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

ROZPOZNÁVÁNÍ A KLASIFIKACE DOPRAVNÍCH SITUACÍ

RECOGNIZING AND CLASSIFICATION OF TRAFFIC SITUATIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JIŘÍ ZBOŘIL**
AUTHOR

VEDOUCÍ PRÁCE **doc. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2022

Vysoké učení technické v Brně
Fakulta informačních technologií

Zadání bakalářské práce

Student: **Zbořil Jiří**
Program: Informační technologie
Název: **Rozpoznávání a klasifikace dopravních situací**
Recognizing and Classification of Traffic Situations
Kategorie: Umělá inteligence
Zadání:

1. Seznamte se s moderními metodami identifikace vozidel a objektů vyskytujících se v dopravních scénách a klasifikací aktivit pomocí neuronových sítí.
2. Shromážděte dostupné datové sady pro průběžné testování systému.
3. Na základě získaných poznatků navrhněte a implementujte systém, který dokáže rozpoznávat a klasifikovat vybrané dopravní situace ve videu z dohledových kamer.
4. Vyhodnoťte výsledky systému na reprezentativním vzorku dat.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle domluvy s vedoucím

Pro udělení zápočtu za první semestr je požadováno:

- funkční prototyp řešení

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/
Vedoucí práce: Smrž Pavel, doc. RNDr., Ph.D.
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 1. listopadu 2021

Abstrakt

Cílem této práce je identifikace a klasifikace nebezpečných situací z přehledových kamer, monitorujících dopravní provoz. Příkladem takových situací jsou nebezpečné stání podél silnice a dopravní nehody, na které se práce zaměřuje. Vytvořený systém využívá detektor objektů, analyzující průměrné snímky v daném intervalu, algoritmů K nejbližších sousedů a K Means a opětovnou detekci objektů v lokálně zvětšené oblasti pro výběr kandidátů. Detekované objekty, nacházející se mimo silnici jsou přiložením masky silnice eliminovány. Modul pro zpracování anomálie pak vypočítá její interval a klasifikaci. Naměřené F1 skóre je 0,645, S4 skóre 0,535 a procentuální úspěšnost klasifikace anomálie 80 %.

Abstract

The aim of this thesis is to identify and classify dangerous situations from surveillance cameras, monitoring traffic. An example of such situations is dangerous standing near by the road and car crash, on which this work focuses. The created system uses object detector, analyzing average images in given interval, K nearest neighbor and K Means algorithm and re-detection of enlarged local area in a frame to select anomaly candidates. Detected objects, that do not belong on the road are eliminated by attaching created road mask. At the very last phase, the interval, together with the classification is determined. Calculated F1 score is 0.645, S4 score 0.535 and precision of classification 80 %.

Klíčové slova

detekce anomálie, klasifikace situace, analýza videa, detekce objektů, modelování pozadí, výběr kandidátů

Keywords

anomaly detection, classification of situation, video analysis, object detection, background modeling, candidate selection,

Citace

ZBOŘIL, Jiří. Rozpoznávání a klasifikace dopravních situací. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Pavel Smrž, Ph.D.

Rozpoznávání a klasifikace dopravních situací

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením doc. RNDr. Pavel Smrž, Ph.D. Další informace mi poskytl Ing. Jakub Špaňhel. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

. . . . . .

Jiří Zbořil

11. května 2022

Poděkování

Tímto bych chtěl poděkovat doc. RNDr. Pavlu Smržovi, Ph.D., za jeho odbornou pomoc a vedení této práce. Dále bych chtěl poděkovat Ing. Jakubu Špaňhelovi, který mi poskytl použitou datovou sadu.

Obsah

Kapitola 1

Úvod

Záznamy z dopravních kamer, jejichž počet se neustále zvyšuje, jsou používány k monitorování provozu za účelem redukce přestupků a trestných činů, jako například vysoká rychlost, ale také pro získání celkového přehledu o dopravní situaci operátory dohledových center, případně zrychlení zásahu a včasnému odstranění překážek a problémů. Při nebezpečné situaci na silnici může být totiž doba nahlášení dlouhá. S ohledem na rostoucí počet dopravních kamer je práce monitorování provozu zdlouhavá a únavná. Ve světe plném dopravních prostředků jsou dopravní nehody velký problém a v jejich důsledků přichází o život spoustu lidí. Při vážnější autonehodě může přežití zraněných záviset na vteřinách a včasný příjezd rychlé záchranné pomoci je kritický.

Zlepšováním algoritmů spadající pod oblast umělé inteligence umožnilo rozpoznávat objekty ve snímku v reálném čase. Kombinováním algoritmů z oboru počítačového vidění lze zkonstruovat programy schopné hlubší analýzy z videa. Tyto programy by pak mohly být implementovány do systému, který by analyzoval jednotlivé záznamy z dostupných kamer.

Cílem této práce je návrh a implementace systému, schopného rozpoznávat a klasifikovat vybrané dopravní situace z videa s vysokou rychlostí analýzy. Pro validaci jsou vybrány dopravní situace, mezi které patří nebezpečné stání podél silnice a autonehoda. Jeho realizace by pak mohla pomoct dopravním orgánům průběžně informovat o stavu na silnicích z výběrových dohledových kamer.

Technická zpráva je rozdělena do teoretické a praktické části. V teoretické části budou popsány metodiky a algoritmy pro realizaci této práce. Otevírací kapitola 2 slouží jako hlubší úvod do dané problematiky, kde jsou mimo jiné představeny příklady základních stavebních bloků pro podobné systémy. Základy strojového učení a detektory objektů jsou popsány v kapitole 3. Zpracování videa, modelování pozadí a masky silnice je rozebráno v kapitole 4. Kapitola 5 je pak poslední teoretickou kapitolou, která obsahuje popis metod pro výběr kandidátů dopravní anomálie a zjištění intervalu a klasifikace anomálie. Před samotným testováním v kapitole 7 je v kapitole 6 uveden podrobný popis návrhu a implementace systému. Poslední kapitolou 8 je závěr.

Kapitola 2

Systémy pro detekci dopravních situací

Tato kapitola rozebírá základní problematiku při rozpoznávání dopravních situací. Obecně lze říct, že dopravní situace by se dala klasifikovat jako veškerá činnost na silnicích. Z tohoto důvodu jsou zde vybrány pouze takové situace, které mají určité charakteristiky. Tato práce se zabývá rozpoznáváním silničních nehod nebo nebezpečným stáním. Především je důležité seznámit se s myšlenkou, proč to má vlastně smysl 2.1. Většina těchto systémů má podobnou architekturu, která je rozebrána v 2.2.

2.1 Proč je to důležité?

Postupem času s novějšími technologiemi vznikají tzv. chytrá města, za účelem vylepšení infrastruktury a snížení spotřeby energií, ve kterém tvoří velkou část automatické monitorování dopravy. Policie ČR k roku 2021 evidovala 99 332 dopravních nehod na území České republiky, z toho 470 osob bylo usmrceno a celkově 22205 zraněno. U vážnějších zranění pak může záležet život dotyčné osoby na vteřinách příjezdu záchranné pomoci. Při nebezpečném stání na silnici může být žádoucí takovou situaci nahlásit na dispečink. V případě dohledových center, které slouží pro monitorování stanovených oblastí, může takovýto systém výrazně usnadnit práci zaměstnance, který je schopen monitorovat pouze omezený počet záznamů. Automatizováním takovéto činnosti se pak například zaměstnanci v dohledových centrech mohou zabývat jinými, více žádoucími činnostmi.

Obr. 2.1: Příklad nebezpečného stání na silnici (snímek je součástí videa z datové sady Iowa DOT [23])

Počet kamer na silnicích stále přibývá a nasbíraná data se spíše hromadí, než že by se reálně využívala. Je to z důvodu, že je stát České republiky ještě neumí dobře analyzovat. Současně analyzovat velký počet videí s detektorem objektů je hardwarově náročné a s tím i finančně. Z tohoto důvodu je v této práci, mimo jiné, kladen velký důraz na rychlost zpracování snímků.

2.2 Základní stavební bloky systému

Detekci dopravních situací můžeme řešit obecně jako detekci neobvyklé situace, tzv. anomálie, kterou následně klasifikujeme do určitých tříd. Rozpoznání anomálie z videa je již dlouho let

otevřený problém, jehož přístupy k řešení se rok od roku zlepšují. Přístupů k řešení je mnoho, například [1] rozdělí obraz do částí, které jsou monitorovány. Každá část je pak zodpovědná za předpovídání anomálie. Nerozpoznávají se zde objekty, jde pouze o statistické vyhodnocení na základě dočasně uložených dat, které můžou reprezentovat například směr či rychlost. Tento algoritmus však počítá pouze s dočasně uloženými daty a nepočítá s prostorovou představou anomálie. Přístupy, které se snaží vylepšit tento přístup fungují na principu časoprostorové reprezentace pixelů [6].

Jiné přístupy pak například používají sledování trajektorie objektů, to však může být při výskytu vice objektů nepřesné a časově náročné. Práce [19] dokonce detekuje anomálii využitím modelování pozadí MOG (Mixture of Gaussians). V posledních letech se rapidně zlepšily detektory objektů, jenž umožnily konstruovat algoritmy pro analýzu situací na jejich základě. Ačkoliv detektory objektů existují už dlouhou dobu [32], nefungovaly na principu hlubokého učení, a tak pracovaly s jistým omezením, jako například detekce pouze lidské tváře. Současně tak dosahovaly nižší přesnosti. Z důvodu pomalé rychlosti zpracování by dřív takovéto systémy nebyly možné, ale díky novějším verzím tvoří detektor objektů jejich důležitou část. Systémy se většinou skládají z následujících částí.

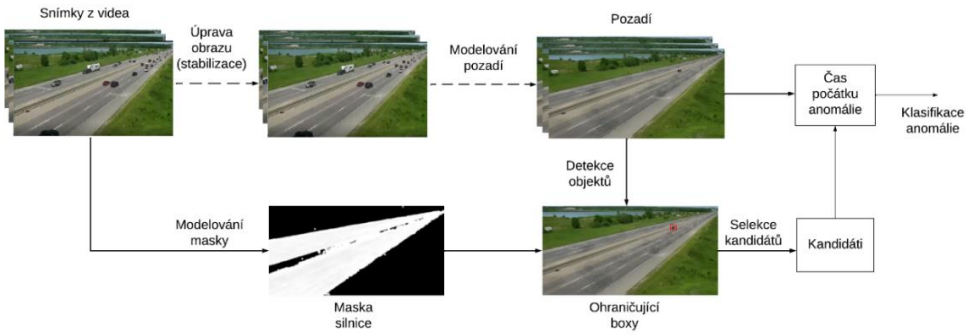
Detektor objektů. Používá se pro určení oblasti, ve které se nachází objekt, patřící do určité třídy. Nejčastěji je jeho výstupem střed, rozměry ohraničující oblasti, pravděpodobnost, na kolik si je systém jistý, zda se na snímku nachází objekt a pravděpodobnost patřičné třídy. Detektory objektů budou podrobněji popsány v 3.3.

Modelování pozadí. Výstupem je množina pozadí, indexovaných s jistou periodou. Obecně lze říct, že se jedná o potlačení dynamických částí v sekvenci obrázků a posilnění statických.

Modelování masky. Protože se dopravní nehody stávají na silnici, je důležité vymodelovat masku silnice. Tímto způsobem lze eliminovat prvky (například vozidlo stojící na zatravněné ploše mimo vozovku), ležící mimo silnici.

Úprava videa. S natáčením venkovní oblasti vzniká spoustu externích faktorů, které můžou ovlivňovat kvalitu snímků. Mezi nejčastější pak patří třepání kamery (silný vítr), nečistota nebo kapky vody na obrazu, šum nebo nesprávná aktualizace snímku.

Tyto části jsou nedílnou součástí a vyskytují se v téměř každém přístupu k řešení [34, 10, 5]. Následuje výběr potenciálních bodů, vystopování intervalu anomálie a její klasifikace. V těchto bodech se jednotlivá řešení liší nejčastěji. Například [5] používá matici informací o stavu pixelů v čase a následně porovnáním s určeným prahem vyhodnotí, zda se například nejedná o vozidlo, které pouze dlouho čeká na červenou. Obrázek 2.2 schovává výběr kandidátů do černé skříňky, je zde pouze znázorněno obecné schéma na bázi výše uvedených metod. V práci [30] je představen příklad systému, který funguje odlišně, a to na základě hlubokého učení shluků sekvencí (pozitivní, negativní příklady), avšak takovéto systémy vyžadují obrovské množství dat a učení by zabralo dlouhou dobu, protože se jedná o videa.



Obrázek 2.2: Příklad systému pro detekci anomálie a klasifikace

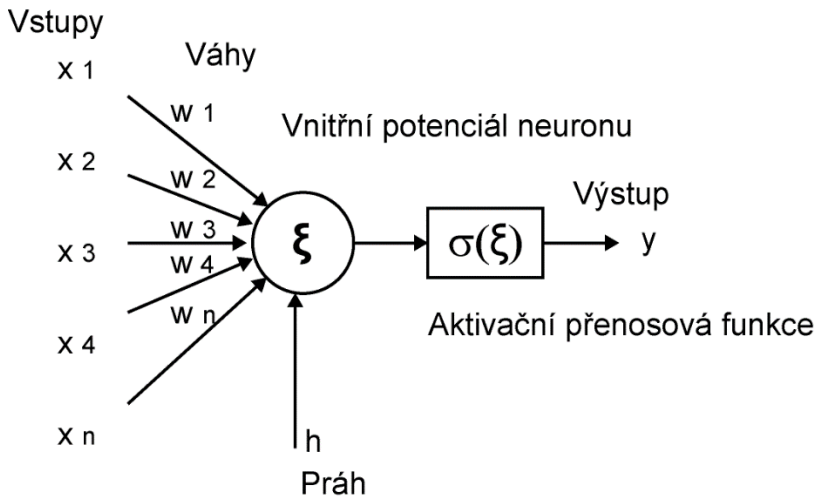
Kapitola 3

Detekce objektů

Je nezbytné zjistit oblast, kde se anomálie vyskytla. To umožnují různé modely detektorů, mezi nejznámější patří YOLO, SSD nebo modely založené na R-CNN. Jejich výstupem je ohraničující oblast kolem detekovaného objektu s pravděpodobností a třídou, do které patří. Nejprve jsou v této kapitole rozebrány teoretické základy pro detektory objektů, které budou detailněji rozebrány v podkapitole 3.3.

3.1 Strojové učení

Systémy, které dynamicky mění svoji konfiguraci na základě vnějších podmínek jsou založeny na metodách z lineární algebry, statistiky a pravděpodobnosti a diferenciálního počtu. Vytváří tak teoretické základy pro složitější systémy, které jsou na bázi hlubokého učení. Mezi základní úlohy patří: klasifikace a regrese. Klasifikace je z matematického hlediska zobrazení z množiny X do množiny Y, kde prvky množiny Y by mohly být například názvy tříd. Regrese je zobrazení z množiny X do číselné hodnoty *y,* která značí odhadovanou hodnotu [22]. Původní inspirací byl princip fungování neuronu, který při poslání elektrického impulzu jiné buňce zesílí toto nervové spojení.



Obrázek 3.1.1: model umělého neuronu (převzato z https://cs.wikipedia.org/wiki/Umělá_neuronová_síť)

Umělý neuron. Jedná se o matematickou funkci, která je inspirována neuronem. Provádí skalární součin vektorů **x**= [x_1, x_2, …, x_n] a w= [w_1, w_2, …, w_n], který je definován jako:

$$(1)\quad \mathbf{x} \cdot \mathbf{w} = \sum_{j=0}^{n} \boldsymbol{x_j} \cdot w_j$$

Protože x_0 je většinou připsaná hodnota 1, tak w_0 se nazývá tzv. bias, proto pak skalární součin začíná od indexu 1 a k celkovému výsledku je přičten bias *h*. Výsledek téhle operace je vstupem pro přenosovou funkci, která může být definovaná mnoha funkcemi.

$$(2)\quad y = \xi(\sum_{j=1}^{n} \boldsymbol{x_j} \cdot w_j + h)$$

Nejjednodušší reálný model sítě skládající se z umělých neuronů je Perceptron, který obsahuje pouze jeden umělý neuron. První zmínka o Perceptronu pochází již z roku 1958 [29]. Často se používá jako binární klasifikátor. Vstupem funkce jsou pozitivní a negativní záznamy. V případě, že model nesprávně určí kategorii, změní si svojí konfiguraci vah. Funkce tedy konverguje ke globálnímu minimu a úspešnost klasifikace záleží na vstupních datech.

Přenosová funkce. Mapuje výstupní hodnotu z váženého součtu na jinou hodnotu z intervalu ξ. Těchto funkcí je poměrně mnoho, jejich výběr je komplikovaný a výrazně ovlivňuje rychlost učení a přesnost odhadu. Existují dokonce způsoby, jakým automaticky nastavit vhodnou aktivační funkci [3]. V této sekci budou rozebrány nejčastější typy těchto funkcí.

Skoková funkce. Při použití binárního klasifikátoru, například v podobně jednoduchého Perceptronu, je vhodné použít skokovou funkci, která vrací 0 pro hodnotu menší než předem určený práh a 1 pro hodnoty vyšší.

$$(3)\ f(x) = 0, pro\ x \geq h$$
$$f(x) = 1, pro\ x < h$$

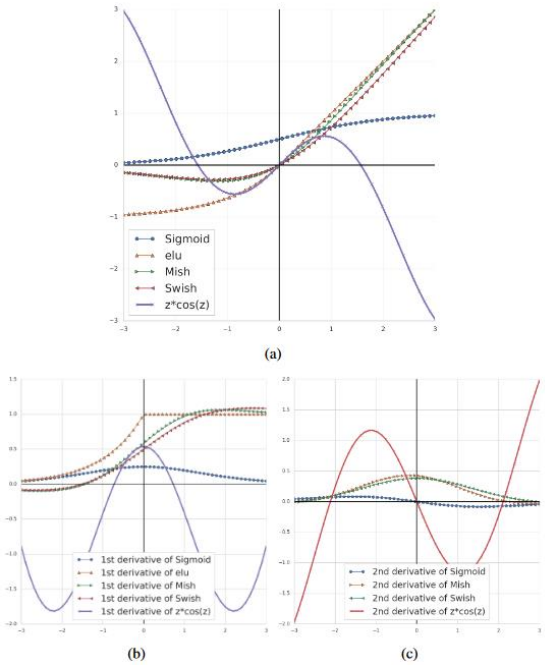
Sigmoid funkce. Mapuje vstupní nelineární hodnoty do intervalu hodnot nejčastěji mezi (0; 1) nebo (-1;1). Jedním z důsledků je, že se výstup dá vyjádřit jako pravděpodobnostní funkce. Těchto typů funkcí je více. Mezi nejčastější patří logistická sigmoid funkce, hyperbolický tangens nebo funkce arkus tangens. Zde bude podrobněji rozebrána pouze logistická sigmoid funkce, protože se v neuronových sítí vyskytuje nejčastěji.

$$(4)\ f(x) = \frac{1}{1 + e^{-x}}$$

Výše zmíněné funkce nejsou lineární. Model, který se skládá primárně z funkcí typu sigmoid narazí na problém mizejícího gradientu. Váhy jsou v modelu upravovánu po směru negativního gradientu [25] a proto vlivem menší velikosti gradientu jsou parametry modelu upravovány pomaleji, a dochází k pomalejšímu učení. Hlouběji bude tento problém rozepsán v této kapitole níže. Tento problém částečně eliminuje funkce ReLu (Rectified linear unit), která má biologicko-matematické podložení [3].

$$(5)\ f(x) = \max(0, x)$$

Pro průměrně velký svazek dat (batch size) [17] produkuje funkce derivaci rovno 0, tedy pro záporné hodnoty, a derivaci rovno 1 pro kladné, to umožnuje sestupování gradientu pokračovat i při širších vrstvách. ELUs, ReLUs, LReLUs, SELU jsou další příklady funkcí, které optimalizují problém mizejícího gradientu. Bližší informace o těchto funkcí lze najít v [3]. I přes to, funkce tohoto typu jsou v dnešní době nejpoužívanější, [25] představil novou aktivační funkci GCU v roce 2021 definovanou jako: C(z) = z · cos(z), která má eliminovat nedostatky ReLu (popsané v [25]).



Obrázek 3.1.2 Srovnání výpočetní složitosti funkcí (převzato z [25])

Vrstva neuronové sítě. Část neuronové sítě, která přijímá informace z předešlé vrstvy, provede transformaci a odešle informace do vrstvy další. Skládá se z neuronů a každý neuron je napojený na všechny neurony z předchozí a následující vrstvy. Výjimkou je první a poslední vrstva. Rozdělují se podle transformací, které provádí.

Neuronové sítě. Posloupnost různých typů vrstev. Cílem těchto systémů je najít takovou vnitřní konfiguraci, aby se dosáhlo co nejmenší chybovosti. Z matematického hlediska se jedná o hledání globálního minima funkce se spoustou proměnných. Modely se učí najít vysokou míru abstrakce v datech a zjišťuje nelineární vztahy mezi vstupem a výstupem. Využívají se v odvětvích jako počítačové vidění nebo například zpracování přirozeného jazyka.

Ztrátová funkce. Určuje reálnou hodnotu, která reprezentuje ztrátu mezi predikovanou hodnotou a požadovanou hodnotou. Na základě výsledku z této funkce model upravuje svoje parametry, tak ,aby výsledek ztrátové funkce byl co nejmenší. Mezi nejznámější patří:

Střední kvadratická chyba. Vyjadřuje chybovost podle střední hodnoty druhé mocniny rozdílu mezi odhadovanou a skutečnou veličinou. Používá se většinou při lineární regresi, kde výsledkem je číslo.

$$(8)\ MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y'_i)^2$$

Křížová entropie. Nejčastěji se používá při multi-klasifikačním modelu neuronové sítě. Vypočítává chybovost mezi obecně n-pravděpodobnostními funkcemi. Každá klasifikační třída díky softmax funkci je rozdělení pravděpodobnosti [16].

$$(9)\ \mathrm{CEF} = -\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log\left(\mathrm{h}\theta(x_i)_j\right)$$

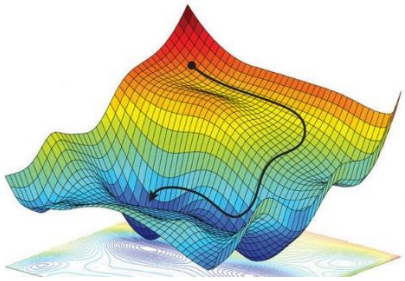
Zde bylo zmíněno pouze pár funkcí, samozřejmě jich je mnoho a naleznou mnoho uplatnění pro různé datové sady. Jiné, nejčastější ztrátové funkce pro regresní ztráty jsou MSE, L1, MBE, u klasifikační ztráty pak může být příkladem SVM.

Učení neuronové sítě s učitelem. Příchozí data modelu jsou anotována, aby model poznal, že udělal chybu při odhadu. Pro dobře navrženou síť obecně platí, že čím více trénovacích dat, tím lepší přesnosti dosáhne. Protože by nebylo vhodné dát na vstup modelu celou datovou sadu na jednou (nedostatek paměti, pomalejší trénování) a zároveň při malém počtu vstupních dat (například 1) by gradient dosahoval nižší přesnosti, zavedl se tzv. „batch-size“, tedy velikost

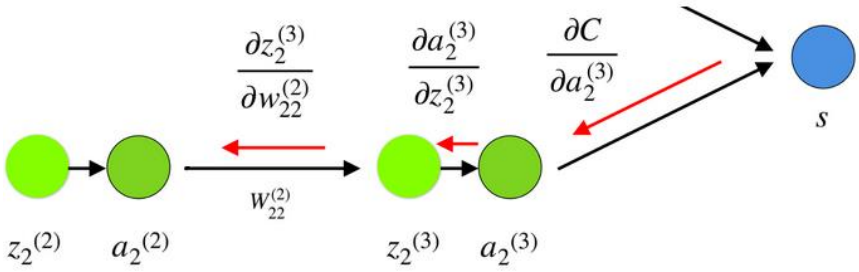
svazku, která může být libovolná. Většinou se volí na základě ideálního počtu (empiricky změřeno), nebo je velikost limitovaná stanicí, kvůli nedostatku operační paměti.

Stochastický gradientní sestup (SGS). Iterativní, optimalizační algoritmus pro hledání globálního minima funkce za podmínky, že je funkce diferencovatelná a částečně diferencovatelná. Iterativně se postupuje ve směru záporného gradientu vynásobeným koeficientem rychlosti učení, na jehož volbě silně závisí. Při malých hodnotách je zapotřebí mnoho iterací a učení trvá déle. Naopak u vysokých hodnot může dojít k nesprávnému určení minima, protože by se mohlo sestoupit o příliš velkou hodnotu.

Zpětná propagace chyby. Algoritmus, který řetězcovým pravidlem upravuje jednotlivé váhy modelu směrem od konce, tak aby se dosáhlo co nejmenší chybovosti. Váhy a prahy modelu jsou nejprve náhodně určeny. Gradient se postupně spočítá násobením parciálních derivací podle hodnot v jednotlivých vrstvách.



Obrázek 3.1.3: Příklad sestupu gradientu (převzato z https://easyai.tech/en/ai-definition/gradient-descent/)



Obrázek 3.1.4: Příklad zpětné propagace chyby (převzato z https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd)

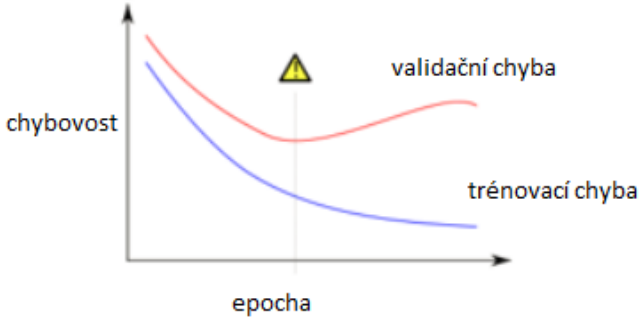
Problémy neuronové sítě. Tyto problémy obecně platí pro všechny typy neuronových sítí a je důležité je zmínit, protože novější detektory objektů uplatňují různé prvky, které je eliminují.

Problém mizejícího gradientu. Jak lze vidět na obrázku 3.1.4, jednotlivé váhy jsou upravovány parciálními derivacemi ztrátové funkce s ohledem na aktuální váhu. Přenosové funkce typu sigmoid jsou v rozsahu (0, 1). Postupně násobením čísel v tomto rozsahu se velikost gradientu snižuje exponenciálně podle počtu vrstev. V prvních vrstvách tedy může být velikost gradientu skoro nulová a může dojít až k úplně nesprávnému chování modelu.

Overfitting. Model může najít takovou aproximaci funkce, která až příliš odpovídá datům, na který byl naučen. Při odhadu dalších dat bude v důsledku přeučení modelu docházet k nesprávným výsledkům. Neodhadne tedy přibližnou funkci, podle které by se řídil, ale ideální funkci, která je více komplexní. Existuje více způsobů řešení.

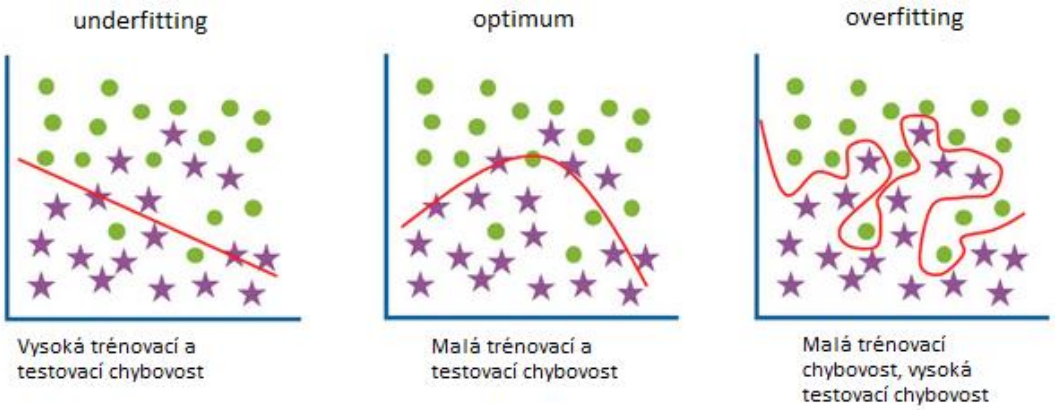
Rozdělení datové sady – vznikají dvě množiny dat – trénovací a validační. V každé epoše (fáze, kdy se trénuje a následně validuje, pro dosáhnutí lepšího výsledku jich je více) se model nejprve učí a následně se na základě validace upravují tzv. hyper-parametry (neupravují se při učení).

Předčasné zastavení – na obrázku 3.1.5 je znázorněn ideální bod, ve kterém dosahuje model nejmenší validační chyby. Předčasným ukončením modelu v iteraci, kdy se začne rozlišovat validační chyba, může model dosahovat lepších výsledku. Více přístupu k řešení overfittingu je popsáno v [33].



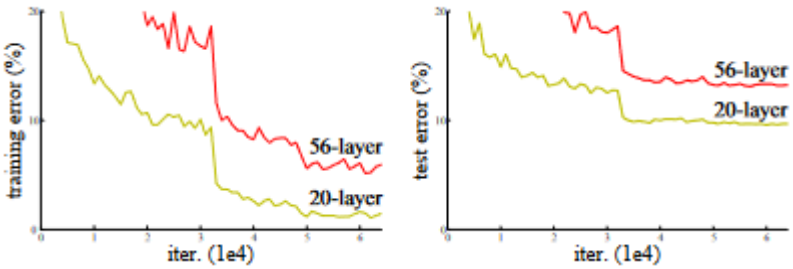
Obrázek 3.1.5: Příklad znázorňující průběh učení modelu (převzato z [33])

Underfitting. Opační efekt overfittingu. Model našel takovou aproximaci funkce, která koreluje s daty jenom do určité míry. Underfitting se dá optimalizovat délkou trénování.



Obrázek 3.1.6: Příklad underfittingu, optima a overfittingu (převzato z https://medium.com/mlearning-ai/understanding-overfitting-and-underfitting-in-layman-terms-e4c82a28e2d2)

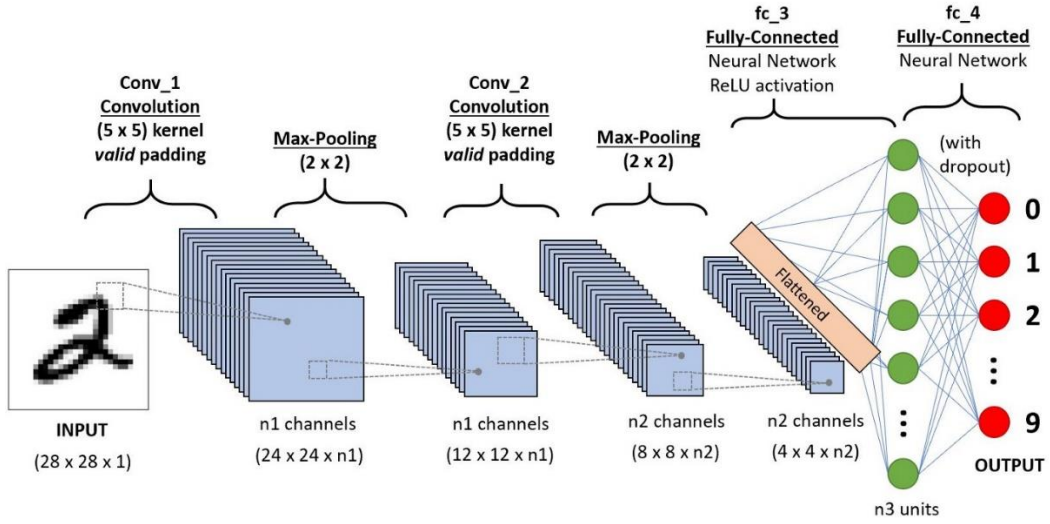
Hloubka sítě. Ačkoliv by někdo mohl očekávat, že zvětšováním hloubky neuronové sítě bude vést k menší chybovosti, práce [15] ukazuje jinak. Při zvětšování hloubky sítě se testovací chybovost zastaví a následně se začne dokonce zvětšovat. Co je ale více překvapující je, že to není důsledkem overfittingu [14, 15].



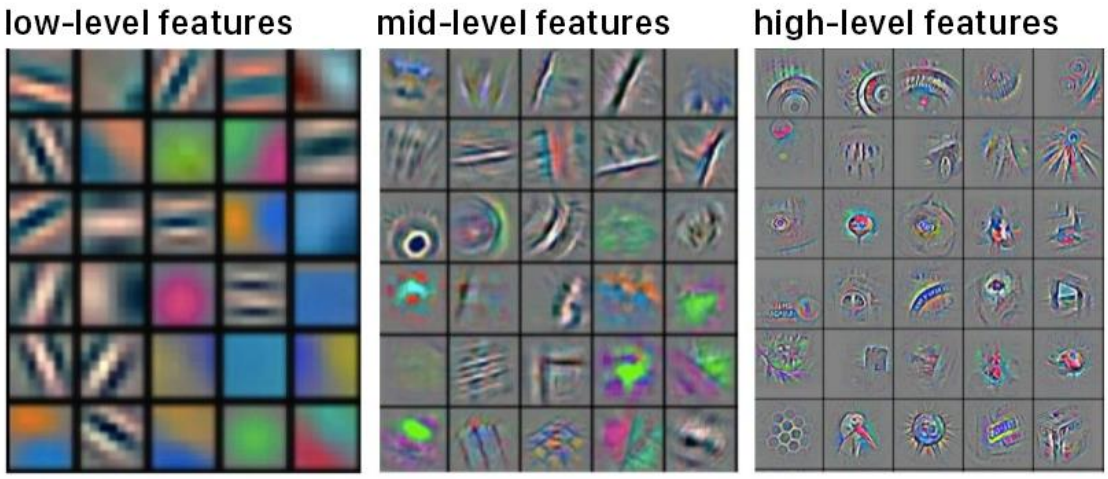
Obrázek 3.1.7: Srovnání trénovací a trénovací chybovosti menšího a většího modelu (převzato z [14])

3.2 Konvoluční neuronové sítě

Základem principu fungování jsou matice, tedy obecně obdélníkový či čtvercový tvar, ve kterým jsou na jednotlivých indexech čísla. Obrázek je matice pixelů **w** x **h** x **c**, kde c je počet kanálů. V případě šedo tónového obrazu se c = 1, pro RGB c = 3. Pixel je pak reprezentován jako 8bitové číslo. Nemusí však jít jenom o obrázek. Konvoluční neuronové sítě (CNN) pracují obecně s 2D daty, které spolu souvisí, například hudba.



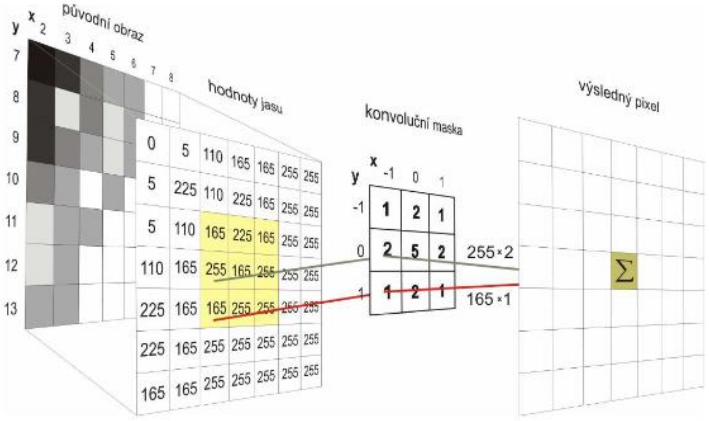
Obrázek 3.2.1: Příklad konvoluční neuronové sítě (převzato z https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)



Obrázek 3.2.2: Vizualizace skládání vlastností obrazu (převzato z https://tvirdi.github.io/2017-10-29/cnn/)

Konvoluční vrstva. Základní vrstva, která provádí operaci konvoluce. Operuje s maticemi jádro a filtr. Hodnoty v jádru jsou vynásobeny koeficienty z filtru a následně sečteny do výsledné hodnoty. Filtr se posouvá po jádře a na každé unikátní pozici se provede konvoluce. Velikost posunu může být libovolná. Technikou padding pak můžou být přidány 0 hodnoty kolem okrajů pro zvětšení velikost.

$$(6)\ (f * h)(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x-i, y-j) \cdot \mathrm{h(i,j)}$$



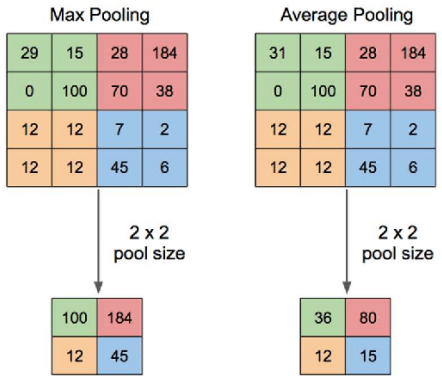
Obrázek 3.2.3: Příklad operace diskrétní konvoluce (převzato z https://cs.wikipedia.org/wiki/Konvoluce)

Rozměr filtru se může v různých vrstvách lišit. Stejně jako při klasických neuronových sítí, váhy v CNN jsou na pozicích ve filtru. Postupem zpracování dat model nastavuje hodnoty ve filtrech. Čím větší rozměr filtru, tím větší komplexnost a doba učení. Současně zde bude více map vlastností (tvz. Features), proto se využívají různé techniky pro redukci komplexnosti. Zavádí se 1 x 1 x n filtr pro redukci počet kanálů a počet vlastností obrazu. Protože filtr je tak malý nabízí se otázka, zda se vůbec jedná o konvoluci, či spíše skalární součin vektoru po hloubce **c**= [c_1, c_2, …, c_n]. Tato vrstva se vyskytuje v modelu často a pro redukci 2D dat se z pravidla vyskytuje mezi těmito vrstvami tzv. pooling vrstva.

Pooling vrstva. Redukuje počet 2D dat, tak aby se co nejvíc zachovala celková informace. Snižuje tak míru komplexnosti, zvyšuje rychlost učení a zároveň potlačuje overfitting. Nejčastěji se jedná o 2x2 pooling vrstvy, tedy nejmenší, aby se neztrácelo tolik informace. Vybraný segment je většinou jedinečný, operace tedy pracuje s nekolizními informacemi. Typů metod pro výběr nejvhodnější reprezentace informace pro daný segment je mnoho, ale nejčastější jsou dvě. Více informací o těchto vrstvách je v [13].

Average pooling. Ze segmentu je vypočítaná průměrná hodnota, tyto hodnoty se nadále mapují na stejný index. Poprvé již byla představena v roce 1990.

Max Pooling. Maximální hodnota ze segmentu je vybraná a následně projektována.



Obrázek 3.2.4 Příklad max a average poolingu (převzato z https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451)

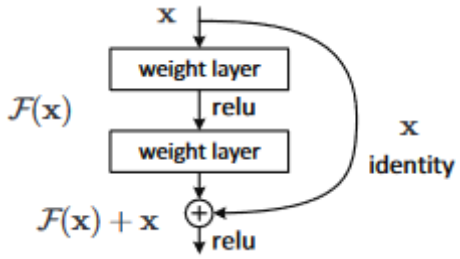
Plně propojená vrstva. Více rozměrná data se transformují do jednoho rozměru, stejně jako je tomu při běžných neuronových sítí. Bývají zpravidla jako jedny z posledních vrstev modelu, kde se zpracovávají získané vysoko úrovňové vlastnosti a jsou transformovány do výsledných hodnot. Tyto vrstvy se v podstatě učí, jak klasifikovat tyto vlastnosti a kategorizovat je tak, aby podobné vlastnosti měly lineárně podobné hodnoty (u klasifikačních modelů je výstupem třída, která by mohla reprezentovat objekt z reálného světa). Na obr. 3.2.1 se jedná o poslední 2 vrstvy.

Softmax vrstva. Poslední vrstva modelu je realizovaná jako aktivační funkce pro výstupy z plně propojené sítě. Normalizuje výstupní vektor tak, aby hodnoty byly rozdělením pravděpodobnosti. Funkce σ: $\mathbb{R}^K \rightarrow (0,1)^K$ je definována jako:

$$(7)\quad \sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \text{ pro } i = 1, \ldots, K \text{ a } \mathbf{z} = (z_1, \ldots, z_k), \text{ kde z je prvkem } \mathbb{R}^K, \text{ pro } K > 0.$$

Funkce provede exponenciální funkci nad každou složkou vektoru **z** a tuto složku podělí celkovým součtem exponenciálních funkcí. Při klasifikačním modelu je zvolena taková hodnota produkovaná touto funkcí, která má nejvyšší hodnotu, tedy pravděpodobnost.

Residuální blok. Hlubší neuronové sítě jsou náročnější na trénování, protože mají více parametrů a zároveň můžou podléhat problému degradace při zvyšování hloubky. Residuální blok využívá spojení s jednou z dalších vrstev. Tímto způsobem se můžou do modelu přidat extra vrstvy a přes přidanou vrstvu vznikne nové spojení. Jak lze vidět na obrázku 3.2.5 model se učí parametr F(x) s ohledem na vstup residuálního bloku. Tato metoda je hojně využívaná v detektorech objektů. [14]



Obrázek 3.2.5: Příklad residuálního bloku (převzato z [14])

3.3 Detektor objektů

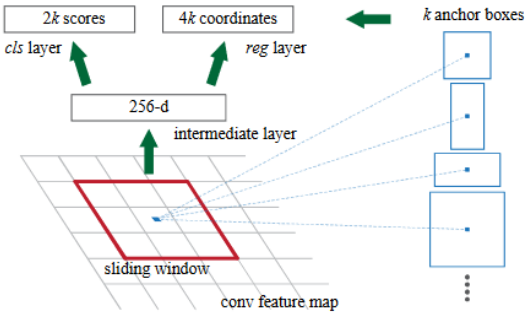
Speciální případ konvoluční neuronové sítě. Může být jak regresivního, tak klasifikačního typu. V počítačovém vidění jde o velmi důležitý prvek a jeho využití nalezneme v mnoha oborech. Například v medicíně je používán pro detekci zhoubného nádoru. Jedním z populárních projektů v automatizaci je samořídící auto, které tyto modely taktéž využívá. Mezi další příklady patří rozpoznání tváře (např. pro odemčení zámku), rozpoznání typu půdy, rozpoznání chodců atd. Ačkoliv existují detektory, které nejsou na bázi CNN, např. Viola-Jones [32], SIFT nebo HOG (histogram orientovaných gradientů) budou zde popsány pouze detektory typu R-CNN a YOLO, protože jsou ve spojitosti s danou problematikou používány nejčastěji.

3.3.1 Faster R-CNN

Vylepšení modelu R-CNN (Region-based CNN). Je rozdělen do dvou úloh, a to lokalizace a klasifikace. Klíčovým prvkem jsou navrhované oblasti v obrázku, jenž reprezentují ohraničení potenciálního objektu, který bude následně detekován. Starší modely tohoto typu využívají „selective search“ [31] pro najití navrhované oblasti, který je ale časově náročný. Stejně jako

jeho předchůdce využívá CNN pro mapu vlastností obrazu. Pro predikci navrhovaných oblastí používá tzv. Region proposal network (RPN).

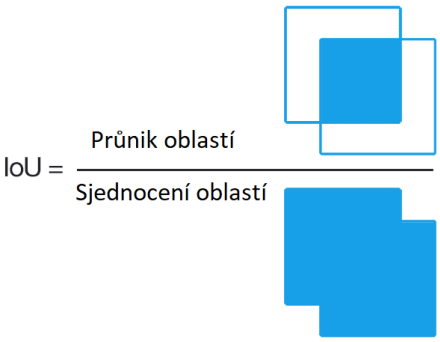
RPN. Ve snímku navrhne čtvercovité oblasti, které potenciálně reprezentují objekt, společně s pravděpodobností výskytu. Pro generaci množiny oblastí je postupně posouvané malé okénko (n x n, v originální práci [28] n=3), které je vstupem pro malou síť. Výstupem této malé sítě je vlastnost o nižší dimenzi, následně je tato vlastnost vstupem pro dvě plně propojené vrstvy, které jsou regresního a klasifikačního typu. [28]



Obrázek 3.3.1: Síť pro navrhované oblasti (převzaté z [28])

Anchors. Nad každým okénkem je RPN predikováno několik potenciálních oblastí, kde maximální počet oblastí je *k*. Na obrázku 3.3.1 je vidět, že model predikuje 2*k* pro pravděpodobnost výskytu objektu, tedy binární klasifikace a 4*k* pro souřadnice (x, y, šířka, výška). Anchor představuje potenciální ohraničující box pro daný objekt. Pro každé okénko se stanový 3 různé velikosti anchorů a pro každou velikost 3 různé rozměry (v poměrech 1:1, 1:2, 2:1), celkem tedy 9 anchorů.

Intersection over union (IoU). Podíl průniku dvou oblastí a jejich sjednocení. Protože průnik je vždy menší nebo rovno sjednocení, IoU = <0;1>.



Obrázek 3.3.2 Vizualizace IoU (převzato z https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/)

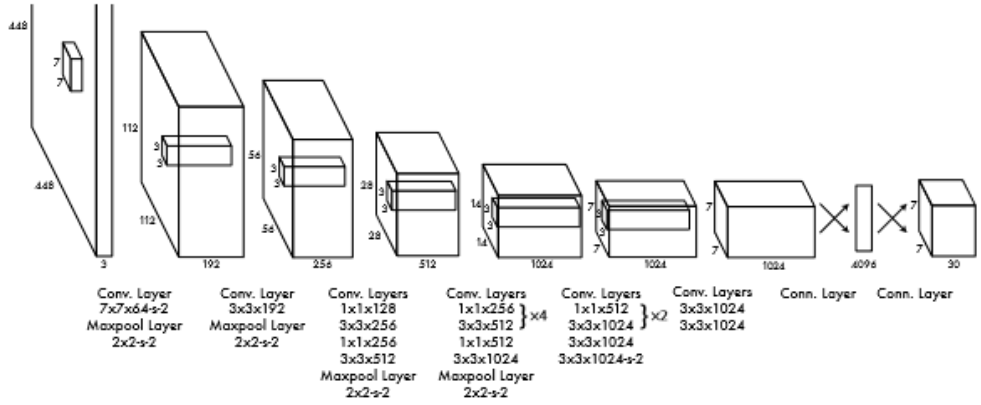
Při učení RPN sítě je anchor boxům na základě velikost IoU mezi nimi a pravým ohraničením přiřazen příznak pozitivní/negativní. Pozitivní příznak se přiřadí bud těm, kteří mají IoU větší než 0,7, nebo anchor boxu s nejvyšším IoU pro daný pravý box.

Detektory objektů na principu Faster R-CNN jsou oproti modelu YOLO pomalejší, protože ověření, zda je na obrázku objekt probíhá nad každým segmentem zvlášť.

3.3.2 YOLO

Jeden z nejpoužívanějších algoritmů pro detekci objektů. Poprvé byl představen v roce 2016 [27] a díky své vysoké rychlosti umožnil velký převrat při analýze objektů v živém videu. Jedná se o plně propojenou konvoluční neuronovou síť, a díky tomu je invariantní vůči velikost vstupního obrazu, ačkoliv pro paralelní zpracování dat (batch size) musí být zachovaná stejná

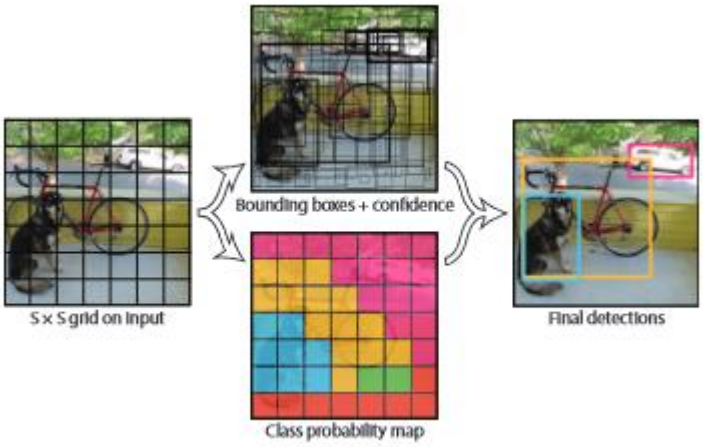
velikost. YOLO slučuje ohraničující boxy kolem detekovaných objektů a jejich třídy do jednoho regresního problému [27]. Faster-R-CNN oproti YOLO častěji chybně detekuje segmenty pozadí, protože nemá k dispozici větší kontext. Jeho rychlost závisí také na počtu detekovaných objektů a malé objekty v dálce nemusí být detekovány. Proto některé přístupy k identifikaci anomálie (např. [5]) používají Faster RCNN. Při implementaci do kamerové infrastruktury by však kvůli pomalé rychlost nemusely být dostačující.



Obrázek 3.3.4: Architektura systému YOLO (převzato z [27])

Princip. Nejprve je vstupní obraz rozdělen do mřížek o velikosti S x S. Každá mřížka předpovídá B ohraničujících boxů. Ohraničující box je definován jako množina předpovědí (x, y, w, h, c), kde:

- x, y – souřadnice středu ohraničujícího boxu s ohledem na danou mřížku
- w, h – relativní šířka a výška vůči celkovému rozměru obrazu
- c – pravděpodobnost výskytu objektu
- C – množina pravděpodobností jednotlivých tříd

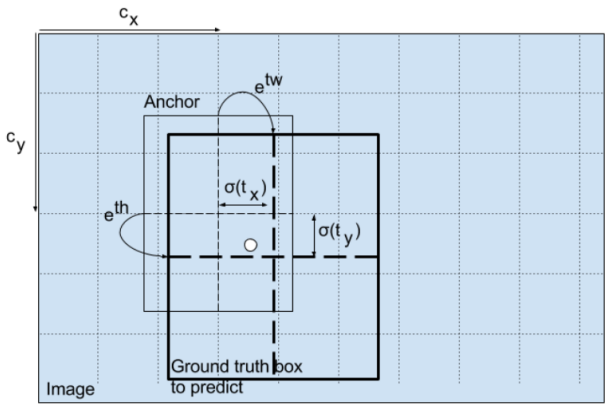


Obrázek 3.3.5: Obecný princip činnosti modelu YOLO (převzato z [27])

Výstupem ze sítě je pak tenzor o velikosti S x S x (B * 5 + C). Počet B se může lišit podle objektu, YOLO v3 používá B=3 pro každou mřížku. Mřížka má určité pole kolem sebe, kam vidí. YOLO se při trénování učí, která mřížka odpovídá danému ohraničujícímu boxu. Parametr S je vypočítán

na základě velikosti vstupního obrazu a kroku sítě, například: 416 x 416 a s krokem 32 bude S rovno 13. Mřížka nejblíže středu je zodpovědná za detekci objektu. Jelikož predikuje celkem 3 ohraničující boxy nabízí se otázka, který si zvolí.

Anchors. Predikování šířky a výšky modelem by mohlo vést k nestabilním gradientům při trénování. Zavadí se tedy posun (off-set) od předdefinovaných ohraničujících boxů nazývaný anchor. Zvolený anchor bude ten, který má nejvyšší IoU s trénovacím boxem. Jak lze vidět na obrázku 3.3.6, off-set je definovaný exponenciální funkcí. Je to za účelem lepšího trénování modelu.



Obrázek 3.3.6: Předpovídání ohraničujícího boxu na základě předdefinovaného anchor boxu (převzato z https://christopher5106.github.io)

V YOLOv2 byly zavedeny anchor boxy, které co nejvíce vystihují pravé ohraničující boxy za pomocí algoritmu K-means.

Název You Only Look Once vychází právě z toho, že oproti modelech založené na R-CNN neprochází jednotlivé segment zvlášť, ale na jednou v podobě mřížek. Jednotlivé verze YOLO se od sebe liší primárně v architektuře než v principu fungování. Například YOLOv3 využívá jako páteřní síť Darknet-53. Pro srovnání YOLOv3 přibližně pracuje v rychlosti 50FPS (z pravidla záleží na snímku, většinou však více než 30FPS) a 80.17 % MaP (průměrná přesnost), Faster R-CNN 7FPS (starší verze) a 87.60 % MaP. V roce 2020 byl představen YOLOv4, který dosahuje větší přesnosti a rychlosti než jeho předchozí verze.

Kapitola 4

Zpracování pozadí

Ve skutečnosti se dá rozpoznávání dopravní situace (v tomhle případě nebezpečné) interpretovat jako určení anomálie, něčeho, co není pro danou množinu sekvencí obrázků obvyklé. Nabízí se tedy otázka, jak zjistit, co je neobvyklé? Pozemní dopravní situaci vytváří všichni účastníci pozemní komunikace. Protože by bylo náročné počítat s tolika různými případy, je tato práce zaměřena pouze na motorové vozidla, které dokáže detektor objektů klasifikovat (auto, autobus, nákladní automobil a motorka). Při analýze videí z neobvyklých dopravních situací lze vypozorovat, že potenciálně nebezpečná vozidla jsou taková, která se eventuálně přestanou pohybovat na delší dobu. Vozidla se tedy pro určitý interval stanou součástí pozadí. Pokud detektor objektů klasifikuje vozidlo, které dříve nebylo součástí pozadí bude na něj nahlíženo jako na možnou anomálii. Anomálie nastává pouze na silnici. Pro zvýšení přesnosti systému se eliminují anomálie, které byly klasifikovány mimo silnici. Kamery, které aktivně pořizují záznamy jsou vystaveny vnějšími vlivy, jenž ovlivňují kvalitu záznamu. Dynamické změny v pozadí může ovlivňovat například silný vítr nebo déšť, a tak se úloha značně komplikuje. Stabilizace kamery potlačí šum vzniklý třepáním. Celkový proces zpracování pozadí je rozdělen do více částí:

1. Stabilizace snímků
2. Odečítání pozadí
3. Modelování masky silnice
4. Modelování pozadí

4.1 Stabilizace kamery

Protože při třepání dochází k rozdílu mezi snímky primárně na okrajích, jednodušší metody ořízly snímek v oblasti průniku všech snímků. To sice eliminuje šum, ale tato metoda by mohla oříznout vozidla. Pro zachování velikosti snímku a současně potlačení šumu se nabízí metoda založená na porovnávání daných bodech v obraze (point feature matching).

Metoda založena na porovnávání bodů. Skládá se ze tří částí:

1. Určení pohybu obrazu
2. Výpočet trajektorie pohybu a transformace
3. Aplikování transformace

Porovnávat všechny pixely mezi sebou by bylo výpočetně náročně. Pro výběr těchto bodů existuje algoritmus „Good features to track“ [12], který vybere takovou podmnožinu pixelů, u kterých bude nejlepší výsledek. Následně se vypočítá trajektorie mezi těmito body, která se využije pro výpočet transformace mezi snímky. Výsledný, stabilizovaný snímek je pak snímek, nad kterým byla aplikována vypočítaná transformace.

4.2 Modelování pozadí

Výstupem této fáze jsou snímky, nad kterými je následně spuštěn detektor objektů. Tato úloha by se dala jinak pojmout jako výpočet statického snímku v daným intervalu ve videu. Cílem je získat snímky, kdy se vozidlo stane součástí pozadí (takové, které tam dříve nebylo). Pro výpočet pozadí se provádí průměrování pixelů v intervalu. Formálně:

$$(8)\ A(x,y)^{t+1} = (1-\alpha)A(x,y)^{t} + \alpha \mathrm{I}(\mathrm{x},\mathrm{y})^{t+m}$$

Kde:

- $t \in <1*s, 2*s, \ldots, n>$, *n* je celkový počet snímku a s velikost kroku
- I (x, y) - pixel ve snímku
- A (x, y) - pixel v průměrném snímku
- m a α jsou vhodně zvolené konstanty, $\alpha \in (0;1)$, $m \in \mathbf{N}^{+}$

Rovnice (8) má charakteristiku filtru s nekoneční impulzní odezvou (IIR). Konstanta alfa zde určuje, jak moc velká váha je přikládaná předchozímu pozadí, respektive novému snímku. Při volbě malé velikosti by se nový snímek skládal z posledního snímku jenom z mála. Dopravní vozidlo, které by nebylo v provozu na kratší interval by se nemuselo na snímku vyskytnout. Naopak u vysokých hodnot by snímek mohl obsahovat i dynamické části (tedy vozidla). Průměrovat snímek po snímku by nemuselo dávat tak přesné výsledky, protože při vysokých FPS by se vozidla stávaly součástí pozadí a zároveň by to bylo výpočetně náročné. Koeficient *s* tedy udává vzorkovací frekvenci. Každý $A(x,y)^{t}$ je vstup pro detektor objektů. Před počátkem jednotlivých iterací je nejprve nastaven A (x, y) na počáteční snímek. Je důležité podotknout, že pozadí se začne modelovat až po několika iteracích, jestliže tedy dojde k anomálii právě v tomto bodě, bude přehlídnuta. Pro modelování pozadí, tak aby se v něm nevyskytovaly žádné objekty, které reálně pozadí nejsou, je zaveden dynamické přizpůsobení α. Pixely, které tvoří pozadí jsou adaptovány rychle, zatímco ty, které tvoří popředí pomalu [11, 8]. Tento princip využívá skoro každé řešení spojené s danou problematikou [34, 10, 5].

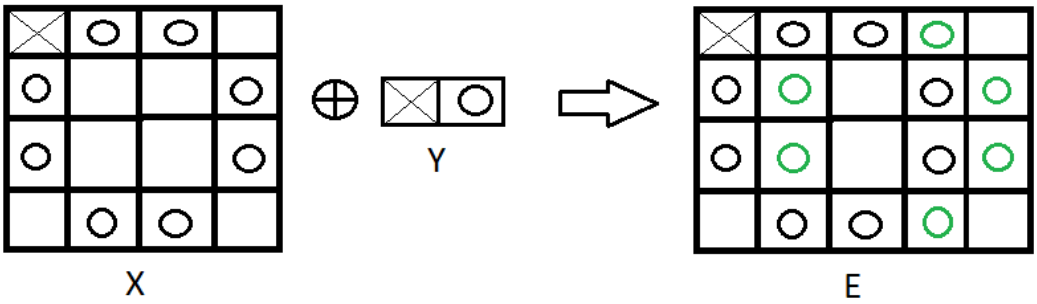
4.3 Morfologické operace

Jedná se o transformaci obrazu do jiného obrazu za pomocí tzv. strukturního elementu, který má menší velikost než zpracovávaný obraz. Strukturní element je *n x m* matice o hodnotách 0 nebo 1, která je obrazem postupně posouvána a nad každým okénkem je vykonána patřičná operace. Zároveň se specifikuje bod počátku obrazu a střed strukturního elementu. Tyto hodnoty specifikují tvar a strukturní element hraje v morfologické operaci stejnou roli jako konvoluční filtr. Základní morfologické operace jsou dilatace a eroze.

Dilatace. Komplementární operace k erozi. Dá se vyjádřit jako skládaní bodů do množiny vyjádřené jako součet vektorových složek.

$$(9)\ \mathrm{X} \oplus \mathrm{Y} = \{\mathrm{e} \in E^{2}; \mathrm{e} = \mathrm{x} + \mathrm{y}, \mathrm{x} \in \mathrm{X}, \mathrm{y} \in \mathrm{Y}\}$$

Mějme tedy příklad: $X = \{(0,1),(0,2),(1,0),(2,0),(1,3),(2,3),(3,1),(3,2)\}$
a $Y = \{(0,0),(1,0)\}$
Výsledek množiny je tedy definován součtem všech složek z množiny X se složkami z Y:
$\mathrm{X} \oplus \mathrm{Y} = \{(0,1),(0,2),(1,0),(2,0),(1,3),(2,3),(3,1),(3,2),(1,1),(1,2),(3,0),(3,3),(4,1),(4,2)\}$
Pro tento konkrétní strukturní element se vlastně jedná o sjednocení původního obrazu a stejného obrazu posunutého o 1 pixel po ose x.



Obrázek 4.3.1: Příklad morfologické dilatace

Jelikož se jedná o součet složek dvou množin, výsledný obraz bude obsahovat vždy více nebo rovno prvků množině X. Tato operace se nejčastěji používá s velikostí strukturního elementu 3x3 s hodnotami 1 a středem v bodě (1,1) pro zaplnění malých děr, zálivů a dochází ke zvětšení obrazu (pro 3x3 o 1 vrstvu). Dilatace je komutativní, asociativní a invariantní vůči posunu.

Eroze. Je to duální operace k dilataci, vyjádřená jako:

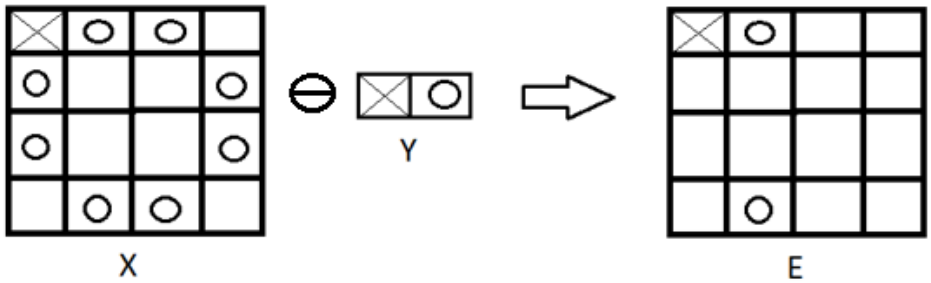
$$(10)\ \mathrm{X} \ominus \mathrm{Y} = \{\mathrm{e} \in E^2; \mathrm{e} + \mathrm{y} \in \mathrm{X}, \forall \mathrm{y} \in \mathrm{Y}\}$$

Tedy neformálně řečeno výsledkem operace jsou všechny složky x, takové, pro které platí, že všechny jejich součty s prvky z množiny Y patří do množiny X. Opačně, jak je tomu u dilatace bude počet výsledných prvků vždy menší nebo roven počtu X.

Pro stejný příklad: $X = \{(0,1), (0,2), (1,0), (2,0), (1,3), (2,3), (3,1), (3,2)\}$

a $Y = \{(0,0), (1,0)\}$

Bude výsledek: $\mathrm{X} \ominus \mathrm{Y} = \{(1,0), (1,3)\}$



Obrázek 4.3.2: Příklad morfologické eroze

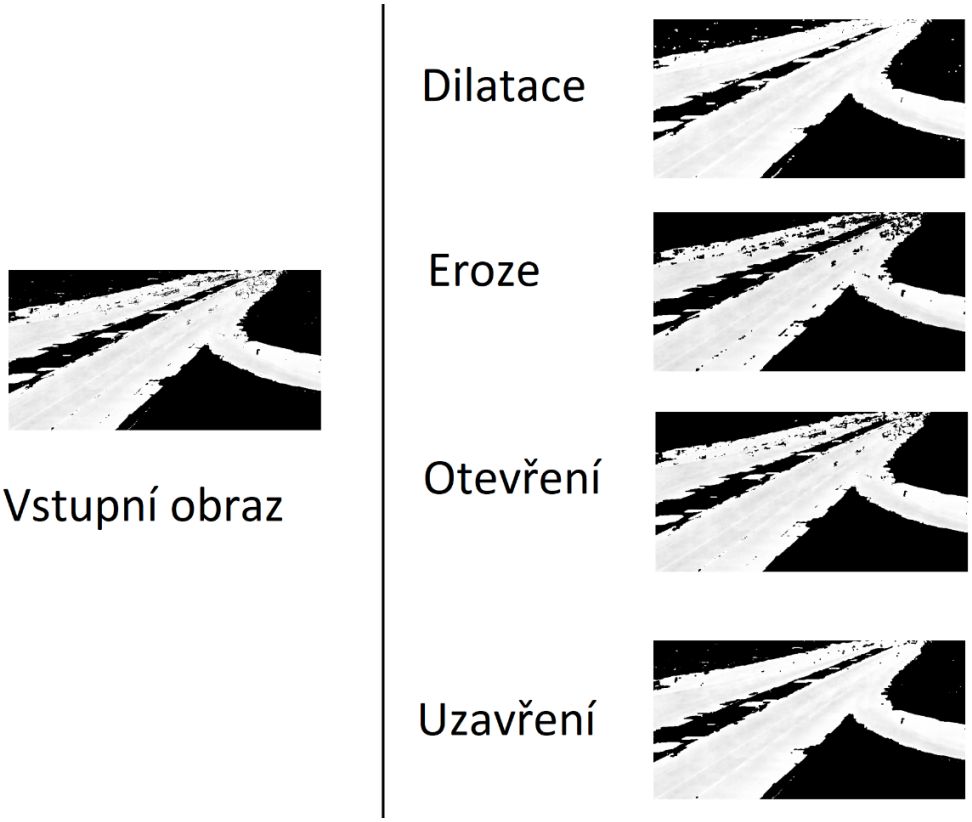
Pro erozi platí:

- Anti extenzivnost: $(0,0) \in \mathrm{Y} \Rightarrow \mathrm{X} \ominus \mathrm{Y} \subseteq \mathrm{X}$
- Invariance vůči posunu
- Jelikož není inverzní transformací k dilataci, ale duální, postupnou aplikací jednotlivých operací lze získat nový operátor

Otevření a uzavření. Morfologické operace založeny na dilataci a erozi. Otevření je transformace eroze následovaná dilatací: $X\mathrm{o}\mathrm{Y} = (\mathrm{X} \ominus \mathrm{Y}) \oplus \mathrm{Y}$. Naopak uzavření je dilatace následovaná erozí:

$$X \bullet \mathrm{Y} = (\mathrm{X} \oplus \mathrm{Y}) \ominus \mathrm{Y}$$

Otevření tedy nejprve odstraní tenké spojení, šum a následně zvětší obraz. Míra odstranění spojení a šumu je definováno velikostí matice. Uzavření naopak nejprve sloučí tence oddělené vrstvy a následně zmenší obraz. Obě operace jsou idempotentní - $f\big(f(x)\big) = f(x)$, opakované provádění stejné operace dokola již nezmění výsledek.



Obrázek 4.3.3: Příklady jednotlivých morfologických operací s velikosti matice 7x7 (pro detaily si obraz přibližte)

Tyto operace se ve spojení s odečítání pozadí používají primárně pro potlačení šumu, který byl vytvořen dynamickými změny v pozadí.

4.4 Odečítání pozadí

Jedná se poměrně o velký úkol v oboru počítačové vidění a zpracování obrazu, jehož smyslem je najít změny v jednotlivých snímcích, které ideálně budou reprezentovat objekty v pohybu. Vzniká tak maska, která modeluje ohraničení těchto objektů. Tyto objekty jsou dále zpracovány (například detektorem objektů). Vymodelování masky segmentu, kde se objekty hýbou je pak definován jejich sjednocením. Nežádoucí dynamické změny výrazně komplikují tuto úlohu, protože zavádí do obrazu s maskou šum. Různé algoritmy modelují masku jinak. Tento šum se dá do určité míry potlačit použitím morfologických operací.

MOG (Mixture of Gaussians). Cílem je oddělit popředí od pozadí. Detekovat popředí pouze z jednoho obrázku by bylo velice náročné, za použití hlubokých neuronových sítí. Pozadí tvoří takové pixely, které se o moc nezmění v čase. Pokud se najednou pixel změní příliš, nejspíše se změnil v důsledku jiného objektu. Problémem pak můžou být pixely, které se nezmění v důsledku objektu (velká světelná změna, hýbající se pozadí, kapky na kameře apod.). MOG je obecně model pro data, která se nedají vystihnout jednoduchou pravděpodobnostní funkcí, ale jejich skládáním s různými parametry. Každá pravděpodobností funkce, ze které se skládá jejich směsice se nazývá komponenta a má koeficient π', který vyjadřuje její procentuální zastoupení. Pixel v čase je právě složen z více komponent. Pravděpodobnost, že pixel leží na pozadí, je definován jako: $P(\underset{x}{\rightarrow}|BG)$. Změna světla může být buď lineární, nebo skoková. Nový snímek může obsahovat nový objekt, nebo nějaký předchozí objekt odstraňuje. Pro adaptaci na tuto změnu se daná množina snímku (nazývaná trénovací množina) rozšíří o tenhle prvek a naopak

starší snímky se vyřadí. Protože přibyl do množiny nový prvek, je nutné vypočítat pravděpodobnost, že jednotlivé pixely leží na pozadí. Zároveň může nový prvek obsahovat objekty na popředí. Pravděpodobnost, že je pixel součástí pozadí nebo popředí:

$$(11) P\left(\underset{x}{\rightarrow} \middle| \chi_T, BG + FG\right) = \sum_{m=1}^{M} \pi'_m \mathrm{N}(\underset{x}{\rightarrow}; \hat{y}_m, \sigma'^2_m I)$$

Kde v rovnici (11):

- M – celkový počet komponent
- $\hat{y}_m$ – odhadovaná průměrná hodnota
- $\underline{\sigma}'_m$ – odhadovaná směrodatná odchylka
- I – jednotková matice o stejných rozměrech jako kovarianční matice
- π' – váha jednotlivých rozdělení pravděpodobnosti, $\pi' \in < 0; 1 >$ a $\sum_{m=1}^{M} \pi'_m = 1$
- χ_T – trénovací množina

Pro nový prvek je potom hledána blízká komponenta s největším π'_m, která by ho dobře definovala. Blízký rozdělení je například takové, které má Mahalanobisovu vzdálenost menší jak 3σ. Pokud pro prvek není žádné blízké rozdělení, je vytvořena nová komponenta. Jestliže je přesažen maximální počet komponent, pak je vyřazena komponenta s nejnižším π'_m. Pozadí by pak statisticky mělo mít nejvyšší váhu. Když vstoupí do scény nový objekt, začne se modelovat komponenta a postupně se bude navyšovat její váha. Jestliže bude objekt v popředí na scéně dlouho, stane se součástí pozadí. Problémem je pak volba M, tedy počet komponent. Jednotlivá komponenta je daná patřičným shlukem dat, pro jehož řešení pak lze použít například algoritmus K means nebo metodu maximální věrohodnosti. [36].

Obrázek 4.4.1: Výsledek odečítání pozadí použitím MOG2 pro 30 snímek z videa

4.5 Modelování silnice

Obecně modelování segmentu, ve kterém se objevují dynamické prvky. Spojením s bodem potenciální anomálie se eliminují nesprávně určené body. Při kameře, která se otáčí se tato úloha velmi komplikuje. Následně zde budou popsány dvě metody, kterými tato úloha lze realizovat.

Metoda založená na MOG. Pro jednoduchost se zde bere v potaz pouze off-line zpracování snímku. To znamená, že je k dispozici vstupní video o konkrétní velikost. Naopak on-line zpracování by bylo z živé kamery, u které by byl postup komplikovanější. Postupným skládáním výsledků snímku v čase po odečtení pozadí lze vymodelovat masku. Provádět výpočty s každým snímkem ve videu je výpočetně náročně, takže se počítá s každým n-tým snímkem. Tady by při vysoké hodnotě n nebo v konkrétních případech mohlo dojít k neúplnému modelování masky, ale při běžném provozu a relativně dlouhém videu je maska postačující. Tomuto modelu lze nastavit parametr, který udává z kolika posledních snímků provádí výpočty a práh, který udává minimální vzdálenost od rozdělení, aby se bral v potaz. Jelikož se na dynamické změny přizpůsobí až po několika snímcích, může se stát, že se v obraze projeví šum, tedy pixely, které nebudou součástí vozidla. Toto lze do určité míry potlačit buď tak, že před připojením n-tého snímku do současné masky je nejprve do modelu přivedeno pár předchozích snímku, aby se

adaptoval na změny, ačkoliv velké změny se stále projeví. Druhý způsob je až ve fázi po zpracování, použitím výše zmíněných morfologických operací. Na obrázku 4.3.3 jsou demonstrovány jednotlivé operace nad výslednou maskou, kde nejlepší výsledek dává otevření.

Metoda založená na detektoru objektů. Každý n-tý snímek je vstupem do detektoru objektů. Nejprve je vymodelovaná maska, která vznikla spojením všech výsledků z detektoru objektu. Následně je provedena normalizace obrazu a binarizace, která eliminuje části, jež nejsou součástí silnice např.: vozidla na parkovišti v pozadí. Výsledkem je maska silnice. Tento přístup je principiálně lehčí než metoda založená na MOG, ale je výpočetně náročnější a závisí na kvalitě detektoru. Tento přístup je použit například v [10, 5].

Kapitola 5

Výběr kandidátů a zpracování anomálie

Kombinováním průměrného snímku v jednotlivých časových úsecích s objektem detektorů by za ideálních podmínek stačilo pro rozpoznání anomálie. Každé detekované vozidlo, které se dočasně stane součástí pozadí, bude potenciální anomálie. Vzniká zde však problém, kdy dojde k detekci vozidla, které není součástí anomálie. K těmto případům dochází nejčastěji kvůli:

- Nesprávně detekovanému vozidlu – shluk pixelů je klasifikován jako třída s vozidlem, i když objekt ve skutečnosti nepatří do této třídy
- Pomalu se pohybujícímu vozidlu – při dopravní koloně nebo delší stání u semaforu se může vozidlo stát dočasně pozadím
- Vozidlu, stojícímu mimo hlavní vozovku – v pozadí se může například vyskytnout parkoviště

Systém nemůže poznat, jestli se opravdu jedná o anomálii pouze z jednotlivých detekovaných objektů. Pro ověření zde dochází k výběru takových prvků z této množiny, které skutečně splňují kritéria. Po tomto procesu bude mít systém k dispozici snímky od <T; T+n>, kde T je počátek a T+n konec průběhu události. Poslední úlohou je tedy klasifikace dopravní situace.

5.1 Výběr kandidátů

Eliminovat vozidla, které netvoří anomálii by z jednoho snímku bylo víceméně nemožné. Sjednocením všech bodů, ve kterých došlo k detekci vzniká množina složená ze shluků. Při vážnější dopravní situaci bude vozidlo stát na místě po delší dobu. Body, které se zde vyskytují krátce se nepodílí na situaci. Pro odstranění těchto bodů je vhodný algoritmus K nejbližší soused. Konkrétně:

$$(12)\ d_P(k) \geq l; P = (X_i, Y_i)$$

Kde v rovnici (12):

- $d_P(k)$ −vzdálenost bodu P od k-tého nejbližšího bodu
- $l -$ vhodně zvolená konstanta
- k ≪ l, tedy k je mnohem menší než l
- Záměnou rovností v rovnici (12) lze odstranit špatně klasifikované body, které tvoří velmi hustý shluk v průběhu celého videa.

$$(13)\ d_P(k) \leq l; P = (X_i, Y_i)$$

Kde k >> l.

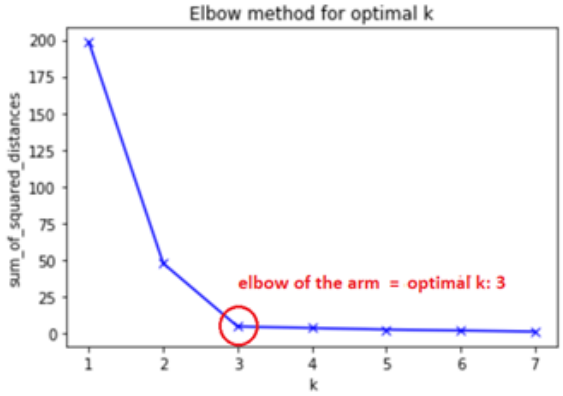
Výsledkem této operace nad množinou bodů bude podmnožina s lepšími vzorky. Každý shluk bodů se následně sjednotí použitím k-means algoritmu.

K-means. Z množiny bodů *M* najde středy shluků. Na počátku se určí *k* středových bodů, které náleží na M. Každý bod je přiřazen do shluku, jehož středový bod je nejblíže. Nové středové body se vypočítají jako průměr bodů ve shluku, a přepočítají se jednotlivé vzdálenosti bodů od nových středových bodů. Tímto způsobem algoritmus konverguje k řešení. Jakmile jsou nové středové body stejné jako v předchozí iteraci je algoritmus ukončen. Dřívější verze algoritmů vybíraly počáteční body náhodně. Při vysokém počtu bodů v jednom shluku a malým v ostatních, by mohlo docházet k vybírání počátečních bodů pouze v nejhustším shluku. Práce [4] představuje vylepšení, které nachází počáteční body, jež jsou dále od sebe:

1. Urči náhodný bod náležící na M
2. Pro každý bod, který nebyl zvolen urči vzdálenost D(x) od nejbližšího již zvoleného bodu
3. Vyber nový bod náhodně s pravděpodobností definovanou pravděpodobností funkcí $\frac{D(x_i)^2}{\sum_{n=1}^{N} D(x_n)^2}$
4. Vrat se na bod 2, pokud již není zvolen k náhodných bodů

Jednotlivé body z videa jsou získány dynamicky, takže zde není k dispozici informace o počtu shluků. Metody pro výběr čísla K jsou následující.

Metoda loktu. Algoritmus K-means se spouští s hodnotou *k* v intervalu <1; n>. Na konci každé iteraci se vypočítá metrika WCSS, která je definovaná jako součet všech vzdáleností od bodů k přiřazenému středovému bodu, na druhou. Graf těchto hodnot bude připomínat tvar loktu. V bodě, kdy se změní křivka na přibližně lineární závislost je výsledná hodnota K. Automaticky vyčíst tuto hodnotu je pro program složitější a zároveň nemusí zde vždy vyjít ideální číslo.



Obrázek 5.1.1: Graficky znázorněná metoda loktu (převzato z https://stackoverflow.com/questions/62443970/finding-the-optimal-number-of-clusters-using-the-elbow-method-and-k-means-clust)

Metoda siluet. Měří, jak moc se bod hodí do přiřazeného shluku v porovnání s ostatními. Výsledkem je index v intervalu <-1;1>. Výpočet indexu probíhá iterativně od k = <2; n>:

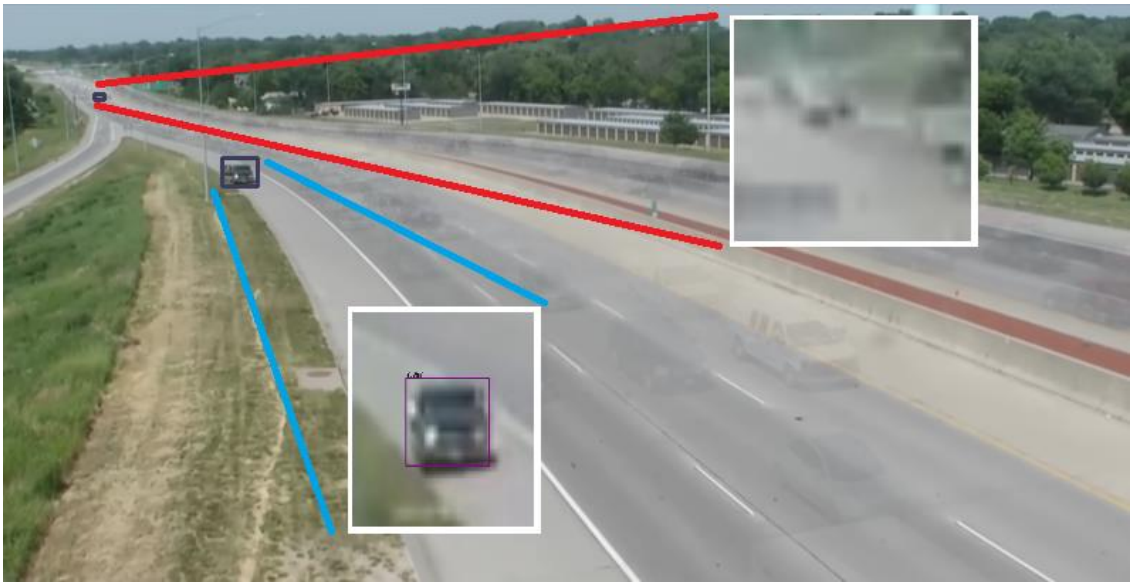
1. Pro každý prvek $i \in C_I$ vypočítej průměrnou vzdálenost *a(i)* od ostatních bodů v C_I
2. Pro každý prvek $i \in C_I$ vypočítej nejmenší vzdálenost b(i) od všech ostatních bodů, které nenáleží C_I
3. Pro každý shluk $C_I, |C_I| \geq 2$ vypočítej hodnotu siluety $s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i)}$ a s(i) = 0, pro $|C_I| = 1$
4. Zprůměruj s(i) přes všechny shluky a vrať $\check{s}(k)$

Pokud by byla hodnota $\check{s}(k)$ blízko -1, tak jsou body rozděleny do shluků velice špatně. Naopak blízko 1 jsou rozděleny adekvátně. Výsledná hodnota *k* je tedy taková iterace, ve které bude $\check{s}(k)$ nejvyšší, $SC = max_i \check{s}(i)$. Problémem této metody je, že pracuje s k>1, protože při k=1 nelze určit hodnotu b(i).

Po této operaci je znám celkový počet shluků a jejich středové body, které označují místa anomálie. Po přiložení masky silnice vznikne podmnožina bodů definovaná jako jejich průnik. I přes všechny metody pro výběr bodů se zde můžou objevit body, které budou ležet na silnici a budou mít podobné vlastnosti jako anomálie. Vzniká to v důsledku nedokonalého detektoru objektů, který špatně klasifikuje pixely jako objekty.

Lokální zvětšení rozlišení. K jednotlivým středům je nutné znát šířku a výšku zde detekovaného vozidla, který definuje oblast anomálie. Každá oblast je následně zvětšena na

patřičnou velikost pro detektor objektů a opět zde dojde k detekci objektů. Jestliže nebude objeven žádný objekt, který by představoval dopravní vozidlo, je střed odstraněn z množiny. Jedná se o jednoduchou, ale efektivní metodu, která však může být při použití detektorech, založené na R-CNN časově náročná. Zde je však nutné rozhodnout o prahu jistoty detektoru. Při menších hodnotách může i přes to dojít k chybné klasifikaci.



Obrázek 5.1.2: Příklad eliminace kandidátů použitím lokálního zvětšení

5.2 Určení intervalu anomálie

V jednotlivých shlukách je společně se souřadnicemi bodu dostupný i snímek, ve kterém se objevil. Jamile se vozidlo zastaví, začne se postupně stávat pozadím a zvyšuje se tak jeho průhlednost. V důsledku nízké průhlednosti je detekováno až po několika snímcích. Pro určení přesného snímku je vhodné použít podobnost struktury obrazu. [10]

Podobnost struktury (SSIM). Vyjadřuje hodnotu v rozmezí <-1;1>, která určuje podobnost dvou obrazů. Vstupní obrazy jsou rozděleny do okének, které jsou mezi sebou porovnávány. Porovnání je založeno na statistickém vyhodnocení pixelů. Původní algoritmus porovnává pouze šedobílé obrazy, avšak byl rozšířen i na RGB, kde je každá složka porovnávaná zvlášť. Původně byl představen v [35].

Následně jsou vybrány hraniční body podle snímku. K bodům je důležité uchovat i šířku a výšku detekovaného objektu. Tato oblast je porovnávaná se stejnou oblastí ve snímkách, které předchází nebo nadchází porovnávanému snímku, podle toho, jestli se jedná o počátek nebo konec intervalu použitím SSIM. Výsledný snímek je takový, u kterého klesne hodnota SSIM pod stanovený práh.

5.3 Klasifikace anomálie

Pro klasifikace situace z videa je přesný SlowFast realizovaný dvěma částmi. Jedna část, která operuje s vyšší periodou snímků zachytává sémantické informace a druhá část, s vysokou periodou zachycuje pohyb (např.: mávání, vrtění, třepání). [12]. Ačkoliv je tento model přesný, je více komplexní a důvodů, proč se vozidlo přestalo na silnici hýbat není mnoho. Nejčastěji nastane jeden z následujících případů:

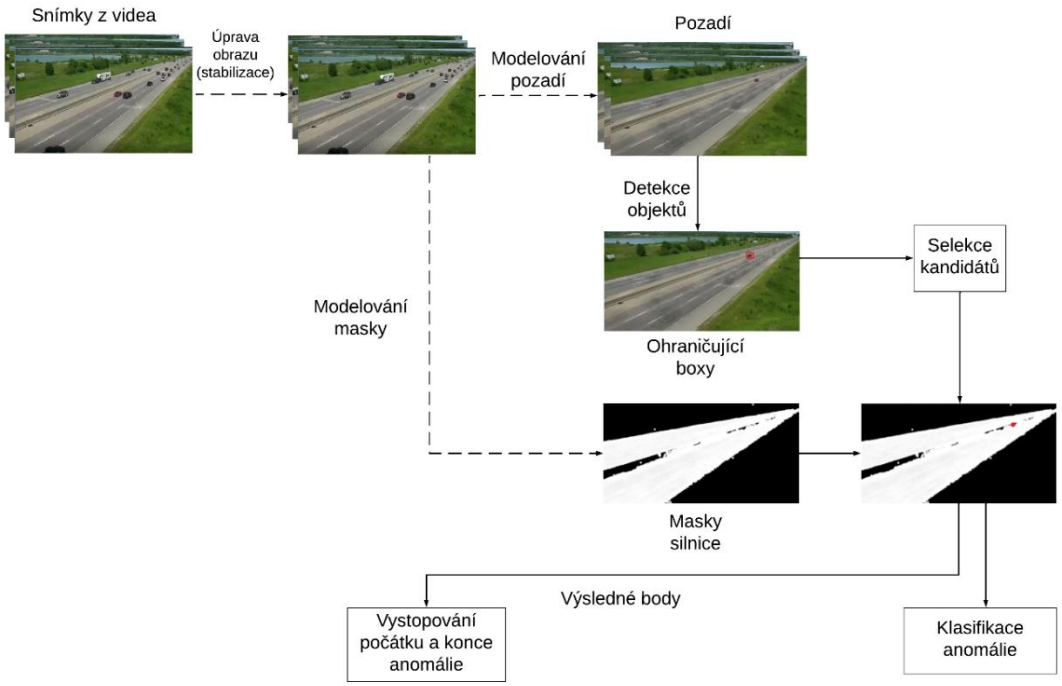
- Nebezpečné stání
- Dopravní nehoda

V nejhorším případě se vozidlo zastaví přímo v místě, kde je běžný provoz, tak, že omezí okolní vozidla. Pro zjištění, zda se vozidlo vyskytuje v silnici lze upravit masku silnice, konkrétně jí zúžit o jistou velikost. Jestliže i přes to bude vozidlo v silnici, bude situace vyhodnocena jako velmi nebezpečná a potenciálně se jedná o dopravní nehodu. V opačném případě se s nejvyšší pravděpodobností jedná o nebezpečné stání (i když dopravní nehoda je současně nebezpečné stání).

Kapitola 6

Návrh a implementace

Tato práce realizuje program, který je schopen z předem nahraného videa určit výskyt, interval průběhu a klasifikaci dopravní situace. Dopravní situací by se mohlo rozumět široká škála událostí, mezi které by mohly být i vysoce výjimečné situace. Takový program by bylo náročné realizovat, a tak bude zaměřen pouze na vybrané situace, které sdílejí stejnou vlastnost. Dopravní nehody a nebezpečné stání u silnice sdílejí tu samou vlastnost, a to takovou, že se vozidlo eventuálně přestane pohybovat. Tímto způsobem lze na program pohlížet, jako na detektor statických vozidel, v úsecích, ve které se za normálních okolností nevyskytují. Při průměrném počtu snímků za sekundu 30-60 a délce trvání videa 5-30 min bude video obsahovat celkem 9000-108000 snímků. Navíc testování musí zahrnovat spoustu videí s různými parametry. Aby nebylo třeba vysoko výkonných GPU a zároveň dlouhého času pro testování, je zvolen rychlejší algoritmus, který je inspirován z [10]. Pomalé algoritmy, které sice mají vyšší přesnost, by stejně se současnou technikou pravděpodobně nebyly možné implementovat do infrastruktury. Stavební bloku systému jsou podobné jako v 2.2:



Obrázek 6.1: Schéma principu navrhnutého programu (inspirováno schématem 2.2)

Navrhnutý program bude mít kostru programu podobnou jako v [10] s rozdílnými přístupy k řešení v některých částech a s extra částí pro klasifikaci situace. Celkově se tedy bude skládat z následujících modulů:

- Analyzátor videa a zpracování snímků
- Detektor objektů
- Selektor kandidátů

- Modul pro zpracování anomálie

Výstupem programu pak bude interval anomálie a k ní její patřičná klasifikace. Při situaci, kdy nastane více anomálií ve videu, bude ke každé vypočítán její interval a klasifikace. Oproti AI city challenge se tak bude lišit, kde se kontroluje pouze první detekovaná anomálie.

Tyto moduly budou obsahovat funkce spojené s danou problematikou a mohou využívat pomocných metod, které nejsou tolik konkrétní a mohou být řešením více problémů současně. Spouštěcí část programu pak bude sloužit k provázání výše definovaných modulů v takovém pořadí, aby by bylo dodržené navrhnuté schéma z obrázku 6.1.

Zvolený programovací jazyk je Python s Anaconda3 interpretem (Python verze 3.9.7) pro jednoduchou instalaci nových balíčků. Python obsahuje velmi širokou škálu knihoven pro ML, zpracování videa a obrazu a různé transformace. Pro ML je použit framework PyTorch, který díky své přehledné dokumentaci, rozsáhlých knihovnách a udržovatelnosti patří mezi nejpoužívanější frameworky pro ML vůbec. Urychlení výpočtů spočívá v jejich přesunutí na GPU, v takovém případě je nutné mít na koncové stanici nainstalovaný CUDA. Celková práce je pro jednoduchost realizovaná jako konsolová aplikace bez grafického rozhraní, ačkoliv by v budoucnu mohla být o tuto možnost rozšířena. Syntaxe spuštění je následující:

```
python3 anomaly_detector.py <video_name>
```

Kde `video_name` je povinný parametr se vstupním videem. Při prvním spuštění budou staženy váhy modelu, který byl naučený na datové sadě MS COCO. Následně budou v této kapitole postupně rozebrány jednotlivé moduly. Nejprve bude vždy zmíněný návrh patřičné části a následně její implementace.

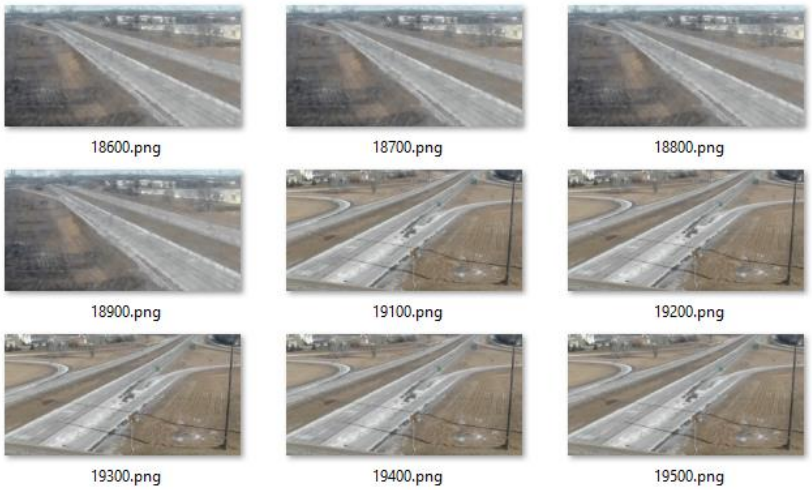
6.1 Analyzátor videa a zpracování snímků

Vstupem bude video s předem neznámou délkou, FPS, šířkou, výškou a počtem kanálů. Videem se pak bude postupně procházet a v každé iteraci bude dostupný jeden snímek z daného časového úseku. Protože kvalita nahraného videa z dopravních silnicích může být libovolná, a kvůli problémům popsané v kapitole 4.1, je důležité snímek stabilizovat. Současně s analyzátorem bude probíhat modelování pozadí a silnice. Při vyšších FPS není nutné procházet každý snímek, protože většina snímků nese velmi podobnou informaci. Stejně jako v práci [10] je použit velikost kroku s = 100, tedy každý 100. snímek je zpracován. Průměrováním obrazu postupně vzniká pozadí v každém 100. stabilizovaném snímku, ale prvních několik zaznamenaných pozadí není dostatečně zprůměrováno. Jakmile se analyzátor dostane na poslední snímek, začne zpětný chod pro vytvoření prvních pozadí, které nebyly vymodelovány správně. Zpětné průměrování začíná na 30 snímku a postupuje až po první snímek. Nad každým stabilizovaným snímkem je pro výpočet dynamických prvků ve scéně aplikován algoritmus MOG, a jejich sjednocení udává masku silnice. Součástí použité datové sady jsou videa, ve kterých kamera změní svůj úhel a začne snímat jiný obraz. Adaptaci na tento problém pak bude více masek silnice v jednotlivých časových úsecích definovanými změnou kamery. Při změně kamery nastane rozdíl ve struktuře obrazu (5.2). Pro každý interval se pak modeluje pozadí a maska zvlášť, tedy zpětný chod modelování prvních snímků pozadí pak končí v počátku tohoto intervalu. Pro zjištění masky silnice, kterému bodu anomálie náleží budou uchovány intervaly. Kvůli vysoké velikosti snímků se při běhu programu vytvoří nová složka se stejným názvem jako je název videa, do které se následně uloží jednotlivé snímky pozadí, vymodelované masky silnice a textový soubor s intervaly.

Pro analýzu videa je použita knihovna OpenCV, která současně obsahuje algoritmus pro odečítání pozadí (MOG2). MOG2 je zde nastaven tak, aby prováděl výpočty z posledních 25 snímků s prahem 100, bez stínů, a ještě před průchodem videa je naučen na prvních 25 snímcích. Snímky z dopravních kamer můžou obsahovat silný šum, které by ovlivňovaly vypočítané hodnoty. Pro jejich identifikaci je pak prvně určen snímek, který šum neobsahuje

tak, že zde budou detekovány objekty s jistotou výskytu alespoň 70 %. Při každé iteraci s velikostí kroku 100 je pak vybrán první snímek v pořadí, který nebude mít hodnotu ssim blízkosti 0 (pozn.: funkce pro výpočet ssim v Pythonu je v rozmezích <0;1>, nikoliv <-1;1>). Tyto hodnoty byly především empiricky zjištěny pro nejlepší výsledky. Při delším videu je lepší čerpat data z více snímků. Obecně lze říct, že s rostoucím prahem změny algoritmu pro odečítání pozadí bude do určité velikosti model masky přesnější, bohužel při velikosti kroku 100 zde nebude dostačující počet dynamických prvků pro modelování. Na druhou stranu nižší práh pomůže s modelováním silnic, na kterých se objevilo málo vozidel. Stíny jsou v MOG algoritmu vypnuty, protože nejsou pro modelování potřebné. Pro lepší výsledky po odečtení pozadí je na stabilizovaný snímek aplikováno Gaussovsské rozostření s velikostí matice 7x7.

Výsledná maska silnice je transformována užitím morfologických operací, konkrétně uzavření a otevření s velikostí matice 7x7 a následnou 15x15 dilatací pro vyplnění a zvětšení masky. Stabilizace snímku je zajištěna knihovnou VidStab, volně dostupnou z GitHubu[1]. Snímky jsou průměrovány podle rovnice (8) s $\alpha = 0{,}1$ a $m = 30$. Pro výpočet SSIM je použita knihovna skimage.metrics. Práh pro SSIM je nastaven jako polovina průměrné hodnoty ssim. Poté, co klesne hodnota ssim pod stanovený práh, vypočítá se snímek, kdy dojde opět ke stabilizaci kamery, a to tak, že hodnota ssim bude opět vysoká. Změna pohledu kamery může nějakou dobu trvat, a tak může být pár snímků pozadí vynecháno. V implementaci se počítá s offsetem u snímku, protože stabilizátor vrací vždy n-tý poslední stabilizovaný snímek (zde n=25). MOG2 se automaticky přizpůsobuje na dynamické změny (voda, špína na kameře), ale protože se zde bere každý 100. snímek, nemusí se adaptace projevit. Časově náročnější varianta by pak mohla počítat s více snímky, tak aby se maska stihla přizpůsobit před samotným uložením pozadí. Při realizaci je postačující odstranění těchto bodů použitím morfologických operací.



Obrázek 6.1.1: Příklad uložení pozadí v úseku změny pohledu kamery (19000. snímek je záměrně vynechán, protože by neobsahoval snímek silnice)

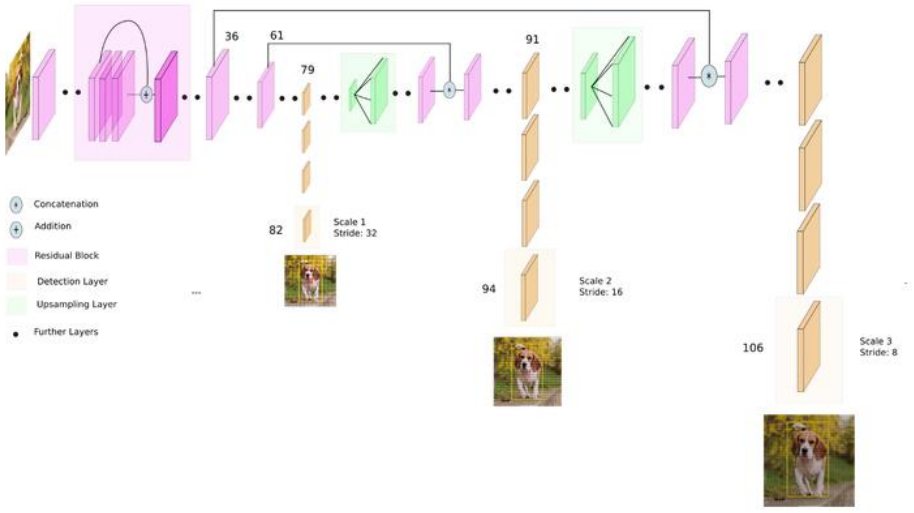
Obrázek 6.1.2: Adekvátní masky silnice k obrázku 6.1.1

[1] https://github.com/AdamSpannbauer/python_video_stab

Intervaly pro obrázek 6.1.2 by pak byly <0; 18900>, <19100; 26789>

6.2 Detektor objektů

Za účelem vyšší rychlosti je v této práci zvolen model YOLOv3, který má velmi vysokou rychlost zpracování a relativně vysokou přesnost odhadu. Je založen na sítí Darknet, která měla původně 53 vrstev. Pro lepší detekci obsahuje YOLOv3 navíc dalších 53 vrstev. Aplikuje 1x1 konvoluci napříč třemi různými velikostmi vlastností ve třech různých místech. Používá křížovou entropii jako ztrátovou funkci a pro jistotu výskytu objektu a predikci distribuci tříd logistickou regresi. Váhy pro tento model budou převzaty z předem naučeného modelu na datové sadě MS COCO, která obsahuje 80 klasifikačních tříd. Pro potřeby této práce jsou však dostačující třídy auto, motorka, autobus a nákladák. Strukturu modelu a jeho hyper-parametry definuje konfigurační soubor s příponou .cfg. Parser následně vytvoří model podle konfiguračního souboru ve frameworku PyTorch. Model bude přijímat obraz o předem stanovené velikosti a jeho výstupem bude tenzor o velikosti m x 85, kde m je počet detekovaných objektů. Zároveň bude volat pomocné metody definované v jiném souboru, mezi kterými bude například NMS algoritmus pro výběr nejlíp reprezentující oblasti pro daný objekt. Kód detektoru bude převzat ze stránky GitHub[2], kde je publikovaný jako open-source, s drobnými úpravy v některých funkcích.



Obrázek 6.2.1: Architektura modelu YOLOv3 (převzato z https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b)

Analyzátor videa vrátí cestu k vytvořenému adresáři. Pro snadný průchod při zpracování jsou jednotlivá pozadí očíslována podle snímku, ve kterém byly vytvořeny. Kvůli detektoru objektů jsou pozadí transformována na velikost 416x416 a z BGR jsou převedeny na RGB. Pokud je to možné, převede se výpočet na GPU. Objekty na pozadí jsou detekovány detektorem. Z celkových 80 čísel identifikující třídy je zvolena třída s maximální hodnotou. Jestliže tato třída nespadá do množiny stanovených tříd je množina popisujících vlastností odstraněna. Současně jsou odstraněny takové množiny, které mají procentuální jistotu výskytu objektu menší než 10 %. Takhle nízké číslo je zde zvoleno z důvodu vzdálených objektů, které by jinak nemusely být brány v potaz. I přes to nejsou některé velmi vzdálené objekty detekovány. Současně se zvyšuje riziko nesprávné detekce, jehož řešení popisuje kapitola 5.

[2] https://github.com/ayooshkathuria/YOLO_v3_tutorial_from_scratch

NMS je implementován rychlejší verzí, která optimalizuje jeden vnořený cyklus a výstupy z detektoru, které mají IoU větší než 0,7 jsou eliminovány. Detekované souřadnice a rozměry jsou přidány do seznamu společně se snímkem, ve kterém došlo k detekci. Selektoru kandidátů stačí pro výpočet pouze souřadnice, proto jsou rozměry odděleny do jiného seznamu. Indexy mezi těmito seznamy jsou navzájem korespondující.

6.3 Selektor kandidátů

V seznamu jsou k dispozici jednotlivé body, reprezentující detekci ve snímku s pozadím. Předpokládá se tedy, až na výjimky, že se jedná o vozidla, které tvoří anomálii. Selektor by pak měl být zodpovědný za výběr takové podmnožiny bodů, které doopravdy reprezentují anomálii. Takové body by měly mít následující vlastnosti:

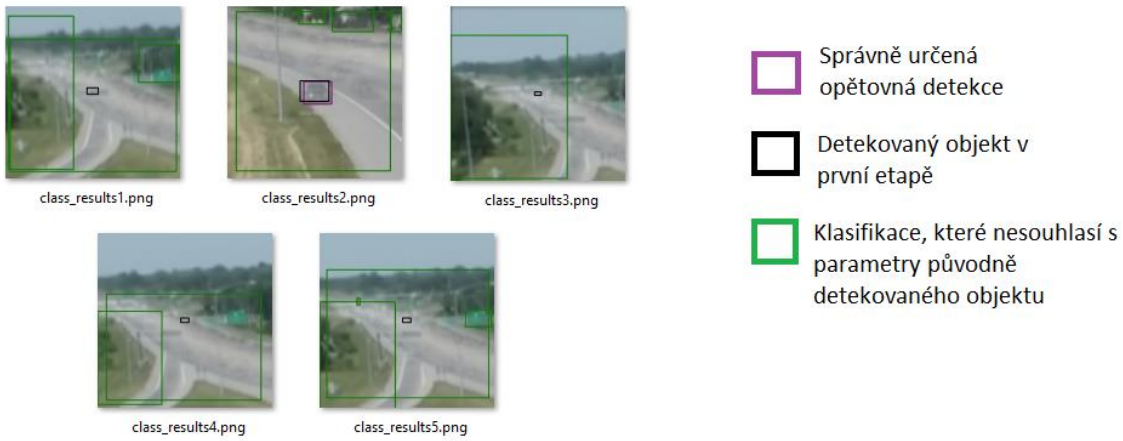
- Tvoří shluk s malou průměrnou vzdáleností a jsou časově po sobě jdoucí
- Bude ležet na silnici

Pro eliminaci bodů, vyskytující se krátký časový úsek (např. pomalu se pohybující vozidlo), bude použita rovnice (12). Zobrazením bodů bez ohledu na snímek detekce do 2D plochy vzniknou shluky bodů. U každého shluku by se mělo jednat o stejný detekovaný objekt. Z bodů se vypočítá pár reprezentujících bodů, které budou společně s detekovaným rozměrem tvořit místo anomálie pro další zpracování. Ideálně by pro každé detekované vozidlo mělo být právě jedno místo anomálie. Použitý algoritmus pro výpočet středových bodů bude K-means s euklidovskou metrikou vzdálenosti. Jednotlivé shluky můžou obsahovat různý počet bodů, protože se vozidlo vyskytuje ve videu s předem neurčitou dobou. Výběr počátečních bodů by potom neměl být náhodný, z důvodu vyššího výskytu v hustších shlukách. Z tohoto důvodu bude pro výběr počátečních bodů zvolen algoritmus popsaný v kapitole 5.1 [4]. Počet výskytů vozidel může být ve videu libovolný, ale aby algoritmus neiteroval do nekonečna bude zvolena maximální hodnota vozidel, tedy K. Rozdělení bodů do i-shluků bude v každé iteraci zhodnoceno jednou z metrik popsané v 5.1. Na základě použité metriky bude vybrán výsledný seznam shluků a jejich středů. Následně budou přiloženy masky silnice pro jednotlivé středy a ty, které nejsou na silnici budou vymazány. Z důvodu možného výskytu špatně klasifikovaného objektu, dojde v každé oblasti k opětovné detekci. Místa, ve kterých se vyskytuje vozidlo bude detekováno a takové body se ponechají.

Kvůli předem neznámému počtu bodů bude v této práci vybráno *k* v rovnici (12) jako odmocnina z celkového počtu bodů a $l = 2k$. Rovnice (13) zde nebude použita, protože v jejím důsledku docházelo v implementaci k eliminaci správných bodů. Původně kvůli tomuto zde bylo zavedeno lokální zvětšení. Pro zhodnocení kvality rozdělení bodů do shluků bude vybrána metoda siluet, z důvodu popsané v sekci 5.1. K-means je iterováno s hodnotami k v intervalu *<2; min(počet unikátních bodů, 15)>.* Aby se dal vypočítat koeficient siluety, začíná interval od hodnoty 2. Před samotným K-means je vypočítaná průměrná vzdálenost od okolních bodů a jestliže je menší než 5 je *K=1*, takto se ošetří případ, kdy je detekováno pouze jedno vozidlo. Ze seznamu bodů jsou odstraněny duplicitní prvky podle souřadnic, protože detekce často nastane ve stejném bodě a výrazně to urychlí výpočet.

Veškeré výpočty pro výběr kandidátů jsou implementovány bez použití knihovny, která by to realizovala. Z každého shluku je vybrána nejvyšší hodnota snímku výskytu a podle toho je vybrána maska silnice. Silnici tvoří bílé pixely. Bod je zachován, jestliže je průměrná hodnota v okolí 9 pixelů se středem (x, y) větší než 100 (intenzita pixelu je v rozsahu 0-255). Detektor objektů nemusí při opětovné detekci klasifikovat vozidlo kvůli špatnému snímku, proto se provádí detekce nad několika snímky v intervalu anomálie a jestliže dojde k detekci alespoň v jednom snímku, tvoří bod anomálii. Zvětšením malé oblasti vzniká rozmazaný snímek, a tak za účelem lepší detekce jsou okraje oblasti zvětšeny o 50 pixelů. Při takovém zvětšení oblasti může dojít k překrytí jednoho vozidla s jinými, a tak zde dochází k rozpoznání, zda detekovaný objekt koresponduje s danými souřadnicemi a rozměry. Jestliže je euklidovská vzdálenost

souřadnic a rozměrů menší než 20, jedná o tu samou oblast. Hranice prahu pro jistotu objektu je zde nastavena na 50 %. Lokální zvětšení potlačí hlášení anomálie, ale kvůli nedokonalosti detektoru objektů k tomu i přes to může dojít. Výsledkem těchto operací je požadovaná podmnožina.



Obrázek 6.3.1: Příklad výběru kandidátů lokálním zvětšením obrazu

Na výše uvedeném obrázku byl pro ukázku výběru ohraničující oblasti nastaven práh pro jistotu objektů na 1 %.

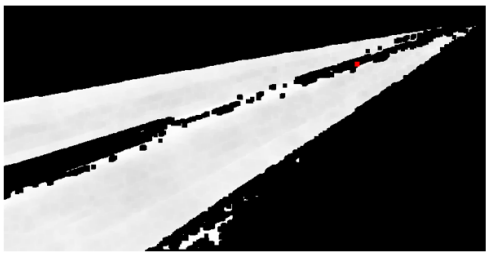
6.4 Modul pro zpracování anomálie

Finální fáze programu spočívá v určení intervalu anomálie ze vstupních bodů a její klasifikaci. Ze selektoru kandidátů budou dostupné středové body shluků, první a poslední bod podle snímků detekce a jejich rozměry. Protože průhlednost vozidla se podle rovnice (12) postupně zvyšuje/snižuje, není snímek detekce přesný. Zároveň je brán v potaz pouze každý 100. snímek. Určení přesného snímku pak bude realizováno postupným porovnáním strukturální podobnosti popsané v sekci 5.2. Pro klasifikaci anomálie pak bude postačovat způsob popsaný v sekci 5.3.

Získání konkrétního snímků probíhá odděleně pro počátek a konec. Z hraničního bodu je vybráno číslo snímku a oblast, která je definována souřadnicemi a rozměry v prvku. Následně je oblast oříznuta ze snímku pozadí uloženého ve složce. Ze vstupního videa je vybrán další snímek v pořadí z důvodu nízké průhlednosti v průměrovaném snímku a stejná oblast je oříznuta. Extrahované oblasti snímky jsou strukturálně porovnány. Protože čas určený čas anomálie je klasifikován jako správný, jestliže interval je s přesností 10 s, je zde velikost kroku opět 100. Nejprve je vypočítána původní hodnota ssim mezi prvními dvěma snímky a tato hodnota, vynásobená konstantou 0,7 je práh. Jestliže sekvence alespoň 4 snímků klesne pod tenhle práh, je první snímek ze sekvence určen jako hraniční interval anomálie. Porovnáním více po sobě jdoucích snímků se zamezí potenciálnímu šumu, kdyby například jiné vozidlo vjelo do popředí. Výsledný interval je v každé iteraci uložen do seznamu. Nezávisle na tomto algoritmu nastává podle kapitoly 5.2 klasifikace anomálie použitím masky silnice do dvou tříd: nebezpečné stání a dopravní nehoda. Společně s transformovanou maskou silnice je po první fázi (6.1) uložena netransformovaná maska. Pro odstranění okrajů silnice je použita transformace eroze (4.3). Vozidlo může být různě vzdálené a šířka silnice není předem známa. Při vyšší velikosti matice se může silnice úplně vytratit, naopak při malé velikosti může dojít k chybné klasifikaci autonehody. Z tohoto důvodu je rozměr matice určen dynamicky na základě rozměru detekovaného vozidla, konkrétně: $m = (n\,x\,n); n = log_2(S_i)$ kde S_i je plocha ohraničujícího boxu. Kdyby funkce pro *n* byla lineární, docházelo by u vyšších hodnotách k vysokému odstranění pixelů silnice, proto je zde zaveden logaritmus. V případě, že průměr

okolních *m* pixelů kolem středu bodu je větší než 240, vyhodnotí se situace jako dopravní nehoda, jinak jako nebezpečné stání. Výhoda této metody je její rychlost.

původní maska silnice



maska po operaci eroze. Červený bod značí detekované vozidlo

Obrázek 6.4.1: Příklad modelování masky pro klasifikaci situace a přiložení detekovaného bodu (zde je klasifikovaná situace nebezpečné stání u vozovky)

```
Dangerous standing by the road starting at: 590, ending at: 896
Dangerous standing by the road starting at: 592, ending at: 796
```

Obrázek 6.4.2: Příklad výstupu programu

Kapitola 7

Testování

V této kapitole bude rozebrán způsob ověřování výsledků implementovaného systému, společně s použitými datovými sady. Vyhodnocení výsledků bude rozděleno na dvě části:

- Úspešnost určení anomálie a její interval
- Úspěšnost určení klasifikace anomálie

Zatímco je klasifikace určena pouze jako procentuální přesnost, určení anomálie a jejího intervalu je definován podle tzv. S_4 skóre.

$$(14)\ S_4 = F_1 * (1 - NRMSE)$$

Kde:

$$(15) F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Metrika F_1 se používá pro určení klasifikační přesnosti. Objevují se zde proměnné:

- Precision – podíl správně určených klasifikací vzhledem k celkovému počtu správných klasifikacích
- Recall – podíl správně určených klasifikací vzhledem k celkovému počtu určených klasifikací

Hodnota je v rozsahu <0;1>, kde 1 značí nejvyšší možnou kvalitu systému. Protože interval je dán čísly, tak se ještě zavádí normalizovaná odmocnina ze střední kvadratické chyby.

$$(16) MSE = \sum_{i=0}^{n} \frac{(\check{y}_i - y_i)^2}{n}$$

Kde v rovnici (16):

- $\check{y}_i$ značí odhadovanou hodnotu
- y_i značí reálnou hodnotu

$$(17) NRMSE = \frac{\min(300, \sqrt{MSE})}{300}$$

Anomálie je určena jako správná, jestliže je interval určen s tolerancí 10 s (součtem začátku a konce) [10]. Odchylka se pak počítá pouze od jejího počátku. V použité datové sadě jako v [23] je FPS 30, a tak maximální odchylka určení intervalu vychází na 300. Výsledná hodnota RMSE je pak normalizovaná právě touto hodnotou a její výsledek, stejně jako hodnota S_4 je opět v interval <0;1>.

7.1 Datová sada

Použitá datová sada byla čerpána z páté AI City Challenge pro detekci anomálií [23]. Datová sada se jmenuje Iowa DOT podle lokace odkud byly čerpány. Jedná se o celkově 250 videí s rozlišením 800x410, 30FPS a každé má délku přibližně 15 min. Video může obsahovat jednu nebo současně více anomálií, a to bud dopravní nehodu nebo zastavené vozidlo v nebezpečném úseku. Datová sada je rozdělena na dvě složky:

- Trénovací sada – 100 videí s celkem 18 anomáliemi s anotovanými intervaly
- Testovací sada – 150 videí

Ačkoliv každé video obsahuje záznam ze silnice, tak často definuje unikátní problém, ze kterým je třeba se vypořádat. Datová sada tak pokrývá velkou část problémů, které můžou nastat při pořizování záznamů z dohledových kamer. Mezi těmito problémy je:

- Nekvalitní záznam kamery v důsledku třepání, šumu (černé snímky), dočasného výpadku, tmy, vody nebo špíny na kameře
- Vozidla v pozadí, které pak může systém vyhodnotit jako anomálii
- Zácpa na silnici nebo dlouhé čekání na semaforu nebo dopravní značce
- Změna záběru kamery
- Velmi vzdálená anomálie

Všechny tyto problémy pak můžou být od méně závažného (šum ve snímkách, ze kterými se nepočítá), až po velice závažný (minutové čekání na značce STOP), kdy je třeba systém adaptovat na daný problém. Jednotlivé videa pak testují, do jaké míry se dokáže systém vypořádat s jednotlivými problémy.

7.2 Průběh testování a vyhodnocení

Datová sada sice obsahuje videa pro trénování, ale pro algoritmus, který není na bázi učení, může být použita pro testování. Současně je anotována, takže vyhodnocení bude snazší. Testování bude probíhat na osobním počítači s CPU AMD Ryzen 5, GPU NVIDIA GeForce GTX 1050 Ti a 16GB RAM, jedná se tedy na dnešní poměry o průměrně výkonný stolní počítač. Použitá datová sada bude pouze část pro trénování, celkově tedy 100 videí, což je stejný počet jako v předchozí testovací sadě AI City Challenge [24]. Nejprve bude uvedeno celkové vyhodnocení a následně důvody špatného vyhodnocení. Z celkového počtu 18 anomálií program detekoval 10 správně (interval s odchylkou maximálně 10 s) a ze zbylých 82 videí pouze 3 klasifikoval nesprávně. Ze zbývajících 8 špatně určených anomálií detekoval 2 se špatným intervalem, zbývajících 6 nezaregistroval vůbec.

Špatně určený interval. Modul pro zpracování anomálie porovnává hodnoty SSIM mezi snímky v oříznutém segmentu. Problém pak může nastat v případě, kdy se segment změní v čase v důsledku změny pohledu kamery. Algoritmus K-means shlukuje podle souřadnic, není zde žádná informace o vozidlu. Sjednotit tento interval by pak znamenalo, že program je schopný se přesně adaptovat na změnu pohledy a současně určit, zda se jedná o stejné vozidlo, které tvoří anomálii. Druhý důvod pak může být z důvodu nepřekročení nastaveného prahu, protože rozdíl SSIM není tak velký. Například pozadí za vozidlem je trošku podobné vozidlu.

Obrázek 7.2.1: Příklady dvou pohledů kamery

V obrázku 7.2.1 lze pak vidět, že detekce dojde ve dvou odlišných místech a porovnáváním konkrétního segmentu dojde k detekci intervalu při změně pohledu kamery.

Nezaregistrované anomálie. Špatně detekovaný interval anomálie je lepší než žádný, protože i tak by systém mohl nahlásit její počátek v případě detekce. Důvodů pak může být více a to např.:

- K detekci vozidla dojde, ale při opětovné detekci lokálním zvětšením nepřesáhne jistota nastavený práh nebo v důsledku špatně vymodelované masky silnice je eliminováno
- K detekci nedojde, protože je buď vozidlo příliš vzdálené, nebo je vozidlo pro detektor moc náročné detekovat

Obrázek 7.2.2: Opětovnou detekci zvětšeného segmentu nepřesáhnou vozidla nastavený práh, pravděpodobně kvůli vyzařovanému světlu

Obrázek 7.2.3: K detekci vozidla dojde, ale přiložením masky bude eliminováno, protože neleží na silnici

Obrázek 7.2.4: Detektor objektů není schopen určit, zda se jedná o vozidlo v takové vzdálenosti (pozn. domnívám se, že u pravého snímku by to bylo náročné i pro člověka)

Obrázek 7.2.5: Příklad vozidel, které detektor nedetekuje se spolehlivostí alespoň 10 %

Jak lze vidět ve výše uvedených případech, jedná se o poměrně náročné případy, které je ale třeba řešit. Vzdálené vozidla lze vyřešit lokálním zvětšením, na to je ale třeba nejprve zjistit, jak moc si danou oblast přiblížit. Konkrétní mechanismus je popsán v [10], bohužel kvůli nedostatku času nebyl implementován.

Špatné hlášení anomálie. V tomto případě program nahlásí, že se vyskytla anomálie, i přes to, že k žádné nedošlo. Mezi hlavními důvody je:

- Model masky silnice není perfektní a vozidlo v pozadí je klasifikováno jako anomálie
- Zácpa na silnici nebo dlouhé čekání na semaforu nebo dopravní značce
- I přes lokální zvětšení dojde ke špatné klasifikaci

Obrázek 7.2.6: Chybně nahlášená anomálie, protože vozidlo čeká na zelenou příliš dlouho (pozn. ohraničující boxy jsou stejné jako v obrázku 6.3.1, levý a pravý obrázek jsou z různého videa)

Obrázek 7.2.7: Detektor objektů si je na 75 % jistý, že v ohraničeném boxu se nachází vozidlo

Správně hlášené anomálie. Pro představu zde bude znázorněno několik příkladů, kdy program detekoval interval správně.

Obrázek 7.2.8: Správně určené intervaly byly u všech detekovaných vozidel, avšak pouze červené se testovaly

Obrázek 7.2.9: Další případy správně detekovaných intervalů

Klasifikace anomálie. Určení typu anomálie proběhlo úspěšně v 8 případech z 10 (nutno podotknout, že ve zbylých 2 videích, ve kterých byl nesprávně určen interval byla správně určena klasifikace). Ve dvou případech, kdy došlo k nesprávnému určení bylo v důsledku:

- K-means algoritmus určil středový bod mezi více body, který pak měl souřadnice v silnici
- Autonehoda nastala v blízkosti okraje

Konkrétně na obrázku 7.2.8 algoritmus K means vypočítal střed mezi dvěma vozidly na levé silnici. V dalším videu pak nastala autonehoda, kde se ale vozidlo přestalo hýbat na straně vozovky, takže klasifikace nebyla vyhodnocena správně.

Vyhodnocení. Precision tedy vychází na 10/18 a recall na 10/13. F_1 = 0,6452 (zaokrouhleno na deseti tisíciny) a RMSE pak vychází na 51,088. Výsledné S_4=0,617. Klasifikace anomálie pak dosahovala přesnosti 80 %.

	F1	RMSE	S4	ACP
Výsledek	0,6452	51,0882	0,53516	0,8

Tabulka 7.2.10: Výsledky jednotlivých metrik měření úspěšnosti program (ACP – anomaly classification precision, tedy “přesnost klasifikace anomálie”)

Doba zpracování jednoho videa pro použitou datovou sadu je v rozmezí 2-4 min.

Srovnání s podobnými systémy. Srovnávány zde budou výsledky z prací, které se zúčastnily soutěže AI City Challenge 2021. Protože takové projekty neklasifikují anomálii, bude porovnáno pouze F1 a S4 skóre. Porovnání nebude v poměru 1:1, protože srovnávací projekty byly testovány na odlišné datové sadě, která obsahuje o třetinu více videí a některé videa můžou být pro správné určení náročnější.

	F1	RMSE	S4
Výsledná hodnota	0.5926	8.2386	0.5763

Obrázek 7.2.11: Výsledky z podobné práce, která se účastnila soutěže a umístila se na 2. místě (převzato z [10])

1	113	Firefly	0.9695
2	51	SIS Lab	0.5763
3	106	CETCVLAB	0.5438
4	72	UMD_RC	0.2952
5	91	HappyLoner	0.2909
6	26	Orange-Control	0.2386
7	49	PapaNet	0.1703
8	132	Team_Gaze_NSU_UAP	0.0958

Obrázek 7.2.12: Seřazené výsledky z prvních 8 týmu podle S4 skóre (převzato z [10])

Při srovnání s týmem č.2 ([10]) dosahuje tato práce vyššího F1 skóre, zatímco odhad časového interval není tak přesný. Pravděpodobně to je v důsledků, že práce [10] používá lepší mechanismus pro jeho výpočet. Například nad funkcí změny SSIM v čase je aplikován Savitzky-Golay filtr, pro eliminaci šumu. Ačkoliv je tato práce testována na jiných videích, v porovnání s výše uvedenými týmy dosahuje relativně dobrých výsledků. Současně je nutno brát v potaz, že výše uvedené práce jsou řešené týmy, které se skládají z profesionálů v daném oboru.

Kapitola 8

Závěr

Cílem této práce je navrhnout a implementovat program, který dokáže zanalyzovat vstupní video a rozpoznat výskyt anomálie. Anomálie je zde definovaná jako nebezpečné stání podél silnice nebo dopravní nehody. Tyto událostí sdílejí tu samou složku a to, že se vozidlo eventuálně přestane hýbat. Základní myšlenka je, že program vyhodnotí jako anomálii taková vozidla, která tvoří pozadí a současně jsou na silnici. Následně je anomálie zpracována a výsledkem je její interval a typ do které patří. Protože počítat s každým snímkem by bylo časově velmi náročné a současně tolik informací není třeba, počítá se v této práci pouze s každým stým snímkem.

Kamery na silnici natáčí v různé kvalitě a snímky můžou být ovlivňovány přírodními podmínkami. Z tohoto důvodu jsou nejprve snímky stabilizovány a poškozené snímky jsou nahrazeny jiným snímkem v časové blízkosti. Pozadí v jednotlivých časových úsecích je získáno průměrováním snímků. Současně s vytvářením pozadí je modelována maska silnice sčítáním výsledků po odečtení pozadí od snímku. Kvůli adaptaci na šum v pozadí je pro odečítání pozadí použit algoritmus MOG. Poté, co jsou získány všechny pozadí, je začne detektor objektů analyzovat. Použitý detektor objektů je pro svoji rychlost a relativně vysokou přesnost YOLOv3. Protože anomálie může nastat v dálce a kvalita snímků může být nízká je nastavena hodnota jistoty objektu YOLO na 0,1. Často se pak může stát, že nastanou nesprávné detekce, které je nutno odstranit. Jednotlivé body detekce jsou pak spojeny a tvoří shluky. Pro odstranění takových bodů, které jsou mezi ostatními odpadlíci, je použit algoritmus K nejbližší soused. Shluky jsou vypočítány algoritmem K Means. Jestliže středový bod shluku neleží na masce silnice, je odstraněn. Oblast, ve které se nacházejí body ve shluku je oříznuta a zvětšena pro opětovnou detekci, za účelem eliminace nesprávně určených kandidátů.

Po této operaci jsou k dispozici takové shluky bodů, jenž reprezentují vozidlo, které opravdu tvoří anomálii. Porovnáním oblasti vozidla s dalšími snímky v pořadí je použitím algoritmu SSIM určen interval anomálie. Jakmile leží středový bod na masce silnice, která byla zmenšena použitím morfologické operace, jedná se o vozidlo, které leží uprostřed silnice a klasifikuje se jako dopravní nehoda. V opačném případě se jedná o nebezpečném stání podél silnice.

Pro testování je použit průměrně výkonný stolní počítač. Použitá datová sada je čerpána z práce [23], která obsahuje celkem 250 videí v rozlišení 800x410 s 30FPS a délkou trvání zhruba 15 min. Z datové sady je využito pouze 100 videí, které jsou anotovány a sjednoceny v jedné složce. Výsledné skóre F1 je 0,6452 a S4 skóre 0,53516. Přesnost určení dopravní situace je 80 % a rychlost analýzy videa pro použitou datovou sadu 2-4 min. Při srovnání s ostatními řešitelskými týmy byly zjištěny dobré výsledky.

Vylepšení aktuálního systému by pak mohlo spočívat v detekci i velmi vzdálených objektů, vylepšení algoritmu pro výpočet masky silnice a optimalizace rychlosti analýzy. Současně úprava modulu pro zjištění času počátku anomálie by zvýšila celkové skóre, protože jak ukázalo měření, byla tato část, v porovnání s jiným řešením, kritická.

Odkazované zdroje

[1] Adam. A and E. Rivlin and I. Shimshoni, et al. *Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors*. [online]. in IEEE Transactions on Pattern Analysis and Machine Intelligence, březen 2008. vol. 30, č. 3, s. 555-560. [cit 2022-04-1]. DOI: 10.1109/TPAMI.2007.70825. Dostupné z: https://ieeexplore.ieee.org/document/4407716

[2] Agarap, Abien Fred. *Deep Learning using Rectified Linear Units (ReLu)*. [online]. Březen 2018. [cit 2022-03-26]. DOI: 10.48550/ARXIV.1803.08375. Dostupné z: https://arxiv.org/abs/1803.08375

[3] Alberto Marchisio and Muhammad Abdullah Hanif and Semeen Rehman, et al. *A methodology for automatic selection of activation functions to design hybrid deep neural networks*. [online]. 2018. [cit 2022-03-26]. DOI: 10.48550/ARXIV.1811.03980. Dostupné z: https://arxiv.org/abs/1811.03980

[4] Arthur, David and Vassilvitskii, Sergei. *K-Means++: The Advantages of Careful Seeding*. [online]. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms, Leden 2007. Č. 8. s. 1027-1035. [cit 2022-04-10]. DOI: 10.1145/1283383.1283494.Dostupné z: https://www.researchgate.net/publication/220778887_K-Means_The_Advantages_of_Careful_Seeding

[5] Bai, S. and He, Z. and Lei, Y., et al. *Traffic Anomaly Detection via Perspective Map based on Spatial-temporal Information Matrix*. [online]. CVPR Workshops, 2019. [cit 2022-04-08]. Dostupné z: https://www.semanticscholar.org/paper/Traffic-Anomaly-Detection-via-Perspective-Map-based-Bai-He/579da787acbe561b05beea60162488a15f82ceea

[6] Benezeth, P. and Jodoin, V. Saligrama and C. Rosenberger. *Abnormal events detection based on spatio-temporal co-occurences*. [online]. IEEE Conference on Computer Vision and Pattern Recognition, Miami, USA, červen 2009. S. 2458-2465. [cit 2022-03-19]. DOI: 10.1109/CVPR.2009.5206686. Dostupné z: https://ieeexplore.ieee.org/document/5206686

[7] Bochkovskiy and Alexey and Wang, et al. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. [online]. Duben 2020. [cit 2022-04-14]. DOI: 10.48550/ARXIV.2004.10934. Dostupné z: https://arxiv.org/abs/2004.10934

[8] Bouwmans, Thierry. *Traditional and recent approaches in background modeling for foreground detection: An overview*. [online]. Computer Science Review, květen 2014. Svazky 11–12, s. 31-66. [cit 2022-04-20]. ISSN 1574-0137. DOI: 10.1016/j.cosrev.2014.04.001. Dostupné online: https://www.sciencedirect.com/science/article/pii/S1574013714000033?via%3Dihub

[9] Bouwmans, Thierry and Baf, Fida and Vachon, Bertrand. *Statistical Background Modeling for Foreground Detection: A Survey*. [online]. Leden 2010. [cit 2022-04-20]. DOI: 10.1142/9789814273398_0008. Dostupné z: https://www.researchgate.net/publication/215737750_Statistical_Background_Modeling_for_Foreground_Detection_A_Survey

[10] Doshi and Y. Yilmaz. *Fast Unsupervised Anomaly Detection in Traffic Videos*. [online]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, USA, červenec 2020. S. 2658-2664. [cit 2022-04-08]. DOI: 10.1109/CVPRW50498.2020.00320. Dostupné z: https://ieeexplore.ieee.org/abstract/document/9150642

[11] El Baf and T. Bouwmans and B. Vachon. *A fuzzy approach for background subtraction*. [online]. 15th IEEE International Conference on Image Processing, San Diego, USA, Říjen 2008. S. 2648-2651. [cit 2022-04-19]. DOI: 10.1109/ICIP.2008.4712338. Dostupné z: https://ieeexplore.ieee.org/document/4712338

[12] Feichtenhofer, Christoph and Fan, Haoqi and Malik, et al. *SlowFast Networks for Video Recognition*. [online]. Prosinec 2018. [cit 2022-04-29]. DOI: 10.48550/ARXIV.1812.03982. Dostupné z: https://arxiv.org/abs/1812.03982

[13] Gholamalinejad, Hossein and Khosravi, Hossein. *Pooling Methods in Deep Neural Networks, a Review*. [online]. Září 2020. [cit 2022-03-28]. DOI: 10.48550/ARXIV.2009.07485. Dostupné z: https://www.researchgate.net/publication/344277235_Pooling_Methods_in_Deep_Neural_Networks_a_Review

[14] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing, et al. *Deep Residual Learning for Image Recognition*. [online]. Prosinec 2015. S. 770-778. [cit 2022-03-30]. DOI: 10.48550/ARXIV.1512.03385. Dostupné z: https://arxiv.org/abs/1512.03385

[15] He, Kaiming and Sun, Jian. *Convolutional neural networks at constrained time cost*. [online]. Prosinec 2014. s. 5353-5360. [cit 2022-03-30]. DOI: 10.48550/ARXIV.1412.1710. Dostupné z: https://arxiv.org/abs/1412.1710

[16] Ho and S. Wookey. *The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling*. [online]. in IEEE Access, Prosinec 2019. Svazek 8, s. 4806-4813. [cit 2022-03-29]. DOI: 10.1109/ACCESS.2019.2962617. Dostupné z: https://ieeexplore.ieee.org/document/8943952

[17] Ioffe, Sergey and Szegedy, Christian. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. [online]. Únor 2015. [cit 2022-03-29]. DOI: 10.48550/ARXIV.1502.03167. Dostupné z: https://arxiv.org/abs/1502.03167

[18] Jianbo Shi and Tomasi. *Good features to track*. [online]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA, červen 1994. s. 593-600. [cit 2022-04-27]. DOI: 10.1109/CVPR.1994.323794. Dostupné z: https://ieeexplore.ieee.org/document/323794

[19] Jodoin, J. Konrad and V. Saligrama. *Modeling background activity for behavior subtraction*. [online]. Second ACM/IEEE International Conference on Distributed Smart Cameras, Niteroi, Brazílie, září 2008. S. 1-1. [cit 2022-03-19]. DOI: 10.1109/ICDSC.2008.4635683. Dostupné z: https://ieeexplore.ieee.org/document/4635683

[20] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey. *ImageNet Classification with Deep Convolutional Neural Networks*. [online]. Neural Information Processing Systems, Leden 2012. [cit 2022-03-28]. DOI:10.1145/3065386. Dostupné z: https://www.researchgate.net/publication/267960550_ImageNet_Classification_with_Deep_Convolutional_Neural_Networks

[21] Le Cun and B. Boser and J. S. Denker, et al. *Handwritten digit recognition with a back-propagation network*. [online]. Advances in neural information processing systems 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, červen 2020. s. 396–404. [cit 2022-03-29]. Dostupné z: https://papers.nips.cc/paper/1989/hash/53c3bce66e43be4f209556518c2fcb54-Abstract.html

[22] Liaw, Andy and Wiener, Matthew. *Classification and Regression by RandomForest*. [online]. Forest, Listopad 2001. Svazky 2/3. [cit 2022-03-27]. https://www.researchgate.net/publication/228451484_Classification_and_Regression_by_RandomForest

[23] Naphade, Milind, et al. *The 5th ai city challenge*. [online]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, duben 2021. [cit 2022-04-2]. DOI: 10.48550/ARXIV.2104.12233. Dostupné z: https://arxiv.org/abs/2104.12233

[24] Naphade, et al. *The 4th AI City Challenge*. [online]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, USA, červenec 2020. S. 2665-2674. [cit 2022-04-23]. DOI: 10.1109/CVPRW50498.2020.00321. Dostupné z: https://ieeexplore.ieee.org/document/9150577

[25] Noel, Mathew and L, Arunkumar and Trivedi, Advait, et al. *Growing Cosine Unit: A Novel Oscillatory Activation Function That Can Speedup Training and Reduce Parameters in Convolutional Neural Networks*. [online]. Srpen 2021. [cit 2022-03-28]. DOI: 10.48550/ARXIV.2108.12943. Dostupné z: https://arxiv.org/abs/2108.12943

[26] Peri, et al. *Towards Real-Time Systems for Vehicle Re-Identification, Multi-Camera Tracking, and Anomaly Detection*. [online]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, USA, červenec 2020. S. 2648-2657. [cit 2022-04-1]. DOI: 10.1109/CVPRW50498.2020.00319. Dostupné z: https://ieeexplore.ieee.org/document/9150744

[27] Redmon, Joseph, et al. *You only look once: Unified, real-time object detection*. [online]. Proceedings of the IEEE conference on computer vision and pattern recognition, červen 2015. [cit 2022-04-09]. DOI: 10.48550/ARXIV.1506.02640. Dostupné z: https://arxiv.org/abs/1506.02640

[28] Ren, Shaoqing, et al. *Faster r-cnn: Towards real-time object detection with region proposal networks*. [online]. Advances in neural information processing systems 28, červen 2015. [cit 2022-04-07]. DOI: 10.48550/ARXIV.1506.01497. Dostupné z: https://arxiv.org/abs/1506.01497

[29] Rosenblatt, Frank. *The perceptron: a probabilistic model for information storage and organization in the brain*. [online]. Psychological review, 1958. Svazek. 65, č. 6, s. 386-408. [cit 2022-03-23]. DOI:10.1.1.335.3398. Dostupné z: https://doi.org/10.1037/h0042519

[30] Sultani, Waqas and C. Chen and Mubarak Shah. *Real-World Anomaly Detection in Surveillance Videos*. [online]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, leden 2018. S. 6479-6488. [cit 2022-03-21] DOI: 10.48550/ARXIV.1801.04264. Dostupné z: https://www.researchgate.net/publication/322498470_Real-world_Anomaly_Detection_in_Surveillance_Videos

[31] Uijlings, Jasper and Sande, K. and Gevers, T. et al. *Selective Search for Object Recognition.* [online]. International Journal of Computer Vision 104, září 2013. s. 154-171. [cit 2022-04-01]. DOI: 10.1007/s11263-013-0620-5. Dostupné z: https://www.researchgate.net/publication/262270555_Selective_Search_for_Object_Recognition

[32] Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. [online]. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, USA, prosinec 2001. S. I-I, [cit 2022-03-20]. DOI: 10.1109/CVPR.2001.990517. Dostupné z: https://ieeexplore.ieee.org/document/990517

[33] Ying, Xue. *An Overview of Overfitting and its Solutions*. [online]. Journal of Physics: Conference Series, únor 2019. [cit 2022-03-29]. DOI: 1168. 022022. 10.1088/1742-6596/1168/2/022022. Dostupné z: https://www.researchgate.net/publication/331677125_An_Overview_of_Overfitting_and_its_Solutions

[34] Zhao, Yuxiang and Wenhao Wu and Yue He, et al. *Practices and A Strong Baseline for Traffic Anomaly Detection*. [online]. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, květen 2021. [cit 2022-04-08] DOI: 10.48550/ARXIV.2105.03827. Dostupné z: https://arxiv.org/abs/2105.03827

[35] Zhou Wang and A. C. Bovik and H. R. Sheik, et al. *Image quality assessment: from error visibility to structural similarity*. [online]. in IEEE Transactions on Image Processing, duben 2004. Svazek 13, č. 4, s. 600-612. [cit 2022-04-23]. DOI: 10.1109/TIP.2003.819861. Dostupné z: https://ieeexplore.ieee.org/document/1284395

[36] Zivkovic, Zoran. *Improved Adaptive Gaussian Mixture Model for Background Subtraction*. [online]. Proceedings - International Conference on Pattern Recognition. 2, září 2004. Svazek 2, s. 28 – 31. [cit 2022-04-25]. DOI: 10.1109/ICPR.2004.1333992. Dostupné z: https://www.researchgate.net/publication/4090386_Improved_Adaptive_Gaussian_Mixture_Model_for_Background_Subtraction

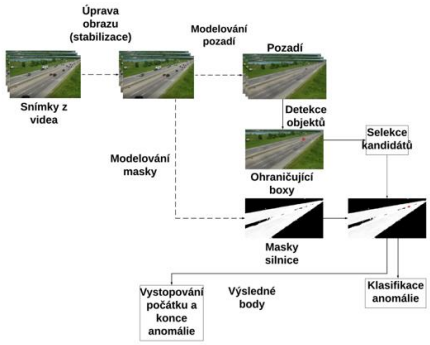
Příloha A

Plakát

Rozpoznávání a klasifikace dopravních situací

Cílem této práce je identifikace a klasifikace nebezpečných situací z přehledových kamer, monitorujících dopravní provoz. V budoucnu by se tak vytvořený program mohl podílet na vývoji chytrého města.

Schéma



Ve skutečnosti se vozidla v daných dopravních situací eventuálně přestanou pohybovat a stanou se dočasně součástí pozadí v oblasti silnice. Vnější podmínky ovlivňují kvalitu videa z dopravních kamer a úloha se tak značně komplikuje.

Dosažené výsledky

Počet videí	TP	FP	TN	FN	F1	S4	Přesnost klasifikace
100	10	8	79	3	0,6452	0,5352	80 %

Testovací videa byly 15min dlouhé s rozlišením 800x410 a 30FPS. Doba analýzy se pohybovala v rozsahu 2-4min.

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Autor: Jiří Zbořil
Vedoucí: doc. RNDr. Pavel Smrž, Ph. D

Obr. A. 1: Plakát prezentující práci, její cíle a výsledky