VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

MAPOVÁNÍ TRAJEKTORIÍ POHYBU CHODCŮ V ZÁZNAMU POŘÍZENÉM DRONEM

MAPING PEDESTRIAN TRAJECTORIES IN A VIDEO RECORD CAPTURED BY DRONE

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE — **Bc. FILIP ŠŤASTNÝ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. FILIP ORSÁG, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2019/2020

Zadání diplomové práce

Student: **Šťastný Filip, Bc.**
Program: Informační technologie Obor: Inteligentní systémy
Název: **Mapování trajektorií pohybu chodců v záznamu pořízeným dronem**
Mapping of the Pedestrian Movement Trajectory in a Video Recording Captured by a Drone
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte a sumarizujte možnosti detekce osob v obraze pomocí neuronových sítí. Zaměřte se především na řešení, která umožňují detekci osob z výšky.
2. Seznamte se s GPS systémy a komerčně používanými drony. Zjistěte, jakým způsobem je možné určit polohu osoby v záznamu z kamery, pokud známe polohu kamery, natočení a úhel záběru.
3. Navrhněte algoritmus pro mapování trajektorií pohybu chodců v úhlu záběru kamery. Jednotlivé trajektorie budou mít své ID a budou určené souřadnicemi.
4. Navržený algoritmus mapování implementujte. Výstupem programu budou trasy jednotlivých osob zobrazitelné v programu Google Earth.
5. Proveďte experimenty na videích pořízených z dronů. Na základě zjištěných poznatků proveďte zhodnocení funkce algoritmu pro mapování trajektorií pohybu osob. Zaměřte se na vyhodnocení jednoznačnosti trajektorie pro konkrétní osoby.

Literatura:

- STEINER, Ivo a Jiří ČERNÝ. GPS od a do z. 4., aktual. vyd. Praha: eNav, 2006, 264 s. : il. ISBN 80-239-7516-1.
- CIPOLLA, Roberto, Sebastiano BATTIATO a Giovanni Maria FARINELLA. Computer vision: detection, recognition and reconstruction. Berlin: Springer, 2010, 350 s. : il., fot. ISBN 978-3-642-12847-9.

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

Vedoucí práce: **Orság Filip, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 3. června 2020
Datum schválení: 31. října 2019

Abstrakt

Tato práce se zabývá detekcí chodců ve videozáznamu pořízeném dronem pomocí neuronových sítí. Pro sledované osoby poté řeší určování jejich GPS souřadnic na základě aproximace jejich pozice s využitím nadmořské výšky zemského povrchu. Dále mapuje jejich trajektorie v čase pomocí re-identifikace a informací dostupných z dronu. Výsledné trajektorie pak umožňuje zobrazit v programu Google.

Abstract

This master's thesis deals with pedestrian detection using neural networks in a video record captured by drone. Pedestrians are tracked, and their GPS coordinates are calculated using digital elevation models and mapped based on their identity and an information provided by the drone.

Klíčová slova

detekce osob, neuronové sítě, dron, UAV, konvoluční neuronové sítě, sledování osob, mapování trajektorií

Keywords

pedestrian detection, neural networks, drone, UAV, convolutional neural networks, person tracking, trajectory mapping

Citace

ŠŤASTNÝ, Filip. *Mapování trajektorií pohybu chodců v záznamu pořízeném dronem.* Brno, 2020. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Filip Orság, Ph.D.

Mapování trajektorií pohybu chodců v záznamu pořízeném dronem

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doktora Filipa Orsága. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Filip Šťastný
10. června 2020

Poděkování

Děkuji panu doktoru Filipu Orságovi za vedení této práce. Také děkuji panu inženýru Tomáši Goldmanovi za tvorbu zadání a pomoc. Dále bych rád poděkoval panu doktoru Michalu Hradišovi za vedení dobrovolného semináře pro zájemce o konvoluční neuronové sítě, kde jsem získal základ pro některé znalosti využité v této práci. Také děkuji všem, kdo se podíleli na tvorbě testovacího datasetu.

Obsah

1 Úvod **3**

2 Drony, jejich systémy a použití **4**
2.1 Komerční drony 4
2.1.1 Typy dronů dle konstrukce 5
2.1.2 Přehled komerčně používaných dronů 6
2.2 Vybavení a systémy komerčních dronů 7
2.2.1 GPS 7
2.2.2 Výškoměry 9
2.2.3 Polohové a pohybové senzory 9
2.3 Zákonná omezení provozu komerčních dronů 10
2.3.1 Regulace dronů v rámci Evropské unie 10
2.3.2 Omezení dronů legislativou České republiky 10

3 Umělé neuronové sítě **12**
3.1 Umělý neuron 12
3.2 Architektura umělých neuronových sítí 14
3.3 Učení neuronových sítí 15
3.3.1 Učení s učitelem 16
3.3.2 Přeučení a nedoučení neuronových sítí 16
3.4 Konvoluční neuronové sítě 18
3.4.1 Architektura konvolučních neuronových sítí 18
3.5 Detekce osob pomocí konvolučních neuronových sítí 20
3.5.1 Detekce osob z výšky 20

4 Určení pozice a sledování osob **22**
4.1 Určení pozice osob 22
4.1.1 Vzdálenost na základě předpokládané velikosti objektu 23
4.1.2 Triangulace 24
4.1.3 Rekonstrukce 3D scény 24
4.1.4 Určení vzdálenosti objektu na rovném povrchu 25
4.1.5 Určení pozice objektu s využitím nadmořské výšky 25
4.2 Re-identifikace a sledování osob ve videozáznamu 26
4.2.1 Principy strojové re-identifikace 27

5 Implementace **29**
5.1 Návrh 29
5.1.1 Použité nástroje 30

5.1.2 Implementační konvence a modularita aplikace . . . 33
5.1.3 Návrh aplikace . . . 35
5.2 Vstupní data a jejich zpracování . . . 36
5.2.1 Načtení videozáznamu . . . 37
5.2.2 Zpracování obrazu . . . 38
5.2.3 Načtení letového záznamu dronu . . . 39
5.2.4 Úpravy a opravy letových dat . . . 39
5.3 Detekce osob v obraze . . . 40
5.3.1 Implementované detektory osob . . . 40
5.3.2 Export výsledků detekce . . . 42
5.4 Určení GPS pozice osob . . . 43
5.4.1 Určení směrového vektoru osoby . . . 44
5.4.2 Databáze nadmořské výšky . . . 45
5.4.3 Metoda určení vzdálenosti objektu na rovném povrchu . . . 46
5.4.4 Metoda určení pozice objektu s využitím nadmořské výšky . . . 46
5.5 Sledování osob a re-identifikace . . . 48
5.6 Výstup aplikace a zpracování výsledků . . . 50
5.6.1 Soubory typu GPX . . . 51
5.6.2 Soubory typu KML a KMZ . . . 51
5.7 Uživatelské rozhraní a ostatní náležitosti . . . 51

6 Testování a výsledky **53**
6.1 Testovací prostředí . . . 53
6.1.1 Testovací dataset . . . 53
6.1.2 Sada identit testovacího datasetu . . . 54
6.1.3 Testovací hardware . . . 54
6.2 Experimenty a vylepšení jednotlivých částí aplikace . . . 56
6.2.1 Předzpracování obrazu . . . 56
6.2.2 Vliv nepřesností letovém záznamu . . . 58
6.2.3 Detekce osob . . . 60
6.2.4 Výpočet GPS pozice chodců . . . 63
6.2.5 Re-identifikace osob a její vyhodnocení . . . 66
6.2.6 Zobrazení a úpravy výsledků . . . 71
6.3 Výsledky na testovacím datasetu . . . 73
6.3.1 Ukázka výsledků na záznamu č. 1 . . . 74
6.3.2 Ukázka výsledků na záznamu č. 2 . . . 77
6.3.3 Ukázka výsledků na záznamu č. 3 . . . 80
6.3.4 Ukázka výsledků na záznamu č. 4 . . . 82

7 Závěr **85**

Literatura **87**

8 Obsah přiloženého media **92**

Kapitola 1

Úvod

Moderní technologie pomáhají rozvíjet a zjednodušovat rozličné oblasti moderní společnosti. V neposlední řadě mezi ně patří i bezpečnost, konkrétně monitorování osob či oblastí. K tomuto účelu v dnešní době často slouží drony, které mohou např. sledovat nebo prozkoumávat oblasti z nadhledu, bezpečné vzdálenosti a bez přímé účasti pozorovatele. Drony takto pořizují videozáznamy, které je však dále potřeba zpracovat.

Tato práce se věnuje problematice sledování osob ve videozáznamu pořízeném dronem. S využitím počítačového vidění se snaží v obraze automaticky detekovat chodce. Konkrétně k tomu využívá konvolučních neuronových sítí. Na základě dostupných informací z dronu a záznamu poté vypočítá pozice osob v reálném světě formou GPS souřadnic.

Pro nalezené osoby jsou v průběhu záznamu vytvářeny identity. K nim jsou pak nově příchozí nálezy v dalších snímcích přiřazovány na základě navržených postupů pro rozeznávání a sledování osob. Pohyb chodců v průběhu času je pak ukládán jako GPS trajektorie, aby šla sledovat historie jejich pohybu na mapě v geografických aplikacích.

Takovéto zpracování videozáznamu tedy může značně usnadnit již zmíněnou snahu o monitorování oblastí a osob v nich, a tak pomáhat zvyšovat bezpečnost na nejrůznějších místech světa. Podobná opatření jsou např. zaváděna pro hlídání a zabezpečení důležitých objektů, pro zvýšení bezpečnosti na letištích a v bankách nebo dokonce v centrech některých velkých měst.

V některých případech použití však mohou nastat i eticky, či dokonce legislativně problematické otázky. I když tedy používání dronů a počítačové vidění může pomáhat v mnoha oblastech, nemusí to nutně platit vždy a všude. Je proto důležité dodržovat při jejich používání a aplikaci výsledků této práce minimálně legislativu platnou v dané zemi a dbát na to, aby nebyla ohrožena bezpečnost (např. provoz na letištích) či neoprávněně narušeno soukromí jiných lidí. Základy pravidel platných v České republice a Evropské unii jsou proto uvedeny v kapitole 2.

V kapitolách 2, 3 a 4 jsou probrány teoretické základy potřebné k porozumění fungování systému mapování trajektorií. Konkrétní návrh algoritmu a implementace jednotlivých složek práce jsou poté rozebrány v kapitole 5. Zhodnocení dosažených výsledků, ověření funkcionality, řešení problémů a provedená vylepšení jsou poté uvedena v kapitole 6.

Kapitola 2

Drony, jejich systémy a použití

Drony, někdy označované jako UAV[1], jsou bezpilotní letouny schopné provozu bez pilota či jiné lidské posádky [35]. Jejich řízení je zajištěno autonomně palubním počítačem nebo mohou být ovládány dálkově.

Bezpilotní letouny se často používají pro armádní účely, jelikož při jejich nasazení do akce není ohrožen život pilota v případě poruchy či poškození. Dalším důvodem je také absence kokpitu a vybavení pro posádku, díky čemuž se nabízí jak mnohem více prostoru pro užitkovou váhu, tak lepší letové vlastnosti.

V současnosti ovšem narůstá použití bezpilotních letounů i v civilním sektoru. Zde se pro ně vžilo převážně označení dron. Pojem dron ale může popisovat více typů zařízení (ne nutně pouze vzdušných). V kontextu této práce však bude používán jako označení pro komerčně používané bezpilotní letouny.

V civilním sektoru jsou taková zařízení využívána z různých důvodů. Nejčastěji mezi ně patří ekonomické a praktické pohnutky, které vyplývají např. z následujících vlastností komerčních dronů:

- Nízká pořizovací cena – ceny se pohybují v řádově tisících až desetitisících korun českých za běžné až velmi pokročilé modely. Ovšem základní zařízení pro zábavu a rekreaci lze pořídit i v řádech stovek Kč.
- Schopnost dosáhnout zajímavých pozorovacích úhlů – umožní např. pořizování přehledného videozáznamu z perspektivy při venkovních sportech, prohlídkách pojistných událostí[2] nebo také ve filmovém průmyslu. Zde nahrazují používání letadel a helikoptér, čímž podstatně snižují náklady.
- Malá velikost a bezpečná vzdálenost uživatele – lze s nimi pořizovat např. záběry nebezpečných druhů a rizikových oblastí v přírodě.

2.1 Komerční drony

V následující kapitole budou uvedeny obvyklé typy komerčně používaných dronů z hlediska jejich konstrukce. Ta totiž udává klíčové letové vlastnosti, které předurčují použití v konkrétních odvětvích – např. rozhodují o tom, jaký přístroj použít v leteckých závodech a jaký ve filmovém průmyslu.

[1] Obecně užívaná zkratka z angl. označení pro bezpilotní letouny – *Unmanned Aerial Vehicle*.
[2] Příklad: www.ekontech.cz/clanek/cemu-je-dron-pojistovne-usnadnuje-praci-setri-cas-naklady

Dále budou porovnány některé z běžně rozšířených modelů střední a vyšší cenové třídy. Na ně jsou kladeny nejrůznější požadavky z pohledu vybavení a schopností, ale v základu se ty nejčastější shodují v následujících bodech:

- nízká cena,
- dobrá stabilita,
- možnost nést kameru,
- dostatečná doba letu,
- ovladatelnost (rychlost, obratnost, dosah),
- menší rozměry (přenositelnost).

Obrázek 2.1: Ukázka kvadrokoptéry[3].

2.1.1 Typy dronů dle konstrukce

Konstrukce dronů v civilním sektoru se nejčastěji skládá z těla s několika rameny, na kterých se nacházejí vrtule poháněné elektromotory. Na rozdíl od velkých armádních bezpilotních letounů se zde obvykle nepoužívají křídla, která při nízké hmotnosti a malých vzdálenostech doletu nejsou potřeba. Drony jsou pak stabilní a dokáží se vznášet na místě. Celý přístroj je zpravidla ovládaný řídicí jednotkou, která pracuje pouze pomocí regulace otáček rotorů [36].

Platí, že více vrtulí znamená vyšší stabilitu. Naopak se ale snižuje ovladatelnost – dron je jednak těžkopádnější, jednak roste komplikovanost ovládání pro řídicí jednotku [36]. Příklad běžně používaného dronu lze vidět na obrázku 2.1. Na základě konstrukce lze tedy drony dělit do těchto kategorií:

[3]Převzato z: https://static.bhphotovideo.com/explora/sites/default/files/styles/top_shot/public/ts-dji-mavic.jpg

- Trikoptéry – mají tři ramena s vrtulemi. Ty se všechny točí stejným směrem, aby se snáze dosáhlo rovnováhy a stability při vznášení se. Směr letu je určován pomocí naklápění některého z rotorů. Trikoptéry tak bývají obratnější než modely s čtyřmi či více motory, ale méně stabilní.

- Kvadrokoptéry (nebo také kvadrikoptéry) – čtyři vrtule zajišťují dobrou stabilitu a zároveň jsou vhodné pro široké použití díky jednodušší konstrukci, a tím příznivé ceně. Dva protilehlé rotory se točí stejným směrem, zatímco zbylé dva opačným. Pohyb je potom zajištěn zvyšováním a snižováním výkonu motorů. Tyto modely tedy nebývají obratné jako trikoptéry, avšak díky zmíněným vlastnostem jsou nejčastěji používaným typem dronů.

- Drony s více než čtyřmi rotory – mají typicky sudé počty motorů, ramena rozložena do tvaru pravidelných mnohoúhelníků. Střídavě je polovina vrtulí levotočivých a polovina pravotočivých. Tato zařízení bývají stabilnější a díky více motorům mohou často nést i těžší náklad (např. více kamer). Podstatnou výhodou těchto modelů je odolnost vůči případnému poškození či výpadku rotoru – drony dokáží pokračovat v letu a přistát bez újmy drahého vybavení, se kterým pracovaly.

2.1.2 Přehled komerčně používaných dronů

Vlastnosti běžně užívaných komerčních dronů jsou dány především jejich cenou. V tabulce 2.1 jsou porovnány modely střední a vyšší cenové kategorie (řádově tisíce až deseti tisíce Kč). Lze si povšimnout, že některá zařízení navzdory svým malým rozměrům dokáží nést navíc až polovinu vlastní hmotnosti. Je dobré poznamenat, že velikostně přístroje nepřesahují jeden metr, což umožňuje jejich snadnou přepravu a operování v bezprostřední blízkosti překážek.

Dále tabulka 2.1 obsahuje údaje o rychlosti dronů – všechny uvedené modely jsou schopné letět rychleji, než je schopen běžet průměrný člověk. Zmíněné pozitivní parametry komerčních dronů jsou ale částečně vykoupeny velikostí baterie a jejich výdrží v letu – žádný nepřekročil u letové výdrže hranici 30 minut. Dalším údajem je maximální výška letu. Zde se modely velmi liší v rozmezí od 122 m až po extrémních 6 km. Vrchní hranice stoupavosti je limitována také dosahem vysílání ovladače a v praxi často omezována legislativou dané země, např. v České republice je výškový limit stanoven na 300 m (více v kapitole 2.3).

Model	Hmotnost [g]	Rozpětí [mm]	Baterie [mAh]	Náklad [g]	Kvalita kamery [MP]/[pix]	Výška letu [m]	Rychlost [m/s]	Doba letu [min]
Phantom1	840	350	2200	330	-	200	10	15
Phantom2	1000	350	5200	300	-	500	15	25
Phantom3	1280	590	4480	-	12/4K	120	16	25
Phantom4	1380	350	5350	462	12/4K	6000	20	28
MavicPro	743	-	3930	-	12/4K	-	18	27
Q500 4K	1100	565	5400	600	12/4K	122	8	25
3D Solo	1500	460	5200	420	-	122	24.6	25
Spark	300	170	1480	-	12/4K	500	13.9	16

Tabulka 2.1: Přehled běžně užívaných dronů a jejich vlastností [12] a rozšířeno o model Spark od firmy DJI).

Pro potřeby této práce byl vytvořen dataset videozáznamů pořízených dronem DJI Spark. V tabulce 2.1 lze vidět, že v porovnání s ostatními se jedná o menší, lehký model vhodný pro běžné použití. Přesto bez problémů nese kameru zavěšenou na dvouosém stabilizátoru. Ten umí redukovat výkyvy v bočním náklonu a sklápění přídě až do 85°, díky čemuž dokáže dron v praxi pořídit velmi stabilní záběry. Více informací o tomto modelu lze nalézt na stránkách výrobce[4].

2.2 Vybavení a systémy komerčních dronů

Aby mohly drony plnit nejrůznější úkoly v civilním sektoru, jsou vybaveny mnoha systémy, od v současnosti samozřejmých (např. gyroskop) až po specializované (např. laserový výškoměr). Vybavení dronů potřebné různé činnosti a běžně instalované komerční součásti budou rozebrány v rámci této podkapitoly.

Bezpilotní letouny řídí obvykle uživatel dálkovým ovladačem, je tedy potřebné zajistit kvalitní komunikaci mezi ovladačem a letounem, ale stále častěji se objevují systémy pro podporu autonomních činností jako např. návrat k místu vzletu, automatické přistávání či stabilizace při vznášení se na místě i ve větru.

Většina přístrojů je v praxi limitována svou přesností. Někdy se může jednat o zanedbatelné odchylky (např. milimetry až centimetry při měření vzdálenosti lidarem), jindy o výrazné chyby (např. odchylky v řádech metrů u GPS navigace). Pokud dron využívá informace např. k autonomnímu přistání, takové nepřesnosti by mohly vést i k havárii. Pro odstranění šumu se proto využívají různé filtry, přičemž jeden z nejpoužívanějších je Kalmanův filtr (bližší popis v kapitole 5.2.4).

2.2.1 GPS

GPS (*Global Positioning System*) [47][7], oficiálně pojmenovaný jako NAVSTAR, je globální polohový systém sloužící k určování polohy v armádě i civilním sektoru. Je vytvořený a spravovaný Ministerstvem obrany Spojených států amerických. Pomocí GPS je možné určit místo výskytu s přesností na jednotky až desítky metrů [32]. Koordináty jsou možné určit jak v rámci souřadnic 2D (zeměpisná výška a šířka), tak ve 3D (přidána nadmořská výška). Využívá se i při navigaci dronů. Obzvláště užitečný je pro autonomní sledování uživatele nebo návrat ke vzletové pozici [37].

GPS systém byl uveden do provozu v roce 1994, kdy se na oběžné dráze nacházelo celkem 24 GPS družic. Systém však byl plně zpřístupněn až v roce 2000, kdy se USA rozhodly vypnout umělé omezení vysílání pro civilní potřeby. Do té doby mohli využívat plný potenciál GPS pouze armádní složky USA. Počet družic se časem rozšiřoval pro uspokojení celosvětových potřeb. V současné době je aktivních 31 satelitů, což umožňuje zisk spolehlivějších a přesnějších výsledků, ale není podmínkou provozu.

V jeden okamžik by z libovolného místa na Zemi mělo být možné přijímat signál min. 4 z družic, s aktuálním navýšeným počtem satelitů by však tato hodnota neměla klesnout ani pod 6. Nad Českou republikou se běžně vyskytuje 7 až 8 kusů [41]. Ukázku pokrytí GPS lze vidět na obrázku 2.2, kde modře vyznačené místo na Zemi přijímá signál z 9 satelitů, zatímco ostatní pokrývají zbytek planety.

Družice krouží kolem planety ve výšce lehce přes 20 000 km, perioda oběhu dosahuje přibližně 12 hodin. Dosahují velmi vysoké rychlosti kolem 3,8 km/s, což ovlivňuje čas na

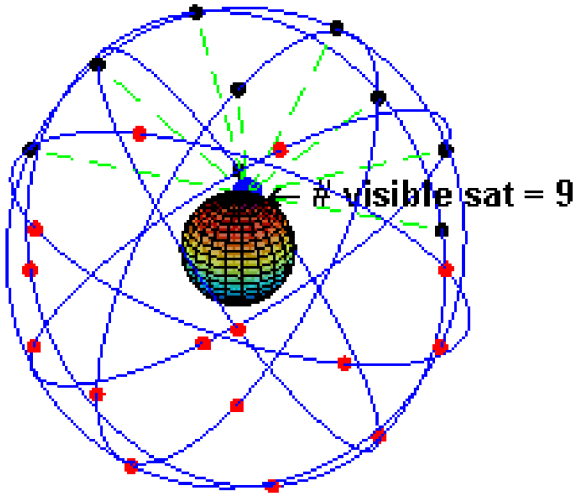
[4]Dostupné online: www.dji.com/cz/spark/info

jejich palubě – dle teorie relativity se nepatrně zpožďuje za pozemskými hodinami. Z tohoto důvodu jsou družice vybaveny extrémně přesnými atomovými hodinami a na zemi se nachází síť měřících stanic, které průběžně signál vyhodnocují a provádí korekci satelitů.

Signál satelitů obsahuje (mimo jiné) čas vyslání a polohu družice. Stanice, která tato data obdrží, je schopná na základě rozdílu časů vyslání a doručení a pozic satelitů vypočítat svoji polohu. Pro určení 2D souřadnic jsou zapotřebí informace z nejméně 3 družic, pro dopočítání 3D koordinátů pak minimálně ze 4. V praxi však bývá použito signálů více, aby bylo měření přesnější. I z důvodu, že přijímače nemají dostatečně kvalitní hodiny, a tak využívají signál z jiné GPS družice jako referenční časový údaj.

Při letu dronu se GPS může využívat nejen pro 2D lokalizaci, ale i jako výškoměr. V tom případě ovšem nastává problém s přesností nejen kvůli obvyklé chybě měření (šum, odchylka hodin, refrakce signálu v atmosféře atp.), ale i kvůli možné nevhodné reprezentaci modelu planety. Standardně se totiž používá pro výpočet tvar elipsoidu zavedený směrnicí WGS84, který může oproti reálnému geoidu navozovat rozdíly od -100 m do +70 m [32]. Na jeho zpřesnění bylo uvedeno několik dalších korekčních směrnic a není vždy jasné, kterou přesně daný GPS lokátor používá. Tento fakt je tedy třeba brát na vědomí, pokud je vyžadována minimální chyba např. při autonomním letu na zadané souřadnice. Zpravidla bývá vertikální chyba přibližně dvakrát větší, než chyba horizontální.

Přesnost určení polohy může být zvýšena i pomocnými metodami. Jedna z nich se nazývá diferenciální GPS. Využívá nehybnou základnu, která se nachází na přesně vyměřené pozici. Druhý přijímač se pohybuje po okolí. Základna vypočítá rozdíl mezi svojí reálnou lokací a pozicí udávanou GPS signálem. Tuto odchylku poté odešle pohyblivému přijímači, který ji přidá ke své lokaci udávané GPS. V závislosti na vzdálenosti od základny je tímto způsobem možné dosáhnout přesnosti např. v řádech centimetrů.



Obrázek 2.2: Ukázka pokrytí planety systémem GPS[5].

[5]Převzato z: https://en.wikipedia.org/wiki/Global_Positioning_System#/media/File:GPS24goldenSML.gif

2.2.2 Výškoměry

Pro některé aktivity dronů (např. automatické přistání) je znalost vzdálenosti od povrchu esenciální. Proto bývají vybaveny senzory na měření letové výšky. U komerčních modelů se nejčastěji jedná o gps, barometry, sonary a laserové výškoměry. Využití radarů není v civilním sektoru běžné. Disponují zbytečně velkým dosah a neposkytují dostatečnou přesnost nad členitým terénem.

Barometr [33] je zařízení sloužící k zjištění letové hladiny dronu. Funguje na principu měření atmosferického tlaku, který se s rostoucí nadmořskou výškou snižuje. Barometr je tedy v podstatě tlakoměr, který převádí naměřený tlak na výškové údaje, které následně digitálně či analogicky zobrazuje. Zjištěné hodnoty tlaku tím pádem musí být vztažené k referenčnímu výškovému bodu, např. hladina moře. Neudává tedy reálnou aktuální vzdálenost přístroje od povrchu.

Pokud ovšem barometr měří relativně k místu vzletu a dron se pohybuje nad rovnou plochou, lze s těmito hodnotami pracovat jako s vzdáleností od povrchu. I tak se ale jedná převážně o přibližné navigační údaje, jelikož počasí a tlak vzduchu se mohou měnit i po dobu letu, a tím zkreslit data z barometru. V rámci krátkých letů a mírných doletových vzdáleností lze data považovat za poměrně přesná, při delších trasách (řádově kilometry a více) však na ně nelze spoléhat, tedy nemohou být použity jako rozhodující údaj např. pro automatické navádění na přistávací plochu.

Sonar [31][28] slouží k určení přesné vzdálenosti od povrchu či překážek, obecně ho lze využít i k mapování 3D struktury terénu. Informace ze sonaru jsou velmi přesné, ale přístroj je použitelný pouze do vzdálenosti jednotek, max. desítek metrů. Je tedy vhodný např. pro automatické přistání či stabilizaci vznášejícího se dronu.

Sonar vysílá max. několikrát za sekundu impulsy vysokofrekvenčních zvukových vln, které se odráží od cíle a vrací se zpět k vysílači. Ten je následně i jejich přijímačem. Na základě doby mezi vysláním a zpětným obdržením signálu přístroj určí vzdálenost překážky. To vše za předpokladu, že jsou známé vlastnosti prostředí (v případě dronů se jedná o teplotu a tlak vzduchu), kterým se zvuk šíří, a tak je možné vypočítat jeho rychlost.

Lidar [28], nebo také laserový výškoměr, slouží stejně jako sonar k určení přesné vzdálenosti od povrchu. Na rozdíl od něj je použitelný na větší vzdálenosti, ale v praxi se obdobně jedná spíše jen o desítky metrů. Název je složený ze slov „light“ a „radar“, což v překladu z angličtiny znamená světelný radar a naznačuje princip činnosti zařízení.

Přístroj vysílá světelný paprsek, který se odráží od povrchu předmětu a po návratu je senzorem zaznamenán. Z rychlosti světla a návratové doby paprsku pak lze dopočítat vzdálenost překážky. Výhodou lidaru je jeho frekvence měření, která jeho díky rychlosti světla podstatně větší, než u sonaru. Nevýhodou je možnost vlivu prostředí na světelný paprsek, který se může šířit nedostatečně či nevhodně např. ve smogu nebo při sněžení.

2.2.3 Polohové a pohybové senzory

Znalost natočení, směru a rychlosti pohybu je nezbytná pro správné fungování dronů. Ať už se jedná o autonomní přistání, nebo jen běžnou činnost řídicí jednotky při koordinaci pohybů. Zmíněné údaje jsou proto měřeny a získávány z gyroskopu, akcelerometru, magnetometru a jejich variací v kombinaci s GPS, navíc jsou jejich výstupy často zastřešovány a agregovány multifunkční měřící jednotkou [51].

Gyroskop [25] je senzor sloužící k určení orientace v prostoru. Jinými slovy detekuje natočení dronu kolem svých os, popř. vůči povrchu, pokud byla provedena kalibrace gyro-

skopu před vzletem. Sestává z několika kol, která jsou zasazena v ložiscích tak, aby mohla volně rotovat. U běžných komerčních modelů se využívá gyroskop se 6 osami.

Akcelerometr [51] pomáhá určit pohyb dronu – jeho směr a velikost akcelerace. Sestává ze součástí, které jsou schopny detekovat zrychlení přístroje a převádět je na elektrický proud. Tyto součástky mohou pracovat na vícero principech – ve dronech se využívají termální senzory na bázi rozpínavosti plynů, které jsou méně náchylné na vibrace produkované vrtulemi.

Magnetometr [51] je elektronická obdoba kompasu, běžně používaná v letectví. Dokáže měřit směr magnetického pole a jeho velikost, což napomáhá k určení polohy a natočení dronu vůči povrchu.

2.3 Zákonná omezení provozu komerčních dronů

Drony nabízí skvělé podmínky pro sledování osob nebo monitorování oblasti. Nikdy dřív nebylo létání takto snadné a dostupné každému. Nebylo jednoduché dostat kameru do míst, kam se člověk sám nedostane. S rozvojem nových možností se tedy vyskytlo i mnoho příležitostí a potenciálně nepříjemných až nelegálních činností. V současnosti už je však většina těchto možností prozkoumaná a ochrana před nimi uzákoněná platnou legislativou. V této kapitole budou rozebrány základní podmínky a omezení provozu civilních bezpilotních letounů jednak z hlediska zákonů Evropské unie (dále jen EU), jednak z pohledu legislativy České republiky.

2.3.1 Regulace dronů v rámci Evropské unie

Evropská unie reguluje provoz létajicích stojů na základě Nařízení Evropského parlamentu a Rady (EU) 2018/1139 ze dne 4. července 2018 o společných pravidlech v oblasti civilního letectví a o zřízení Agentury Evropské unie pro bezpečnost letectví [19].

Podle tohoto nařízení jsou drony členěny do 5 kategorií, na které se vztahují omezení [34]. Drony uvedené v kap. 2.1 v této práci spadají do kategorie C2 (hmotnost méně než 4 kg). Proto nemohou létat přímo nad lidmi, ale mohou se pohybovat v jejich blízkosti a v obydlených oblastech, pokud udržují bezpečnou vzdálenost od nezúčastněných osob.

Dále se na pořizování videozáznamu vztahují zákony spojené s ochranou osobní svobody a soukromí dle platné normy GDPR (*General Data Protection Regulation*) [18]. Podle tohoto nařízení je zakázáno sbírat a uchovávat osobní údaje jiných osob. Ve vztahu k videozáznamu tedy zakazuje uchovávat snímky, které by splňovaly některá z následujících kritérií [46]:

- Obsahují jasně viditelnou tvář osoby.
- Umožňují identifikaci osoby na základě její pozice, adresy či jiných údajů.
- Zachycují detaily osobního či pracovního života.
- Zaměřují konkrétní osobu, obzvlášť po dobu delšího časového úseku.
- Mohly by být využity k ohrožení či nevhodnému jednání s člověkem.

2.3.2 Omezení dronů legislativou České republiky

V rámci ČR se na provoz dronů vztahuje několik zákonů a předpisů:

- letecký předpis – pravidla létání L2, doplněk X[6],
- zákon o civilním letectví[7],
- stanovisko č. 1/2013 od Úřadu pro ochranu osobních údajů[8].

Podle zmíněných zákonů se na drony do 20 kg vztahují podmínky rekreačního používání [46][45]. To limituje horizontální vzdálenost od nezúčastněných osob na min. 100 m, resp. 150 m od obydlených oblastí. Dále je povolena max. výška letu 300 m. To ovšem pouze v případě, že není zvýšená oblačnost a dron se nepohybuje v mracích, k oblačnosti je povoleno přiblížit se max. 1500 m horizontálně a 300 m vertikálně. Toto opatření zajišťuje viditelnost přístroje a prevenci kolize s případnými jinými letouny, které by navíc mohly být i pilotované. Pro drony v hmotnostní kategorii 0–0,91 kg platí, že nesmí přesáhnout letovou hladinu 100 m.

Pro všechny drony dále platí (stejně jako v legislativě EU), že nesmí létat nad lidmi bez jejich svolení. Také se musí vyhýbat bezletovým plochám, jako jsou např. letiště, jaderné elektrárny, vojenské prostory atp. Je zakázáno shazovat dronem za letu jakékoliv předměty. Je povinný oční kontakt se dronem, popř. na něj musí vidět spolupracující pozorovatel, tedy nesmí se létat jen podle záběrů z připevněné kamery.

Navíc se uplatňují i zmíněná pravidla pro ochranu osobních údajů při pořizování záznamu. Z těch lze vybrat např. následující body [45]:

- Nenahrávat obydlí a soukromé aktivity bez svolení.
- Nepořizovat záběry, které snižují lidskou důstojnost.
- Pokud je pořízena nahrávka v rozporu s výše uvedeným, je povinnost ji bez zbytečného odkladu smazat.
- V jasně vyznačeném veřejném prostoru lze záznamy pořizovat volně.
- Je povoleno nahrávat za účelem ochrany svých zájmů (kontrola majetku) a chránit zdraví ostatních i bez jejich předchozího souhlasu.

Dále je také upravena komunikace mezi ovladačem a dronem na základě předpisů týkajících se využití rádiového spektra v ČR, které jsou dostupné na webu Českého telekomunikačního úřadu[9].

[6] https://lis.rlp.cz/predpisy/predpisy/dokumenty/L/L-2/data/effective/doplX.pdf
[7] https://www.zakonyprolidi.cz/cs/1997-49
[8] https://www.uoou.cz/files/stanovisko_2013_1.pdf
[9] https://www.ctu.cz

Kapitola 3

Umělé neuronové sítě

Lidské tělo je ovládáno centrálním nervovým systémem. Ten je složen z množství různě propojených neuronů. Podněty z okolí jsou zaznamenány pomocí smyslových orgánů a množství receptorů v celém těle. Vzruchy se pak šíří přes nervové dráhy nejčastěji do mozku. Mozek podnět v daném kontextu zpracuje, určí odpovídající reakci a také si ukládá potřebné informace související s událostí. Může také upravovat spojení mezi neurony na základě výsledků provedené reakce tak, aby v budoucnu vyhodnotil podobnou situaci co nejvhodněji, jinými slovy se učí.

Jak vyplývá ze samotného názvu, umělé neuronové sítě [38][5] se snaží napodobit princip fungování reálné nervové soustavy. Sestávají z vrstev tvořených umělými neurony, které mohou být navzájem různě propojeny. Architektura a části umělých neuronových sítí jsou podrobněji popsány dále v této kapitole. Vrstev a neuronů bývá v běžných neuronových sítí mnoho. V kombinaci s ohromným množstvím jejich propojení, ovlivňujících vstupní informaci, pak sítě dokáží obsáhnout velké množství informací a učit se přizpůsobovat reakce na různé vstupy.

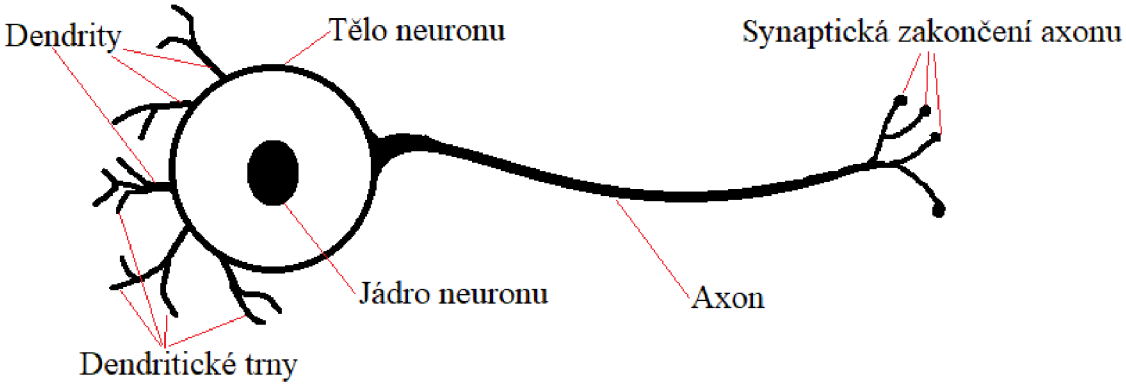
Díky zmíněným schopnostem je možné umělé neuronové sítě použít v mnoha situacích, kde klasicky navržené algoritmy nemusí podávat dostatečně dobré výsledky, nebo v případech, kdy by bylo téměř nemožné požadované algoritmy vytvořit. Běžně jsou tak neuronové sítě užívané v oblasti počítačového vidění (detekce objektů, rekonstrukce a generování obrazu atp.), zpracování přirozené řeči (rozpoznání řeči, generování textů s daným obsahem, překlady cizích jazyků atp.), umělé inteligence (řídicí systémy, hraní her atp.), dolování z dat, v robotice, pro úlohy klasifikace a mnoho dalších.

3.1 Umělý neuron

Umělý neuron [5][38] vznikl snahou o napodobení fungování lidského nervové soustavy, jejíž základem je biologický neuron. Těch se v mozku nachází miliardy a jsou vzájemně různě propojeny.

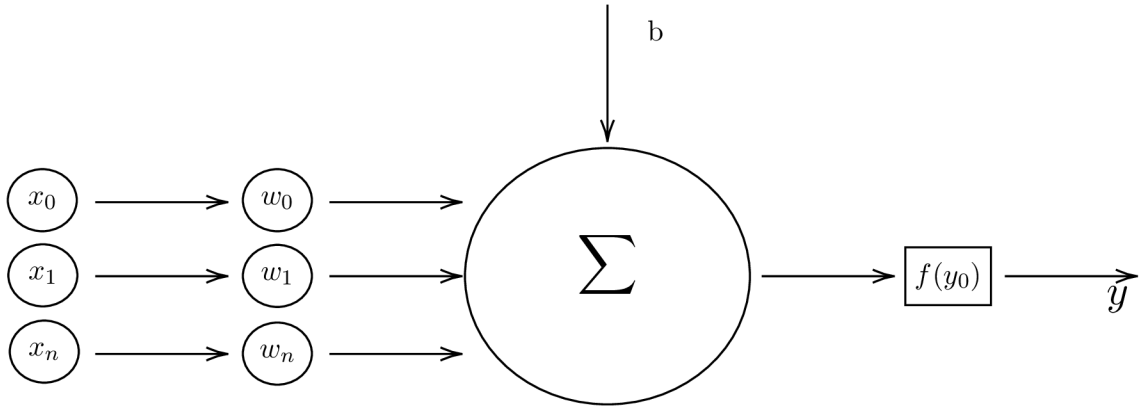
Biologický neuron, resp. jeho zjednodušený model použitý pro výzkum umělých neuronů, a princip jeho fungování lze vidět na obrázku 3.1. Nervové buňky přijímají podněty přes dendritická zakončení. Vstupní informace zpracují a pomocí axonu vyšlou výstupní signál. Axon se větví na množství synaptických zakončení, která mohou být napojena na dendrity dalších neuronů, kde se uvedený proces opakuje.

Umělý neuron je sestaven analogicky ke zjednodušenému modelu biologického neuronu. Následný popis principu jeho funkce a matematické souvislosti v této sekci vychází ze



Obrázek 3.1: Znázornění biologického neuronu.

zdrojů [5] a [38]. Koncept umělého neuronu je zachycen na obrázku 3.2. Vstupní signály mají přiřazeny váhové koeficienty. Jejich multiplikace se poté v těle sečtou i s biasem (vysvětlen dále v kapitole) a na získanou hodnotu se následně aplikuje aktivační funkce (popsána dále v kapitole). Výsledný údaj putuje přes axon do dalších buněk.



Obrázek 3.2: Koncept umělého neuronu.

Matematicky popisuje funkci neuronu rovnice:

$$y = f\Big(\sum_{i=0}^{n} x_i w_i + b\Big)$$

kde x_i značí hodnoty na příslušných vstupech neuronu a w_i příslušné parametry váhy. Označení `b` znamená bias a $f(y_0)$ představuje aktivační funkci, která je aplikována na sumu v neuronu. Finálním výstupem je pak hodnota `y`.

Bias je speciální vstup, který do neuronu přichází navíc (mimo vstupní signály). Někdy je označován jako prahová hodnota, jelikož může určovat hranici, kterou musí suma dosáhnout, aby neuron vyslal signál. Přísluší mu konstantní hodnota, obvykle nastavená na hodnotu 1, jeho váha se však může měnit. Přítomnost biasu není nutnou podmínkou funkčnosti neuronu, ale pro mnoho typů úloh je nezbytný.

Umělý neuron dokáže na základě učení pozměnit váhy vstupů, které pak umocňují, resp. utlumují, vliv jednoho neuronu na druhý. Neuron tedy díky schopnosti vysílat sig-

nál (hodnoty blízké 1), resp. ho nevysílat (hodnoty blízké 0), může fungovat jako binární klasifikátor.

Aktivační funkce jsou aplikovány na výstup z těla neuronu. Mohou být jak nelineární, tak lineární – ty se ovšem obvykle nepoužívají. Pokud by totiž v neuronové síti byly pouze neurony s lineárními funkcemi, byl by její výstup také lineární a celá síť by tak mohla být nahrazena jedním či několika neurony – dále tedy nebudou uvažovány. Pro možnost učení sítě je potřeba, aby funkce byla diferencovatelná (na celém definičním oboru), tedy aby v každém bodě bylo možné aproximovat směr jejího růstu, tzv. gradient.

Mezi nejvíce používané varianty patří **sigmoidální funkce** a **ReLU** (Rectified Linear Unit). První zmíněná se používala nejvíce v minulosti, a to díky své schopnosti převést výstupní hodnoty na čísla mezi 0 a 1. Matematicky je popsána rovnicí

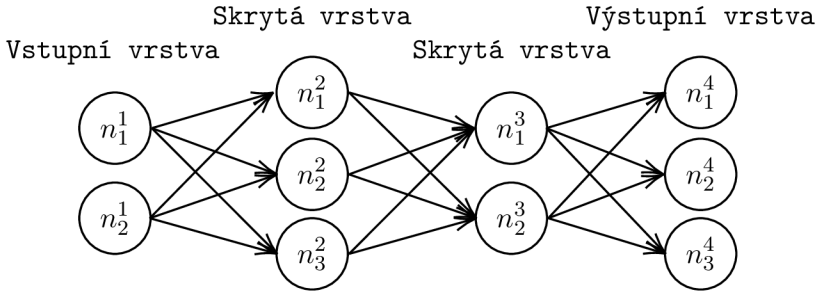
$$\sigma(x) = \frac{1}{1 + e^{-}x}$$

kde x značí výstupní hodnotu z těla neuronu:

V současnosti je asi nejpoužívanější aktivační funkcí ReLU, jelikož bylo ukázáno, že podstatně urychluje trénování konvolučních neuronových sítí (viz 3.4) [3]. Vyskytuje se proto v naprosté většině moderních sítí pro zpracování obrazu a signálů. Matematicky ji popisuje rovnice $f(x) = max(0, x)$. Funkce prahuje výstupní hodnotu `x` těla neuronu na nule (nižší hodnoty nepropouští, vyšší ano).

3.2 Architektura umělých neuronových sítí

Neuronové sítě se skládají z umělých neuronů, které jsou poskládány do jednotlivých vrstev. Sítě obvykle obsahuje vstupní, skryté a výstupní vrstvy – znázorněno na obrázku 3.3. Vstupní vrstva přijímá data ke zpracování. Její výstupy jsou napojeny na neurony skryté vrstvy, která může být napojena na další skrytou vrstvu atp., přičemž poslední z nich předává informace na výstupní vrstvu. Z axonů jejích neuronů se poté vyčtou finální výstupní hodnoty pro reakci sítě na daný vstup. Výstupní neurony také obvykle nemívají přiřazenou aktivační funkci, jejich hodnoty jsou přímo brány jako např. pravděpodobnost příslušnosti do dané třídy při klasifikaci.



Obrázek 3.3: Základní schéma plně propojených vrstev dopředné neuronové sítě.

Podle struktury lze neuronové sítě rozdělit do dvou skupin [5][38]:

- **Sítě s dopředným šířením signálu** – mají neurony propojeny v acyklickém grafu. Cykly se nesmí vyskytovat, jelikož by vytvářely nekonečnou odezvu. Neurony jsou uspořádány do vrstev (viz obrázek 3.3), které jsou navzájem propojeny a dopředně se jimi šíří odezva na vstupní signál. Charakteristické je použití plně propojených vrstev.

- **Sítě se zpětnou vazbou** – také nazývány rekurentní neuronové, liší se přítomností cyklů v propojení neuronů. Výstupy z jedné vrstvy mohou být napojeny zpět na její vstup, což umožňuje uchování předchozí informace. Takto vytvořená paměť pomáhá udržet souvislosti a používá se při generování obrazu, hudby a dalších úlohách. Princip a ukázku využití tzv. *Long Short-Term Memory* sítě (typ sítě se zpětnou vazbou) pro generování ručně psaného textu lze pro zajímavost vidět v článku [2] (s. 1–6 a 24–34). Rekurentní neuronové sítě však nejsou vhodné pro detekci osob, a tak dále nebudou uvažovány.

Plně propojená vrstva [5] je základní vrstvou neuronových sítí. Na vstupu přijímá n-dimenzionální vektor hodnot, z nichž každá je napojena na všechny neurony ve vrstvě. Mezi neurony jedné vrstvy a napříč různými vrstvami žádná další propojení neexistují (propojeny jsou pouze vrstvy po sobě následující), jak lze vidět na obrázku 3.3. Počet synapsí, tedy trénovatelných parametrů sítě, je pak v jedné vrstvě (při použití bias vstupu) určen dle rovnice $N = n(i + 1)$, kde `N` je výsledný počet propojení souvisejících danou vrstvou, `n` značí počet jejích neuronů a `i` udává počet hodnot na jejím vstupu.

Jelikož je každý váhový parametr trénovatelný, roste s jejich počtem i potenciální množina znalostí, které je síť schopna pojmout. Obecně se proto doporučuje volit velikost sítě úměrně náročnosti řešeného problému. Pokud je příliš malá, nemusí jít vhodně natrénovat a v dané úloze selže. Pokud je síť naopak příliš velká, může dojít k přeučení na trénovacích vstupech, pročež síť bude reagovat na nepodstatné detaily konkrétních dat a nebude schopna správně určit neznámé vstupy. Trénování sítě se dále věnuje kap. 3.3.

3.3 Učení neuronových sítí

Funkce neuronových sítí spočívá v tom, že na vstup je přiveden vzorek dat, pro který je očekáván odpovídající výstup sítě. Nově vytvořená síť však podává výsledky víceméně náhodné a je potřeba ji natrénovat, nebo-li učit. Jak bylo popsáno, neuronové sítě obsahují váhové parametry neuronových propojení. Při procesu učení se právě jejich hodnoty upravují tak, aby bylo dosaženo cílených výstupů.

Existují tři základní typy učení neuronových sítí [22]:

- **Učení s učitelem** („supervised learning“) – trénovací množina sestává z dvojic vstupních hodnot a jim odpovídajících očekávaných výstupů. Pro každý vzorek síť vygeneruje výsledek, který se následně porovná s očekávaným výstupem, a na základě rozdílu se upraví váhové parametry. Vzorky bývají vybírány náhodně a celý proces se opakuje v cyklech, dokud síť nepodává uspokojivou odezvu. Pro naučení sítě je tedy potřeba dataset s předem připravenými známými výsledky. Tato metoda je obvykle používána pro trénování sítí k detekci osob, bude tedy rozebrána podrobněji dále v kapitole.

- **Učení bez učitele** („unsupervised learning“) – trénovací množinu tvoří pouze vstupní vzorky. Metoda se snaží docílit stejných výstupů pro jednotlivé skupiny vstupů s podobnými vlastnostmi. Pro určení podobnosti vzorků může být použita např. euklidovská vzdálenost výstupních vektorů. Na základě míry shodnosti se pak upravují váhy sítě a celý proces se opět opakuje v epochách.

- **Zpětnovazební učení** („reinforcement learning“) – trénovací množinu tvoří pouze vstupní vzorky. Síť dokáže určovat, zda je daný výstup dobrý, nebo špatný, pomocí hodnotící funkce, která určuje velikost odměny. Parametry vah se upravují tak, aby výsledná odměna byla co největší.

3.3.1 Učení s učitelem

Jak bylo popsáno, při učení sítí s dopředným šířením signálu s učitelem jsou odezvy vygenerované pro vstupní vzorky porovnány s jejich očekávanými výstupy, na základě čehož se potom upravují váhy sítě. K určení velikosti rozdílu mezi výstupy se používají **chybové funkce** (loss funkce) [5], např. lze počítat střední kvadratickou odchylku. Pomocí loss funkce lze získat výslednou chybu sítě dle rovnice:

$$L = \frac{1}{N} \sum_{i=0}^{N} L_i$$

kde `L` značí velikost chyby a `N` je počet dvojic (vstup a očekávaný výstup) v trénovacím datasetu. Cílem učení je právě tuto chybu minimalizovat, k čemuž se používá zpětná propagace chyby, tzv. **backpropagation** [38].

Jedná se o metodu, která počítá parciální derivace loss funkce ve vztahu k jednotlivým váhovým parametrům sítě. Na jejich základě jsou následně váhy upraveny. Velikost změn je však ještě ovlivněna **učícím koeficientem** (learning rate) [5]. Ten nabývá hodnot v rozmezí 0 až 1, kde 0 by znamenala, že neproběhnou úpravy žádné. Kvůli zpětné propagaci chyby je nutné, aby chybová funkce i všechny aktivační funkce (jak bylo popsáno v kapitole 3.1) byly diferencovatelné, aby bylo možné určit gradient.

Sítě s vícero skrytými vrstvami jsou často označovány jako **hluboké neuronové sítě**. V praxi se mohou používat i desítky či více vrstev. V takových případech narůstá počet trénovatelných parametrů sítě do extrémních počtů, které se na současných počítačích trénují příliš dlouho (např. dny i týdny). Kvůli tomu se používají různé metody k optimalizaci učení, jako je např. **stochastic gradient descent**, jehož princip lze najít v [5].

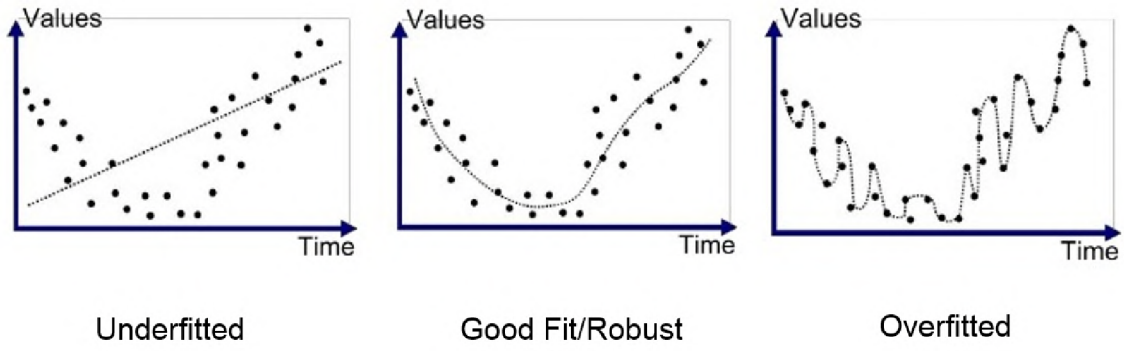
Dalšími přístupy jsou tzv. offline učení či **batch training**. Jako online učení je označován přístup, kdy se pro jednotlivé vzorky datasetu vygenerují výstupy a váhy sítě jsou po každém z nich upraveny. V případě offline, resp. batch učení, jsou odezvy sítě spočítány pro celý dataset, resp. jeho stanovenou část. Parametry vah se pak zpětně propagují jednorázově až na základě chyb celé množiny vzorků. Více na toto téma, včetně porovnávání výsledků a vhodnosti těchto metod, lze najít v článku [15].

3.3.2 Přeučení a nedoučení neuronových sítí

Dobře natrénovaná neuronová síť by měla nejen podávat správné výsledky pro naučené vstupy, ale především být schopna **generalizovat**, tedy zvládat správně vyhodnotit i data, na kterých nebyla naučená. Proto je důležité vybrat trénovací vzorky tak, aby umožnily dostatečně rozeznat všechny vlastnosti klíčové pro danou úlohu.

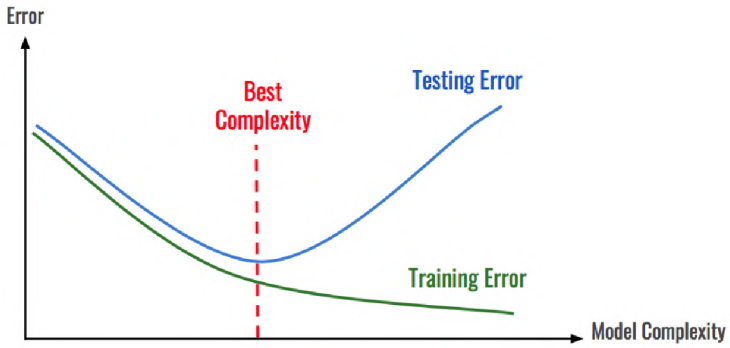
Na příkladu lze uvést např. síť, která rozhoduje, zda se jedná o jablko, či nikoli. V jejím trénovacím datasetu by měla být nejenom nejrůznější jablka, ale také objekty, které jimi nejsou, přestože jsou jim podobné - např. hrušky. To aby síť dokázala nalézt hranici mezi jablky a jinými předměty. Pokud by data obsahovala pouze jablka, síť by na datasetu podávala správné výsledky, ale na jiných příkladech by s nejvyšší pravděpodobností naprosto selhala.

Aby bylo síť možné správně naučit, je také potřeba velké množství dat. Čím více vzorků, tím lépe se naučí rozlišovat jejich různé vlastnosti. K rozšiřování datasetu se v praxi používá **augmentace dat** – její popis a aplikaci rozličných postupů lze najít v článku [8]). Pokud dat není dostatek, často to vede k přeučení sítě na dané množině dat a její neschopnosti generalizovat.



Obrázek 3.4: Zleva jdoucí ukázky výstupů sítě nedoučené, správně natrénované a přeučené [6].

Přeučení sítě (overfitting) [6] je situace, kdy se síť naučí klasifikovat vzorky z trénovacího datasetu tak přesně, že ztrácí schopnost generalizace, tudíž se snižuje její přesnost pro jiné než trénovací vstupy. Opakem je potom **nedoučení sítě** (underfitting) [6], kdy síť klasifikuje příliš obecně a nepřesně. Zmíněné situace lze pozorovat na obrázku 3.4, kde se vlevo nachází ukázka výstupů nedoučené sítě, uprostřed graf odezvy vhodně natrénované sítě a vpravo výsledky přeučené sítě. Vertikální osa představuje hodnoty vstupních vzorků, horizontální osa čas průběhu.



Obrázek 3.5: Graf chyby sítě na testovací a trénovací množině a určení bodu přeučení [27].

Hranici přeučení sítě lze přibližně detekovat tak, že je celý dataset rozdělen do dvou částí – trénovací a testovací data. Na trénovacích datech se síť naučí a na testovacích vzorcích se pak ověří její přesnost. Zatímco chyba na trénovací množině bude postupně neustále klesat, nepřesnost na testovacích datech bude klesat jen do určitého okamžiku, načež začne růst. V okolí bodu tohoto zlomu se pohybuje hranice, kdy je síť optimálně natrénovaná, což lze vidět na obrázku 3.5. Vertikální osa udává chybu sítě, horizontální osa počet trénovacích cyklů. Křivka „Training error“ znázorňuje nepřesnost v průběhu učení na trénovacím datasetu, „Testing error“ pak na testovací množině. Označení „Best Complexity“ popisuje zmíněný zlomový bod pro přeučení.

Přeučení sítě lze také předcházet dalšími postupy, jako je např. metoda **dropout**. Při ní jsou ze sítě odebrány některé neurony, které jsou následně navráceny po procesu trénování. Více o této metodě lze najít v článku [27].

3.4 Konvoluční neuronové sítě

Konvoluční neuronové sítě [5] jsou variantou umělých neuronových sítí. Jejich architektura je uzpůsobena zpracování obrazu, ve kterém dokáží např. detekovat objekty mnohem spolehlivěji, než tradiční algoritmy. Dále jsou pro ně charakteristickými úlohami problémy převeditelné do struktury podobné obrazové podobě a zpracování zvuku.

Konvoluční neuronové sítě se začaly používat v průběhu 80. let, a to především pro rozpoznávání ručně psaných číslic [30]. Jejich širší rozvoj však byl zpomalen především nedostatečnou výpočetní kapacitou. Zlomový okamžik pro jejich rozšíření pak představoval rok 2012, kdy v soutěži ImageNet Large Scale Visual Recognition Challenge[1] zvítězila konvoluční neuronová síť vytvořená Alexandrem Krizhevskym [3]. Ta navíc šokovala svojí chybovostí 15.3 % a absolutní převahou nad tradičními algoritmy, z nichž nejlepší měl na druhém místě chybovost 26.2 %. Výsledky konvolučních neuronových sítí v dalším ročníku soutěže dosáhly přibližně 6,8 %, čímž se přiblížily úrovni nepřesnosti 5,1 %, které dosáhlo lidské pozorování. Sítě v dalších ročnících tuto hranici dokonce překonaly, a to s chybovostí kolem 3 % [48]. Výsledky soutěže jsou podrobněji srovnávány v článku [39].

V návaznosti na dobré výsledky konvolučních neuronových sítí se začaly objevovat nástroje usnadňující jejich tvorbu a učení, např. TensorFlow[2] či Keras[3]. Dále se začaly vyvíjet metody urychlení na stávajících počítačích, např. použitím grafickým karet, které mají velké množství výpočetních jader. Začal se také vyvíjet specializovaný hardware určený pro trénování sítí, např. neural compute stick od Intelu[4]. To vše usnadňuje širší využití konvolučních neuronových sítí v praxi a jejich další rozvoj.

3.4.1 Architektura konvolučních neuronových sítí

Pokud není uvedeno jinak, informace v této kapitole jsou čerpány ze zdroje [5], kde lze případně dohledat podrobnější vysvětlení užitých pojmů. Konvoluční neuronové sítě mají architekturu uzpůsobenou pro zpracování obrazu. Klasické neuronové sítě nejsou pro tyto úlohy vhodné, jelikož by měly příliš velký počet trénovatelných parametrů – např. pro vstupní obrázek o výšce i šířce 200 pixelů a hloubce 3 barev by bylo třeba 120 000 spojení v jediné vrstvě. U hlubokých neuronových sítí či při použití obrázků s větším rozlišením by tedy učení bylo extrémně výpočetně náročné a neefektivní. Dále by se také objevovaly tendence k přeučení na nepodstatných detailech obrazu a k nedostatečné generalizaci.

Konvoluční neuronové sítě ke zpracování obrazu využívají především **konvolučních vrstev**, ve kterých jsou neurony uspořádány do 3D struktury vrstev podobně jako vstupní obraz (výška, šířka a hloubka neboli počet kanálů). Těmto vrstvám se také říká konvoluční filtry. Sítě tak mají významně nižší počet trénovatelných parametrů. Schéma lze vidět na obrázku 3.6.

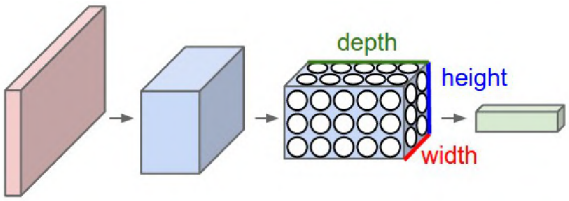
Pro každý vstup proběhne postupně konvoluce všech jeho kanálů s filtry dané vrstvy. Výsledkem jsou tedy 2D odezvy jednotlivých konvolučních filtrů seřazeny za sebou do 3D výstupu. V průběhu učení jsou parametry filtrů upravovány a jejich hodnoty lze zobrazit.

[1] Soutěž probíhala každoročně v letech 2010–2017, viz https://image-net.org/challenges/LSVRC. Věnovala se porovnávání algoritmů detekce a klasifikace objektů v obraze. Stala se asi nejznámějším srovnávacím kritériem a významnou platformou pro rozvoj těchto algoritmů. Poté záštitu převzal Kaggle: www.kaggle.com.

[2] Framework pro práci s neuronovými sítěmi. Dostupný z: www.tensorflow.org.

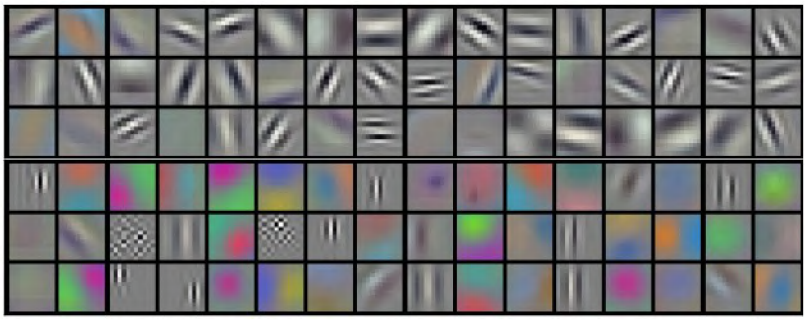
[3] Vysokoúrovňové API pro práci s neuronovými sítěmi. Dostupné z: https://keras.io.

[4] Výpočetní zařízení velikosti USB flash disku, specializované svým výkonem pro trénování neuronových sítí. Více informací na stránkách produktu: https://software.intel.com/en-us/neural-compute-stick.



Obrázek 3.6: Schéma konvoluční vrstvy [5].

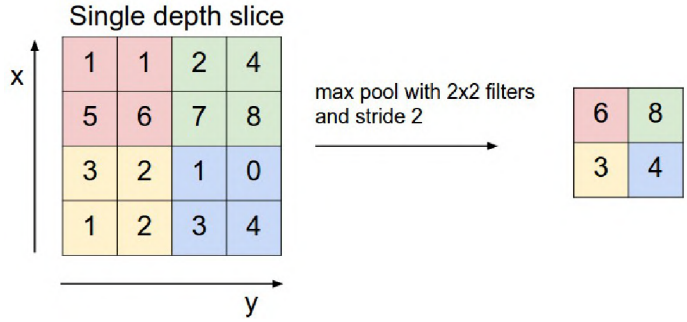
Takovéto jádra konvolučních filtrů lze vidět na obrázku 3.7, kde jsou v horní části patrné jádra reagující na hrany, zatímco ve spodní části spíše na barvy.



Obrázek 3.7: Ukázka naučených jader konvolučních filtrů [5].

Počítáním konvoluce se však podle velikosti filtru snižuje také velikost výstupu oproti vstupním rozměrům. Tomuto se dá zabránit tzv. **zero paddingem**, kdy se doplní okraje vstupního obrázku o nulové pixely, takže výsledek nabývá stejných rozměrů jako vstup.

Pooling vrstva se často vkládá za konvoluční vrstvy jednak aby se snížil počet dimenzí a tím i výpočetní náročnost, jednak také ovlivňují lepší zobecňování sítě a zabraňují přetrénování. Vrstva rozděluje vstupní matici hodnot na čtvercové části, které se nesmí překrývat. Poté z každé oblasti vybere či dopočítá jedinou hodnotu, která poté oblast reprezentuje na výstupu vrstvy. Ke zvolení hodnoty z čtvercové submatice se často používá funkce výběru maxima. Princip takové pooling vrstvy lze vidět na obrázku 3.8.



Obrázek 3.8: Princip fungování pooling vrstvy [5].

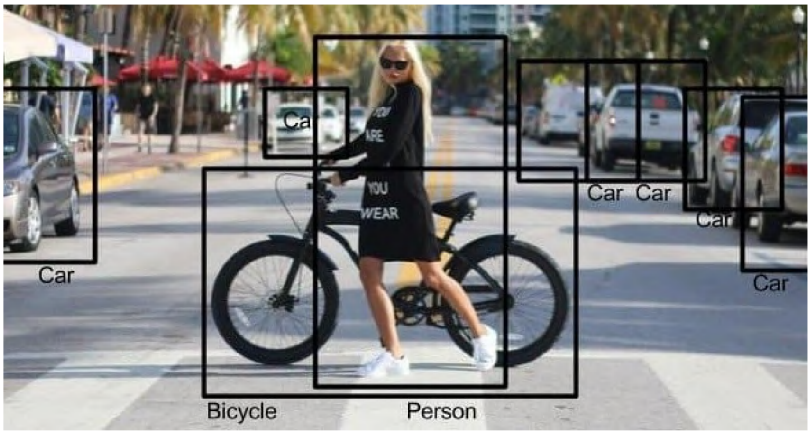
Opakováním konvolučních a poolingových vrstev dokáže síť extrahovat příznaky z obrazu. Obvykle lze po naučení sítí pozorovat, že jádra první vrstvy odpovídají nízkoúrovňovým rysům (např. hrany), zatímco filtry následujících vrstev postupně detekují příznaky

vyšších úrovní. Výsledná data jsou poté přivedena do plně propojených vrstev, které fungují jako klasifikátor na základě získaných příznaků.

3.5 Detekce osob pomocí konvolučních neuronových sítí

Tato kapitola se zabývá rozpoznáváním osob v obraze, resp. v jednotlivých snímcích videozáznamu, pomocí konvolučních neuronových sítí. Jak bylo popsáno v kapitole 3.4, jejich využití v praxi se neustále rozšiřuje a přesnost detekce zlepšuje.

Pro detekci chodců jsou používány neuronové sítě určené k detekci nejrůznějších objektů obecně. Ty nalézají ohraničení objektu v obrázku, tzv. bounding box, a zároveň určují, o jaký objekt se jedná. Ukázku tohoto přístupu lze vidět na obrázku 3.9.



Obrázek 3.9: Ukázka detekovaných objektů v obrázku a jejich bounding boxů [29].

Jednou z takových sítí je např. SSD (*Single Shot MultiBox Detector*) [50]. Jedná se o velmi přesnou síť, která využívá k predikci nejen data z poslední výstupní vrstvy, ale také přímo na závěr vyhodnocuje i některé údaje z dřívějších vrstev. Ukázku jeho výstupu lze vidět na obrázku 3.9. Další oblíbenou sítí je např. YOLO (*You Only Look Once*) [26], která vyniká nejen přesností, ale především rychlostí – na výkonných grafických kartách je schopná zpracovat i 30 snímků za sekundu (v závislosti na rozlišení obrazu).

V současnosti existuje nespočet variant různých sítí pro detekci osob, které jsou trénované na různých datasetech. Přehled výsledků různých algoritmů např. na datasetu Caltech[5] lze najít na webové stránce Papers With Code[6].

3.5.1 Detekce osob z výšky

Detekce osob při pohledu z výšky pomocí neuronových sítí je problematická v případě, kdy byla síť naučena rozpoznávat postavy na datasetu obsahujícím převážně snímky pořízené z podobné výškové úrovně a vzdálenosti, ve které se vyskytují detekovaní lidé. Většinu populárních komerčních nástrojů pro detekci osob tak nelze spolehlivě použít - např. řešení z nejznámější soutěže pro detekci objektů ImageNet (podrobněji zmíněno v kap. 3.4) se soustředí na snímky natáčené při pohledu z auta atp. Jejich výsledky jsou nedostačující při úhlech pohledu z výšky.

[5] Databáze videozáznamů chodců snímaných z jedoucího vozidla s nízkým rozlišením kamery. Dostupná z: www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/.

[6] Dostupný na adrese: https://paperswithcode.com/.

Tato problematika lze v některých případech obejít pomocí existujících řešení, která se nezaměřují na detekci celé postavy či obličeje, ale na jednotlivé části těla, např. kolena, prsty či nos. V tomto případě by tedy mohly být např. hlava a končetiny úspěšně detekované i při pohledu z výšky, při kterém by za normálních okolností jiná síť detekující celé postavy selhala.

Touto problematikou se zabývá např. systém pro detekci částí lidského těla OpenPose[7]. Rozlišuje až 135 podčástí osoby, ze kterých lze určovat její pozici. Ukázka výsledků tohoto systému je vidět na obrázku 3.10. Ačkoli dokáže rozpoznat osoby z nejrůznějších úhlů (stačí i pouze jejich části), jeho nevýhodou je, že přesnou detekci zvládá jen pro objekty vzdálené jednotky až nižší desítky metrů (také v závislosti na rozlišení snímku). Při vyšších vzdálenostech rozpozná pouze malý počet bodů, které nestačí k zformování siluety člověka.

Obrázek 3.10: Ukázka výstupu OpenPose [49].

K tomuto účelu by tedy bylo ideální vytvořit či najít vhodný dataset, který poskytuje záběry osob z výšky a při nejrůznějších pozorovacích úhlech a vzdálenostech. Na něm poté natrénovat neuronovou síť, která bude schopná osoby z těchto nejrůznějších pohledů detekovat.

Jeden z datasetů splňujících zmíněné podmínky je např. *MultiDrone Public DataSet*[8]. Obsahuje záběry chodců i cyklistů, dále nabízí pohledy na jednotlivé osoby i skupinky nebo zástupy lidí – např. záběry z ulice či běžeckého závodu. Vše je nasnímáno z kamery dronu, který se pohybuje v různých výškách – od úrovně lidské postavy až po několik desítek metrů na povrchem.

Dalším datasetem obsahujícím záběry chodců z výšky je *VisDrone2018*[9] Ten ovšem obsahuje výhradně záběry z velkých výšek (desítky metrů), takže sítě naučené na tomto datasetu by měly problémy s detekcí osob při bližším pohledu. Podobné záběry nabízí také Stanford Drone Dataset[10]. Více vhodných databází lze dohledat např. na seznamu datasetů pro monitorování osob – CVonline[11].

[7] Dostupný z: https://github.com/CMU-Perceptual-Computing-Lab/openpose.
[8] Dostupný z: https://multidrone.eu/multidrone-public-dataset/.
[9] Dostupný z: https://aiskyeye.com/.
[10] Dostupný z: https://cvgl.stanford.edu/projects/uav_data/.
[11] Dostupný z: https://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm#surveillance.

Kapitola 4

Určení pozice a sledování osob

Tato kapitola rozebírá určení pozice detekovaného člověka ve videozáznamu vzhledem k pozici dronu. Navržené metody vycházejí z předpokladu, že osoba byla v obraze úspěšně detekována a potřebné údaje (např. výška letu či údaje o natočení dronu a úhlu záběru) jsou získatelné z dříve popsaných senzorů přístroje. Tyto informace pak budou použity k určení přibližných GPS souřadnic osoby.

Dále se kapitola věnuje sledování osob pomocí re-identifikace detekovaných objektů. Díky tomu lze přiřadit získané GPS souřadnice k jednotlivým identitám a v mapě potom zaznamenat trajektorie konkrétních osob.

4.1 Určení pozice osob

Videozáznam je série snímků. Pokud se podaří určit pozici osoby v jednotlivých záběrech, je snadné tyto údaje propojit s časovou sekvencí videozáznamu. Detekce osoby probíhá také vždy jen na jednotlivých snímcích videa. Dále uvedené metody budou tedy přistupovat k videozáznamu jako k sérii fotek a pro zjednodušení budou navrhovat, jak určit pozici osoby detekované v obraze, ne v celém videozáznamu.

Pro určení reálné pozice osoby z videozáznamu je nejprve nutné osobu úspěšně detekovat, čehož lze dosáhnout s využitím metod popsaných v kapitole 3. Záleží přitom na přesnosti detekce, jelikož i malý rozdíl několika pixelů v obraze může znamenat významnou odchylku zjištěné pozice u subjektů vzdálených od dronu.

Dále je nutné lokalizovat dron v prostoru (vertikální i horizontální souřadnice), znát jeho natočení a úhel připevněné kamery vůči němu. Z těchto údajů je možné následně zjistit směr pohledu kamery v prostoru. Tyto informace lze získat pomocí systémů bezpilotních letounů popsaných v kapitole 2.2. Jak ovšem bylo zmíněno, údaje mohou někdy být poměrně nepřesné. To může značně zkreslit výslednou pozici subjektu.

Potřebným údajem je také zorné pole kamery, které vypovídá o tom, jak velká výseč reálného okolí je zachycena na snímku. Od toho se pak odvíjí určování 3D směrového vektoru, který probíhá od dronu k osobě.

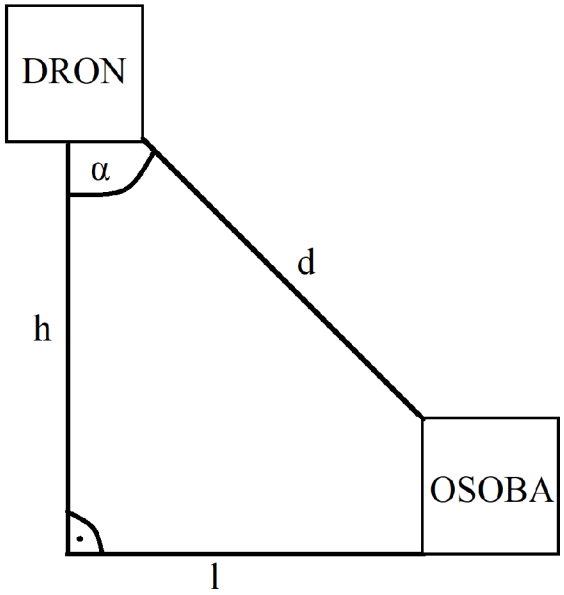
Následně je potřeba zjistit vzdálenost sledovaného subjektu od dronu (přímou či horizontální, dle metody). Právě oním problémem se zabývají metody navržené v této kapitole. Nakonec lze s využitím všech zmíněných informací vypočítat reálnou pozici osoby z obrazu.

Několik základních metod, kterými lze vzdálenost osoby od dronu určit, je popsáno dále v této kapitole. K jejich přesnějšímu vysvětlení je použit obrázek 4.1. Čtverec s označením

`DRON` symbolizuje bezpilotní letoun, který sleduje objekt označený jako `OSOBA`. Trojúhelník, který je spojuje, charakterizují následující body:

- Strana `h` – značí výšku dronu (vertikální vzdálenost od povrchu).
- Strana `l` – symbolizuje horizontální vzdálenost mezi dronem a objektem. Svírá pravý úhel se stranou `h`.
- Strana `d` – určuje přímou vzdálenost dronu a objektu.
- Úhel `α`– popisuje natočení dronu (vůči kolmici k zemskému povrchu).

Obrázek 4.1 symbolizuje průřez 3D prostorem. Je zanedbána hloubka scény, avšak princip určení vzdálenosti subjektu je po zohlednění natočení dronu v prostoru identický.



Obrázek 4.1: Výchozí údaje pro určení pozice sledovaného objektu.

4.1.1 Vzdálenost na základě předpokládané velikosti objektu

Metoda umožňuje určit vzdálenost osoby od dronu, pokud je známa velikost sledovaného objektu a ohnisková vzdálenost použité kamery. Velikost sledované osoby lze obecně odhadnout z populačních statistik. Není také třeba určovat velikost celé osoby, může stačit pouze část těla, pokud např. na snímku zbytek postavy chybí.

Ohniskovou vzdálenost kamery lze dopočítat experimentálně. Je třeba umístit předmět o známé velikosti před záznamový aparát a změřit reálnou vzdálenost od objektivu. Následně lze počítačově určit ohniskovou vzdálenost kamery na základě porovnání reálné velikosti objektu a jeho velikosti v záznamu.

Je tedy možné odhadnout, jak velký by měl být detekovaný člověk. Trojčlenkou lze následovně dopočítat, jak daleko od objektivu by musel stát, aby jeho reálná velikost odpovídala velikosti v obraze zachyceném danou kamerou. Výsledkem je pak právě vzdálenost `d` (viz obrázek 4.1).

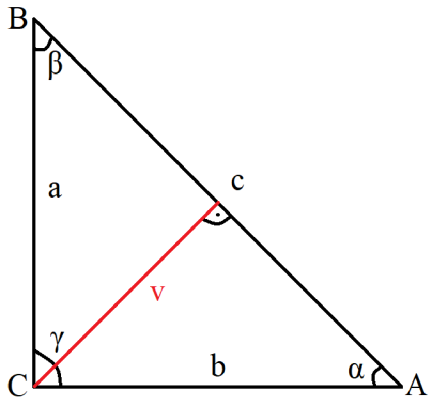
Nevýhodou této metody je však nepřesnost plynoucí z obecného odhadu velikosti, jelikož každý člověk je jinak vysoký a má jiné tělesné proporce. Taková chyba může být zanedbatelná při malých vzdálenostech a pro část populace odpovídající odhadu. Avšak může vést k podstatným zkreslením pozice při sledováním na větší vzdálenost, nebo pokud by cílem bylo např. dítě.

4.1.2 Triangulace

Metoda triangulace je obecně známý postup užívaný k vypočítání souřadnic objektů zaznamenaných ze dvou kamer současně. Princip je zachycen na obrázku 4.2. Jsou zapotřebí 2 body `A` a `B`, které společně s cílovým objektem `C` tvoří vrcholy trojúhelníku. Pomyslná výška `v` je pak vzdálenost, kterou lze zjistit. Vnitřní úhly α a β jsou známé, doplněním do součtu 180° se dopočítá i úhel γ. Sinovou větou lze zjistit délky stran. Vzdálenost `v` pak lze určit dle vzorce 4.1:

$$v = a \times \tan \beta \tag{4.1}$$

V kontextu záznamu z dronu představují body `A` a `B` dvě kamery, jejichž úhel upevnění je předem změřený. Nevýhodou je tedy potřeba použití dvou kamer místo jediné, což některé konvenční drony nemusí umožňovat, jak bylo popsáno v kap. 2.1.2. Dalším záporem je míra nepřesnosti při použití kamer svírajících velmi malý úhel (dron nenabízí příliš místa), kdy může triangulace dosahovat velké chyby, nebo i zcela selhat – např. pokud nejsou osoby detekované dostatečně přesně, strany `a` a `b` z obrázku 4.2 se nemusí protnout a bod `C` vůbec nevznikne.



Obrázek 4.2: Ukázka triangulace.

4.1.3 Rekonstrukce 3D scény

Rekonstrukce 3D scény [13] může být prováděna při použití více kamer snímajících prostředí z různých úhlů nebo při použití jediné kamery, která dynamicky mění svůj úhel pohledu v čase. Nevýhodou prvního zmíněného způsobu je jednak malá vzdálenost pozorovacích bodů, jednak potřeba dvou kamer, což nemusí dron umožňovat (viz kap. 2.1.2).

Hlavní nevýhodou obou způsobů, především pak druhého uvedeného, je náchylnost na proměnlivost scény. Pokud totiž objekty ve scéně mění umístění, zatímco jsou snímány, dojde k nepřesnostem a zkreslení výsledku v závislosti na míře a rychlosti obměn. Obzvláště v případě, kdy je třeba sledovat pohybujícího se člověka, by tedy mohly být výsledky velmi

nepřesné. Další riziko přináší pohyb kamery, který může snižovat kvalitu záznamu (rozmazaný obraz), a tím znesnadnit rekonstrukci scény. Při použití ve statickém prostředí by však tato metoda mohla dosahovat velmi dobrých výsledků.

4.1.4 Určení vzdálenosti objektu na rovném povrchu

Metoda vychází z předpokladu, že nadmořská výška v místě vertikálně pod dronem a nadmořská výška bodu, na kterém stojí sledovaná osoba, jsou shodné (viz obrázek 4.1). Pokud je známa orientace dronu (`α`) i výška nad zemí (`h`), lze dopočítat horizontální vzdálenost (`l`) pomocí goniometrické funkce tangens. Konkrétně dosazením do vzorce

$$l = h \times \tan \alpha.$$

V tomto případě není ani nutné dopočítávat vzdálenost `d`, jelikož díky předpokladu rovného povrchu lze určit horizontální pozici osoby přičtením získané hodnoty `l` k horizontálním GPS souřadnicím dronu. Vertikální složku pozice subjektu pak lze získat odečtením výšky `h` od vertikální souřadnice letounu.

Nevýhodou této metody je ovšem onen předpoklad rovného zemského povrchu. Pokud by se sledovaný člověk nacházel na vyvýšeném, resp. sníženém místě, vertikální souřadnice subjektu by byla samozřejmě nepřesná. Avšak také jeho horizontální pozice by kvůli tomu byla určena dále, resp. blíže ke dronu. Při natáčení osob nad nebo v úrovni dronu by pak metoda udávala nekonečně velkou vzdálenost osoby, takovýto záznam by tedy neposkytl žádné výsledky.

4.1.5 Určení pozice objektu s využitím nadmořské výšky

Metoda je svým přístupem podobná přístupu navrženému v kap. 4.1.4 – snaží se dopočítat vzdálenost `l` z obrázku 4.1. Vychází ovšem z předpokladu, že sledovaná osoba se pohybuje po zemském povrchu, jehož nadmořskou výšku známe.

Zemský povrch byl z velké části zmapován např. v roce 2000 organizací NASA (*United States National Aeronautics and Space Administration*) v projektu SRTM (*Shuttle Radar Topography Mission*)[1]. Ten například poskytuje veřejně dostupné informace o nadmořské výšce v bodech s horizontálními rozestupy 90 m a pokrytím planety z 80 %. Od roku 2009 pak Japonsko pomocí zařízení ASTER (*Advanced Spaceborne Thermal Emission and Reflection Radiometer*) umístěným na satelitu Terra zmapovalo až 99 % zemského povrchu a společně s NASA zpracovalo data o nadmořské výšce s horizontálním rozlišením 30 m. Údaje jsou volně dostupné v rámci projektu GDEM (*Global Digital Elevation Model*)[2].

S těmito daty, známým natočením dronu (úhel `α`), jeho pozicí a výškou nad terénem (`h`) je pak možné dopočítat vzdálenost, ve které polopřímka $\overrightarrow{DRON–OSOBA}$ protíná zemský povrch. Díky tomu lze zjistit i horizontální vzdálenost bodu od dronu. Translací GPS pozice je pak možné určit souřadnice, na kterých se pozorovaná osoba nachází.

Jelikož metoda vychází z předpokladu, že se osoba nachází na zemském povrchu, může dojít k mylnému výsledku v případě, kdy se člověk pohybuje po vyvýšeném objektu (např. budova nebo most).

Další nevýhodou je nepřesnost plynoucí z kvality databáze nadmořských výšek. Informace mohou být v některých regionech neúplné, popř. nemusí dosahovat potřebného rozlišení. Např. pokud se sledované osoby pohybují blízko dronu a nebo výšková data popisují

[1] https://www2.jpl.nasa.gov/srtm
[2] https://asterweb.jpl.nasa.gov/gdem.asp

geograficky členitou oblast pouze s mřížkou v řádech stovek metrů, kvalita výsledků bude pravděpodobně nízká.

4.2 Re-identifikace a sledování osob ve videozáznamu

Tato kapitola se věnuje sledování osob a řeší, jak jedince vzájemně rozlišit a identifikovat ve videozáznamu. K tomu je nutné vymezit několik pojmů:

- `Identita` člověka je unikátní soubor rysů, které ho odlišují od ostatních. Jejími základními pilíři jsou tyto 3 body: fyzické rysy (např. obličej), znalosti (např. heslo, styl vyjadřování) a vlastnictví (např. řidičský průkaz).
- `Verifikace` znamená porovnání identit jedna ku jedné pro potvrzení (stejná osoba) či vyvrácení shody.
- `Identifikace` je proces porovnávání dané identity s množinou všech známých identit s cílem nalezení/vyvrácení shody.
- `Re-identifikace` značí opětovnou identifikaci. V kontextu této práce tedy jde o snahu sledovat osoby napříč snímky videozáznamu.

V současnosti se sledování osob uplatňuje nejen v komerční sféře (např. detekce podezřelého chování v bankách či mapování pohybových vzorů v obchodních domech za účelem zlepšení služeb a tržeb), ale i ve státní [20]. Výrazná je především potřeba nasazení těchto systémů pro zajištění bezpečnosti (např. na letištích nebo bezpečnostní kamery ve městech).

Lidé mají pro identifikaci ostatních k dispozici mnoho faktorů (např. celkový vzhled, obličej, oblečení, hlas, vyjadřování, kontext). Zdánlivě běžný a snadný úkol se však značně komplikuje při strojové automatizaci. Z videozáznamu lze získat jen limitované informace. Navíc i ty pak mohou být dále omezené – např. nízkým rozlišením obrazu, špatným osvětlením, subjekty daleko od kamer a v různých úhlech natočení atp. V rámci této práce se dokonce může pohybovat i sama kamera, takže proměnlivost prostředí a objektů (potažmo jejich rysů) se ještě zvyšuje.

Kvůli zmíněným okolnostem se tomuto tématu výzkumníci věnují již řadu let. Významnější vývoj započal kolem roku 2003, velký nárůst byl zaznamenán od roku 2008 dále a neutuchá ani v poslední době [20]. Ba naopak s rozvojem neuronových sítí a strojového učení se zájem o odvětví rychle zvyšuje, tento trend lze pozorovat i na obrázku 4.3, který ukazuje rostoucí počet článků souvisejících s tímto tématem na významných konferencích počítačového vidění (CVPR[3], ICCV[4], ECCV[5]).

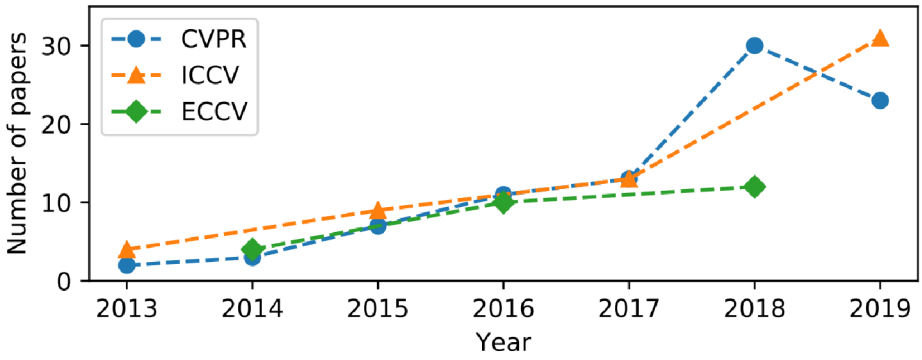
Obecně lze přístupy k re-identifikaci rozčleňovat a hodnotit na základě různých kritérií [42], např. počet sledovaných subjektů, typ použitých rysů, přesnost, výpočetní efektivita atp. Zde je vhodné zmínit především kontextové systémy, které využívají i informace dostupné mimo videozáznam, např. natočení kamery.

Vzhledem k určování GPS souřadnic osob v této práci se tedy přímo nabízí možnost využití jejich pozice pro zpřesnění výsledků identifikace. Např. pokud je ve stejném snímku vícekrát označen stejný člověk, lze porovnat jeho lokaci s předchozím snímkem (s přihlédnutím k časové souvislosti) a určit, která možnost je chybná, resp. pravděpodobnější.

[3] Conference on Computer Vision and Pattern Recognition. Informace o ročníku 2020 dostupné z: https://cvpr2020.thecvf.com.

[4] Internal Conference on Computer Vision. Informace o ročníku 2019 dostupné z: https://iccv2019.thecvf.com.

[5] European Conference on Computer Vision. Informace o ročníku 2020 dostupné z: https://eccv2020.eu.



Obrázek 4.3: Množství článků o re-identifikaci osob na konferencích věnujících se počítačovému vidění v letech 2013 – 2019 [54].

4.2.1 Principy strojové re-identifikace

V této kapitole budou popsány základy počítačové re-identifikace s důrazem na rozdíl mezi klasickými algoritmy a řešeními založenými na neuronových sítích. Informace jsou čerpány převážně z knihy [20].

Na vstupu systémů pro re-identifikaci se předpokládají snímky osoby – obvykle výřez obrazu i s blízkým okolím (např. po detekci pomocí CNN). Poté je provedena extrakce příznaků[6] sledované osoby. Následně je vytvořen vektor těchto rysů. Proces verifikace, potažmo identifikace, pak probíhá jako porovnávání vektorů pro jednotlivé subjekty.

Re-identifikace založená na extrakci rysů klasickými algoritmy. Před nástupem trendu neuronových sítí stavěla identifikace osob především na klasických algoritmech pro extrakci příznaků, které jsou základním stavebním kamenem. Typicky se jedná o barvu, texturu, tvary (detektory rohů, hran atp.) či nejrůznější lokální rysy (např. LBP, LRP). V ideálním případě by se neměly příliš měnit při změnách osvětlení, v různých pozorovacích úhlech atp. Ovšem takové vlastnosti obecně nelze garantovat a je tak potřeba přizpůsobit rysy danému datasetu a podmínkám (jedna vs. více kamer, osvětlení v budově vs. na ulici, nízké vs. vysoké rozlišení atp.). Je zřejmé, že nalézt takové příznaky není snadné, což vede k chybám identifikace.

Příznaky bývají zakódovány do různých forem histogramu nebo do vektoru s fixní délkou, mezi kterými lze určit podobnost např. pomocí Eukleidovské vzdálenosti. Metrikou pro porovnávání identit může být také metoda nejbližšího souseda, SVM[7] atp. Příklad systému s využitím obdobného postupu lze najít v článku [44].

Re-identifikace pomocí neuronových sítí. S rozvojem hlubokého učení se trend přesunul k používání neuronových sítí, konkrétně konvolučních (díky jejich vhodnosti pro zpracování obrazových úloh, jak bylo popsáno v kapitole 3.4). Tyto systémy zapouzdřují proces klasického vybírání a extrakce příznaků a díky svému internímu fungování jsou schopné adaptovat se na různé datasety.

Konvoluční neuronové sítě fungují znamenitě při extrakci příznaků z obrazu. Ovšem pokud by se rysy extrahovaly globálně a bez další korekce, snadno by docházelo k nepřesnostem. Navíc značná část vstupního snímku bývá tvořena pozadím či jinými rušivými

[6] Jinak řečeno též rys či vlastnost, ekvivalent anglického výrazu „feature“.

[7] Metoda strojového učení „Support Vector Machine“, popis a vysvětlení lze nalézt např na: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

elementy. Lze tedy uvést příklad, kdy má sledovaná osoba modrou košili, na zádech černý batoh a stojí na silnici před červeným automobilem. Globální rysy extrahované při čelním pohledu na subjekt s automobilem v pozadí (modrá oblast trupu, červené pozadí s obrysy a hranami) budou značně rozdílné oproti těm extrahovaným ze zadního pohledu (černá oblast trupu, jednolitá šedá silnice v pozadí).

Moderní přístupy se proto snaží vylepšit kvalitu extrahovaných rysů třeba tak, že síť přimějí soustředit se na důležité vlastnosti subjektu / lokální oblasti zájmu. Další cestou vedoucí k řešení uvedeného problému je učení/tvorba lepších metrik, které mají vhodně porovnávat příznaky. Takto vylepšené přístupy re-identifikace s využitím neuronových sítí lze nalézt např. v článcích [23] a [11].

Díky dostupnosti moderních frameworků pro hluboké učení (TensorFlow, PyTorch atp.), stále se vyvíjejícím optimalizačním metodám a velkému množství datasetů pro re-identifikaci osob (přes 30 často používaných zdrojů, viz jejich shrnutí[8]) se výsledky znatelně zlepšují [54]. Přestože v nich lze místy najít chyby, které se lidskému pozorovateli jeví jako banální, celkové výsledky nejnovějších metod v některých situacích dokonce překonávají hranici lidského pozorování. Takovou síť popisuje článek [52] – její úspěšnost na datasetu Market-1501 [53] činí 94,4 %, zatímco lidé za stejných podmínek dosahují přibližně 93.5 %.

V kontextu této práce je potřeba ještě zmínit, že uvedené příklady pracují s relativně kvalitními záběry osob – je vidět větší část těla a rozlišení snímků umožňuje i viditelnost některých důležitých detailů, např. obličeje. To však při sledování osob dronem z výšky nemusí platit, např. při větších vzdálenostech může být výřez člověka tvořen jen malým počtem pixelů. V takových situacích je nutné volit algoritmus obezřetně a nelze očekávat stoprocentní funkčnost samostatných uvedených algoritmů. Problém tedy může vyžadovat i použití dalších informací, kupříkladu již zmíněných kontextových údajů (znalost času, GPS pozice atp.).

[8] The Awesome Person Re-Identification Datasets. Dostupné z: https://github.com/NEU-Gou/awesome-reid-dataset

Kapitola 5

Implementace

Mapování trajektorií pohybu chodců v záznamu pořízeném dronem je poměrně komplexní proces. V následující kapitole budou popsány návrh algoritmu a implementace systému, a to jak z pohledu potřebných algoritmů, tak použitých nástrojů, architektury a navržených rozhraní v rámci aplikace. Pro přehlednost bude kladen důraz na několik hlavních bodů, které budou patřičně rozebrány:

- vstupní data a jejich zpracování,
- detekce osob v obraze,
- určení GPS pozice detekovaných subjektů,
- sledování osob,
- výstupní data a jejich úpravy.

Jak bylo dříve zmíněno, existuje množství různých dronů a záznamových zařízení. Na systém také může být kladeno vícero rozličných požadavků při využití v praxi (osvětlení, kvalita záznamu, vzdálenost od sledovaných osob, úhel pohledu atp.). Proto musí být kladen důraz také na modulárnost aplikace a vhodný objektový návrh tak, aby v případě potřeby bylo možné systém jednoduše rozšířit.

5.1 Návrh

V této kapitole bude rozebrán návrh aplikace a další důležité principy, na kterých staví – počínaje volbou a zdůvodněním použitých nástrojů z obecného hlediska, přes styl implementace a architekturu na úrovni zdrojového kódu až po celkový průběh aplikace.

Systém sestává z klíčových částí zmíněných v nadřazené kapitole. Ty jsou samy o sobě obsáhlé celky, jejichž detailní návrh a implementace přesahují rámec této práce. Proto jsou v některých oblastech použita veřejně dostupná řešení, která bylo možné vhodně upravit pro potřeby aplikace.

Důraz tedy musí být kladen nejen na zpracování konkrétního datasetu, ale i na modularitu systému. Mělo by tak být možné jednotlivé části aplikace pohodlně nahradit specializovanými moduly upravenými pro konkrétní potřeby, pokud by aplikace byla nasazena v odlišných situacích.

V návrhu budou popsány pouze základní použité nástroje a části aplikace především z obecného hlediska. Konkrétní použitá řešení, jejich varianty a verze budou popsány až v konkrétních podkapitolách věnovaných implementaci jednotlivých částí programu.

5.1.1 Použité nástroje

Nejprve je potřeba popsat rozdíl mezi online a offline zpracováním obrazu, jelikož je klíčový pro výběr nástrojů:

- V případě online (jinak též „real-time“) zpracování se jedná o úpravy v reálném čase, např. přímo ve dronu, který natáčí video. Už při běžném pořizování 30 či 60 snímků za sekundu jsou pak velmi vysoké nároky na rychlost zpracování, čímž je vyloučeno mnoho pokročilejších a výpočetně náročnějších metod, nebo je potřeba specializovaný hardware na míru.

- Offline zpracování oproti tomu zachází s videozáznamem pořízeným v minulosti. Není tedy limitováno výpočetním výkonem v čase, resp. podmínky jsou mnohem volnější. Např. čas na zpracování 1 snímku se může klidně pohybovat v řádech vteřin i minut, aniž by měl dopad na praktické použití aplikace, zatímco při online zpracování by na stejnou operaci připadly desítky milisekund. V tomto případě tedy není omezen počet použitých nástrojů a lze využít i komplexnější řešení jako jsou náročnější modely neuronových sítí atp.

Tato aplikace odpovídá druhé z uvedených variant, jelikož dodatečně využívá záznam, který byl pořízen v minulosti, provádí tedy offline zpracování obrazu. Čili nejsou kladeny požadavky na extrémní omezení výpočetní náročnosti, čímž se nabízí mnohem více použitelných nástrojů – především strojové učení, včetně náročnějších umělých neuronových sítí, což je důležité z hlediska této práce, jelikož takové metody dosahují lepších výsledků při detekci a sledování osob (viz kap. 3.5 a 4.2).

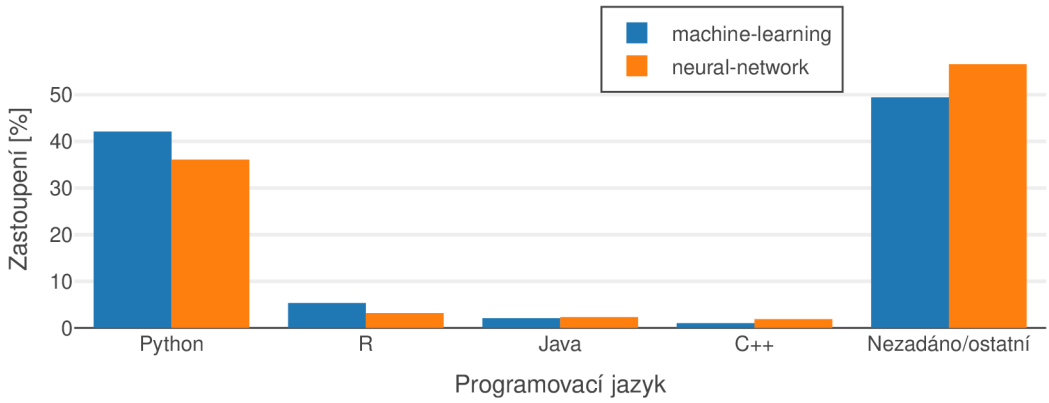
Volba programovacího jazyka

V grafu, který je součástí analýzy na webu [1], lze vidět vývoj popularity programovacích jazyků v oblasti strojového učení (konkrétně kategorie „machine learning“ a „data science“). Je zde patrný vzestup jazyka *Python*, kde postupně získává náskok přibližně 30 %. Výsledky byly získány pomocí srovnání počtu pracovních nabídek dle programovacího jazyka, avšak za období v letech 2012 až 2016.

Pro ověření situace i v současnosti byla v rámci této práce provedena aktualizovaná analýza obdobným způsobem s využitím dat ze serveru pro řešení programátorských otázek. Potvrdila zmíněný trend a ukázala výraznou dominanci *Pythonu* (více než v roce 2016), viz graf 5.1.

Je tedy patrné, že v tomto jazyce se řeší i majorita moderních nástrojů pro detekci objektů, či pro re-identifikaci osob. V rámci této práce je tedy také využit *Python*, aby byla aplikace pohodlně rozšiřitelná a držela se běžných konvencí. Navíc je také platformě nezávislá – vyžaduje pouze interpret jazyka a instalaci použitých nástrojů (instrukce potřebné k zprovoznění programu jsou uvedeny v návodu README.md ve zdrojovém kódu). Konkrétně je aplikace implementována ve verzi *Python 3.7*.

Python 3 je vysokoúrovňový, interpretovaný[1] a dynamicky typovaný[2] jazyk. Byl vydán v roce 2008 a jeho syntaxe je intuitivní a relativně podobná mluvené řeči. Zmíněné vlastnosti v praxi ústí v rychlejší a pohodlnější (větší flexibilita, méně restrikcí) tvorbu aplikací. Jsou vykoupeny horší optimalizací výkonu, většími požadavky na paměť a vyššími nároky na kvalitu kódu u větších projektů, avšak tato negativa u offline zpracování obrazu nejsou podstatná. Díky tomu je *Python* jeden z nejpoužívanějších nástrojů jak pro začátečníky, tak v oblasti výzkumu a strojového učení



Obrázek 5.1: Zastoupení programovacích jazyků ve vybraných tématech na webu *Stack Overflow*[3]. Z celkového počtu otázek označených „machine-learning“ (39 tisíc) a „neural-network“ (15 tisíc) byly vybrány otázky zároveň označené některým programovacím jazykem. Z uvedených hodnot je jasně patrná popularita *Pythonu*. Jazyky s menším podílem než 1 % jsou zahrnuty do skupiny „Nezadáno/ostatní“. Ta dosahuje vysoké hodnoty, jelikož označování otázek není povinné a také mohou být dotazy obecného charakteru. Data jsou platná k dubnu 2020.

Použité externí knihovny

V projektu je využito několik externích *Python* knihoven. Nejdůležitější z nich zde budou popsány. Každá z nich může vyžadovat instalaci dalších balíčků, avšak všechny potřebné závislosti lze pohodlně nainstalovat přes oficiální balíkovací systém *pip*. Ten komponenty stáhne a nainstaluje ve vyžadované verzi z online repositáře *PyPI*[4].

Celkový seznam všech použitých knihoven je uveden v souboru `requirements.py`, což je běžná praxe, jelikož je pak možné všechny závislosti *pipem* nainstalovat jediným příkazem.

[1] Zdrojový kód je vykonáván interpretem, který řídí akce na nižších úrovních. Díky tomu není potřeba zdrojový soubor nejprve překládat do strojového kódu. S takovými programy se tedy většinou pracuje snadněji a mají lepší přenositelnost mezi platformami (do jisté míry), jelikož na cílovém přístroji stačí mít k dispozici interpret a není potřeba překládat projekt na míru operačnímu systému.

[2] Proměnná může nést různé hodnoty, nezávisle na jejich datovém typu, který se tak může dynamicky za běhu aplikace měnit (nemusí být znám v době tvorby zdrojového kódu). Datový typ proměnné tedy není udávám proměnnou samotnou, ale její hodnotou.

[3] *Stack Overflow* je online portál pro řešení nejrůznějších programátorských otázek, kde uživatelé pokládají otázky a členové komunity na ně odpovídají. Stránka udává více než 50 milionů unikátních návštěvníků každý měsíc. Dostupné online: `https://stackoverflow.com`.

[4] Celým názvem *Python Package Index* je oficiální online repositář pro knihovny jazyka *Python*. Více na adrese: https://pypi.org.

Jsou zde také uvedeny příslušné verze, se kterými byl projekt testován a které tím pádem garantují jeho funkčnost.

Díky architektuře aplikace a její modulárnosti (popsáno v kap. 5.1.2) však lze program spustit i s podstatně menším množstvím závislostí. Minimální požadované knihovny pro běh aplikace jsou proto uvedeny v souboru `requirements_minimal.py`. Jejich použitím se uživatel může vyhnout instalaci nadbytečných balíčků, které v daném případě nepotřebuje. Nepotřebná rozšíření tak nezabírají místo v paměti a nezpomalují běh programu. Toto je však doporučeno pouze pro případ testování a ladění aplikace.

NumPy[5] je základní knihovna používaná při vědeckých činnostech a práci s větším množstvím dat. Jejím základem jsou N-rozměrná pole. Nad nimi nabízí rozhraní pro důležité operace, které jsou navíc optimalizované pro výkon při práci s vysokým počtem prvků. Na tomto balíčku staví i mnoho dalších významných knihoven (OpenCV, Pandas atp.). V tomto projektu je NumPy jednak vyžadováno jako jejich prerekvizita, jednak používáno i samostatně napříč celou aplikací, např. pro geometrické operace nebo práci s obrazem.

OpenCV[6] je framework zaměřený na počítačové vidění. Nabízí množství funkcí od obecného zpracování obrazu až po pokročilejší metody, jako je např. detekce obličeje. V projektu je používána pro práci s videem, analýzu obrazu a jeho úpravy atp.

Pandas[7] představuje nástroj pro rychlou a efektivní analýzu/zpracování dat. Na rozdíl od NumPy je jejím základním prvkem dvojrozměrná struktura, tedy pracuje s informacemi, které lze reprezentovat 2D tabulkou. Nabízí spoustu běžně užívaných funkcí při úpravě dat a umí pohodlně zacházet se soubory typu CSV, proto je v aplikaci využita ke zpracování a uchovávání letového deníku.

TensorFlow[8] je velmi obsáhlý framework vhodný pro potřeby strojového učení. Umožňuje tvorbu i trénování umělých neuronových sítí s využitím přehledného vysokoúrovňového rozhraní a mnoho dalšího. V této práci slouží jako základ pro detektor osob (k dispozici je však více typů detekce, není tedy nutně vyžadován).

Matplotlib[9] nabízí vykreslování interaktivních grafů nejrůznějších typů a také jejich exportování do souborů běžně používaných formátů. Navíc dokáže přímo pracovat i s objekty vytvořenými knihovnou NumPy. V projektu je využíván k vykreslování detailních informací v testovacím režimu a také umožňuje zobrazení trajektorií v 3D náhledu.

GeographicLib[10] nabízí širokou škálu geodetických operací, od konverzí souřadnic mezi různými systémy až po výpočty pracující s gravitací a magnetickým polem Země. Originální verze je implementována pro programovací jazyk *C++*, pro *Python* je však dostupné rozhraní umožňující používání geografických operací. V projektu zajišťuje operace nad GPS souřadnicemi (výpočty vzdáleností, přesuny atp.).

PyTorch[11] je moderní framework určený pro potřeby strojového učení. Jedná se o obdobu knihovny TensorFlow, ale je lépe integrovaný do jazyka `Python` (např. lepší možnosti ladění). V tomto projektu je využíván neuronovou sítí, která zajišťuje re-identifikaci osob.

[5] Více informací: https://numpy.org.
[6] Více informací: https://opencv.org.
[7] Více informací: https://pandas.pydata.org.
[8] Více informací: https://www.tensorflow.org.
[9] Více informací: https://matplotlib.org.
[10] Více informací: https://geographiclib.sourceforge.io.
[11] Více informací: https://pytorch.org.

Další použité nástroje

Google Earth[12]. Jedná se o program, který umožňuje prohlížení virtuálního glóbu. Ten zachycuje reálnou podobu planety s použitím satelitních snímků, navíc nabízí i 3D znázornění povrchu. V tomto projektu slouží k zobrazení výsledných trajektorií sledovaných osob na reálnou mapu.

Open Elevation[13]. Jedná se o open-source nástroj, který umožňuje tvorbu serveru s údaji o nadmořské výšce. S ním pak lze komunikovat pomocí http dotazů, kdy na zaslané požadavky obsahující GPS pozice odpovídá hodnotami nadmořské výšky. *Open Elevation* je ovšem sám o sobě pouze nástroj, který neuchovává žádná data. Informace čerpá ze souborů typu GeoTIFF, které obsahují potřebná data.

GeoTIFF je datový formát určený pro ukládání a přenos geodetických dat. Vychází ze standardního formátu TIFF, který slouží pro práci s rastrovou grafikou. Přidává však metadata, která udávají např. prostorové umístění, model geoidu, datum či souřadný systém. Běžně se používá k ukládání informací o nadmořské výšce či jiných informacích o zemském povrchu, které lze zobrazit díky typu formátu i stylem obrázku (např. s intenzitou barev dle nadmořské výšky). Podrobné informace lze nalézt na stránkách NASA[14].

GDAL[15]. Jedná se o konzolový nástroj pro práci a úpravy souborů obsahujících geoprostorové informace (např. zmíněný GeoTIFF). Umožňuje jejich analýzu, úpravy, konverzi mezi vektorovými a rastrovými typy atp. V práci byl použit např. k extrakci oblasti zájmu z datových souborů nadmořských výšek.

QGIS[16] je geografický informační systém, který umožňuje vizualizaci, úpravy a práci s mapami, modely povrchu i dalšími daty v nejrůznějších formátech přehledně v grafickém rozhraní. Při použití digitálního výškového modelu zde lze např. analyzovat výškový profil sledované oblasti. Na rozdíl od běžných GIS aplikací, které jsou často velmi drahé, staví *QGIS* na open source principu a je dostupný zdarma.

Shell skripty[17]. V projektu je k dispozici několik shell skriptů pro úpravu videa (např. změna rozlišení či odstranění audio stopy). Ty však uživatel při běžné práci s aplikací nepoužívá, slouží pouze pro usnadnění manuálního předzpracování videa v případě potřeby.

PyCharm[18]. Aplikace byla implementována ve vývojovém prostředí *PyCharm*. Nabízí rozsáhlou škálu podpůrných funkcí, kontrol kódu a možnosti vlastních rozšíření, díky čemuž pomáhá udržet přehlednost a kvalitu zdrojového kódu. Pro vzdělávací účely navíc nabízí plnou verzi zdarma.

5.1.2 Implementační konvence a modularita aplikace

Jazyk *Python* nabízí různá programovací paradigmata. V projektu je místy využito imperativního stylu[19] (např. hlavní řídící konstrukce a některé pomocné knihovny), avšak

[12] Domovská stránka: https://www.google.com/earth.
[13] Domovská stránka: https://open-elevation.com.
[14] Více na: https://earthdata.nasa.gov/esdis/eso/standards-and-references/geotiff.
[15] Domovská stránka: https://gdal.org/.
[16] Domovská stránka: https://qgis.org.
[17] Označení „shell“ zahrnuje v praxi mnoho implementací skriptovacích jazyků (např. *bash* či *zsh*). Obecně se jedná o standard *Shell Command Language*, který definuje náležitosti pro komunikaci s *POSIX* systémy. Definice dostupná online: https://pubs.opengroup.org/onlinepubs/009695399/utilities/xcu_chap02.html.
[18] Domovská stránka: https://www.jetbrains.com/pycharm.
[19] Více informací lze najít online, např. https://en.wikipedia.org/wiki/Imperative_programming.

základem je převážně OOP[20] (v této práci bude dále předpokládáno, že čtenář je obeznámen s principy objektově orientovaného programování a s ním spojenými pojmy). Z tohoto principu vychází i ABC třídy (z angl. „abstract base class“). Umožňují definici objektových API[21] pomocí základních tříd, jejíchž všichni potomci musí dané rozhraní implementovat. Tím lze zajistit konzistenci a modulárnost aplikace.

Téma práce je poměrně komplexní. Aplikace proto není implementována s fixním použitím jednoho daného nástroje pro každou úlohu, jelikož by tak silně omezovala svoje použití na konkrétní podmínky/dataset a pro přizpůsobení jinému vstupu by vyžadovala zásahy do celé programové struktury.

Namísto toho jsou pro klíčové části (detekce osob, letový deník atp.) definovány požadavky na rozhraní (jinak řečeno také API) formou základního abstraktního objektu (implementováno pomocí zmíněných ABC tříd). Z něho poté musí vycházet každá další konkrétní implementovaná varianta, která může být funkčně odlišná, avšak díky principu dědičnosti poskytuje stejné rozhraní. Následkem toho mohou být všechny takové objekty (mající stejný původ) navzájem libovolně zaměňovány.

Díky popsanému postupu je aplikace snadno přizpůsobitelná a rozšiřitelná – lze implementovat jen potřebnou část, přičemž zbylá struktura zůstává neovlivněna. Zmíněné varianty základních tříd implementující stejnou část algoritmu pomocí rozdílné funkcionality jsou v této práci nazývány jako moduly. O modulech lze ovšem v kontextu *Pythonu* hovořit na více úrovních – názvosloví samo o sobě tedy není jednoznačné, takže při nedostatečném kontextu není zřejmý přesný význam. Proto je nyní potřeba uvést rozdíly v rámci této práce a zavést jednoznačné názvosloví používané v této kapitole pro případy, kdy nebude dostatečně jasný kontext:

- Základní modul – ABC třída, která definuje objektové rozhraní potřebné pro danou část programu.
- Funkční modul – jedná se o objekt, který nabízí totožné rozhraní jako základní modul, ale používá odlišný vnitřní postup. Dva takové objekty stejného typu lze zaměňovat (např. detektor osob založený na Haarových příznacích nahradit detektorem osob využívajícím konvoluční neuronové sítě).
- Souborový modul – proměnné, funkce a objekty sdružené v jednom souboru zdrojového kódu. Teoreticky může obsahovat více funkčních modulů.
- Knihovní modul – jinak jej lze pojmenovat též jako knihovna, balíček či package. Jedná se o balík (může být zkomprimovaný či jinak zabalený do jednoho celku) nebo adresář, který může obsahovat více souborových modulů. Balíček je možné importovat jako celek, nebo z něj vybírat jednotlivé souborové moduly (potažmo přímo jejich metody/objekty).

Výsledný program bude nakonec testován na datasetu popsaném v kap. 6.1.1, který byl pořízen pro účely této práce dronem DJI Spark. Proto aplikace poskytuje minimálně veškeré funkční moduly potřebné pro zpracování testovacího datasetu. Avšak z popsané architektury

[20] Zkr. pro „objektově orientované programování“, čož je hojně využívané programovací paradigma, které umožňuje efektivní znovupoužití komponent a přehlednou architekturu aplikací, kdy jednotlivé objekty jsou znázorněny podobnou formou, jak jsou vnímány v reálném světě. Více informací lze najít online, např. https://en.wikipedia.org/wiki/Object-oriented_programming.

[21] Zkr. angl. výrazu „Application Programming Interface“, jinak řečeno též rozhraní. Jedná se o množinu funkcí/vlastností, které objekt/knihovna/program nabízí pro interakci.

je zřejmé, že v některých částech aplikace může být implementováno i více funkčních modulů zajišťujících totožný úsek algoritmu – a to jak pro dosažení lepších výsledků (např. různé typy detektorů osob), tak za účelem větší přenositelnosti (např. více typů dronů).

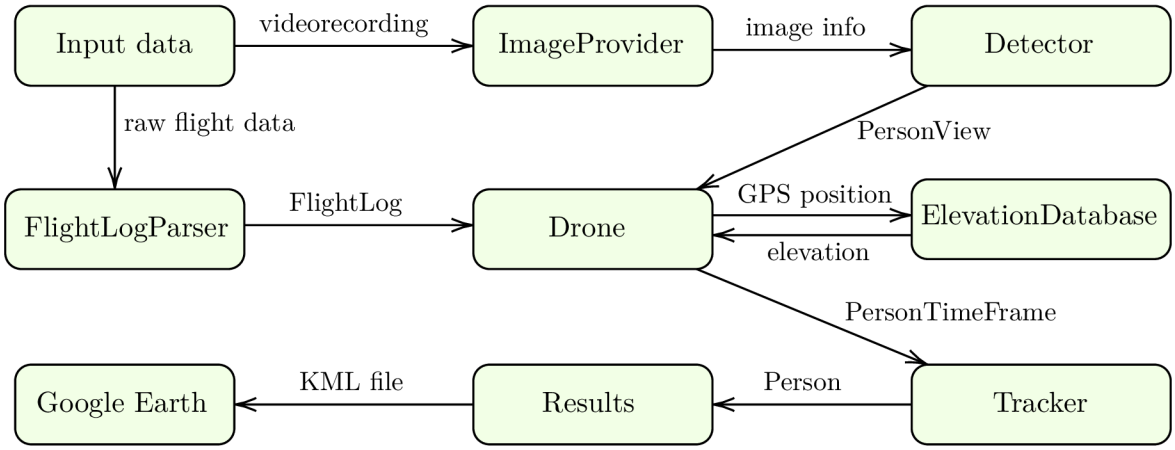
Problém u takového návrhu ovšem nastává ve chvíli, kdy více funkčních modulů vyžaduje různé externí balíčky. Narůstají tak externí závislosti aplikace (potažmo nároky na paměť atp.), ale především může nastat situace, kdy dva objekty vyžadují totožnou knihovnu v nekompatibilních verzích (např. TensorFlow 1 vs. TensorFlow 2). Takovým konfliktům je potřeba předejít.

Proto jsou zdrojové kódy funkčních modulů separovány do jednotlivých souborových modulů. Ty jsou dále sdruženy do knihovních modulů dle svého typu (např. všechny detektory osob jsou sdruženy do adresáře `detectors`). V daném okamžiku tedy lze používat pouze moduly přímo vyžadované pro konkrétní situaci/nastavení, bez nutnosti instalovat externí závislosti, které zrovna nejsou používány. Díky tomu je také možné projekt v základu spustit i s menším množstvím závislostí.

Pokud jednotlivé funkční moduly umožňují přizpůsobení pomocí interních parametrů, které nespadají do předdefinovaného rozhraní základních modulů, může uživatel nastavovat jejich hodnoty v konfiguračním souboru `config.py`. Parametry v něm musí být dostatečně dokumentovány, popř. v rámci této práce jsou další informace uvedeny implementační kapitole 5 (v příslušných kapitolách).

5.1.3 Návrh aplikace

Pro přehlednost je návrh popsán s jistou mírou abstrakce, na jednotlivé celky algoritmu nahlíží obecně. U těchto částí budou průběžně v následujícím textu vyznačeny i názvy příslušných základních modulů, aby s jejich využitím mohly být přehledně znázorněny v návrhovém diagramu aplikace 5.2. Konkrétní implementované varianty jednotlivých modulů však budou popsány až dále v kapitole (počínaje kap. 5.2).



Obrázek 5.2: Návrhový diagram aplikace.

Základní vstupní data aplikace jsou tvořena pořízeným videozáznamem a letovým deníkem dronu. Videozáznam vstupuje do modulu `ImageProvider`, který slouží ke zpracovávání snímků. Ten načte nahrávku po jednotlivých záběrech, přičemž uživatel může např. specifikovat konkrétní úsek videa, či urychlit chod aplikace přeskakováním snímků atp. Každý obraz zde také může být upraven dle potřeby. Výstup pak tvoří jednotlivé snímky s časovými razítky záběrů.

Letový deník může obsahovat spoustu informací, které je potřeba odstranit či upravit, a je potřeba jej synchronizovat s časem pořízení videozáznamu. Dále mohou být údaje ze senzorů nepřesné (několik příkladů nepřesností bylo popsáno v kapitole 2), je proto vhodné je zpracovat a odfiltrovat chyby. K tomu se běžně používá např. Kalmanův filtr (více v kap. 5.2.4). To zajišťuje modul `FlightLogParser`.

V každém snímku je následně potřeba vyhledat chodce, což zajišťuje modul `Detector` pro detekci osob. Vzhledem k náročnosti prostředí (různé objekty, úhly pohledu, vzdálenosti, pohyblivý snímač atp.) je potřeba použít moderní řešení, která jsou založena na konvolučních neuronových sítích. Nabízí se tedy možnost využití nástrojů zmiňovaných v kapitole 3.5, nebo natrénování vlastního detektoru na míru konkrétnímu projektu. Výsledky jsou nakonec zformovány do objektů obsahujících obrazové souřadnice jednotlivých osob a jejich náhled v záběru.

Když je známo umístění osob v obraze, natočení i pozice dronu a jeho výška letu, je možné subjekty lokalizovat v reálném světě, což zajišťuje modul `Drone`. Pro určování vzdálenosti chodců od dronu a výpočet jejich GPS souřadnic jsou využity metody navržené v kap. 4.1. Pokud použitý postup vyžaduje další komponenty, je nutné implementovat také tyto – kupříkladu algoritmus 4.1.5 staví na znalosti nadmořské výšky zemského povrchu, proto pro něj musí existovat databáze, které se může dotazovat na nadmořskou výšku pro konkrétní souřadnice.

Takto byla tedy zjištěna GPS pozice dané osoby. Předchozími kroky pak celkově byly získány následující informace: čas snímku, náhled detekovaného člověka, jeho souřadnice v obraze a GPS pozice. Jejich spojením je vytvořena stopa sledované osoby v čase. Nutno podotknout, že v diagramu 5.2 je označena jako `single track`, což ovšem neznamená celkovou trasu (jak by dvojznačný anglický výraz mohl být vyložen také), ale jen jednu stopu v daném časovém okamžiku.

Pro získání výsledných trajektorií chodců je potřeba jednotlivé stopy napříč videozáznamem správně přiřadit k příslušným osobám. K tomu slouží modul pro re-identifikaci s názvem `Tracker`, který vychází z metod popsaných v kapitole 4.2. Uchovává v sobě záznam všech známých identit a seznam stop jim příslušejících. Každou nově příchozí stopu identifikuje a přiřadí ke známému člověku, nebo na jejím základě vytvoří identitu novou.

V tomto bodě tedy existuje seznam identit, kde každá z nich má přiřazený seznam stop. Spojením všech GPS souřadnic ze stop dané osoby vznikne její jednoznačná trajektorie, což zajišťuje modul `Results`. Výsledné trasy následně upraví různými metodami (např. agregace pozic nebo Kalmanův filtr). Poté mohou být exportovány do souboru (např. typu KML, více informací o tomto formátu v kap. 5.6.2), jenž lze načíst a zobrazit v aplikaci Google Earth.

Celý průběh aplikace tak, jak byl popsán v předchozím textu s využitím označení základních modulů a jejich komunikace, je možné vidět v návrhovém diagramu 5.2. Případná rozšíření (např. databáze nadmořských výšek) budou zmíněna v následujících kapitolách, které popisují konkrétní implementaci různých řešení.

5.2 Vstupní data a jejich zpracování

V této kapitole bude detailněji popsáno zpracování vstupních dat a jejich úpravy. Vstupní data jsou tvořena dvěma klíčovými celky – videozáznamem a letovým deníkem dronu. Tyto prvky musí poskytnout uživatel při spuštění aplikace. Z nahrávky jsou postupně načítány snímky, které mohou dle potřeby projít i volitelným předzpracováním. Letový deník je generován dronem za letu při pořizování videozáznamu. V každém zařízení se však takový výstup

může lišit, proto je potřeba jej zpracovat a extrahovat z něj pouze podstatné informace, které jsou nutné k lokalizaci sledovaných osob.

5.2.1 Načtení videozáznamu

Snímky jsou načítány a zpracovávány modulem typu `ImageProvider`. Jeho úkolem je poskytovat jednotlivé obrázky společně s časovou známkou záběru. Ta je velmi důležitá. Používá se k při výpočtu GPS pozice osoby, kdy je potřeba znát údaje o dronu a jeho orientaci v daný okamžik (více v kap. 5.4). Dále je možné ji využít pro vylepšení re-identifikace při sledování osob (více v kap. 5.5). Následně je uplatněna ještě ke konci celého programu při generování trajektorií sledovaných chodců a upřesnění jejich průběhu (více v kap. 5.6).

Na každý načtený snímek je možné aplikovat nejrůznější úpravy obrazu. Ty jsou provedeny také v rámci tohoto modulu, ovšem zajišťuje je objekt `ImageTweaker` speciálně určený k tomuto účelu (více v kap. 5.2.2). Nejsou tedy implementovány přímo v jednotlivých variantách `ImageProvideru`. Je tomu tak proto, aby byly metody přístupné přes přímé API `ImageTweakeru` s možností přizpůsobení jejích parametrů a pořadí (nejsou tedy všechny ukryty za jedinou inicializační metodou s omezeným rozhraním).

`ImageProvider` je implementován formou iterátoru. Ten umožňuje odložené vyhodnocování (běžně používaný je spíše angl. výraz „lazy evaluation“, který bude dále užíván i v této práci) – tzn. jednotlivé prvky (snímky) jsou načteny až v momentě, kdy jsou potřeba. Díky tomu neaktivní záběry nezabírají místo v paměti a iterátor tak představuje potenciálně nekonečnou datovou strukturu. Konkrétní varianty `ImageProvideru` a jejich možnosti jsou pak popsány níže.

ImageProviderFromVideo

Načítá záběry ze souboru videozáznamu, dokud nedosáhne konce. Pro komfort uživatele nabízí možnost stanovení konkrétního úseku videa pomocí indexu počátečního a koncového snímku. Jelikož jsou běžně videozáznamy pořizovány se snímkovou frekvencí 30 či 60 fps[22], modul dále umožňuje volbu přeskakování/vynechávání snímků – tzn. uživatel zadává, kolik záběrů má být ignorováno předtím, než je jeden navrácen. Aplikace tak může být výrazně urychlena, pokud stačí vyhodnotit např. pouze jeden obraz za sekundu.

Tento modul je určen pro použití v aplikaci ve standardním režimu a při běžném použití by neměla vznikat potřeba jiných variant `ImageProvideru`. Pro testovací a ladící účely však jsou k dispozici i další implementace.

StaticImageProvider a RandomImageProvider

Tyto moduly slouží výhradně k ladícím/testovacím účelům. Pracují na podobném principu. Lze jim nastavit maximální počet snímků, které mohou poskytnout, a dále počet snímků za sekundu, jehož pomocí jsou dopočítány časové známky jednotlivých záběrů. `StaticImageProvider` pak vždy navrací jeden stejný přednastavený snímek, zatímco `RandomImageProvider` pokaždé generuje nový obraz s náhodnými hodnotami pixelů a požadovaným rozlišením.

[22]Počet snímků za sekundu – běžně užívané značení z angl. „frames per second“.

5.2.2 Zpracování obrazu

Zachycený obraz kamerou není dokonalý a může obsahovat např. šum, artefakty způsobené osvětlením či jiné vysokofrekvenční složky. Běžnou praxí v počítačovém vidění je proto snímky upravit. Cílem je odstranit zmíněné vady, ale zároveň zachovat důležité rysy, pomocí kterých budou později extrahovány potřebné informace (např. při detekci a re-identifikaci osob).

Na každý načtený snímek mohou být aplikovány různé metody pro úpravu obrazu pomocí třídy `ImageTweaker`. Ten umožňuje přednastavení požadovaných operací v libovolného pořadí a s volitelnými parametry. Níže jsou popsány některé z nabízených úprav.

Odstranění šumu

Gaussův filtr je jedna z nejpoužívanějších metod pro odstranění šumu v obraze. Probíhá tak, že na celý obraz je postupně aplikována konvoluce s maskou, přičemž její velikost udává, jak širokým okolím bude každý pixel ovlivněn. Hodnoty polí masky jsou potom určeny Gaussovou funkcí. V rovnici 5.1 lze vidět vzorec její 2D podoby a dále hodnoty masky o velikosti 3×3, kde x, y značí souřadnice v obraze a σ udává směrodatnou odchylku normálního rozložení a M značí masku[43].

$$
\begin{aligned}
G(x,y) &= \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \\
M &= \frac{1}{16} \begin{Vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{Vmatrix}
\end{aligned}
\tag{5.1}
$$

Výsledek aplikace Gaussova filtru se vizuálně jeví jako rozmazání obrázku a zahlazení pixelů, které neodpovídaly svému okolí. Současně jsou však i rozostřeny hrany (dle velikosti masky), což může být nežádoucí. Použití metody je proto vhodné zvážit v závislosti na situaci (např. pokud bude dále použita detekce hran), při zpracování obrazu neuronovými sítěmi by však obecně neměl nastat problém. Rozmazání lze pozorovat na obrázku 6.4.

Bilaterální filtr [9] slouží též k odstranění šumu v obraze, na rozdíl od většiny dalších běžně používaných filtrů však dovede zachovat, resp. až zvýraznit hrany. Navíc i v barevných snímcích tak, aby reflektoval vnímání barev a hran lidským okem. Využívá principu Gaussova filtru, navíc ovšem zvažuje také vzájemné rozdíly intenzity pixelů. Kvůli tomu je výpočetně náročnější, avšak nová hodnota daného pixelu pak není ovlivněna body, které jsou od něj odděleny hranou. Ukázku je možné nalézt v kap. 6.2.1.

Úpravy kontrastu

Ekvalizace histogramu se používá pro zlepšení kontrastu obrazu. Změny jsou však vypočítávány na základě histogramu celého snímku jako celku, což by mělo zlepšit kvalitu obrazu jako celku, ale může to vést k negativním změnám v některých jeho částech.

V aplikaci je proto implementována metoda pro adaptivní ekvalizaci histogramu, která řeší obraz po menších částech a změny aplikuje lokálně, kde je třeba. Ve výsledku by tak mělo dojít např. k odhalení tmavých oblastí a navrácení detailů v přespříliš světlých místech, kde osvětlení odstranilo podstatné rysy.

V oblastech, kde se vyskytuje šum, však proto může dojít k jeho umocnění. Tomu se snaží zabránit algoritmus metody CLAHE (z angl. „contrast limited adaptive histogram equalization“), která limituje extrémní hodnoty kontrastu. Ukázku výsledku této metody lze vidět v tutoriálu knihovny OpenCV v sekci Histogram Equalization [4].

5.2.3 Načtení letového záznamu dronu

Zpracování surového letového záznamu je zajištěno modulem typu `FlightLogParser`, který na vstupu přijímá informace ve formě, v jaké je zaznamenává dron. V případě testovacího datasetu 6.1.1 se jedná o dron DJI Spark, proto je implementována jediná konkrétní varianta, a to `FlightLogFarserDjiSpark`, která je dále popsána.

Jelikož letový deník obsahuje data z celého letu, ale pro účely práce jsou potřebná pouze data v době natáčení kamery, je prvním krokem rozdělení do dílčích částí, odpovídajících videozáznamům. Toto je rozlišeno podle atributu kamery, který udává, zda je v natáčení aktivní.

Jelikož surová data obsahují mnoho přebytečných informací a možné chyby, jsou dále části záznamu překonvertovány do předdefinované formy – výstupem je tedy soubor typu CSV s jasně stanovenými názvy sloupců a použitými jednotkami. Všechny případné údaje bez časového razítka jsou smazány. Pro ladící účely umožňuje hodnoty vykreslit do grafu.

5.2.4 Úpravy a opravy letových dat

Vhodně naformátovaný CSV soubor je dále načten modulem `FlightLog`. Ten sjednocuje použitá data a nabízí nad nimi potřebné rozhraní. Dále také umožňuje aplikovat úpravy vedoucí k zpřesnění výpočtů. V práci je implementován funkční modul `FlightLogDjiSpark` pro dron, který byl použit při pořizování testovacího datasetu.

Aby bylo možné určit GPS pozice sledovaných osob dle dříve navržených metod a zaznamenat jejich trajektorie v čase, musí výsledný letový záznam obsahovat následující údaje: čas záznamu, GPS souřadnice dronu, jeho výšku nad povrchem a přesnou orientaci v prostoru.

Natočení je specifikováno třemi údaji – roll (otočení kolem osy x), pitch (otočení kolem osy y) a yaw (otočení kolem osy z). Toto názvosloví bude použito v celé práci (české názvosloví není napříč zdroji jednotné). Bližší vysvětlení orientace os a určování natočení dronu lze nalézt v článku [10].

V kap. 2 byly popsány komerčně používané drony a jejich systémy, uvedeny byly také některé jejich nepřesnosti. V praxi pak mohou nastat další chyby způsobené okolními jevy (špatný GPS signál, vliv počasí atp.), přičemž i minimální odchylka např. v natočení dronu se může promítnout jako několikametrový rozdíl pro vzdálené sledované osoby. Vstupní údaje tak musí být vhodně zpracovány a v případě potřeby odfiltrovány chyby.

V práci je k tomuto účelu použit Kalmanův filtr (popsán níže). Další možností je nastavení pevných odchylek pro jednotlivé údaje o natočení dronu.

Kalmanův filtr

Kalmanův filtr [21] je velmi často používán v letectví. Je možné ho využít k vícero účelům. Může sloužit jako filtr šumu, kdy odstraňuje/upravuje hodnoty, které neodpovídají průběhu vývoje měření. Dále může být použit k predikci neznámého následného stavu či vynechaného údaje při měření. V případě měření různými senzory s redundantními informacemi je s ním možné takováto měření sjednotit a zpřesnit.

V této práci je aplikován na údaje z letového deníku dronu, kde slouží jako filtr nepřesných měření. Dále je aplikován na výsledné trajektorie sledovaných osob, kde slouží pro jejich vyhlazení, resp. větší přiblížení k reálnému pohybu.

Implementován je s použitím knihovny `pykalman`, která nabízí pro použití jeho různé varianty. Bližší vysvětlení a konkrétní výpočetní vztahy pro implementaci lze najít v článku [21].

5.3 Detekce osob v obraze

V rámci této práce bylo zpracováno a vyzkoušeno několik řešení detekce osob pomocí konvolučních neuronových sítí. Teoretické hledisko tohoto přístupu bylo popsáno v kap. 3.5. Po implementační stránce jej zaštiťuje základní modul jménem `Detector`. Do něj vstupuje obraz spolu s časovou známkou snímku. Výstupem je seznam náhledů nalezených osob.

Interakce s modulem je zajištěna jasně definovaným rozhraním, a to metodami `detect` a `detect_multi`. První z nich zpracovává snímky jednotlivé, zatímco druhá zmíněná jich pojme více současně. Obě funkce vedou ke stejnému výsledku, avšak některé detektory mohou umožňovat paralelní zpracování více snímků zároveň a tímto postupem lze značně urychlit dobu běhu.

Výstupní náhled detekované osoby pak představuje objekt `PersonView`. Ten ukládá souřadnice bounding boxu (obdélník ohraničující nalezený objekt) a výřez záběru s danou osobou, přičemž nad nimi nabízí několik pomocných funkcí (např. pro získání obrazových souřadnic předpokládané pozice chodidel). Pro neuronové sítě, jejichž výstup je maska nalezeného objektu místo bounding boxu, je tedy nejprve nutné pro masku dopočítat ohraničující obdélník.

5.3.1 Implementované detektory osob

Aplikace nabízí několik variant detektorů založených na principu konvolučních neuronových sítí. Vzhledem k modulárnosti projektu lze bez obtíží zapojit případné další moduly. Pokud by to tedy konkrétní situace použití vyžadovala, je možné natrénovat i specializované detektory na míru daným potřebám.

Návrh a trénování neuronové sítě pro detekci chodců však nespadá do rámce této práce, proto je jsou využita veřejně dostupná řešení, která bylo možné upravit pro potřeby této práce (kompatibilní nástroje, splnitelné externí závislosti, umožněno licencí atp.). Implementované moduly jsou popsány níže.

OpenPoseBinaryDetector

Vychází z projektu *OpenPose*, který hledá a spojuje jednotlivé části lidského těla (ukázka a další informace v kap. 3.5.1). Výhoda je, že u něj příliš nezáleží na různých úhlech pohledu. Podporováno je jak CPU, tak GPU zpracování a různé verze modelů těla (odlišné části těla, obličeje atp.). Volbu typu detekce lze v modulu nakonfigurovat.

Další kladnou vlastností je absence falešných pozitiv (výskyt pouze extrémně vzácně), jelikož systém skládá osobu z jejích částí. Pokud tedy několik bodů detekuje duplicitně s menší hladinou jistoty, téměř nikdy nepostačují pro tvorbu celé platné osoby, popř. jsou použity jako součást validní detekce.

Tato vlastnost je na druhou stranu vykoupena velmi vysokými nároky na výpočetní výkon. Např. při sledování vzdálenějších osob, kdy je potřeba detekci provádět na snímku s vysokým rozlišením, tak tento detektor není ideální z hlediska výpočetní a paměťové náročnosti.

Proto je možné nastavit rozlišení obrazu, které `OpenPoseBinaryDetector` použije při samotné detekci. Jelikož nároky na zpracování snímku v originálním rozlišení by mohly být extrémní právě kvůli zmíněné výpočetní a paměťové náročnosti, toto nastavení lze měnit speciálně pro potřeby modulu, aniž by ovlivnilo původní snímek. Souřadnice detekovaných osob jsou ve výsledku přeškálovány do původního snímku, takže snížení kvality obrazu omezí pouze tuto část programu a ne další (např. re-identifikace tak může probíhat opět v původní kvalitě).

Dále je implementováno omezení minimálních vyžadovaných částí těla k úspěšnému navrácení detekované osoby, aby nebyla pokaždé navrácena její jiná část (např. hlava v jednom snímku a celá osoba v druhém). Stanovena je také minimální hodnota jistoty detekce, která musí být dosažena pro uznání platné části těla. Tato hodnota může být relativně tolerantní, jelikož detektor nemá příliš problémy s duplicitními detekcemi právě kvůli vyžadování minimálního počtu detekovaných částí, což předchází falešně pozitivním výsledkům poměrně dobře.

Pro modul nebyla použita platformě univerzální implementace v jazyce *Python*, jelikož se na testovacím hardwaru (viz kap. 6.1.3) jevila jako podstatně pomalejší oproti použití předkompilované binární verze (k nalezení na stránkách projektu[23]), která je spouštěna přes *Python* rozhraní *subprocess*, které umožňuje komunikaci mimo prostředí programu. Vzniká tak režie navíc, ale detekce přesto proběhne rychleji. Modul je proto dostupný pouze pro operační systém Windows, jelikož binární spouštěcí soubory nejsou aktuálně pro jiné systémy vydávány.

TensorFlowV1ModelZooDetector

V rámci projektu TensorFlow existuje nabídka modelů neuronových sítí natrénovaných na různých datasetech. `TensorFlowV1ModelDetector` umožňuje použít libovolný z těch, které podporují detekci osob. Seznam všech poskytovaných modelů a potřebné informace lze najít na stránce [14]. Údaje jednotlivých modelů jsou exportovány do souboru, který obsahuje jak strukturu dané sítě, tak naučené hodnoty jejích parametrů.

V rámci projektu byly vyzkoušeny detektory natrénované na datasetu COCO[24], které mají na výstupu bounding box (ne masku), např. „SSD with Resnet 50 v1 FPN feature extractor, shared box predictor and focal“ či „Faster R-CNN with Inception v2, configuration for MSCOCO Dataset“.

Výhodou oproti `OpenPoseBinaryDetectoru` je konzistence ohraničujících obdélníků osob napříč snímky. Modely z TensorFlow zoo by také měly dosahovat podstatně rychlejších časů detekce (nehledají desítky bodů na těle, ale osobu jako celek). Pro přehled jsou předpokládané doby zpracování a uvedeny na oficiální stránce [14]. Zde je také možné nalézt jejich hodnocení pomocí metriky mAP [17]. Ta zde sice není udávána čistě pro detekci osob, ale napříč všemi kategoriemi objektů, přesto přibližně vypovídá o kvalitě modelu.

V rámci `TensorFlowV1ModelZooDetector` je možné konfigurovat hladinu detekční jistoty neuronové sítě (síť ji udává pro každý potenciální výsledek), která musí být dosažena, aby byl nalezený objekt považován za platný. Nižší hodnota znamená správné určení většiny osob, avšak mohou se objevovat falešné detekce nebo duplicitní nálezy více bounding boxů pro jednu osobu. Vyšší hodnota pak znamená méně falešných nálezů, ale zároveň zvedá počet ignorovaných objektů, které by jinak byly validní (např. osoby vzdálenější nebo nevhodně osvětlené). Tuto problematiku lze vidět na obr.6.7.

[23] https://github.com/CMU-Perceptual-Computing-Lab/openpose/releases
[24] Dostupnéz:http://cocodataset.org.

Algoritmus 1 Algoritmus pro získání detekovaných osob používaný v modulu `TensorFlowV1ModelZooDetector`. Proměnná $PERSON$ značí typ detekovaného objektu pro člověka, $THRESHOLD$ je minimální nastavená hodnota detekční jistoty, $TOLERANCE$ je maximální povolený překryv dvou bounding boxů.

obtain all detected *objects* from the neural network
for every $o \in objects$ **do**
  if (type of $o == PERSON$) **and** (confidence of $o \geq THRESHOLD$) **then**
    append o to *candidates*
  end if
end for
for every $x \in candidates$ **do**
  for every $y \in candidates$ **do**
    if (bounding box overlap of x by $y \geq TOLERANCE$) **and** ($x \neq y$) **then**
      if confidence of $x \leq$ confidence of y **then**
        delete x from *candidates*
      else
        delete y from *candidates*
      end if
    end if
  end for
end for
return *candidates*

Pro částečné řešení problému duplicitních výsledků je implementována možnost stanovení maximálního povoleného překryvu osob. Pokud se dva (nebo více) bounding boxy překrývají z větší části, než umožňuje limit (např. více než polovinou plochy), je uznán pouze výsledek s vyšší hodnotou jistoty a druhý ignorován. Celkový uvedený postup pro získávání výsledných osob je pak v pseudokódu popsán algoritmem 1.

DetectorUsingPredefinedSerializedResults

Umožňuje jednoduše načíst výsledky předchystané detekce osob z disku, čímž obchází nutnost opakování reálné detekce při každém běhu aplikace. Výsledky mohou být předgenerovány a uloženy na disk (viz kap. 5.3.2). Při běhu aplikace pak mohou být výsledky pouze načítány, což podstatně zrychlí běh programu oproti reálné detekci pomocí neuronových sítí. Tento postup je však spíše určen pro testovací a ladící účely.

5.3.2 Export výsledků detekce

Snímky jsou aplikací zpracovávány po jednom. Za běhu programu tedy není možné nejprve detekovat osoby ve všech záběrech a poté s nimi dále pracovat (přílišné nároky na operační paměť, vznikla by nutnost omezení maximálního počtu snímků). Pokud by tak chtěl uživatel vyzkoušet různé varianty nastavení a optimalizací (např. při sledování osob), musel by s každým dalším během aplikace znovu provádět detekce osob, což je výpočetně náročný proces.

Proto existuje modul `DetectionsExporter`, který umožňuje provést samostatnou detekci na zadaném videozáznamu pomocí zvoleného detektoru osob. Výsledky jsou pak ulo-

ženy na disk – pro každý snímek je dle časového razítka vytvořen soubor obsahující serializované detekce.

Při načítání snímků pak je potřeba dbát na to, že v různých verzích knihoven *NumPy* či *OpenCV* se mohou časové razítka snímků mírně lišit. Názvy výsledných souborů pak nemusí přesně odpovídat a je potřeba je načítat s tolerancí, jejíž velikost by se měla odvíjet od snímkové frekvence záznamu.

5.4 Určení GPS pozice osob

Metody k určení GPS pozice sledovaných osob byly navrženy v kap. 4.1. Jejich výstupem je GPS pozice obsahující zeměpisnou šířku a délku, popř. také nadmořskou výšku. Lokalizaci osob zajišťují funkční moduly typu `Drone` pomocí metody `calculate_gps_position_of_person`.

K výpočtu pozice sledovaných chodců může modul využívat veškerá data získaná v předchozích krocích aplikace. Jedná se tedy o letový deník, ze kterého lze určit orientaci dronu v prostoru (hodnoty vybírány na základě časové známky snímku), a vstupní obrázek s nalezenými osobami a jejich obrazovými souřadnicemi.

Orientaci dronu a kamery v prostoru (jinými slovy směr pohledu) lze popsat 3D vektorem v North–East–Down souřadnicovém systému (dále jen NED), který se běžně používá v letectví. Jeho osa `x` směřuje k severu, `y` k východu a `z` dolů směrem k povrchu Země (většina událostí se děje pod hladinou letu, proto je definice positivního růstu souřadnic tímto směrem vhodnější k použití v praxi). Pro zjednodušení bude v této kapitole považována orientace dronu a kamery za totožný údaj (odstínění této problematiky je řešeno modulem `FlightLog`, viz kap. 5.2.4).

Ve všech navrhovaných metodách je pak potřeba určit také směr, kterým leží sledovaná osoba, tedy 3D směrový vektor polopřímky $\overrightarrow{dron–osoba}$. K tomu jsou však potřeba nejen obrazové souřadnice detekovaného subjektu, ale také informace o zorném poli záznamového zařízení. K tomuto účelu byl vytvořen základní modul `Camera`. Blíže je proces popsán v kap. 5.4.1 níže.

K přesnějším výsledkům při určování pozice osoby v prostoru mohou dále pomoci informace o nadmořské výšce v oblasti, pročež byl navržen modul `ElevationDatabase` (viz kap. 5.4.2 níže), který přístup k takovým datům umožňuje.

`Drone` mimo jiné mohou využívat i další moduly (např. při získávání a úpravách výsledků, ve kterých může být zobrazena také jeho trajektorie). Nabízí proto i několik povinných atributů pro pohodlnější získávání informací, jako např. jméno, pozici, výšku letu či natočení v prostoru. Dále také metody jako např. `update_to_time`, která umožňuje veškeré interní vlastnosti modulu synchronizovat do požadovaného časového okamžiku. Rozhraní je jasně definováno i dostatečně komentováno a snaží se držet obvyklých konvencí[25], nebude však detailně rozebíráno, jelikož není uživateli běžně přístupné.

DroneDjiSpark

V rámci této práce existuje konkrétní modul pro dron DJI Spark (více informací v kap. 2.1.2), kterým byl pořízen testovací dataset (viz kap. 6.1.1). Pro něj byly implementovány metody navržené v kap. 4.1.4 a 4.1.5, které budou dále rozebrány v kap. 5.4.3 a 5.4.4).

[25] Více informací: https://developer.dji.com/mobile-sdk/documentation/introduction/flightController_concepts.html.

5.4.1 Určení směrového vektoru osoby

Pouhý 3D vektor pohledu dronu v prostoru pro výpočet GPS pozice osoby nestačí, jelikož směřuje do centra snímku. Chodci se však mohou vyskytovat v libovolném bodě obrazu, pročež je nutné určit směrový vektor polopřímky $\overrightarrow{dron–osoba}$ pro každého jednotlivce (v práci bude zkracován též jako směrový vektor osoby či směr osoby).

K určení směrového vektoru osoby slouží modul `Camera`. Uchovává údaje o pixelové výšce a šířce obrazu, které je možné získat z videozáznamu, ale především informaci o velikosti zorného pole kamery (stupně horizontálního a vertikálního FoV), jenž je charakteristické pro konkrétní typ záznamového zařízení a je potřeba ho nakonfigurovat. V rámci práce je nabízena varianta `CameraDjiSpark`, která popisuje parametry kamery[26] poskytované ve výchozí výbavě dronu DJI Spark.

Pokud výrobce u kamery FoV neuvádí, je možné tento údaj zjistit pomocí knihovny OpenCV a kalibračních měření. Pokud je k dispozici pouze horizontální, resp. vertikální složka, lze druhou část dopočítat pomocí nabízených metod `calculate_horizontal_fov` a `calculate_vertical_fov`. Ty vychází ze vztahů uvedených v rovnicích (tzn. prosté násobení jedné složky poměrem stran nestačí):

$$\begin{aligned} \tan\left(\frac{fov_{horizontal}}{2}\right) &= \frac{width}{distance}, \\ \tan\left(\frac{fov_{vertical}}{2}\right) &= \frac{height}{distance}, \\ ratio &= \frac{width}{height}, \end{aligned}$$

kde $width/height$ udávají šířku/výšku viditelné výseče, která se mění v závislosti na vzdálenosti pozorovatele od obrazu $distance$. Horizontální/vertikální složka zorného pole je označena proměnnou fov, $ratio$ pak vyjadřuje poměr stran obrazu.

Samotný výpočet směrového vektoru polopřímky $\overrightarrow{dron–osoba}$ pak probíhá ve dvou krocích. V prvním je vypočítán rozdíl mezi směrovým vektorem pohledu dronu a směrovým vektorem osoby. Ten je rozložený na vertikální a horizontální složku (v orientaci podle NED konvence) stanovenou pomocí rovnic:

$$\begin{aligned} angle_{horizontal} &= \frac{person_x - center_x}{width * fov_{horizontal}}, \\ angle_{horizontal} &= \frac{person_y - center_y}{height * fov_{vertical}}, \end{aligned}$$

kde proměnná $angle$ značí úhel odchylky směrového vektoru osoby, $center$ udává souřadnice středu obrazu, $person$ značí pomyslný bod, ve kterém leží osoba (např. střed bounding boxu, nebo jeho spodní hrany jako předpoklad místa, kde chodec stojí). Souřadnice v ose x, resp. y, indexují obraz horizontálně, resp. vertikálně. Ostatní proměnné odpovídají konvencím stanoveným v předchozích rovnicích.

Druhým krokem je již finální výpočet 3D směrového vektoru $\vec{v}$ pro danou osobu v souřadném systému pohledu dronu, přičemž směr os dodržuje NED konvenci – tzn. osa x roste ve směru pohledu dronu, y kolmo směrem k pravému křídlu, z prochází tělem dronu shora dolů. Vzorec pro jeho výpočet je dán vztahem (proměnné popsány v předchozích rovnicích):

$$\vec{v} = distance * [1, \tan(angle_{horizontal}), \tan(angle_{vertical})].$$

[26] Více informací v oficiální specifikaci: https://www.dji.com/cz/spark/info#specs.

5.4.2 Databáze nadmořské výšky

Třída `ElevationDatabase` definuje rozhraní pro snadný přístup k informacím o nadmořské výšce přímo v prostředí *Pythonu*. V podstatě se jedná o proxy objekt[27]. Aplikace tak může implementovat vlastní databázi, nebo v případě potřeby komunikovat s jinými nástroji. Zároveň však na žádném konkrétním externím programu není závislá.

`ElevationDatabase` nabízí metody `get_elevation` a `get_elevations`, které jsou vzájemně zastupitelné. Jejich vstupem jsou GPS pozice (zeměpisná šířka a délka) a výstupem údaj o nadmořské výšce (uváděné v metrech nad mořem), pokud tedy databáze údaj pro požadovanou lokaci obsahuje (jinak nevrací žádná data). Ve výsledné aplikaci jsou implementovány dvě konkrétní varianty tohoto modulu.

FlatEarthElevationDatabase

Tento modul odpovídá na všechny požadavky stejnou hodnotou nadmořské výšky. Ze základu navrací 0 m. n. m., ale umožňuje volbu libovolného čísla. Může být využit, pokud pro danou oblast nejsou k dispozici reálná data o zemském povrchu.

OpenElevationDatabase

Tento modul je v , který dokáže komunikovat s libovolnými *Open Elevation* (více informací v kap. 5.1.1) servery. Modul lze nakonfigurovat tak, aby čerpal data z veřejně dostupného *Open Elevation* API[28], nebo z libovolného serveru stejného typu.

Veřejně dostupné API je však pro vysoký zájem a extrémní provoz často nedostupné, popř. přetížené, pročež jednotlivé dotazy trvají desítky sekund i déle. Taková situace je pro praktické využití v tomto projektu nevyhovující – obvykle je vyžadováno několik dotazů pro určení GPS pozice každé jednotlivé osoby.

Ideální variantou je proto spuštění vlastního *Open Elevation* serveru, např. lokálně na počítači, kde běží i samotná aplikace pro mapování trajektorií chodců. Oficiální návod, jak toho dosáhnout, je k dispozici v github repositáři nástroje, ovšem kvůli delší neaktivitě projektu jsou některé body (např. skripty pro získání datasetu) nefunkční. Některé z těchto problémů jsou opraveny v odvozeném projektu[29].

Pro aktualizovaný a ověřený postup ke zprovoznění *Open Elevation* serveru byl v rámci této práce vytvořen návod, který lze nalézt v souboru `open_elevation_tutorial.md`. Obsahuje také několik odkazů pro stažení veřejně dostupných datasetů (DEM, zkr. angl. „Digital Elevation Model“) a posloupnost bodů pro jejich případnou úpravu. Některé totiž např. nabízí pouze soubory s daty pro plochu velikosti Evropy nebo obsahují porušené části, které je potřeba odstranit.

V rámci projektu jsou nabídnuty údaje o nadmořských výškách v oblasti, kde vznikl testovací dataset (viz 6.1.1), a to ze dvou zdrojů z výše zmíněného návodu. V prvním případě se jedná o projekt „The TanDEM-X 90m Digital Elevation Model“[30] s horizontálním rozlišením přibližně 90 metrů (v úrovni rovníku). V druhém případě jde o data z projektu „Copernicus Land Monitoring Service – EU-DEM“[31], jehož rozlišení je 25 metrů a vertikální

[27] Popis návrhového vzoru „proxy“ lze nalézt na: https://refactoring.guru/design-patterns/proxy.
[28] Veřejné API je dostupné na stránce: https://api.open-elevation.com.
[28] https://github.com/Jorl17/open-elevation/blob/master/docs/host-your-own.md
[29] https://github.com/Developer66/open-elevation
[30] Více informací: https://geoservice.dlr.de/web/dataguide/tdm90.
[31] Více informací: https://www.eea.europa.eu/data-and-maps/data/copernicus-land-monitoring-service-eu-dem.

přesnost přibližně 7 metrů – jedná se tedy o velice podrobný dataset (mapuje pouze oblast Evropy).

5.4.3 Metoda určení vzdálenosti objektu na rovném povrchu

Metoda navržená v kap. 4.1.4 byla implementována, ovšem shledána jako nedostačující (více v kap. 6.2.4), jelikož je striktně omezena pro záběry osob na rovném povrchu/ploše, pročež není uživateli přímo nabízena. Poněvadž byl její teoretický algoritmus popsán v odkazované kapitole a je téměř celý obsažen v postupu metody z kap. 4.1.5, nebude její implementace blíže popsána.

V případě potřeby může být simulována použitím metody z kap. 4.1.5, kdy namísto databáze reálných nadmořských výšek bude předána varianta, která vždy vrací stejnou (libovolnou) nadmořskou výšku (viz kap. 5.4.2). Originální verze je mírně rychlejší (neprobíhá aproximace při zjišťování povrchu pomocí dotazů do `ElevationDatabase`), rozdíl je však zanedbatelný a výsledek je při příslušném nastavení totožný.

5.4.4 Metoda určení pozice objektu s využitím nadmořské výšky

Metoda navržená v kap. 4.1.4 je v podstatě varianta metody popsané v kap. 4.1.4 rozšířená o využití znalosti nadmořské výšky povrchu, po kterém se osoby pohybují. Pro svou funkci tedy vyžaduje modul typu `ElevationDatabase`, s jehož pomocí aproximuje polohu osoby s požadovanou přesností.

Celkový průběh z vysokoúrovňového pohledu je popsán algoritmem 2, který je v následujícím textu podrobněji vysvětlen a doplněn o potřebné závislosti.

Algoritmus 2 Vysokoúrovňový algoritmus pro určení pozice osob metodou navrženou v kap. 4.1.5. Algoritmus je podrobněji vysvětlen v kap. 5.4.4.

```
1: get 3D position and orientation of drone from FlightLog by frame timestamp
2: calculate drone elevation using its position and ElevationDatabase
3: if drone elevation unknown then
4:     return null
5: end if
6: get person direction vector from Camera
7: transfer person direction vector from drone-coordinates to NED-coordinates
8: approximate closest ground point in person direction using elevation database
9: if approximation failed then
10:    return null
11: else
12:    return approximated ground position as person position
13: end if
```

Nejprve je nutné z leteckého deníku získat data vypovídající o umístění dronu v 3D prostoru, konkrétně pozici (výšku nad terénem a zeměpisnou šířku a délku) a natočení (yaw, pitch, roll). Následně je z `ElevationDatabase` získána nadmořská výška bodu, nad kterým se dron vznáší. Pokud není známa, algoritmus nemůže pokračovat (nevrací žádnou pozici), v opačném případě je k ní přičtena výška dronu nad terénem, čímž se určí nadmořská výška samotného dronu.

Jelikož nadmořská výška dronu je určena relativně vůči povrchu a polopřímka $\overrightarrow{dron–osoba}$ (na níž leží hledaná osoba) vychází z této pozice dronu, nejsou pro určení GPS souřadnic

osoby důležité absolutní hodnoty nadmořských výšek v databázi, ale pouze reliéf povrchu. Z tohoto důvodu tedy pro výslednou zeměpisnou šířku a délku nevadí, pokud jsou celková data plošně položena např. o 100 m níže/výše než reálný zemský povrch. Pokud tedy daný model dronu nabízí i informaci o nadmořské výšce, je i přesto obecně lepší ji dopočítat relativně k použité databázi.

Následně je pomocí modulu `Camera` získán směrový vektor polopřímky $\overrightarrow{dron–osoba}$, který je uváděn v souřadném systému dronu (osa x roste ve směru pohledu, y jde k pravému křídlu, z prochází dronem od shora k podvozku). Pro určení GPS souřadnic tak musí být nejprve převeden do souřadného systému North–East–Down. K této operaci je využito následující rotační matice (vysvětlení lze najít zde [10]), kde $y/p/r$ značí yaw/pitch/roll úhly natočení dronu:

$$M = \begin{Vmatrix} \cos y \cos p & \cos p \sin y & -\sin p \\ \cos y \sin r \sin p - \cos r \sin y & \cos r \cos y + \sin r \sin y \sin p & \cos p \sin r \\ \sin r \sin y + \cos r \cos y \sin p & \cos r \sin y \sin p - \cos y \sin r & \cos r \cos p \end{Vmatrix}$$

Jakmile je znám směr (jehož výchozím bodem je pozice dronu), kterým se člověk nachází, jediná chybějící informace k výpočtu GPS pozice osoby je vzdálenost $|dron–osoba|$. Ta může být s použitím speciálních zařízení v jiných metodách měřena či odhadována.

Tato metoda ovšem vychází z předpokladu, že dron podobnými specializovanými zařízeními nedisponuje, pročež musí být zmíněná vzdálenost aproximována pomocí znalosti nadmořské výšky povrchu (po němž se dle očekávání chodci pohybují), jak jest popsáno v následující podkapitole 5.4.4.

Aproximace polohy pomocí nadmořské výšky

Aproximace nejbližšího bodu zemského povrchu v daném směru (implementována v interním knihovním modulu pro geodetické operace) je zachycena algoritmem 3. Vstup pos_0 představuje počáteční pozici dronu (včetně jeho nadmořské výšky), $\overrightarrow{v}_{person}$ je směrový vektor osoby, *maxDist* je maximální vzdálenost od dronu, ve které se může osoba nacházet, *desiredAccuracy* určuje požadovanou finální přesnost výsledné pozice, *stepSize* je velikost počátečního kroku a *stepReductionCoef* udává, jak moc se krok změní při každé iteraci.

Funkce tedy vychází z počáteční pozice dronu a hledá po krocích ve směru osoby, kde se protne pohled dronu s povrchem země, na němž dle předpokladu stojí sledovaný člověk. Stanovení maximální vzdálenosti osoby předchází nekonečnému hledání v případě, kdy by nedošlo k protnutí pohledu s povrchem, např. nepřesnými údaji v letovém záznamu. Přesnost je dále implicitně ovlivněna kvalitou použité databáze nadmořských výšek, explicitně je dána velikostí kroku, který byl použit k aproximaci.

Čím menší je však krok, tím více výpočtů a dotazů na `ElevationDatabase` je provedeno, tudíž není žádoucí ihned prohledávat celý prostor s požadovanou přesností (např. úsek 500 m rozdělit po 1 cm). Proto existuje jak možnost stanovení finální přesnosti, tak počátečního kroku a rychlosti, jakou se má jeho velikost zmenšovat. Při každé další iteraci je pak prohledáván pouze prostor velikosti předchozího kroku.

Může ovšem nastat i případ, kdy bude počáteční krok příliš velký, pročež přeskočí např. menší nerovnost neprotne povrch. Pro tento případ je algoritmus navržen tak, aby neskončil bez výsledku, ale opakoval další iteraci hledání s novým menším krokem v celém původním rozsahu. V konkrétním případě nechť nabývá max. vzdálenost 500 m, krok 10 m a redukční koeficient jest 0.1. Po úspěšné první iteraci bude nová hodnota vzdálenosti pouhých 10 m a krok 1 m. Při neúspěchu pak bude nový krok zmenšen také na 1 m, ale funkce prohledá znovu celých 500 m.

Z optimalizačních důvodů je pak také implementováno dávkové prohledávání prostoru, aby nebyly všechny potřebné hodnoty zvlášť počítány a `ElevationDatabase` kontaktována pro každý jednotlivý krok. `DroneDjiSpark` umožňuje dále nastavit i volitelnou maximální horizontální (či sférickou) vzdálenost mezi nalezenou osobou a GPS pozicí dronu, např. pro omezení nežádoucích prvků v okolí.

Algoritmus 3 Abstraktní algoritmus pro aproximaci nejbližšího bodu, ve kterém směrový vektor pohledu dronu protíná zemský povrch. Názvosloví popsáno v kap. 5.4.4.

while *desiredAccuracy* was not reached **do**
  do steps in the direction $\overrightarrow{v}_{person}$ from pos_0 until *maxDist* is reached
  find the first position pos_i below ground
  if $stepSize \leq desiredAccuracy$ **then**
    if pos_i was found **then**
      return pos_i
    else
      return *null*
    end if
  end if
  $previousStepSize = stepSize$
  $stepSize = stepReductionCoef * stepSize$
  if pos_i was found **then**
    $pos_0 = pos_{i-1}$
    $maxDist = previousStepSize$
  end if
end while

5.5 Sledování osob a re-identifikace

Tato kapitola popisuje implementaci části aplikace, která zajišťuje rozlišení jednotlivých nalezených osob, tvorbu identit a přiřazení nových nálezů k těmto identitám. Toto zajišťuje modul `Tracker`, který uchovává známé identity a přiřazuje k nim nové stopy osob.

Stopa osoby v čase (třída `PersonTimeFrame`) se skládá z časového razítka snímku, ve kterém byla detekována (nejedná se tedy o stopu ve významu trasy, ale o stopu jako jeden bod výskytu osoby v daném okamžiku), z bounding boxu ohraničujícího její výskyt v původním záběru obraze, náhledu této osoby (výřez obrazu) a její GPS pozice. Pro optimalizaci programu také může obsahovat další volitelná data – např. soubor dříve extrahovaných rysů, které pak nemusí být znovu vypočítávány při porovnávání s nově příchozími osobami.

Základy postupů používaných pro re-identifikaci osob byly popsány v kap. 4.2. V rámci práce jsou implementovány 2 funkční moduly zajišťujicí re-identifikaci, konkrétně se jedná o `HistogramTracker` a `AlignedReIdTracker`, které budou popsány níže.

Moduly lze použít pro bezkontextovou re-identifikaci založenou čistě na obrazovém vstupu jednotlivých osob. V tomto případě jsou extrahovány rysy pro porovnávané náhledy osob

Implementována byla ovšem i kontextová vylepšení, která vychází ze základního porovnávání rysů a dále ho zpřesňují/omezují s využitím známých informací:

- Pozice na základě obrazových souřadnic – porovnávané osoby se musí nacházet poblíž nové stopy na základě pozice bounding boxů určených detektorem osob. Pokud

jejich vzdálenost přesahuje maximální povolenou hodnotu, jsou ignorovány. Povolená vzdálenost je vypočítána dle zadané maximální povolené rychlosti osoby a toleranční odchylky. Metoda je náchylná na prudké změny kamery, které mohou vést k falešně negativnímu výsledku i v případě blízkých osob, nebo naopak v opačném případě nemají kýžený efekt.

- Pozice na základě GPS souřadnic – postup je obdobný jako v případě omezení na základě obrazových souřadnic, avšak využívá se určená GPS pozice osob. Toto vede k odstranění citlivosti na prudké pohyby kamery, jelikož GPS pozice by měla být určena nezávisle na natočení dronu. Problémem je nepřesnost vnášená jinými částmi algoritmu (např. letový deník či výškový model povrchu).

- Limit 1 stopy pro 1 osobu v 1 snímku – pro všechny stopy v rámci jednoho snímku jsou určeny odpovídající identity na základě předchozích kritérií. Jelikož se ale každá osoba může ve snímku nacházet pouze jedenkrát, je ve výsledku přiřazena ke každé identitě max. 1 stopa, přestože mohla identita vyhovovat více vstupům. Vítězná stopa je vybrána na základě nejvyšší podobnosti, ostatní (chyby či duplicity) jsou ignorovány. Toto je zajištěno funkcí `track_all_in_video_frame`.

AlignedReIdTracker

Modul vychází z neuronové sítě typu AlignedReID [52]. Konkrétně je implementováno využití volně dostupného modelu[32] stavějícího na architektuře Resnet50 (je využita jako tzv. *backbone*, což je typ struktury jedné části konvoluční neuronové sítě) a natrénovaného na datasetu Market-1501 [53].

Síť je trénována s použitím kombinace globálních a lokálních příznaků extrahovaných ze snímků s osobami [52]. Lokální příznaky (které jsou patřičně zarovnány tak, aby si odpovídaly mezi porovnávanými snímky) pomáhají vhodně natrénovat síť. Pro výsledné porovnávání však jsou již použity čistě globální rysy.

Neuronová síť bere jako vstup obrázky osob o velikosti 128×256 pixelů (jinak velké obrázky musí být patřičně přizpůsobeny). Na výstupu udává vektory příznaků. Získané vektory jsou uloženy do stop osob pro další použití. Výsledná podobnost osob se pak porovnává na základě střední euklidovské vzdálenosti získaných vektorů, které byly normalizovány. Vzdálenost se počítá jako:

$$\sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}$$

kde u_i, v_i značí příslušné prvky porovnávaných vektorů a n udává délku vektorů.

Rozhodnutí, zda stopa přísluší dané identitě, je založeno na uživatelem stanoveném limitu podobnosti. Ta se může pohybovat v intervalu $\langle 0; 1 \rangle$, přičemž 0 udává perfektní shodu a 1 naprostou rozdílnost. Limit je v modulu udáván parametrem `similarity_threshold`.

Dále lze nastavit porovnávání na základě pozice osob (tolerance a maximální rychlost) a vyčerpávající porovnávání společně s maximálním počtem použitých stop pro jednu osobu v rámci jednoho porovnávání. Zapnutí vyčerpávajícího porovnávání umožňuje při identifikaci nově příchozí stopy použít stanovený počet stop od každé porovnávané identity. V opačném případě je stopa porovnávána pouze s posledními (nejaktuálnějšími) údaji každé osoby.

[32] Dostupný z: https://github.com/michuanhaohao/AlignedReID

HistogramTracker

Tento modul vychází ze základní techniky analýzy obrazu. Extrakce rysů probíhá formou počítání multidimenzionálního histogramu. Pro každou barevnou složku obrazu je stanoven počet binů (tj. intervalů pro různé intenzity barvy), v jejichž rozmezí je spočítáno množství příslušných pixelů. Výsledné hodnoty jsou normalizovány a následně zkombinovány do vektoru o délce N^3, přičemž N udává počet binů. Tato technika se ukázala jako efektivnější oproti prostému porovnávání jednotlivých barevných složek.

Histogramy dále nejsou porovnávány pomocí euklidovské vzdálenosti. Na testovacím datasetu 6.1.1 se ukázalo, že samotná metrika podává poměrně nekonzistentní výsledky. Byly proto implementovány 2 různé metriky. Prvním je průnik histogramů, který se počítá jako

$$\sum_{i=1}^{n} min(k_i, l_i)$$

kde k_i, l_i udávají počet pixelů pro danou barevnou kombinaci a n udává celkový počet kombinací v histogramu.

Druhou metodou je Hellingerova vzdálenost. Více o této metodě a jejím výpočtu lze najít v dokumentaci knihovny OpenCV [40]. Obě použité metody pracují rozdílně. Zatímco vyšší hodnota průniku histogramů (parametr `intersect_threshold`) znamená vyšší podobnost, u Hellingerovy metody (parametr `hellinger_threshold`) je to přesně naopak, tzn. nižší hodnota udává větší podobnost. Výsledná stopa je přiřazena, jen pokud obě metody dosáhnou požadované hranice podobnosti a zároveň maxima podobnosti u stejné identity.

5.6 Výstup aplikace a zpracování výsledků

Výstupem programu jsou trajektorie sledovaných chodců určené jejich GPS pozicemi v čase. Výsledky musí jít zobrazit v programu Google Earth, pročež musí být exportovány do souboru. Veškeré zpracování a ukládání výsledků zajišťuje třída `Results`. Pro přehlednost výsledků byla zavedena možnost ukládání i pro souřadnice dronu, které jsou získány přímo z letového deníku.

Třída implementuje možnosti exportu výsledků do souborů typu GPX, který je běžně používaným standardem pro ukládání a přenos GPS záznamů. V Google Earth však nabízí pouze omezenou podporu, jelikož není nativním formátem této aplikace. Proto bylo implementováno také exportování do souborů typu KML. Oba typy souborů a způsob ukládání souřadnic budou popsány v této kapitole. K jejich tvorbě je využita standardní *Python* knihovna pro práci s formátem XML, není proto vyžadována žádná externí závislost.

Dále je umožněno trajektorie osob upravit několika způsoby. Prvním z nich je odstranění osob, které nedosahují minimálního zvoleného počtu stop. Druhým je odstranění osob, které se vyskytly na kratší, než minimální stanovenou dobu (např. několik sekund). Tímto by mělo dojít k odstranění chybných identit, které často vznikají z falešně pozitivních detekcí nebo při falešně negativní re-identifikaci).

Dále mohou být stopy jedné osoby agregovány dle časového intervalu – pro GPS pozice za daný čas jsou zprůměrovány hodnoty zeměpisné šířky a délky. Takto vzniklé nové stopy pak nahrazují původní trajektorii, čímž lze odstranit přílišnou členitost trasy. Na GPS pozice výsledných stopy lze také aplikovat Kalmanův filtr pro vyhlazení trajektorie.

Výsledné zobrazení je pak programově řízeno pomocí systémového volání, kdy je cesta ke KML souboru předána jako parametr aplikaci Google Earth. Tím dojde k jejímu zobrazení

v aplikaci. Toto ovšem není možné provést se souborem GPX, který Google Earth umožňuje otevřít pouze manuálně. V tomto případě, nebo pokud je zadána neplatná cesta k aplikaci Google Earth, je soubor otevřen ve výchozí aplikaci přiřazené k danému formátu.

5.6.1 Soubory typu GPX

Formát GPX [16] vznikl za účelem sjednocení a zjednodušení sdílení dat mezi geografickými aplikacemi. Jeho aktuální verze byla stanovena v roce 2004 a je postavena na schématu XML[33], což je obecně známý a velmi rozšířený standard pro uchovávání a přenos dat nezávisle na platformě. Díky tomu je snadné s GPX soubory pracovat, pročež je využívají stovky programů a webových aplikací (např. Google Earth).

Soubor GPX může být tvořen různými strukturálními prvky[34], které většinou obsahují GPS data a další doplňující informace. Kupříkladu bod `wptType` reprezentuje jakýkoliv bod zájmu, `rteType` symbolizuje libovolnou trasu složenou ze zmíněných prvků atp. Pokud by předdefinované typy nedostačovaly potřebám, je možné formát rozšířit i o vlastní doplňky.

Pro záznam trajektorií chodců se jeví jako nejvhodnější element typu `trkType`. Skládá se ze segmentů `trksegType`. Mohou sloužit k oddělení úseků např. v případech, kdy GPS zařízení ztrácelo a znovu nacházelo signál. Segmenty uchovávají cestu jako seznam seřazených bodů `wptType`, které obsahují zeměpisnou délku a šířku. Volitelně pak mohou nést i další informace - např. nadmořskou výšku, čas navštívení, název a popis místa, atd.

5.6.2 Soubory typu KML a KMZ

Standard KML[35] vznikl jako nativní formát pro aplikaci Google Earth (resp. jejího předchůdce) za účelem přenášení nejen geografických informací, ale také doplňujících aspektů, např. různých geometrických tvarů a vizuálních stylů. Díky tomu dokáže přenášet GPS trasy, kresby a mnoho dalšího. Podobně jako formát GPX, i KML vychází z XML standardu. Formát KMZ pak představuje komprimovanou verzi KML souboru.

Osoby jsou představovány prvkem `Placemark`. Ten může obsahovat trasy, které ukládá element `gx:Track`. V něm může být obsaženo libovolné množství prvků `gx:coor`, což jsou vlastně body určené GPS souřadnicemi a volitelnou nadmořskou výškou. Ke každému bodu jde také přiřadit čas navštívení elementem `when`.

Rozdíl v ukládání výsledků oproti formátu GPX je pak možnost nastavení barvy a šířky zobrazených trajektorií. Každé osobě je tedy vygenerována a přiřazena náhodná barva, aby byly snadno rozlišitelné.

5.7 Uživatelské rozhraní a ostatní náležitosti

Jelikož práce nezobrazuje žádné grafické výsledky, ale exportuje výsledné trajektorie do souboru, který je poté zobrazen v Google Earth, nebylo vytvořeno grafické rozhraní. Uživatelské rozhraní je místo toho tvořeno konfiguračním souborem *config.py* a spouštěcím souborem *run.py*.

Možnosti nastavení jsou detailně vysvětleny v konfiguračním souboru. Princip spuštění aplikace je popsán v souboru *README.md*. Veškerý kód v práci je důkladně dokumentován

[33] Podrobnější informace lze nalézt online: www.w3schools.com/xml/xml_whatis.asp.
[34] Dokumentaci obsahující všechny prvky lze nalézt na: www.topografix.com/GPX/1/1.
[35] Oficiální dokumentace: https://developers.google.com/kml/documentation.

a přizpůsoben tak, aby uživatel mohl snadno také vytvořit vlastní spouštěcí moduly dle potřeb použití.

V rámci práce bylo vytvořeno také mnoho podpůrných funkcí a interních knihoven, např. modul pro geografické výpočty a práci s GPS, geometrické operace, pro práci s obrázky, skripty pro úpravy videozáznamů (střih, odstranění zvukové stopy, změna velikosti) atp. Tyto dílčí složky však v práci nejsou rozebrány, jelikož pro běžného uživatele nejsou přímo určeny a v případě úprav aplikace jsou dobře dokumentovány.

Kapitola 6

Testování a výsledky

V této kapitole budou popsány metody a nástroje použité při testování a dosažené výsledky jak pro celou aplikaci, tak pro experimenty s dílčími částmi. Použité názvosloví a principy vychází z informací uvedených v kapitole popisující implementaci 5, kde jsou vysvětleny jednotky, zkratky, pojmy a konvence běžně používané v následujícím textu.

Konkrétní varianty funkčních modulů nabízených v aplikaci byly implementovány převážně pro podmínky testovacího datasetu (viz 6.1.1) a obdobných situací. Ovšem jak bylo dříve popsáno, aplikace je navržena tak, aby bylo možné ji snadno přizpůsobit jiným podmínkám. Proto budou dále uvedeny a řešeny i problémy, které se v testovacím datasetu přímo neprojevily.

6.1 Testovací prostředí

Pro potřeby práce byl vytvořen dataset videozáznamů pořízených dronem. Z něj byl poté vytvořen také dataset identit pro porovnávání výsledků re-identifikačních algoritmů. Oba budou popsány v rámci této kapitoly a pro bližší představu budou uvedeny názorné ukázky. Celé datasety pak lze nalézt na přiloženém paměťovém médiu.

Dále zde budou popsány hardwarové a softwarové vlastnosti testovacích zařízení, aby bylo možné přehledně porovnávat výpočetní náročnost algoritmů a jejich vhodnost pro různé situace.

6.1.1 Testovací dataset

Dataset vytvořený pro účely této práce byl pořízen dronem DJI Spark (viz kap. 2.1.2) používajícím výchozí kameru s rozlišením 1920 × 1080 pixelů, 30 snímky za sekundu a horizontálním zorným polem 81,9 °. Obsahuje několik videozáznamů s délkou v rozmezí jednotek až desítek vteřin a zabírá převážně osoby vzdálené 10 až 50 m od objektivu z mírného nadhledu.

K nahrávkám je k dispozici také letový záznam obsahující parametry zaznamenané senzory dronu, které postačují k úspěšné lokalizaci osob vybranými metodami navrženými v této práci. V průběhu letu však dron zaznamenal několik záběrů postupně oddělených časovým rozestupem, proto bylo nutné letový deník rozdělit na úseky dle údajů o stavu kamery. Počátek logu a reálného nahrávání tak nemusí být přesně synchronizován, což je třeba zohlednit při práci s datasetem (možná chyba, více v kap. 6.2.2).

Všechny osoby, jejichž tváře jsou na snímcích rozpoznatelné, byly s využitím záznamů obeznámeny předem a souhlasily s jejich zveřejněním. Natáčení probíhalo v dostatečné

Obrázek 6.1: Náhled na oblast natáčení. Modrý obdélník na levé polovině snímku označuje odhad rozmezí výskytu dronu. Červeně je přibližně uprostřed snímku vyznačena oblast, ve které se pohybovaly sledované osoby. Vpravo je pak uvedena orientace snímku (sever symbolizuje písmeno N, z angl. „North“).

vzdálenosti od obytných zón. Záznamy byly dle potřeby upraveny pomocí skriptů dostupných v rámci práce (viz 5.7), proběhlo např. odstranění zvukové stopy či zkrácení některých souborů.

V datasetu se vyskytuje několik osob. Pro lepší orientaci je na obr. 6.1 vidět satelitní náhled na místo natáčení, přičemž napříč ukázkami se dron i osoby pohybují přibližně v zakreslených rozmezích. Většina záznamů v datasetu je pořízena při pohledu dronu směrem na východ. Konkrétní ukázky snímků lze vidět např. v kap. 6.3.

Chodci se pohybují po šikmé stráni. Jejich nadmořská výška se tedy v průběhu času mění. Úpatí stráně, tedy nejnižší bod celého místa, se nachází u levého okraje snímku a směrem doprava nadmořská výška roste. Přibližný výškový profil sledované oblasti je vyznačen na obr. 6.2a.

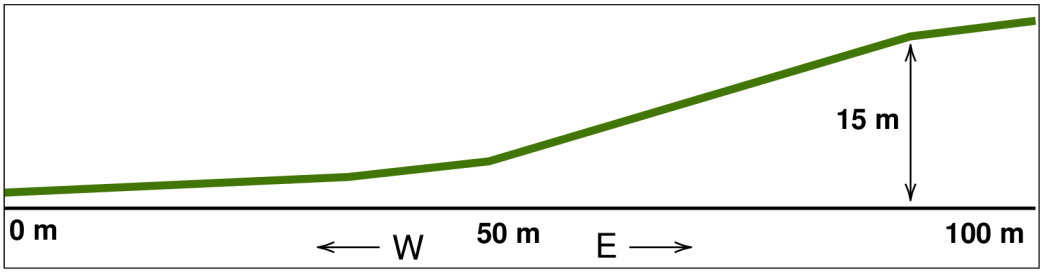
6.1.2 Sada identit testovacího datasetu

Pro potřeby vyhodnocení re-identifikačních nástrojů vznikla v rámci práce sada identit testovacího datasetu (dále označována zkr. SITD). Ve 100 snímcích náhodně vybraných napříč celým datasetem byly detekovány osoby detektorem „Faster R-CNN with NASNet-A“ (viz 6.2.3). Jejich náhledy poté byly manuálně roztříděny a došlo k odstranění chybných či neúplných detekcí.

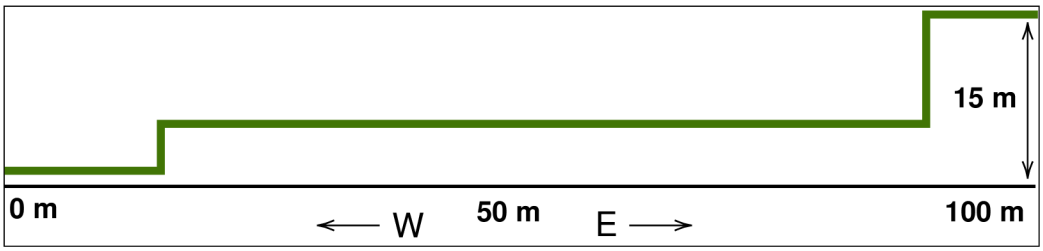
Ve výsledku SITD sestává z 6 různých identit tvořených celkem 384 obrázky o rozměrech v rozmezí 19x50 až 115x241 pixelů. Sledované osoby jsou zaznamenány v různých pózách, vzdálenostech a úhlech pohledu. Názornou ukázku lze vidět na obr. 6.3. SITD je dostupný na přiloženém paměťovém médiu ve složce *data*.

6.1.3 Testovací hardware

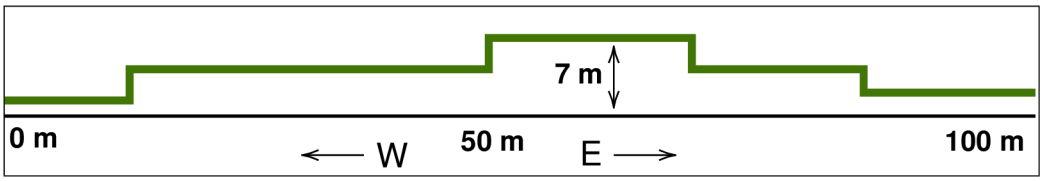
Testování aplikace probíhalo pod vícero operačními systémy s různým hardwarovým vybavením:



(a) Skutečný výškový profil (na základě reálného pozorování a porovnání s Google Earth).



(b) Výškový profil dle digitální výškové databáze „The TanDEM-X 90m Digital Elevation Model“.



(c) Výškový profil dle digitální výškové databáze „Copernicus Land Monitoring Service – EU-DEM“.

Obrázek 6.2: Porovnání přibližných výškových profilů západo-východního řezu místa natáčení jak je zachyceno na obr. 6.1. Profily byly získány pomocí uvedených metod (více informací v kap. 5.4.2 a 6.2.4). Na obrázcích je zeleně vyznačen profil povrchu. Černou barvou je pak uveden směr západu (W) a východu (E), přibližná horizontální vzdálenost a převýšení signifikantních bodů.

1. PC s Windows 10 (64bit, verze 1909), procesor i5-2500K (4 jádra, takt 4 GHz), 16 GB operační paměti, grafická karta NVIDIA RTX 2060 (6 GB grafické paměti).

2. MacBook Pro s operačním systémem High Sierra (64 bit, verze 10.13.6), procesor i7-7700HQ (4 jádra, takt 3,8 GHz), 16 GB operační paměti, grafická karta AMD Radeon Pro 555 (2 GB operační paměti).

3. Virtuální počítač s Ubuntu (64 bit, verze 18.04), procesor virtualizovaný (2 jádra, takt 3 GHz), 8 GB operační paměti, grafická karta virtualizovaná (128 MB grafické paměti).

Základní aplikace funguje na všech uvedených platformách za předpokladu, že jsou k dispozici požadované prerekvizity (např. interpret jazyka Python 3.7 atp.). Pro měření náročnosti operací a jejich porovnávání byl v naprosté většině případů používán systém číslo 1, jelikož má k dispozici aktuální grafickou kartu značky NVIDIA, která je podporována většinou nástrojů strojového učení, které urychlení pomocí grafických jader nabízejí. Pokud

Obrázek 6.3: Ukázka snímků osob z SITD.

byl použit jiný systém, bude toto explicitně uvedeno na daném místě. S využitím grafické karty je však spojeno také spoustu omezení a chyb, např. nefunguje při použití více procesů atd.

6.2 Experimenty a vylepšení jednotlivých částí aplikace

V této kapitole budou rozebrány experimenty s dílčími částmi aplikace a jejich testování. Dosažené výsledky budou zhodnoceny jak v kontextu testovacího datasetu, tak z obecného hlediska.

Nalezené problémy budou vysvětleny s použitím názorných ukázek. Dále budou navržena jejich řešení a možná vylepšení jednotlivých částí algoritmu. Výsledky implementovaných úprav budou také otestovány a zhodnoceny.

6.2.1 Předzpracování obrazu

Bylo vyzkoušeno několik variant obrazových úprav se záměrem vylepšit detekci osob odstraněním šumu nebo úpravami histogramu (podrobněji popsány v kap. 5.2.2). Výsledky byly hodnoceny na základě úspěšnosti detektorů osob na takto upravených snímcích, konkrétně pomocí hladiny jistoty, kterou neuronové sítě udávají pro jednotlivé nalezené objekty.

Porovnávaly se výsledky upravených vstupů vůči originálu pro různé verze použitých detektorů (viz 6.2.3). Hodnotilo se množství chybějících vs. získaných validních objektů, počet vylepšených vs. snížených hodnot jistoty detekce a její celkový přírůstek vs. úbytek. Byly testovány desítky kombinací detektorů a parametrů pro níže popsané metody, v tab. 6.1 s konkrétními hodnotami jsou tedy pro názornost uvedeny pouze některé z dále zmíněných ukázek.

Odstranění šumu

Při použití metod pro odstranění šumu nebylo pozorováno výrazné zlepšení, ba spíše naopak, a to u všech typů použitých detektorů. Ve většině sledovaných případů se sice hladina jistoty detekcí neměnila, relativně často však došlo k jejímu snížení, čili ke zhoršení. Občasně pak může dojít i k vynechání některého subjektu.

Method	Parameters	Miss	Extra	Higher (sum)	Lower (sum)
Gaussian Blur	ksize 5x5	8	1	21 (1,60)	56 (-3,02)
Bilateral Filter	d 7, sigmaColor 30, sigmaSpace 30	6	1	27 (1,12)	49 (-1,68)
CLAHE	clipLimit 1, tileGridSize 7x7	1	1	33 (1,57)	40 (-0,99)

Tabulka 6.1: Ukázka změn v detekci osob po provedení obrazových úprav. Testovací sada sestávala z 50 náhodně vybraných snímků napříč datasetem. Uvedené údaje byly získány při použití detektoru *Faster R-CNN with NASNet-A featurization* (viz 6.2.3), který v neupravených záběrech určil 138 osob. Vysvětlení parametrů metod lze najít v návodu knihovny OpenCV [4]. Sloupce miss/extra, udávají počet chybějících, resp. nově získaných validních nálezů oproti originálnímu obrazu. Sloupce higher/lower udávají kolik objektů mělo vylepšenou, resp. zhoršenou jistotu detekce a celkový součet všech těchto rozdílů (běžná hodnota jednoho nálezu se pohybuje v intervalu 0 až 1).

Při aplikaci Gaussova rozmazání lze zhoršení předpokládat, jelikož i pro lidské oko je snazší identifikovat objekty v ostrém obraze. Toto se také potvrdilo až na jeden ze 4 zkoušených detektorů, a to i při volbě velmi jemných parametrů filtru (např. maska o velikosti 3x3 pixely). Při výraznějším filtrování pak detektory podávaly úměrně zhoršující se výsledky, detailnější popis v následujícím odstavci se tedy vztahuje pouze na mírnější nastavení.

U detektoru *Faster R-CNN with Inception v2*, který za normálních podmínek produkoval více falešně pozitivních nálezů, byl pozorován jejich mírný pokles. Přibližně v polovině případů, kdy se jistota detekce změnila, došlo k jejímu zvýšení (převážně u vzdálenějších objektů). Ve druhé polovině však spolehlivost klesala. U ostatních detektorů však bylo pozorováno zlepšení jen zřídka, zatímco ke zhoršení docházelo mnohem častěji, místy se vyskytly i zcela chybějící detekce. Typickou ukázku lze vidět na obr. 6.4. Gaussův filtr se tak jeví jako nevyhovující.

Po aplikaci bilaterálního filtru, který uchovává či dokonce zvýrazňuje hrany, z lidského pohledu kvalita kontur osob zůstává zachována a rozmazání pozadí navíc odstraní nepodstatné informace. Výsledky potvrdily předpoklad a byly oproti Gaussovu filtru mírně lepší. Ukázka se nachází na obr. 6.4, kde je vidět, že i při výrazném rozmazání pozadí zůstávají osoby vizuálně zachovány lépe, než při jemné konfiguraci Gaussova filtru.

Při použití mírného nastavení byly změny víceméně zanedbatelné. Při použití výraznějších parametrů pak byly rozdíly patrné a v mnoha případech vedly i ke zlepšení detekce, ovšem přibližně stejným podílem bylo zastoupeno i zhoršení, avšak méně často se vyskytlo úplné vynechání validní detekce (dle typu detektoru a nastavení filtru).

Obecně je známo, že konvoluční neuronové sítě mohou sice ze své podstaty (viz 3.4) být náchylné na extrémní rozdíly intenzity u několika málo pixelů. Proto se jejich filtrování jeví jako možné řešení, avšak při praktických experimentech na testovacím datasetu s uvedenými algoritmy se ukázalo jako nejednoznačné. Výsledky však mohou být ovlivněny faktem,

Obrázek 6.4: Porovnání úspěšnosti detekce v závislosti na použitých filtrech šumu. Uvedená čísla znamenají hladinu jistoty detekce označené osoby. Shora jdoucí části jsou: a) původní obraz, b) jemná varianta Gaussova rozmazání, c) výrazná verze bilaterálního filtru.

že v testovacím datasetu dosahuje kvalita obrazu poměrně vysoké úrovně (dostatečné a nepřehnané osvětlení, kvalita kamery atp.). Tedy přestože odstranění šumu zde není velkým přínosem, v jiných záznamech by mohlo pomoci.

Úprava kontrastu

K úpravě kontrastu byla využita metoda CLAHE (*Contrast Limited Adaptive Histogram Equalization*), která provádí lokální adaptivní ekvalizaci histogramu (více informací v kap. 5.2.2). Vesměs vedla k lepším výsledkům než filtrování šumu. Příklad naměřených hodnot lze vidět v tab. 6.1.

U detektoru *Faster R-CNN with Inception v2* ve většině konfigurací pozitivní efekt znatelně převážil negativa. CLAHE se tak jeví jako dobrá volba pro tento model. U ostatních případů byl efekt při použití jemných variant parametrů víceméně neutrální. Použití CLAHE tedy na tomto datasetu nevede k významnějším změnám. Avšak je pravděpodobné, že v případech špatného osvětlení (např. natáčení s tmavými oblastmi či na příliš světlých plochách) by mohla metoda vést k výrazným zlepšením.

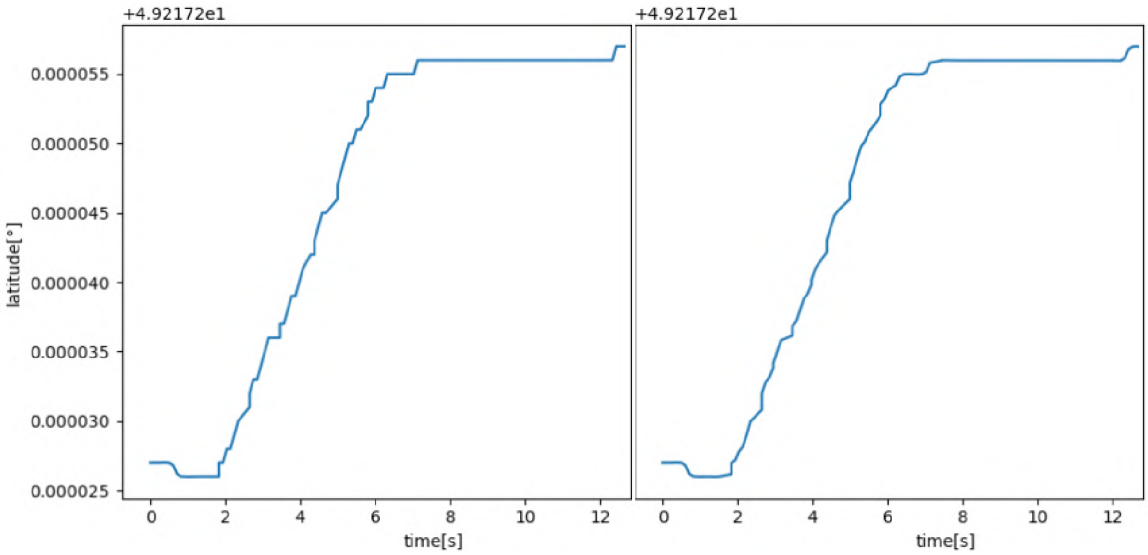
6.2.2 Vliv nepřesností letovém záznamu

Letový záznam je s videonahrávkou synchronizován dle záznamu o aktivaci videokamery. Prvním problém tedy nastává v případě, že aktivace kamery není přesně zaznamenána v okamžiku, kdy se reálně spustí natáčení. V testovacím datasetu bylo vypozorováno, že u

několika záznamů se tak začátek videozáznamu s prvním zapsaným údajem může rozcházet o 100 ms až 300 ms. I tak krátký okamžik však může vést k odchylkám určených pozic – např. i malý rozdíl v natočení dronu během této doby může ve výsledku významně odchýlit pozice osob (čím vzdálenější, tím více jsou ovlivněny).

Tento problém je možné řešit manuálním nastavením zpoždění/předbíhání záznamu, což ovšem vyžaduje bližší analýzu uživatelem. Automaticky by bylo možné ho řešit detekcí posunu všech osob stejným směrem a kompenzovat takovéto plošné translace změnou do protipohybu. U každé osoby se však zmíněná chyba projevuje odlišně - pokud se např. chodec pohybuje v protisměru odchylky, bude jeho posun menší a kompenzace pak u něj přílišná, nebo u ostatních osob nedostatečná. V praxi by dále taková úprava mohla znamenat kompenzaci i v momentech, kdy se všechny osoby pohybují stejným směrem přirozenou chůzí (a když by letový deník udával správné údaje). Proto kompenzace v práci nebyla implementována.

Potenciálním zdrojem chyb je též možná nepřesnost letových senzorů (viz 2.2). Na údaje z letového deníku je proto aplikováno vyhlazení pomocí Kalmanova filtru. Názorná ukázka je vidět na obr. 6.5.



Obrázek 6.5: Ukázka aplikace Kalmanova filtru na zeměpisnou šířku GPS pozic dronu během videozáznamu z testovací sady. Vlevo lze vidět původní údaje z letového záznamu, vpravo po úpravě. Na ose y je hodnota přírůstku zeměpisné šířky v desetinných stupních oproti 49,2172 °, osa x udává čas. Je možné pozorovat mírné vyhlazení křivky a plynulejší průběh skokových změn.

Některé chyby však ani takto nemohou být odstraněny. Dron např. používá stabilizaci kamery, takže hodnoty roll a pitch jsou mimo úseky, kdy probíhá výraznější otáčení kolem daných os, udávány jako přesně nulové. Aplikace Kalmanova filtru ani jiné úpravy se tedy neprojeví. V obraze je však možné pozorovat, že chvění probíhá v těchto směrech, i pokud jsou záznamy neměnné. Velikost odchylky je jen minimální (např. méně než 1 °) – i takové hodnoty však vnáší chyby.

Další problém byl pozorován v případě chybně uvedených úhlů natočení dronu – jak bylo uvedeno dříve, v měření hardwarových senzorů se při letu postupně kumulují nepřesnosti, které by měl částečně odstranit Kalmanův filtr. Na testovacím datasetu však byly

pozorovány odchylky udávaného natočení oproti reálné situaci po 20 minutách letu i více než 10 °i při použití filtru. To je dáno jednak faktem, že filtr je použit jen na daný úsek letového deníku a ne na celý průběh (jelikož záznamy byly natáčeny postupně s přerušeními), jednak faktem, že ani Kalmanův filtr nedokáže chyby způsobené počasím a měřením odstranit kompletně, ale jen částečně. Proto je zavedena možnost manuální kompenzace těchto údajů (ukázka viz kap. 6.3.4).

Další potenciální chybovost vnáší problém nedostatečně častého zaznamenání údajů, které probíhá 10× za sekundu. Pokud má videozáznam vyšší frekvenci snímků (např. 30 za sekundu), u některých nejsou použity dostatečně aktuální údaje – ve výsledku se však i takto malé rozdíly mohou viditelně promítnout do pozic vzdálenějších osob.

Problém by mohlo být možné řešit pomocí prokládání údajů v kratších intervalech, např. lineárním proložením či předpovědí Kalmanova filtru. V některých úsecích by pravděpodobně mohlo nastat mírné zpřesnění výsledků. V tomto případě by však bylo ideální znát i manuální vstupy ovladače dronu, aby mohla být lépe reflektována dynamika změn. Tyto údaje však v testovacím datasetu nejsou k dispozici.

6.2.3 Detekce osob

K detekci osob byly testovány celkem 4 modely neuronových sítí. Nejprve byl vyzkoušen nástroj *Open Pose*, který je obecně považován za poměrně robustní řešení v běžné detekci osob. Vážou se k němu ovšem některá klíčová omezení. Proto byl následně implementován modul pro využívání sítí dostupných z oficiální nabídky modelové zoo *TensorFlow*, který tyto chyby řeší.

Modely nejsou vybírány pouze na základě metriky mAP (z jejich trénovacích datasetů), ale bylo u nich ověřováno především to, jak se chovají v testovací sadě použité v této práci a zda jsou jejich výsledky vhodné pro další použití při re-identifikaci osob. V této kapitole jsou popsány problémy, které se při testování detekce chodců vyskytly, a jejich případná řešení.

Detekce s využitím Open Pose

Open Pose úspěšně zvládá detekci chodců z výšky a při pohledu z různých úhlů. První problém však nastává u vzdálenějších osob, kdy síť lépe pracuje na snímcích s vyšším rozlišením. Zde prudce narůstá její výpočetní náročnost.

Na testovací sestavě se časy detekce jednoho snímku pomocí procesoru pohybovaly při rozlišení 1280 × 720 pixelů na testovací sestavě č. 1 (viz 6.1.3) v řádech jednotek minut, při použití grafické karty se proces urychlil na jednotky sekund, ovšem kvůli limitu grafické paměti neumožnil větší rozlišení, než 640p. Z tohoto důvodu docházelo k méně přesným detekcím vzdálených osob. Uvedené hodnoty navíc byly dosaženy pouze s využitím předkompilované binární verze nástroje, která je platformě omezená. Nativní implementace v Pythonu dosahovala ještě znatelně horších hodnot. Řešením by tedy bylo extrémní navýšení výkonu/kapacity.

Další problém nastává ve chvíli, kdy je na dvou těsně po sobě jdoucích snímcích rozdílně rozpoznána např. končetina – výsledný bounding box pro stejného člověka se pak může velmi podstatně lišit. Příklad lze vidět na obr. 6.6. Toto způsobuje výrazné problémy při re-identifikaci osob.

Uvedenému problému je částečně předcházeno pomocí stanovení základních částí těla potřebných pro úspěšnou detekci (např. vyžadovat alespoň trup, hlavu a minimálně jednu dolní končetinu). Ovšem toto řešení je pouze částečné, jelikož každá přidaná restrikce může

Obrázek 6.6: Ukázka možných rozdílů bounding boxů stejné osoby v těsně po sobě jdoucích snímcích při použití *Open Pose* bez vyžadování klíčových částí. Vlevo byla správně detekována celá osoba. Vpravo nebyla nalezena pravá část těla (vč. ramena a boku), chodidla a levá ruka.

vést k úbytku detekcí – kupříkladu vyžadování obou boků a ramen vede ke stabilizaci výsledku, avšak vyřadí všechny osoby otočené bokem ke kameře, přestože jsou jinak zřetelně viditelné.

Detekce pomocí modelů z TensorFlow zoo

Kvůli klíčovým nedostatkům *Open Pose* bylo implementována možnost využití neuronových sítí z *TensorFlow* zoo. Ty nemají problém s přílišně se měnící velikostí bounding boxů a mohou dosahovat velkých rychlostí. Celkem byly testovány 3 varianty modelů, které byly testovány na datasetu COCO[1]. Konkrétně se jedná o:

1. SSD with Resnet 50 v1 FPN feature extractor[2],
2. Faster R-CNN with Inception v2[3],
3. Faster R-CNN with NASNet-A featurization[4].

Model č. 1 dosahuje na datasetu COCO hodnocení mAP 35. Umělé neuronové sítě typu SSD[5] patří k nejrychlejším, přesto přesným architekturám. Na testovacím datasetu však bylo dosaženo poměrně nízkého počtu správných detekcí v porovnání s *Open Pose*.

Především pak síť špatně detekovala vzdálenější objekty, pročež se jeví jako nevhodná pro tento dataset a dále nebyla používána. Toto pozorování bylo následně potvrzeno článkem [24], kde je možné vidět, že SSD detektory mají obecně problémy s detekcí menších objektů (čili vzdálenější osoby v tomto případě).

Model č. 2 dosahuje na datasetu COCO hodnocení mAP 30, přičemž jeho rychlost je také extrémně dobrá. Na testovacím hardwaru trvá zpracování jednoho snímku v rozlišení 1920 × 1080 pixelů přibližně 100 ms s použitím grafické karty, resp. 5× až 10× déle s procesorem.

[1] Dostupné z: http://cocodataset.org.

[2] Detaily konfigurace viz: https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/ssd_resnet50_v1_fpn_shared_box_predictor_640x640_coco14_sync.config.

[3] Detaily konfigurace viz: https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/faster_rcnn_inception_v2_coco.config.

[4] Detaily konfigurace viz: https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/faster_rcnn_nas_coco.config.

[5] Zkr. angl. „Single Shot Detector“ je architektura, která využívá „přímé“ detekce. Na rozdíl od jiných známých typů neuronových sítí, kde se často generují oblasti zájmu a až poté se v nich hledají objekty.

Na testovacím datasetu dosahuje dobrých výsledků v počtu úspěšných detekcí. Problém však vyvstává s velkým množstvím falešně pozitivních detekcí. Všechny uvedené modely sice udávají hladinu jistoty detekce pro každý objekt v rozmezí (0;1>, kterou je možné využít pro odfiltrování těchto nálezů. Avšak při přísnějších podmínkách síť ignoruje velké množství osob, zatímco při volnějších vrací často pro jednu osobu i vícero duplicitních detekcí.

Pouhé nastavení hranice jistoty tedy nestačí k dosažení optimálních výsledků. Tento problém je proto řešen přidáním limitu maximálního překrytí nalezených osob. V detektoru je tak možné nastavit větší toleranci jistoty detekce, čímž dojde k nálezu více objektů (včetně duplicit), které jsou následně odfiltrovány – z každé skupiny vzájemných překryvů je vždy ponechán pouze 1 výsledek dle nejvyšší hladiny jistoty.

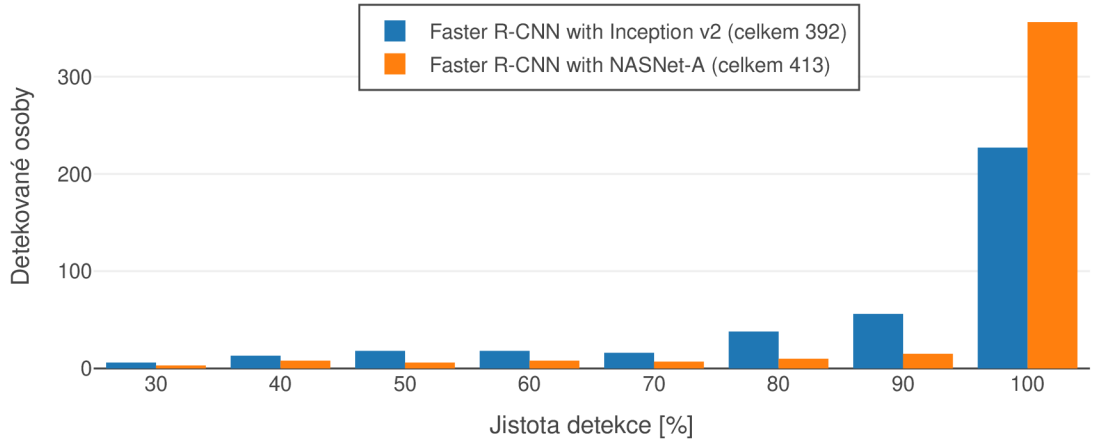
Obrázek 6.7: Ukázka možných problémů v detekci. Výsledky byly získány s použitím modelu „Faster R-CNN with Inception v2“. Čísla udávají jistotu detekce příslušného objektu. Nahoře je požadována min. jistota 0,7 bez zákazu překryvů. Uprostřed je vyžadována min. jistota 0,4 bez zákazu překryvů. Dole je vyžadována min. jistota 0,3 se zákazem překryvů nad 50 %. Černé rámečky označují veškeré nálezy, zatímco červené ukazují pouze ty, které dosahují požadované jistoty detekce. Zeleně jsou pak označeny osoby, které dosahují požadované jistoty a zároveň mají nejvyšší hodnotu ze všech lokálních překryvů.

Příklad lze vidět na obr. 6.7. S použitím min. limitu jistoty 0,7 na vrchním snímku byly nalezeny 3 validní osoby (zbytek nedosahuje požadované hodnoty). Aby byla uznána za platnou další osoba, musel být limit snížen na hodnotu 0,4 v prostředním snímku. Tím ovšem přibyly navíc ještě 2 nechtěné duplicitní nálezy. Řešení lze vidět ve spodním snímku, kde byl použit limit min. jistoty 0,3 a zároveň filtr pro odstranění překrývajících se bounding boxů. Tím byly duplicity odstraněny, navíc také použití nižšího limitu umožnilo odhalení další osoby.

Hodnota povoleného překryvu může odstranit i validní osoby v okamžiku, kdy kolem sebe prochází a míjí se – to ovšem nevadí. V případech většího překryvu by totiž mohlo dojít k záměně identit (např. při překryvu 50 % už nelze říct, která osoba je která). Filtr překryvů tedy předchází oběma těmto problémům zároveň.

Model č. 3 dosahuje na datasetu COCO hodnocení mAP 43, z uvedených modelů tedy nejlepší výsledek. To je ovšem vykoupeno několikanásobně vyšší výpočetní náročností. Na testovacím hardwaru trvá zpracování jednoho snímku v rozlišení 1920×1080 pixelů přibližně 1–1,5 s při využití grafické karty, resp. 20× déle s procesorem.

Na testovacím datasetu však podává nejlepší výsledky ze všech uvedených detektorů v naprosté většině případů. Srovnání výsledků modelů 2 a 3 (ostatní varianty jsou pro nedostatečnost vynechány) lze vidět na obr. 6.8. Na základě těchto údajů je následně možné volit minimální vyžadovanou hladinu jistoty.



Obrázek 6.8: Porovnání počtu validních výsledků dle jistoty jejich detekce (hodnoty jistoty zaokrouhleny na desítky) pro nastínění efektu při omezení její minimální hodnoty. Porovnání proběhlo na 120 náhodně vybraných snímcích z testovacího datasetu. Byl aplikován filtr pro odstranění překrývajících se detekcí. Detektor „Faster R-CNN with Inception v2“ celkem našel 392 validních osob. Model „Faster R-CNN with NASNet-A“ zajistil 413 platných výsledků.

Jistota detekcí modelu č. 3 se většinou pohybuje blízko hodnoty 1 a duplicitní výsledky téměř nevytváří. Občas navrací falešně pozitivní výsledky pro nežádoucí objekty, ovšem nemá problém např. se správným ignorováním stínu osob (což je např. u předchozí sítě velký problém). Ze všech testovaných detektorů se tak jeví jako nejlepší volba.

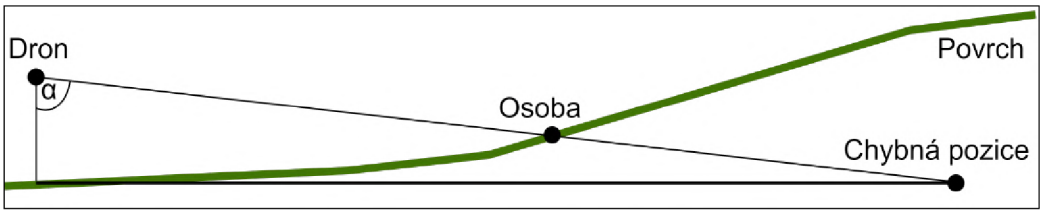
6.2.4 Výpočet GPS pozice chodců

K určení GPS pozice sledovaných chodců byly vyzkoušeny 2 navržené postupy, konkrétně metoda z kap. 4.1.4 „určení vzdálenosti objektu na rovném povrchu“ (viz) a metoda z kap. 4.1.5 „určení pozice objektu s využitím nadmořské výšky“. První z nich na většině testovacího datasetu podala naprosto nevyhovující výsledky. Druhá tyto nedostatky řeší, ovšem vyžaduje databázi nadmořských výšek povrchu v daném místě, s čímž se pojí další komplikace. V této kapitole budou popsány zmíněné problémy a jejich případná řešení.

Testování metody pro určení pozice objektu na rovném povrchu

K určení pozice chodců byla nejprve testována metoda navržená v kap. 4.1.4. Ta nevyžaduje externí závislosti a je velmi rychlá, ovšem lze ji použít jen v případě, že se sledované osoby pohybují po rovné ploše. V testovacím datasetu proto přijatelně fungovala pouze ve výjimečných případech, kdy se dron společně se sledovanými subjekty pohybuje na rovné ploše před úpatím stráně.

Naopak např. v záznamu 6.19 se osoby pohybují po stráni a dron je sleduje zespodu (ze zmíněné rovné části), a proto zde nastává extrémní nepřesnost, která prakticky kompletně znehodnocuje výsledky. Informační grafiku, která vysvětluje tento problém, lze vidět na obr. 6.9. Tento problém vede k nedostatečným výsledkům na většině testovacího datasetu, pročež byla metoda shledána nedostatečnou a nahrazena.



Obrázek 6.9: Ukázka chybného určení GPS pozice osoby při neznalosti nadmořské výšky (zakresleno do výškového profilu oblasti, viz obr. 6.2a). Dron předpokládá, že osoba se vyskytuje ve stejné nadmořské výšce, jako povrch pod ním. Pozice chodců na stráni jsou proto určeny výrazně dále od dronu, než kde se reálně nachází. Čím větší bude úhel α pro pozorovaný objekt, tím dále bude určen – při hodnotě 90 ° a více pak pozici nelze určit vůbec.

Určení pozice metodou využívající nadmořské výšky a problémy s tím spojené

Metoda výpočtu GPS pozice osob s využitím nadmořské výšky 5.4.4 vyžaduje ke své práci databázi nadmořské výšky povrchu v dané oblasti. Její přesnost tak závisí na základních faktorech (stejně jako předchozí metoda) - např. přesnost údajů v letovém záznamu, kvalita detekce osob, natočení snímku (odhad chodidel je předpokládán na spodní hraně bounding boxu). Především ale záleží na kvalitě použité databáze výšek.

Databáze nadmořských výšek vychází z DEM (*Digital Elevation Model*), což jsou data pro dané území s určitým rozlišením a přesností. V závislosti na těchto vlastnostech pak nastává problém, pokud DEM nepopisuje reálný reliéf povrchu dostatečně přesně (příklady budou popsány dále v této kapitole).

Potenciálním problémem metody je dále předpoklad, že se chodci pohybují po povrchu. Pokud se tedy sledovaná osoba nachází např. na budově, výsledek bude podobně nepřesný, jako v případě předchozí metody. Obecně by však chodci měli v naprosté většině putovat přímo po zemském povrchu a v testovacím datasetu tomu tak skutečně je, tento problém tedy nebyl pozorován.

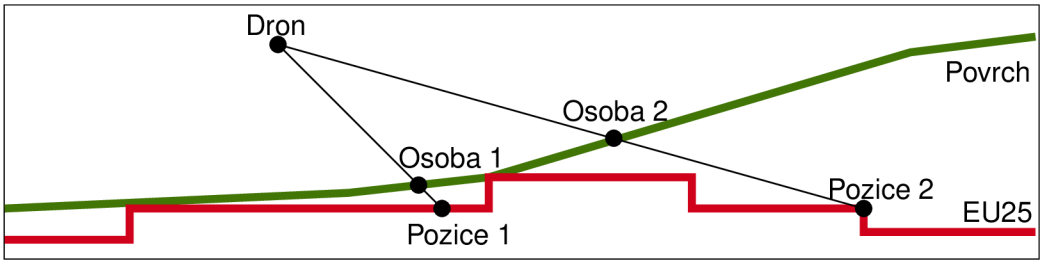
V rámci práce byly testovány 2 verze DEM (více informací v kap. 5.4.2):

1. „Copernicus Land Monitoring Service – EU-DEM“ s horizontálním rozlišením přibližně 25 m (dále označován EU25),

2. „The TanDEM-X 90m Digital Elevation Model“ s horizontálním rozlišení 90 m (dále označován TDM90).

Nejprve byl vyzkoušen model EU25, který slibuje vysokou přesnost a rozlišení, což je vhodné pro kratší vzdálenosti na testovacím datasetu. Výsledky ovšem na velké části záběrů nesplnily očekávání a obsahovaly mnohem větší chybu, než by se dalo u DEM s takovýmto rozlišením očekávat.

Následnou podrobnější analýzou výškového profilu v testovací oblasti bylo zjištěno, že v EU25 se právě v místě natáčení vyskytuje chyba/nepřesnost – výškový profil tak odpovídá reálnému prostředí pouze na západní polovině sledované oblasti (nižší část svahu), jak lze vidět na obr. 6.2c. Osoby pohybující se východněji (vyšší část stráně) jsou pak lokalizovány nepřesně. Důvod je znázorněn na obr. 6.10.



Obrázek 6.10: Ukázka chybného určení GPS pozice osoby při použití DEM EU25. Zeleně je zakreslen skutečný výškový profil, červeně EU25 (viz 6.2). Skutečná poloha osob se promítne do vyznačených pozic. V prvním případě se jedná o poměrně přesný výsledek. U druhé osoby nepřesnost prudce narůstá kvůli chybě modelu povrchu EU25 v dané oblasti.

TDM90 nenabízí tak detailní rozlišení, ovšem na testovacím datasetu byly získány lepší výsledky převážně pro osoby pohybující se na východní polovině nahrávací plochy. Podstatu výškového profilu v místě totiž vystihuje věrněji, i když ne příliš přesně. Rozdíly lze vidět na obr. 6.2. Celkové byl TDM90 s EU25 na testovacím datasetu potvrzen jako dostačující ve většině případů, přestože vnáší výrazné chyby (více výsledků lze vidět v kap. 6.3).

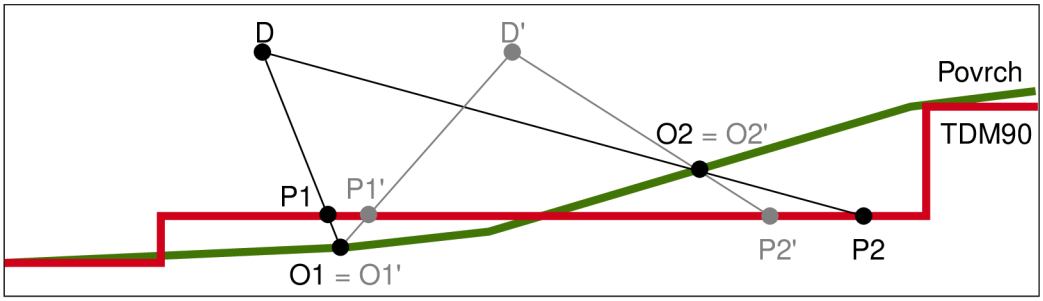
Nedostatečná přesnost se pak může ve výsledných trajektoriích projevovat posunutím oproti reálné lokaci, nebo může způsobit další zkreslující efekty. Např. na obr. 6.11 je ukázáno, jak se mění výsledná pozice osoby, přestože člověk stojí na místě a hýbe se pouze dron. Statická osoba se pak zdánlivě pohybuje společně s dronem, nebo do protipohybu.

Tento problém bohužel nelze řešit např. zohledněním pohybu dronu v trasách všech osob. Každý člověk se totiž může nacházet v lokaci s jiným rozdílem nadmořských výšek, tudíž není jasné, jakým směrem pohyb kompenzovat a jak moc.

Dalším problémem nedostatečného rozlišení a přítomnosti velkých skoků/hran v DEM je sdružování osob v pásmech do stejné pozice. Ukázku lze vidět na obr. 6.12. V oblasti testovacího datasetu se takovéto schodovité skoky v obou testovaných DEM vyskytují. Jejich linie jde přibližně od jihu k severu. Osoby v odpovídajícím pásmu reálného pohledu se tak promítnou do této linie, tzn. jejich GPS pozice se nebude měnit směrem na východ/západ (ale pouze severo-jižně), dokud nepřekonají hranice pásma.

Absolutní hodnota nadmořských výšek v TDM90 se napříč celým modelem pohybuje asi o 40m výše, než je reálný povrch. To je ovšem v implementaci ošetřeno tím, že výpočet je vztažen k relativnímu reliéfu povrchu a ne konkrétním hodnotám. Tento problém tedy výsledky nijak neovlivnil.

Nezávisle na DEM pak může nastat situace, že sledované osoby jsou příliš vzdálené od dronu a uživatel nemá zájem je sledovat nebo je zřejmé, že se jedná o chybu – takový problém může nastat v případu osoby 2 na obr. 6.10. Např. pokud se sledované osoby pohybují ve vzdálenosti 50 m, ale znázorněnou chybou se pozice promítne do 300 m vzdáleného bodu,



Obrázek 6.11: Při nedostatečné přesnosti DEM se sledovaná osoba se může jevit jako pohybující se, přestože stojí na místě. Na obrázku jsou zachyceny 2 okamžiky v čase, mezi kterými došlo pouze k pohybu dronu. D značí dron, $O1, O2$ značí reálnou pozici osoby, $P1, P2$ značí určenou pozici osoby při použití DEM TDM90 – to vše v prvním časovém okamžiku. Verze těchto proměnných, které jsou vykreslené šedou barvou a označené čarou (apostrofem), symbolizují aktualizované údaje (druhý časový okamžik). Zeleně je zakreslen skutečný výškový profil, červeně TDM90 (viz 6.2). Lze pozorovat, že osoba 1 se ve výsledku zobrazí jako po vykonání pohybu směrem doprava, zatímco u osoby 2 vznikne iluze pohybu vlevo.

je takový výsledek zřejmě chybný. K řešení této situace je zavedeno omezení maximální vzdálenosti sledované osoby od dronu – vzdálenější osoby budou ignorovány.

6.2.5 Re-identifikace osob a její vyhodnocení

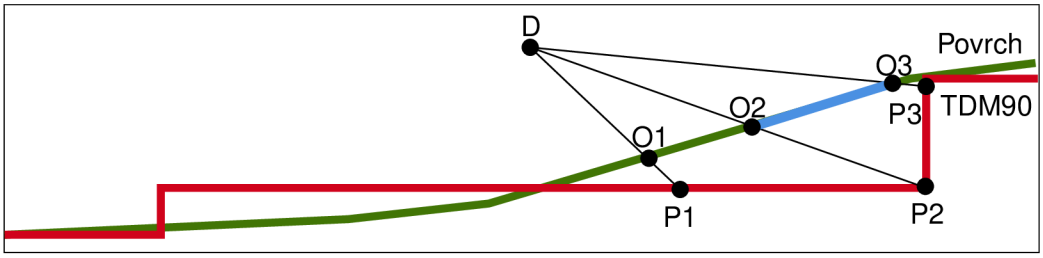
Pro re-identifikaci osob bylo vyzkoušeno několik přístupů. Nejprve budou obecně popsány z hlediska použití a poté budou uvedeny metody pro jejich hodnocení společně s dosaženými výsledky. Pro extrakci rysů byly využity 2 základní principy (více informací v kap. 5.5). U každého je uvedena také přibližná výpočetní doba extrakce na testovacím hardwaru pro snímky osob o rozměrech 128x256 pixelů (přibližně odpovídá horní hranici velikosti osob v testovacím datasetu):

1. Počítání histogramů – jedná se o základní techniku analýzy obrazu. Rysy jsou vypočteny za méně než 1 ms (výpočet probíhá na procesoru). Metoda je vhodná spíše do méně proměnlivého prostředí a je silně ovlivnitelná mnoha faktory (osvětlení, pozadí, úhel pohledu atp.).

2. AlignedReID [52] – přístup využívá neuronovou síť. Čím větší detail osoby je k dispozici, tím lépe algoritmus pracuje (na rozdíl od histogramů, které z tohoto příliš nezískají). Extrakce příznaků osoby pomocí procesoru trvá přibližně 0,5 s, zatímco s použitím grafické karty klesne doba na 30 ms. Jedná se tedy o výpočetně náročnější metodu, ale dosahuje „state of the art“ výsledků v re-identifikaci osob (viz 4.2).

Nad popsanými principy jsou postaveny moduly `HistogramTracker` a `AlignedReIdTracker` (resp. další jejich varianty), které umožňují získané příznaky porovnávat různými přístupy, s různým nastavením tolerance atp. Proto je možné např. používat bezkontextový přístup jako čisté porovnání extrahovaných rysů, nebo naopak kontextově porovnávat i pozici osob v obraze, GPS pozici, čas snímku, počet osob atp.

V práci je kladen důraz na vyhodnocení jednoznačnosti trajektorií chodců, proto bylo navrženo několik metod pro vyhodnocení výsledků re-identifikace osob:



Obrázek 6.12: Při schodovitých skocích v DEM se sledované osoby v daném pásmu promítnou do stejných výsledných souřadnic. D značí dron, $O1, O2, O3$ značí reálné pozice osob, $P1, P2, P3$ značí určené pozice osob. Zeleně je zakreslen skutečný výškový profil, červeně TDM90 (viz 6.2). Osoba 1 se ve výsledku zobrazí dle očekávání. Osoby 2 a 3 se ovšem promítnou do bodu se stejnými horizontálními souřadnicemi (odlišnými pouze vertikálně). Totéž platí pro všechny další případné osoby vyskytující se v modře vyznačeném pásmu mezi osobami 2 a 3.

1. vyhodnocení na předem definované sadě identit (pouze pro bezkontextové varianty),
2. porovnání výsledných trajektorií s GPS pozicemi naměřenými uživateli při tvorbě testovacího datasetu,
3. analýza výsledků lidským pozorovatelem.

Vyhodnocení bezkontextové re-identifikace na sadě identit

Aby vyhodnocení nástrojů pro re-identifikaci co nejvíce vypovídalo o vhodnosti použití v kontextu práce, nebyly použity externí datasety identit. Místo toho byla vytvořena a manuálně upravena sada identit SITD (více informací v kap. 6.1.2), která se skládá výhradně z osob detekovaných v testovacích videozáznamech (viz 6.1.1).

Pomocí SITD mohou být hodnoceny pouze bezkontextové varianty algoritmů, jelikož kromě náhledů osob nenabízí žádné další údaje (např. čas či pozici osoby). Výsledky však mohou posloužit pro srovnání kvality použitých principů a pro nastavení citlivosti/parametrů re-identifikačních modulů.

Vzorové i výsledné identity (jinak též řečeno osoby) sestávají z množiny stop (neboli náhledů osob). Pro hodnocení správnosti výsledků bylo zavedeno několik metrik, které se navzájem doplňují a samy o sobě nejsou dostatečně popisné. Porovnává se:

- Pokrytí vzorových identit (PVI) – ke každé vzorové identitě (seřazené sestupně dle počtu stop) je přiřazena unikátní výsledná identita 1:1 (nemůže dále být přiřazena jiným vzorovým identitám) na základě toho, která výsledná osoba má ve vzorové identitě nejvyšší zastoupení stop. Stopy této osoby v rámci vzorové identity jsou pak považovány za správné, zatímco ostatní za chybné. Na základě podílu stop správné výsledné identity je poté určeno procento pokrytí vzorové osoby, potažmo pokrytí celého vzorového datasetu. Ideální hodnota je 100 %.
- Pokrytí výsledků (PV) – obdoba PVI, ovšem hodnota pokrytí je odvozena v opačném směru. Udává tedy, jak moc jsou pokryty výsledné identity vzorovými osobami. Ideální hodnota je 100 %.
- Poměr počtu osob (PPO) – udává poměr počtu výsledných identit vůči vzorovým. Vypovídá např. o přísnosti použitých parametrů, kdy může být vytvářeno mnoho

nových identit s malým počtem stop, nebo naopak sloučeno více vzorových identit do jednoho výsledku. Dále umožňuje rozlišit mezi kvalitou algoritmů, u kterých ostatní parametry dosahují totožných hodnot. Ideální hodnota je 1.

- Počet neurčených stop (PNS) – udává počet snímků, u kterých selhala identifikace a nebyly přiřazeny žádné osobě. Ideální hodnota je 0.

Při každém měření je soubor stop náhodně seřazen, což může znatelně ovlivnit následnou re-identifikaci (běžně v řádech jednotek procent, ovšem mohou nastat i výraznější výjimky). Proto bylo každé měření prováděno opakovaně a spočítána průměrná hodnota.

Princip metody	**Vyčerpávající porovnání**	**Úpravy obrazu**	**PVI** [%]	**PV** [%]	**PPO**	**PNS**
Histogram	ne	–	31,2	30,8	14,03	0
Histogram	ano	Gaussův f.	57,3	58,5	4,63	0
Histogram	ano	–	58,5	58,9	3,6	0
Histogram	ano	Bilaterální f.	61,7	63,0	4,07	0
AlignedReID	ne	–	62,2	62,2	2,75	0
AlignedReID	ano	CLAHE	87,9	86,5	1,22	0
AlignedReID	ano	Gaussův f.	88,9	88,9	1,13	0
AlignedReID	ano	Bilaterální f.	92,6	92,6	1,12	0
AlignedReID	ano	–	95,2	95,2	1,07	0

Tabulka 6.2: Přehled vybraných výsledků re-identifikačních algoritmů na SITD, seřazeno dle hodnoty PVI. Každé měření bylo prováděno 10× a výsledky zprůměrovány. U všech uvedených výsledků byly použity stejné (optimální dosažené) hladiny tolerance (12 kontejnerů na barevnou složku, *min. intersect 3* a *max. Hellinger 0,5* pro `HistogramTracker`, similarity 0,7 pro `AlignedReIdTracker` – vysvětleno v kap. 5.5). Vyčerpávající porovnání říká, zda byla porovnávána jen 1 poslední stopa (hodnota pole = ne), nebo srovnání probíhalo napříč všemi stopami jedné identity (hodnota pole = ano). Dále jsou uvedeny testované úpravy obrazu (viz kap. 5.2.2). PVI, PV, PPO a PNS udávají výsledky metrik navržených v kap. 6.2.5.

V tabulce 6.2 je možné vidět některé z naměřených údajů. Hodnota PNS byla ve všech případech 0, jelikož algoritmy implementované v rámci této práce v bezkontextovém režimu

vždy přiřadí snímek k osobě, popř. vytvoří osobu novou. V jiných algoritmech nebo v kontextovém režimu však může být stopa úplně vyřazena, např. když se neshoduje pozice osoby s výsledkem porovnání histogramů. Metrika je proto přesto uvedena.

Hodnoty PV a PVI se často shodují nebo jsou velmi blízké, přesto je nutné uvést obě, což lze ilustrovat na příkladu. Pokud by tedy výsledné identity obsahovaly např. po 10 správných stopách ze vzorových identit o 50 stopách, hodnota PV by dosáhla 100 %, zatímco pouhých 20 % metriky PVI by správně vypovídalo o chybě.

Bylo testováno více variant tolerance/citlivosti re-identifikačních modulů (podrobněji vysvětleno v kap. 5.5), hodnoty v tabulce jsou uvedeny pro přibližně optimální zjištěné konfigurace. Obdobně byly vyzkoušeny různé intenzity obrazových úprav. Gaussův filtr výsledky ovlivnil buď jen neznatelně, nebo spíše negativně. Podobně se neosvědčil ani algoritmus CLAHE, který nedosáhl ani v jednom testovaném případě lepších hodnot, než s originálním obrazem.

Bilaterální filtr šumu se ovšem ukázal jako možné vylepšení metod založených na histogramech. Splynutím barev a odstraněním šumu však dochází k větší podobnosti histogramů, proto je při jeho použití vhodné mírně snížit toleranci porovnávání, aby nedocházelo k příliš častým falešným shodám.

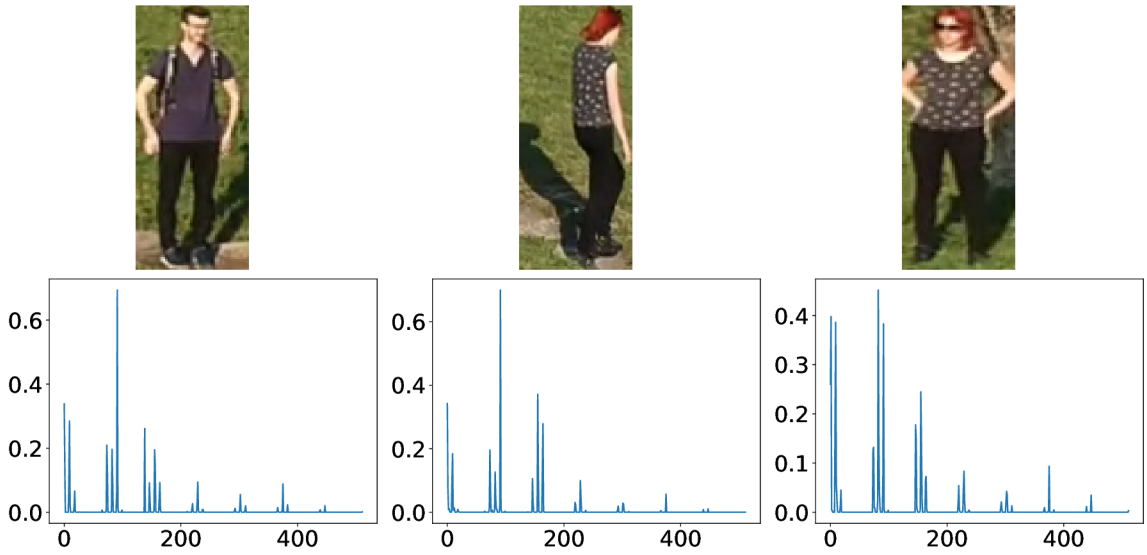
Z naměřených hodnot je dále patrné, že použití vyčerpávajicího porovnání (nová stopa je srovnávána se všemi stopami existujících identit) extrémně zlepšuje výsledky, zatímco při re-identifikaci na reálném záznamu se oproti použití pouze 1 posledního snímku každé identity tak znatelný rozdíl neobjevuje. Toto je dáno faktem, že v SITD jsou snímky zpracovány v náhodném pořadí, tudíž se mohou i dva po sobě jdoucí výrazně lišit. V reálném videozáznamu je oproti tomu osoba většinou v zachycena v podobné pozici, jako v předchozím snímku.

Z tab. 6.2 je patrné, že použití histogramů nedosahuje příliš dobrých výsledků (nejlepší varianta s úpravami obrazu je srovnatelná s nejhorším výsledkem AlignedReID). Histogramy by tak bylo vhodné použít jedině v kombinaci s další informací (např. pozice osoby), nebo pro videozáznamy s velmi vzdálenými osobami, kde není patrný dostatečný detail pro jiné přístupy.

Na na obr. 6.13 je možné vidět ukázku problematické situace. Na první pohled je zřejmé, že histogramy snímku 1 a 2 jsou si podobnější než histogramy 2 a 3, přestože snímek 1 patří rozdílné osobě. Totéž potvrzuje např. výpočet průniku histogramů 1-2 s hodnotou 3,83 oproti snímkům 2-3 s výsledkem 3,47 (vyšší znamená větší podobnost). Histogram je totiž ovlivněn barvami pozadí (které tvoří značnou část náhledů 1 a 2) stejně, jako samotnou osobou. Dále nerozlišuje ani texturu, ani pozici pixelů v obraze. Proto se v něm podobně projeví i tmavé tričko se světlými popruhy batohu osoby 1 a tmavé tričko se světlými body osoby 2.

Oproti tomu metoda AlignedReID správně přiřadí snímkům 1 a 2 hodnotou odlišnosti 0,82, zatímco snímky 2 a 3 ohodnotí odlišností 0,59 (nižší znamená větší podobnost). Celkově vychází výsledky této neuronové sítě velmi dobře. Je však třeba podotknout, že ze SITD byly odstraněny nekompletní a neplatné snímky. Výsledky na SITD tedy nemusí přesně odpovídat realitě.

Navíc může dojít k záměně identit, např. kvůli jednomu snímku se špatným odleskem v obraze, což by vedlo k znehodnocení takovýchto trajektorií. Je proto potřeba s těmito alternativami počítat a výsledné podmínky přizpůsobit reálnému použití (pravděpodobně snížením tolerance podobnosti).



(a) Snímek 1. (b) Snímek 2. (c) Snímek 3.

Obrázek 6.13: Ukázka problematické re-identifikace s použitím histogramů. K jednotlivým náhledům jsou přiřazeny jejich histogramy (variace barevných složek zkombinované do 1 rozměru, normalizované) – body na ose x reprezentují varianty barevných kombinací, osa y udává jejich relativní množství. V kap. 6.2.5 je rozebráno, proč je situace problematická.

Největší problém pozorovaný u této sítě byl s rozlišováním obrázků různých poměrů, jelikož model používá obrázky osob dané velikosti. Naprostou většinu obrázků je proto potřeba zvětšit či zmenšit. Toto běžně nezpůsobuje problémy.

Avšak případech, kdy je stejná osoba detekovaná např. s mnohem širším okolím než na jiném snímku, je po změně velikosti velmi rozdílný poměr proporcí osoby a pro AlignedReID se jeví jako odlišný člověk. Toto často způsobuje např. detekce pomocí modelu *Open Pose*, ukázku lze vidět na obr. 6.6.

Porovnání výsledků s hodnotami naměřenými v terénu

Při natáčení testovacího datasetu používaly účastníci mobilní telefony s aplikacemi pro záznam GPS pozice. Jejich trasy by tak teoreticky mělo být možné porovnat s výslednými trajektoriemi sledovaných osob. Zařízení se však lišila hardwarovým i softwarovým vybavením, pročež jsou zaznamenané trajektorie velmi odlišné – neshoduje se řádově ani počet zaznamenaných pozic, ani přesnost údajů. Tato metoda tedy ve výsledku nemohla být na testovacím datasetu použita.

Analýza výsledků kontextové re-identifikace lidským pozorovatelem

Subjektivně byly pozorovány a hodnoceny výsledné trajektorie a průběžné problémy celé aplikace (bližší ukázky lze vidět v kap. 6.3). Na základě různých odhalených nedostatků v použití bezkontextových algoritmů byla navržena vylepšení, která využívají i další dostupné informace.

Jak bylo popsáno dříve, rozlišování osob pomocí histogramů je samo o sobě nedostačující a dochází k častým záměnám identity (viz 6.13). Tato chybovost je částečně řešena zohledněním pozice osoby. Jelikož chodci se mohou pohybovat jen omezenou rychlostí, je

při ověřování totožnosti zohledněno, zda osoba mohla reálně urazit vzdálenost mezi dvěma porovnávanými body.

Modul tedy umožňuje nastavení maximální rychlosti pohybu. Pro každou stopu je pak kontrolováno, zda osoba v čase mezi porovnávanými snímky mohla takovou rychlostí vzdálenost překonat. Porovnávání pozic může být provedeno dvěma způsoby:

- Na základě obrazových souřadnic. Tato varianta funguje poměrně spolehlivě při použití statické kamery. Práce by však měla zvládnout zpracovat i videozáznamy z pohyblivého zdroje, které se v testovacím datasetu vyskytují. V takovém případě však toto omezení může při rychlých změnách pohledu/pozice dronu podávat falešně negativní výsledky při striktním nastavení. Naopak při volnějším nastavení správně reflektuje rychlé pohyby, ale podstatně se snižuje efekt při stabilním snímku.

- Na základě GPS pozice. Zde není problémem dynamika kamery, jelikož GPS pozice by měla být ve výsledku stejná. Do finální pozice osob se však v tomto případě promítá mnoho nepřesností (viz předchozí kapitoly – např. přesnost hardwarových senzorů dronu, výškový model povrchu Země, atp.). Tolerance tak nemůže být nastavena striktně a může stále docházet k záměnám identit osob v blízkém okolí.

U re-identifikace pomocí AlignedReID k záměnám identit dochází zřídka. To je navíc řešeno možností vyčerpávajicího porovnání všech stop osoby, kdy je velká pravděpodobnost, že i když nejasná stopa osoby 1 pronikne do stop osoby 2, bude nový náhled osoby 1 správně přiřazen spíše validním stopám osoby 1, než několika málo problematickým kusům, které se dostaly do stop osoby 2 chybně.

I zde však může nastat situace, že se identity smíchají. Proto je zavedena možnost stanovení maximálního množství stop, které mohou být u každé osobu použity k porovnávání s nově příchozím náhledem (např. posledních 10, což značně zvyšuje šanci na korektní spárování oproti použití jediného posledního náhledu). Omezí se tak situace, kdy chybně přiřazená stopa osoby 1 k osobě 2 ovlivnila přiřazení dalších stop osoby 1 k osobě 2 např. o 200 snímků později.

Další problém může nastat, pokud se nově příchozí stopy osob identifikují jedna po druhé. Může tak potenciálně dojít k chybnému přiřazení více náhledů různých osob k jedné identitě v rámci jediného snímku videozáznamu. Je však zřejmé, že v rámci snímku by se neměla jedna osoba vyskytovat vícekrát (až na výjimky typu fantomových objektů v obraze). Proto je byla zavedena možnost přiřazení všech osob v jednom snímku tak, aby se žádná identita neopakovala. Pokud je pak více stop přiřazeno k jedné identitě, rozhodne se na základě nejvyšší hodnoty podobnosti a ostatní stopy jsou ignorovány jako chybné.

6.2.6 Zobrazení a úpravy výsledků

Cílem práce je zobrazení výsledných trajektorií chodců v programu Google Earth. Jelikož stěžejní část celkového algoritmu zajišťuje převážně detektor osob, výpočet GPS souřadnic a následná re-identifikace osob, mohlo by se pouhé uložení a zobrazení výsledků zdát, jako snadná operace. I zde se však vyskytuje několik problémů, které jsou řešeny v této kapitole.

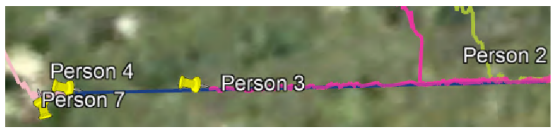
Export výsledků a nevhodná vizualizace

Pro zobrazení výsledků v Google Earth byla nejprve implementována možnost exportování tras do souboru typu GPX (viz 5.6.1). Výsledek je vidět na obr. 6.14. Tento formát lze v programu manuálně otevřít a zobrazit, ovšem s limitovanými možnostmi. Všechny osoby

proto mají trajektorii stejné barvy, což se ukázalo jako naprosto nepřehledné při zobrazení více trajektorií na jednom místě.

Přestože je tedy GPX obecně užívané řešení pro šíření GPS údajů, v Google Earth jsou hlavním nosičem soubory typu KML/KMZ (viz 5.6.2), které umožňují i přenosy grafiky a dalších detailů. Byla proto implementována také možnost exportu do KML souboru, přičemž jednotlivým osobám jsou přiřazeny náhodně generované barvy, aby byly od sebe dobře rozlišitelné. Takovýto výsledek je dále možné automaticky otevřít v Google Earth přímo z aplikace a konkrétní značení tras je navíc možné upravit dle libosti (šířka trajektorie atp.). Příklad finálního zobrazení trajektorií je vidět na obr. 6.15.

Obrázek 6.14: Formát GPX.



Obrázek 6.15: Formát KML.

Obrázek 6.16: Vizualizace trajektorií v programu Google Earth.

Extrémní členitost trajektorie

Pokud záznam disponuje vysokou snímkovou frekvencí a osoba je rozpoznána téměř v každém záběru, výsledná trajektorie obsahuje velké množství bodů. Ty oscilují kolem reálné trasy kvůli nejrůznějším nepřesnostem (senzory dronu, mírné posuny v detekci, nedokonalý model nadmořské výšky atp.). Problémem tedy může být extrémní členitost tras, kdy výsledek budí až chaotický dojem. Ukázku lze vidět na obr. 6.17.

Obrázek 6.17: Ukázka optimalizace výsledné trajektorie. Vlevo se nachází surová trasa sledované osoby sestávající ze 185 stop za čas 10 sekund. Vpravo je výsledná trajektorie agregovaná po intervalech délky 0,5 s a vyhlazená pomocí Kalmanova filtru.

Pro optimalizaci takovéto trasy je možné využít dvou nabízených metod. První z nich je agregace stop dle časového intervalu. Umožňuje zprůměrovat množství přespříliš detailních GPS pozic do jednoho bodu za signifikantní časový interval. Jelikož v reálném světě se osoba může pohybovat jen omezenou rychlostí, drobné rozdíly v poloze jsou zanedbatelné a výsledku by taková operace neměla ubírat na správnosti (při vhodné volbě intervalu).

Druhou možností je aplikovat Kalmanův filtr(viz 5.2.4) na výsledné pozice. Jelikož však stopy osoby nemusí ležet v pravidelných intervalech, mohla by výsledná trajektorie vypadat poněkud schodovitě, pokud by delší dobu nebyla známa nová pozice a odhad vycházel ze starých informací. Ideální možností je proto zkombinovat Kalmanův filtr s předchozí agregací pozic za časový interval. Výsledek je vidět na obr. 6.17.

Výskyt nežádoucích objektů

Tento problém vzniká již při detekci osob. Jedná se o falešně pozitivní výsledky, tedy nežádoucí objekty, označené jako validní osoby. Obvykle se jedná pouze o výpadky jednorázové

(např. náhodná chyba způsobená šumem) či krátkodobé (např. stín osoby při daném úhlu pohledu), méně často se pak vyskytuje perzistentní chyba v delším časovém úseku (např. označení odloženého oblečení jakožto osoby).

V kontextu detekce lze tyto problémy řešit jen obtížně (např. kompletní výměnou použitého algoritmu). Krátkodobé výpadky však lze poměrně dobře řešit v rámci úpravy výsledných trajektorií. Proto byla implementována možnost odstranění objektů:

- neobsahujících dostatečný počet stop,
- existujících pouze v krátkém časovém intervalu.

Limity pro minimální počet stop i pro nejkratší požadovaný časový interval lze nakonfigurovat. Při vhodně zvolených parametrech je pravděpodobnost odstranění důležité informace velmi nízká – např. u osob detekovaných po méně než 1 sekundu se bude pravděpodobně jednat o chybu, ale i pokud by se jednalo o validní nález, je takto okrajový výskyt poměrně nezajímavý. Přínos v odstranění chyb je naopak znatelný, a to obzvláště u detektorů s nízkou prediktivní hodnotou pozitivního nálezu.

6.3 Výsledky na testovacím datasetu

V teto kapitole budou uvedeny výsledky dosažené na testovacím datasetu (viz 6.1.1). Bude uvedeno několik ukázek za použití různých konfigurací aplikace pro demonstraci možných problémů či vylepšení (část jich byla rozebrána již v předchozích kapitolách, zde však budou názorné ukázky). Výsledné trajektorie budou vyhodnoceny lidským pozorovatelem (strojové měření bylo popsáno již v kap. 6.2.5).

Rozložení osob a pohledu dronu odpovídá ve všech následujících záznamech popisu v kap. 6.1.1. Na obr. 6.1 je možné vidět náhled z ptačí perspektivy. 6.1. Výškový profil terénu je zachycen na obr. 6.2. Ke každému záznamu bude uvedeno pro názornost několik snímků zachycujících jeho dění (obrázky v sérii postupují zleva doprava, shora dolů), společně bude také popsáno chování dronu (např. náklony větší než 5 °).

Uvedené výsledky budou vycházet z různých nastavení (jejich případné detailnější vysvětlení lze vždy dohledat v kap. 5 a 6.2). Pro přehlednost je však nejprve uvedena společná základní konfigurace, u každého výstupu zvlášť pak budou zmíněny rozdíly oproti následujícímu nastavení:

- Nejsou použity žádné obrazové úpravy – ani při předzpracování záznamu, ani při re-identifikaci. Rozlišení použitého videozáznamu je 1920×1080 pixelů.
- Na údaje z letového deníku dronu je aplikován Kalmanův filtr (viz 6.2.2).
- Pro detekci osob je použita neuronová síť Faster R-CNN with NASNet-A, která dosáhla nejstabilnějších výsledků (viz 6.2.3). Minimální hladina jistoty detekce byla nastavena na 0,5 a maximální povolená hodnota překrytí osob na 40 %.
- GPS pozice osob je vypočítána s využitím databáze nadmořské výšky založené na DEM (*Digital Elevation Model*) TDM90 (*TanDEM–X 90m Digital Elevation Model*) či EU25 (*Copernicus Land Monitoring Service – EU-DEM*) (více informací v kap. 6.2.4). Maximální vzdálenost chodců od dronu je nastavena na 100 m.
- Re-identifikaci osob zajišťuje modul AlignedReID, který dosáhl nejpřesnějších výsledků při měření bezkontextových variant použitých algoritmů na sadě identit testovacího datasetu (viz 6.2.5). Od toho je také odvozena hodnota maximální odlišnosti

osob 0,65, taktéž je použito vyčerpávající porovnání (pro max. 30 posledních snímků každé identity). Z kontextových vylepšení je zaveden limit 1 stopy pro 1 identitu v rámci 1 snímku. Není použito omezení na základě rozdílu pozice (ani souřadnicemi obrazovými, ani GPS), jelikož pro re-identifikaci pomocí AlignedReID není ve většině případů potřeba.

- Z výsledných tras jsou odstraněny identity s méně než 30 stopami (frekvence videozáznamu je 30 fps a validní osoby běžně dosahují stovek stop za několik sekund) a s kratším výskytem než 1 sekundu. Trajektorie jsou agregovány do 0,5 s intervalů, vyhlazeny Kalmanovým filtrem a exportovány do KML formátu.

6.3.1 Ukázka výsledků na záznamu č. 1

Snímky z videozáznamu lze vidět na obr. 6.18. Délka je 27 vteřin. Zpočátku se je dron od osob vzdálen přibližně 40 m směrem na západ (pohled kamery míří na východ) a vznáší se ve výšce 7,5 m, poté se přiblíží o 10–15 m a otočí doleva o 5 ° (yaw). Následně přerušovaně vystoupá do výšky 10,5 m se stejným sklonem kamery, takže nejbližší osoba spodní hraně obrazu na chvíli zmizí ze záběru. Poté se dron v této výšce vzdálí o 10 m a rotuje (roll) přibližně o 15 ° vlevo, poté se vyrovná a rotuje přibližně pod úhlem 10 ° doprava (poslední snímek). Celkem je zachyceno 6 osob, které celou dobu stojí na svých místech.

Obrázek 6.18: Ukázka průběhu videozáznamu 1. Sever směrem vlevo. Detaily jsou popsány v kap. 6.3.1.

Na tomto záznamu budou ukázány převážně základní problémy v re-identifikaci založené na principu histogramů. Re-identifikace s použitím AlignedReID zde nevykazuje problémy a dosahuje přesných výsledků.

Trajektorie vykreslené na obr. 6.19 jsou tedy extrémně chybné a neodpovídají záznamu. Prvním klíčovým problémem je, že sahají do vzdálenosti stovek metrů, přičemž osoby stály 20–40 m od dronu. Toto je způsobeno použitím původní metody pro výpočet pozic chodců, která nevyužívala databáze nadmořských výšek. Pro názornost také není omezena maxi-

mální vzdálenost osob od dronu – při jejím použití by zde totiž většina výsledků zmizela.

Obrázek 6.19: Ukázka chybných výsledků z prvních 5 s videozáznamu 1 pro demonstraci řešených chyb. Sever směrem k levému hornímu rohu. Na letový deník nebyl aplikován Kalmanův filtr. Pro určení pozice chodců byla použita metoda bez znalosti nadmořské výšky (viz 6.2.4). Není omezena maximální vzdálenost od dronu. Re-identifikace proběhla na principu histogramů v bezkontextové variantě a s parametry získanými měřením v kap. 6.2.5, ovšem bez vyčerpávajícího porovnání a bez omezení 1 stopa pro 1 identitu za snímek. Výsledky bez dalších úprav exportovány do souboru GPX.

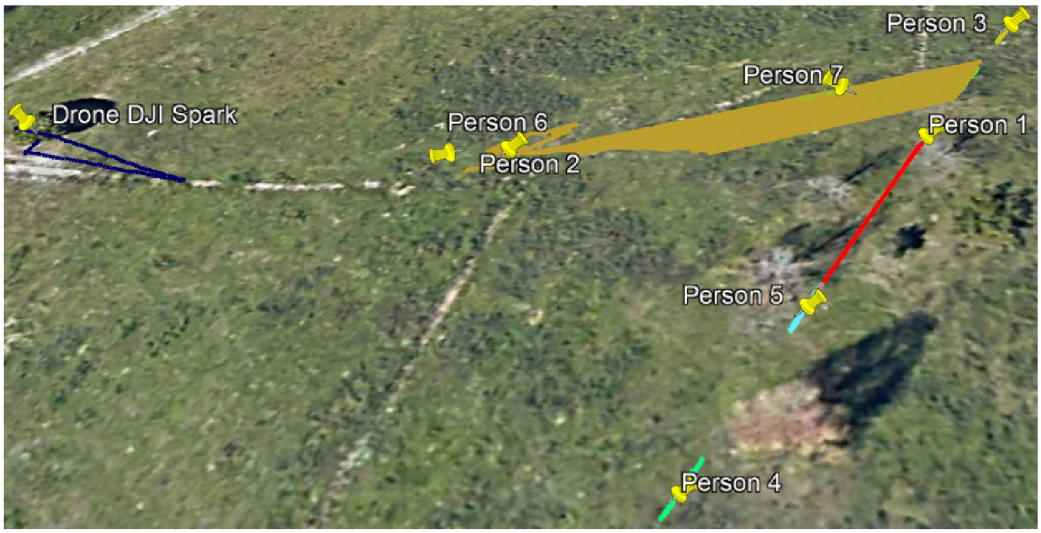
Během 5 s, pro které jsou vykresleny trajektorie na obrázku, se dron pohybuje směrem vpřed. Zpočátku je velmi malý úhel mezi horizontálou a směrem pohledu na osoby, pročež jsou lokalizovány velmi daleko od dronu. Postupně se úhel zvětšuje (dron pohlíží na chodce více směrem dolů), což je jejich pozice určí blíže. Tento princip byl vysvětlen na obrázcích 6.9, resp. 6.11.

Druhým problémem je použití nejslabší metody re-identifikace (dle měření v kap. 6.2). Přestože je tak na záznamu 6 osob, ve výsledcích je určeno 17 (některé nejsou vidět, jelikož se nachází daleko mimo obrázek). Kvůli přísným kritériím a porovnávání pouze posledního snímku proto rozdělováním skutečných identit vzniká mnoho falešných.

Na obr. 6.20 jsou předchozí problémy napraveny a je proto použito celých 27 vteřin záznamu. Metoda lokalizace osob pomocí nadmořské výšky s využitím DEM TDM90 vedla k mnohem přesnějšímu určení pozic osob, které přibližně odpovídají záznamu. Při použití tolerantnějšího nastavení re-identifikace dále nevzniklo tolik falešných identit.

Oproti tomu však nastal nový problém – po chvíli došlo ke splynutí identity 6 a 7 ve variantu 2, jejíž trajektorie kvůli rychlému přeskakováním mezi pozicemi vytvořila souvislou plochu. Kvůli nepoužití kontextu tak obsáhne jedna identita i více stop za snímek. Podobně také osoba 1 přeskakuje na pozici osoby 5 a zpět, což mezi nimi vytvořilo souvislou linii.

Tyto problémy byly vyřešeny použitím kontextových omezení re-identifikace, jak je uvedeno na obr. 6.21. Všechny sledované osoby byly rozlišeny správně a nedošlo ani k záměnám, ani k tvorbě chybných identit. Největším problémem/chybou je zde pohyb osoby 2, která však celou dobu stojí na místě. Je způsoben nedostatečnou přesností modelu povrchu TDM90 v daném místě, což se při delších pohybech dronu může promítnout do výsledků jako pohyb stacionárního objektu (viz obr. 6.11).



Obrázek 6.20: Ukázka chybných výsledků na celém videozáznamu 1 pro demonstraci řešených chyb. Sever směrem k horní hraně obrázku. Na letový deník nebyl aplikován Kalmanův filtr. Pro určení pozice chodců byla využita databáze nadmořských výšek TDM90. Není omezena maximální vzdálenost od dronu. Re-identifikace proběhla na principu histogramů v bezkontextové variantě se zvětšenou tolerancí podobnosti oproti obr. 6.19, tentokrát s vyčerpávajicím porovnáním, ale stále bez omezení 1 stopa pro 1 identitu za snímek. Výsledky bez dalších úprav exportovány do souboru KML pro lepší rozlišení trajektorií.

Obrázek 6.21: Ukázka finálních výsledků na videozáznamu 1. Sever směrem k horní hraně obrázku. Byla použita DEM TDM90. Byly aplikovány všechny předdefinované postupy (popsané na začátku kap. 6.3), s výjimkou re-identifikace, která proběhla na principu histogramů v kontextové variantě s vyčerpávajicím porovnáním, s limitem 1 stopa pro 1 identitu za 1 snímek a také s omezením na základě GPS pozice. Pro přehlednost všech demonstrovaných problémů byly odstraněny ukazatele konečných pozic osob.

Tento posun je vidět i na osobě 7. Ostatní osoby však na pohyb dronu vpřed/vzad nereagují stejným způsobem, jelikož se nachází na zlomové hranici, kde DEM obsahuje schodovitý nárůst nadmořské výšky (schod v DEM lze vidět v pravé části obr. 6.2b, vy-

světlení pak na obr. 6.12). Kvůli tomu u osob v několikametrovém pásmu reálného pohledu (v oblasti stromořadí, kde sledovaní chodci stojí) není ve výsledku vykazován posun ve východním/západním směru, pokud toto pásmo nepřekročí.

Dále je patrná oscilace osoby 7 kolem její reálné pozice a u osob v zmíněném pásmu je vidět přibližně severo-jižní posun. To je částečně způsobeno zmíněným problémem DEM, částečně otáčením (yaw) a rotací dronu (roll). Změny v orientaci dronu jsou sice ve všech směrech reflektovány algoritmem pro výpočet pozice, ovšem kvůli dříve popsaným chybám (např. nepřesnost senzorů, synchronizace záznamu s letovým deníkem či nedostatečně časté záznamy) se přesto projevují do výsledných trajektorií.

Lze pozorovat rozdíl mezi obr. 6.20 a 6.21 v roztažení tras, např. na osobě 4. V druhém uvedeném obrázku je nepřesnost mnohem menší díky použitým úpravám letového záznamu a výsledků.

Na trajektorii osoby 2 jsou vidět dvě ostřejší vybočení – první z nich (na pravé straně) je způsobeno otočením dronu vlevo o 5 ° na začátku záznamu. Lze dopočítat, že při vzdálenosti osoby 30 metrů (měřeno pomocí Google Earth) by odchylka způsobená čistě nezohledněním natočení měla dosahovat přibližně 2,6 m. Ve výsledné trajektorii se však promítá jako vychýlení o přibližně 1 m (výsledek je přitom ovlivněn i dalšími chybami), což je pro určení poměrně

Druhé vybočení (na levé straně) trajektorie osoby 2 je způsobeno opakovanou rotací (roll) ke konci záznamu o 10-15 °. Zde se na nepřesnosti podílí další typ problému – chodidla osoby jsou předpokládána ve spodní části bounding boxu, pokud ale dojde k rotaci snímku, chodidla jsou reálně jinde a pozice je tak určena s odpovídající chybou.

Celkově pak lze pozorovat mírně větší rozestupy mezi osobami v severo-jižním směru. I při zanedbání ostatních chyb by měl být směr výskytu osoby zachován. Nepřesnost je tedy s nejvyšší pravděpodobností způsobena nesprávným údajem o horizontálním úhlu pohledu kamery, který je ve skutečnosti mírně menší, než udává výrobce. Popř. kamera hodnoty nedosahuje při natáčení v rozlišení, s jakým byl vytvořen testovací dataset.

Shrnutí výsledků na záznamu 1

Na záznamu 1 bylo demonstrováno použití histogramů k re-identifikaci a s tím spojené problémy i vylepšení. Finální výsledky lze popsat jako velmi dobré z pohledu rozlišení identit a jednoznačnosti trajektorií chodců. Celkový algoritmus zvládá úspěšně reflektovat pohyb dronu, včetně změn nadmořské výšky dronu i orientace kamery.

Pro 5 z 6 osob byla správně určena GPS pozice s max. odchylkou do 5 m. Největší komplikací se zde ukázala být nepřesná databáze nadmořských výšek TDM90, která způsobila zdánlivý pohyb stacionární osoby v rozmezí přibližně 15 m. Bohužel nemohla být nahrazena DEM EU25, jelikož v místě pohybu osob obsahuje ještě větší chybu (napříč své jinak vyšší přesnosti).

6.3.2 Ukázka výsledků na záznamu č. 2

Snímky z videozáznamu lze vidět na obr. 6.22. Délka je 18 vteřin. Dron se vznáší na místě ve výšce 2,7 m, tudíž dochází jen k velmi drobným změnám v natočení a pozici (max. 1 m). Pohled kamery míří na východ. Celkem jsou zachyceny 4 osoby. Osoba 1 (číslováno zleva doprava) přejde od levého okraje k pravému, poté se otočí a vrátí do středu záběru. Osoba 2 je nejblíže dronu, stojí na místě a v polovině záznamu udělá několik kroků (částečně překryta větví stromu) směrem doleva. Osoba 3 se nehýbe. Osoba 4 je vzdálená, zpočátku

kompletně zakrytá korunou stromu, poté udělá několik kroků vpravo, otočí se a jde nahoru po stráni (dál od dronu).

Obrázek 6.22: Ukázka průběhu videozáznamu 2. Sever směrem doleva. Detaily jsou popsány v kap. 6.3.2.

Výsledné trajektorie na obr. 6.23 zjevně vůbec neodpovídají realitě. Příčinou je nepřesnost v TDM90. Přímo mezi dronem a osobami se nachází schodovitá změna v nadmořské výšce o 4 m, osoby tak jsou výše než dron. Dron se však nachází pouze 2,7 m nad zemí. Směrový vektor všech osob tak prochází touto pomyslnou clonou mezi dronem a osobami, a proto se všechny trajektorie nacházejí na jedné linii. Princip byl vysvětlen na obr. 6.12.

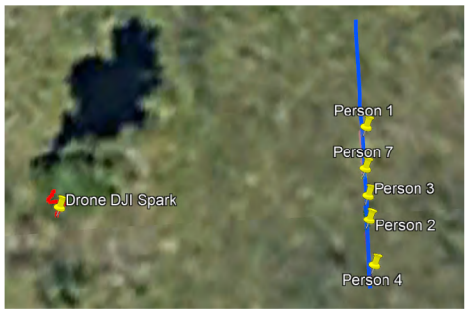
Tento problém lze vyřešit použitím DEM EU25, kde je poměrně přesně zachycen povrch v první polovině svahu v oblasti testovacího datasetu – a právě zde se vyskytují osoby v záznamu 2. Výsledky při jeho použití lze vidět na obr. 6.24.

Jak je vidět, pohyb osob velmi věrně odpovídá záznamu. Relativní chyba v rámci jednotlivých trajektorií (vůči určeným pozicím) je téměř nulová a nepřesahuje 1 m. Absolutní odchylka trajektorii od reálných pozic je však patrná – všechny blízké osoby jsou zobrazeny 5–7 m blíže ke dronu.

Tato chyba je způsobena opět schodovitostí DEM (ovšem zanedbatelnou oproti TDM90), kde se dron podle EU25 vyskytuje na podstatně nižší hladině než osoby. Ve skutečnosti je přibližně ve stejné výšce, ale ihned za dronem směrem na východ se nachází prudké několikametrové snížení, které se v DEM promítá už v místě dronu (horizontální rozlišení v desítkách metrů).

Nečekaným prvkem je dále osoba s číslem 7, přestože na snímku se pohybují pouze 4 chodci a jejich trajektorie (osoby 1 až 4) jsou platně zaznamenané. Kousek za osobou 1 byl totiž odložen na zemi batoh s bundou, jak je vidět na obr. 6.25d. Předměty byly chybně detekovány jako osoba na více než 200 snímcích napříč celým záznamem, čímž vznikla nová identita.

Dále se v záznamu osoba 3 pohybuje chvíli překrytá korunou stromu (viz obr. 6.25a). Její obraz kvůli odlišnosti nebyl přiřazen správně, kvůli čemuž vznikla osoba 5. Osoba 6 pak vznikla z paže chodce, který stál mimo záběr – na moment se do záběru dostala jeho ruka



Obrázek 6.23: Ukázka výsledků na videozáznamu 2. Sever směrem k horní hraně obrázku. Použita předdefinovaná nastavení s DEM TDM90.

Obrázek 6.24: Ukázka výsledků na videozáznamu 2. Sever směrem k horní hraně obrázku. Byla použita předdefinovaná nastavení s DEM EU25.

(viz obr. 6.25e)a byla chybně detekována jako platná osoba. Obě tyto nežádoucí identity byly automaticky odstraněny při úpravě výsledků, jelikož obsahovaly malý počet celkových stop (konkrétně 6 a 2) a v záznamu nevydržely ani po dobu jedné sekundy.

Další problematická situace je vidět na obr. 6.25b, kde se nachází osoba 4 zakrytá větvemi stromu, zatímco na obr. 6.25b je již celá dobře viditelná. Rozdíl je vskutku markantní a porovnávání pomocí histogramů zde selhává. AlignedReID však problematickou situaci dokázalo vyřešit správně.

V ukázkách záběrů ze záznamu je na obr. 6.22 vidět, že 3 osoby jsou oblečeny ve stejných odstínech tmavě šedé a černé, kvůli čemuž vypadají vskutku velmi podobně. V jistých částech snímku se navíc tyto osoby částečně překrývají (3 a 1, 3 a 2), když osoba 3 prochází

(a) Částečně překrytá osoba.

(b) Vzdálená osoba dobře viditelná.

(c) Vzdálená osoba za stromem.

(d) Odložený batoh.

(e) Ruka osoby mimo snímek.

Obrázek 6.25: Ukázka problematických částí v záznamu 2.

napříč scénou. Také tuto situaci AlignedReID dokázalo správně vyřešit a nedošlo k záměně identit. Tomu částečně předchází i omezení překrytí osob na maximálně 40 %.

Shrnutí výsledků na záznamu 2

Záznam je celkově náročný z pohledu re-identifikace osob. Popsaná úskalí však byla velmi dobře vyřešena pomocí příznaků získaných neuronovou sítí AlignedReID. Jak bylo uvedeno, re-identifikace selhala pouze v několika případech překrytí větvemi u osoby 3.

Vznik neplatné identity zapříčinila chyba způsobená falešně pozitivním výsledkem detektoru a posun trajektorií oproti reálné pozici je zapříčiněn převážně nedostatečným detailem digitálního modelu. Relativní pohyb osob je jinak zachycen velmi přesně. Chyba tedy nevychází z principu navrženého algoritmu, ale z jeho dílčích složek.

6.3.3 Ukázka výsledků na záznamu č. 3

Snímky z videozáznamu lze vidět na obr. 6.26. Délka je 10 vteřin. Dron se pohybuje přibližně ve výšce 7 m. Pohled kamery směřuje na východ. Celkem jsou zachyceny 4 osoby v proměnné vzdálenosti 10–50 m od kamery. Zpočátku osoba 1 (číslováno zleva doprava) udělá několik kroků vlevo, poté se dron začne přibližovat. Zároveň s ním osoba 1 běží k osobě 2 (nejblíže dronu). Ostatní lidé stojí na místě.

Obrázek 6.26: Ukázka průběhu videozáznamu 3. Sever směrem doleva. Detaily jsou popsány v kap. 6.3.3.

Na obr. 6.27 lze vidět, že výsledné trajektorie odpovídají pohybu osob v záznamu. Tvar trasy osoby 4 je poměrně přesně zachycen. Je zaznamenána jak její počáteční chůze směrem

na sever (z obrázku není příliš patrné, jelikož je překryta následným návratem při cestě k osobě 1), tak fáze běhu. Obě části odpovídají skutečnosti.

Obrázek 6.27: Ukázka výsledků na videozáznamu 3. Sever směrem k horní hraně obrázku. Byla použita předdefinovaná nastavení s databází nadmořských výšek TDM90.

Re-identifikace osob proběhla úspěšně. Nebyla zaznamenána žádná chybná identita. Podobný výsledek je možné očekávat, jelikož osoby se pohybují dostatečně daleko od sebe a nepřekrývají se, také nejsou zacloněny stromy či jinými objekty a navíc se pohybují poměrně blízko dronu. Detekce osob proto taky proběhla bezproblémově.

Hlavní patrnou nepřesností je zdánlivý pohyb osoby 1, která celou dobu stojí na místě. Způsobuje to již dříve popsaná nepřesnost v DEM (viz obr. 6.11).

Druhým problémem je posun všech trajektorií o 5–15 m směrem na sever oproti reálným pozicím. Toto je převážně způsobeno nepřesnými údaji o natočení dronu z letového deníku. Videozáznam byl pořízen až po 20 minutách letu, přičemž byl dron celou dobu ovlivňován větrem a pravděpodobně se u některých senzorů sčítala chyba měření. V záznamu 1 (viz kap. 6.3.1), který byl pořízen na začátku letu, takový posun není patrný.

Trajektorie osob 2 a 3 nabývají tvaru krátké úsečky místo předpokládaného stacionárního bodu. Podobná nepřesnost již byla popsána a vysvětlena v záznamu 1 (viz kap. 6.3.1 pro osoby na stejných místech. Odchylka však nepřesahuje 2 m.

Dále je možné pozorovat větší rozestupy mezi osobami v severo-jižním směru (dle měření v Google Earth) oproti reálnému záznamu. Toto je způsobeno mírně odlišnou hodnotou horizontálního zorného úhlu kamery, než jaké ve skutečnosti dosahuje (bylo vysvětleno a pozorováno též u záznamu 1).

Shrnutí výsledků na záznamu 3

Relativní odchylka (vůči určené pozici) trajektorií osob na záznamu 3 se pohybuje v řádech jednotek metrů. Při zvážení současného pohybu dronu i sledovaných cílů (i při běhu) je taková nepřesnost v kontextu scény poměrně uspokojivá.

Největším problémem se ukázaly být nepřesnosti dat z letového záznamu a výškové databáze. Absolutní chyba vůči reálné pozici tak kvůli nasčítané chybě hardwarových senzorů, které udávají natočení dronu, nabývá i 15 m, a to formou posunu celé scény.

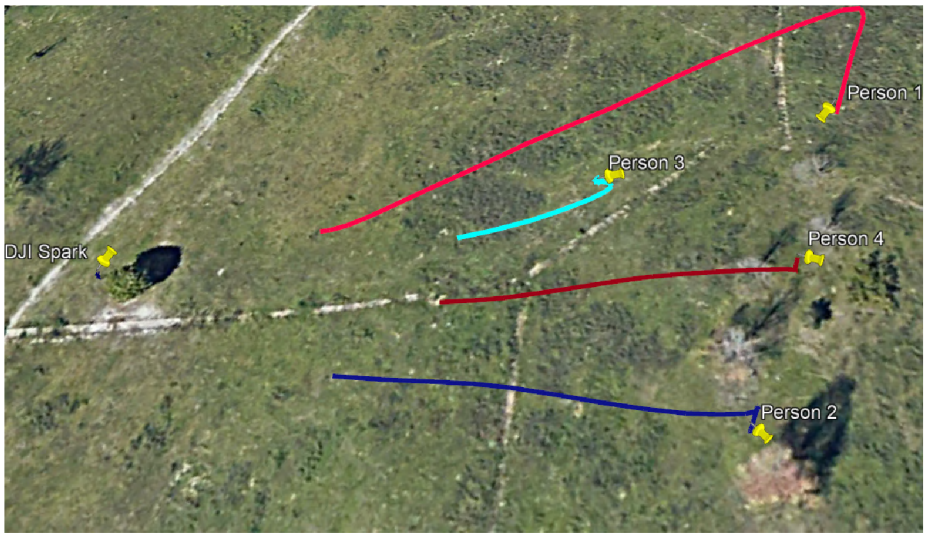
6.3.4 Ukázka výsledků na záznamu č. 4

Snímky z videozáznamu lze vidět na obr. 6.28. Délka je 15 vteřin. Dron se vznáší přibližně ve výšce 7 m a dochází jen k minimálním vychýlení od pozice a orientace. Pohled kamery směřuje na východ. Celkem jsou zachyceny 4 osoby, které začínají ve vzdálenosti přibližně 10 m od dronu a postupně se vzdalují o 30–40 m, stoupají nahoru po svahu. Osoba 2 (číslováno zleva doprava) se zastaví přibližně v půlce. Osoba 1 míří oproti ostatním nejprve více směrem na sever a poté se stočí zpět k centru záběru.

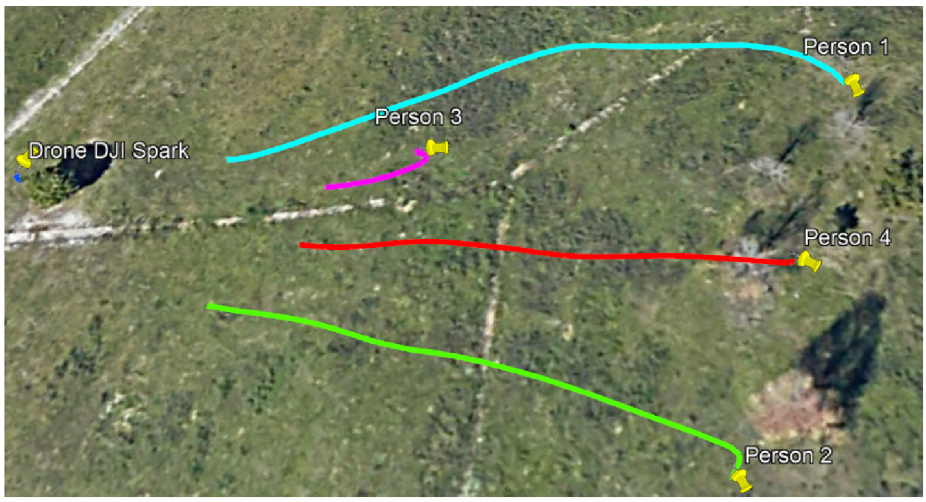
Obrázek 6.28: Ukázka průběhu videozáznamu 4. Sever směrem doleva. Detaily jsou popsány v kap. 6.3.4.

Na obr. 6.29 jsou vidět výsledné trajektorie osob, které jsou oproti reálným pozicím posunuty přibližně o 10 m směrem na sever a podobně také směrem na východ (což se ovšem příliš neprojevilo kvůli schodovitému pásmu v TDM90, jak bylo popsáno na obr. 6.12).

Tato chyba je způsobena převážně velmi nepřesnými záznamy v letovém deníku. Jelikož byl záznam 4 pořízen až po 20 minutách letu, průběžná chyba hardwarových senzorů dronu se nasčítala až do hodnot 5–10 °. Tato chyba by mohla být napravena rekalibrací dronu, popř. úpravou hodnot v letovém deníku.



Obrázek 6.29: Ukázka výsledků na videozáznamu 2. Sever směrem k horní hraně obrázku. Použita předdefinovaná nastavení s DEM TDM90.



Obrázek 6.30: Ukázka výsledků na videozáznamu 2. Sever směrem k horní hraně obrázku. Byla použita předdefinovaná nastavení s DEM TDM90. Navíc byla změněna hodnota otočení dronu (yaw) o přírůstek 10 °, tedy směrem doprava, a hodnota náklonu (pitch) o −5 °, tedy směrem dolů.

Na obr. 6.30 jsou pak výsledné trajektorie po aplikaci fixního přírůstku na údaje letového deníku. Tyto trasy již dosahují méně než 5 m odchylky od reálné předlohy. Na první pohled je patrná chyba v koncové poloze osoby 3, což je z hlavní části zapříčiněno nedostatečnou přesností DEM.

Dále je možné pozorovat větší severo-jižní rozestupy mezi osobami 1, 2 a 4, které by měly stát pouze v rozmezí poblíž rostoucích stromů. Podobně jako v předchozích záznamech je toto způsobeno mírně odlišnou hodnotou zorného pole kamery, než které reálně dosahuje (viz 6.3.1).

Jinak jsou však trajektorie osob velmi přesné. Chyba se pohybuje v řádech jednotek metrů. Současně věrně kopírují reliéf reálného pohybu sledovaných subjektů, včetně popsaného stočení osoby 1.

Z hlediska re-identifikace v záznamu nenastaly větší problémy. Vznikla pouze jedna falešná identita kvůli chybě detektoru, který označil stín osoby 3 jako validní detekci. Chybná detekce však proběhla jen na několika snímcích a pro nedostatečný počet stop byla automaticky odstraněna z výsledků.

Shrnutí výsledků na záznamu 4

Výsledné trajektorie po úpravě odchylky senzorů dronu poměrně přesně odpovídají vstupnímu záznamu s chybou přibližně 5 m. Nepřesnost měření dronu, která narostla do velikosti jednotek až desítek stupňů, byla způsobena dlouhou dobou letu, kdy se průběžně kumuluje chyba. Dále se také projevila nepřesnost DEM, podobně jako v předchozích záznamech.

Kapitola 7

Závěr

V rámci práce byl uveden přehled komerčně užívaných dronů, jejich typů a vybavení. Byly vysvětleny principy funkce jednotlivých senzorů. Dále práce rozebírá stavbu a fungování konvolučních neuronových sítí, jejich využití k detekci osob v obraze a možné variace, jejichž vhodnost je porovnávána v kontextu zpracování videozáznamu pořízeného z dronu, tedy při pohledu z výšky a různých úhlů pohledu.

Na základě uvedených informací jsou navrženy metody pro zjištění GPS souřadnic sledovaných chodců, a to bez nutnosti speciálního vybavení dronu. Dále jsou popsány obecné principy metod pro re-identifikaci osob a konkrétní trendy v této oblasti.

Jelikož téma práce zahrnuje širokou množinu použití, architektura výsledné aplikace je navržena modulárně tak, aby bylo možné ji snadno přizpůsobit vyžadovaným podmínkám. Program implementuje dvě z navržených metod pro určení GPS pozice osob a jejich sledování. K tomuto účelu využívá digitálních databází nadmořských výšek zemského povrchu, které jsou veřejně dostupné. Byly ukázány problémy a nepřesnosti spojené s jejich využitím.

Pro detekci jsou využívána volně dostupná řešení detektorů osob pomocí konvolučních neuronových sítí. Byly otestovány 4 různé varianty detektorů, jejichž úspěšnost a vhodnost do různých situací byla rozebrána. Pro výsledné experimenty byla vybrána nejpřesnější varianta, která na datasetu MS–COCO dosahuje hodnoty mAP (*Mean Average Precision* 43. Na základě získaných poznatků byla navržena a aplikována potřebná vylepšení, např. pro odstranění duplicitních detekcí.

Pro účely práce byl vytvořen dataset několika videozáznamů a letových informací s využitím dronu DJI Spark. Na něm byly navržené metody a celá aplikace otestována. Práce se soustředí také na jednoznačnost rozlišení sledovaných osob a jejich trajektorií, pročež byla za účelem vyhodnocení re-identifikace osob vytvořena sada identit obsahující 384 snímků 6 osob z pořízeného datasetu.

Byly implementovány nejen základní metody re-identifikace pomocí histogramů, ale také využití „state of the art“ řešení dané problematiky pomocí strojového učení – konkrétně neuronovou sítí AlignedReID, která podává na testovacím datasetu úspěšnosti 95,2 % i bez použití dalších obrazových úprav. Oproti tomu přístup na bázi histogramů dosáhl i s úpravami maxima přibližně 60 %. Pro další zlepšení a řešení některých problémů byla zavedena zpřesnění výsledků formou kontextových omezení re-identifikace.

Dále byl vyzkoušen vliv několika typů obrazových úprav na kvalitu detekce osob a také na re-identifikační algoritmy. Při použití neuronových sítí však nedošlo v naprosté většině ke zvýšení přesnosti, spíše naopak. Mírně pozitivní vliv v řádech jednotek procent byl pozorován u bilaterálního filtru. Výsledky byly měřeny na uvedených datasetech, které dosahují dobré kvality obrazu, proto pravděpodobně nebylo třeba dalších úprav.

Na testovacím datasetu se většinou vzdálenost chodců od dronu pohybuje v rozmezí 10–60 m. Horní hranice nepřesnosti určených GPS pozic dosahuje ve většině testovaných případů řádu jednotek metrů.

Nejvíce problematickou částí aplikace se ukázaly být databáze nadmořských výšek, které svou malou přesností (např. horizontální rozlišení 90 m) mohou zavádět chybu v desítkách metrů i do výsledných trajektorií. Je proto důležité mít k dispozici kvalitní model povrchu v dané oblasti.

Další problematickou složkou jsou záznamy měření senzorů dronu, které se časem sčítají. I přes filtrování Kalmanovým filtrem proto bylo na 2 testovaných záznamech po 20 minutách letu pozorováno chybně udávané natočení o 10 °, resp. o 5 ° v případě sklopení. To vedlo k vychýlení trajektorií o 15 m.

Při zkalibrovaném letu a vhodném výškovém modelu však, i přes vzájemné umocňování všech v práci zmíněných chyb z různých zdrojů, většinou chyba nepřesahuje 5 m. Z výstupu aplikace tak lze poměrně přesně zjistit, kudy se osoby pohybovaly, a to s použitím čistě softwarového řešení.

Chyby v průběhu aplikace mohou pramenit z více různých zdrojů. Data o pozici dronu jsou limitována podstatou GPS systému. Senzory dronu mohou uvádět nepřesná/nedostatečná data jak ze své podstaty, tak i vlivem okolí (např. počasí). Detekce osob v obraze díky strojovému učení dosahuje skvělých výsledků, přesto dochází k falešně negativním i pozitivním výsledkům poměrně často. Metoda pro určení pozice chodců vychází z aproximace pomocí nadmořské výšky zemského povrchu, který nemusí být v daném místě dostatečně přesně zmapován. Algoritmy pro re-identifikaci osob mohou relativně často chybovat, mají k dispozici pouze omezené informace oproti lidskému vnímání.

Uvedené chyby se práce snaží rozebrat a ve většině případů nabízí alternativy pro jejich řešení. I přesto existují další vylepšení, která by se dala aplikovat pro konkrétní situace. Problémové části proto aplikace umožňuje díky své architektuře a modulárnosti měnit podle potřeby bez větších zásahů.

Celkově se také nezadržitelně vyvíjí oblast strojového učení. Nástroje použité v aplikaci tak díky tomu bude v blízké budoucnosti (možná i v čase tvorby této práce) možné nahradit novějšími verzemi a lepšími modely vytvořenými na míru konkrétním situacím.

Výsledné trasy chodců mohou být exportovány do několika typů souborů běžně používaných pro přenos GPS souřadnic. Ty mohou být načteny a zobrazeny na satelitní mapě pomocí programu Google Earth či jiných geografických aplikací.

Bylo ukázáno, že určit GPS pozice chodců v záznamu a zaznamenat jejich trajektorie je možné i s využitím běžných modelů komerčních dronů, bez nutnosti speciálního hardwarového vybavení. Byly potvrzena funkčnost implementovaných metod a také odhaleny jejich limity a nepřesnosti.

Literatura

[1] Adarsh Verma: *Most Popular Programming Languages For Machine Learning And Data Science*. Prosinec 2016, [Online; navštíveno 11.3.2019].
URL https://fossbytes.com/popular-top-programming-languages-machine-learning-data-science

[2] Alex Graves: *Generating Sequences With Recurrent Neural Network. Computing Research Repository*, 2013.

[3] Alex Krizhevsky et al.: *ImageNet Classification with Deep Convolutional Neural Networks. Advances in neural information processing systems*, ročník 25, 2012: s. 1097–1105.

[4] Alexander Mordvintsev and Abid Rahman Khan: *OpenCV-Python Tutorials*. 2013, [Online; navštíveno 12.5.2020].
URL https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html

[5] Andrej Karpathy: *Convolutional Neural Networks for Visual Recognition*. [Online; navštíveno 30.3.2019].
URL http://cs231n.github.io

[6] Anup Bhande: *What is underfitting and overfitting in machine learning and how to deal with it.* Březen 2018, [Online; navštíveno 3.4.2019].
URL https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

[7] autorů, K.: *Americký družicový navigační systém NAVSTAR GPS*. Leden 2007, [Online; navštíveno 20.3.2019].
URL http://www.czechspaceportal.cz/3-sekce/gnss-systemy/gnss-mimo-evropu/americky-navstar-gps

[8] Bharath Raj: *Data Augmentation | How to use Deep Learning when you have Limited Data*. Duben 2018, [Online; navštíveno 3.4.2019].
URL https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

[9] Carlo Tomasi and Roberto Manduchi: Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, s. 839–846.

[10] CH Robotics: *Understanding Euler Angles*. CHRobotics, [Online; navštíveno 16.2.2020].
URL http://www.chrobotics.com/library/understanding-euler-angles

[11] Chen, B.; Deng, W.; Hu, J.: Mixed High-Order Attention Network for Person Re-Identification. 2019, 1908.05819.

[12] Chott, V.: *Přehled bezpilotních prostředků (Drones) v oblasti elektronických komunikací*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 50 s.

[13] Cipolla, R.; Battiato, S.; Farinella, G. M.: *Computer Vision: Detection, Recognition and Reconstruction*. Berlin: Springer, 2010, ISBN 978-3-642-12847-9.

[14] Contributors of TensorFlow repository: *Tensorflow detection model zoo*. 2020, [Online; navštíveno 14.5.2020].
URL https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

[15] D Randall Wilson and Tony Ramon Martinez: *The general inefficiency of batch training for gradient descent learning*. *Neural Networks*, ročník 16, 2003: s. 1429–1451.

[16] Dan Foster: *GPX: the GPS Exchange Format*. Stow (Massachusetts): TopoGrafix, c1998-2018, [Online; navštíveno 18.3.2020].
URL https://www.topografix.com/gpx.asp

[17] Everingham, M.; Gool, L. V.; Williams, C. K. I.; aj.: *The PASCAL Visual Object Classes (VOC) Challenge*. *International Journal of Computer Vision*, ročník 88, č. 2, 2010: s. 303–338.
URL https://doi.org/10.1007/s11263-009-0275-4

[18] Evropský parlament a Rada Evropské unie: *Nařízení Evropského parlamentu a Rady (EU) 2016/679*. 2016, [Online; navštíveno 9.6.2020].
URL https://eur-lex.europa.eu/legal-content/CS/TXT/?uri=uriserv:OJ.L_.2018.212.01.0001.01.CES&toc=OJ:L:2018:212:TOC

[19] Evropský parlament a Rada Evropské unie: *Nařízení Evropského parlamentu a Rady (EU) 2018/1139*. 2018, [Online; navštíveno 9.6.2020].
URL https://eur-lex.europa.eu/legal-content/CS/TXT/?uri=uriserv:OJ.L_.2018.212.01.0001.01.CES&toc=OJ:L:2018:212:TOC

[20] Gong, S.; Cristani, M.; Yan, S.; aj.: *Person Re-Identification*. New York: Springer, 2014, ISBN 978-1-4471-6296-4.

[21] Greg Welch and Gary Bishop: *An Introduction to the Kalman Filter*. University of North Carolina, 2006, [Online; navštíveno 13.1.2020].
URL https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[22] Gustavo Machado: *ML Basics: supervised, unsupervised and reinforcement learning*. Říjen 2016, [Online; navštíveno 1.4.2019].
URL https://medium.com/@machadogj/ml-basics-supervised-unsupervised-and-reinforcement-learning-b18108487c5a

[23] Hou, R.; Ma, B.; Chang, H.; aj.: Interaction-and-Aggregation Network for Person Re-identification. 2019, 1907.08435.

[24] Huang, J.; Rathod, V.; et al., C. S.: Speed/accuracy trade-offs for modern convolutional object detectors. 2016, 1611.10012.

[25] Hájek, L.; Hambálek, T.: *Gyroskopy*. 2010, [Online; navštíveno 18.3.2019].
URL http://fyzsem.fjfi.cvut.cz/2010-2011/Zima10/proc/gyroskopy.pdf

[26] Joseph Redmon and Santosh Divvala and Ross Girshick and Ali Farhadi: *You Only Look Once: Unified, Real-Time Object Detection*. 2016, doi:10.1109/CVPR.2016.91.

[27] Julien Despois: *Memorizing is not learning!* Březen 2018, [Online; navštíveno 3.4.2019].
URL https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42

[28] Kilián, K.: *Čím se LIDAR liší od radaru a jaká je jeho role v autonomních vozidlech.* Říjen 2018, [Online; navštíveno 15.3.2019].
URL https://www.mrk.cz/clanek.php3?id=718

[29] Koustubh Sinhal: *Object Detection using Single Shot Multibox Detector*. 2018, [Online; navštíveno 9.6.2020].
URL https://cv-tricks.com/tag/single-shot-detector

[30] Kuhihiko Fukushima: *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics*, ročník 36, 1980: s. 193–202.

[31] Küffer, F.: *Vše o sonarech.* Říjen 2006, [Online; navštíveno 15.3.2019].
URL https://www.mrk.cz/clanek.php3?id=718

[32] Lauermann, M.: *GPS A BAROMETRICKÁ VÝŠKA: KONEČNÁ ODPOVĚĎ.* Září 2011, [Online; navštíveno 17.3.2019].
URL www.pg-leonardo.cz/index.php?option=com_content&view=article&id=300

[33] Lichtenberg, P.: *BAROMETRICKÉ VÝŠKOMĚRY*. [Online; navštíveno 15.3.2019].
URL http://lichtco.cz/side_vysk.htm

[34] Malfliet, E.: *Implementing the new European drone legislation: what will change?* Květen 2018, [Online; navštíveno 11.1.2019].
URL
www.unifly.aero/news/the-european-drone-legislation-what-will-change

[35] Mišák, P.: *Co je to dron a jaké může mít využití?* Červenec 2015, [Online; navštíveno 25.2.2019].
URL https://www.droni.cz/co-je-to-dron/

[36] Mišák, P.: *Kvadrokoptéra nebo octocoptéra?* Srpen 2016, [Online; navštíveno 25.2.2019].
URL https://www.droni.cz/kvadrokoptera-octocoptera-dron

[37] Novák, J. A.: *Výhody a zrady GPS*. Květen 2016, [Online; navštíveno 20.3.2019].
URL http://www.droneweb.cz/zaciname/item/52-gps-drony

[38] Novák, M.: *Umělé neuronové sítě: teorie a aplikace*. Praha: C. H. Beck, 1998, ISBN 80-7179-132-6.

[39] Olga Russakovsky and Jia Deng and others: *ImageNet Large Scale Visual Recognition Challenge*. *International Journal of Computer Vision*, ročník 115, 2015: s. 211–252.

[40] OpenCV developers team: *Histograms*. 2011-2014, [Online; navštíveno 29.5.2020].
URL https://docs.opencv.org/2.4/modules/imgproc/doc/histograms.html

[41] Přispěvatelé Wikipedie: *Global Positioning System*. 2019, [Online; navštíveno 20.3.2019].
URL https://cs.wikipedia.org/wiki/Global_Positioning_System

[42] Riccio, D.; Marsico, M. D.; Distasi, R.; aj.: A comparison of approaches for person re-identification. 03 2014.

[43] Robert Fisher and Simon Perkins and Ashley Walker and Erik Wolfart.: *Gaussian Smoothing*. 2003, [Online; navštíveno 19.4.2020].
URL https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

[44] Shengcai Liao et al.: Person re-identification by Local Maximal Occurrence representation and metric learning. June 2015, doi:10.1109/CVPR.2015.7298832.

[45] Sochor, M.: *Kvadroptéra a zákon v ČR*. Prosinec 2016, [Online; navštíveno 12.01.2019].
URL www.chytredrony.cz/kvadrokoptery-a-zakony/kvadroptera-a-zakon-v-cr

[46] SPRL, S. P.: *Drone Rules*. 2019, [Online; navštíveno 12.01.2019].
URL http://dronerules.eu

[47] Steiner, I.; Černý, J.: *GPS od A do Z*. Praha: eNav, 2006, ISBN 80-239-7516-1.

[48] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; aj.: *Rethinking the Inception Architecture for Computer Vision*. Červen 2016, doi:10.1109/CVPR.2016.308.

[49] The Robotics Institute of Carnegie Mellon University: *A Computer That Reads Body Language*. 2017, [Online; navštíveno 9.6.2020].
URL https://www.ri.cmu.edu/a-computer-that-reads-body-language

[50] Wei Liu and others: *SSD: Single Shot MultiBox Detector*. Září 2016.

[51] Winkler, C.: *How Many Sensors are in a Drone, And What do they Do?* Červenec 2016, [Online; navštíveno 18.3.2019].
URL https://www.sensorsmag.com/components/how-many-sensors-are-a-drone-and-what-do-they-do

[52] Zhang, X.; Luo, H.; Fan, X.; aj.: AlignedReID: Surpassing Human-Level Performance in Person Re-Identification. 2017, 1711.08184.

[53] Zheng, L.; Shen, L.; Tian, L.; aj.: Scalable Person Re-identification: A Benchmark. 2015.

[54] Zhou, K.; Xiang, T.: Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch. 2019, 1910.10093.

Kapitola 8

Obsah přiloženého media

- Zdrojový kód včetně dat vytvořených datasetů a dalších dat, včetně souboru README
- Tato práce ve formátu PDF v adresáři `\thesis\`
- Zdrojové soubory pro tuto práci ve formátu LaTeXv adresáři `\thesis\latex\`