



Zadání bakalářské práce

Název:	Analýza videozáznamů evakuačních experimentů – zúžení
Student:	Vít Pospíšil
Vedoucí:	Ing. Pavel Hrabák, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Cílem práce je analyzovat záznamy z evakuačního experimentu pořízených v r. 2014 na FJFI (Hrabák et al. 2018). Důraz je kladen na extrakci trajektorií v blízkosti zúžení (východu) a identifikaci jednotlivých účastníků – ti byly vybaveni kontrastními čepičkami s unikátním binárním kódem.

1. Proveďte rešerši vhodných nástrojů pro extrakci trajektorií (multiple-object-tracking, detekce binárních kódů apod.)
2. Prozkoumejte možnosti využití nalezených nástrojů pro analýzu poskytnutých videozáznamů. Pomocí vhodných algoritmů extrahujte trajektorie a přiřadte jim správné ID.
3. Navrhněte a implementujte algoritmus pro detekci neúplných trajektorií a jejich napojení.
4. Výsledky porovnejte s manuálně či polo-automaticky získanými daty (poskytnete školitel).
5. Získaná data použijte pro statistickou analýzu vybrané evakuační charakteristiky.

P. Hrabák, M. Bukáček, M. Krbálek (2018) Transp. A: Transp. Sci. 14(5-6), 375-391.

Bakalářská práce

**ANALÝZA
VIDEOZÁZNAMŮ
EVAKUAČNÍCH
EXPERIMENTŮ –
ZÚŽENÍ**

Vít Pospíšil

Fakulta informačních technologií
Katedra aplikované matematiky
Vedoucí: Ing. Pavel Hrabák, Ph.D.
11. května 2022

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Vít Pospíšil. Odkaz na tuto práci.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci: Pospíšil Vít. *Analýza videozáznamů evakuačních experimentů – zúžení*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratek	ix
Úvod	1
1 Teoretická část	3
1.1 Experiment	3
1.1.1 Videozáznamy	4
1.1.2 Identifikátory osob	4
1.2 Pohyb davu	4
1.3 Metody získání dat z videa	5
1.3.1 Detekce objektů	5
1.4 Metody strojového učení	5
1.5 Neuronové sítě	6
1.5.1 Konvoluční neuronové sítě	6
1.5.2 R-CNN algoritmy	6
1.5.3 YOLO algoritmy	7
1.6 Segmentace obrazu	8
2 Implementační část	9
2.1 Použité technologie	9
2.2 Model neuronové sítě	9
2.2.1 Anotace objektů	9
2.2.2 Faster R-CNN	10
2.2.3 Učení neuronové sítě	10
2.2.4 Optimalizace běhu	10
2.3 Detekce identifikátorů	11
2.4 Oprava neúplných trajektorií	11
2.5 Segmentace kódu	12
2.5.1 Transformace obrazu	12
2.5.2 Čtení kódu	12
2.6 Ukládání dat	13
2.7 Závěr	13
3 Analytická část	15
3.1 Porovnání trajektorií	15
3.1.1 Poskytnutá data	15
3.1.2 Způsob porovnání	15
3.1.3 Výsledky porovnání	15

3.2	Strategie průchodů	17
3.3	Rozestupy	17
3.3.1	Porovnání jednotlivých běhů	17
3.3.2	Porovnání rozestupů chodců	18
4	Závěr	19
	Obsah přiloženého média	23

Seznam obrázků

1.1	Schéma organizovaného experimentu [1]	3
1.2	Identifikátor osoby	4
1.3	Struktura R-CNN [4]	6
1.4	Struktura Fast R-CNN [5]	7
1.5	Struktura Faster R-CNN[6]	7
1.6	Zpracování obrazu algoritmem YOLO [7]	8
2.1	Průběh učení neuronové sítě	10
2.2	Opravená trajektorie	11
2.3	Vlevo: Segmentovaný identifikátor, vpravo: přečtený kód 43 z transformovaného identifikátoru	12
3.1	Porovnání trajektorií chodce s kódem 6 v šestém běhu	16
3.2	Vlevo: porovnání strategií průchodů, vpravo: porovnání nejrychlejšího modrého a nejpomalejšího červeného průchodu	17
3.3	Porovnání časových rozestupů běhu číslo 2 a číslo 3	18

Seznam tabulek

3.1	Porovnání získaných trajektorií s poskytnutými	16
3.2	Porovnání časových odstupů jednotlivých průchodů	18

Děkuji vedoucímu práce Ing. Pavlu Hrabákovi, Ph.D. za jeho rady a trpělivost při tvorbě této bakalářské práce. Dále bych chtěl poděkovat své přítelkyni a rodině, kteří při mně vždy stáli a podporovali mě.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 11. května 2022

.....

Abstrakt

Tato bakalářská práce se zaměřuje na zpracování dat získaných experimentem simulované evakuace osob místem zúžení. Z experimentu byly pořízeny videozáznamy, ze kterých je potřeba extrahovat pomocí neuronových sítí a strojového vidění trajektorie pohybu jedinců a následně je porovnat a analyzovat. Výsledkem práce jsou získané trajektorie chodců, jejich porovnání s trajektoriemi získanými jedním z organizátorů experimentu, Ing. Markem Bukáčkem, a výsledky statistické analýzy trajektorií.

Klíčová slova dynamika davu, zúžení místa průchodu, konvoluční neuronové sítě, strojové vidění, segmentace obrazu, statistická analýza rozestupů, trajektorie pohybu

Abstract

This bachelor thesis focuses on the processing of data obtained by the experiment of simulated evacuation of people through narrowed section. Multiple video recordings were made of the experiment, from which is needed to extract the trajectories of individual's movements using neural networks and computer vision, and then compare and analyze them. The result of the thesis is to obtain trajectories of pedestrians, their comparison with the trajectories obtained by organisator of the experiment, Ing. Marek Bukáček, and to perform statistical analysis of trajectories.

Keywords crowd dynamics, narrowing of the place of passage, convolutional neural network, computer vision, image segmentation, statistical analysis of spacings, movement trajectory

Seznam zkratek

COCO	Common Objects in Context
CUDA	Compute Unified Device Architecture
FPN	Feature Pyramid Network
IoU	Intersection over Union
R-CNN	Regions with Convolution Neural Networks
RoI	Region of Interest
RPN	Region Proposal Network
YOLO	You Only Look Once

Úvod

Pohyb davu těsně před zúžením je nepředvídatelný jev. Každý jedinec zvažuje s každým krokem, kterým směrem se vydá. Tím vzniká nekonečné množství trajektorií. Některé jsou rychlejší, některé pomalejší a některé se vyhýbají nejvíce zhuštěným místům. Takovéto situace nastávají každý den, ať už při nástupu do autobusu, průchodu eskalátory nebo průchodu turnikety. Většina těchto všedních situací má vliv pouze na naši náladu, obzvláště pokud spěcháme. Mnohem horší mohou být následky při náhlé evakuaci osob, kde je cílem zajistit co nejrychlejší průchod všech zúčastněných osob. Tato práce si bere za cíl zpracovat nejčastější trajektorie průchodu zúžením při evakuačním cvičení. Zároveň díky použití identifikačních kódů jednotlivců je možné porovnávat nejen jednotlivé trajektorie, ale i různé průchody jednotlivých chodců.

Analýza trajektorií z tohoto experimentu už byla zpracována jedním z organizátorů experimentu a vyučujícím na fakultě jaderné a fyzikálně inženýrské, Ing. Markem Bukáčkem, v roce 2018 [1]. V této práci se zabývá analýzou rychlosti průchodů účastníků experimentu zúžením. Tuto práci rozšiřuji o popis způsoby získání trajektorií a analýzou délky rozestupů jednotlivých chodců.

V teoretické části této práce se zabývám rešerší problému a technologií pro zpracování videa, získáním trajektorií a přečtením binárních kódů. Zároveň zde uvádím základní teorii technologií a algoritmů, které dále používám v části implementační. V implementační části popisují mnou použité technologie a postup řešení úkolů automatické extrakce trajektorií, přečtení a přiřazení binárního kódu ke správné trajektorii a implementaci algoritmu detekce a opravy neúplných trajektorií. V analytické části získané trajektorie porovnávám s trajektoriemi získanými Ing. Markem Bukáčkem. Dále v této kapitole provádím časovou analýzu strategií průchodu a analýzu časových rozestupů v zúženém prostoru.

Mojí motivací je získání informací ohledně rozhodování lidí v hustém davu, pro získání lepšího přehledu o jeho dynamice, který může vést k plynulejšímu průchodu zúžením. Zároveň je pro mne důležitou motivací i způsob získání těchto strategií. Získávání informací z videozáznamů za pomoci vycvičených neuronových sítí pro relativně přesné sledování pohybu, které se více a více dostává do podvědomí lidí.

Výsledná data lze využít samotném návrhu zúžení nebo informování a moderování davu v případě nutnosti evakuace.

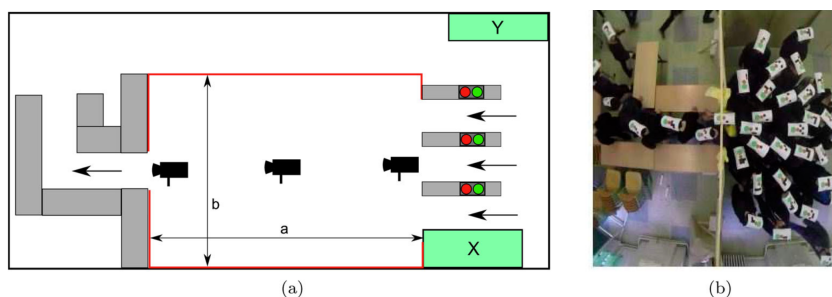
Kapitola 1

Teoretická část

Tato kapitola je zaměřená na řešení jednotlivých problémů, kterými se tato práce zabývá. Cílem teoretické (řešební) části práce je objasnit teoretickou podstatu řešeného tématu, nalézt a porovnat různé nástroje pro zpracování videa, detekce a sledování pohybu objektů, segmentace obrazu a extrakce trajektorií pohybu chodců. Zároveň zde popisují experiment, který mám za úkol zpracovat.

1.1 Experiment

Práce je založena na experimentu [2] z 29. 4. 2014 pořádaného Ing. Markem Bukáčkem, Ing. Pavlem Hrabákem, Ph.D. a doc. Mgr. Milanem Krbálkem, Ph.D. Experimentu se účastnilo 76 dobrovolníků, kteří měli za úkol co nejrychleji projít předpřipravenou obdélníkovou místností (na obrázku 1.1 (a)) o délce 7,2 m a šířce 4,5 m ukončené zúžením o šířce 0,6 m, a poté se znovu vrátit na začátek. Každý chodec prošel místností maximálně 6 krát během jednoho běhu experimentu. Dobrovolníci byli vpouštěni do místnosti v krátkých moderovaných intervalech za sebou, aby se udržel konstantní příchod nových chodců. Každý dobrovolník byl označen papírovou čepičkou obsahující unikátní binární kód pro jednoduché automatizované rozpoznávání jednotlivých chodců. Konec místnosti byl uměle zúžen tak, aby bylo možné procházet pouze v řadě za sebou. Z experimentu byly pořízeny videozáznamy z více kamer. Mým úkolem je zpracovat 11 videozáznamů zachycující místo zúžení (obrázek 1.1 (b)). Každý videozáznam zachycuje jednotlivý běh experimentu.



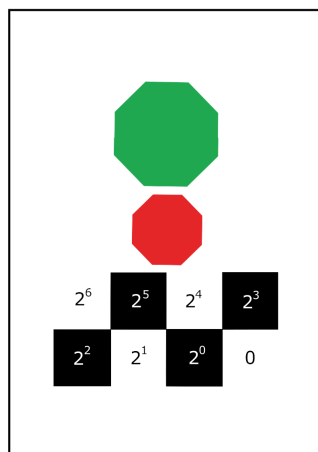
■ Obrázek 1.1 Schéma organizovaného experimentu [1]

1.1.1 Videozáznamy

Poskytnuté videozáznamy jsou pořízeny z kamery umístěné nad místem zúžení pod stropem. Zachycují pohyb účastníků experimentu zhruba 2 metry před zúžením až po jeho konec, kde se dobrovolníci otáčejí a vracejí na začátek experimentu. Videozáznamy jsou pořízeny v rozlišení 1920×1080 px ve 48 snímcích za vteřinu. Vše je pořízeno za umělého konstantního nasvícení místnosti stropními světly. Videozáznamy k zpracování jsou číselně označeny od 1 do 12. První videozáznam neobsahuje žádné události pro zpracování, a proto se zabírám následujícími 11 videozáznamy.

1.1.2 Identifikátory osob

Každý účastník experimentu má na hlavě umístěný papírový identifikátor skládající se z červeného a zeleného osmiúhelníku a 8bitového kódu. Červený menší osmiúhelník se nachází uprostřed papíru, zelený větší osmiúhelník je těsně nad červeným a binární kód je rozmístěn ve dvou řadách a čtyřech sloupcích těsně pod červeným osmiúhelníkem. Binární kód je reprezentován osmi obdélníky. Pokud je obdélník bílý, reprezentuje bit s hodnotou 0, pokud černý, reprezentuje bit s hodnotou 1. Poslední obdélník nacházející se vpravo dole reprezentuje kontrolní bit, který je reprezentován vždy hodnotou 0. Zbytek obdélníků reprezentuje 7bitový identifikátor osoby, která jej nosí. V experimentu byly použity identifikátory s rozmezím hodnot od 1 do 76.



■ Obrázek 1.2 Identifikátor osoby

1.2 Pohyb davu

Pohyb davu je pohyb většího množství lidí ve stejném směru, kdy může docházet k rozdílným rychlostem pohybu vedoucím k neustálé změně uskupení davu. Předbíhání může vést ke zrychlení průchodu jedinců s agresivnější strategií na úkor pasivnějších jedinců. Motivací takového chování často bývá nedostatek času, netrpělivost nebo snaha se vyhnout nekomfortní tlačenci. Dav nejčastěji mění uskupení, pokud narazí na překážku. Takovou překážkou mohou být schody, semafor u přechodu, nečekané zastavení či odbočení některého z chodců, nebo zúžení, do kterého se ve stávajícím stavu dav nevejde. Poslední příklad je simulován zpracovávaným experimentem. V tuto chvíli je třeba dav zeštíhlit. Osoby, které by se v tomto uskupení nevešly do zúženého prostoru, se musejí vtěsnat mezi ostatní. To způsobuje změnu rychlosti pohybu davu. V těchto případech mají chodci možnost vybrat si strategii. Buď se budou snažit zpomalit dav obejít

až k překážce, a tam se zařadit, nebo budou čekat a pomalu se pohybovat s davem. Případně nějakou kombinací agresivity a trpělivosti.

1.3 Metody získání dat z videa

Videozáznamy obsahují opravdu velké množství dat. Obsahují desítky snímků za vteřinu a každý snímek obsahuje milióny pixelů (bodů), které jsou reprezentovány nejčastěji 8 bity pro každou barevnou složku (červená, zelená a modrá). Zpracování takového množství informací vyžaduje velké množství výkonu a procesorového času. Naštěstí pro tato data jsou navrženy grafické karty, které to zvládají mnohonásobně rychleji.

V této práci se zaměřuji na detekování a klasifikaci objektů ve videu a jejich sledování.

1.3.1 Detekce objektů

Video můžeme rozdělit na jednotlivé snímky, ve kterých se můžeme pokusit detekovat očekávaný objekt. Video ale na rozdíl od jednotlivých snímků má výhodu kontinuity. To znamená, že pokud jsme detekovali objekt na jednom snímku, lze očekávat, že se objekt objeví i na dalším snímku v blízkém okolí, kde se nacházel předtím. Toto můžou narušit rychle pohybující se nebo mizející objekty, případně když má video nízkou snímkovou frekvenci.

Klasifikací objektů myslíme to, že detekované objekty umíme od sebe odlišit. Každý klasifikační model využívá alespoň dvou tříd klasifikace. Jedna třída reprezentující pozadí (objekt, který nechceme detekovat ani klasifikovat) a zbylé třídy, které nás zajímají a chceme je odlišovat.

Pro detekci a klasifikaci objektů se používají neuronové sítě s vycvičenými modely. Pro obvykle detekované objekty jako jsou osoby, automobily, zvířata apod. existují předcvičené modely. Pro ostatní je třeba takový model vycvičit. K tomu je zapotřebí vyšší desítky až stovky anotovaných objektů na snímcích, které se budou podsouvat algoritmu učení. Během tohoto procesu probíhá ladění parametrů modelu určených k detekci a klasifikaci. Výsledný model pak dokáže s vysokou pravděpodobností najít a správně klasifikovat hledaný objekt.

1.4 Metody strojového učení

Strojové učení můžeme rozdělit na supervizované a nesupervizované. Hlavní rozdíl je, že supervizované učení využívá anotovaná/označená data v průběhu učení. Díky tomu je vhodné pro detekci a klasifikaci. U nesupervizovaného učení trénovací dataset nevyžaduje manuální označení dat. Užívá se pro seskupování podobných dat. Protože se tato práce zaměřuje na detekci a klasifikaci dat, budeme se dále bavit o supervizovaném učení.

Aby supervizované učení mohlo upravovat hodnoty vnitřních parametrů za vidinou zlepšení predikce, musí vědět úspěšnost predikce v nynějším stavu. Proto je důležité rozdělit anotovaná data na trénovací a testovací, případně ještě validační. Učení probíhá v epochách, během kterých se ladí vnitřní parametry. Po každé epoše proběhne testování na testovacích datech, které nebyly součástí trénování. Tím se ověří, zda nové parametry mají vyšší úspěšnost než parametry předchozí. Po skončení poslední epochy probíhá validace modelu na validační množině anotovaných dat. Výsledná úspěšnost reprezentuje přesnost naučeného modelu.

1.5 Neuronové sítě

Neuronové sítě jsou disciplínou strojového učení a umělé inteligence. Jedná se pokus replikovat schopnosti rozhodování lidského mozku. Skládají se z množiny neuronů uspořádaných ve vrstvách, kde každá následující vrstva přebírá na vstup výstup vrstvy předchozí. Tato síť přijímá na vstupní vrstvě vektor vstupních dat. Po vstupní vrstvě následují vrstvy skryté. Poslední vrstva se nazývá výstupní a vrací výstupní vektor.

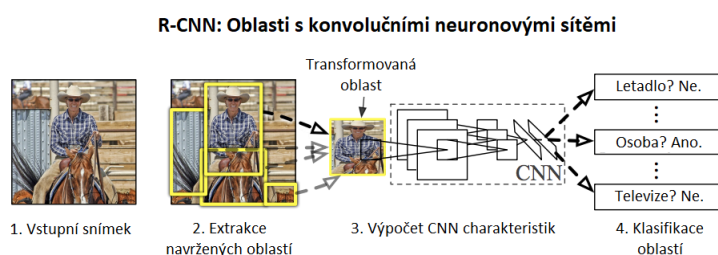
Každý neuron může mít jeden nebo více vstupů z předchozí vrstvy, každý ovlivněný vahou (vnitřním parametrem), díky které je možné některé vlastnosti vstupních dat potlačit a některé posílit, ale pouze jeden výstup. Samotný neuron se pak rozhoduje dle své aktivační funkce.

1.5.1 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou algoritmy používající Deep Learning (hluboké učení). Jsou inspirovány fungováním lidského mozku při zpracovávání vizuálních dat. Dokáží rozpoznat různé tvary, struktury a přiřadit jim různé váhy. Tyto charakteristiky se učí samy ze vstupních dat. Jedná se o podskupinu neuronových sítí skládajících se z konvolučních a poolingových vrstev. Mezitím co konvoluční vrstvy se starají o extrakci vlastností z dat, poolingové vrstvy se starají o redukci dat. Oproti obecným neuronovým sítím jsou ideální pro zpracovávání větších dat jako jsou digitální snímky[3].

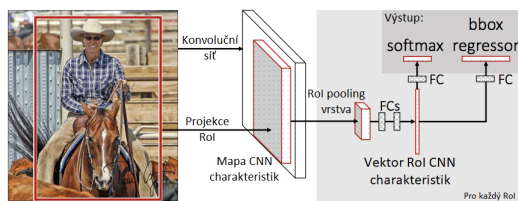
1.5.2 R-CNN algoritmy

Algoritmy používající konvoluční neuronové sítě R-CNN slouží k detekci i klasifikaci objektů z digitálního snímku. Původní algoritmus R-CNN zpracovává každý snímek sekvencí modulů. Nejdříve ho rozdělí do 2000 oblastí, poté každou oblast transformuje na čtverec a výsledek vloží do konvoluční sítě, která vyprodukuje vektor o 4096 dimenzích. Následně je v klasifikátoru ke každému vektoru přiřazena třída.



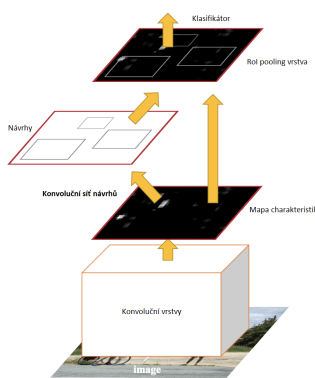
■ **Obrázek 1.3** Struktura R-CNN [4]

Protože původní R-CNN algoritmus nebyl dostatečně rychlý, navrhl stejný autor rychlejší variantu Fast R-CNN. Oproti původnímu algoritmu nerozděluje vstupní snímek na oblasti, ale vkládá ho celý do konvoluční neuronové sítě, která vytvoří konvoluční mapu vlastností. Následně z této mapy identifikujeme oblasti s vyšší pravděpodobností, které předáme pooling vrstvě. Poté lze redukovaná data klasifikovat. Výsledný algoritmus je mnohonásobně rychlejší při zachování vysoké přesnosti.



■ **Obrázek 1.4** Struktura Fast R-CNN [5]

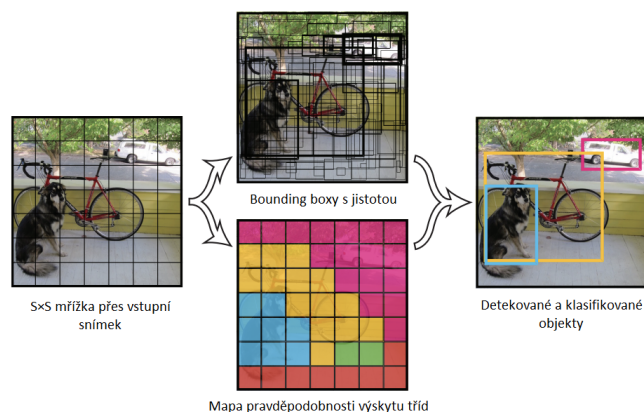
Další etapou vývoje R-CNN algoritmů je znovu rychlejší varianta Faster R-CNN, která eliminuje selektivní vyhledávání v konvoluční mapě a nahrazuje ho separovanou neuronovou sítí určenou čistě k vyhledávání vlastností v mapě. Tato úprava přináší další několikanásobné zrychlení učení, detekce i klasifikace oproti předchozímu algoritmu Fast R-CNN a dovoluje nám ho použít i v případě detekce a klasifikace videa v reálném čase.



■ **Obrázek 1.5** Struktura Faster R-CNN[6]

1.5.3 YOLO algoritmy

Algoritmy ze skupiny YOLO (You Only Look Once) mají za úkol provádět detekci a klasifikaci v reálném čase. Na rozdíl od R-CNN algoritmu se na detekci i klasifikaci podílí pouze jedna konvoluční neuronová síť. Vstupní snímek je nejdříve rozdělen do čtvercové mřížky. Poté jsou pro každý region mřížky určeny oblasti a pravděpodobnosti výskytu objektů. Následně vyfiltrováním oblastí s vyššími pravděpodobnostmi získáme detekované objekty. Největší výhodou YOLO algoritmů je jejich rychlost. Dokáží detekovat objekty v desítkách snímků za vteřinu. Nevýhodou je však to, že mívají problémy s detekcí malých objektů.



■ **Obrázek 1.6** Zpracování obrazu algoritmem YOLO [7]

1.6 Segmentace obrazu

Segmentace obrazu se používá pro hledání specifických kontur, oddělení nebo odfiltrování různých částí obrazu. Získané kontury se poté hodí při počítání objektů, kontrole kvality zboží nebo získání informace o pozici objektu na snímku. Nejjednodušší způsob segmentace je pomocí hraniční hodnoty pixelu. Ta filtruje všechny pixely s hodnotou v rozmezí intervalu hraničních hodnot. Výsledkem je vytvořená binární maska, kde vyfiltrované pixely nabývají nejvyšší hodnoty a zbylé hodnoty nejnižší.

Kontury, oblasti vyznačené maskou, lze dále filtrovat podle tvaru či velikosti.

Implementační část

Cílem implementační části práce je vytvořit funkční algoritmus pro získání trajektorií a binárních kódů pohybujících se objektů. Následně k získaným trajektoriím přiřadit odpovídající binární kód. Pokud některá z trajektorií nebude kompletní, navrhnout algoritmus pro detekci neúplných trajektorií a jejich následné napojení. Získaná data zpracovat a uložit pro následnou analýzu.

2.1 Použité technologie

Z jazyků vhodných na zpracovávání velkých dat a vhodných pro strojové učení byl vybrán Python [8]. Disponuje velkou podporou knihoven pro práci s obrazem, n-rozměrnými maticemi i graficky akcelerovaným strojovým učení. Pro práci s obrazem byla vybrána obsáhlá knihovna OpenCV [9], pro práci s daty knihovna NumPy [10]. Pro implementaci strojového učení knihovna PyTorch [11] spolu s balíčkem pro strojové vidění TorchVision a podporou CUDA pro možnost zpracování pomocí grafických karet. Dále doplněno o implementaci sledování trajektorií algoritmu SORT [12]. Pro anotaci trénovacích dat byl použit program opencv annotation, který je součástí knihovny OpenCV.

2.2 Model neuronové sítě

Aby bylo možné sledovat dobrovolníky ve videu, je potřeba vytvořit model neuronové sítě, která je bude detekovat. Existují různé předučené modely pro detekování osob, ale jelikož dobrovolníci jsou snímáni shora a nosí na hlavě papírový identifikátor, není možné takovéto modely použít. Naštěstí při experimentu měli téměř všichni účastníci oblečené černé oblečení v kontrastu s bílým identifikátorem. Mohl jsem se tedy zaměřit na učení takového modelu, který bude detekovat právě identifikátory.

2.2.1 Anotace objektů

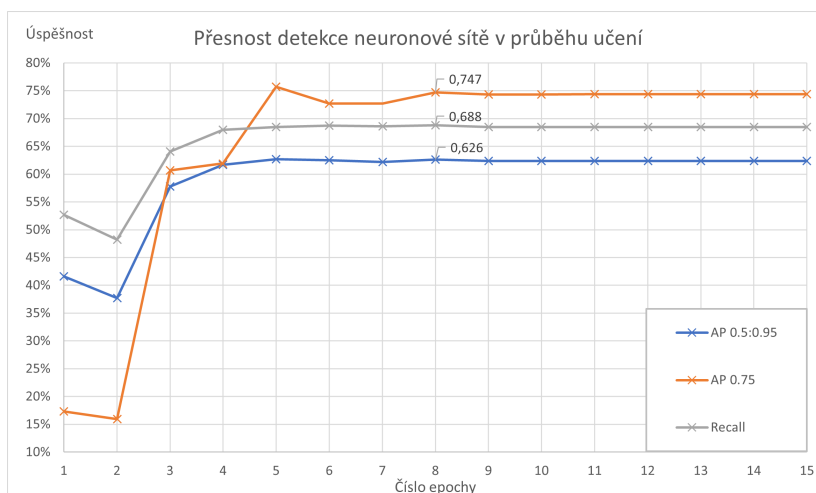
Prvním krokem k supervizovanému naučení modelu je příprava dat k učení. Jediná dostupná data jsou videa ke zpracování, z nichž je zapotřebí získat jednotlivé snímky vhodné pro účel trénování. Snímků jsem vytvořil dohromady 38, kde každý z nich obsahoval až 45 identifikátorů vhodných k učení. Následně jsem na každém snímku manuálně anotoval obdélníkem všechny pozice identifikátorů a uložil je do formátovaného souboru.

2.2.2 Faster R-CNN

Pro detekční model jsem použil předtrénovaný model Faster R-CNN ResNet-50 FPN na obrovské databázi COCO [13]. Jedná se o model skládající se z dvou modulů konvolučních neuronových sítí. První modul poskytuje regiony, kde by se mohl objekt nacházet a druhý modul je Fast R-CNN detektor detekující objekty v poskytnutých regionech. ResNet-50 je konvoluční neuronová síť o hloubce 50 vrstev, která má za úkol vyhledávat charakteristiky v konvolučních mapách. Implementace je poskytnutá knihovnou torchvision.

2.2.3 Učení neuronové sítě

Pro naučení neuronové sítě je třeba anotovaná data rozdělit do skupin. Jelikož dataset obsahuje celé snímky s náhodným počtem výskytů, rozhodl jsem se dataset náhodně zamíchat a ponechat 10 snímků jako testovací dataset. Zbytek snímků byl označen jako trénovací. Trénování probíhalo ve 30 epochách. Po každé epoše byly sledovány 2 metriky. První metrikou je IoU, která udává poměr obsahů průniků odhadnuté a anotované oblasti vůči samotné anotované oblasti. Pro mě byla tato metrika zajímavá ve 2 variantách. Varianta AP 0.75, která počítá procentuální úspěšnost detekcí, kde IoU je větší než 0,75. Druhá varianta AP 0.5:0.95, která počítá průměr úspěšností detekce oblastí pro hodnoty IoU od 0,5 do 0,95. Druhou metrikou, která nás zajímá je Recall. Recall udává poměr počtu správně detekovaných objektů vůči počtu všech anotovaných objektů na snímku. Po 30 epochách učení lze vidět na grafu 2.1, že nejvyšší úspěšnosti jsme dosáhli rychle, a po 10 epochách se metriky už vůbec neměnily. Rozhodl jsem se tedy použít váhy z 8. epochy, kde je metrika Recall nejvyšší a IoU 0.75 druhá nejvyšší dosažená.



■ Obrázek 2.1 Průběh učení neuronové sítě

2.2.4 Optimalizace běhu

Učení i detekce objektů ve videozáznamu může trvat hodiny, když se nepoužije správná implementace a správný hardware. Jedním z důvodů, proč jsem vybral knihovnu PyTorch, byl příslib podpory běhu na grafických kartách AMD. Bohužel tato podpora je stále v experimentálním stavu a nebyl jsem schopen běh na mé grafické kartě zprovoznit. Naštěstí se mi podařilo získat grafickou kartu s podporou CUDA, díky které jsem byl schopen několikanásobně urychlit proces učení i detekce. Proces učení 10 epoch zabral na 8 vláknech procesoru téměř 6 hodin. Oproti tomu stejný proces na 3840 CUDA jádrech zabral pouze 6 minut.

2.3 Detekce identifikátorů

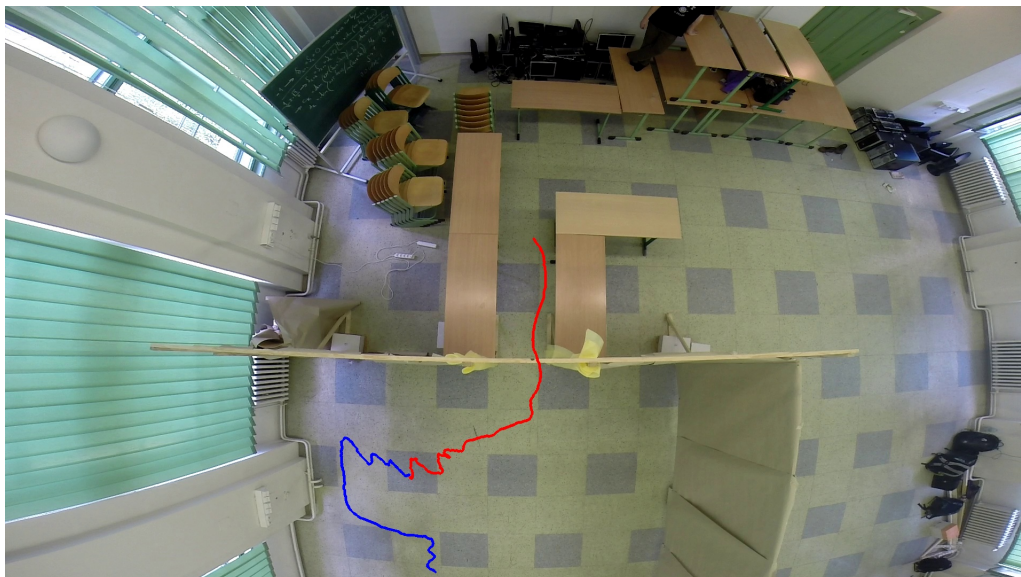
Pro detekci identifikátorů byla použita mnou naučená neuronová síť. Detekování identifikátorů probíhá v každém snímku videa zvlášť. Jelikož v záběru kamery je vidět i cesta dobrovolníků zpět, vytvořil jsem si masku, která mi dovoluje se zaměřit pouze na chodce před a v místě zúžení. Do neuronové sítě poté předávám maskou vyfiltrovaný snímek. Výsledkem je množina predikcí, kde ke každé predikci je přiřazena její jistota. Z této množiny jsou odstraněny predikce s jistotou menší 80 %. Výsledky predikce každého snímku jsou předány algoritmu SORT [12], který má za úkol sledovat každý detekovaný objekt napříč snímky. To mi dovoluje vytvořit ucelené trajektorie. Jako úplné považuji trajektorie, které začínají na začátku záběru kamery a končí v místě zúžení. Všechny ostatní trajektorie je potřeba opravit.

2.4 Oprava neúplných trajektorií

Někdy se stane, že se některé identifikátory překryjí, ztratí z dohledu kamery, nebo se chodec otočí tak, že identifikátor není rozpoznatelný neuronovou sítí. Pokud se trajektorie rozdělí vlivem krátkého zastínění, vzniknou 2 na sebe navazující trajektorie, které lze spojit.

Hledání dvojic trajektorií začíná vyhledáním trajektorií, které nezačínají ve vstupní oblasti a trajektorií, které nekončí v cílové oblasti. Některé se mohou vyskytovat na obou seznamech, pokud se sledování přerušilo vícekrát. Následně je spočítána metrika vzdálenosti mezi posledním bodem každé nekončící trajektorie a prvním bodem každé nezačínající trajektorie. Kandidáti s nejmenší hodnotou metriky nepřekračující určitou mez jsou spojeni. Toto se opakuje, dokud existují trajektorie ke spojení.

Dále kontroluji, zda trajektorie nezačíná v blízkosti startu. V tomto případě doplním trajektorii o nejbližší bod nacházející se ve vstupní oblasti. Shodně opravuji trajektorie končící v blízkosti cíle.



■ Obrázek 2.2 Opravená trajektorie

2.5 Segmentace kódu

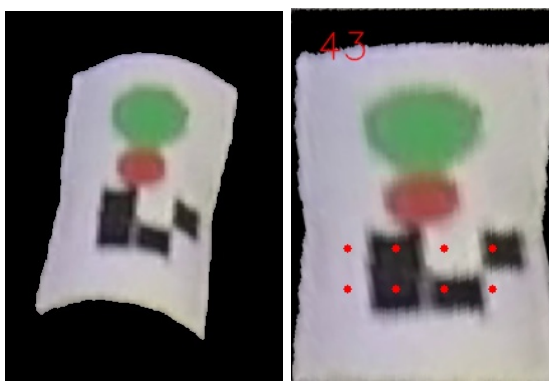
Pokaždé, když se identifikátor nachází přímo pod kamerou v místě zúžení, je předán k přečtení binárního kódu. Jelikož jsou identifikátory vyrobeny z papíru a umístěny na hlavě, jsou deformovány. Aby bylo možné úspěšně číst, je třeba je nejdříve transformovat do roviny a určit správnou orientaci. Pro lepší výsledky segmentace i jednodušší vizuální kontrolu kontur jsem výřez určený k segmentaci nejdříve interpoloval na snímek 4 krát větší a následně rozmazal Gaussovým filtrem za účelem snížení šumu na snímku.

2.5.1 Transformace obrazu

Aby bylo možné transformovat papír do roviny, je třeba nejdříve přesně určit, kde se papír nachází. Jelikož jsou kódy vytisknuté na bílém papíře, lze ho jednoduše segmentovat pomocí prahování. Tím jsem získal konturu papíru, okolo které jsem spočítal bounding box (obdélník s co nejmenším obsahem, ve kterém se nachází zmiňovaná celá kontura). Pro získání rohů kontury jsem zkoušel použít algoritmus detekce rohů, ale kvůli nedostatečnému rozlišení zpracovávaného výřezu jsem nebyl schopen spolehlivě získat správné body. Proto jsem se spokojil s rohy bounding boxu. Podle velikostí stran bounding boxu mapuji vrcholy na rohy papíru. Pro zbavení se zaoblení papíru, způsobeného nošením na hlavě, používám zarovnání ke spodní hraně výřezu. Toho jsem docílil rotací jednotlivých sloupců obrazu, dokud spodní pixel nepřekročil prahovou hodnotu papíru.

2.5.2 Čtení kódu

Pro čtení binárního kódu je zapotřebí nejdříve určit, kde by se měl na papíře nacházet. Toho jsem dosáhl odhadnutím pozice středu kódu ve směru od zeleného osmiúhelníku k červenému osmiúhelníku. Středů osmiúhelníků jsem získal barevnou segmentací snímku a následně odhadem těžiště kontury. Ze získaných středů jsem vytvořil vektor a naškáloval ho tak, aby ukazoval do středu binárního kódu. Dále jsem určil normálový vektor těchto bodů a lineárními kombinacemi získaných vektorů odhadl, kde by se měly nacházet bílé a černé obdélníky reprezentující jednotlivé bity kódu. V získaných bodech jsem poté určil, zda je oblast reprezentována bílou nebo černou barvou. Z prvních 7 bitů jsem složil číslo reprezentující kód, a pokud byl poslední bit reprezentován bílou barvou, označil jsem získaný kód za validní.



■ **Obrázek 2.3** Vlevo: Segmentovaný identifikátor, vpravo: přečtený kód 43 z transformovaného identifikátoru

2.6 Ukládání dat

Každá získaná trajektorie má vlastní identifikační číslo poskytnuté algoritmem SORT. Tato čísla nekorespondují s binárním kódem chodce, proto jsem si při každém úspěšném přečtení binárního kódu zapamatoval záznam obsahující identifikační číslo trajektorie a přečtený kód. Jelikož ten samý kód může být přečtený vícekrát, ukládám všechny úspěšné pokusy, z kterých následně vybírám ten nejčastěji obsažený. Toto umožňuje ignorovat některé špatně přečtené kódy. Pro každý zpracovaný videozáznam poté ukládám 2 soubory. V prvním souboru jsou všechny nalezené trajektorie s vlastním identifikačním číslem. Každá trajektorie je seznam bodů určených časem (číslem snímku) a souřadnicemi korespondující s pixelem. Druhý soubor obsahuje dvojice čísel. První číslo reprezentuje identifikační číslo trajektorie a druhé je k němu přečtený kód z identifikátoru.

2.7 Závěr

Vytvořil jsem program, který je schopen zpracovat videozáznam a extrahovat z něj trajektorie pohybu chodců a přiřadit je ke správnému identifikačnímu kódu dobrovolníka. Toho jsem docílil za pomoci neuronové sítě a metod zpracování obrazu. Implementoval jsem algoritmus pro detekci neúplných trajektorií i pro jejich opravu. Získaná data jsem uložil do strukturovaných souborů vhodných pro další zpracování.

Celý proces zpracování jednoho 4 minutového videa trvá zhruba 30 minut.

Analytická část

Cílem analytické části práce je porovnat získané trajektorie s trajektoriemi získanými Ing. Markem Bukáčkem [1]. Zanalyzovat nejčastější a nejzajímavější strategie průchodu. Porovnat rychlosti průchodů jednotlivých strategií. Provést statistickou analýzu velikosti rozestupů jednotlivců v davu.

3.1 Porovnání trajektorií

Jedním z úkolů této práce je porovnání získaných trajektorií s daty získanými Ing. Markem Bukáčkem, který trajektorie extrahoval v roce 2018 [2] programem napsaným v jazyce MATLAB.

3.1.1 Poskytnutá data

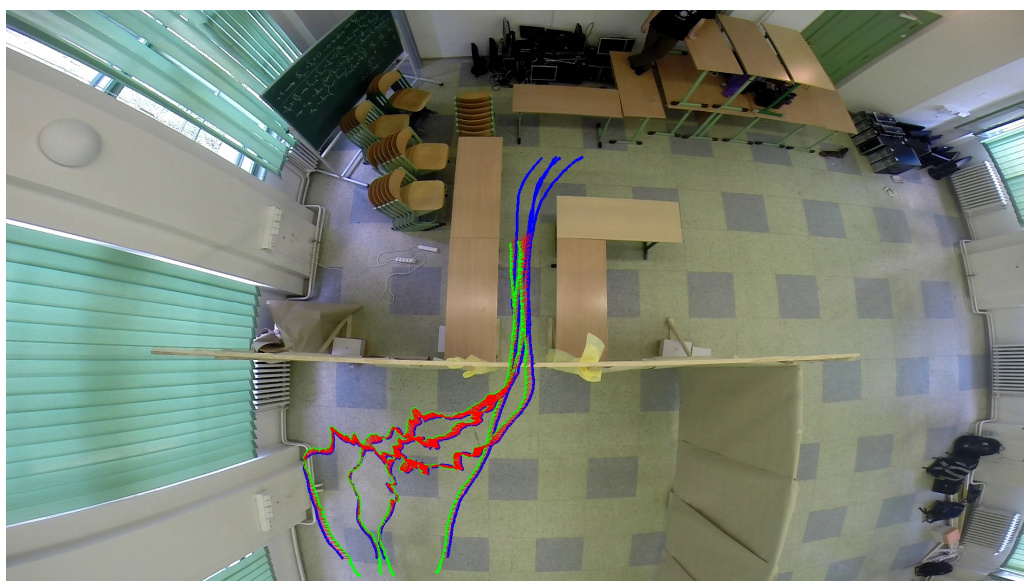
Data mi byla poskytnuta vedoucím bakalářské práce Ing. Pavlem Hrabákem, Ph.D. ve formátu csv. Soubor obsahoval trajektorie přiřazené k číslu běhu (číslo videozáznamu) a kódu chodce. Jednotlivé trajektorie byly reprezentovány sekvencí bodů. Každý bod byl určen časem v sekundách korespondujícím s videozáznamem a souřadnicemi. Bohužel souřadnice nekorespondovaly s obrazem, a proto jsem je musel přeškálovat a posunout.

3.1.2 Způsob porovnání

Porovnávám pouze trajektorie, které jsou celé a určené (mají přiřazený kód chodce). Určení, které 2 trajektorie se mají porovnat, vychází z kódu chodce a z času průchodu. Správně nalezené trajektorie jsou poté porovnány na základě součtu vzdáleností bodů nacházejících se ve stejném snímku a následně vydělené jejich počtem, $\Delta p = \sum_{i=1}^l L_2(p1_i, p2_i)/l$. Tímto získávám průměrnou odchylku trajektorií v pixelech. Na obrázku 3.1 lze vidět mnou získané zelené trajektorie, poskytnuté modré trajektorie a červeně vyznačené rozdíly trajektorií.

3.1.3 Výsledky porovnání

Při porovnávání výsledků jsem si všiml, že v mnoha případech mnou získané data obsahují jiný počet trajektorií, častěji vyšší, jednotlivých chodců, než poskytnutá data. Dokonce trajektorie posledního běhu úplně chybí. Při vizuální kontrole jsem zjistil, že mnou získané trajektorie existují, ale pouze nebyly obsaženy v mně poskytnutém souboru. Porovnávání jsem tedy omezil pouze na data obsažené v obou datasetech.



■ **Obrázek 3.1** Porovnání trajektorií chodce s kódem 6 v šestém běhu

Změřil jsem odchylku pro každý běh zvlášť a zapsal do tabulky 3.1. Průměrná odchylka datasetů činí 11,7 pixelu, což je do značné míry ovlivněno mnou odhadnutými parametry pro přeskálování a posunutí poskytnutých dat. I přes to, že u některých trajektorií byl špatně přečten nebo vůbec nepřečten kód, lze z průniku množin trajektorií označit algoritmus pro získávání trajektorií za funkční.

Průchod	Počet získaných trajektorií	Počet poskytnutých trajektorií	Počet porovnávaných trajektorií	Průměrná odchylka
2	192	158	153	11,572 px
3	287	270	256	11,648 px
4	307	291	272	11,382 px
5	263	258	253	11,361 px
6	292	268	240	12,273 px
7	290	259	230	12,320 px
8	166	140	126	11,154 px
9	162	148	125	11,253 px
10	151	143	122	10,888 px
11	219	140	181	12,273 px
12	124	0	0	—
Průměrná odchylka získaných trajektorií				11,677 px

■ **Tabulka 3.1** Porovnání získaných trajektorií s poskytnutými

3.2 Strategie průchodů

Získané trajektorie lze zanalyzovat z pohledu strategie. Efektivita průchodu se přímo odráží na čase stráveném v prostoru davu. Samozřejmě nejrychlejší průchod místností bude ve chvíli, kdy je prázdná a žádný dav není vytvořený. Proto musíme porovnávat průchody z doby, kdy už je před zúžením dav vytvořen.

Z množiny trajektorií jednotlivých průchodů jsem si nejdříve vyfiltroval pouze úplné trajektorie. Z těch jsem poté vybral ty, které začínají v čase, kdy už je dav vytvořený. Poté jsem je seřadil dle doby průchodu. Následná vizualizace vlevo na obrázku 3.2 zobrazuje porovnání 20 nejrychlejších oproti 20 nejpomalejším trajektoriím z videozáznamu číslo 4. Jak lze vidět, rychlejší a agresivnější chodci vybírají častěji trasu okolo davu (vizualizována modře), zatímco chodci vybírající cestu středem stráví delší dobu posouváním se s davem (vizualizováno červeně). Dokonce, pokud porovnáám nejrychlejší průchod s nejpomalejším průchodem (obrázek 3.2 vpravo), zjistím, že nejrychlejší průchod trval pouze 3,5 s oproti 30 s u nejpomalejšího průchodu.



■ **Obrázek 3.2** Vlevo: porovnání strategií průchodů, vpravo: porovnání nejrychlejšího modrého a nejpomalejšího červeného průchodu

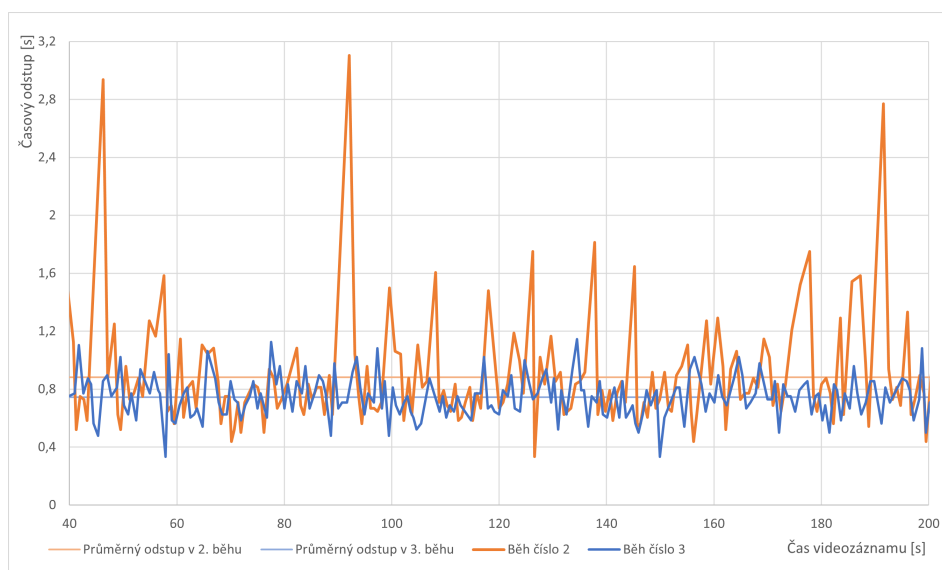
3.3 Rozestupy

Po naměření časových rozestupů při průchodu zúžením, je snadné dopočítat propustnost tohoto zúžení. To je velmi důležitý údaj pro návrh tras, kde se pohybuje velké množství lidí. Správným návrhem zúžení lze poté regulovat počet lidí v prostoru za ním.

Pro analýzu rozestupů z videozáznamů jsem využil získaná data trajektorií. K těm jsem připojil informaci o prvním vstupu do cílové oblasti. Trajektorie jsem poté seřadil dle času průchodu zúžením a extrahoval informaci o časovém odstupu od předchozí trajektorie.

3.3.1 Porovnání jednotlivých běhů

Pro porovnání jsem si připravil vizualizaci 3.3 dvou odlišných průchodů místností. Průchod zachycený na videozáznamu číslo 2 je plynulý a netvoří se tak dav před zúžením. To dovoluje chodcům zařadit se do fronty dříve před zúžením a nemusí kvůli tomu tolik zpomalit chůzi. Naopak při průchodu číslo 3 se dav začíná tvořit rychle a chodci se více přetlačují o prioritu průchodu. To pozitivně ovlivňuje rychlost průchodu zúžením, ale naopak znemožňuje vcházet do místa zúžení plynule. V grafu 3.3 lze vidět, že při průchodu číslo 2 jsou časové odstupy méně stabilní a lze tak vyčíst časy, kdy bylo místo zúžení volné. Naopak průběh číslo 3 místo zúžení saturoval velmi dobře. Nejčastější časový odstup byl 0,75 sekundy, což odpovídá 80 průchodům za minutu. Z tabulky 3.2 lze vidět, že časové úseky, kdy dav před zúžením nebyl, se nacházejí i ve videozáznamech číslo 5 a 12.



■ **Obrázek 3.3** Porovnání časových rozestupů běhu číslo 2 a číslo 3

Průchod	Průměrný odstup	Medián odstupu
2	0,882 s	0,792 s
3	0,745 s	0,75 s
4	0,767 s	0,75 s
5	0,828 s	0,791 s
6	0,784 s	0,75 s
7	0,724 s	0,729 s
8	0,728 s	0,729 s
9	0,721 s	0,729 s
10	0,715 s	0,688 s
11	0,767 s	0,688 s
12	0,878 s	0,802 s

■ **Tabulka 3.2** Porovnání časových odstupů jednotlivých průchodů

3.3.2 Porovnání rozestupů chodců

Cílem tohoto porovnání je najít souvislost mezi délkou odstupu od předchozího chodce a mezi rychlostí průchodu, případně volbou strategie průchodu. Pro porovnání jsem si zpracoval seznam chodců a jejich obvyklé časové odstupy a seřadil je. Nejkratší odstupy měl chodec s kódem 37 a to 0,645 s, což odpovídá 93 průchodům za minutu. Naopak nejdelší odstup si pravidelně držel chodec s kódem 68 a to 1,087 s. To odpovídá 55 průchodům za minutu.

Dále mě zajímalo, zda existuje souvislost mezi velikostí rozestupu při průchodu zúžením a mezi agresivitou chodce a jeho rychlostí průchodu místností. Bohužel se mi nepodařilo najít data, která by tuto souvislost potvrdila. Při bližším vizuálním zkoumání videozáznamů jsem došel k závěru, že ačkoliv se chodci, co dav obcházejí, pohybují rychleji, v místě zúžení se často zaseknou a přetlačují s jiným chodcem, což náhodně prodlužuje časový odstup.



Kapitola 4

Závěr

Cílem práce bylo vytvořit funkční algoritmus pro detekci dobrovolníků účastnících se experimentu evakuace a automaticky extrahovat jejich trajektorie. Toho jsem docílil naučením neuronové sítě detekovat identifikátory dobrovolníků. Rovněž jsem vyřešil problém kontinuálního sledování jednotlivců a napojování jejich neúplných trajektorií. V místě zúžení jsem transformoval detekovaný identifikátor do strojově čitelného tvaru, přečetl a přiřadil identifikační kód k chodci. Po zpracování všech videozáznamů jsou získaná data připravena ke statistické analýze rozestupů chodců v místě zúžení a porovnání získaných dat s daty získanými Ing. Markem Bukáčkem. Porovnáním jsem si ověřil funkčnost a přesnost mého způsobu získávání trajektorií. Dále jsem získal časové rozestupy při průchodu zúženým prostorem. Pomocí těchto rozestupů se dá identifikovat, zda se před zúžením vytvořil dav. Podle získaných trajektorií jsem porovnal nejrychlejší průchody místností s těmi nejpomalejšími. Z tohoto porovnání jsem zjistil, že nejrychlejší průchod davu před zúžením je snažit se ho obejít a zařadit se co nejbližší vstupu do zúžení. Tento styl průchodu může být až 10× rychlejší, než se zařadit na konec davu a pomalu se pohybovat vpřed. Bohužel se mi nepodařilo najít souvislost mezi agresivitou chodce a časovým rozestupem, který daný chodec nechává před sebou.

Přínosem bakalářské práce je získání informací o pohybu a chování lidí v davu při průchodu zúženým prostorem. Lze využít při plánování konstrukcí evakuačních prostor, nutných překážek v místech průchodu větších davů či při výcviku záchranných složek řešících evakuace objektů. Dále by se na tuto práci dalo navázat dalšími analýzami nebo zkoumáním psychologických podnětů jedinců vyskytujících se v hustém davu.

Bibliografie

1. BUKÁČEK, Marek; HRABÁK, Pavel; KRBÁLEK, Milan. Microscopic travel-time analysis of bottleneck experiments. *Transportmetrica A: transport science*. 2018, roč. 14, č. 5-6, s. 375–391. Dostupné z DOI: 10.1080/23249935.2017.1419423.
2. BUKÁČEK, Marek; HRABÁK, Pavel; KRBÁLEK, Milan. Experimental study of phase transition in pedestrian flow. *Transportation Research Procedia*. 2014, roč. 2, s. 105–113. Dostupné z DOI: 10.1016/j.trpro.2014.09.014.
3. SAHA, Sumit. *A comprehensive guide to Convolutional Neural Networks-the eli5 way* [online]. Towards Data Science, 2018 [cit. 2022-04-20]. Dostupné z: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
4. GIRSHICK, Ross; DONAHUE, Jeff; DARRELL, Trevor; MALIK, Jitendra. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014. Dostupné také z: https://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf.
5. GIRSHICK, Ross. Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015. Dostupné také z: https://openaccess.thecvf.com/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf.
6. REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross; SUN, Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015, roč. 28. Dostupné také z: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>.
7. REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 779–788. Dostupné také z: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf.
8. *Python 3.10.4 documentation* [online]. [B.r.] [cit. 2022-04-10]. Dostupné z: <https://docs.python.org/3/>.
9. *Open Source Computer Vision* [online]. [B.r.] [cit. 2022-04-10]. Dostupné z: <https://docs.opencv.org/>.
10. *NumPy* [online]. [B.r.] [cit. 2022-04-10]. Dostupné z: <https://numpy.org/>.
11. *Pytorch* [online]. [B.r.] [cit. 2022-04-10]. Dostupné z: <https://pytorch.org/>.

12. BEWLEY, Alex; GE, Zongyuan; OTT, Lionel; RAMOS, Fabio; UPCROFT, Ben. Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, s. 3464–3468. Dostupné z DOI: 10.1109/ICIP.2016.7533003.
13. LIN, Tsung-Yi; MAIRE, Michael; BELONGIE, Serge; HAYS, James; PERONA, Pietro; RAMANAN, Deva; DOLLÁR, Piotr; ZITNICK, C Lawrence. Microsoft coco: Common objects in context. In: *European conference on computer vision*. 2014, s. 740–755. Dostupné z DOI: 10.1007/978-3-319-10602-1_48.

Obsah přiloženého média

readme.txt	stručný popis obsahu média
src	
├─ impl	zdrojové kódy implementace
└─ thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
└─ thesis.pdf	text práce ve formátu PDF
data	
├─ received	adresář s poskytnutými daty
└─ acquired	adresář se získanými daty
video	adresář s poskytnutými videozáznamy