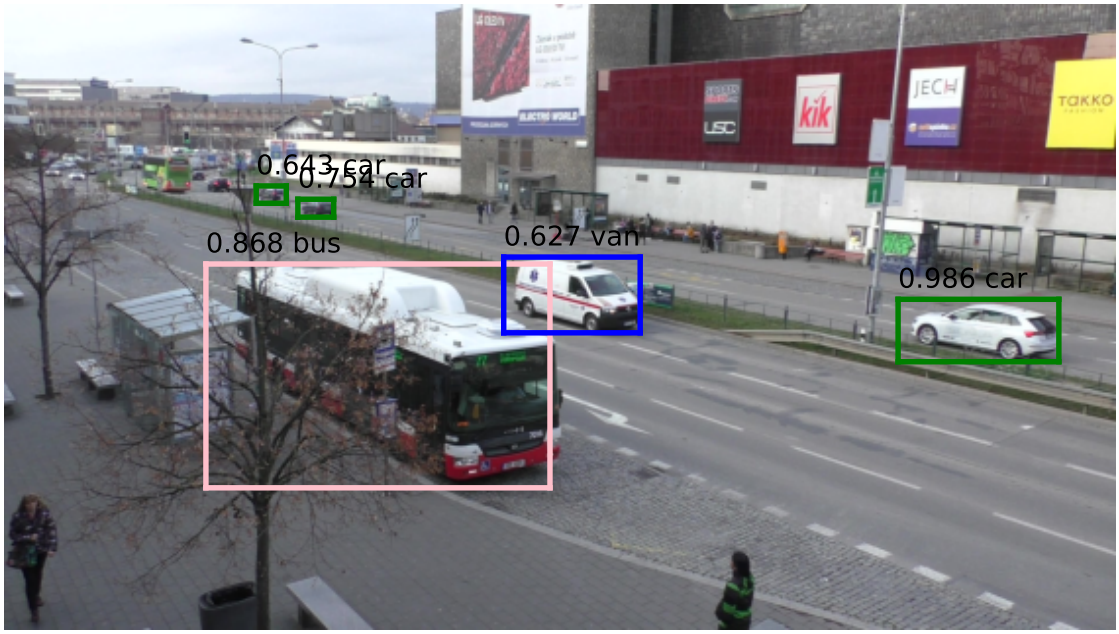


Obrázek 4.18: *Confusion matrix* detektoru YOLACT++ při klasifikaci obtížných detekcí. Měřeno pro práh *IoU* 0,75 a *confidence* 0,5. Sloupce znázorňují reálný typ vozidla, řádky označují výstup klasifikace.

YOLOv4

Od YOLOv4 lze vlivem jednoúrovňové architektury očekávat spíše skromnější výkon v ohledu dosažených hodnot mAP [5]. Při pohledu na tabulku 4.2 je vidět, že se detektor sice ve všech kategoriích ($mAP^{IoU=.75}$, $mAP^{IoU=.50:.05:.95}$ i $mAP^{IoU=.50}$) umístil až na třetím místě, avšak s rozdíly vždy pouze v řádu jednotek procent. Stejně jako ostatní detektory se dle očekávání hodnoty dosažené mAP snižují pro striktnější kategorie.

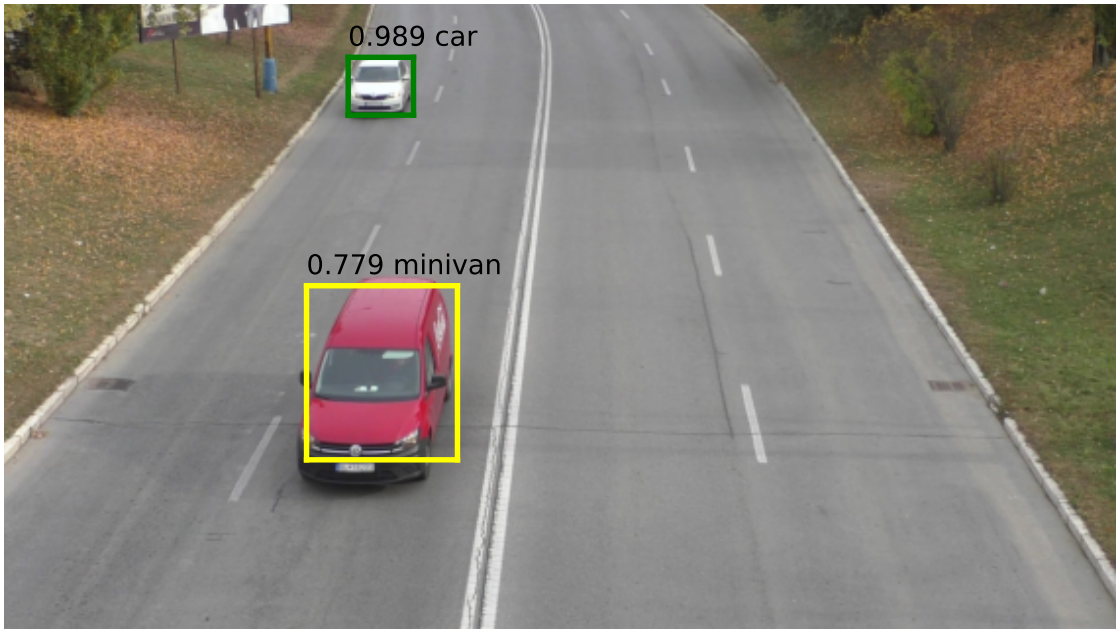
Detailnější náhled do výkonu detektoru nabízí obrázky 4.21 a 4.22 s grafy křivek *recision-recall* dosažených při detekci. Srovnání detekce pro prahy IoU 0,50 a 0,75 má očekávaný charakter se snížením dosahované *precision* i *recallu*. Nižší hodnoty mAP jsou pravděpodobně způsobeny nižším maximem křivek na ose *recallu*, než je tomu u ostatních detektorů (grafy viz. například obrázky 4.7 a 4.14). Detektor tak vykazuje nižší citlivost a zachycuje menší množství detekcí než ostatní testované sítě.



Obrázek 4.19: Vizualizace výstupu detektoru YOLOv4 na testovacím obrázku z lokace Vaňkovka. Název třídy a *confidence* je zobrazen pouze pro detekce s *confidence* přesahující hodnotu 0,5.

V tabulce 4.3 je možné vidět dosažené hodnoty mAP pro obtížné detekce. V této disciplíně se detektor umístil na střední příčce ve všech kategoriích ($mAP^{IoU=.75}$, $mAP^{IoU=.50}$ i $mAP^{IoU=.50:.05:.95}$) a to s více než 10% odstupem od obou dalších sítí. Detekce je tak zřetelně lepší než výkon Mask R-CNN, avšak také výrazně zaostává za YOLACT++. Z absolutního hlediska se jedná samozřejmě o velké zhoršení oproti běžné detekci, nicméně i přesto lze pozorovat například překročení prahu 50% pro kategorii $mAP^{IoU=.50}$. Detailnější pohled nabízí obrázek 4.23, kde jsou vidět odlišné charaktery detekce pro různé třídy. Například pro třídu *minivan* detektor dosahuje jen velmi nízkých hodnot *recallu*, avšak zhruba konstantní, relativně vysoké hodnoty pro ostatní třídy vyzvedávají celkový výkon detektoru pro obtížné detekce.

Při klasifikaci obtížných detekcí detektor dosáhl nejvyšší úrovně MCC ze všech testovaných sítí (viz tabulka 4.4). Pro konkrétnější rozbor je možné využít graf na obrázku 4.24,

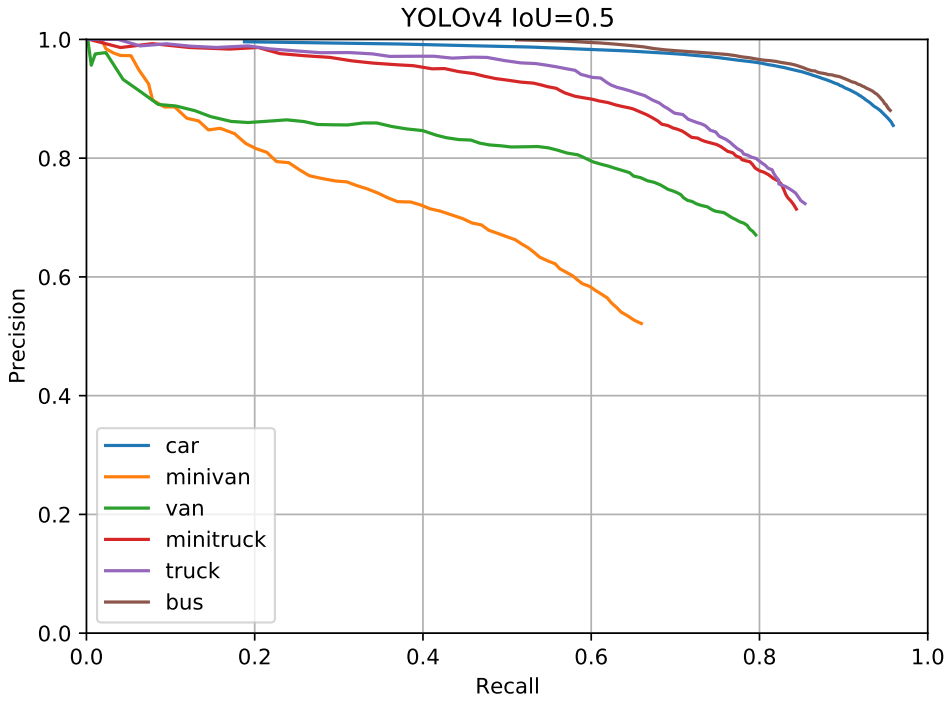


Obrázek 4.20: Vizualizace výstupu detektoru YOLOv4 na testovacím obrázku z lokace most Jazdiareň. Název třídy a *confidence* je zobrazen pouze pro detekce s *confidence* přesahující hodnotu 0,5.

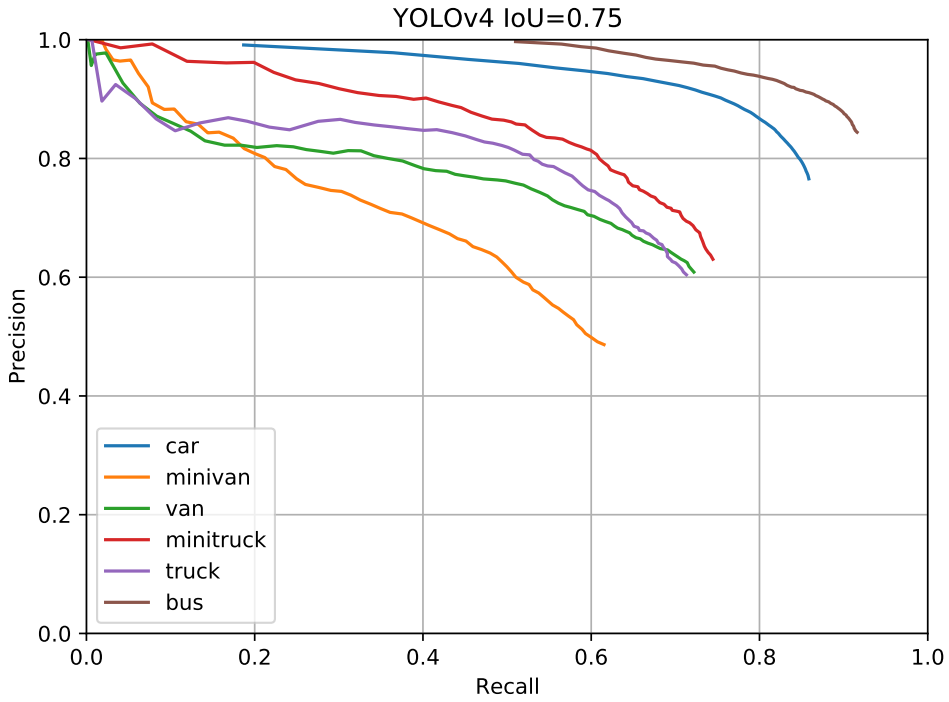
kde je vidět, že se naměřené hodnoty pohybují ve středních až vysokých hodnotách a pro třídy *car* a *bus* dokonce v hodnotách maximálních. Tyto výsledky dle očekávání souhlasí s dosaženou vysokou hodnotou *MCC* a celkovým výkonem při testování obtížných detekcí. Při průzkumu *confusion matrix* na obrázku 4.25 a jejím porovnání s předchozími detektory (viz obrázky 4.11 a 4.18) je patrné, že lepší hodnocení klasifikace YOLOv4 plyne pravděpodobně z nižšího počtu záměn vozidel třídy *car* za třídu *minivan*. Ostatní trendy (například záměny *truck* za *minitruck*) lze pozorovat v podobném rozsahu jako u ostatních detektorů.

Dosaženou rychlost detekce lze vidět v tabulkách 4.2 a 4.3, ze kterých je patrné, že i přes její větší rozpětí pro různé disciplíny je velmi vysoká a v kontextu ostatních testovaných detektorů bezkonkurenční. Detektor tak potvrzuje, že je vhodný pro použití v reálném čase – například pro detekci na živém přenosu z dopravních kamer a další. Tyto výsledky jsou konzistentní s očekáváním chování YOLOv4. [5].

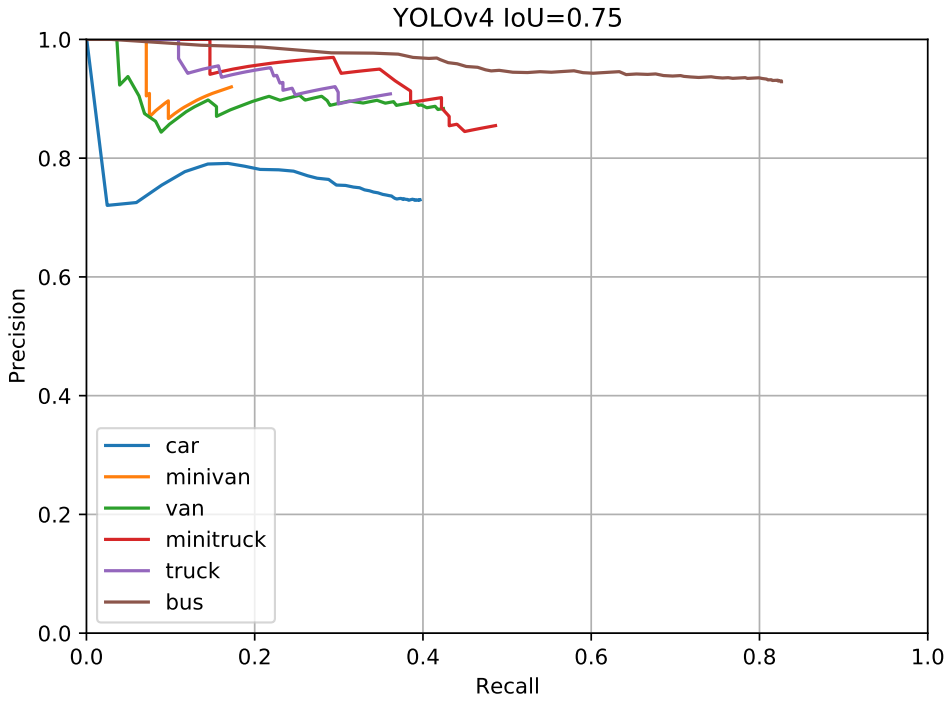
Vizualizace výstupu detektoru je možné vidět na obrázcích 4.19 a 4.20.



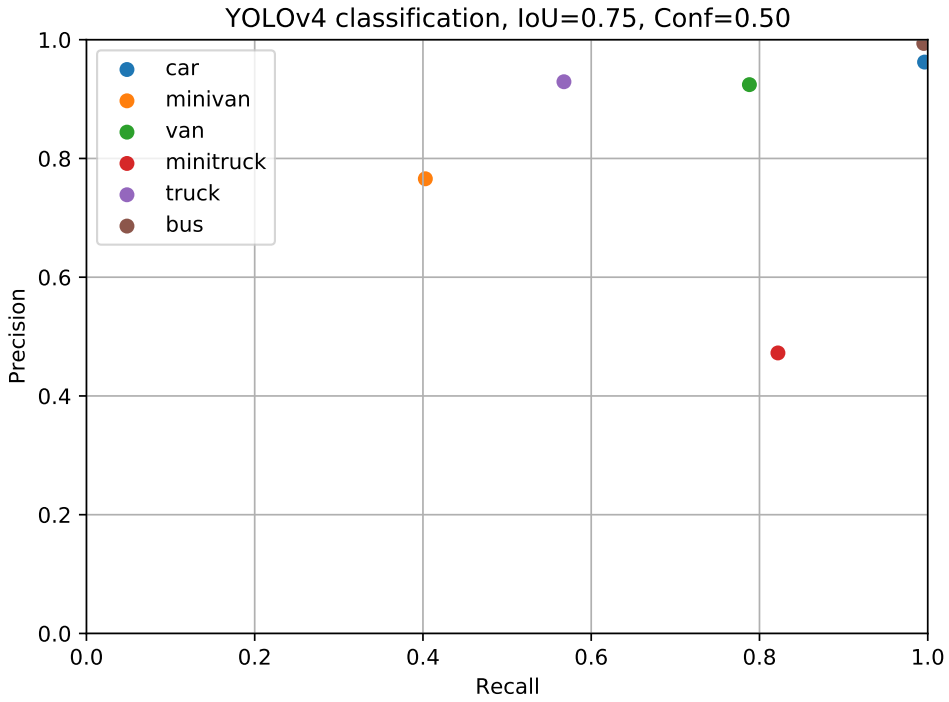
Obrázek 4.21: Graf zobrazující křivky *precision-recall* dosažené detektorem YOLOv4 pro jednotlivé třídy při obecném testování s prahem překrytí *IoU* na úrovni 0,5.



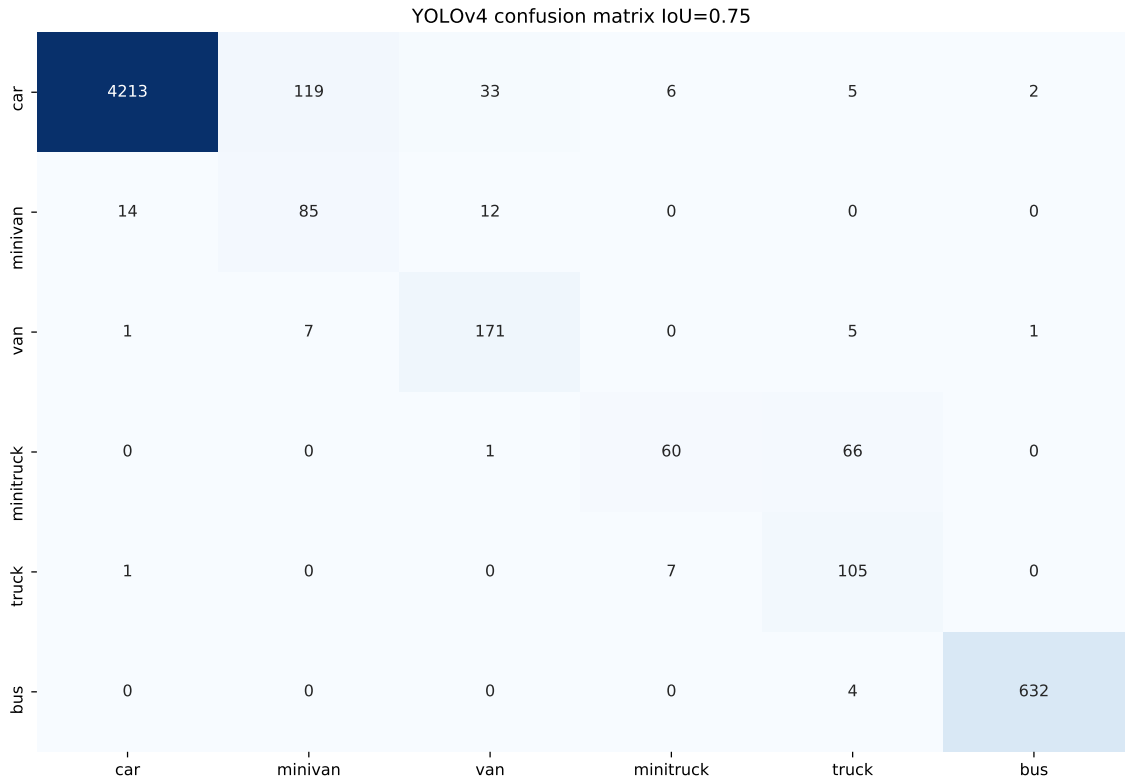
Obrázek 4.22: Graf zobrazující křivky *precision-recall* dosažené detektorem YOLOv4 pro jednotlivé třídy při obecném testování s prahem překrytí *IoU* na úrovni 0,75.



Obrázek 4.23: Graf zobrazující křivky *precision-recall* dosažené detektorem YOLOv4 pro jednotlivé třídy, při testování na obtížných detekcích s prahem překrytí *IoU* na úrovni 0,75.



Obrázek 4.24: Hodnoty *precision* a *recall* jednotlivých tříd dosažené detektorem YOLOv4 při klasifikaci obtížných detekcí. Měřeno pro práh *IoU* 0,75 a *confidence* 0,5.



Obrázek 4.25: *Confusion matrix* detektoru YOLOv4 při klasifikaci obtížných detekcí. Měřeno pro práh *IoU* 0,75 a *confidence* 0,5. Sloupce znázorňují reálný typ vozidla, řádky označují výstup klasifikace.

Srovnání tříd

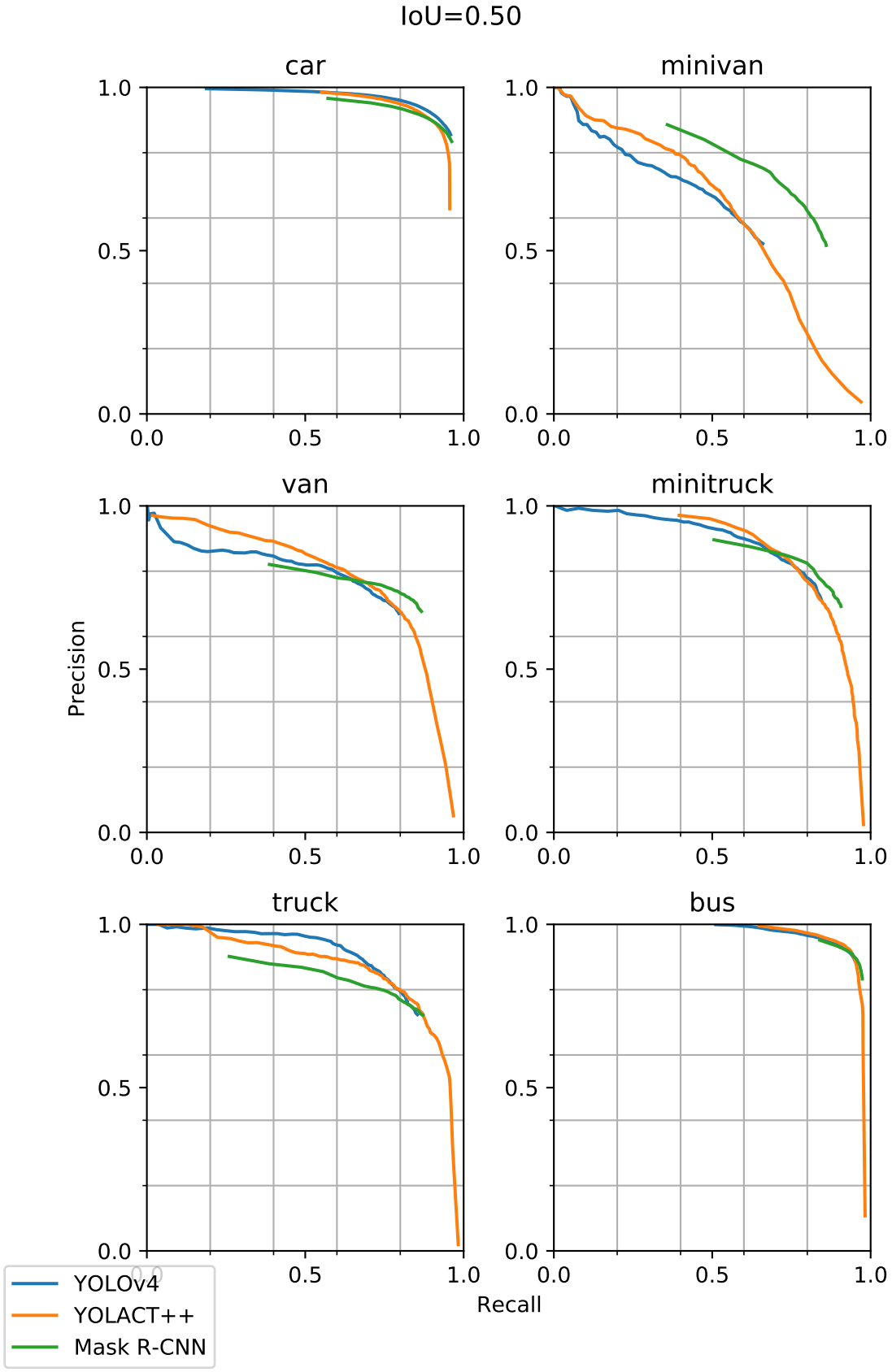
Na obrázcích 4.26 a 4.27 je zobrazeno srovnání křivek *precision-recall* pro různé třídy pro hlubší průzkum jejich chování. Na tvaru křivek je vidět, že v detekci různých sítí pro jednu třídu jsou jen malé rozdíly v tomto ohledu. Každý detektor samozřejmě dosáhne různé *precision* a různého *recallu*, z čehož plyne jejich rozdílné hodnocení. Průběh těchto dvou hodnot napříč hladinami *confidence* je však pro všechny velmi podobný.

Zajímavější je tak posouzení rozdílů v detekci jednotlivých tříd, které se může razantně lišit. Toto je zvláště patrné pro třídy *car* a *bus*, které vykazují mimo většího vyhlazení křivek zvýšenou granularitou výsledků, také vyšší dosažené hodnoty *precision*.

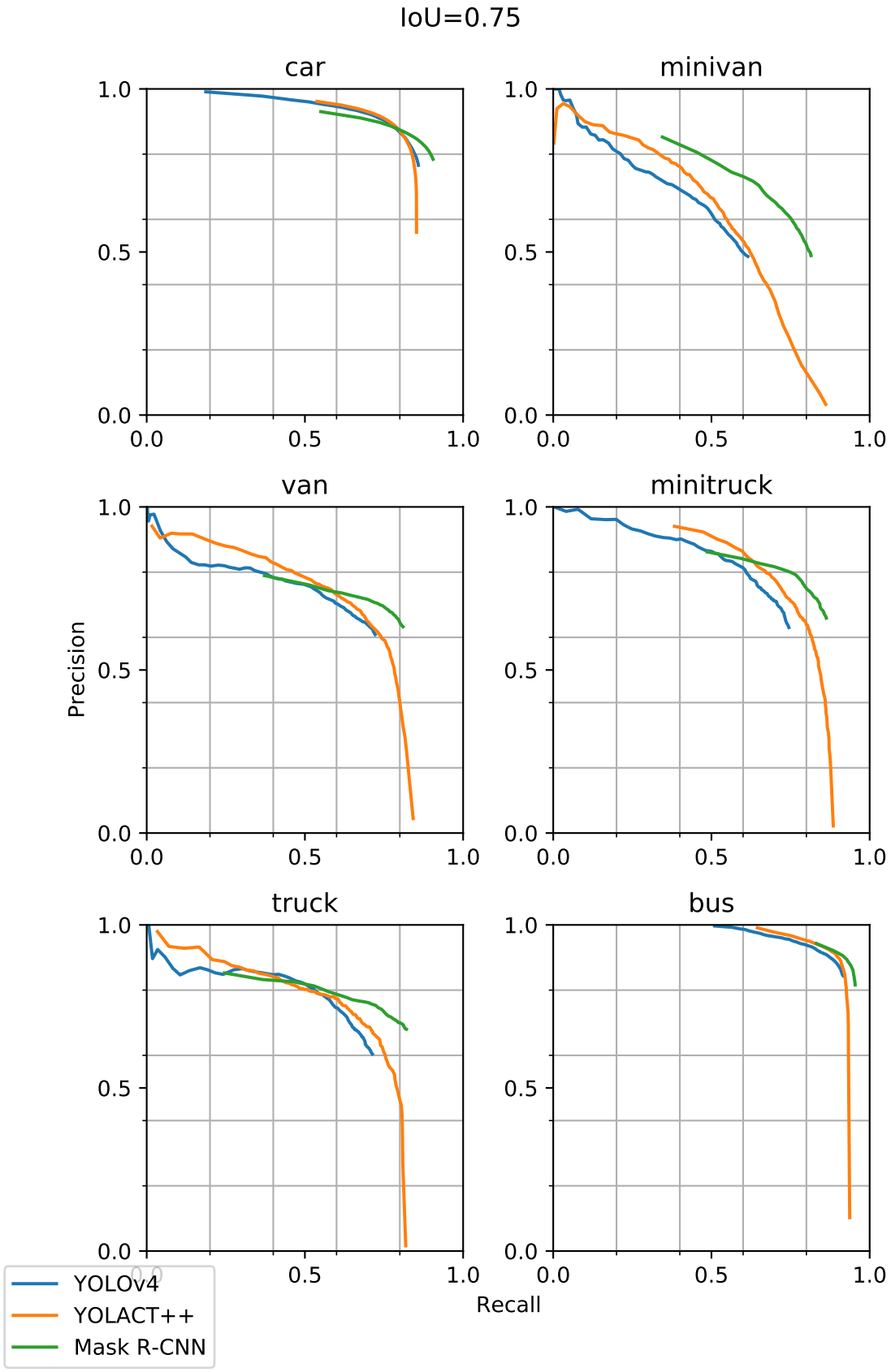
Možnou příčinou zvýšených hodnot *precision* pro tyto třídy by samozřejmě mohlo být jejich větší zastoupení v učící části datové sady (viz. tabulka 3.2). Nicméně tato hypotéza není zcela konzistentní při pohledu na třídu *van*, pro kterou detektory vykazují relativně nízké úrovně *precision* i přesto, že se k její četnosti (viz tabulka 3.1) třída *bus* blíží daleko více než k dominantní třídě *car*. Možným vysvětlením tohoto jevu pro třídu *bus* je jednotlitý vzhled instancí této třídy. Z povahy věci má většina autobusů v datové sadě uniformní vzhled, jako například barevné schéma městské hromadné dopravy nebo společnosti provozující dálkové autobusy. Detektor tak může být naučen velmi dobře rozpoznávat tyto a podobné instance jako se nachází v datové sadě, nicméně při testování na datech z jiných lokací by mohla kvalita detekce pro třídu *bus* klesat mnohem více, než pro třídy ostatní.

Nejnižší hodnoty *precision* dosahovali detektory většinou pro třídu *minivan*. Tvary křivek *precision-recall* (viz obrázky 4.26 a 4.27) se pro tuto třídu vyýrazně liší od trendů pozorovatelných u tříd ostatních. Zastoupením v části datové sady pro učení se sice jedná o minoritní třídu (viz tabulka 3.2), to však nevysvětluje její postavení vůči chování pro třídy *truck* a *minitruck*. Možnou příčinou je tak obecně vizuální podobnost vozidel této třídy s vozidly tříd *car* a *van*. Ta se projevovala již při přípravě datové sady, kdy i pro lidské oko nebylo vždy na první pohled zcela jasná klasifikace těchto vozidel.

Pohled na *confusion matrix* jednotlivých detektorů (viz obrázky 4.11, 4.18 a 4.25) odhaluje, že velké množství záměn vzniká klasifikováním vozidel třídy *truck* jako *minitruck*. Pravděpodobnou příčinou tohoto jevu je obecná vizuální podoba těchto tříd v kombinaci s charakterem obtížných detekcí. Například při vjezdu vozidla do záběru kamery při jejím umístění přibližně souběžně s vozovkou, jsou zakryté zřejmě nejpatrnější odlišovací rysy, jako je například počet náprav vozidla. Tomuto jevu možná nahrává i potenciální různorodost vzhledu instancí tříd *truck* a *minitruck* v provedení nákladového prostoru. Je možné, že data pro učení obsahují výrazně větší zastoupení jednoho typu provedení nákladového prostoru, například sklápěčkového, pro třídu *minitruck*. Při testování by pak detektor mohl častěji predikovat třídu *minitruck* pro sklápěčková vozidla třídy *truck*, protože by provedení nákladového prostoru hrálo při klasifikaci větší roli než prvky, které od sebe třídy ve skutečnosti odlišují.



Obrázek 4.26: Grafy zobrazující křivky *precision-recall* dosažené detektory pro jednotlivé třídy pro práh *IoU* 0,50.



Obrázek 4.27: Grafy zobrazující křivky *precision-recall* dosažené detektory pro jednotlivé třídy pro práh *IoU* 0,75.

Kapitola 5

Závěr

Cílem práce bylo porovnat výkonnost dostupných vícetřídních detektorů na vhodné datové sadě. Text práce nejprve čtenáře seznamuje s potřebným teoretickým základem a dále konkrétně popisuje připravenou datovou sadu, proces učení detektorů a provádění testů, jejichž výsledky se snaží vhodně interpretovat.

Prvním výstupem práce je připravená datová sada, kterou lze použít k učení a testování dalších detektorů. Tato část práce se ukázala jako obzvláště časově náročná, kvůli nutnosti manuální revize obrázků. Mimo množství dat pro učení a testování poskytuje také možnost specializovanějšího experimentování na obtížných detekcích. Prostor pro její rozšíření je zejména v nevyváženosti zastoupení jednotlivých tříd, které by se dalo realizovat podvzorkováním či ignorováním části dat s výskytem majortiních tříd (zejména třídy *car* reprezentující osobní auta). Tento proces by však vyústil ve ztrátu značného objemu dat, které by bylo vhodné doplnit jejich dalším rozšířením. Velikost třídně vyvážené sady by tak byla vždy zhruba limitována četností nejméně zastoupené třídy. Dalším možným vylepšením by bylo rozšíření části pro obtížné detekce, vedoucí k větší granularitě výsledků.

Výstupem druhým jsou naměřené výsledky při testování detektorů, které obecně prokazují schopnost detekovat vozidla, pro všechny testované sítě. Tuto skutečnost lze v první řadě interpretovat jako kontrolu a potvrzení použitelnosti připravené datové sady. Ta je tak i přes své nedostatky vhodná pro hluboké učení a testování detektorů k dosažení kvalitních výsledků.

V řadě druhé lze interpretovat zajímavé dosažené výsledky. Mask R-CNN se prokázal jako nejlepší testovaný detektor pro striktní požadavky na lokalizaci objektů, avšak pouze pro obecné testování. Velmi špatný výkon detektoru pro obtížné objekty se dá považovat za překvapivý. Vzhledem k tomu, že při obecném testování je třeba detekovat i obtížné objekty, mohou být výsledky Mask R-CNN při obecném testování sníženy právě tímto vlivem. Je možné, že by detektor dosáhl větší kvality detekce, kdyby byl testován pouze na jednoduchých, ničím nezakrytých objektech.

Detektor YOLOv4 se ukázal jako kompetentní varianta s vysokou běhovou rychlostí. V testovaných disciplínách se prokazoval průměrně dobrými výsledky.

Výsledky detektoru YOLACT++ naznačují vysokou citlivost detekce se stále dobrou přesností. Obzvláště zajímavé je jeho chování při obtížných detekcích, kdy došlo jen k malému snížení výkonu oproti běžnému testování, což zcela kontrastuje s velkým úpadkem pozorovaným u ostatních detektorů.

Literatura

[1] Abdulla, W. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow* [online]. Github, 2017 [cit. 2021-05-02]. Dostupné z: https://github.com/matterport/Mask_RCNN.

[2] *AlexeyAB/darknet* [online]. Github, listopad 2013 [cit. 2021-05-01]. Dostupné z: https://github.com/AlexeyAB/darknet.

[3] Ballard, D. H. a Brown, C. M. *Computer Vision*. Prentice-Hall, 1982.

[4] Bewley, A., Ge, Z., Ott, L., Ramos, F. a Upcroft, B. Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, s. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.

[5] Bochkovskiy, A., Wang, C.-Y. a Liao, H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv preprint arXiv:2004.10934*. 2020.

[6] Bolya, D., Zhou, C., Xiao, F. a Lee, Y. J. YOLACT: Real-time Instance Segmentation. In: *ICCV*. 2019.

[7] Bolya, D., Zhou, C., Xiao, F. a Lee, Y. J. YOLACT++: Better Real-time Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2020.

[8] Brownlee, J. *Overfitting and Underfitting With Machine Learning Algorithms* [online]. Srpen 2019 [cit. 2021-05-04]. Dostupné z: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/#:%7E:text=Overfitting%20in%20Machine%20Learning&text=Overfitting%20happens%20when%20a%20model,as%20concepts%20by%20the%20model.

[9] Brownlee, J. *A Gentle Introduction to Object Recognition With Deep Learning* [online]. Jan 2021 [cit. 2021-04-02]. Dostupné z: https://machinelearningmastery.com/object-recognition-with-deep-learning/.

[10] *COCO - Common Objects in Context* [online]. [cit. 2021-04-02]. Dostupné z: https://cocodataset.org/#home.

[11] *Convolutional Neural Network Algorithms* [online]. [cit. 2021-05-02]. Dostupné z: https://docs.ecognition.com/v9.5.0/eCognition_documentation/Reference%20Book/23%20Convolutional%20Neural%20Network%20Algorithms/Convolutional%20Neural%20Network%20Algorithms.htm.

[12] Everingham, M., Van Gool, L., Williams, C. K., Winn, J. a Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*. Springer. 2010, sv. 88, č. 2, s. 303–338.

[13] Goodfellow, I., Bengio, Y. a Courville, A. *Deep Learning*. Amsterdam, Netherlands: Amsterdam University Press, 2016.

[14] Gorodkin, J. Comparing two K-category assignments by a K-category correlation coefficient. *Computational Biology and Chemistry*. 2004, sv. 28, č. 5, s. 367–374. DOI: https://doi.org/10.1016/j.compbiolchem.2004.09.006. ISSN 1476-9271. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1476927104000799.

[15] Guo, J., Han, K., Wang, Y., Zhang, C., Yang, Z. et al. Hit-Detector: Hierarchical Trinity Architecture Search for Object Detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[16] He, K., Gkioxari, G., Dollár, P. a Girshick, R. Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. 2017, s. 2961–2969.

[17] He, K., Zhang, X., Ren, S. a Sun, J. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[18] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv preprint arXiv:1704.04861*. 2017.

[19] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A. et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[20] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B. et al. Feature Pyramid Networks for Object Detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[21] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. et al. SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N. a Welling, M., ed. *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, s. 21–37.

[22] Mohajon, J. *Confusion Matrix for Your Multi-Class Machine Learning Model* [online]. Duben 2021 [cit. 2021-05-02]. Dostupné z: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

[23] O'Shea, K. a Nash, R. An introduction to convolutional neural networks. *ArXiv preprint arXiv:1511.08458*. 2015.

[24] Redmon, J. a Farhadi, A. YOLO9000: Better, Faster, Stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[25] Redmon, J. a Farhadi, A. Yolov3: An incremental improvement. *ArXiv preprint arXiv:1804.02767*. 2018.

[26] Ren, S., He, K., Girshick, R. a Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2016.

[27] Schraudolph, N. a Cummins, F. [online]. [cit. 2021-01-03]. Dostupné z: https://cnl.salk.edu/~schraudo/teach/NNcourse/ann-overview.html.

[28] Sharma, S. *Epoch vs Batch Size vs Iterations - Towards Data Science* [online]. Březen 2019 [cit. 2021-01-7]. Dostupné z: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9.

[29] Sobel, I. An Isotropic 3x3 Image Gradient Operator. *Presentation at Stanford A.I. Project 1968*. Únor 2014.

[30] Soviany, P. a Ionescu, R. T. Optimizing the Trade-off between Single-Stage and Two-Stage Object Detectors using Image Difficulty Prediction. 2018.

[31] *Tzutalin/labelImg* [online]. Github, 2016 [cit. 2021-05-07]. Dostupné z: https://github.com/tzutalin/labelImg.

[32] Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W. et al. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.

[33] Xie, S., Girshick, R., Dollar, P., Tu, Z. a He, K. Aggregated Residual Transformations for Deep Neural Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[34] Zeng, N. *An Introduction to Evaluation Metrics for Object Detection* [online]. Prosinec 2018 [cit. 2021-04-17]. Dostupné z: https://blog.zenggyu.com/en/post/2018-12-16/an-introduction-to-evaluation-metrics-for-object-detection/.

[35] Zhang, X. *Simple Understanding of Mask RCNN - Xiang Zhang* [online]. Duben 2020 [cit. 2021-05-05]. Dostupné z: https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95#:%7E:text=Mask%20RCNN%20is%20a%20deep,a%20image%20or%20a%20video.&text=First%2C%20it%20generates%20proposals%20about,based%20on%20the%20input%20image.