Technická zpráva k projektu VI20172020068

Nástroje a metody zpracování videa a obrazu pro zvýšení efektivity operací bezpečnostních a záchranných složek (VRASSEO)

Algoritmy - Úprava, nastavení, kalibrace

Prosinec 2020

Abstrakt

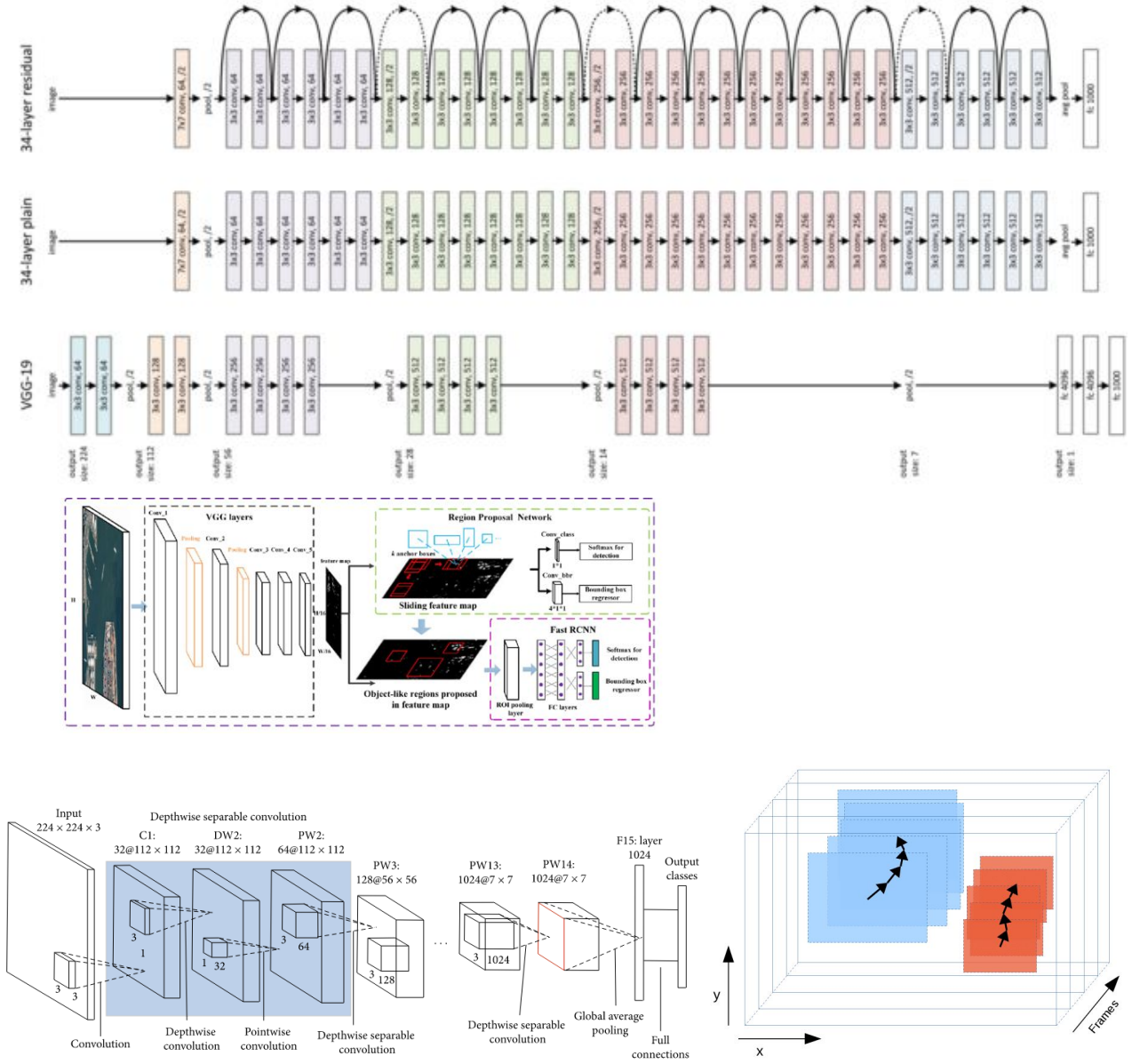
Byla dokončena metoda pro sledování pohybu objektů ve scéně a klasifikaci anomálních jevů, jak na základě modelu pozadí, tak i s využitím moderních technik CNN. Dále byl dokončen vývoj a testování metody pro detekci požáru v obraze, který je založen na sémantické segmentační metodě DeepLabV3 ve statických snímcích. Probíhá integrace těchto metod do systému, popř. vybraných metod také příprava samostatného nástroje. Vyvinuli jsme a na reálných datech ověřili detektor výstražných značek ADR, který může být použit pro vytipování potenciálně nebezpečných vozidel v zájmových úsecích silnic. Pro pořizování HDR videosekvencí se při běhu používá jediný parametr nastavovaný na základě okolního osvětlení a samotné pořizování HDR záznamu je velmi robustní a není třeba jej dále nastavovat. Pro přehrávání HDR (převod HDR do SDR) není třeba žádné parametry nastavovat, použitý algoritmus sám zjišťuje potřebné minimální a maximální hodnoty pixelů a nastavuje podle něj převod, je tedy “bezobslužný” a robustní. Je připravena nová verze detektoru a klasifikátoru zbraní, včetně nové verze generátoru snímků zbraní ve scéně. Je připraven algoritmus určení natočení obličeje a inovovaná verze generátoru snímků obličejů z 3D modelu. Probíhaly off-line experimenty zaměřené na hledání složených událostí ve videu využitím dat extrahovaných snímacím modulem pro detekci automobilů a osob.

Obsah

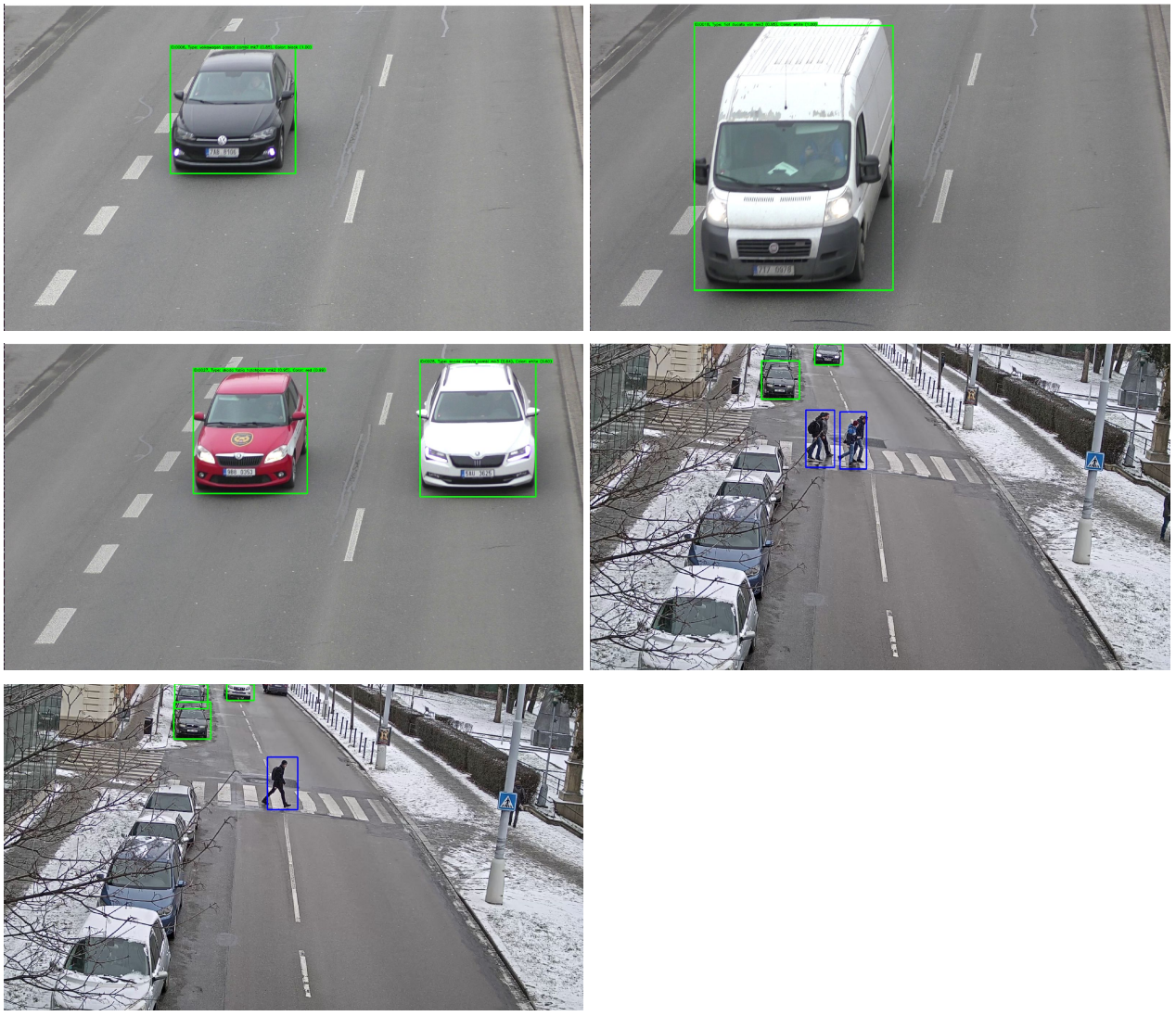
Sledování pohybu ve scéně

Byly navrhnuty a natrénovány modely konvolučních neuronových sítí pro detekci a klasifikaci vozidel. Souběžně s detekcí vozidel lze detekovat i osoby ve snímku. Pro detekci a klasifikaci byly implementovány a vyhodnoceny sítě Faster-RCNN [1] a MobileNet [2]. Detekce vozidel byla trénována na dvou datasetech a to COD20k [3] a UA-DETRAC [4]. Detekce osob byla trénována na datasetu SPID [5]. Základem použitých konvolučních neuronových sítí je extraktor příznaků ResNet [6]. Klasifikace vozidel byla trénována na datasetu BoxCars [7].

Klasifikace vozidel je prováděna na úrovni značky vozidla, modelu vozidla i konkrétní modelové řady. Součástí řešení je i sledování pohybu detekovaných objektů pomocí IoU trackeru, který dokáže sledovat objekt na základě předchozích detekcí daného objektu. Tím je možné objekt sledovat po celou dobu, kdy byl v obraze detekován.



Obr. 1: Architektura natrénovaných neuronových sítí.



Obr. 2: Ukázka výstupu algoritmu pro detekci vozidel a chodců.

Pořizování a zpracování HDR videosekvencí

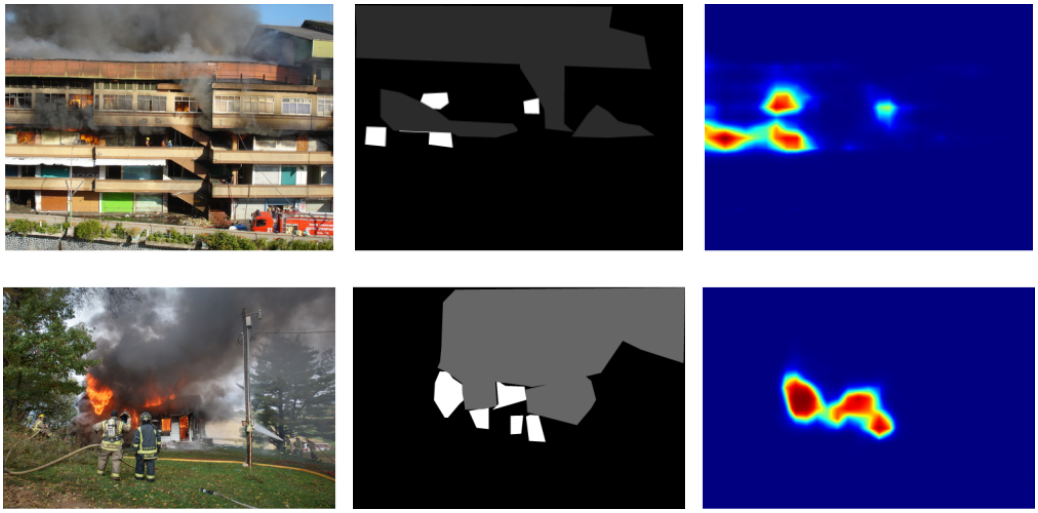
Pro pořizování HDR (High Dynamic Range) [8] videosekvencí se při běhu používá jediný parametr nastavovaný na základě okolního osvětlení a samotné pořizování HDR záznamu je velmi robustní a není třeba jej dále nastavovat. Drobné změny průměrného jasu jsou filtrovány v robustním mapovacím algoritmu, který převádí HDR sekvenci na LDR (Low Dynamic Range - rozsah zobrazitelný na standardních zobrazovacích zařízeních). Pro mapování HDR sekvence do LDR je použit algoritmus založen na původním algoritmu Durand a Dorsey [9] rozšířen o pokročilou kontrolu parametrů pro řízení mapování videosekvencí. Algoritmus pro zachování temporálního charakteru videa adaptuje parametry mapování nejen na základě parametrů aktuálního snímku, ale využívá informace z předchozích snímků (budoucí snímky se nepoužívají protože by mohlo dojít k situaci, kdy algoritmus začne reagovat na změnu jasu dříve než reálně nastane). Ukázka výsledků mapování HDR videa je ukázán na následujícím obrázku 3.

Obr. 3: Ukázka výstupu algoritmu mapování HDR na LDR na různých sekvencích.

Segmentace ohně v obraze

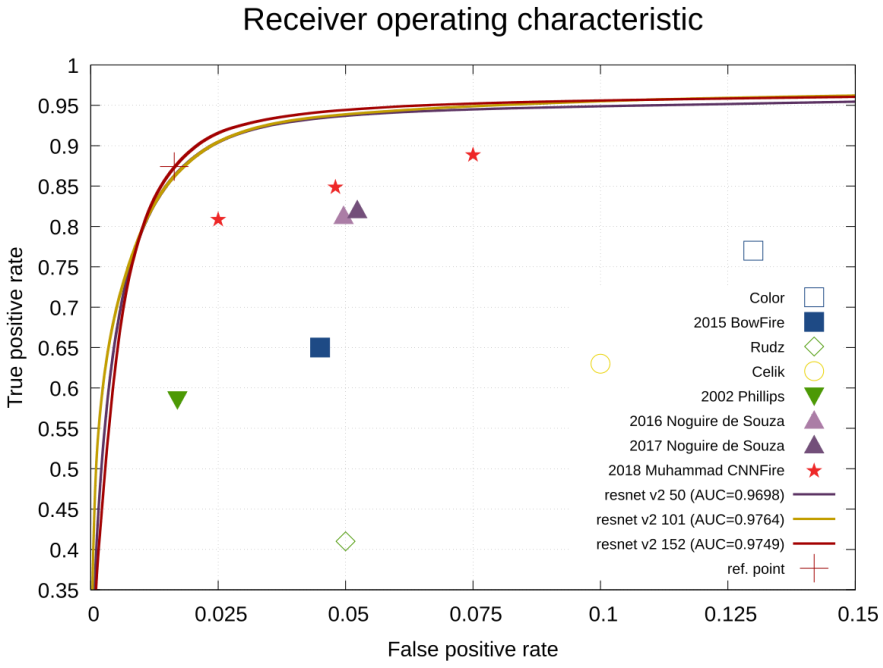
Navrhli jsme nový přístup k lokalizaci ohně v obrazech na základě nejmodernější metody sémantické segmentace DeepLabV3 [10]. Sestavili jsme datovou sadu 1775 obrázků obsahujících oheň z různých zdrojů, pro které jsme vytvořili polygonové anotace. Datová sada je rozšířena o obrázky neobsahující oheň z datové sady SUN397 [11].

Vstupem je tedy obraz a výstupem jsou pravděpodobnosti výskytu obrazu pro každý pixel. Na následujícím obrázku (Obr. 4) jsou v prvním sloupečku příklady vstupních obrázku v datové sadě, jejich anotace (druhý sloupec). Šedou barvou je vyznačena poloha kouře, bílou barvou poloha ohně. Ve třetím sloupci jsou pravděpodobnosti výskytu ohně.



Obr. 4: Ukázka vstupních (vlevo) a anotovaných (uprostřed) dat a pravděpodobnostní mapa výskytu ohně a kouře (vpravo).

Metoda segmentace natrénovaná s naší datovou sadou dosáhla lepších výsledků než nejnovější metody s datovou sadou BowFire. Konkrétně 83.8% TPR at 1.5% FPR a přesnosti (accuracy) 97.8% pro zvolený operační bod. Pro lepší představu a porovnání s ostatními byly hodnoty TPR a FPR znázorněny v ROC křivce (viz Obr. 5).



Obr. 5: ROC křivka

Při vyhodnocení na vytvořené datové sadě bylo dosaženo přesnosti (accuracy) 99%, metrika Intersection over Union dosáhla 70.5%. Věříme, že vytvořená datová sada usnadní další vývoj metod detekce požáru a segmentace, a že by tyto metody měly být založeny na obecných segmentačních sítích.

Publikace a vytvořená datová sada: https://www.fit.vut.cz/research/publication/12124/

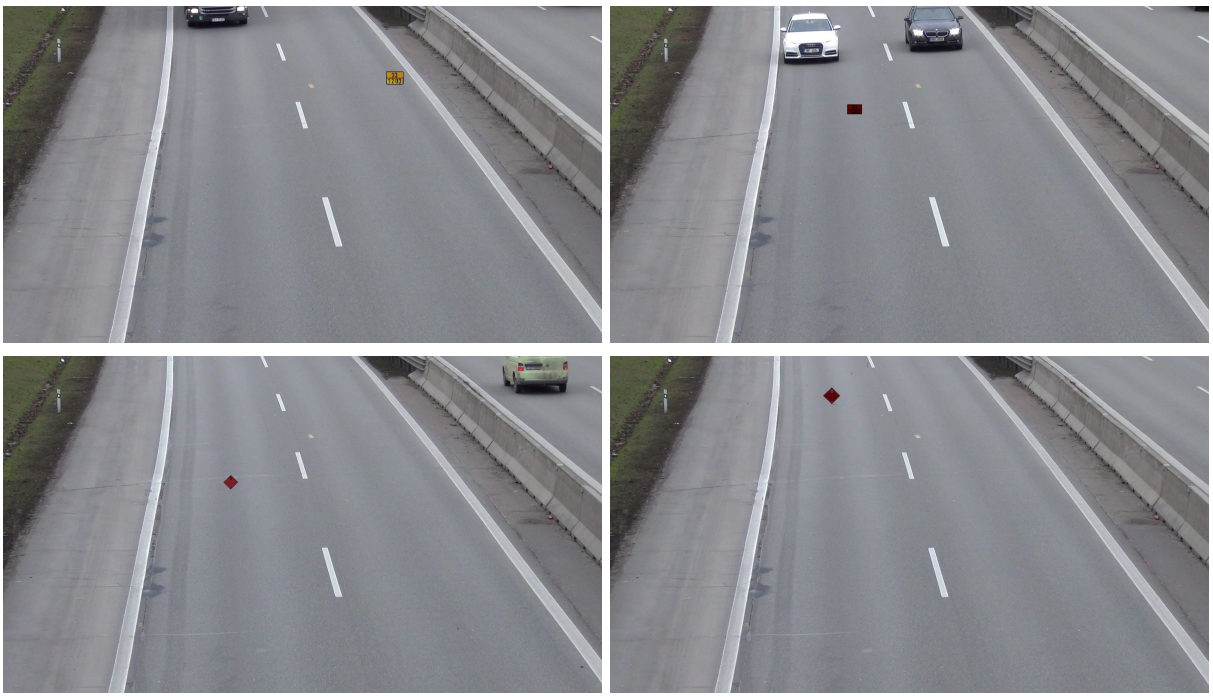
Pro další zpracování lze ve výstupním obraze filtrovat regiony pomocí prahu. Ve výchozím nastavení se berou v úvahu pouze regiony s hodnotou větší než 0,5. Tyto regiony jsou pak opsány obdélníkem (bounding box) a jsou zobrazeny jeho souřadnice a průměrný likelihood pro daný region.

Detekce značek ADR o nebezpečném nákladu

V současné době se při přepravě nebezpečných nákladů používají především ADR označení [12]. ADR je Evropská dohoda o mezinárodní silniční přepravě nebezpečných věcí. Mezi nebezpečné vlastnosti patří například hořlavost, výbušnost, uvolňování a tlak plynů, žíravost, toxicita - jedovatost, samovolná reakce, radioaktivita, oxidace, infekčnost, rakovinotvornost a další. První typ ADR označení, kterým musí být označeno větší množství přepravovaného nákladu, je tzn. ADR tabulka. Jedná se o oranžovou tabulku, na které je označen typ nebezpečné látky, která je přepravována. Pokud je tato tabulka čistě oranžová bez jakéhokoli textu, znamená to, že je přepravováno více nebezpečných nákladů najednou. Dále je náklad označen ADR značkami ve tvaru čtverce postaveného na jeden z jeho vrcholů, které označují třídu nebezpečnosti nákladu. Kromě těchto značek se práce zabývala i detekcí značek označujících nebezpečné látky nejen při přepravě, ale i obecně na výrobcích.

Pro detekci výstražných značek byl zvolen nástroj YOLO [13] ve verzi YOLOv3-Tiny. YOLO na rozdíl od ostatních detektorů (faster R-CNN, SSD, atd.), které pracují při detekci ve dvou fázích (vyhledávání hraničních oblastí objektů a jejich následné rozpoznávání), řeší detekci objektu jako jediný regresní problém. Tato architektura sítě sleduje celý obraz v době trénování a testování, takže jeho předpovědi jsou ovlivněny globálním kontextem v obraze. Tento sjednocený model má několik výhod ve srovnání s ostatními detekčními systémy. Na druhou stranu vznikají specifické nároky na vhodnou datovou sadu. Pro vlastní implementaci byl zvolen framework Darknet ve verzi od AlexeyAB [14], který byl upraven pro požadavky trénování.

Jelikož neexistuje žádná vhodná datová sada pro řešení problému detekce těchto značek, byl vytvořen generátor syntetické datové sady v jazyce Python (Obr. 6). Tento program malé ADR značky vkládá do obrázků z kamer s projíždějícími auty. Při vkládání je značka vykreslena s náhodnou pozicí a velikostí v předem nastaveném rozmezí a její pozice je zaznamenána v textovém souboru, který je dále využit ke trénování modelu. Značce jsou při vkládání taktéž měněny vizuální vlastnosti, které lépe simulují reálné zhoršené podmínky při detekci. Mezi tyto vlastnosti patří změna kontrastu, změna jasu, šum, rozmazání a zkosení jak v ose *x*, tak v ose *y*. Poslední dotrénování modelu bylo uskutečněno nad několika snímky z reálného prostředí, které byly ručně doanotovány. Ukázalo se, že kombinace velkého množství syntetických dat v kombinací s velmi malým množství reálných dat (cca 20) vede k dobré úspěšnosti detektoru.

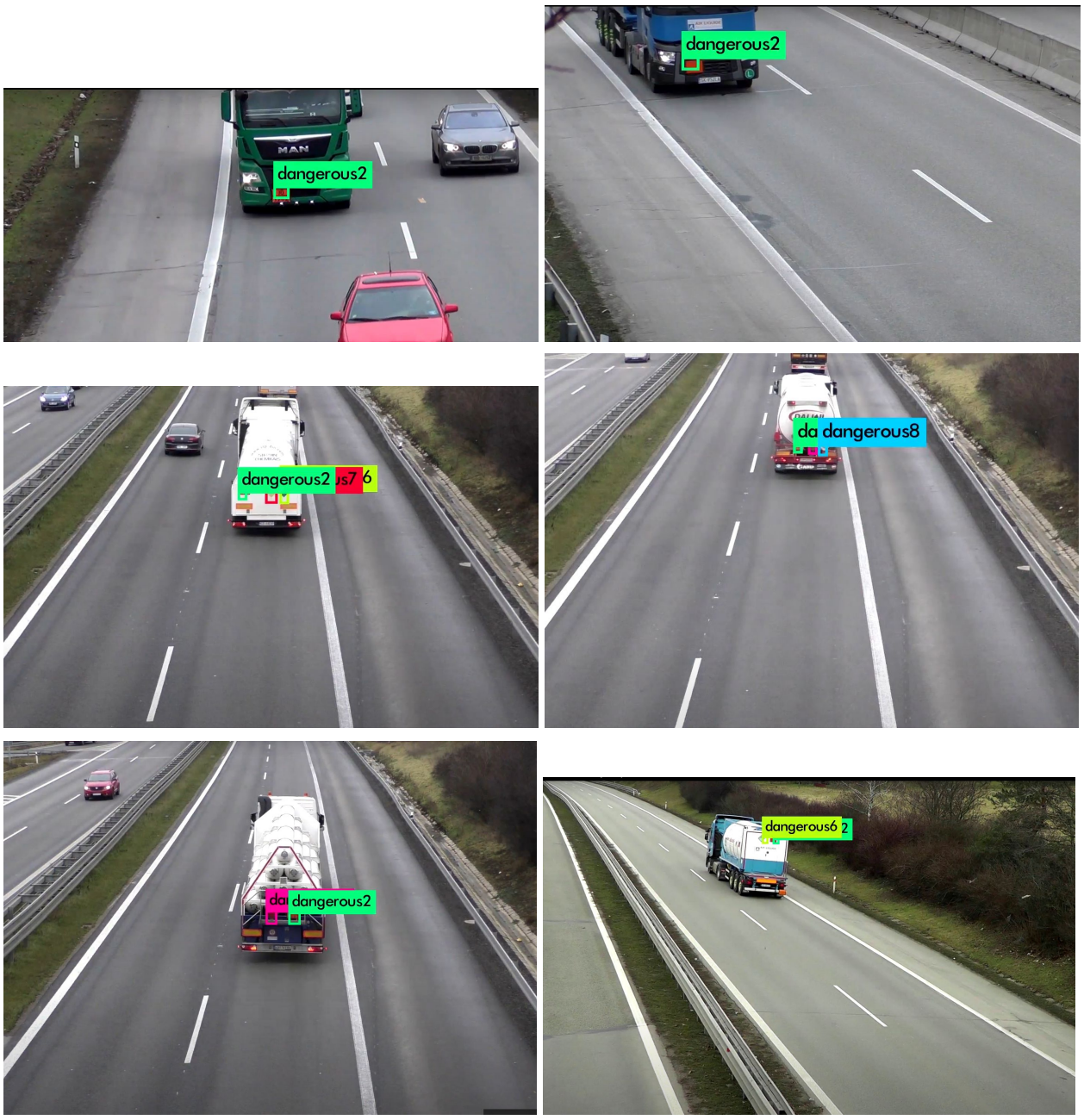


Obr. 6: Ukázka syntentické datové sady

Značky pro označení nebezpečných nákladů mimo přepravu byly testovány na obrázcích z internetu, viz Obr. 7.

Obr. 7: Příklad pokusných obrázků použitých při vývoji detektoru

Zbylé ADR značení bylo testováno na velkém množství videí z dopravy. Jednalo se o záznam z kamer, které byly umístěny většinou nad vozovkou a monitorovaly provoz. Ve videích byl provoz monitorován ve směru jízdy, proti směru jízdy i pod úhlem z boku. Příklady některých detekcí nad těmito videi lze vidět na obrázcích níže. Tyto ukázky byly pořízeny jako snímky z detekovaného videa.



Obr. 8: Příklady automobilů s výstražnými značkami nalezených na dříve pořízených záznamech z dálničních kamer

Detekce anomálie v pohybu davu osob

Potřebou uživatele je sledování mnohdy i rozsáhlého prostoru a rychlá reakce v případě výskytu neobvyklé a potenciálně nebezpečné situace v davu. Pojem potenciálně nebezpečná situace lze blíže specifikovat jako zrychlení, zastavení, rozdělení, srocení nebo změna směru pohybu davu. Vstupní data pro řešení této úlohy mohou představovat jak videozáznamy pořízené statickou kamerou, ale i umístěnou na dronu ve stacionární poloze. Vlivem povětrnostní situace, trajektorie letu, dostupné vzdálenosti od cílového místa a dalších okolností mohou být data snímána se záchvěvy někdy i výraznějšího charakteru je nutno obraz stabilizovat. Ze stabilizovaných úseků videa jsou pak získávány informace o optickém toku, které slouží k zakódování pohybu mezi snímky. Příznaky o pohybu slouží k modelování běžného pohybu ve snímku, který je rozdělen na

mřížku buněk. Pohyb, který není běžný, je dostatečně vzdálen od modelu běžného pohybu je pak klasifikován jako anomálie.

Ačkoliv je **stabilizace obrazu** při snímání dronem řešena v snímacím zařízením (jak závěsné zařízení pro kameru, tak vlastní kamera), vzniká řada situací, kdy toto nestačí - např. krátký výpadek v přenosu dat, nečekaná změna nastavení expozice, prudší závan větru, výkyv pozice dronu operátorem apod. Drobné korekce nestability záznamu jsou řešeny detekcí klíčových bodů a jejich sledováním do následujícího snímku. Z trajektorie jejich posunu je určena transformační matice, která umožňuje posunutí odpovídajících snímků přes sebe. Dochází tedy k odstranění pohybu pozadí, což by mělo negativní vliv na výsledek detekce anomálie.

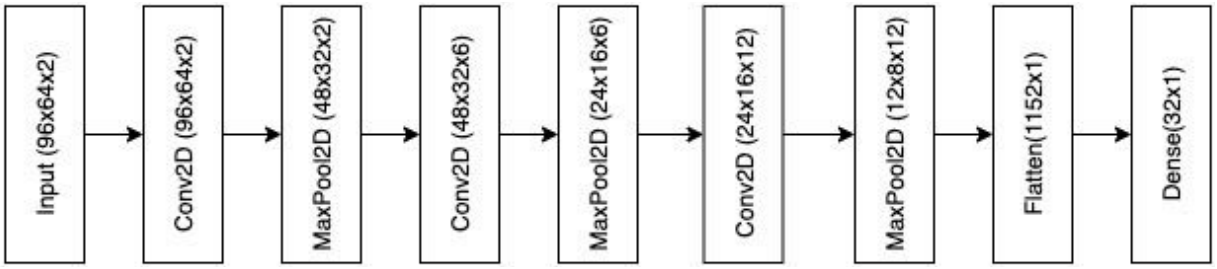
Stabilní oblasti videa, ve kterých je prováděna detekce klíčových bodů a následná analýza jejich pohybu, jsou specifikovány uživatelem skrze zadání masky. Pro detekci klíčových bodů je využito rohového detektoru, posun detekovaných bodů je vypočten pomocí optického toku a výsledná transformační matice je pak vypočtena metodou RANSAC. Vzájemné posuny jednotlivých snímků jsou dále akumulovány v transformační matici vyjadřující vztah mezi prvním a aktuálním snímkem. To je klíčové pro zobrazení heatmapy vyjadřující úroveň anomálie přímo do mapového podkladu.

V případě silného záchvěvu ve videu není možné s dostatečnou přesností využít předchozího postupu. Pro tento účel byla využita detekce klíčových bodů metodou FAST a získání obrazových příznaků metodou BRIEF. Tyto metody se ukázaly pro zadaný úkol jako dostatečně robustní i přes jejich nízkou výpočetní náročnosti. Korespondence snímků je vypočtena hledáním odpovídajících dvojic pomocí metody pro aproximativní hledání nejbližších sousedů. Tyto klíčové body jsou rovněž využity v situaci, kdy dojde k úplné ztrátě původní scény, a je nutné provést zpětné dohledání.

Detekce anomálií v pohybu davu osob je založena na **příznacích charakterizujících změnu polohy obrazových bodů v obraze**. Základem je optický tok mezi následujícími snímky vypočten Farnebackovou metodou. Tento optický tok je dále transformován do souřadnicového systému modelu pozadí. Tím je zajištěno správné mapování optického toku fyzického místa scény na odpovídající místo v souřadnicovém systému modelu i v případě posunu kamery.

Myšlenkou použitého přístupu je extrakce série příznakových vektorů v čase a následná detekce anomálie spočívající v hledání outlierů pro dané místo v obraze. Obraz je tedy rozdělen do pravidelné mřížky. **Metody detekce anomálie si pro každou buňku udržují model pozadí**, který je pro dané místo scény obvyklý. Porovnáním aktuálního pohybu s modelem pozadí je vypočtena úroveň anomálie pro aktuální snímek. Výstupem analýzy je mapa specifikující úroveň anomálie pro každý bod souřadnicového systému modelu.

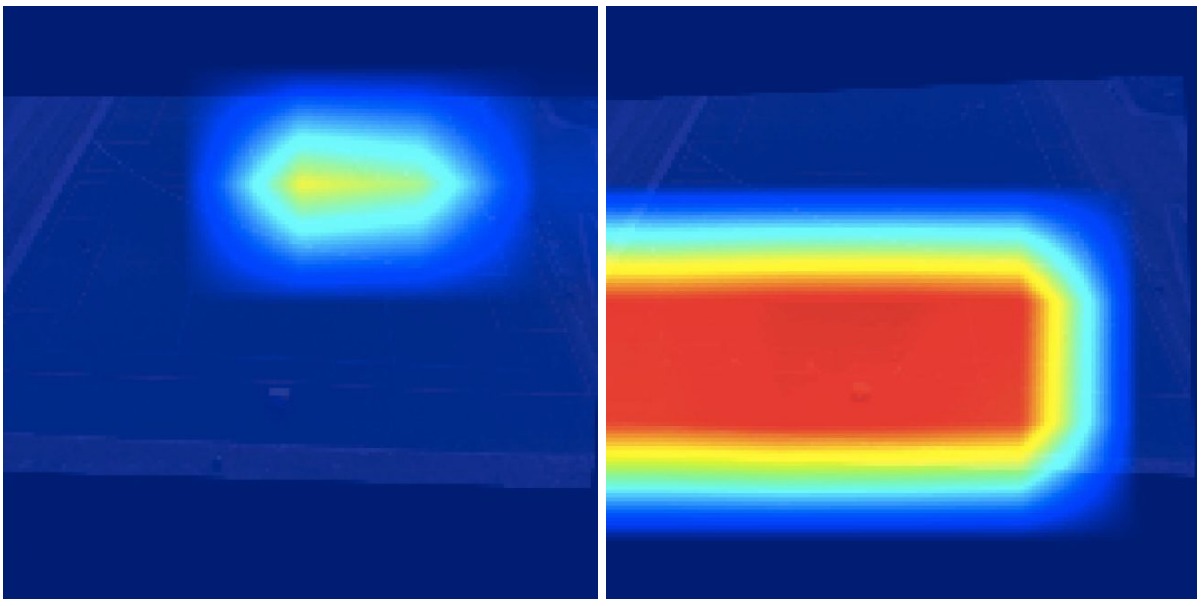
Optický tok v dané buňce je tedy transformován s využitím autoenkodéru. V našem řešení jsme využili neuronové sítě s architekturou typu Denoising Convolutional Autoencoder (viz obr. 9.), které inspirované prací [15]. Tato neuronová síť se rozděluje na část kodéru a dekodéru. Kodér nejprve provádí kompresi vstupního optického toku dané buňky do příznakového vektoru o 32 dimenzích. Na základě tohoto vektoru usiluje symetricky odpovídající dekodér o rekonstrukci původního vstupu. Do příznakového vektoru tedy musí být zakódována pouze klíčová informace. Neuronová síť byla natrénována přístupem bez učitele na datové sadě Train Station Dataset [16].



Obr. 9: Architektura použité neuronové sítě typu Denoising Convolutional Autoencoder.

Příznakové vektory dané buňky jsou shlukovány pomocí metody Birch [17]. Tato metoda umožňuje online shlukování příznakových vektorů po jednotlivých dávkách. U nově příchozího příznaku je porovnána jeho vzdálenost od středů existujících shluků. Příznak je poté přiřazen k existujícímu shluku, nebo mu je v případě překročení nastavitelného prahu vytvořen nový shluk. Ke každému shluku je uchováván počet jemu náležících vzorků. Nově příchozí vektor je tedy přiřazen k některému ze shluků a s využitím celkového počtu vzorků daného shluku lze normalizovat úroveň anomálie.

Metoda byla pro účely projektu **testována na videích** pořízených přímo za řešení úlohy specifikované projektem. Videa obsahují příklady pořízené jak statickou kamerou, tak kamerou umístěnou na mobilním snímacím zařízení. Základním vzorem je situace, kde je opakovaný pohyb osob(y), který se po nějaké době nečekaně změní. Metoda byla testována na situacích reprezentující požadavky potenciálních uživatelů, a na této omezené doméně vykazuje očekávané chování. Příkladem je situace se záznamem sportovního hřiště snímaného pomocí dronu z výšky přibližně 60 metrů. Po obvodu hřiště se pohybuje skupina 23 osob podle nacvičeného scénáře. Scénář zahrnuje náhlé zastavení, změnu směru pohybu, chaotický útěk a simulovaný silný záchvěv dronu. Od metod se očekává vyhodnocení vysoké úrovně anomálie v oblastech s výskytem výše popsaných jevů. Obrázek 10. zobrazuje zvýšený výskyt nestandardního pohybu ve scéně.



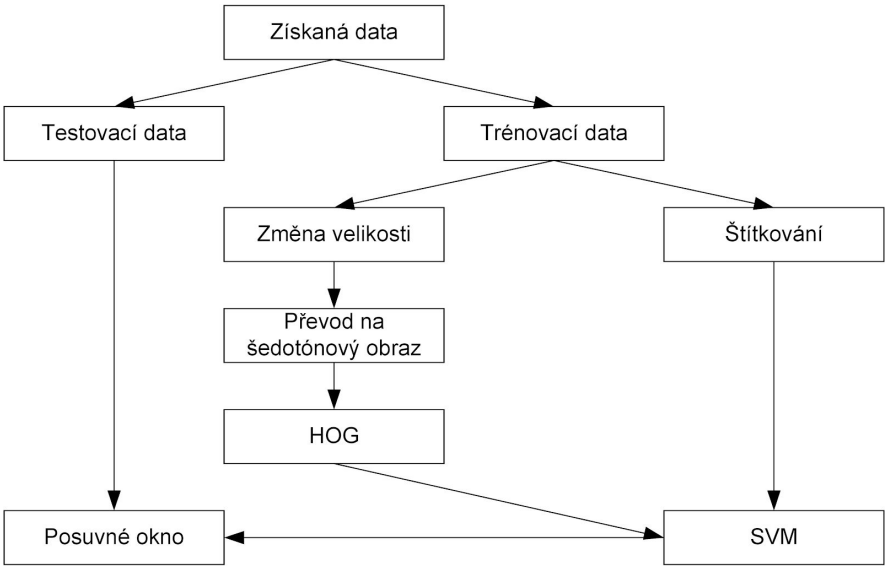
Obr. 10. Ukázka výstupu metody při výskytu nečekané změny pohybu osob: náhlá změna směru pohybu (vlevo) a chaotický útěk (vpravo).

Detektor zbraní ve scéně

Prvním krokem je definice vhodného datasetu. Vzhledem k tomu, že jich volně k dispozici není velký počet, bylo rozhodnuto využít detekci jen krátkých zbraní. Z pohledu statistik je tento typ zbraní nejčastěji používaný při páchání trestné činnosti. A tomu také odpovídá rozsah a množství dostupných datasetů.

Klasické metody (HOG a SVM)

Jako první je sestrojen model, který využívá klasických metod, které nevyužívají konvoluční neuronové sítě. Popis jednotlivých kroků vykonávaných v rámci prvního modelu je možné vidět na obrázku 11.



Obrázek 11: Sekvence kroků u modelu využívající klasické metody.

Pro tento model je využit dataset, který je dostupný z práce [18]. Dohromady má 9.857 obrázků rozdělených do 102 tříd. Třída krátkých zbraní (AAAPistol) má 795 obrázků. Ostatní třídy obsahují obrázky, které nejsou zbraně. Pro tento model byl dataset upraven tak, že byl počet tříd zredukován na dvě (obsahuje zbraň a neobsahuje). Obrázky jsou předzpracovány a první úpravou je transformace na stejnou velikost (128x128). Další změna je převedení do odstínů šedi.

Práce s modelem pokračuje tvorbou pole labelů pro vstupní data a pole deskriptorů pro všechny vstupní obrázků (k tomu se využívá HOG metody). Pro klasifikaci se využije SVM. K natrénování klasifikátoru se použije pole příznaků a labelů. Poslední důležitou částí je posuvné okno. Tam se místo obvyklé změny velikosti okna upravuje velikost prohledávaného obrázku. Velikost okna tím zůstane fixní. K uložení a práci s různými velikostmi obrázků využíváme tzv. model obrazových pyramid. Je to víceúrovňová reprezentace obrázku. Ve spodní vrstvě se nachází původní velikost obrázku a v každé další zmenšenina toho původního, dokud se nedosáhne určená minimální velikost.

Zdrojové obrázky datasetu použitého při testování zobrazovaly zbraně jako selektivní objekty, tedy s uniformním pozadím a bez přítomnosti čehokoliv jiného v obraze. Pro detekci jsme jako vstupní data použili reálné (filmové) scény, kde se nacházely osoby držící zbraň. Ačkoli byl model schopen ve fázi testování pozitivně klasifikovat s vysokou mírou pravděpodobnosti, výsledky při detekci nebyly uspokojivé. Vykazoval totiž vysokou míru falešných nálezů, především ve scénách, jako je například obloha. Zbraň byl schopen detekovat zejména podle

oblasti spouště, která se však při zbrani, kterou drží ruka, nenachází v takovém tvaru, jaký detektor hledal. Toto zjištění považujeme za klíčový důvod, proč tento model nedetekoval zbraň drženou rukou.
Nepodařilo se nám najít žádné nastavení parametrů HOG deskriptoru, které by ve finální fázi detekce reálných obrazů zvoleného datasetu vedlo, nebo se alespoň dostatečně přibližovalo k výsledkům, které bychom mohli považovat za v praxi použitelné. Tento model tak nepovažujeme jako úspěšný ani z hlediska detekčních výsledků, ani z hlediska rychlosti.

YOLO

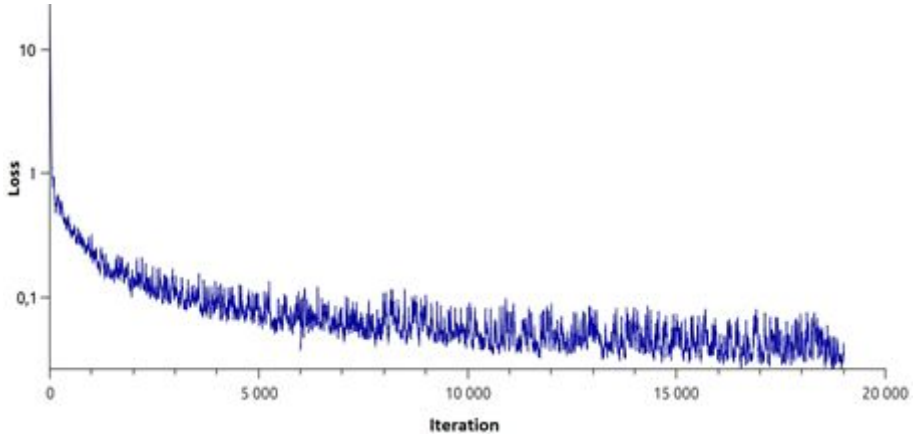
Prioritou tohoto modelu byla dosažení detekce zbraně v reálném čase. Pro otestování modelu takového typu je potřeba přihlédnout i k časové náročnosti a limitech dostupného hardwaru. Po zvážení možností byla využita síť Yolo_v3, konkrétně její Tiny verze, která sice nedosahuje takovou přesnost, ale je poměrně malá. Díky tomu se dá v rozumném čase natrénovat. Architektura této sítě je zobrazena v tabulce 1.

Tabulka 1: Architektura Tiny Yolo_v3 modelu.

#	Typ vrstvy	Filtr	Velikost	Krok	Vstup	Výstup
1	Konvoluční	16	3×3	1	416×416×3	416×416×16
2	Maxpool		2×2	2	416×416×6	208×208×16
3	Konvoluční	32	3×3	1	208×208×16	208×208×32
4	Maxpool		2×2	2	208×208×32	104×104×32
5	Konvoluční	64	3×3	1	104×104×32	104×104×64
6	Maxpool		2×2	2	104×104×64	52×52×64
7	Konvoluční	128	3×3	1	52×52×64	52×52×128
8	Maxpool		2×2	2	52×52×128	26×26×128
9	Konvoluční	256	3×3	1	26×26×128	26×26×256
10	Maxpool		2×2	2	26×26×256	13×13×256
11	Konvoluční	512	3×3	1	13×13×256	13×13×512
12	Maxpool		2×2	2	13×13×512	13×13×512
13	Konvoluční	1.024	3×3	1	13×13×512	13×13×1024
14	Konvoluční	256	1×1	1	13×13×1024	13×13×256
15	Konvoluční	512	3×3	1	13×13×256	13×13×512
16	Konvoluční	18	1×1	1	13×13×512	13×13×18
17	Yolo					
18	Route					

19	Konvoluční	128	1×1	1	13×13×256	13×13×128
20	Upsample			2	13×13×128	26×26×128
21	Route					
22	Konvoluční	256	3×3	1	26×26×384	26×26×256
23	Konvoluční	18	1×1	1	26×26×256	26×26×18
24	Yolo					

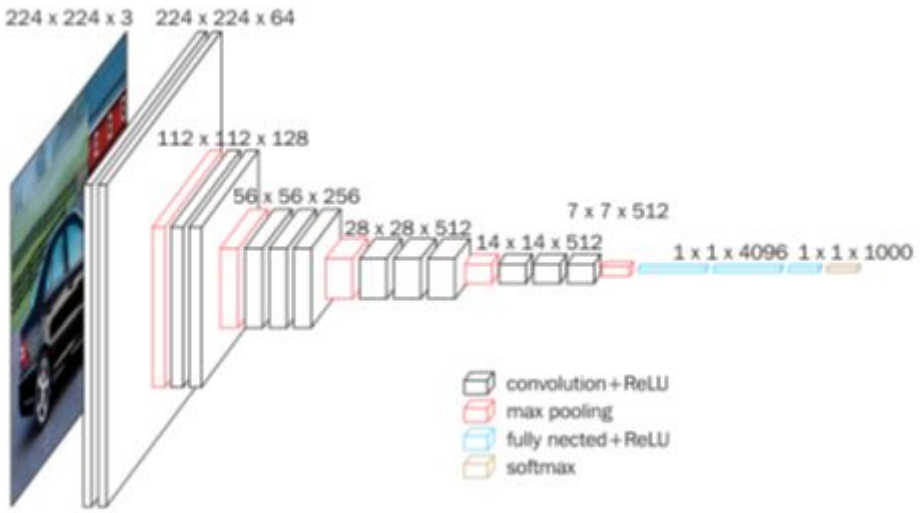
Dataset byl v tomto případě zvolen opět ze stejné práce a je určený pro jejich implementaci CNN. Obsahuje 3.000 obrázků v různých scénách. Z těch bylo vybráno 1.000 obrázků. Pro využití dat v tomto modelu bylo nutné je speciálně anotovat. Anotace definuje třídu a obdélník, ve kterém se objekt nachází. Obdélník je určen poměrem *x*, *y* souřadnic, výšky a šířky k hodnotám celého obrázku. Tento druh anotace lépe funguje při použití různých velikostí obrázků.
Během trénování, jsme ukládali váhy každou 1.000 iteraci. Pro každou váhu jsme následně vyhodnotili *average precision* na testovacích datech. Nejvyšší hodnotou jsme dostali při 19.000 iteracích. Hodnota *average precision* byla 32,16 %, což je velmi blízko referenční hodnotě, kterou Tiny Yolo_v3 dosahuje, a to 33,1 %. Průběh chybové funkce je na obrázku 12.



Obrázek 12: Průběh chybové funkce při trénovaní Tiny Yolo_v3.

Model s využitím CNN a sliding window

Na rozdíl od předchozího modelu je cílem statistická přesnost detekce zbraně. Vyzkoušeny byly dvě architektury VGG16 a vlastní navržená architektura. VGG16 (uvedena na obrázku 13) již byla předtrénovaná a toto trénování bylo doplněno tak, aby vyhovovalo klasifikaci zbraní. Vzhledem k tomu, že klasifikátor na bázi CNN potřebuje fixní velikost vstupního obrazu, a není známá velikost zbraně v poměru k velikosti obrázku, bylo potřebné tento aspekt ošetřit. Na konec modelu byly připojeny dvě plně-propojené vrstvy (první má počet filtrů 1.024 a druhá počet tříd, které mají být klasifikovány).



Obrázek 13: Architektura VGG16 [19].

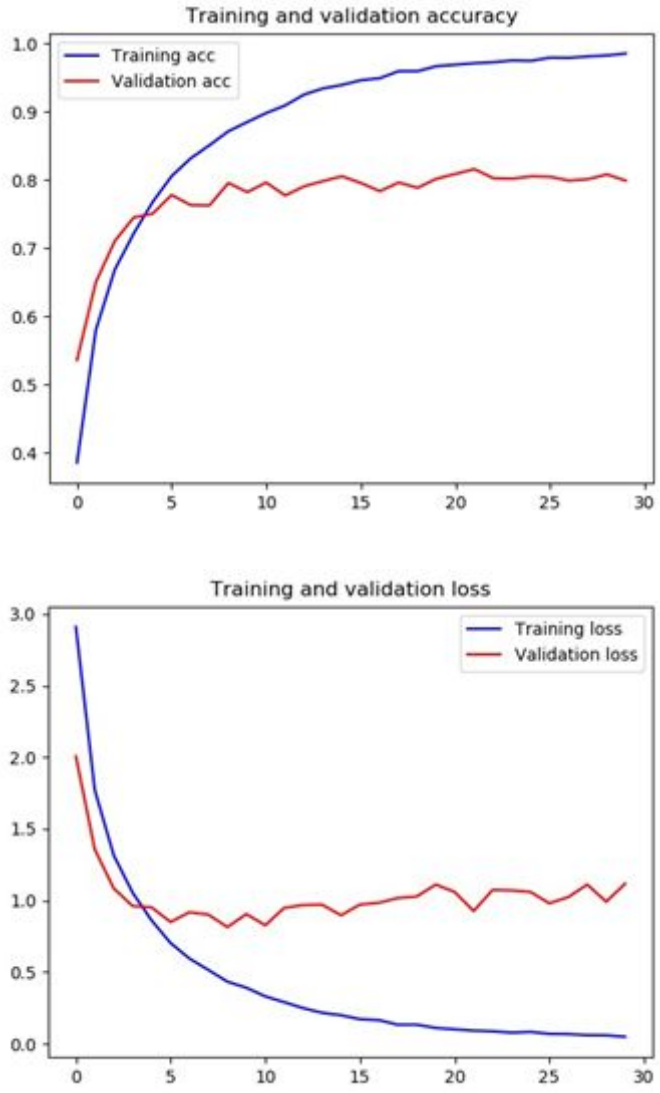
Prvotní návrh architektury je v tabulce 2. Jedná se o tři konvoluční a maxpool vrstvy. Jako aktivační funkce je využito ReLu. Pro zvýšení zobecnění modelu se využívají i dropout vrstvy. Hodnota dropout vrstev byla nastavena na 0,2. Jedna dropout vrstva je vložena i mezi plně-propojené vrstvy (tentokrát s hodnotou 0,5).

Tabulka 2: Architektura vlastního návrhu modelu.

#	Typ vrstvy	Filtr	Velikost	Krok	Vstup	Výstup
1	Konvoluční	32	3×3	1	64×64×3	64×64×32
2	Maxpool		2×2	2	64×64×32	32×32×32
3	Konvoluční	64	3×3	1	32×32×32	32×32×64
4	Maxpool		2×2	2	32×32×64	16×16×64
5	Konvoluční	128	3×3	1	16×16×64	16×16×128
6	Maxpool		2×2	2	16×16×128	8×8×128

Pro trénování je použitý stejný dataset jako v prvním modelu (HOG a SVM). Ten obsahuje 102 tříd 9.857 obrázků, a z toho 795 zbraní (krátkých). Velké množství obrázků však obsahuje čistě bílé pozadí. Prozkoumán byl i vliv počtu tříd na klasifikaci. Proto byla vyzkoušena varianta se 2 (obsahuje zbraň a neobsahuje zbraň) i původní ze 102 třídami.

Jelikož dosahoval binární klasifikační model na bázi VGG16 vyšší průměrné přesnosti, rozhodli jsme se využít právě tento model pro tvorbu klasifikátoru. Průběh trénování můžeme vidět na obrázku 14.



Obrázek 14: Průběh trénování klasifikace do 102 tříd.

Na testovacích datech dosáhl *average precision* 97,79 %. Při výsledné detekci tato verze modelu prokazovala mnohem lepší výsledky než předchozí. Na vzorku 12 scén, které dohromady obsahovaly 13 zbraní, jsme experimentálně určovali hodnotu prahu. Při hodnotě prahu 0,999 byl počet skutečně positivních nálezů 12, skutečně negativních také 12, a jeden případ falešně negativní detekce.

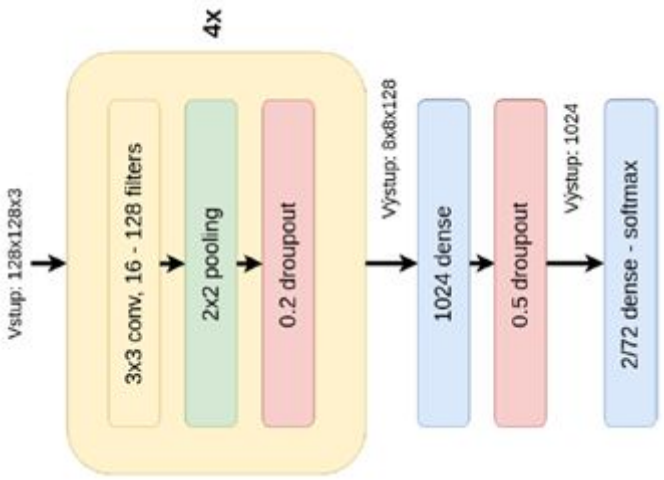
Příklady úspěšné detekce jsou uvedeny na obrázku 15.

Obrázek 15: Příklady úspěšných detekcí zbraní.

Popis zbraní

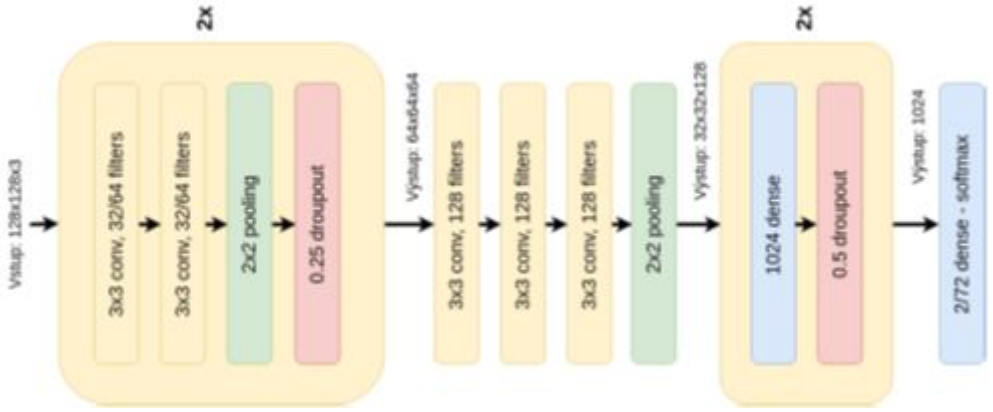
Tato část je věnována klasifikaci popisu zbraní. V tomto případě se jedná o délku zbraně a její natočení. Hlavním nástrojem byly konvoluční neuronové sítě. Pro možnost porovnání výsledků byly navrženy dvě sítě.

První z nich je inspirovaná architekturou AlexNet – model obsahuje 15 vrstev (4 konvoluční, 4 maxpooling, 5 dropout a 2 plně-propojené vrstvy). Velikost vstupních dat do první vrstvy je 128×128×3. Ve všech konvolučních vrstvách jsou použity filtry velikosti 3×3 s krokem 1 a nulovým zarovnáním. V modelu se s každou konvoluční vrstvou zdvojnásobuje počet filtrů z počátečních 16 až na 128 v poslední vrstvě. Pooling (resp. maxpooling) vrstvy mají velikost filtru 2×2 s posunem 2 v každé ose. Za nimi se pak nachází dropout vrstva s nastavením 0,2 (tzn. 20 % náhodných propojení se ignoruje). Po 4 blocích konvoluční, pooling, a dropout vrstvy následuje dense vrstva s počtem propojení 1.024 a dropout vrstva s nastavením 0,5. Jako poslední je dense vrstva s 2 nebo 72 propojeními a softmax klasifikátorem. Počet výstupů závisí na tom, zda se určuje typ nebo náklon zbraně. V celém sítu jsou použity ReLu aktivační funkce. Architektura sítě je uvedena na obrázku 16.



Obrázek 16: AlexNetLike navrhovaná architektura.

Druhý navržený model je inspirovaný architekturou VGG sítí. Pro zrychlení trénování je síť o dva bloky vrstev menší a konvoluční vrstvy obsahují méně filtrů. Celkově síť obsahuje 2 bloky s 2 konvolučními (počet filtrů 32 a 64), pooling a dropout (0,2) vrstvami. Poté následují 3 konvoluční vrstvy (počet filtrů 128) a pooling vrstva. Jako poslední jsou 2 bloky s dense (počet propojení 2.048) a dropout vrstvou (0,5). Poslední výstupní vrstva obsahuje 2 nebo 72 propojení se softmax klasifikátorem. Každá konvoluční vrstva obsahuje filtry o velikost 3×3, krokem 1 a použitím nulového doplňku. Pooling vrstvy jsou typu max, velikost filtru 2×2 a posun 2 po každé ose. V celé síti je použita ReLu aktivační funkce. Architektura sítě je znázorněna na obrázku 17.



Obrázek 17: VGGLike navržená architektura.

Typ (délka) zbraně

Při klasifikaci délky zbraně jsou využity i klasické metody. Pro ně je potřeba předzpracovat vstupní data pomocí konverze do odstínů šedi a HOG. Na tato data byl využit klasifikátor K-Nearest-Neighbour (KNN, k nejbližších sousedů) a SVM. Ke klasifikaci byly však použity i neuronové sítě. Předzpracování dat pro tento přístup byl jen v normalizaci RGB hodnot, nastavení rovnoměrné velikosti stran obrázků a augmentaci dat pro zvětšení počtu vstupních dat.

Dataset pro tuto klasifikaci byl vybrán jako kompilát 3 zdrojů (IMFDB, ImageNet a Google). IMFDB je databáze záběrů z filmů, ve kterých se nacházejí zbraně – obsahují nejen celé scény, ale i samostatné obrázky zbraní, které se v dané scéně nacházejí. Pro klasifikaci je však potřeba, aby obrázek obsahoval pouze zbraně, a proto bylo potřeba obrázky ručně vyfiltrovat. Následně bylo nutné udělat ručně klasifikaci na dlouhé a krátké zbraně. ImageNet je databáze obrázků, která obsahuje víc než 14 milionů obrázků ve více než 21 tisících kategorií. Výhodou této databáze je, že je již anotovaná. Pro doplnění a zvětšení počtu obrázků je možno využít i služeb vyhledávání Googlu. Celkový počet využitých obrázků je uveden v tabulce 3.

Tabulka 3: Podrobné počty trénovacích dat.

Zdroj	Počet krátkých zbraní	Počet dlouhých zbraní
IMFDB	647	670
ImageNet	730	94
Google	0	86
Dohromady	**1.377**	**850**

Následně proběhla augmentace dat. To je jedna z možností, jak navýšit jejich množství a zlepšit generalizaci. Použité metody jsou: rotace (o úhel 0-180°), překlopení (vertikální nebo horizontální), posun (v ose *x* nebo *y* až o 15 % délky, resp. šířky). Při rotaci obrázků mohou vzniknout místa bez určené barvy – v takovém případě je barva těchto „neznámých“ pixelů vyplněna barvou hraničního pixelu.

Orientace

Dalším cílem bylo určení náklonu zbraně v obraze. Náklon se zjišťuje ve všech třech osách. Pro zjednodušení se využívají pojmy z letectví (roll, yaw a pitch). Klasifikace natočení probíhala s využitím CNN. Výsledek poslední vrstvy softmax je 72 výstupů – ty určují, o jaký úhel je zbraň natočena. Každá ze 72 tříd zastupuje rozpětí 5°. Předzpracování je stejné jako při klasifikaci délky, tzn. normalizace, úprava rozměru vstupu na čtverec a augmentace dat. Pro každý náklon byla natrénovaná samostatná CNN. Přesnost sítě pro určení natočení je průměrný rozdíl mezi skutečnými a klasifikovanými úhly.

Byla využita databáze Free3D. To je databáze 3D modelů zbraní včetně texturních informací v různých formátech. Tím bylo vyřešeno získání 3D modelů. K tomu, aby z nich vznikly anotované verze různých natočení, byl využit SYDAgenerator (nejnovější verzi lze nalézt zde: https://www.fit.vutbr.cz/~igoldmann/app/sydagenerator/). Ten pracuje s 3D modelem a obrázkem pozadí. 3D model umí do pozadí (scény) vložit s využitím jakéhokoliv natočení ve všech třech osách, zároveň u toho automatizovaně vytvořit anotaci vytvořené rotace. Využitím 10 pozadí a pěti 3D modelů (3 pro dlouhé zbraně a 2 pro krátké) bylo vytvořeno dostatečné množství dat pro otestování přístupu s využitím CNN - pro *pitch* to bylo 1.480 obrázků, pro *roll* 4.450 obrázků a pro *yaw* potom 4.584 obrázků.

Přesnost je zde definovaná pomocí tzv. chybové matice. A vychází se vzorce *Přesnost* = (TP+TN) / (TP+TN+FP+FN), kde TP (*true positive*), FP (*false positive*), FN (*false negative*) a TP (*true positive*). V tabulce 4 je uvedeno celkové srovnání výsledků pro určení typu zbraně. Z těchto výsledků je patrné, že navrhovaná architektura AlexNetLike dosáhla nejlepší částečné, ale hlavně i celkové úspěšnosti a to až 83,14 %. Barevně jsou vyznačeny nejlepší (zelené) a nejhorší (červené) výsledky pro jednotlivé přesnosti.

Tabulka 4: Celkové srovnání pro určení typu zbraně.

Metoda	Dlouhá zbraň – přesnost	Krátká zbraň – přesnost	Celková přesnost
SVM	61 %	59 %	**59,65 %**
SVM – linear	57 %	62 %	**59,29 %**
KNN - 1	70 %	69 %	**69,49 %**
KNN - 5	70 %	68 %	**69,22 %**
AlexNetLike	94 %	73 %	**83,14 %**
VGGLike	87 %	48 %	**67,06 %**

Tabulka 5 dále zobrazuje výsledky pro jednotlivé osy a dvě navrhované architektury. Je jasně vidět, že architektura AlexNetLike řádově překonala úspěšností druhé navrhované architektury. V hlavičce tabulky je uvedeno *p*, které označuje hodnotu prahové hodnoty pro určení správné predikce. Barevně jsou znovu vyznačeny nejlepší a nejhorší výsledky.

Tabulka 5: Souhrn srovnání dosažených výsledků pro určení náklonu zbraně.

Osa rotace	Metoda	Přesnost pro $p = 5$	Přesnost pro $p = 10$
Pitch	AlexNetLike	85,14 %	**92,34 %**

	VGGLike	4,05 %	**7,66 %**
Roll	AlexNetLike VGGLike	91,02 % 3,74 %	**95,51 %** **5,39 %**
Yaw	AlexNetLike VGGLike	49,71 % 3,78 %	**54,65 %** **5,96 %**

Rozpoznání natočení obličeje

Obličejové rysy (*facial landmarks*) jsou dnes jednou z nejpoužívanějších biometrických charakteristik. Vzestup těchto metod úzce souvisí s úspěchy hlubokých (konvolučních) neuronových sítí (CNN nebo DCNN). Tím se výrazně zvýšila úspěšnost rozpoznávání obličeje.

Tyto metody se však stále potýkají s problémy, pokud není obličej umístěn frontálně k fotoaparátu. Metody, které se snaží vyrovnat s neobvyklou orientací obličeje, jsou založeny na klíčových bodech obličeje (*facial key points* - FKP) [20]. FKP jsou obvykle detekovány tak, že určují polohu důležitých částí obličeje (jako jsou oči, nos, rty atd.). Malá rotace nebo naklonění hlavy není problém - FKP jsou v podobných pozicích a jejich vzájemný vztah je rovněž obdobný.

Jedním z cílů našeho výsledku je umožnit použití v realistických scénářích. Znamená to, že nelze předpokládat, že poloha obličeje bude dokonalá, nebo že kvalita obrazu bude splňovat některé běžné standardy. Nejedná se o kvalitu v rozsahu použité kamery nebo rozlišení, ale v rozsahu uživatelské spolupráce se systémem dohledu. Dnes se pro tyto účely vytvářejí speciální databáze zvané „*in-the-wild*“. Tyto databáze jsou získávány (nebo upravovány) ze skutečných videí, kdy osoby nejsou poučeny a ani nedodržují žádné pokyny pro nasnímání.

Faktem zůstává, že úspěšnost rozpoznávání tváří rychle klesá, pokud není algoritmus připraven na tento druh obrázků. Z tohoto důvodu existuje několik dostupných databází tváří “*in-the-wild*”. V naší práci je použita databáze *Wider Facial Landmarks in-the-Wild* [21, 22], neboť obsahuje problematické obrázky a rovněž všechny tyto obrázky jsou ručně anotovány s 98 FKP (jak již byly popsány v předchozí části). Ukázky obrázků lze vidět na obrázku 18. Každý řádek tohoto obrázku také ukazuje jeden typ „defektů“ přítomných v databázi. Jedná se o velkou pózu (první sloupec), výraz (druhý sloupec), neobvyklé osvětlení (třetí sloupec), (nadměrné) použití make-upu (čtvrtý sloupec), okluzi (pátý sloupec) a rozmazání (šestý sloupec).

Obrázek 18: Ukázkové obrázky z databáze WFLF (převzato z [21]).

Celkově databáze obsahuje 10.000 obrázků tváří s těmito 98 anotacemi FKP - anotací atributů (typy „defektů“), což umožňuje lepší pochopení toho, co se děje s testovaným algoritmem.

Předzpracování a augmentace dat

Neuronové sítě vyžadují standardizovanou velikost obrázků. To je první krok k určení vhodné velikosti. Pokud je obrázek příliš malý, pak existuje vysoká možnost, že by některé FKP byly mimo obraz. Na druhou stranu větší velikost znamená větší CNN a delší dobu trénování detektoru. V [20] zmíněný přístup používá velikost 98×98 pixelů - jak je vidět, obličej se této velikosti stěží hodí, velmi často chybí body na bradě a tvářích. Toto rozlišení není dostatečné, ve [23] je velikost obrázku definována jako 224×224 pixelů, což vypadá realističtěji. Protože se v této práci používá více FKP, bylo rozhodnuto mírně zvětšit velikost na 288×288 pixelů. Před změnou velikosti obrázků v databázi na požadovanou velikost je třeba provést jeden důležitý krok - je to převod obrázku do odstínů šedé. Jiné barevné kanály by přidaly CNN více složitosti (dimenze).

Existuje několik způsobů, jak změnit velikost obrázků v databázi. Může to být jen oříznutá část obrazu, změna měřítka celého obrázku atd. Obvykle se k tomu používají metody rozšiřování dat. To znamená, že z jednoho obrázku v databázi je generováno několik obrázků pro trénování CNN. Možnosti jsou: Změna měřítka celého obrázku - kde je celý obrázek mírně zmenšen nebo zvětšen a poté jsou použity metody oříznutí; převrácení - vodorovné a / nebo svislé převrácení celého obrázku; oříznutí - kde je část obrázku požadované velikosti oříznuta z celého obrázku; oříznutí měřítka obrázku - kde je oříznutá část zmenšena a poté znovu oříznuta. Souřadnice anotovaných dat musejí být změněny odpovídajícím způsobem (zmenšeny, převráceny nebo oříznuty). V této práci je použito pouze oříznutí (ale další možnosti jsou zmíněny jako možnost budoucího rozšíření).

Model CNN

Náš model CNN je založen především na modelu NamishNet [20], ale s jednou důležitou změnou. NamishNet byl navržen tak, aby našel pouze jeden bod, výsledkem sítě je jedna sada souřadnic. Myšlenka byla, že by mělo být spuštěno 15 CNN, aby se získalo všech 15 FKP. To je stěží představitelné u více FKP, proto musela být změněna poslední sada vrstev. Architektura navržené sítě je uvedena v tabulce 6.

Tabulka 6: Architektura navrženého modelu CNN pro detektor FKP.

#	Typ vrstvy	Filtry	Jádro	Jiné
1	Convolution 2D	16	(5, 5)	Padding (same)
2	Activation			Function (ReLu)
3	Convolution 2D	16	(5, 5)	
4	Activation			Function (ReLu)
5	Convolution 2D	16	(3, 3)	
6	Activation			Function (ReLu)
7	MaxPooling 2D		(5, 5)	Strides (2, 2) Padding (valid)

8	Convolution 2D	32	(3, 3)	
9	Activation			Function (ReLu)
10	Convolution 2D	32	(3, 3)	
11	Activation			Function (ReLu)
12	Convolution 2D	32	(3, 3)	
13	Activation			Function (ReLu)
14	MaxPooling 2D		(3, 3)	Strides (2, 2) Padding (valid)
15	Convolution 2D	64	(3, 3)	
16	Activation			Function (ReLu)
17	Convolution 2D	64	(3, 3)	
18	Activation			Function (ReLu)
19	Convolution 2D	64	(3, 3)	
20	Activation			Function (ReLu)
21	MaxPooling 2D		(3, 3)	Strides (2, 2) Padding (valid)
22	Convolution 2D	128	(3, 3)	
23	Activation			Function (ReLu)
24	Convolution 2D	128	(3, 3)	
25	Activation			Function (ReLu)
26	Convolution 2D	128	(3, 3)	
27	Activation			Function (ReLu)
28	MaxPooling 2D		(3, 3)	Strides (2, 2) Padding (valid)
29	Flatten			
30	Dropout			Rate (0.2)
31	Dense	392	Function (ReLu) Regularizer L2 (0.001)	
32	Dropout			Rate (0.2)
33	Dense	196		

Jistě si lze všimnout, že nedochází pouze ke změnám v architektuře. Postupem času se design modelu CNN změnil a více se odchyloval od původního modelu. Navržená architektura získá dobré výsledky u malých testovacích dávek.

Model lze rozdělit do pěti částí: první z nich je pořizování vstupního obrazu a použití konvolučních vrstev s malým počtem filtrů (16), ale spíše vysokou velikostí jádra (5 a 3); poté je podvzorkován vrstvou maxPooling (také docela výrazně s velikostí poolu 5); druhá až čtvrtá vrstva jsou podobné, nastavení jsou běžnější (velikost jádra 3) a jediná věc, která se mění, je množství filtrů použitých v konvolučních vrstvách (od 32 do 128); všechny tyto vrstvy ve čtyřech částech jsou střídány s aktivačními vrstvami, které používají funkci ReLu; pátá vrstva se zplošťuje z 3D na 1D, poté jsou dropout a dense vrstvy, které nakonec končí 196 hodnotami (98 FKP krát dvě souřadnice).

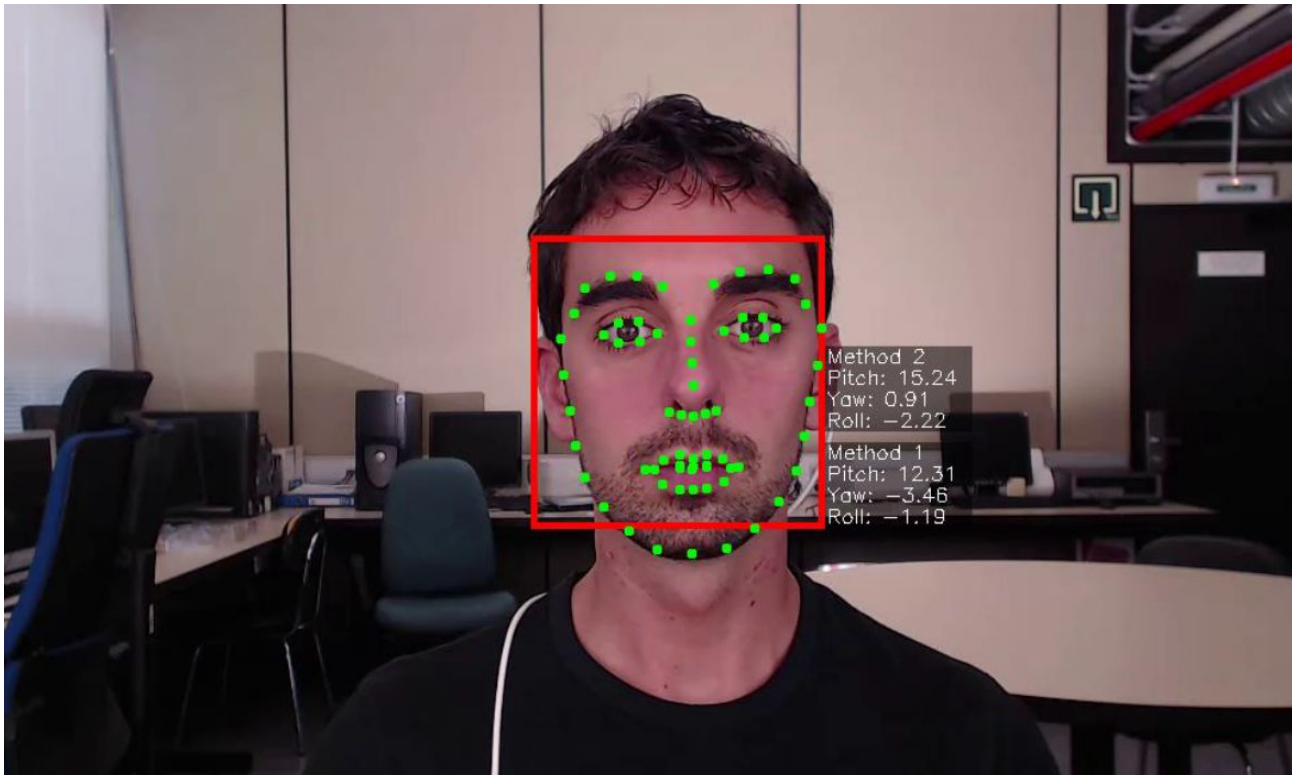
Byla definována architektura, nastavení a celkový přístup k řešení, předběžné výsledky vypadají slibně, ale to nutně neznamená, že se řešení v následujícím optimalizačním období nezmění. V rámci zobrazení obrázků, které byly použity pro testování, jsou výsledky uvedeny na obrázku 19, který zobrazuje ukázkový obrázek, který nebyl ve výcvikové sadě CNN. Samotný obrázek ukazuje mnoho „vad“ přítomných v obrazu. Na tváři jsou vousy, brýle a nějaký předmět, který zakrývá obličej. Hlava je také trochu nakloněná dolů. I když jsou nosní body mírně mimo střední část nosu, celková struktura ostatních bodů je velmi blízko jejich optimální poloze.

Obrázek 19: Ukázkový obrázek s detekovanými klíčovými body (FKP).

Obrázek 20 ukazuje více vzorků, které jsou výrazněji zaměřeny na rotaci hlavy, což je jedno z hlavních zaměření našeho detektoru. Pokud číslujete obrázky zleva doprava, shora dolů, jsou velmi slibné výsledky na třetím obrázku, kde je hlava osoby v zásadě v poloze bočního portrétu a body jasně sledují okraje obličeje. Podobně významné výsledky jsou zobrazeny na čtvrtém obrázku, kde se hlava otáčí v jiné ose a body jsou na bodech. Na druhé straně poslední devátý obrázek zobrazuje osobu, která se dívá vzhůru a je velmi blízko polohy bočního portrétu. Na tomto obrázku mají FKP patřící k lícím offset ke skutečné tváři.

Obrázek 20: Ukázkové obrázky s otočenými nebo nakloněnými tvářemi.

V případě využití obličejových bodů (FKP) jsme schopni dopočítat na základě jejich hustoty a změny pozice navzájem vůči sobě i rotaci obličeje v osách *pitch*, *yaw* a *roll*. Tyto 3 osy přesně defingují rotaci hlavy v prostoru, přičemž nerespektují antropologický model, který má střed otáčení jiný, než je v reálných podmínkách - tento je totiž umístěn na páteři. Ukázka výsledku našeho detektoru natočení obličeje je na obrázku 21. Jsou zde uvedeny výsledky dvou metod, které jsme testovali, přičemž obě vykazují odchylku v řádu několika stupňů, což není pro zpracování dat kritické. Připravujeme se na rozsáhlé testování za využití SYDAgenerator (nejnovější verzi lze nalézt zde: https://www.fit.vutbr.cz/~igoldmann/app/sydagenerator/), kdy je možné vygenerovat z 3D modelu libovolně rozsáhlou databázi s různě natočenými obličeji, přičemž generátor sám data anotuje, tj. ukládá informace o *pitch*, *yaw* a *roll* do separátního souboru. Tato data lze pak porovnat s výsledky zvolených metod.



Obrázek 21: Ukázka reálného rozpoznání natočení obličeje.

Pokročilejší metody dotazování a detekce

V rámci projektu byly zkoumány možnosti definice, vyhledávání a detekce složitěji definovaných událostí, než umožňuje pouhá filtrace (offline a online) nastavením jednoduchých podmínek pro hodnoty atributů detekovaných událostí. Další oblastí bylo přibližné podobnostní vyhledávání a možnost jeho využití.

Problematika složitých událostí je ve světě zkoumána jak z hlediska teoretického (zejména formální popis definice událostí založený na kalkulu událostí (event calculus), např. [25]), tak praktického využití (např. [26] pro lodní dopravu). Tyto přístupy zpravidla jsou založeny na vzájemném vztahu trajektorií pohybujících se objektů.

V našem případě jsme byli omezeni dostupnými informacemi extrahovanými z videa. Protože jsme chtěli podstatu pokročilejšího dotazování a detekce založit na informacích dostupných ze všech snímacích modulů, omezili jsme se na časo-prostorovou složku společné definice elementární události detekované na snímku videa, tj. čas snímku a ohraničující obdélník objektu.

Jako příklady takových složitějších událostí byly zvoleny události *vystoupení osoby z automobilu* a *zastavení provozu*, ke kterému může dojít například v důsledku nehody.

Událost vystoupení osoby z automobilu

Vystoupení osoby je složenou událostí, která kombinuje elementární události detekce automobilu a detekce osoby. Doplňujícími podmínkami pro událost vystoupení osoby pak bylo, že automobil stojí a osoba je v jeho blízkosti. Vzhledem k tomu, že modul sledování objektů poskytuje také informaci o nových objektech ve videu, lze odpovídajícího atributu s výhodou využít a omezit výskyt osoby v blízkosti nově detekované osoby. Událost vystoupení osoby z automobilu je potom

definována jako situace, kdy je ve videu detekována nová osoba v blízkosti stojícího automobilu. Blízkost je vyjádřena překrytím ohraničujících obdélníků obou objektů.

Jádro příkazu SQL, který takové složené události najde, má potom podobu:

```
SELECT …
FROM car_data cd, person_data pd
WHERE pd.obj_class='person' AND pd.obj_track_status='new' AND cd.obj_class='car'
      AND pd.frame_ts = cd.frame_ts AND cd.obj_speed_px IS NULL
      AND overlap(pd.obj_bbox,cd.obj_bbox);
```

kde *car_data* je tabulka, ve které jsou uložená data elementárních událostí detekce automobilu, analogicky *person_data* je tabulka, ve které jsou uložená data elementárních událostí detekce osoby, a *overlap()* je funkce vyhodnocující překrytí ohraničujících obdélníků. Pokud snímací modul detekuje jak automobily, tak osoby, jde o dotaz nad jednou tabulkou.

Při offline dotazování je typicky příkaz doplněn podmínkou pro časový interval, který nás zajímá. Ukázka detekované události je na obr. 22.

Experimenty ukázaly, že při detekci se objevuje poměrně hodně falešných poplachů vlivem nedokonalého trasování pohybu objektu, v tomto případě osoby. V některých případech je osoba označena jako nová, i když je skutečnosti na předchozím snímku už byla. Toto se negativně projevuje hlavně ve sledovaném prostoru, kde je výraznější pohyb osob. Pokud jde o místo s malou četností pohybu osob, mělo by být množství falešných poplachů výrazně nižší. V každém případě i při existenci falešných poplachů může takové dotazování zefektivnit prohlížení videa při hledání událostí tohoto typu.

Stejný princip lze použít i pro online detekci. Rozdíl spočívá jen v tom, že dotaz na událost se provádí opakovaně s určitou periodou.

Událost zastavení provozu

Událost zastavení provozu jsme definovali jako situaci, kdy jsou na snímku stojící automobily a po jistou dobu se na snímku neobjeví žádný pohybující se automobil. Takto definovaná událost je potom parametrizovaná minimálním počtem stojících automobilů a minimálním intervalem, ve kterém se neobjevil žádné jedoucí automobil.

Jádro příkazu SQL (v tomto případě s procedurálním rozšířením), který takové složené události najde, má potom podobu:

```
IF car_count >= car_count_threshold THEN
    IF NOT EXISTS (SELECT *   -- Bylo nějaké auto v pohybu?
                           FROM car_data cd
                           WHERE cd.obj_class='car' AND cd.obj_speed_px >= speed_px_min
                               AND cd.frame_ts BETWEEN last_frame_ts - rest_time AND last_frame_ts)
      THEN res = TRUE;
   END IF;
END IF;
```

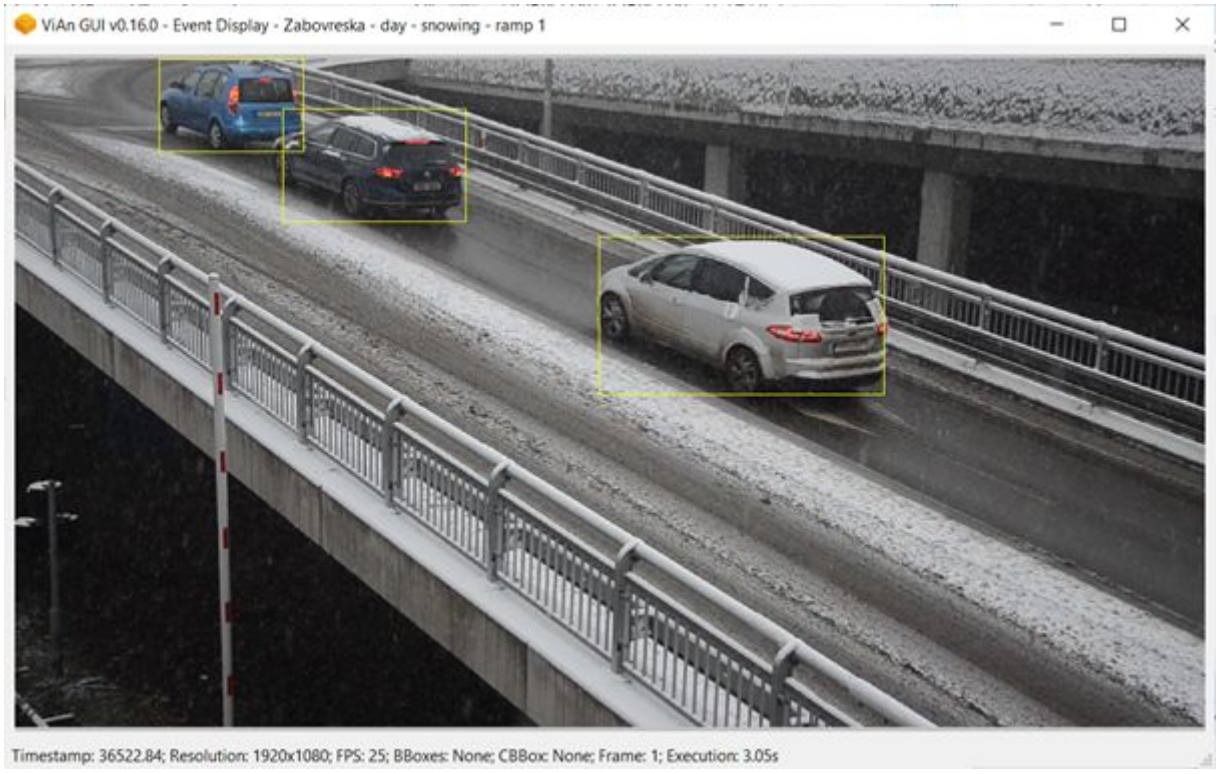
kde *car_count_threshold* je nastavený minimální počet stojících vozidel, *car_count* je počet vozidel na snímku (příkaz pro zjištění zde zahrnutý není), *car_data* je tabulka, ve které jsou uložená data elementárních událostí detekce automobilu, výraz *last_frame_ts - rest_time AND last_frame_ts* vymezuje minimální interval bez pohybu automobilů. Proměnná *last_frame_ts* je zjištěné časové razítko posledního snímku intervalu (opět příkaz pro zjištění není zahrnutý) a *rest_time* je parametr události udávající požadovanou minimální délku intervalu bez pohybu. Parametr *speed_px_min* udává minimální rychlost automobilu, aby byl považován za pohybující se.

Obr. 22: Příklad detekce události typu vystoupení osoby z automobilu

Při offline dotazování je typicky zadán ještě časový interval, který nás zajímá. V takovém případě je příkaz zjišťující zastavení provozu prováděný opakovaně v rámci celého intervalu. Ukázka detekované události je na obr. 23. Vzhledem k tomu, že jsme neměli k dispozici vhodné video se zastavením provozu z důvodu nějaké mimořádné události, bylo použito video křižovatky a detekováno zastavení před semaforem.

Opět lze použít stejný přístup i pro online detekci. Rozdíl spočívá jen v tom, že dotaz na událost se provádí opakovaně s určitou periodou.

Každou takovou složitější událost je potřeba řešit samostatnou implementací. Typicky se použijí uložené podprogramy. V našem případě jsme použili uložené funkce napsané v jazyce PL/pgSQL, protože ViAn Server používá SŘBD PostgreSQL.



Obr. 23: Příklad detekce události typu zastavení provozu

Podobnostní vyhledávání

V rámci projektu bylo také provedeno několik experimentů s přibližným podobnostním vyhledáváním vektorů rysů. K tomuto účelu byly využity knihovny FLANN (C++) a NearPy (Python).

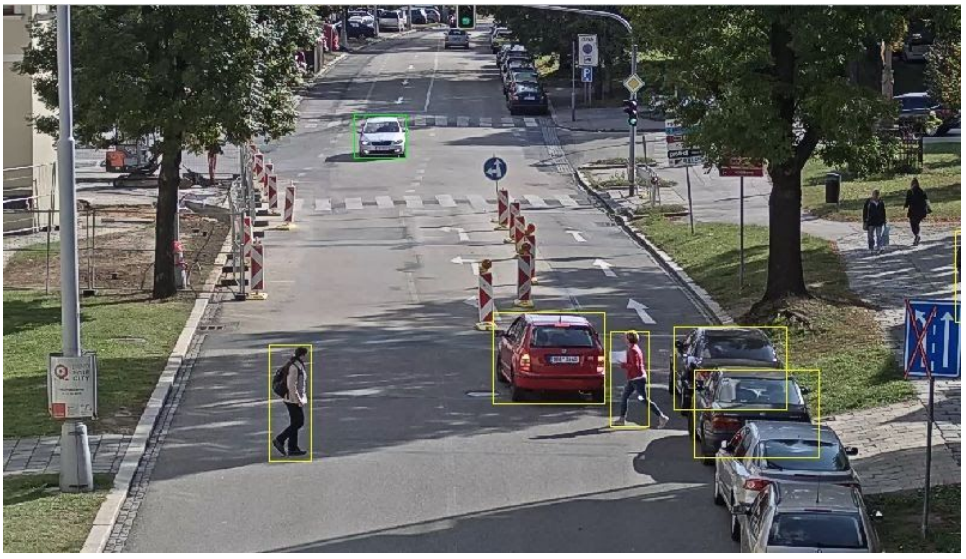
Vzhledem k lepším výsledkům i jednoduššímu použití byla nakonec využita knihovna NearPy [24], která se stala součástí finálního řešení.

Tato knihovna nejprve uloží a indexuje všechny vstupní vektory rysů. Pro každý indexovaný vektor se vygeneruje hash, který je reprezentován jako jedna hodnota typ řetězec. Tento hash je využit jako klíč koše (seznamu vektorů), kam je vektor uložen. Je založen na principu LSH (locality sensitive hashes), kdy podobné vektory mají tendenci být ve stejných koších. Cílem je pak vyhledání nejpodobnějšího koše, ve kterém najdeme nejpodobnější vektory.

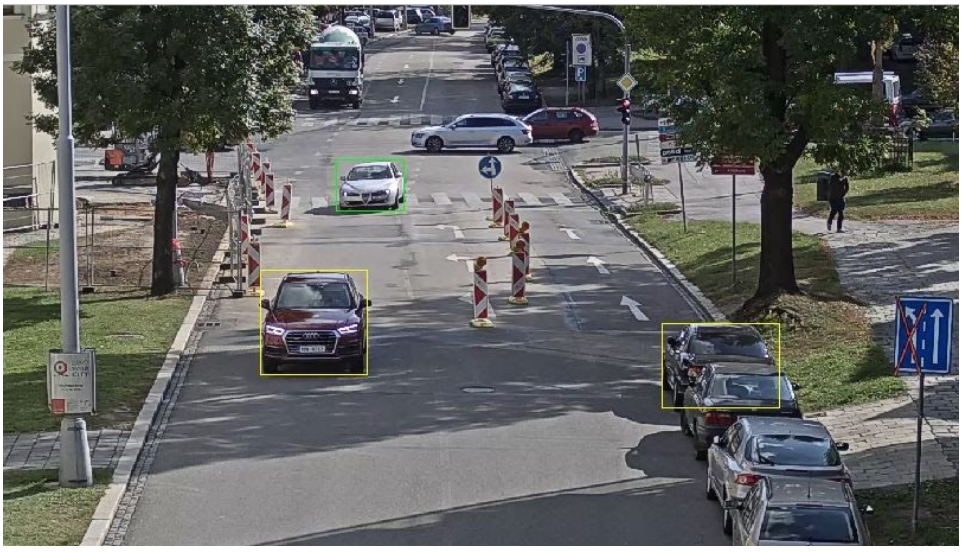
Knihovna poskytuje několik variant LSH, z nichž byla využita varianta RandomBinaryProjections [21]. Tato metoda provádí projekci daného vektoru na n náhodných normalizovaných vektorů v prostoru a vrátí binární řetězec. Jestliže vektor leží na pozitivní straně n-tého normálového vektoru, pak n-tý znak řetězce je '1', v opačném případě '0'. Takto LSH provádí projekci vektoru na jeden z možných košů.

Při zadání vektoru, který je dotazem, je pomocí LSH vyhledán koš nebo několik košů, které tomuto dotazu nejvíce odpovídají. Tím vznikne množina kandidátů a pomocí zvolené metriky podobnosti (např. kosinové vzdálenosti) v nich je vyhledán výsledek.

Zde následuje příklad výsledku přibližného podobnostního vyhledávání. Obrázek 24 (vozidlo v zeleném rámečku) představuje dotaz, obrázek 25 potom výsledek podobnostního vyhledávání v dalších 5 minutách záznamu.



Obr. 24: Příklad dotazu pro podobnostní vyhledávání



Obr. 25: Příklad výsledku podobnostního vyhledávání

Reference

1. Shaoqing Ren and Kaiming He and Ross Girshick and Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv 1506.01497, 2016
2. Andrew G. Howard and Menglong Zhu and Bo Chen and Dmitry Kalenichenko and Weijun Wang and Tobias Weyand and Marco Andreetto and Hartwig Adam: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv 1704.04861, 2017
3. R. Juránek, A. Herout, M. Dubská and P. Zemcík, "Real-Time Pose Estimation Piggybacked on Object Detection," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 2381-2389, doi: 10.1109/ICCV.2015.274.
4. Longyin Wen and Dawei Du and Zhaowei Cai and Zhen Lei and Ming-Ching Chang and Honggang Qi and Jongwoo Lim and Ming-Hsuan Yang and Siwei Lyu: arXiv 1511.04136, 2020
5. SPID: Surveillance Pedestrian Image Dataset, http://best.sjtu.edu.cn/Data/View/972
6. Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun: Deep Residual Learning for Image Recognition, arXiv 1512.03385, 2015
7. J. Sochor, J. Špaňhel and A. Herout, "BoxCars: Improving Fine-Grained Recognition of Vehicles Using 3-D Bounding Boxes in Traffic Surveillance," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 97-108, Jan. 2019, doi: 10.1109/TITS.2018.2799228.
8. Erik Reinhard et al.: “High dynamic range imaging: acquisition, display, and image-based lighting”. Morgan Kaufmann, 2010.
9. F. Durand and J. Dorsey: “Fast bilateral filtering for the display of high-dynamic-range images”, in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 2002. p. 257-266.
10. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR abs/1706.05587 (2017). http://arxiv.org/abs/1706.05587
11. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492, June 2010. https://doi.org/10.1109/CVPR.2010.5539970
12. Přeprava nebezpečných látek a věcí v režimu ADR, Dokumentace BOZP.cz, 28.2.2018, https://www.dokumentacebozp.cz/aktuality/adr-preprava-nebezpecnych-latek-a-veci/
13. Joseph Redmon and Ali Farhadi: YOLOv3: An Incremental Improvement, arXiv 1804.02767, 2018
14. AlexeyAB: Yolo v4, v3 and v2 for Windows and Linux, GitHub https://github.com/AlexeyAB/darknet
15. Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song and Nicu Sebe: Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. DOI: 10.5244/C.29.8, 2015.
16. B. Zhou, X. Wang and X. Tang: Understanding Collective Crowd Behaviors: Learning a Mixture Model of Dynamic Pedestrian-Agents. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012
17. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. SIGMOD Rec. 25, 2 (June 1996), 103–114. DOI:https://doi.org/10.1145/235968.233324
18. Hosang, J.; Benenson, R.; Dollár, P.; aj.: What makes for effective detection proposals? IEEE transactions on pattern analysis and machine intelligence, 2016: s. 814–830

19. Girshick, R.; Donahue, J.; Darrell, T.; aj.: Rich feature hierarchies for accurate object detection and semantic segmentation. UC Berkeley, 2014.
20. Namish A., Grimberge A.K., Vyas R.: Facial Key Points Detection using Deep Convolutional Neural Network – NaimishNet. CVPR, 2017
21. Wider Facial Landmarks in-the-Wild, https://wywu.github.io/projects/LAB/WFLW.html
22. Wu W., Chen Q., Yang, S., Wang Q., Cai. Y., Zhou Q.: Look at Boundary: A Boundary-Aware Face Alignment Algorithm. CVPR, 2018
23. Nishad G.: Facial Keypoint Detection: Detect relevant features of face in a go using CNN & your own dataset in Python. [Online; visited 20.12.2019]. URL https://towardsdatascience.com/facial-keypoint-detection-detect-relevant-features-of-face-in-a-go-using-cnn-your-own-dataset-e09cf359c2bc
24. Krause-Sparmann, O.: NearPy: ANN search in large, high-dimensional data sets (in python), GitHub, 2013. https://pixelogik.github.io/NearPy/
25. Artikis, A., Sergot, M., Paliouras, G.: An Event Calculus for Event Recognition. In: *IEEE Transactions on Knowledge and Data Engineering*, Volume 27, Issue 4, April 1 2015, pp. 895 - 908.
26. Pitsikalis, M. et al.: Composite Event Recognition for Maritime Monitoring. In: Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems, June 2019, pp. 163–174.
27. Gionis, A., Indyk, P. ,Motawani, R.: Similarity search in high dimensions via hashing. In: Proceedings of the International Conference on Very Large Data Bases (VLDB), 1999